Kommunikation & Recht

K&R
10 | Oktober 2023
26. Jahrgang
Seiten 633 - 704

dfv Mediengruppe
Frankfurt am Main

VI. Resümee

Die §§ 95a ff. UrhG räumen technischen Maßnahmen einen weitreichenden urheberrechtlichen Schutz ein. Gleichwohl begründen technische Maßnahmen im Kontext von Verbraucherverträgen über die Bereitstellung digitaler Inhalte häufig einen Produktmangel i. S. v. § 327e BGB, sofern nicht eine Vereinbarung gem. § 327h BGB getroffen wurde. Technische Maßnahmen wirken sich regelmäßig dann mangelbegründend aus, wenn sie dem Verbraucher die Ausübung der urheberrechtlichen Schrankenbestimmungen verwehren oder die Nutzbarkeit durch Nebenfolgen einschränken. Teilweise kann schon das bloße Vorhandensein technischer Maßnahmen einen Mangel bedeuten. Die Nacherfüllung i. S. v. § 327l BGB kann insbesondere durch Abänderung oder Entfernung der technischen Maßnahme oder erneute Bereitstellung des digitalen Inhalts ohne die technische Maßnahme erfolgen. Liegt der Mangel in der Verhinderung der Schrankenausübung, können zur Bestimmung der zulässigen Nacherfüllungsweise zudem die im Rahmen des § 95b UrhG (weiteren) anerkannten Mittel Orientierung bieten.

Mittelbar wird durch die DIRL bzw. §§ 327 ff. BGB eine urheberrechtliche Fehlentwicklung – zumindest für die breite Masse der Verbraucherverträge über digitale Inhalte – korrigiert: Das Urheberrecht gestattet *de lege lata*, dass technische Maßnahmen die Schrankenbestimmungen weitestgehend außer Kraft setzen. Dies birgt die Gefahr, dass der fein austarierte Interessenausgleich des Urheberrechts aus dem Gleichgewicht gerät.[66] Indem nun solche technischen Maßnahmen, die die Ausübung der Schranken unterbinden, häufig einen Mangel des digitalen Inhalts begründen, wird die Einsetzbarkeit derartiger technischer Maßnahmen eingedämmt.[67] Nicht zuletzt durch Instrumente wie § 2 Abs. 1 und 2 S. 1 Nr. 1 lit. c UKlaG, die Musterfeststellungsklage sowie die kommende Abhilfeklage wird dieses Ergebnis keine Theorie bleiben, sondern kann effektiv durchgesetzt werden.

Johannes Nowesky
Jahrgang 2000; seit dem Wintersemester 2019/2020 Studium der Rechtswissenschaft an der Universität Bayreuth, dort Belegung des Schwerpunktes „Märkte der digitalen Welt"; seit Oktober 2021 studentische Hilfskraft am Institut für deutsches und europäisches Verwaltungsrecht an der Ruprecht-Karls-Universität Heidelberg (Prof. Dr. Dr. h. c. Wolfgang Kahl, M.A.).

66 *Specht* (Fn. 34), S. 179 ff., 214; *Specht*, GRUR 2019, 253, 258 f.; vgl. ferner *Schack*, ZUM 2002, 497, 510.
67 Dies bezweifelnd *Specht* (Fn. 34), S. 406 f.; *Specht*, GRUR 2019, 253, 259.

RA Dr. Florian Deusch und Prof. Dr. Tobias Eggendorfer*

Strafbarkeit von IT-Sicherheitsforschern und Pentestern

Zugleich Kommentar zu LG Aachen, Beschluss vom 27. 7. 2023 – 60 Qs 16/23, K&R 2023, 693 ff. (Heft 10)

Kurz und Knapp

Der Beitrag untersucht die Strafbarkeit nach dem „Hackerparagrafen" § 202a StGB,[1] die sich bei der Prüfung unsicherer Softwareanwendungen ergeben kann, anlässlich eines Beschlusses des LG Aachen (Seite 693 ff. in diesem Heft). Praktisch relevant ist das für IT-Sicherheitsforscher, Penetrationtester, IT-Anbieter und -Anwender. Durch Befragung des Angeschuldigten konnten die Autoren dessen technische Darstellungen mit aufnehmen, die plausibler erscheinen als der lückenhafte Sachverhalt des Landgerichts.

I. Problemaufriss

Das AG Jülich[2] hatte über einen Strafbefehlsantrag zu entscheiden, der den angeschuldigten IT-Experten dafür bestrafen sollte, dass er sich anlässlich einer beauftragten Softwareprüfung Zugang zu einem Datenbanksystem eines IT-Anbieters verschafft habe. Es lehnte den Strafbefehl ab, weil das Datenbanksystem nach dessen Beurteilung nicht „besonders gesichert" i. S. d. § 202a StGB war. Das LG Aachen hielt eine Sicherung aber für gegeben, hob den Ablehnungsbeschluss auf und verwies die Sache zur erneuten Entscheidung an das AG zurück.

Laut Beschluss des LG Aachen habe der Angeschuldigte ein Programm des anzeigeerstattenden IT-Anbieters M „decompiliert" und so ein Passwort aus dem Programm entnommen. Mit diesem Passwort habe der Beschuldigte dann Zugangsdaten[3] und Inhalte der Datenbank ausgelesen und Kundendaten auf seinen eigenen Computer kopiert.

* Mehr über die Autoren erfahren Sie am Ende des Beitrags.
1 Alle Links im Betrag zuletzt abgerufen 11. 9. 2023. Ausgehend von der Bezeichnung des § 202c StGB als „Hackerparagraf" werden die §§ 202a und 202b bisweilen ebenso benannt, siehe z. B. *Biselli*, Sicherheit für die Sicherheitsforschung, https://netzpolitik.org/2022/hackerparagrafen-sicherheit-fuer-die-sicherheitsforschung/. Andere Straftatbestände, insbesondere §§ 202b, c StGB, 303a, b StGB und § 23 GeschGehG sind ebenfalls relevant, bleiben aber einer gesonderten Betrachtung vorbehalten.
2 AG Jülich, 10. 5. 2023 – 17 Cs 230 Js 99/21-55/23.
3 Die Ausführungen des LG Aachen lassen hier schon Fragen aufkommen: Der Angeschuldigte soll das Passwort für die Datenbank ausgelesen haben, also die Zugangsdaten der Datenbank, um damit die Zugangsdaten der Datenbank zu erhalten. Wie genau sich das LG Aachen diesen Vorgang technisch vorstellt, bleibt offen.

Dieser Sachverhalt im Beschluss des LG Aachen klingt aus technischer Sicht nur eingeschränkt nachvollziehbar und lässt entscheidungserhebliche Fragen offen.

Deshalb stellt Abschnitt II den Sachverhalt dar, soweit ihn die Autoren aus dem Beschluss und den Darstellungen des Angeschuldigten entnehmen konnten. Das zeigt auch, wie herausfordernd es sein kann, IT-bezogene Sachverhalte als Grundlage juristischer Entscheidungen aufzubereiten und zu untersuchen. Abschnitt III bewertet den herausgearbeiteten Sachverhalt strafrechtlich; Abschnitt IV fasst die Ergebnisse zusammen.

II. Entscheidungserheblicher Sachverhalt und Herausforderungen seiner Feststellung

1. Feststellungen des Gerichts und Einlassungen des Angeschuldigten

Laut dem Sachverhalt im Beschluss des LG Aachen bietet das IT-Unternehmen M ein Warenwirtschaftssystem für Online-Händler an und stellt dazu eine Schnittstelle für die Kundensysteme bereit. M verwalte die Daten von ca. 600 000 - 700 000 Endkunden der Händler.

