die datenschleuder.

das wissenschaftliche fachblatt für datenreisende
ein organ des chaos computer club

ISSN 0930-1054 • 2007
€2½
Kein Postvertriebsstück mehr ~~€11301F~~

#91

Liebe Chaoten, genauer gesagt: liebe Mitglieder einer neo-terroristischen Vereinigung. Ja! Ihr lest richtig. Dank eines Handstreichs unseres Staatssicherheitsministers haben wir geschafft, wovon Generationen von Hackern vor uns nur träumen konnten: Endlich auf Augenhöhe mit der Rote Armee Fraktion, den Damen und Herren von Al-Qaida, der Animal Liberation Front und dem Schwester-CCC in Belgien, den Cellules Communistes Combattantes.

Was ist passiert? In den Paragraphen 129a StGB, der die Bildung einer terroristischen Vereinigung unter Strafe stellt, sind Vereinigungen, deren Ziele den Straftatbestand der Computersabotage nach Paragraph 303b StGB umfassen, aufgenommen wurden. Zudem wird der Paragraph 303b so angepaßt, daß Verstöße gegen den Paragraphen 202c StGB, also grob gesagt der Umgang mit sogenannten „Hackertools", als Computersabotage gelten.

Betrachtet man dies zusammen mit den Überlegungen Schäubles, Terroristen vorläufig zu erschießen und gegen Terrorausbildungslager vorzugehen, muß die Einladung zum Chaos Communication Camp im August dieses Jahres mit einer Empfehlung einhergehen, schußsichere Westen, eine Rechtsschutzversicherung, genug finanzielle Polster für die Fehltage auf Arbeit und einen frischen orangen Ganzkörperanzug bereitzuhalten.

Da wir uns aber nicht einfach den totalitären Phantastereien beugen wollen, ruft die Redaktion zum kreativen zivilen Widerstand auf: Kommt zum Camp, diskutiert mit uns über Selbstverteidigungsmaßnahmen gegen Terror, Panikmache und Kriminalisierung beliebiger Vereinigungen, Verkehrsdaten-Vollerfassung und die Stasi 2.0.

Laßt am besten alle verkehrsdatenerzeugenden Geräte zuhause, solange euch – in eurer Eigenschaft als Gefährder – nicht sowieso schon der Besitz verboten wurde. Wo wir gerade dabei sind: Vermeidet das Unterfahren von Mautbrücken, hebt Bargeld sicherheitshalber noch zuhause und zwei Wochen vorher ab, bezahlt damit möglichst am Automaten eine Zugfahrkarte oder an der Tanke das Benzin. Mit dem Flugzeug Anreisende erkundigen sich bitte vorsorglich beim ortsansässigen Kostümverleih nach Bärten und Nasen (möglichst teutonischer und nicht arabischer Façon.)

Auf dem Camp selber sammelt alle Ausscheidungen in einem Beutel, laßt keine geruchsprobengeeigneten Kleidungsstücke zurück und benutzt Getränkebehältnisse mit rauhen, daktyloskopisch ungeeigneten Oberflächen.

Seid bitte nicht enttäuscht, wenn wir die von Outdoor-Veranstaltungen der letzten Monate gewohnten Zauninstallationen, Kampfflugzeugüberflüge und Legebatterie-Erlebnisschlafplätze nicht anbieten können. (Um ehrlich zu sein: Wir **haben** vergeblich versucht, die auf dem Campgelände ausgestellten MIGs aufsteigen zu lassen und Maschendraht ist elend teuer.)

Aber vor allem: Laßt euch von diesen Terroristen keinesfalls euer Verhalten diktieren! Lebt ein freiheitlich-demokratisches Leben ohne Angst und entfaltet euch im Rahmen der Verfassung und eurer Möglichkeiten. Denn SIE hassen uns für unsere Art zu leben und haben eigentlich schon gewonnen, wenn wir anfangen, uns mißtrauisch umzuschauen.

Now, bring me that revolution™! *<erdgeist>*

Inhalt

Hallo Herr Albert,

in Ihrem Artikel fehlt eine Aussge zu den mobilen Überwachungssystemen der BVG U-Bahn. In den Zügen der Linie 7 und 5 sind ca. 300 Geräte der Fa. DResearch verbaut:

http://www.dresearch.de/p/teleobserver_to3100_de.pdf

die per Eplus mit HSCSD an die Leidstelle angeschlossen sind.

http://www.dresearch.de/p/teleobserver_cmu_de.pdf

Bis zu 30 Alarme können parallel aufgenommen werden, siehe auch:

http://www.dresearch.de/company/press/material/release/pm_2006-02_knifer_de.pdf

Die gefühlte Sicherheit in den Zügen wurde damit deutlich verbessert.

Dieses System, was auch live mobil zu mobil-Übertragungen ermöglicht (H263+ mit bis zu 5fps in CIF-Farbe) ermöglichte der „schnellen Eingreiftruppe" bereits sehr erfolgreiche Einsätze. Daraufhin verlagerte sich die „Szene" auf andere Verkehrsmittel, die nun nachrüsten müssen. *<Michael Franke>*

Besten Dank für diesen Hinweis! Die Redaktion hat das an den Autor weitergeleitet. <FrankRo>

Wolfgang Wieland, 72. Sitzung, Bundestag

Dann sagt Kollege Kauder: Nun regt euch doch nicht auf, die Strafprozessordnung ist von 1877 und da gab es noch keine Hacker. So weit hat er Recht.

(Dr. Dieter Wiefelspütz [SPD]: Was? Das heißt „Häcker"!)

– Der Herr Staatssekretär hat es – die Amtssprache ist Deutsch – eingedeutscht.

(Ströbele [GRÜNE]: Ein Chaosclub ist diese Bundesregierung!)

Sie dürfen die englische Aussprache des Begriffs Hacker verwenden, aber ich bleibe bei der deutschen. Hacker gab es 1877 wirklich noch nicht.

Später...

Abschließend: Wir sind von dieser Regierung einiges Chaos gewohnt. Aber es ist doch etwas anderes, wenn unsere Strafverfolgungsorgane nun so handeln wie der Chaos Computer Club. Das wollen wir nicht. Es gibt schärfsten Protest von unserer Seite.

