erschienen in der FIfF-Kommunikation,
herausgegeben von FIfF e.V. - ISSN 0938-3476
www.fiff.de

Mathias Haimerl

Zum technologischen Infantilismus der Industrie und des Healthcare-Sektors

Während die Digitalisierung voranschreitet und die Abhängigkeit von Computern auf einem nie dagewesenen Level ist, stillen Konzerne ihren Datenhunger auf Kosten des Datenschutzes, wirtschaftlicher Schäden und sogar Menschenleben. Das Ungleichgewicht zwischen den Interessen von Software-Unternehmen und den Endanwendern wird medienwirksam verpackt und quelloffene Software marketing-technisch torpediert. Warum betrifft das Problem jeden Einzelnen von uns? Was können wir dagegen tun? Und warum müssen wir etwas dagegen tun?

Die Bedrohung ist hoch. Firmen [8], Gemeindeverwaltungen [35] und Krankenhäuser [20] werden immer mehr zur Zielscheibe digitaler Bedrohungen. Das Problem ist die Basis vieler Systeme – allen voran im Gesundheitswesen. Als Software-Architekt im Gesundheitssektor bin ich immer wieder erstaunt, wie ignorant an grundlegende Probleme herangegangen wird und schwerwiegende – teure oder potenziell tödliche [13, 14] – Risiken businesstauglich in Kauf genommen werden. Ich bin überzeugt, dass die Gesellschaft die Probleme verstehen muss, damit der Gesundheitssektor in Bewegung kommt, denn heute scheint es nicht genug, wenn es brennt, solange mit dem brennenden System noch Geld zu machen ist.

Das Grundproblem – Informationssicherheit

In einer Zeit, in der soziale Medien das Gefühl für Privatsphäre und Datenschutz anscheinend *wegoptimieren,* scheint das Bewusstsein dafür in der Bevölkerung auch bei anderen Themen zu schwinden. Finden wir es zum Teil noch in Ordnung, wenn unsere Bankdaten über die USA geschickt werden [27], so sind sich doch die meisten Menschen im Klaren, dass Gesundheitsdaten sehr persönlich sind. Warum kümmern wir uns nicht darum? Warum gehen wir davon aus, dass der Arzt *schon weiß, was er tut,* wenn er einen Computer bedient? Warum gehen wir davon aus, der IT-Leiter einer Klinik *weiß was er tut,* wenn er ein neues System aus Werbe- oder Kostengründen einführt?

In diesem Kontext möchte ich die Schutzziele der Informationssicherheit beleuchten, im Vergleich von Windows als proprietärer zu Linux als quelloffener Software.

Vertraulichkeit

Spätestens mit Windows 10 hat Microsoft eine nie dagewesene Datensammelwut begonnen, die sich nur für bestimmte Versionen überhaupt deaktivieren lässt – und dort schwer [31]. Der Aufbau dieser Lauschinfrastruktur, genannt *Event Tracking for Windows (ETW),* ist komplex und tief im System verankert. Technisch gesehen könnte dadurch jede Information, die am Computer aufgerufen oder eingegeben wird, an Microsoft übermittelt werden [15]. Allein dieses standardmäßig aktivierte Modul führt zu einer grundsätzlichen Verletzung der Vertraulichkeit. Das *Need-to-know*-Prinzip [32], eines der Grundprinzipien des Datenschutzes wird hierbei vollständig ignoriert.

Microsoft stellt zwar eine immense Liste mit Art und Umfang der gesammelten Daten online zur Verfügung [18], allerdings ist es bei Firmen, die sich selbst kontrollieren, immer fraglich, inwieweit diese Informationen vollständig und wahrheitsgemäß sind – oder ob die von Microsoft gestellten Hürden die Administratoren von einer Deaktivierung der Telemetrie abhält. Laut Microsoft werden diese Daten erfasst, um das System laufend zu verbessern. Ich selbst habe während der Nutzungszeit von Windows (nach XP) keinerlei Verbesserung erfahren. Da ich Windows und Linux parallel nutze, habe ich einen direkten Vergleich.

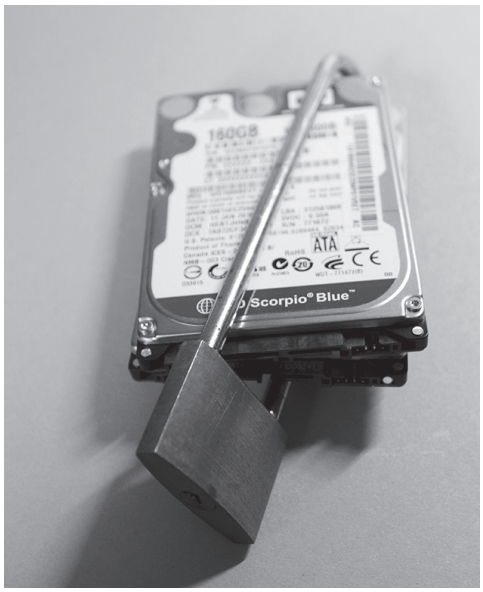
Integrität & Verfügbarkeit

Beide sind Aspekte der Robustheit gegenüber Hackern und Viren, unerwarteten Systemeingaben oder dem Zusammenkommen unterschiedlicher Systemzustände [9]. Alle Windows-Nut-