Laut den ergänzenden Darstellungen des Angeschuldigten betreibe M auf seinem IT-System eine Datenbanksoftware. Diese sei dazu bestimmt, dass Online-Händler ihre Warenbestände an die Online-Bestellungen ihrer Kunden anpassen. Damit die Datenbanksoftware auf dem System des M mit den Warenwirtschaftssystemen der Kunden über die vom LG Aachen bezeichnete Schnittstelle kommunizieren kann, stellt M seinen Kunden ein compiliertes Programm zur Verfügung, das die Kunden auf ihren Rechnern zu installieren haben.[4]

Laut LG Aachen habe der Angeschuldigte mittels eines Decompilers aus der von M erstellten Software einen Quellcode erzeugt. Diesem Quellcode habe er das dort im Klartext gespeicherte Passwort entnommen. Mit diesem Passwort habe er die Zugangsdaten zu den jeweiligen Kundendatenbanken ausgelesen und diese auf seinen Computer kopiert.

Nach den weiteren Darstellungen des Angeschuldigten hat ihn ein Kunde K des M beauftragt, die lokale Software des M bei K zum Laufen zu bringen. Der Angeschuldigte stellte dabei einen Softwarefehler fest. Im Folgenden disassemblierte[5] er Teile der Software, um den Fehler durch gezielte Modifikationen zu beheben.

Dieser Vorgang ist allerdings zunächst unabhängig von der Kenntnisnahme des verfahrensgegenständlichen Passworts. Denn vom Passwort hat der Angeschuldigte erst Kenntnis erhalten, nachdem er die Software zur Fehlerbehebung modifiziert und wieder auf dem lokalen System zum Laufen gebracht hat. Danach beobachtete er routinemäßig den Netzwerkverkehr und stellte dabei fest, dass sich die Software zum Datenbankserver von M verbindet und die notwendigen Verbindungsdaten (Rechnername und Datenbankname) im Klartext überträgt. Vom Passwort hat er nur einen Hash[6] gesehen. Zur Kontrolle, ob das Passwort tatsächlich als Klartext im Programm steht, suchte der Angeschuldigte mit einem Programm wie „Strings“, das z. B. auf Linux- oder Mac-Systemen standardmäßig mitgeliefert wird und auch kostenlos als Open-Source zum Download bereitsteht,[7] nach allen wortartigen Zeichenketten in der Programmdatei von M. In der Nähe der Zeichenfolgen für Rechnernamen und Datenbankname tauchte eine Zeichenfolge auf, die aussieht wie ein Passwort. Mit diesem Passwort versuchte sich der Angeschuldigte dann auf den Datenbankserver des M zu verbinden und erhielt vollen Zugriff. Dort fand er in einer Datenbank mehrere Tabellen für die verschiedenen Kunden des M.

Nach seinen eigenen Angaben hat der Angeschuldigte sodann Screenshots gefertigt, um zu dokumentieren, dass er Zugriff hatte. Außerdem informierte er den Blogger und Webshopping-Experten S. Gemeinsam kontaktieren sie M nach ihren Angaben um 10 Uhr vormittags, informieren über das Sicherheitsproblem und fordern M auf, das Problem bis 13 Uhr zu beheben. M sei am Telefon unhöflich und uneinsichtig gewesen.[8] Allerdings sei um 10:30 Uhr das betroffene System vom Netz getrennt gewesen, neue Kontaktversuche zu M seien gescheitert. Daher habe man um 13 Uhr, da durch das abgeschaltete System kein Sicherheitsrisiko mehr bestand, den Vorfall auf dem Blog des S veröffentlicht.[9]

In der Folge hat M gegen S und den Angeschuldigten Strafanzeigen erstattet. Das Verfahren gegen S (§§ 186, 187 StGB) wurde laut S endgültig eingestellt, im verbliebenen Verfahren hielt das LG Aachen den Angeschuldigten, anders als das AG Jülich, hinreichend verdächtig für eine Strafbarkeit gemäß § 202a StGB.

2. Technischer Hintergrund: Compilieren, Interpretieren, Decompilieren und Disassemblieren

Prozessoren in Computern beherrschen ausschließlich das Binärformat, das Zahlen, Buchstaben und Instruktionen durch Nullen und Einsen darstellt. Diese Darstellung von Programmen ist der Maschinencode. Eine solche Darstellung ist schwer lesbar, weshalb schon früh die einfache Programmiersprache Assembler entwickelt wurde, die leicht merkbare Mnemonics[10] für Prozessor-Instruktionen nutzen. Diese Mnemonics lassen sich 1:1 in eine Binärdarstellung übersetzen. Allerdings sind hier nur einfache Befehle, wie z. B. ADD AX,2 (addiere den Wert 2 zum Registerinhalt von AX) möglich, komplexere Aufgaben wie „gib ‚Hello World‘ aus“, erfordern zahlreiche Instruktionen. Das ist in der alltäglichen Programmierarbeit unpraktisch, weshalb Hochsprachen wie Ada, Basic, C, Java, Pascal usw. entwickelt wurden, die dichter an natürlichen Sprachen liegen und für zahlreiche Aufgaben vorgefertigte Befehle haben. Diese Befehle sind in Buchstaben und Zeichen wiedergegeben, welche für den Menschen lesbar sind z. B. „print ‚Hello world!‘;“.

Damit Prozessoren diese Programme ausführen können, müssen sie zunächst in den Maschinencode übersetzt werden. Diese Aufgabe erledigen Compiler beim Hersteller oder Interpreter zur Laufzeit beim Nutzer der Software. Ob eine Spra-

4 So die Darstellungen des Angeschuldigten gemäß den Blogbeiträgen https://wortfilter.de/warnung-datenleck-beim-jtl-partner-modern-solution-gmbh-co-kg/ und https://wortfilter.de/falsche-anzeige-einer-datenschutzverletzung-durch-modern-solution-gmbh-co-kg, und dem Telefonat, das einer der Autoren mit dem Angeschuldigten geführt hat.

5 Disassemblieren ist der für diesen Vorgang treffendere Begriff als „Decompilieren“, wie im Beschluss des LG Aachen formuliert, zur Differenzierung siehe *Deusch/Eggendorfer*, in: Taeger/Pohle, Computerrechts-Handbuch, 37. EL Mai 2022, Kap. 50.1 Rn. 232k.

6 Ein Hash ist eine Einwegfunktion, also eine nicht-umkehrbare Funktion. Dadurch ist das Passwort unlesbar. Verwenden Sender und Empfänger des Passwortes allerdings dieselbe Hash-Funktion, erzeugt sie denselben Hash-Wert. So reicht es aus, zum Überprüfen eines Kennwortes nicht das Passwort selbst, sondern nur seinen Hashwert zu übertragen.

7 https://en.wikipedia.org/wiki/Strings_(Unix).

8 So der Angeschuldigte und S in einem Telefonat vom 3. 9. 2023.

9 https://wortfilter.de/warnung-datenleck-beim-jtl-partner-modern-solution-gmbh-co-kg/.

10 Mnemonic: ein Kürzel für einen Befehl in der Sprache Assembler, https://de.wikipedia.org/wiki/Mnemonic.

che compiliert oder interpretiert wird, entscheidet zumeist der Entwickler der Software, indem er sie entweder mit einem Interpreter oder Compiler ausliefert.[11] Der Vorteil der Compilierung ist ein schnellerer Ablauf, weil die Übersetzungszeit nur einmal anfällt und nicht bei jeder Instruktion, und die Möglichkeit, den Programmcode zu schützen: Denn bei einer interpretierten Sprache erhält der Nutzer alle Instruktionen in einer Hochsprache, so nur als schlecht lesbaren Maschinencode.[12]

Abbildung 1 zeigt ein sehr einfaches Beispielprogramm in der Programmiersprache C, das zwei fest einprogrammierte Zeichenketten miteinander vergleicht und ausgibt, ob sie gleich oder ungleich sind. Die Ausgabe von dem Programm führt „cat“ durch, das ist ein Befehl, der Textdateien ausgibt. Ersichtlich sind in Zeilen 5 und 6 die fest einprogrammierten Zeichenketten, ebenso wie in 8 und 11 diejenigen, die ausgegeben werden sollen.