(Beifall beim BÜNDNIS 90/DIE GRÜNEN, bei der FDP und bei der LINKEN – Zuruf von der SPD: Hacker Wieland! – Dr. Dieter Wiefelspütz [SPD]: Keine Ahnung! – Hans- Christian Ströbele GRÜNE]: Das ist vielleicht eine Regierung!)

http://dip.bundestag.de/btp/16/16072.pdf

Interessantes Geschenk

Ein hoher Telekomchef schickte mir als kleine Entschädigung für den schlechten Service von T-Com einen 512 MB - USB-Stick.

Ich schloß ihn an und staunte nicht schlecht, als ich die im Anhang befindlichen pdf. Dateien darauf entdeckte. Meine Neugier war geweckt und es sind vertrauliche Daten zur Auswertung einer Studie. Solch unerwartete Geschenke erfreuen mich zwar, aber es ist ein weiterer Beweis dafür, dass die Telekom ein „kleines" Sicherheitsproblem besitzt.

Auf persönlichem Wunsch eines Kumpels soll ich euch das zuschicken. *<Linus OSB>*

Die vertraulichen Dokumente...

...die an die Mail angehangen waren, haben in der Redaktion große Heiterkeit ausgelöst, werden aber aus Platzgründen eventuell in einem eigenen Sonderheft veröffentlicht. <erdgeist>

Zu unserem 23C3-Aufruf

Verwundert habe ich mir die Augen gerieben, als zu einer verstärkten Überwachung von „Problempolitikern" angemahnt wurde (heise.de). Das ist klasse!

Hintergrund: Wir Sportpiloten werden seit geraumer Zeit als „größtmöglich Gefahr" betitelt und müssen nach den Wünschen der Innennminister bzw. Regierung unsere Grundrechte aufgeben und unsere eigene Unschuld beweisen, daß wir keine Gefahr für die Gesellschaft darstellen.

Wie es aussieht gehen die Interessen in die gleichen Richtungen. Könnten Sie sich ggf. eine Art „Zusammenarbeit" auf dieser Ebene vorstellen? Es geht uns darum, gemeinsam gegen den Irrsinn der Politiker vorzugehen. *<frank k.>*

Vielleicht sollten wir ihn darauf hinweisen, daß Piloten wohl am ehesten in der Lage sind, die amtierende Regierung innerhalb von einer Minute auszutauschen... <FrankRo>

Wenn das so einfach wäre, gäbs das doch bestimmt längst zum Clicken...

Müßte unbedingt eine Handy Nummer herausbekommen-ich weiß,daß das über die Betreiber fast unmöglich ist. Vielleicht könnt Ihr mit einer Adresse weiterhelfen-soll auch nicht umsonst sein (> Spende) <PRIVATE XXX@web.de>

„Wir sind die Guten. Wir machen so etwas nicht." <padeluun>

Hilfe bei Crackern

mein Name ist XXX. Wir haben Cracker auf unseren Computern und Funktelefonen. Ich habe die Angelegenheit der Polizei gemeldet, die aber nicht gewillt ist, etwas zu tun.

Wer kann mir in YYY eine Firma oder eine Privatperson nennen, die uns helfen könnte. <XXX@web.de>

Ihr könnt euch selbst helfen: Einfach aufessen.

Mit einem Dip schmecken Cracker oft besonders gut. <Alexander>

Fundunterschlagung...

bei Recherchen im Internet bin ich am 12.9. auf eine Seite eines großen Internetportals, mit nach eigenen Angaben 28 Mio. Nutzern, gestoßen. Dort stand unverschlüsselt eine lange Namensliste mit Adresse, Telefonnummern und noch viel wichtiger: Den Bankverbindungen inkl. Kontonummer und Kreditkartennummer dieser Leute! Ein gefundenes Fressen für alle, die sich auf fremde Rechnung Sachen bestellen und fremde Konten leerräumen wollen.

Ich schickte daraufhin emails an das Internetportal und an die Polizei. Erst am 15.9. bekam ich eine Reaktion von beiden Stellen, und erst am 18.9. war diese Seite nicht mehr aufrufbar. Wie lange die Seite schon im Internet war und ob den Leuten auf dieser Liste Schaden entstanden ist, kann ich nicht sagen. Da müßte die Polizei ermitteln.

Im Zusammenhang an diese Vorgänge habe ich eine Frage an das BSI: Steht mir eigentlich von dem Internetportal so etwas wie ein „Finderlohn" zu? Immerhin kann das Bekanntwerden solcher eklatanter Sicherheitslücken einen erheblichen Imageschaden mit schweren finanziellen Einbußen nach sich ziehen. <XXX@t-online.de>

Nein. <padeluun>

KreditkartenDatenübermittlung

Werden Klarnamen (also Vor und Zuname) beim benutzen eines Geldautomaten einer anderen Bank von der Kreditkarte mit übertragen oder sind die Daten so verschlüsselt, dass nur die Bank, wo das belastete Konto geführt wird, den Karteninhaber identifizieren kann? <Wieland>

Hallo, zur Autorisierung wird der komplette „Track 2", also die 2. Datenspur auf dem Magnetkartenstreifen uebermittelt (zusaetzlich zur PIN). In diesem ist der „Cardholder Name" enthalten. (2 - 26 Zeichen incl. Vor- und Nachname sowie Titel, falls vorhanden).

Der Aufbau der „Track 2" ist unter anderem auch in der ISO/IEC 7813 „Information technology - Identification cards - Financial transaction cards" nachlesbar. <HonkHase>

InternetRatten

Gibt es Leute, Clubs oder Vereine die Internetratten (wie Firma Schmidtlein GbR) das handwerk legt? <Ronald>

Internet-Ratten, schöner Begriff..