zer kennen den *Blue Screen of Death (BSOD).* Der Systemkern von Windows stellt bei diversen Konstellationen seine Arbeit ein und zeigt das an. Weiterarbeiten wird verhindert, der Nutzer aufgefordert zu warten, bis genug Informationen gesammelt wurden. Die ungesicherten Daten des Nutzers sind natürlich trotzdem verloren. Aber ein BSOD ist nicht das Ende. Zwar verliert man alle ungesicherten Daten, kann aber den Computer neu starten und in der Regel weiter arbeiten. Anders sieht es bei Verschlüsselungs-Trojanern bzw. Ransomware aus. Ein kleines Stück Software verschlüsselt das System und alle Daten. Ist das erledigt, wird dem Nutzer nur noch der Preis für das Entschlüsseln angezeigt [24]. Diese Art von Schad-Software verbreitet sich aus gutem Grund am stärksten: Ransomware macht das System unverfügbar. Jede Minute, die ein System nicht verfügbar ist, kostet Geld. Je zentraler, desto höhere Kosten, weshalb zwei von drei Unternehmen das Lösegeld bezahlen [28]. Diese Art von Erpressung funktioniert, was zu neuen, besseren Crypto-Trojanern führt. Ein Teufelskreis auf Kosten der Verfügbarkeit. Zwar ist Windows nicht das einzige Betriebssystem, bei dem Systemabstürze oder Ransomware existieren, aber im Vergleich zu anderen Systemen hat Windows das größte Verbesserungspotenzial [25]. Als ich zu Linux wechselte, hieß es: „unter Linux gibt es keine Viren". Das ist zwar nicht korrekt, aber hat einen wahren Kern: Um Viren *scharf zu schalten,* benötigt man Sicherheitslücken, um Administrator-Privilegien zu bekommen. Erst dann kann man im System weitgehend ungestört sein Unwesen treiben. Diese unter Windows zu finden ist kein Problem [1]. Es gibt einen globalen Schwarzmarkt für bisher undokumentierte ZeroDay-Sicherheitslücken, was neben Hackern auch Staaten für sich entdeckt haben [4]. Linux hat hier zwei entscheidende Vorteile:

1. Open Source

   Dadurch, dass die Quelldateien offen für jeden Interessierten einsehbar sind, werden Fehler meist frühzeitig entdeckt und behoben. Grobe Fehler fallen meist früher auf, allerdings manchmal auch erst Jahre später [2]. In der Zeit vom Finden eines Fehlers, bis der Fehler behoben und an alle Endgeräte ausgeliefert ist, muss ein Virus entwickelt, verteilt und *aktiviert/etabliert* werden. Viele Viren werden erst durch die Ausnutzung mehrerer Lücken gefährlich. Dadurch ist es sehr aufwändig und weniger attraktiv, Viren für Linux zu entwickeln.



Informationssicherheit

2. Kurze Updatezyklen

   Windows-Nutzer kennen die großen, in Zyklen verteilten Sicherheitsupdates [5]. Hier werden viele Lösungen für Sicherheitslücken zusammengepackt und ausgeliefert. Es ist prinzipiell schlecht, die Lösung einer Sicherheitslücke zurückzuhalten, um gebündelt auszuliefern. Hier zeigt sich der zweite große Vorteil von Linux: Modularität [22]. Während Windows als Monolith Gesamtupdates von zentraler Stelle erhält, besteht eine Linux-Distribution aus einer Zusammenstellung von Programmen für verschiedene Aufgaben. Jedes dieser Programme (z. B. Datei-Manager, Editor, Shell, …) hat seine eigenen Entwickler, die ihr Projekt aus Eigeninitiative entwickeln. Daher werden Bugs in quelloffenen Programmen meist innerhalb weniger Stunden behoben. Die meisten Distributionen haben dezentrale Verwaltungen, bei denen eine neue Version angekündigt wird. Endanwender können regelmäßig prüfen, ob ein neues Update vorhanden ist, und dieses mit einem Klick installieren. Oder man automatisiert das Ganze mit einem einmaligen Klick [19]. Es kommt nicht selten vor, dass der Patch für eine Lücke, über die ich eben gelesen habe, bereits zur Installation bereitsteht. Wenn die ausgenutzte Lücke durch die Entdeckung des Virus auffällt und innerhalb weniger Stunden geschlossen wird, kann der Virus einige wenige Computer infizieren, sich aber nicht etablieren.