Abbildung 1:

```
$cat demo.c
#include <stdio.h>
#include <strings.h>

int main ()
{
  char password[]="ganzgeheimespasswort";
  char something[]="wasanderes";

  if ( strcmp (password, something) == 0 )
    { printf ("%s und %s sind gleich\n",password,something);
    }
  else
    { printf ("%s und %s sind ungleich\n",password,something);
    }

  return 0;
}
```

Abbildung 2 zeigt einen Screenshot des Programms nach Ablauf, es gibt tatsächlich die beiden gespeicherten Zeichenketten aus und ob sie gleich oder ungleich sind. Ersichtlich ist zweiteres der Fall.

Abbildung 2:

```
$./demo
ganzgeheimespasswort und wasanderes sind ungleich
```

Abbildung 3 nutzt erneut den eigentlich ungeeigneten Befehl „cat“, der nur Texte darstellen kann, um das compilierte Programm auszugeben. Dabei entsteht ein besonderer Effekt: Weil „cat“ mit einem Text als Eingabe rechnet, interpretiert es alle Binärcodes, als stünden sie für Buchstaben. So entstehen zwar viele Zeichen, die nicht druckbar sind, und auch einiges an Buchstaben-Müll, andererseits sind aber auch die fest einprogrammierten Zeichenketten klar ersichtlich.

Abbildung 3:

```
$cat demo
????
     ? H__PAGEZERO?__TEXT@@__text__TEXT\>?\>?__stubs__TEXTD?$D?
                                                          __cstring__T
EXTh?Ph?__unwind_info__TEXT??H???__DATA_CONST@@@@__got__DATA_CONST@ @H__LIN
KEDIT?@?A4???3???0ÈP?X
                          P0?
Y*(?\>                         /usr/lib/dyld?\?9?{V5?2
      8d'/usr/lib/libSystem.B.dylib&)È???????{??C@@?????=??=??ω??@??=? @??
??q@?(p?*q??h7?????(???(?? >?
?????(???(??|>????_@@?  ???h7??R?{E?????_??@???
@???@??ganzgeheimespasswortwasanderes%s und %s sind gleich
%s und %s sind ungleich
\>44E?4
       ???? P`@@&L\___stack_chk_fail___stack_chk_guard_printf_strcmp_   _mh
_execute_headermain%?|?|\>.AI __mh_execute_header_main___stack_chk_fail___s
tack_chk_guard_printf_strcmp??
                                ????
                                    }]X       ??
                                                 @demo??$?}L?<ºê?¡??E??^[
sWD?<Q???Xo??f????kOX?|?|z?ô?H?,????Xo??f????kOX?|?|z?ô?H?,?^0??O?−???˝
                                                                      ??F
```

Dass es nicht-druckbare Zeichen gibt, zeigt die Ausgabe des Programms „hexdump“: Das zeigt für jeweils 8 Bit deren hexadezimalen Wert im mittleren Bereich an, und in der rechten Spalte das daraus erzeugte Zeichen. Ganz links ist in hexadezimal angegeben, wo in der Datei man sich aktuell befindet („Offset“, also Versatz zum Dateianfang).

```
00003e70  08 01 40 f9 a8 83 1f f8  ff 1f 00 b9 08 00 00 90  |..@.............|
00003e80  08 a1 3d 91 00 01 c0 3d  a0 83 00 d1 a0 03 9e 3c  |..=....=.......<|
00003e90  08 d1 40 f8 08 d0 00 f8  08 00 00 90 08 f5 3d 91  |..@...........=.|
00003ea0  09 01 40 f9 e1 83 00 91  e9 13 00 f9 08 71 40 b8  |..@.........q@.|
00003eb0  28 70 00 b8 2a 00 00 94  08 00 00 71 e8 07 9f 1a  |(p..*......q....|
00003ec0  68 01 00 37 01 00 00 14  e9 03 00 91 a8 83 00 d1  |h..7............|
00003ed0  28 01 00 f9 e8 83 00 91  28 05 00 f9 00 00 00 90  |(.......(.......|
00003ee0  00 20 3e 91 1b 00 00 94  0a 00 00 14 e9 03 00 91  |. >.............|
00003ef0  a8 83 00 d1 28 01 00 f9  e8 83 00 91 28 05 00 f9  |....(.......(...|
00003f00  00 00 00 90 00 7c 3e 91  12 00 00 94 01 00 00 14  |.....|>.........|
00003f10  a9 83 5f f8 08 00 00 b0  08 05 40 f9 08 01 40 f9  |.._.......@...@.|
00003f20  08 01 09 eb e8 17 9f 1a  68 00 00 37 01 00 00 14  |........h..7....|
00003f30  05 00 00 94 00 00 80 52  fd 7b 45 a9 ff 83 01 91  |.......R.{E.....|
00003f40  c0 03 5f d6 10 00 00 b0  10 02 40 f9 00 02 1f d6  |.._.......@.....|
00003f50  10 00 00 b0 10 0a 40 f9  00 02 1f d6 10 00 00 b0  |......@.........|
00003f60  10 0e 40 f9 00 02 1f d6  67 61 6e 7a 67 65 68 65  |..@.....ganzgehe|
00003f70  69 6d 65 73 70 61 73 73  77 6f 72 74 00 77 61 73  |imespasswort.was|
00003f80  61 6e 64 65 72 65 73 00  25 73 20 75 6e 64 20 25  |anderes.%s und %|
00003f90  73 20 73 69 6e 64 20 67  6c 65 69 63 68 0a 00 25  |s sind gleich..%|
00003fa0  73 20 75 6e 64 20 25 73  20 73 69 6e 64 20 75 6e  |s und %s sind un|
00003fb0  67 6c 65 69 63 68 0a 00  01 00 00 00 1c 00 00 00  |gleich..........|
00003fc0  00 00 00 00 1c 00 00 00  00 00 00 00 1c 00 00 00  |................|
00003fd0  02 00 00 00 5c 3e 00 00  34 00 00 00 34 00 00 00  |....\>..4...4...|
00003fe0  45 3f 00 00 00 00 00 00  34 00 00 00 03 00 00 00  |E?......4.......|
00003ff0  0c 00 01 00 10 00 01 00  00 00 00 00 00 00 00 04  |................|
00004000  00 00 00 00 00 00 10 80  01 00 00 00 00 00 10 80  |................|
```

Während der Maschinencode hier noch nicht lesbar ist, sind die Zeichenketten klar zu erkennen.

Um solchen Maschinencode zu analysieren, gibt es sogenannte Disassembler, die den Maschinencode in Assembler-Instruktionen übersetzen. Der so erzeugte Assemblercode ist oft immer noch schwer lesbar, weil auf Speicherstellen statt auf benannte Variablen zugegriffen wird, Funktionsaufrufe auch nur Speicherstellen statt Namen nutzen etc. Daher gibt es noch weitere Tools, wie z. B. das von der NSA entwickelte Ghidra,[13] einen „Decompiler“. Dabei versucht die Software bekannte Folgen von Maschinenbefehlen wieder zu Hochsprachenbefehlen zusammenzusetzen und so Programmcode in C zu generieren. Dabei entsteht nicht der Originalprogrammcode, den der Programmierer geschrieben hat. Auch dieser Code ist oft schwer lesbar, weil auch hier sprechende Bezeichnungen fehlen, aber schon etwas besser als das Disassemblierte. Eine tatsächliche Decompilierung liegt auch hier nicht vor, denn die würde zum Originalcode zurückführen.

Analyse-Möglichkeiten von compilierten Programmen:

Neben der Disassemblierung gibt es noch weitere Analysemöglichkeiten für compilierte Programme. So lassen sich mit einem Debugger Schritt für Schritt alle einzelnen Assembler-Instruktionen des Programms verfolgen und so Erkenntnisse über den Programmlauf gewinnen. Alternativ hilft es auch häufig, Zeichenketten zu extrahieren, wie im vorliegenden Fall. Der Angeschuldigte hat ein Programm wie „strings“ genutzt. Das Programm erkennt wortartige Zeichenfolgen in Maschinencode und gibt sie aus. Die Annahme dahinter ist, dass beim Übersetzen von Programmcode in Maschinencode auch nicht-alphanumerische Zeichen entstehen können – mit der Ausnahme eben von vorher einprogrammierten Texten, wie z. B. Bildschirmausgaben oder anderen Zeichenketten, die beim Compilieren unverändert bleiben.