Es gibt einen Verein, der was tut: Der Generalstaatsanwalt in Frankfurt leitet knapp 3000 Verfahren, die mittlerweile eingegangen sind.

http://www.hna.de/politikstart/00_20061108181900_Wie_Fischen_mit_Dynamit.html <FrankRo>

Mal etwas größer gedacht: Hoteldiebstahl vereitelt

Bereits Ende Dezember 2006 wurde in New York ein Mann wegen Hoteldiebstahls angeklagt; er hatte versucht, durch entsprechende Dokumentenbeschönigung zu seinen Gunsten den Grundbucheintrag des Grandhotel in Soho zu überschreiben. Details unter:
http://wcbstv.com/topstories/local_story_363144841.html

BTX-Hack jetzt bei Google Video

Den legendären Bildschirmtext-Hack des Chaos Computer Club gibt es jetzt bei Google Video in einer Zeitreise nachzuvollziehen:
http://video.google.com/videoplay?docid=-8396178892678063881

Puffmutter erkärt US-Politikern Vorratsdatenspeicherung

...und darf nach einem Gerichtsprozeß nun ihre langjährigen Telefonverbindungsdaten eines von ihr betriebenen Call-Girl-Ringes veröffentlichen. Übrigens mit der Begründung, daß eine Veröffentlichung ja auch im Sinne ihrer Kunden sei, die ob ihrer politischen und sonstigen Funktionen andernfalls Erpressung fürchten müßten. Details:
http://news.yahoo.com/s/ap/20070706/ap_on_re_us/escort_list

Länder sind auch nur Spielkinder: Chinesen haben Anti-Satelliten Waffen

Details siehe:
http://www.aviationweek.com/aw/generic/story_generic.jsp?channel=awst&id=news/CHI01177.xml

Zur Sinn der Reproduktion von Gewaltbildern

Ein zehnjähriger Junge in den USA hat sich Anfang Januar offenbar beim „Nachspielen" der Exekution Saddam Husseins aus Versehen erhängt.

Stichworte: Fernsehen, Medienkompetenz, USA
http://www.spiegel.de/panorama/0,1518,457899,00.html

Nächste Kategorie: Nach SpyWare kommt jetzt GovWare

Mit dem Bundestrojaner hat jetzt auch die Kategorie GovWare ihren festen Bestandteil in der Kategorie-Liste der Schadprogramme eingenommen. Technische Details u. a. in
http://www.heise.de/newsticker/meldung/83538

Besonders absurd anmutend auch der Versuch eines Juristen, einen "Verhaltensleitfaden bei Online-Durchsuchungen" zu erstellen:
http://www.jurblog.de/2007/02/07/verhaltensleitfaden-bei-online-durchsuchungen/

Ein hierzu beachtenswertes Angebot unter:
http://www.bundestrojaner.net/

Bemerkenswert ist allerdings auch der Hinweis auf den Bundestrojaner und ein (reales!) Fallbeispiel einer hierdurch untergeschobenen Straftat:
http://karlweiss.twoday.net/stories/3314674/

Verbrecher pro Einwohner: Vatikan führt

Der Vatikan ist statistisch betrachtet der Staat mit den meisten Verbrechern pro Einwohner. Ob das nun auf die geringe Anzahl von Einwohnern oder dem Prinzip organisierter Religion zurückzuführen ist, sei dem Leser überlassen:
http://www.spiegel.de/international/0,1518,460967,00.html

Zum Abhörskandal in Griechenland

...hat die IEEE Spectrum den Versuch unternommen, mal eine strukturierte technische Analyse zu erstellen:
http://www.spectrum.ieee.org/print/5280

Stasi I: eine FAQ zu konspirativen Wohnungen

http://www.stasi-in-erfurt.de/Wohnungen-FAQ.htm

Stasi II: eine lustige Studie

....über die Beschäftigung ehem. MfS-Angehöriger bei der BStU, wobei die Geschichte mit dem besonders günstigen Gebäudereinigungsdienst in den Räumen der damals noch „Gauck-Behörde" genannten Institution noch gar nicht vorkommt...

http://www.stasi-in-erfurt.de/Wohnungen-FAQ.htm

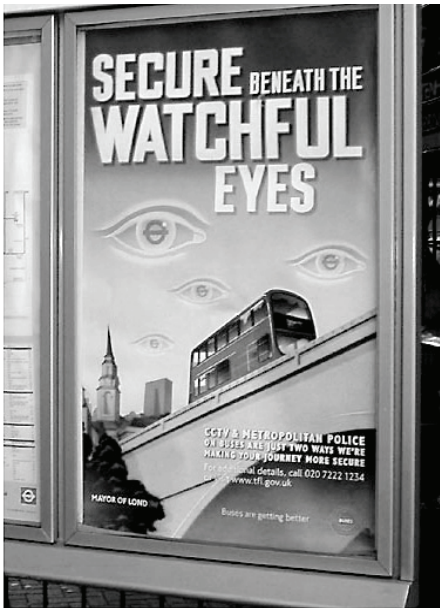
Stasi 2007

Entwurf eines Gesetzes zur Neuregelung der Telekommunikationsüberwachung und anderer verdeckter Ermittlungsmaßnahmen...

http://dip.bundestag.de/btd/16/058/1605846.pdf

Intelsat hat Stress mit Satelliten-Kaperern aus Sri Lanka

http://www.dailynews.lk/2007/04/13/news01.asp



An actual British government poster outside a London Metro station.

Scary Shit: GPS-Empfänger in (!) GSM-SIM-Karte

...und dann auch noch Antennenfunktion via SIM-Toolkit:

http://www.blueskypositioning.com/products.htm

Bald auf Ebay: die Bundesdruckerei

Auch im Jahr 2007 befindet sich die Bundesdruckerei noch nicht auf dem Weg zur Rückverstaatlichung, obwohl dies angesichts der dort verarbeiteten Daten, die ja – wie aus gewöhnlich gut unterrichteten Kreisen verlautete, bereits bei der ersten Privatisierung verlorengingen – wohl naheliegen würde.