Kein Argument: Verbreitung

Ein zentrales Argument, das häufig an dieser Stelle von Microsoft-Fans angeführt wird, ist, dass Linux nicht so verbreitet sei. Es lohne sich nicht, Malware für Linux zu entwickeln. Betrachten wir die Betriebssysteme aller Geräte weltweit, so hat Linux sogar die größte Verbreitung. Wir dürfen dafür nicht nur PCs, sondern auch Server und Smart Devices betrachten. Die Linux Distribution Android gehört mit über 50 % Marktanteil bei Smartphones zu den größten Treibern der Verbreitung von Linux im letzten Jahrzehnt. Natürlich stieg dadurch die Anzahl an Viren mit Android als Ziel. Allerdings hat genau dieser Trend dazu geführt, dass mehr Lücken in Linux gefunden und gestopft wurden [3].

Das Beispiel Android zeigt auch, dass Sammelupdates den Weg für Viren erleichtern. Während bei anderen Distributionen ganz typischerweise Updates für Einzelpakete durchgeführt werden, nutzen alle mir bekannten Hersteller Sammelupdates und *vertagen* dadurch relevante Sicherheitsupdates. Insgesamt scheinen aber mobile Systeme für Hacker zunehmend uninteressant zu werden, wenn man die aktuellen Statistiken von *Kaspersky* [16] mit denen von 2012 [21] vergleicht.

Status Quo: „Vendor Lock-in"

Auch in Arztpraxen und Krankenhäusern werden fast ausschließlich auf Microsoft Windows basierende Systeme verwendet. Das mag sinnvoll erscheinen, denn jemand, der es gewohnt ist privat

Windows zu nutzen, wird das gleiche System am Arbeitsplatz besser bedienen können. Das entspricht allerdings nicht der Realität. Niemand arbeitet mit dem Betriebssystem, sondern mit den Anwendungen. Kein Buchhalter installiert Treiber. Kein Arzt richtet Netzwerke ein. Dafür gibt es IT-Spezialisten. Natürlich gibt es Ausnahmen, wie Ärzte, die sich selbst für ein Linux-System entschieden haben (und damit sehr zufrieden sind).

Nicht alle Windows-Anwendungen lassen sich auf anderen Systemen (Linux, Mac OS, …) installieren, und die meisten Anbieter bieten ihre Programme nur für Windows an. Warum sollten die Firmen den Aufwand betreiben, die meist aufwändige Migration auf weitere Systeme zu stemmen? Aus rein wirtschaftlichen Motiven ist das widersinnig. Es gibt nur einen kleinen Markt. Würden mehr Anwender Linux nutzen, würde es sich eher rentieren. Aber wo kein Wille, da kein Weg. Warum lohnt sich der Aufbruch ins #*Neuland* trotzdem?

Firmen versuchen Interna intern zu halten. Geschäftsgeheimnisse dürfen nicht jedem zugänglich sein. Insbesondere nicht der Konkurrenz oder noch schlimmer: Menschen, die Geheimnisse der Konkurrenz verkaufen. Offensichtlich war dieses Schutzbegehren Microsoft ein Dorn im Auge. Was lag da näher, als die Office-Anwendungen in die Cloud zu verlagern? Der beste Grund, die persönlichen Daten von lokalen auf fremde Rechner zu laden, ist, dass die Software dort läuft. – Das Ganze ist automatisch immer up-to-date und dadurch sicherer … und billiger! Firmen sind reihenweise auf die günstige Lösung aufgesprungen, ohne sich großartig Gedanken darüber zu machen, was das für die Hoheit über die eigenen Daten bedeutet. Microsoft dürfte bei den Vertragsabschlüssen klar sein, dass der Zugriff auf die Daten jederzeit möglich ist. Das Gegenüber spart sich IT-Fachpersonal, das bisher die Computer aktuell halten muss. Das muss es zwar immer noch, aber zumindest nicht die Office-Programme. Seit der Einführung von Office als Cloud-Version hat sich einiges getan: Die Preise sind gestiegen [29] und sicher sind die Daten natürlich auch nicht [10, 6].

Apropos: Die Cloud-Lösung ermöglicht auch, dass man Office auf Linux-Rechnern verwendet. Im Browser. Mit eingeschränktem Funktionsumfang. Bei vollem Preis. Keine Pointe.

Never touch a running system

Angenommen, Windows wäre ein gutes Betriebssystem. Selbst unter dieser Annahme gibt es sehr gute Gründe, die gegen eine Entwicklung von *Windows-only*-Produkten sprechen, denn die oft gehörte Aussage, dass die Nutzer primär Windows nutzen wollen, ist eine *self-fulfilling prophecy:* Heute behaupten wir, dass es sich nicht rentiert, für Linux zu entwickeln, morgen müssen noch mehr Nutzer ungewollt Windows einsetzen, weil es ihre Software für Linux nicht gibt. Dadurch sinkt der Anteil an Personen, die alternative Systeme nutzen … und so fort. Meine Meinung als Software-Architekt: Wenn der Aufwand betrieben wurde, eine Software zur Marktreife zu entwickeln, dann ist der Aufwand marginal, sie für weitere Systeme zu releasen.