Abbildung 5 zeigt ein Beispiel hierfür. Mehrere lesbare Zeichen hintereinander sieht strings dann als wortartig an.

11 Es gibt Ausnahmen, so gab es z. B. für BASIC sowohl Interpreter, die ursprüngliche Idee, als auch später Compiler.
12 *Deusch/Eggendorfer*, in: Taeger/Pohle (Fn. 5), Rn. 232k.
13 https://ghidra-sre.org/.

Abbildung 5:

```
$strings demo
ganzgeheimespasswort
wasanderes
%s und %s sind gleich
%s und %s sind ungleich
$
```

Im vorliegenden Fall des LG Aachen war dies ausreichend, um Passwörter, Rechnername und Datenbankname auszugeben. Nach eigenen Angaben habe der Angeschuldigte auch so das Passwort, das nur gehasht übertragen wurde, ausgelesen. Er habe es anschließend über das zur Fehlerbehebung decompiliert vorliegende Programm überprüft. Anders als in der Begründung des Beschlusses des LG Aachen sei jedoch die Decompilierung weder Voraussetzung noch notwendig gewesen. Technisch ist das so durchaus möglich.

In technischer Hinsicht ist das Vorgehen bei der Entwicklung der Software des Anbieters M in mehrerlei Hinsicht als nicht dem Stand der Technik entsprechend einzustufen: Unverschlüsselte Passwörter fest einprogrammiert abzulegen, gilt als Sicherheitslücke.[14] Außerdem ergibt sich aus der Darstellung, dass die Übertragung zwischen dem Programm und dem Datenbankserver unverschlüsselt erfolgt. Auch das entspricht nicht dem Stand der Technik.[15]

3. Responsible Disclosure/Coordinated Vulnerability Disclosure

Nachdem der Angeschuldigte die Sicherheitslücke festgestellt und dokumentiert hat, hat er zunächst den M als den Anbieter der fraglichen Software informiert und zur Behebung der Lücke aufgefordert. Damit zielte der Angeschuldigte auf die Regeln des „Responsible Disclosure“ ab. Dieses Verfahren ist zwar kein Gesetz, gilt unter Sicherheitsforschern aber als etablierter ethischer Standard beim Umgang mit entdeckten Sicherheitslücken. Dabei informieren die Entdecker von Sicherheitslücken den Hersteller der betroffenen Software vertraulich, geben ihm Zeit für Gegenmaßnahmen inklusive Fehlerbehebung, warten auf Rückmeldung, prüfen, ob die Fehlerbehebung erfolgreich war, und informieren dann erst die (Fach-)Öffentlichkeit über den Vorfall.[16]

Nicht immer funktioniert das in der Praxis so: Einige Anbieter ignorieren die Hinweise von Sicherheitsforschern, so hat einer der Autoren durchaus schon erlebt, dass erst auf eine Presseanfrage nach einer Stellungnahme hektische Betriebsamkeit entstand, andere werden aggressiv – so drohte einer einem der Autoren eine Unterlassungsklage für den Fall einer Veröffentlichung an, um dann zurückzurudern und anzubieten, statt des für den Beitrag über die Lücke vorgesehenen Platzes eine Anzeige zu schalten.[17] Natürlich gibt es auch konstruktive Kooperation, ohne Vorerfahrung mit dem jeweiligen Softwarehersteller ist die Prognose allerdings schwierig.

III. Rechtliche Würdigung

Das Entdecken von Sicherheitslücken gehört zum Alltag von IT-Sicherheitsforschern, Penetrationstestern und gewissenhaften Systemadministratoren, die Softwareanwendungen vor ihrer Installation testen.[18] Daher ist es von hoher Relevanz, wie der Umgang mit entdeckten Sicherheitslücken aus rechtlicher Sicht zu bewerten ist, insbesondere zu § 202a StGB.

1. Objektiver Tatbestand des § 202a StGB

a) Tatobjekt: besonders gesicherte Daten, die nicht für den Täter bestimmt sind

Für eine Strafbarkeit gemäß § 202a StGB ist zu klären, welche Daten betroffen sind, zu denen sich der Täter Zugang verschafft hat.

aa) Feststellung der betroffenen Daten und deren Relevanz für den Strafantrag

Dem Beschluss des LG Aachen ist nicht eindeutig zu entnehmen, auf welche Daten genau das Gericht den hinreichenden Tatverdacht des Angeschuldigten stützt.

Daten sind alle Informationen, die Gegenstand, Mittel oder Ergebnis eines Datenverarbeitungsvorgangs einer Datenverarbeitungsanlage sind.[19] Dies können Ein- oder Ausgabedaten sein, Zugangsdaten, Passwörter oder Programmdaten wie Maschinen- oder Quellcodes. Gemäß § 202a Abs. 2 StGB müssen die Daten elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden (§ 202a Abs. 2 StGB).

Im Fall des LG Aachen kommen hier der Quellcode der lokalen Software in Betracht, das darin enthaltene Passwort sowie die Zugangsdaten und Inhalte auf dem Datenbank-Server des M, außerdem die Datenbanksoftware auf dem Server des M.

(1) Quellcode

Ob das LG Aachen die Zugangsverschaffung *zum Quellcode* bestrafen will, ist dem Beschluss nicht eindeutig zu entnehmen. In Ziffer 1 (vor lit. a) der Beschlussgründe stellt das LG darauf ab, dass „der Angeschuldigte dem Quellcode der Software, den er mittels Decompilierung erlangte, dort hinterlegte Passwörter entnahm und die Daten auf seinen Rechner übertrug.“

Aus technischer Sicht ist es allerdings falsch, auf *den Quellcode* abzustellen, den der Angeschuldigte „durch Decompilierung“ erhalten haben soll. Denn aus einem Maschinencode kann durch einen Disassembler allenfalls eine teilweise Übersetzung des binären Programmcodes ermittelt werden, der disassemblierte Code ist zwar für den Menschen lesbar, entspricht aber nicht dem Quellcode, aus dem der Maschinencode erzeugt wurde. Insbesondere fehlen Kommentare, sprechende Variablen- und Funktionsnamen. Der Angeschuldigte kann allenfalls einen Quellcode mit den Funktionen der Software erzeugt haben, die sein Kunde in compilierter Form erhalten hat. Dies ist aber nicht der Quellcode, mit dem M den Maschinencode erzeugt hat.

14 Siehe dazu u. a. und beispielhaft https://owasp.org/www-community/vulnerabilities/Use_of_hard-coded_password. Den LfDI BaWü hat das auch im Knuddels-Fall zum Anlass für ein Bußgeld genommen, https://gdprhub.eu/index.php?title=LfDI_-_O_1018/115.

15 Siehe u. a. https://cwe.mitre.org/data/definitions/319.html.

16 *Böken*, in: Kipker, Cybersecurity, 2. Aufl. 2023, Kap. 19 Rn. 55; *Vonderau/Wagner*, vom Hörsaal in den Gerichtssaal, IT-Sicherheitsforschung als rechtliches Risiko in: Taeger (Hrsg.), Den Wandel begleiten, DSRI-Tagungsband 2020, 2020, S. 525. Die EU-Cybersicherheitsbehörde ENISA bezeichnet das Responsible Disclosure auch als „Coordinated-Vulnerability-Disclosure-Verfahren, siehe ENISA, Good Practice on Vulnerability Disclosure“, S. 24 (https://www.enisa.europa.eu/publications/vulnerability-disclosure).

17 https://www.linux-magazin.de/ausgaben/2008/12/botenstoff/.