Eine Momentaufnahme:

http://www.heise.de/newsticker/meldung/83189

http://www.finanztreff.de/ftreff/news.htm?id=26826261&r=0&sektion=nachrichten&awert=&u=0&k=0

Geldinstitute informieren pro-aktiv über Kartensicherheit

Nachdem sich die deutschen Geldinstitute etwa 20 Jahre darin geübt haben, das Phänomen der Manipulation von Geldautomaten bzw. von Geldautomatenkarten kleinzureden oder vollständig zu leugnen, haben die Delikte seit einiger Zeit offenbar eine Dimension angenommen, die eine pro-aktive Informierung der Öffentlichkeit zur Erkennung manipulierter Geldautomaten etc. erfordern. Herzzerreißende Zitate nach dem Motto „Im Zweifelsfall könnten Bankkunden daher durch leichtes Rütteln feststellen, ob es sich um einen echten Geldautomaten handle" finden sich unter:

http://www.kartensicherheit.de/

Sprach- und Sprechererkennung offenbar Standard in LI-Systemen

In Mexico City hat Verint offenbar einmal alles aus dem Regal rechts unten geliefert; ein interessanter Einblick unter:

http://www.latimes.com/news/nationworld/world/la-fg-mexico25may25,0,7011563.story?coll=la-home-center

Erfa-Kreise / Chaostreffs

Bielefeld, CCC Bielefeld e.V., Bürgerwache Siegfriedplatz
freitags ab 20 Uhr *http://bielefeld.ccc.de/ :: info@bielefeld.ccc.de*

Berlin, CCCB e.V. (Club Discordia) Marienstr. 11, (✉ CCCB, Postfach 64 02 36, D-10048 Berlin),
donnerstags ab 17 Uhr *http://berlin.ccc.de/ :: mail@berlin.ccc.de*

Darmstadt chaos darmstadt e.V. TUD, S2|02 E215
dienstags ab 20 Uhr *https://www.chaos-darmstadt.de :: info@chaos-darmstadt.de*

Dresden, C3D2/Netzbiotop e.V., Lingnerallee 3, 01069 Dresden
dienstags ab 19 Uhr *http://www.c3d2.de :: mail@c3d2.de*

Düsseldorf, CCCD/Chaosdorf e.V. Fürstenwall 232, 40215 Düsseldorf,
dienstags ab 19 Uhr *http://duesseldorf.ccc.de/ :: mail@duesseldorf.ccc.de*

Erlangen/Nürnberg/Fürth, BitsnBugs e.V. "E-Werk", Fuchsenwiese 1, Gruppenraum 5
dienstags ab 19 Uhr *http://erlangen.ccc.de/ :: mail@erlangen.ccc.de*

Hamburg (die Dezentrale) Lokstedter Weg 72
2. bis 5. Dienstag im Monat ab etwa 20 Uhr *http://hamburg.ccc.de/ :: mail@hamburg.ccc.de*

Hannover, Leitstelle 511 e.V., c/o Bürgerschule Nordstadt, Schaufelder Str. 30, 30167 Hannover
Jeden 2. Mittwoch und jeden letzten Dienstag im Monat, ab 20 Uhr *https://hannover.ccc.de/*

Karlsruhe, Entropia e.V. Gewerbehof, Steinstr. 23
sonntags ab 19:30 Uhr *http://www.entropia.de/ :: info@entropia.de*

Kassel Uni Kassel, Wilhelmshöher Allee 71-73 (Ing.-Schule)
1. Mittwoch im Monat ab 18 Uhr *http://kassel.ccc.de/*

Köln, Chaos Computer Club Cologne (C4) e.V. Vogelsanger Strasse 286, 50825 Koeln
Letzter Donnerstag im Monat ab 20 Uhr *https://koeln.ccc.de :: mail@koeln.ccc.de*

München, muCCC e.V. im Buergerbuero Memmel, Melusinenstr. 18
2. Dienstag im Monat ab 19:30 Uhr *http://www.muc.ccc.de/*

Ulm Café Einstein an der Uni Ulm, montags ab 19:30 Uhr *http://ulm.ccc.de/ :: mail@ulm.ccc.de*

Wien, chaosnahe gruppe wien Kaeuzchen, 1070 Wien, Gardegasse (Ecke Neustiftgasse)
Alle zwei Wochen, Termine auf Webseite *http://www.cngw.org/*

Zürich CCCZH, c/o DOCK18, Grubenstrasse 18 (✉ Chaos Computer Club Zürich, Postfach, CH-8045 Zürich),
abwechslungsweise Di/Mi ab 19 Uhr *http://www.ccczh.ch/*

Aus Platzgründen können wir die Details aller Chaostreffs hier nicht abdrucken. Es gibt aber in den folgenden Städten Chaostreffs mit Detailinformationen unter *http://www.ccc.de/regional/*: Aachen, Aargau, Augsburg, Bad Waldsee, Basel, Bochum, Brugg, Dortmund, Frankfurt am Main, Freiburg im Breisgau, Gießen/Marburg, Hanau, Heidelberg, Ilmenau, Kiel, Leipzig, Mainz, Mülheim an der Ruhr, Münster/Osnabrück, Offenbach am Main, Paderborn, Regensburg, Stuttgart, Trier, Weimar, Wetzlar, Wuppertal, Würzburg.

Zur Chaosfamilie zählen wir (und sie sich) die Häcksen (*http://www.haecksen.org/*), den FoeBuD e.V. (*http://www.foebud.org/*), den Netzladen e.V. in Bonn (*http://www.netzladen.org/*) und die c-base Berlin (*http://www.c-base.org/*).

Die Datenschleuder Nr. 91

Herausgeber (Abos, Adressen, Verwaltungstechnisches etc.)

Chaos Computer Club e.V., Lokstedter Weg 72,
20251 Hamburg, Fon: *+49.40.401801-0*,
Fax: *+49.40.401801-41*, <*office@ccc.de*> Fingerprint:
1211 3D03 873F 9245 8A71 98B9 FE78 B31D E515 E06F

Redaktion (Artikel, Leserbriefe, Inhaltliches, etc.)