Selbst wenn man von einem Tag auf den anderen eine Komplettlösung mit quelloffener Software anbieten könnte, würde man eher auf Ablehnung stoßen, denn die alten Systeme laufen im Moment ja wunderbar. Dass man sich bei komplexen Systemen, deren Integration Stunden, Tage oder Wochen dauert, lieber auf der Momentaufnahme des laufenden Systems ausruht als die Migration auf ein potenziell besseres oder stabileres System in Kauf zu nehmen, ist so nachvollziehbar wie ignorant und führt zu einem destruktiven Trend: *Updatemüdigkeit*. Updates werden aus diffusen Ängsten nicht zeitnah installiert, wodurch Lücken länger offen bleiben als nötig [34, 26].

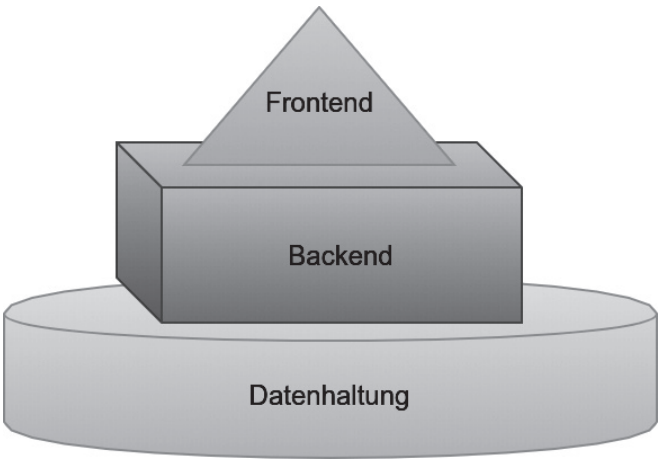
Schlechte Software trainiert uns an, Updates nicht zeitnah zu installieren, wodurch wir Sicherheitslücken aktiv offen halten Die gefühlte Wahrscheinlichkeit von schwerwiegenden Problemen durch die Updates ist höher als die, durch offene Sicherheitslücken bedroht zu werden. Der Schaden eines solchen Angriffs ist aufgrund der potenziellen Höhe schwer zu fassen, nicht nur wirtschaftlich [12], sondern insbesondere, wenn Menschenleben gefährdet sind.

Architektur moderner Software-Systeme

Windows hat maßgeblich zur Verbreitung von privaten Computern beigetragen. Als die Digitalisierung in Büros und Arztpraxen Einzug hielt, kam flächendeckend Windows zum Einsatz. Die anfangs durch Textverarbeitung und Tabellenkalkulation abgebildeten Arbeitsprozesse wurden schnell von Software-Entwicklern aufgegriffen und optimiert. Natürlich auf Windows-Basis. Access, Excel, Word. Eine exe-Datei ausgeführt und das Programm tut, was es soll. Genial. Skaliert nur nicht.

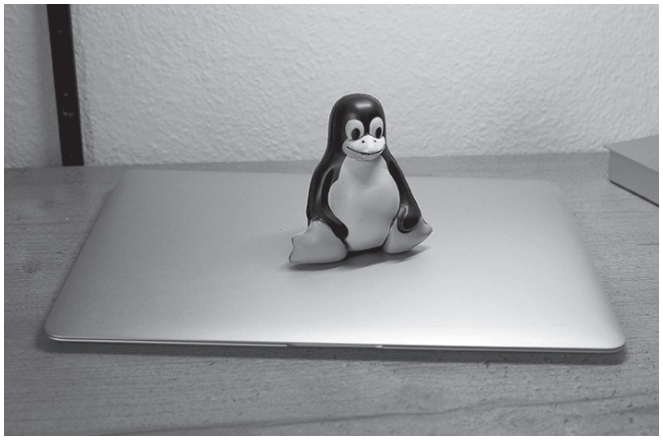
Was in der Verwendung einzelner Verwaltungskomponenten wunderbar funktioniert, lässt sich leider nicht 1:1 auf große Systeme übertragen. Dafür braucht man mehr Rechenleistung als ein normaler Praxiscomputer liefert. Deswegen wird fast alle neu entwickelte Software als verteiltes System aufgebaut. Die eigentliche Software läuft auf einem Server und der Nutzer startet ein Programm zur Visualisierung und Bearbeitung von Daten [30]. Die meisten dieser Server werden übrigens unter Linux betrieben [33]. Und exakt hier ist der Punkt, an dem Systemunabhängigkeit wieder ins Spiel kommt.