18 Siehe auch *Deusch/Eggendorfer*, K&R 2018, 223 ff.

19 BGH, 27. 7. 2017 – 1 StR 412/16, K&R 2018, 793 ff., Rn. 31; *Eisele*, in: Schönke/Schröder, StGB, 30. Aufl. 2019, § 202a Rn. 3; *Brodowski*, in: Kipker (Fn. 16), Kap. 17 Rn. 40, 46.

„Der Quellcode" ist im vorliegenden Fall damit ein untaugliches Tatobjekt.[20]

(2) Passwort

Das Passwort in der lokalen Software könnte ein Tatobjekt sein. Allerdings ist falsch, dass sich der Angeschuldigte das Passwort durch „Decompilierung der Software" verschafft haben soll. Wie die Ausführungen oben in Abschnitt II zeigen, war die Rück-Erschließung eines Quellcodes überhaupt nicht notwendig. Das Passwort war nicht nur im Quellcode als Klartext enthalten, sondern bereits im Maschinencode. Die Zeichenkette des Passworts war lediglich durch allgemein verfügbare Programme wie „strings" wahrnehmbar zu machen. Dabei musste keine Zugangssicherung überwunden werden, das Passwort selbst war nicht geschützt [siehe unten lit. bb)].

(3) Zugangsdaten und Inhalte der Datenbank

Der Beschluss des LG Aachen stellt in Ziffer 1 darauf ab, dass der Angeschuldigte sich „die Daten" mithilfe des Passworts auf seinen Rechner übertrug. Offen bleibt, welche Daten genau gemeint sind. Das Passwort selbst jedenfalls scheidet in diesem Zusammenhang aus, weil es für das LG Aachen die „Zugangssicherung" zu den Daten ist.

Aus dem Sachverhalt des Beschlusses ergibt sich aber, dass sich auf dem Server des M „Kundendatenbanken (…) der jeweiligen Firmenkunden" des M befunden haben, insgesamt 600 000 - 700 000 Endkunden, „verteilt auf die verschiedenen Online-Händler, die ihrerseits Kunden" des M waren.

Die Autoren folgern hieraus, dass es sich bei der „Kundendatenbank" um ein Datenbanksystem handelt, das die Online-Händler nutzen können, um ihre eigenen Kundendaten zu verwalten. M stellt seinen Kunden somit keine „Kundendatenbanken" zur Verfügung, sondern nur IT-Mittel, die die Online-Händler dabei unterstützen, die Daten ihrer eigenen Kunden digital zu verwalten. Dieses Datenbankmanagementsystem (DBMS) ist zu unterscheiden von den Daten, die darin verwaltet werden. Denn eine Datenbank ist gemäß § 87a UrhG die Sammlung von Daten, die systematisch angeordnet sind. Programme, mit denen die Datenbank erstellt oder betrieben wird, gehören aus juristischer Sicht nicht zur Datenbank.[21] Auch technisch wird zwischen dem DBMS und der Datenbank als Ablagestruktur und als Datensammlung unterschieden, wenngleich der technische Sprachgebrauch alles unter dem Sammelbegriff „Datenbank" zusammenfasst.

Tatobjekt können nur „fremde" Daten sein, die nicht für den Täter bestimmt sind. Maßgeblich ist, wer über die Daten formell verfügungsberechtigt ist. Dies ist derjenige, der den Skripturakt der Daten verantwortet (Skribent). Bei Datenverarbeitungen im Auftrag ist dies der verantwortliche Auftraggeber.[22] Im Fall des LG Aachen sind dies die Kunden des M, aber nicht M selbst. Sie geben in das DBMS, das sie von M mieten, die Daten ihrer Kunden ein. Selbst M geht davon aus, dass er Auftragnehmer seiner Kunden im Rahmen einer Auftragsverarbeitung gemäß Art. 28 DSGVO ist.[23]

Für den Angeschuldigten bedeutet dies, dass die Daten nicht für ihn bestimmt sind, da er nicht Kunde des M war (möglicherweise aber dessen Auftraggeber, siehe dazu unten Ziffer 2). Allerdings war auch M insofern nicht verfügungsberechtigt, was für den Strafantrag relevant ist. Denn das Ausspähen von Daten wird gemäß § 205 Abs. 1 S. 2 StGB nur auf Antrag verfolgt. Antragsberechtigt ist nur der Verfügungsbefugte der betroffenen Daten.[24]

Soweit die Inhalte der Datenbank Tatobjekt sind, sind deshalb die jeweiligen Kunden des M befugt, Strafantrag zu stellen, und zwar jeder Kunde in Bezug auf „seine" Daten. Sofern der Angeschuldigte sich ausschließlich Zugang zu den Datenbankinhalten der Kunden von M verschafft hat und kein weiteres Tatobjekt betroffen ist, geht ein Strafantrag des M ins Leere.[25]

(4) Datenbanksoftware

Tatobjekt könnte zudem die Datenbanksoftware sein, die M auf seinem Server betreibt. Auf diese Daten stellt das LG Aachen in seinem Beschluss jedoch nicht ab.

bb) Besonders gesichert

Geeignetes Tatobjekt sind nur solche Daten, die gegen unberechtigten Zugang besonders gesichert sind. Wie die „besondere Zugangssicherung" beschaffen sein muss, ist rechtlich umstritten und vorliegend entscheidungserheblich:

(1) Maßgebliche Kriterien der Zugangssicherung

Das LG Aachen stellt unter Berufung auf den BGH in Ziffer 1 lit. b des Beschlusses darauf ab, dass die Zugangsbeschränkung das Interesse des Verfügungsberechtigten an der Geheimhaltung dokumentiert. Auf die Qualität der Sicherung soll es nicht ankommen. Nur wenn die Sicherung sehr leicht auszuschalten sei und jeder interessierte Laie sie überwinden könne, sei eine Zugangssicherung abzulehnen. Grund für diese weite Auslegung sei, dem Rechtsgut des § 202a StGB möglichst umfassenden Schutz einzuräumen.[26]

Das AG Jülich dagegen stellt zusätzlich darauf ab, ob die „besondere Sicherung" objektiv geeignet ist, den Zugang zu den Daten zu verhindern.[27] Interessant ist, dass auch *Eisele/Nolte* jüngst auf die objektive Eignung der Sicherung abstellen, obwohl die Kommentierung von *Eisele* in Schönke/Schröder auch im Beschluss des LG Aachen in Ziffer 1 lit. b, dritter Textabsatz als Beleg für die entgegengesetzte Meinung herangezogen wird. *Eisele/Nolte* stellen z. B. richtigerweise fest, dass der pauschale Verweis auf eine Firewall keineswegs in jedem Fall eine Zugangssicherung belegt. Firewalls regulieren den Datenverkehr in einem Netzwerk, verhindern aber nicht, dass der sich Täter mithilfe von Schadprogrammen oder anderen Angriffsverfahren Zugang zu Daten eines IT-Systems verschafft.[28]

20 Im Übrigen gehört die Disassemblierung zur Fehlerbehebung am Quellcode zur bestimmungsgemäßen Nutzung der Software gemäß § 69d UrhG; EuGH, 6. 10. 2021 – C-13/20, K&R 2021, 785 ff.; siehe dazu *Deusch/Eggendorfer*, in: Taeger/Pohle (Fn. 5), Kap. 50.1 Rn. 232 ff., 460.

21 *Dreier/Schulze*, UrhG, Kommentar, 7. Aufl. 2022, § 87a UrhG Rn. 5.

22 *Eisele*, in: Schönke/Schröder, StGB (Fn. 19), § 202a Rn. 9; *Cornelius*, in: Taeger/Pohle (Fn. 5), Kap. 102 Rn. 20.

23 https://wortfilter.de/falsche-anzeige-einer-datenschutzverletzung-durch-modern-solution-gmbh-co-kg/.

24 *Eisele*, in: Schönke/Schröder (Fn. 19), § 205 Rn. 4; *Cornelius*, in: Taeger/Pohle (Fn. 5), Kap. 102 Rn. 36.