Redaktion Datenschleuder, Pf 64 02 36, 10048 Berlin,
Fon: +49.30.28097470, <*ds@ccc.de*> Fingerprint:
03C9 70E9 AE5C 8BA7 42DD C66F 1B1E 296C CA45 BA04

Druck

Pinguindruck Berlin, *http://pinguindruck.de/*

ViSdP, Layout und Produktion

Dirk Engling <*erdgeist@erdgeist.org*>

Redaktion dieser Ausgabe

Pavel Mayer, Jane R. Hacker, Constanze Kurz, "Herr Weber", Stephanie Lange, Dexter, Christian Berger, Martin Haase, Bjoern Pahls, Henrik Heigl, Stefan Schumacher, Andy Müller-Maguhn.

Tödliche Sicherheit

Pavel Mayer <pavel@ccc.de>

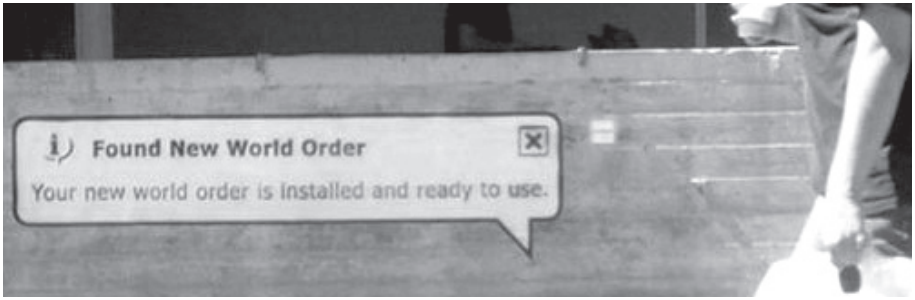
Sinnlose Sicherheitsmaßnahmen im Luftverkehr kosten über tausend Menschenleben an Zeit pro Jahr – eine Angleichung von Sicherheitsmaßnahmen im Luftverkehr an die bestehenden Sicherheitsmaßnahmen im Bus- und Bahnverkehr kann hier Abhilfe schaffen.

Sicherheit durch dummes Geschwätz?

Nach dem Terrorplot von London und den Zugbomben in Deutschland wird allerorten der Ruf nach mehr Kontrollen laut: Gepäckdurchleuchtung auf Bahnhöfen, mehr Kameras, mehr Datenschnüffelei, Verbot von Wasser und Laptops in Flugzeugen und andere Grausamkeiten gegen Reisende werden derzeit als Mittel gegen den Selbstmordattentäter und sonstige Terroristen diskutiert. Allerdings tun sich die Sicherheitsrhetoriker der Unionsparteien vor allem mit völlig unpraktikablen Vorschlägen hervor, wie etwa CDU-"Innenexperte" Clemens Binninger, der Rail-Marshalls (aka bewaffnete Zugbegleiter) einführen will und offenbar noch nie was von der Bundespolizei gehört hat, die seit 1992 die Aufgaben der ehemaligen Bahnpolizei und der Transportpolizei der DDR wahrnimmt und auch für die Sicherheit auf Flughäfen zuständig ist. Einfach eine Milliarde für ein paar tausend zusätzliche Beamte, und schon sind die Züge sicher, dank Röntgenblick unserer neuen Super-BuPos. Blöd nur, wenn die Bombe gar nicht im Zug, sondern im LKW am Bahnübergang ist. Egal, dann stellen wir noch mal fünfhunderttausend Transportbegleiter beim Bundesamt für Güterverkehr ein, das sollte sich doch aus der Autobahnmaut finanzieren lassen – oder wie hat der Herr Innenexperte sich das vorgestellt?

Komplett unrealistisch und damit wirkungsfrei ist der Vorschlag von Norbert Geis, der "die gleichen Sicherheitsmaßnahmen in Zügen wie derzeit in Flugzeugen" fordert, was definitiv das Ende des Zugnahverkehrs bedeuten würde, denn wer würde noch S-Bahn fahren, wenn er eine halbe Stunde vor Abfahrt da sein müsste. Auf so ewas kann man wohl nur kommen, wenn man in Kleinkahl-Edelbach wohnt und in Berlin nur Limousine fährt.

Unions-Fraktionsvize Wolfgang "Susanne-Osthoff-soll-zahlen" Bosbach wirft immerhin die zumindest nicht völlig unvernünftige Wiedereinführung der Kronzeugenregelung in die Diskussion, was wohl den Ausstieg aus einer terroristischen Vereinigung erleichtern soll. Die Effektivität bei Selbstmordattentätern und bei in sehr andersartigen Gesellschaften verankerten Menschen darf wohl bezweifelt werden, aber für einheimische politische Extremisten und Mafiosi kann es was bringen, und es ist vielleicht auch was für pakistanische Migran-



tenkinder, mit denen die Engländer offenbar Probleme haben. Allerdings ist die Wiedereinführung der Kronzeugenregelung eh im Koalitionsvertrag von 2005 vereinbart, insofern verdient diese Forderung das Prädikat "Größtenteils harmlos".

Ob nun Terror-Trittbrettfahrer mit politischem Kalkül oder hilflose Rechtspopulisten, es sind auch die hirnarmen Sensationsjournalisten, die den Schwachsinn verbreiten und die sich zu Komplizen der Terroristen machen, indem sie unter Ausblenden von Tatsachen auflagenfördernd Angst und Schrecken verbreiten, ohne aber die Sicherheit auch nur im geringsten zu erhöhen, sondern im Gegenteil Menschenleben vernichten, indem als Folge sehr viele Menschen Lebenszeit und Geld mit nutzlosen Pseudosicherheitsmaßnahmen verschwenden.