Während man noch vor zehn Jahren für jedes Zielsystem eigene Oberflächen bauen musste, ist das heutzutage weder notwendig noch sinnvoll, denn durch den HTML5-Standard wurde der größte Teil von Systemfunktionalität in einheitlichen Schnitt-



Architektur

stellen für Web-Oberflächen zur Verfügung gestellt. Zugriffe auf Systemkomponenten wie 3D-Beschleunigung und Verschlüsselung sind jetzt nativ, sicher und systemunabhängig implementiert [7]. Es gibt also keinen Grund mehr, neu entwickelte Applikationen nur für Windows zu entwickeln.



Tux, das offizielle Maskottchen des freien Linux-Kernels

Weiterführende Gedanken

Haben Sie irgendeinen Aspekt entdeckt, der Ihnen neu, seltsam oder ungemütlich vorkommt? Für mich ist das Ungemütlichste die Frage, ob ich als Mensch, der die Daten einer Vertrauensperson übergibt, ein Recht darauf habe, dass meine Daten entsprechend den Regeln der Informationssicherheit behandelt werden:

Wenn ich im Amt einen Ausweis beantrage und meine kompletten Daten in einen Windows-PC eingetragen werden, wer sagt mir, dass die Microsoft-Telemetrie diese Daten nicht nach Redmond schickt und dabei Geheimdienste und sonstige Trittbrettfahrer meine Daten ausspähen? Habe ich als Bürger die Verpflichtung, mich dieser Missachtung von Datenschutz zu unterwerfen, weil es eine Ausweispflicht gibt? Wenn ich eine medizinische Behandlung möchte, muss ich dann akzeptieren, dass mein Arzt meine Röntgenbilder in ein System einpflegt, das *insecure-by-design* ist [23]? Wenn ich nach einer Operation aufgrund eines Softwarefehlers mein Bein verliere, habe ich Anspruch auf Schadensersatz oder heißt es einfach „Softwareproblem. Kann man nichts machen?“[17] Warum messen wir als Gesellschaft mit zweierlei Maß, so dass wir bei fehlerhafter Hardware klagen, aber vermeidbare Softwarefehler hingenommen werden? Wie können wir das grundlegend ändern?

Wohin müssen wir gehen?

Bei allen aktuellen Problemen im Bereich der Software scheint es, als gäbe es keine Lösungen. Das stimmt so nicht. Wir müssen uns von unserer Veränderungsresistenz lösen und uns der Fehlbarkeit der Software-Entwickler bewusst werden. Aber wir müssen das intelligent anstellen. Softwarefehler werden immer vorkommen. Aber Softwarefehler können, sollen und müssen immer zurück zum Hersteller gegeben werden. Wenn sich bei Ihrem Auto der Beifahrersitz löst, sobald Sie einen bestimmten Radiosender einstellen, werden Sie das Auto auch zu Ihrem Händler zurückbringen. Warum machen Sie das nicht mit Software?

Oder sehen wir die andere Seite: Mein Arbeitgeber verpflichtet Software-Entwickler, einen Windows-PC zu verwenden. Auch wenn das mehr Probleme verursacht, die Arbeitsleistung darunter leidet und eine zentrale Verwaltung von Rechnern natürlich weitere Sicherheitslücken öffnet. Die Realität sieht leider so aus, dass man ein potenziell unsicheres System als Basis hat und mehrere Schichten an Antiviren-Software, Firewalls und sonstigen Softwarelösungen darauf legt, die meisten wiederum proprietär.

Wir müssen weg von dem Denken, dass Kosten und Sicherheit von Software zusammenhängen oder komplexere Software sicherer ist.

Wir müssen hin zu einer IT-Infrastruktur-Denkweise, die auf wissenschaftlichen Erkenntnissen basiert, anstatt unseren gesamten Datenschatz in die Hände von intransparenten Konzernen zu legen.

Wir müssen aufhören,

- kritische Infrastrukturen auf den Produkten von Herstellern aufzubauen, die mehr an Nutzerdaten Interessiert sind als an Stabilität und Datensicherheit, sich aber als *Heilsbringer-as-a-Service* präsentieren.

- unsere IT-Entscheidungen von Betriebswirtschaftlern treffen zu lassen, die sich von Marketing-Experten der Softwarehäuser beim Golfspielen von ihrer Unfehlbarkeit überzeugen lassen.

- nur High-Level Entwickler auszubilden, die nicht mehr verstehen, wie ein Computer funktioniert, sondern nur noch mit einem konkreten Framework umgehen können.

Wir müssen anfangen, die Softwarekosten nicht nur für Entwicklung, Bereitstellung und Wartung zu berechnen, sondern auch die Effizienz der Arbeit mit dem System zu verrechnen und natürlich den potenziellen Schaden bei Infiltration.