25 Siehe dazu auch unten Abschnitt 2 a.

26 Neben dem LG Aachen wird diese weite Auslegung z. B. auch vertreten von BGH, 13. 5. 2020 – 5 StR 614/19 Rn. 38; *Eisele*, in: Schönke/Schröder (Fn. 19), § 202a Rn. 14; *Cornelius*, in: Taeger/Pohle (Fn. 5), Kap. 102 Rn. 24 - 28, *Beck*, in: Schuster/Grützmacher, IT-Recht, 2020, § 202a StGB Rn. 17 - 19. Auch der Beschluss des BGH, 21. 6. 2015 – 1 StR 16/15, K&R 2016, 53, ist in diese Auffassung einzuordnen. Zwar stellt der Leitsatz 1 auch auf die „Eignung" der Sicherung ab, allerdings liege diese bereits dann vor, wenn sie den Täter „zu einer Zugangsart zwingt, die der Verfügungsberechtigte erkennbar verhindern wollte."

27 AG Jülich, 10. 5. 2023 – 17 Cs 230 Js 99/21-55/23 unter Berufung auf *Hilgendorf*, in: LK-StGB, 13. Aufl. 2023, § 202a Rn. 3. Auch *Brodowski*, in: Kipker (Fn. 16), Kap. 17 Rn. 46, vertritt diese Ansicht. Laut *Härting*, Internetrecht, 7. Aufl. 2023, Kapitel A Rn. 187, soll es auf die „Eignung" der Sicherung ankommen, wobei im Folgesatz allerdings „passwortgeschützte E-Mail-Postfächer" pauschal als geeignet bezeichnet werden.

28 *Eisele/Nolte*, CR 2020, 488, 490.

Für die Auffassung einer geeigneten Zugangssicherung spricht, dass die Rechtsordnung vom Einzelnen verlangen kann, sich hinreichend selbst zu schützen, bevor eine Verhaltensweise strafrechtlich belangt wird. Seit Jahren steigt die Gefährdungslage für IT-Systeme, auch deshalb, weil IT-Anwender und IT-Anbieter verfügbare Schutzmaßnahmen nach dem Stand der Technik schlicht nicht ergreifen bzw. ihre IT-Lösungen fehlerhaft programmieren oder konfigurieren.[29] Für das Rechtsgut des § 202a StGB, die Integrität der angegriffenen Systeme, würde es daher höheren Schutz versprechen, wenn den Verfügungsberechtigten die elementare Relevanz effektiver IT-Sicherheit bewusst wird. Richtigerweise stellt das LG Aachen zwar darauf ab, dass § 202a StGB die völker- und EU-rechtlichen Vorgaben der Cybercrime-Convention umsetzt. Hiernach ist der unbefugte Zugang allerdings nur dann zu bestrafen, wenn „kein leichter Fall vorliegt".[30]

Im Fall des Angeschuldigten war es nicht nur so, dass die Passwort-Sicherung vom Stand der Technik abweicht. Vielmehr ist es aus technischer Sicht ein Kardinalfehler, Passwörter im Klartext in eine Software zu schreiben. Hinzukommt, dass die Passwörter bei der Herstellung der Datenverbindung zwischen dem lokalen IT-System des Nutzers und dem Datenbank-Server des M auch noch im Klartext übermittelt werden. Die Passwörter sind damit nicht nur aus der lokalen Software ohne Mühe auszulesen, sondern auch von jedem Internetnutzer, der die Datenverbindung wahrnimmt. Ein gefundenes Fressen für echte Angreifer, die zielgerichtet nach unverschlüsselten Datenverbindungen suchen.

Ein weiteres Kriterium der Zugangssicherung gilt unabhängig von dem dargestellten Meinungsstreit: Der Zweck der Sicherungsmaßnahme muss „darauf angelegt sein, den Zugriff Dritter auf die Daten auszuschließen oder wenigstens nicht unerheblich zu erschweren." Es genügt nicht, wenn die Zugangssicherung ein bloßer Nebeneffekt ist.[31]

(2) Kritik

Auf der Grundlage der vorgenannten Kriterien ist die Subsumtion des LG Aachen zur besonderen Zugangssicherung kritisch zu hinterfragen. Soweit man darauf abstellt, dass der Angeschuldigte Zugang zum *Quellcode* der lokalen Software hatte, liegt bereits kein hinreichendes Tatobjekt vor (siehe oben).

Darüber hinaus ist es auch in Bezug auf das *Passwort* aus technischer Sicht nicht nachvollziehbar, wenn das LG Aachen darauf abstellt, dieses sei „nur nach einer Decompilierung" abrufbar gewesen. Denn zur Kenntnisnahme des Passworts war keine Decompilierung notwendig, sondern lediglich das Auslesen von Zeichenfolgen durch das für jedermann verfügbare Programm strings. Zudem ist die Compilierung eines Programmcodes keine Zugangssicherung, sondern eine technische Notwendigkeit, um einen Programmcode auf einem Computer ablaufen zu lassen. Das Passwort war somit nicht zugangsgesichert.

Allerdings stellt das Passwort eine Zugangssicherung in Bezug auf die *Inhalte der Datenbank* dar. Dazu ist entscheidungserheblich, welcher Auffassung zur Qualität der Sicherung man folgt. Aus Sicht der Autoren ist es vorzugswürdig, die objektive Eignung der Zugangssicherung einzubeziehen. Diese fehlt im vorliegenden Fall. Ein Passwortschutz ist ungeeignet, wenn man das Passwort im Klartext zur Herstellung der Datenverbindung über das Internet überträgt oder auf einem nicht kontrollierten System, wie dem des Kunden, im Klartext ablegt.

Folgt man dagegen der Auffassung des LG Aachen, das die Erkennbarkeit des Geheimhaltungswillens heranzieht, bleibt die Frage, ob dazu der erforderliche Strafantrag gestellt wurde, da nicht M, sondern nur dessen einzelne Kunden zu den Datenbankinhalten verfügungsberechtigt waren.

Es bleibt somit als mögliches Tatobjekt die *Datenbanksoftware* des M. Es ist aber unklar, ob sich der Angeschuldigte dazu überhaupt Zugang verschafft hat. Laut Sachverhalt des Beschlusses hat der Angeschuldigte das Passwort dazu verwendet, „Zugangsdaten" zur Datenbank zu kopieren und mit diesen Zugangsdaten Inhalte der Datenbank zu kopieren.

(3) Praxishinweis

IT-Sicherheitsforscher und Penetrationstester sollten genau prüfen, ob sich bei der ermittelten Sicherheitslücke möglicherweise eine (wenn auch ungeeignete) Zugangssicherung zeigt.

Solange sich die Rechtsprechung nicht ändert, bleibt ein Strafbarkeitsrisiko, wenn ungeeignete Zugangssicherungen erforscht werden. Allerdings erscheint es dringend nötig, auf solche massiven Qualitätsmängel in Software hinweisen zu dürfen, ohne sich auch nur dem entfernten Risiko einer Strafbarkeit auszusetzen. Hier wäre es hilfreich, wenn die Legislative im Rahmen der Cybersicherheitsinitiative des Bundes entsprechende Klarstellung schafft.

b) Tathandlung: Zugang verschaffen und Zugangssicherung überwinden

Der Täter ist nur strafbar, wenn er sich Zugang zu den verfahrensgegenständlichen Daten verschafft hat. Es ist nicht notwendig, dass der Täter auch tatsächlich Zugriff auf die Daten genommen hat, etwa durch Lesen oder Kopieren. Ausreichend ist, dass er Herrschaft über den Datenträger bzw. das IT-System erlangt hat, auf dem die Daten gespeichert sind. Dabei muss allerdings das Überwinden der Zugangssicherung kausal sein für den Zugang.[32]

Für den Angeschuldigten ist somit relevant, welche Sicherung er überwunden hat, um zu welchen Daten zu gelangen. Auch hier sind die betreffenden Daten abzuschichten:

Quellcode und *Passwort* hatten bereits keine Zugangssicherung (siehe oben).