Sinnlose Passagierschikanen nach 9/11

Nach den Flüssigsprengstoff-Terrorplanungen von London ist klar: die seit 9/11 verschärften Sicherheitsmaßnahmen am Flughafen hätten niemanden daran hindern können, Flüssigsprengstoff an Bord eines Flugzeugs zu bringen, und die neuen Maßnahmen können es vielleicht erschweren, was aber irrelevant ist, weil die Sicherheitsmaßnahmen auch niemanden daran hindern, festen Sprengstoff an Bord zu bringen, sofern er nur ordentlich verarbeitet ist, etwa als Schuh, Baumwollsakko, Laptopbatterie, iPod oder Märchenbuch. Sie hindern auch niemanden daran, mit einer Boden-Luft Rakete ein startendes Flugzeug abzuschießen, oder einen Flugkapitän daran, sich zu entscheiden, in ein Kernkraftwerk oder vollbesetztes Stadion zu fliegen, oder eine Privatmaschine gegen eine Boeing 747 im Landeanflug zu steuern. Und obwohl die Kontrollen am Flughafen keine Sicherheit gegen all die genannten Szenarien bieten, ist Fliegen immer noch sicherer als jede andere Art der Fortbewegung einschließlich zu Fuß gehen. Fliegen ist mittlerweile einfach zu sicher. Das liegt wohl zum einen daran, daß bereits der Sturz aus relativ geringer Höhe für Menschen fatal ist und dadurch allein Höhe bereits Todesangst auslösen kann. Zum anderen war Fliegen früher zunächst nur etwas für vermögende Privatleute, Politiker und Geschäftsreisende, und somit war das Flugzeug als Verkehrsmittel der Elite besonders beliebtes Anschlagsziel, und aus taktischen Gründen wie seiner Beweglichkeit und gut kontrollierbaren Zugänge beliebtes Ziel für Entführungen.

Mittlerweile fliegen aber breite Bevölkerungsschichten, und die Entführungen vom 11. September 2001 haben die Schwelle für Flugzeugentführungen wesentlich erhöht: Arabisch sprechende Entführer müssen nun mit einem gewaltsamen Aufstand der Passagiere rechnen. Allerdings gibt es seit den sechziger Jahren pro Jahr 20-30 Flugzeugentführungen, und vor allem der Luftverkehr zwischen Cuba und den USA scheint regelmäßig gegen den Willen des Piloten zu erfolgen, auch nach 9/11/2001, seit den 1980er Jahren aber überwiegend von Cuba in die USA, während in den 60er und 70er Jahren vor allem Amerikaner Flugzeuge nach Cuba entführt haben.

Flugzeuge stellen auch aufgrund ihrer vermeintlichen Fragilität sicher noch immer ein bevorzugtes Terrorziel dar, aber Nahverkehrszüge und Busse stehen dem nicht viel nach. Da sich beide nicht wirkungsvoll und mit vertretbarem Aufwand schützen lassen, stellt sich die Frage, warum eigentlich der Reisende im Luftverkehr derartigen Behinderungen und Belästigungen ausgesetzt wird, denn die Zahl von Flugzeugentführungen in den letzen 30 Jahren hat sich durch mehr Kontrollen nur unwesentlich verändert, allenfalls verlagert.

Die Kosten der Kontrollen

Die Ärgernisse und Kosten wirkungsloser Sicherheitsmaßnahmen wie Schlangestehen, Tasche auspacken, Hände vom Körper halten, Schuhe ausziehen, Gürtel umdrehen, Abtasten lassen und eventuell den ganzen Kram auspacken, das gibt es nur im Gefängnis und am Flughafen. Nicht beim Militär, nicht beim Passieren des eisernen Vorhangs im kalten Krieg, ja nicht einmal, wenn man den Kanzler oder den Bundespräsidenten treffen will, wird man einer derart entwürdigenden Behandlung ausgesetzt.

Der Verlust von Zeit, Geld und Lebensqualität durch die Kontrollen ist unverhältnismäßig geworden. Bei rund 18 Mio. Starts pro Jahr werden pro Flug ca. 150 Passagiere befördert, das sind 2,7 Mrd. Einsteigevorgänge pro Jahr. Könnte jeder dieser Einsteigevorgänge um 15 Min. verkürzt werden, wären das 675 Mio. Stunden nutzbarer Lebenszeit. Ein Mensch hat in seinem Leben rund 500.000 wache Stunden; die Sicherheitskontrollen kosten also jährlich allein die Zeit von weit über tausend Menschenleben, die Zeit des Sicherheitspersonals nicht mitgerechnet. Rechnet man noch Kosten von ca. $5 je Einsteigevorgang hinzu, sind mindestens 10 Mrd. Dollar verschwendet. Taxiert man nun zugebenermassen etwas zynisch, aber realistisch den Wert eines Menschenlebens (weißer Amerikaner aus der Mittelschicht) auf 10 Mio. Dollar (Lybien hat 2,7 Mrd. für die 270 Toten von Pan Am 103 gezahlt), so kommt der Gegenwert weiterer 1000 Menschenleben hinzu. Vermutlich müssten Terroristen alle vierzehn Tage ein Flugzeug vom Himmel holen, um denselben Effekt zu erreichen, vorausgesetzt, die Menschen würden sich dadurch nicht davon abhalten lassen, einfach weiter zu fliegen, denn das Flugzeug bliebe auch dann noch das sicherste Verkehrsmittel.

Die Kosten des Terrors

Im Durchschnitt sind von 1968-2003 etwa 411 Menschen/Jahr Terroranschlägen zum Opfer gefallen, das ist etwa 1% der jährlichen Verkehrstoten in den USA, und allein in der Schweiz bringen sich jährlich dreimal so viele Menschen selbst um, wie weltweit dem Terror zum Opfer fallen. Allerdings sind Terroranschläge in einem Land Gift für Tourismus (-50%), ausländische Direktinvestitionen (-10 bis -15%), pro-Kopf Konsum (-5%), Investitionen (15-30%), Aktienmarkt (-10% Ertrag), Außenhandel (-4%) und damit Volkseinkommen (-10%) und Wachstum. Andauernder Terror kann dadurch ein Land in seiner Entwicklung erheblich behindern, die Zahlen sind aus gut untersuchten Beispielen im Baskenland, in Griechenland und vor allem Israel, das ohne massive amerikanische Wirtschafts- und Militärhilfe wohl nicht lebensfähig wäre.

Bekämpfung des Terrors: die Praxis

Der viel größere Schaden entsteht also durch die psychologischen Effekte des Terrors, aber die sind umso größer, je mehr Beachtung man dem Terror schenkt. Die effektivste Art, den Terror zu bekämpfen, wäre ihn einfach zu ignorieren.