Vor allem müssen wir uns aber des zugrunde liegenden Problems bewusst sein. Wir müssen uns Softwarefehlern proaktiv nähern. Ist es ein Fehler? Dann muss er behoben werden. Habe ich etwas falsch gemacht? Vermutlich ein UX-Fehler, ergo sollte er behoben werden. Wir leben in einer Zeit von intuitiven Frontends und *User Experience (UX) Design* im Entwicklungsprozess.

Wir müssen aufhören, mehr und mehr Rechenleistung zu verbrauchen. Software wird heute mit übermäßig viel Framework-Unterstützung geschrieben und braucht dadurch viel mehr Rechenleistung und Speicher als nötig. Das wirkt sich auf den Energieverbrauch aus. Schlechte Software erzeugt also mehr CO_2. Wir müssen beginnen, die Nachhaltigkeit von Software zu bewerten.

Wir müssen aufhören, die Software als schlau und die Benutzer als dumm zu sehen. Software soll den Menschen unterstützen. Nicht ihm die Rechte nehmen, nicht ihn herausfordern.

Und: Wir müssen unser Wissen über Softwareprobleme teilen! Niemand sollte heutzutage gezwungen werden, Einschränkun-

gen oder Kompromisse bei Privatsphäre oder Datenhoheit einzugehen. Niemand sollte gezwungen werden, kostenpflichtige Software zu nutzen, um ins Internet zu gehen oder Briefe zu schreiben. Wir zerstören unser Verständnis von Qualität, Datenschutz und Vertraulichkeit. Das müssen wir stoppen.

Die Firmen und insbesondere der Healthcare-Sektor (nicht nur) in Deutschland geben mehr und mehr die digitale Autarkie aufgrund von Geschäftsbeziehungen und Innovationsangst auf. Durch die Schaffung von mehr Abhängigkeit wird weitere Abhängigkeit geschaffen. Deutschland wächst nicht durch die digitale Adoleszenz zu einem ausgewachsenen, IT-kompetenten Land heran, solange wir uns von Microsoft und Google den digitalen Schnuller geben lassen.

Literatur

[1] Ablon L, Bogart A (2017) Zero days, thousands of nights: the life and times of zero-day vulnerabilities and their exploits. RAND Corporation, ISBN 978-0-8330-9761-3

[3] Barrett B (2016) Hack Brief: Years-Old Linux Bug Exposes Millions of Devices. Wired. https://www.wired.com/2016/01/hack-brief-years-old-linux-bug/. ISSN 1059-1028. Zugegriffen: 7. September 2021

[4] Borchers D (2018) Cyber-Sicherheitspolitik: Fünf Prozent Zero-Day-Lücken für staatliche Überwachung von Kriminellen. Heise online. https://heise.de/-4072578. Zugegriffen: 28. Juli 2021

[5] Born G (2018) Windows 10: Microsoft erklärt die Update-Zyklen. https://www.borncity.com/blog/2018/08/02/windows-10-microsoft-erklrt-die-update-zyklen/. Zugegriffen: 7. September 2021

[6] Born G, Wittenhorst T (2021) Cloud-Datenbank-GAU: Microsoft informiert Azure-Kunden über gravierende Lücke. Heise online. https://heise.de/-6176601. Zugegriffen: 7. September 2021

[15] Bundesamt für Sicherheit in der Informationstechnik (BSI) (2020) Analyse der Telemetriekomponente in Windows 10 – Konfigurations- und Protokollierungsempfehlung (Version 1.2). https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/SiSyPHus/Analyse_Telemetriekomponente_1_2.pdf. Zugegriffen: 28. Juli 2021

[9] Eckert C (2012): IT-Sicherheit: Konzepte – Verfahren – Protokolle. 7. überarb. und erw. Aufl. Oldenbourg-Verl. ISBN 978-3-486-70687-1

[10] Einsiedler E (2021) Der EU-US Privacy Shield. Universität Linz, Dissertation. http://epub.jku.at/obvulihs/5833126. Zugegriffen: 7. September 2021

[11] Flade F (2020): Cyberangriff auf Bundestag: Haftbefehl gegen russischen Hacker. – URL https://www.tagesschau.de/investigativ/ndr-wdr/hacker-177.html. Zugegriffen: 7. September 2021

[35] Frankfurter Allgemeine Zeitung (2021): Landkreis liegt lahm: Erster Cyber-Katastrophenfall in Deutschland. In: FAZ.NET. – URL https://www.faz.net/-ikh-admez. ISSN 0174-4909. Zugegriffen: 7. September 2021

[12] Gebauer M (2021) Ransomware-Angriffe kosten im Durchschnitt 570.000 US-Dollar. Heise online. https://www.heise.de/-6158568. Zugegriffen: 7. September 2021