Sofern man für die *Datenbank-Inhalte* trotz des ungenügenden Passwortschutzes entgegen der hier vertretenen Meinung eine Zugangssicherung bejaht, kommt eine Vollendung der Tathandlung insoweit in Betracht (allerdings ist diesbezüglich nicht M, sondern dessen Kunden verfügungsberechtigt).

Es verbleibt die *Datenbanksoftware*. Allerdings gibt es keine Feststellungen im Beschluss des Gerichts, dass das Passwort den Zugang zur Datenbanksoftware schützen sollte. Vielmehr soll sich der Angeschuldigte mit dem Passwort lediglich „Zugangsdaten" beschafft haben, mit denen er sich wiederum Zugang zum Inhalt der Datenbanken auf dem Server des M verschafft habe. Es steht somit nicht fest, ob der Angeschuldigte Zugang zur Datenbanksoftware hatte und, falls ja, ob das Überwinden des Passwortschutzes dafür kausal war.

29 *Deusch/Eggendorfer*, K&R 2016, 152, 153 f.
30 Art. 3 RL 2013/40/EU, ABl. EU Nr. L 218/8, 14. 8. 2013.
31 BGH, 27. 7. 2017 – 1 StR 412/16, K&R 2018, 793 ff., Rn. 40; ebenso *Cornelius*, in: Taeger/Pohle (Fn. 5), Kap. 102 Rn. 25; *Eisele*, in: Schönke/Schröder (Fn. 19), § 202a Rn. 14.
32 *Cornelius*, in: Taeger/Pohle (Fn. 5), Kap. 102 Rn. 31, 32; *Eisele*, in: Schönke/Schröder (Fn. 19), § 202a Rn. 20.

2. Vorsatz, Rechtswidrigkeit und Schuld

Das LG Aachen hat sich mit der Rechtswidrigkeit und Schuld der Tat nicht auseinandergesetzt. Folgende Aspekte sind dazu indes relevant:

a) Einwilligung und Auftrag des Angeschuldigten

Wenn das Ausspähen von Daten durch eine Einwilligung gerechtfertigt ist, ist die Tat nicht rechtswidrig und straflos. Allerdings kann nur der Verfügungsbefugte eine wirksame Einwilligung erteilen. Wenn eine nicht verfügungsbefugte Person die Einwilligung erteilt, kann ein Erlaubnistatbestandsirrtum vorliegen, der den Vorsatz ausschließt, wenn der Täter annimmt, dass die betroffenen Daten ausschließlich der einwilligenden Person gehören. Ein Verbotsirrtum liegt dagegen vor, wenn der Täter weiß, dass die betroffenen Daten nicht der verfügungsbefugten Person gehören, er aber glaubt, die Person sei dennoch berechtigt, ihm den Zugang zu den Daten zu erlauben.[33]

Der Angeschuldigte handelte im Auftrag seines Kunden, der die lokale Software des M nutzen wollte. Es liegt daher zwar nahe, dass der Auftraggeber des Angeschuldigten auch ein Kunde des M war, als solcher könnte er jedoch lediglich einwilligen, soweit seine eigenen Daten betroffen sind. Dem Beschluss des Gerichts ist indes nicht zu entnehmen, wessen Daten genau das Tatobjekt waren.

Praxishinweis:

Insbesondere Penetrationstester sollten vor dem Test feststellen, wer als Verletzter eines Ausspähens von Daten in Betracht kommt und von den identifizierten Stellen wirksame Einwilligungen einholen. Soweit mehrere Verletzte in Betracht kommen, etwa der Anbieter und der Nutzer eines IT-Systems, sollten die Einwilligungen aller Betroffenen eingeholt werden.[34]

Für die Einwilligung gibt es keine Formvorschriften. Aus Nachweisgründen empfiehlt sich indes die Schriftform oder eine qualifizierte elektronische Signatur (§§ 126, 126a BGB). Falls auch personenbezogene Daten betroffen sind, muss die Einwilligung die weiteren Anforderungen des Art. 4 Nr. 11 DSGVO erfüllen.

b) Notstand und Responsible Disclosure

Wenn die Tat, vom Täter gewollt, eine Notstandslage beseitigt, ist sie gemäß § 24 StGB gerechtfertigt.[35]

Sicherheitslücken ermöglichen es Straftätern, betroffene IT-Systeme und, davon ausgehend, weitere Systeme zu schädigen. Somit kann eine Sicherheitslücke eine gegenwärtige Gefahr im Sinne darstellen. Dies gilt auch dann, wenn die Sicherheitslücke nicht akut ausgenutzt wird, in diesem Fall liegt eine Dauergefahr vor.[36]

Eine zielgerichtete Feststellung der Sicherheitslücke und die Information des Herstellers bzw. der Öffentlichkeit nach den Grundsätzen des Responsible Disclosure kann eine Notstandshandlung sein. Dabei dient die Zugangsverschaffung zum IT-System und die Dokumentation der Sicherheitslücke dazu, den IT-Anbieter angemessen zu informieren, damit er die Lücke beseitigen kann. Falls der informierte IT-Anbieter die Lücke nicht beseitigt, besteht die Gefahr fort und die Information der Öffentlichkeit kann notwendig sein, um den betroffenen Nutzern die Gelegenheit zu geben, die Gefahr durch Beendigung der Nutzung zu beseitigen. Allerdings ist in jeder Lage dieses Vorgangs abzuwägen, ob die jeweilige Notstandshandlung (Eindringen in das System, Kopieren von Daten und Information der Öffentlichkeit) erforderlich und angemessen zur Notstandslage ist. Problematisch können Fälle sein, in denen vor dem Eindringen in ein fremdes System noch nicht feststeht, ob eine Sicherheitslücke vorliegt. Dabei ist darauf abzustellen, wie stark die Verdachtsmomente sind, die für die Wahrscheinlichkeit der Sicherheitslücke sprechen. § 34 StGB rechtfertigt kein anlassloses Eindringen in fremde IT-Systeme, um eine bestimmte Anwendung ohne jeden Verdacht zu überprüfen. Konkrete Hinweise, bei denen sich nach dem Stand der IT-Sicherheitsforschung Sicherheitslücken aufdrängen, können dagegen eine nähere Prüfung des betroffenen IT-Systems rechtfertigen. Dies kann es auch umfassen, sich ohne Wissen des Verfügungsberechtigten Zugang zum IT-System zu verschaffen, allerdings nur im Rahmen des Erforderlichen.[37]

Stellt sich nachträglich heraus, dass entgegen der ursprünglichen Annahme doch keine Sicherheitslücke vorliegt, kann ein Erlaubnistatbestandsirrtum vorliegen, der analog § 16 StGB den Vorsatz entfallen lässt.[38] Voraussetzung hierfür ist allerdings, dass der Handelnde von einer Sicherheitslücke ausging, die eine Gefahr verursacht. Diesen Nachweis wird der Handelnde nur führen können, wenn er den Sachverhalt lege artis dokumentiert und aufgrund des Stands der Technik von den bestehenden Verdachtsmomenten auf eine Sicherheitslücke schließen darf.