Leider ist das Gegenteil der Fall. Insbesondere die Maßnahmen nach 9/11 sind an Absurdität kaum zu überbieten: Hunderttausende von Nagelscheren und Taschenmessern landeten im Müll, obwohl allen hätte klar sein müssen, das eine Wiederholung der Vorfälle vom 11. September ausgeschlossen ist, solange nicht mindestens die Hälfte der Passagiere Terroristen sind: anderenfalls wären Entführer nicht in der Lage, sich allzu lange zu behaupten, da die Passagiere davon ausgehen müssen, bereits eh tot zu sein und ohne Rücksicht auf Verluste gegen die Entführer vorgehen werden. Da vermutlich mehr als 99,99999% aller Fluggäste harmlose Reisende sind, wäre es nur konsequent, diese Reisenden in die Lage zu versetzen, sich besser gegen etwaige Entführer zur Wehr zu setzen, statt ihnen demnächst vielleicht noch ihre Kleidungsstücke wegzunehmen. In Deutschland besitzen rund 10% aller Haushalte legal eine Schußwaffe, und es gibt vermutlich zehn Millionen illegale Schußwaffen in Deutschland. Dennoch habe ich in meinem Leben noch keine Schießerei im Zug oder in der Stadt beobachtet, trotz fehlender Metalldetektoren. Wozu also der Quatsch am Flughafen? Zumindest auf innereuropäischen Flügen ist es absolut überzogen, jeden Passagier zu kontrollieren; zur Durchsetzung eines Verbots von Waffen und gefährlichen Gegenständen reichen Stichproben völlig aus, so wie bei jeder vernünftigen Grenzkontrolle.

Die heutigen Kontrollen an Flughäfen sind bereits strenger, als die Kontrollen bei der Überquerung des eisernen Vorhangs zu Hochzeiten des kalten Krieges je waren. Die Reisezeit auf innerdeutschen Flügen erhöht sich um bis zu 50%, weil Gepäck und Passagiere sicherheitsgeprüft werden. Warum nicht bei Non-Stop Flügen einfach mit Koffer zum Flieger, wie bei Zug oder Reisebus auch, und das Gepäck wird direkt verstaut? Fliegen so bequem und sicher wie Zugfahren: das wäre doch was. Das einzige, wozu die Kontrollen an Flughäfen führen, ist, daß nunmehr große Bomben in Zügen platziert werden. Wo ist da der Sicherheitsgewinn für die Gesellschaft? Und wenn Züge kontrolliert würden, dann kommen die Bomben eben in Busse. Und wenn Busse kontrolliert werden, dann eben Volksfeste, Konzerte, Schulen, Wohnhäuser, Hotels, Campingplätze oder Arbeitsagenturen. Wohin der jetzige Weg führt, kann man an Israel sehen. Oder in Bagdad.

Bekämpfung des Terrors: die Theorie

Was also tun? Den Terror zu ignorieren wäre zwar vernünftig, aber Menschen sind bekanntermaßen nicht vernünftig und wollen sehen, das etwas unternommen wird. Es wäre sicher gut, den Menschenrechten weltweite Achtung zu verschaffen, denn Terror speist sich vor allem aus Folter, gewaltsamer Unterdrückung, Rassismus und Gier. Leider gibt vor allem die Großmacht USA weiterhin dem Terror reichlich Nahrung. Das muß aufhören. Allerdings würde es wohl mindestens fünfzig Jahre dauern, bis die Früchte davon spürbar würden, und das ist nicht nur viel länger als eine Legislaturperiode, sondern auch zu lang für die Wähler, und selbst dann wird Terror nicht völlig verschwinden, irgend eine extremistische Gruppe wird wohl immer unzufrieden mit dem Status Quo sein und eventuell zum Terror greifen.

Es gibt einen interessanten sozialwissenschaftlichen Rationalansatz, der weitere Möglichkeiten aufzeigt, als Terrorismus stupide mit Drohungen, Sanktionen und den Einsatz von Polizei und Militär zu bekämpfen, wodurch im Endeffekt eine Gewaltspirale weiter genährt wird, solange man nicht bereit ist, konsequent Völkermord und ethnische Säuberung zu betreiben. Leider hat der Rationalansatz einige schwerwiegende Nachteile, etwa, daß man den Terroristen attraktive friedliche Möglichkeiten geben muß, ihre Ziele (Macht, Einfluß, Eigentum) zu erreichen, wozu in der Regel keine Bereitschaft besteht.

Was zur Zeit abläuft, ist ein sogenanntes Deterrence-Preemption Spiel. Deterrence (Abschreckung) ist das Erschweren von Anschlägen durch Sicherheitskontrollen. Preemption (Zuvorkommen) ist das Ausschalten von Terroristen durch Angriffe auf ihre Basen und das Aufspüren, bevor sie zuschlagen. Jeder Staat kann hier unterschiedliche Schwerpunkte setzen. Die Terroristen kön-

nen wiederum zwischen wenigen, spektakulären oder vielen "normalen" Anschlägen wählen. Die Terroristen in Israel setzen eher auf "normale" Anschläge, dies führt zu exzessiven Preemption-Maßnahmen. In Europa hingegen setzen die Terroristen auf wenige spektakuläre Anschläge, die Preemption-Maßnahmen ineffektiv und damit unzureichend machen und die Gefahr der Kompensation durch "Deterrence" nach sich ziehen, worunter alle Bürger zu leiden haben. Deterrence im Inland zieht als Externalisierungseffekt nach sich, daß Anschläge, auch auf Deutsche, ins Ausland verlagert werden. Deutschland verfügt von Hause aus offenbar aufgrund der ausgeprägten Bürokratie mit Personalausweisen, Meldepflicht und vielfältigem Genehmigungswesen bereits über ein hohes Deterrence-Potential im Vergleich zu vielen anderen Ländern.

Was also folgt daraus? Noch mehr Abschreckung durch Kontrollen verringert die Häufigkeit "normaler" Anschläge in Deutschland und erhöht die Wahrscheinlichkeit weniger, spektakulärer Anschläge in Deutschland und die Häufigkeit normaler Anschläge wie etwa Entführungen im Ausland.

Was tun?