[28] Help Net Security (2021) Only 8back. https://www.helpnetsecurity.com/2021/04/28/ransom-paid/. Zugegriffen: 1. August 2021

[13] Holland M (2021) Vorwürfe an US-Klinik: Möglicherweise erster Todesfall wegen Ransomware-Attacke. https://www.heise.de/-6206624. – Zugegriffen: 2. Oktober 2021

[14] House, M (2021) Attributing Deaths to Ransomware Attacks on Hospitals and Medical Care Facilities. Yale Cyber Leadership Forum. https://bit.ly/3isyzy6. Zugegriffen: 7. September 2021

[8] Jung J (2021) Zwei Drittel der deutschen Unternehmen erleiden Ransomware-Attacken. ZDNet. https://bit.ly/3l1AYRU. Zugegriffen: 7. September 2021

[16] Kaspersky Labs (2020) Mobile Malware: weniger, aber gefährlicher. https://www.kaspersky.de/about/press-releases/2020_mobile-malware-weniger-aber-gefaehrlicher. Zugegriffen: 1. August 2021

[17] von Leitner F. Fefes Blog (Suche nach „Softwareproblem"). https://blog.fefe.de/?q=Softwareproblem. Zugegriffen: 28. Juli 2021

[18] Lich B (2021) Windows 10, Version 21H1, Windows 10, Version 20H2 und Windows 10, Version 2004 – erforderliche Diagnoseereignisse und -felder (Windows 10) – Windows Privacy. https://bit.ly/3Bl1xaB. Zugegriffen: 3. Oktober 2021

[19] Linux Mint (2019) New Features in Linux Mint 19.2 "Tina" Cinnamon Edition – Linux Mint. https://linuxmint.com/rel_tina_cinnamon_whatsnew.php. Zugegriffen: 3. Oktober 2021

[7] MDN contributors (2021) Web APIs. MDN. https://developer.mozilla.org/en-US/docs/Web/API. Zugegriffen: 3. Oktober 2021

[20] Muncaster P (2021) Ransomware Surge Drives 45% Increase in Healthcare Cyber-Attacks. Infosecurity Magazine. https://bit.ly/2Ylegvw. Zugegriffen: 7. September 2021

[21] Namestinov Y, Maslennikov D (2012) Kaspersky Security Bulletin 2012. The overall statistics for 2012. https://bit.ly/3uDdNAl. Zugegriffen: 1. August 2021

[22] Narduzzo A, Rossi A (2003) Modularity in Action: GNU/Linux and Free/Open Source Software Development Model Unleashed. ROCK Working Papers 020. Department of Computer and Management Sciences, University of Trento, Italy. https://ideas.repec.org/p/trt/rockwp/020.html. Zugegriffen 3. Oktober 2021

[23] Pegoraro R (2003) Microsoft Windows: Insecure by Design. Washington Post. https://wapo.st/3l4OlR8. ISSN 0190-8286. Zugegriffen 3. Oktober 2021

[2] Qualys Security Advisory (2021) CVE – CVE-2021-33909. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-33909. Zugegriffen: 28. Juli 2021

[24] Richardson R, North M (2017) Ransomware: Evolution, Mitigation and Prevention. International Management, Review 13, Nr. 1, S. 10–21

Mathias Haimerl

Mathias Haimerl ist Doktorand an der JKU Linz und der Technischen Hochschule Ingolstadt und forscht im Bereich *Inklusion im autonomen Verkehr.* Abseits der akademischen Tätigkeit ist er als Software-Architekt bei *Siemens Healthineers* beschäftigt.

[25] Sarr M (2017) Which OS Crashes Less Often: Mac OS X, Linux or Windows?. https://www.fossmint.com/which-os-crashes-less-often-mac-os-x-linux-or-windows/. Zugegriffen: 28. Juli 2021
[26] Schmidt J (2020) Exchange-Lücke: Immer noch viele Server offen. Heise online. https://www.heise.de/-4947221. Zugegriffen: 9. September 2021
[27] Schuppelius DJ (2020) DKB und Cloudflare. https://deinnetzwerkfachmann.de/allgemein/dkb-und-cloudflare/. Zugegriffen: 28. Juli 2021
[29] Spataro J (2021) Microsoft 365 für Unternehmenskunden: Neue Preise ab März 2022. News Center Microsoft. https://bit.ly/2YgjvMo. Zugegriffen: 22. August 2021
[30] Tanenbaum AS, v. Steen M (2002) Distributed systems: principles and paradigms. Prentice Hall. ISBN 978-0-13-088893-8
[31] Tremmel M (2020) Datenschutzbeauftragter: Windows 10 lässt sich ohne Telemetrie betreiben. Golem.de. https://glm.io/146423. Zugegriffen: 28. Juli 2021
[32] Trommler P (2000) The application profile model. Dissertation. Hochsch.-Verl. an der ETH. ISBN 978-3-7281-2739-6
[33] W3Techs. Usage statistics of operating systems for websites. https://w3techs.com/technologies/overview/operating_system. Zugegriffen: 28. Juli 2021
[34] Wittenhorst T (2018) Verschwundene Dateien: Microsoft zieht Windows-10-Update vorerst zurück. https://www.heise.de/-4182651. Zugegriffen: 28. Juli 2021