Der Angeschuldigte hat, ohne tatbestandsmäßig zu handeln, festgestellt, dass die lokale Software des M Passwörter im Klartext enthält. Dieser Verstoß gegen die Regeln der Sicherheitstechnik legt den Verdacht nahe, dass die Kommunikation mit dem Datenbanksystem aufgrund des Passworts im Klartext auch Unbefugten offensteht, was eine Sicherheitslücke darstellt. Die Dokumentation dieses Umstands durch den Angeschuldigten ist auch erforderlich, um den M angemessen über die Sicherheitslücke zu informieren und die Notstandslage zu beseitigen. Ob es ausgereicht hätte, den M darüber zu informieren, dass in der lokalen Software ein Passwort im Klartext enthalten ist, bleibt fraglich. Denn ohne die genaue Kenntnis der Sicherheitslücke wäre nicht nachvollziehbar, ob M die Lücke und somit die Gefahr für die betroffenen Verfügungsberechtigten auch tatsächlich beseitigt.[39]

Bei dieser Betrachtung zeigt sich auch die *Wirkung des Responsible Disclosure*: (Information des Anbieters über die ermittelte Lücke vor Veröffentlichung):

In der IT-Szene grassiert bisweilen die Vorstellung, ein Vorgehen nach dem Responsible Disclosure kennzeichne den Handelnden als „white hat“ (Hacker), dessen Handeln stets gerechtfertigt sei. Dies ist mitnichten der Fall. Das Responsible Disclosure macht es nicht ungeschehen, wenn Straftatbestände verwirklicht sind. Es lässt auch nicht den Vorsatz entfallen,

33 *Cornelius*, in: Taeger/Pohle (Fn. 5), Kap. 102 Rn. 33, 34. Ob die Erklärung des Verfügungsbefugten eine rechtfertigende Einwilligung oder ein tatbestandsausschließendes Einverständnis darstellen soll, ist streitig, siehe *Wagner*, PinG 2020, 66, 69 sowie *Beck*, in: Schuster/Grützmacher (Fn. 26), § 202a StGB Rn. 26.
34 *Wagner*, PinG 2020, 66, 69; *Böken*, in: Kipker (Fn. 16), Kap. 19 Rn. 78 f.
35 *Perron*, in: Schönke/Schröder (Fn. 19), § 34 Rn. 8, 22, 48.
36 *Wagner*, PinG 2020, 66, 73.
37 *Wagner*, PinG 2020, 66, 73 f.
38 *Perron*, in: Schönke/Schröder (Fn. 19), § 34 Rn. 48.
39 Ob sich aus der Kenntnis des Angeschuldigten sogar eine Garantenpflicht gegenüber den Kunden des M ergeben könnte, angemessene und effektive Maßnahmen zur Gefahrenbeseitigung zu treffen, bleibt einer gesonderten Untersuchung vorbehalten. In diesem Fall wäre der Angeschuldigte sogar verpflichtet gewesen, die Lücke und deren Beseitigung zu dokumentieren, weil er es andernfalls unterlassen hätte, Angriffe durch andere Täter auf das System des M (und dessen Kunden) zu verhindern. Somit könnte die Information des M ohne Nachweis der Lücke sogar zu einer Strafbarkeit des Angeschuldigten gemäß § 13 StGB führen.

weil hierfür nicht das Motiv entscheidend ist, weshalb eine Straftat begangen wird, sondern, ob dem Täter Kenntnis über alle Tatbestandsmerkmale vorlagen. Schließlich stellt das Responsible Disclosure auch keinen selbständigen Rechtfertigungsgrund dar, der die Verwirklichung von Straftatbeständen legitimieren würde. Allerdings kann es eine Unterstützung sein, die Notstandslage und vor allem den Notstandswillen des Handelnden zu dokumentieren. Ein Hacker, dem ein Eindringen in fremde Systeme nachgewiesen wird, wird es schwer haben, sich auf einen Notstandswillen zu berufen, wenn er niemanden über die von ihm entdeckten Gefahren unterrichtet.

Praxishinweis:

IT-Sicherheitsforscher und Penetrationstester sollten in jeder Prüfungsstufe die Verdachtsmomente dokumentieren, die für eine Sicherheitslücke sprechen. Die Handelnden haben stets kritisch zu hinterfragen und auch zu dokumentieren, ob die ermittelten Erkenntnisse jeweils ausreichen, um eine konkrete Gefahrenlage anzunehmen, die weitere Ermittlungen verlangen.

Dabei kann ein Handeln nach den Grundsätzen des Responsible Disclosure die erforderliche Dokumentation unterstützen. Responsible Disclosure ist allerdings kein Freibrief für das Hacken fremder Systeme.

IV. Fazit und Thesen

1. Die Erforschung von IT-Sicherheitslücken ist de lege lata stets mit einem gewissen Risiko der Strafbarkeit, insbesondere aus § 202a StGB, verbunden.

IT-Sicherheitsforschenden und Penetrationstestern ist deshalb zu empfehlen, ihr Handeln sorgfältig von strafbaren Eingriffen in fremde IT-Systeme und Daten abzugrenzen.

2. Für die Beurteilung, ob sich eine solche Strafbarkeit realisiert hat, ist der zugrundeliegende Sachverhalt sorgfältig aufzuklären, dabei ist sachverständige Unterstützung aus der Informatik regelmäßig hilfreich.

3. Für die Beurteilung einer Strafbarkeit aus § 202a StGB ist insbesondere abzuschichten, welche Daten genau betroffen sind, ob dazu jeweils eine geeignete Zugangssicherung vorliegt und wer über die betroffenen Daten verfügungsbefugt ist. Nach Auffassung der Autoren sind nur solche Zugangssicherungen relevant, die geeignet sind, um unbefugte Zugänge zu verhindern. Die offene Kommunikation von Passwörtern im Klartext, sei es bei der Übertragung oder ihrer Speicherung, schließt eine geeignete Sicherung aus.

4. Penetrationstester und IT-Sicherheitsforscher müssen bei ihrer Tätigkeit folgende Fragen klären:

- Gibt es bei der Feststellung der Sicherheitslücke eine Zugangssicherung?
- Solange sich die Rechtsprechung der hier vertretenen Auffassung nicht anschießt, ist auch bei ungeeigneten Zugangssicherungen Vorsicht geboten.
- Wer ist verfügungsbefugt über die Daten, zu deren IT-System die Sicherheitsüberprüfung durchzuführen ist?
- Liegen von allen verfügungsbefugten Personen wirksame Einwilligungserklärungen vor?
- In jeder Lage der Sicherheitsprüfung sind möglichst umfassende Dokumentationen zu treffen. Sicherheitslücken können eine Notstandshandlung rechtfertigen. Allerdings reichen bloße Vermutungen nicht aus; Sicherheitslücken, damit verbundene Gefahren und das Erfordernis weiterer Ermittlungen können nur nach Maßgabe des Stands der Technik angenommen werden.

5. Das Responsible Disclosure Verfahren ist kein Freibrief, im Interesse der Erforschung von Sicherheitslücken Straftaten zu begehen. Es lässt weder automatisch den Vorsatz entfallen noch rechtfertigt es begangene Straftaten. Allerdings unterstützt es die Dokumentation von Notstandssituationen und ist in vielen Fällen dafür sogar unverzichtbar.

Florian Deusch
ist Rechtsanwalt und Fachanwalt für Informationstechnologierecht in der Anwaltskanzlei Dr. Gretter in Ravensburg. Er ist zudem als Datenschutzbeauftragter tätig.

Tobias Eggendorfer
ist Professor für Sicherheit in verteilten Anwendungen an der TH Ingolstadt, davor war er als Abteilungsleiter „Sichere Systeme“ an der Agentur für Innovation in der Cybersicherheit für die Weiterentwicklung der Forschung im Bereich der IT-Sicherheit zuständig. Er ist zudem als IT- und Datenschutzbeauftragter und Lehrbeauftragter tätig.

RA Markus Schröder, LL.M.*

Anonymisieren und seine Ruhe haben?

Kurz und Knapp

Bei Verträgen zur Datenanalyse gerade im Start-up/Fintech-Bereich finden sich häufig Klauseln, wonach die überlassenen personenbezogenen Daten nach Beendigung des Auftrags nicht gelöscht bzw. herausgegeben werden müssen. Vielmehr sollen die Daten „nur“ anonymisiert werden. Diese Daten werden dann durch den (vormaligen) Auftragnehmer für eigene Zwecke, wie das Anlernen von Algorithmen, genutzt. Dabei bestehen datenschutzrechtliche Fragen und solche des Geheimnisschutzes. Diesen Fragen geht der Beitrag nach.

* Der Beitrag geht auf einen Vortrag bei der DSRI-Herbstakademie 2023 zurück, deren Vorträge im Tagungsband, Oldenburger Verlag für Wirtschaft, Informatik und Recht, veröffentlicht wurden. Mehr über den Autor erfahren Sie am Ende des Beitrags.