Markus Reinisch

„Revolutionen finden leibhaftig statt". Die Diskursräume *Straße, Netz* und die Protestlogik in Zeiten des Digitalen

Protestformen als Mittel demokratischer Partizipation spielen sich durch den Einfluss digitaler Medien immer mehr im Virtuellen ab. Trotz immer einfacherer Möglichkeiten, sich mittels sozialer Netzwerke zu organisieren und zu mobilisieren, bleibt der Wert des großen Nein (Armin Nassehi) auf Straßen und Plätzen nach wie vor von großer Bedeutung für die Protestierenden, ihre Anliegen und Sichtbarkeit. In der Öffentlichkeit ein Zeichen zu setzen, mit Gleichgesinnten seine Unzufriedenheit zu artikulieren, macht das Wesen von Protest aus. Zu seiner Logik gehören Gruppendynamiken und Identität, die Bilderproduktion sowie Narrative. Die Diskursräume Netz sowie Straße sollen im Folgenden mit Blick auf diese Aspekte der Protestlogik beleuchtet werden. Dabei gilt es auch Begriffe wie Engagement und Partizipation kritisch unter die Lupe zu nehmen, da sie unter dem Einfluss des Digitalen der Gefahr einer Umdeutung unterliegen. Ferner soll die Frage aufgeworfen werden, ob durch die Social-Media-Infrastrukturen und ihre Aufmerksamkeitsökonomie nicht sogar von einer De-Politisierung auszugehen ist.

Politische Debatten verlagern sich – auch ins Netz

2010 und 2011 erlebte die Welt einen bisher nicht gekannten Höhepunkt an sozialen Bewegungen und politischen Protesten, mit Themen aus verschiedenen Lebensbereichen. Die *Repolitisierung der Gesellschaft* (Edgar Forster) zeigte sich auf Straßen sowie auf Plätzen in vielen Ländern, oft durch Netzwerke in den sozialen Medien organisiert: in Hongkong demonstrierte man gegen den wachsenden Einfluss Chinas, in der arabischen Welt nahm die *Arabellion* an Fahrt auf, in New York gab es Protestcamps gegen den Banken- und Börsenkapitalismus (*Occupy Wall Street*), in Toronto eskalierten G-20-Proteste, in Griechenland und Weißrussland gingen Menschen auf die Straße, in Madrid forderten Tausende *Indignados* eine *Democracia Real Ya!* (echte Demokratie sofort).[1] Zahlreiche Disziplinen der Politik- und Sozialwissenschaft beschäftigen sich (erneut) verstärkt mit diesen Entwicklungen und stellen etwa heraus, welche identitätsstiftenden Merkmale eine Protestgruppe als *Kollektiv*, *Wertegemeinschaft*, *Schwarm* oder sonstige Gemeinschaft ausmacht, welche Dynamiken und Motive dabei vorherrschen oder welche Wirkungsabsichten damit auf ein Publikum (z. B. Politiker:innen) verfolgt und erreicht werden.[2] Dabei beobachten sie beispielsweise eine verstärkte „Verlagerung der politischen Debatte aus den Parteien in soziale Bewegungen und andere zivilgesellschaftliche Organisationen."[3] Und ins Digitale, möchte man ergänzen, wo sich auf verschiedenen Social-Media-Netzwerken neue Formen von Partizipation, Deliberation, Foren und Plattformen für Protestartikulation ergeben haben, die es zum Beispiel aus soziotechnischer Sicht kritisch zu betrachten gilt. Dabei kann unterschieden werden: es gibt Bewegungen, die im Netz entstanden sind und die politischen Debatten immer wieder durch wirkungsstarke Bilder auf der Straße prägten. *Pegida* in Deutschland, die *Gelbwesten* in Frankreich sowie *MeToo* sind nur drei Beispiele dafür. Umgekehrt legen Protestaktionen im öffentlichen Raum auch durch ihre Verbreitung und diskursive Verarbeitung im Internet enorme Mobilisierungskräfte frei. Diese Wechselwirkungen waren vor zehn Jahren beim *Arabischen Frühling* zu beobachten. Der Protestforscher Dieter Rucht spricht von einer „Demokratisierung der Demokratie", d. h. das demokratische System habe sich in seiner Qualität durch die gesellschaftlichen Polarisierungen und sonstigen Herausforderungen zu bewähren.

Neujahrsprotest der Hongkonger auf der Straße am 1.1.2020, Foto: Etan Liam, CC BY-ND 2.0