Wenn man sich nun partout dem Rationalansatz verweigert, wäre mehr Preemption die vernünftigere Strategie. In der Praxis bedeutet das mehr Ressourcen für nachrichtendienstliche Aufklärung, Infiltration von Terrorgruppen und gute Beziehungen und Zusammenarbeit mit den Herkunftsländern. Die entscheidenden Hinweise auf die Identität der Bahnattentäter kamen wohl auch aus dem Libanon.

Auf keinen Fall können wir aber mehr ziellose Repression gegen Millionen harmloser Bürger in Form verschärfter Kontrollen an Flughäfen und Bahnhöfen gebrauchen.

Im Luftverkehr ist der Bogen bereits überspannt, und es wird höchste Zeit, daß hier zurückgerudert wird. Kafkaeske NoFly-Listen und das Erschießen geistig verwirrter Passagiere durch SkyMarshalls wie in den USA ist eine Form staatlichen Exzesses, der durch nichts zu rechtfertigen ist, auch nicht durch 9/11, und der am Ende den Nährboden für ganz neuen Terror aus der Mitte der Gesellschaft bilden wird, denn es ist nur eine Frage der Zeit, bis die Instrumente, die jetzt zur Abwehr einer äußeren Gefahr eingeführt werden, gegen etwaige aufbegehrende Unterprivilegierte im eigenen Land zum Einsatz kommen. Die deutschen Regierungsparteien haben durch ihre Bestrebungen, das Mautsystem zu Überwachungszwecken zu mißbrauchen, weitgehend an Glaubwürdigkeit verloren, und in den USA ist der Mißbrauch von Anti-Terrorgesetzen gegen Bürgerrechtler leider längst kein Einzelfall mehr.

Jeder einzelne – vor allem Politik und Presse – ist aufgerufen, auch dann einen kühlen Kopf zu bewahren, wenn der erste spektakuläre Anschlag in Deutschland Erfolg hat, was eigentlich nur noch eine Frage der Zeit ist, und sich anschließend klug zu verhalten und erst einmal abzuwarten, statt der Versuchung zu erliegen, mit großen Sprüchen große Dummheiten zu verbreiten und zu begehen.

Quelle: *http://acker3.ath.cx/wordpress/archives/26*

Information at your fingertips

Random Jane Hacker <jane@berlin.ccc.de>

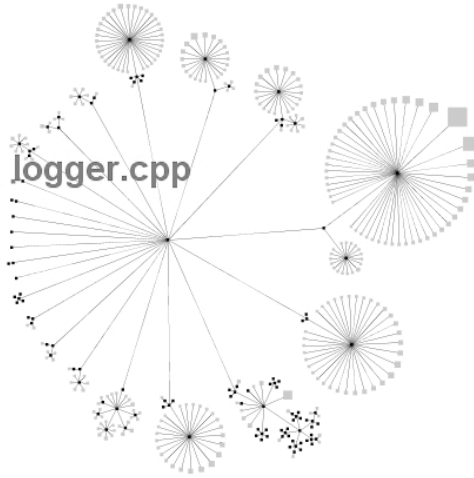
One problem that each of us encounters everyday is searching. In the context of computers, this usually means searching for information. But there's also a related problem, which I want to distinguish from searching here: information generation. This problem arises whenever you need some information that doesn't yet exist explicitly, even though it exists implicitly in some kind of data. The problem is thus one of derivation.

The reason why this information does not yet exist explicitly is most likely that it hasn't been useful enough yet for somebody to excavate it. As an example, imagine you are reading a paper. The paper is discussing some sociological question, such as the future of work. While reading, you ask yourself: Do the authors have some specific geographical background? Do they, for example, cite mainly or exclusively German sources? This would imply that their analysis might be biased towards a specifically German version of the problem they discuss. This is just a flash of a thought: that you might be able to judge the content of the paper better if you could just know the answer in an instant. This question would be easily answered if you had a world map of the locations the paper's citations were published in. The data itself is available, no problem, because the citation entries include the location of publication. However, such a map is unlikely to exist, because it is of such marginal use.

This example showcases what I think is the second grand challenge of information handling in the future: Making information transformation and information fusion ubiquitous, straightforward, effortless. Information at your fingertips. (The other grand challenge is of course searching the existing information.)

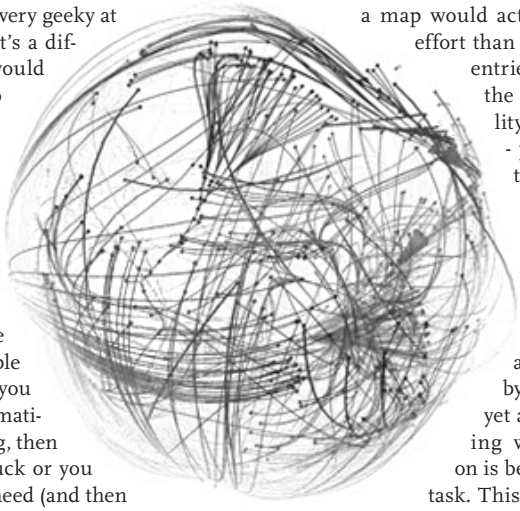
What we want to do in an effortless manner is this: On the one hand, we want to convert information between different representations, according to the task at hand. On the other hand, we want to fuse information from different sources and then jointly convert them between representations. "Representations" may be anything you can think of: A whole host of visualizations, or text, or sound and spoken word, or force-feedback, or direct brain stimulation. You name it. Whatever suits the task best.

There are several prerequisites for this to be effortless. One is digitalization of just about anything. I certainly won't type in all the publication locations from the above-mentioned paper to generate a map from it. It will be much quicker to just read the citation entries and extract my answer from that (subconsciously fusing this information with my knowledge of geography, by the way). So digitalization is the first requirement.



Graph representation of a directory structure, generated using yFiles library

The next prerequisite is in the implementation of transformations. Again, if I anticipate that it will take me an hour to put together a perl script that accomplishes the task of generating a map, I will just read the citation entries and be done (unless I feel very geeky at that moment, but that's a different story). So it would make most sense to have an array of pre-implemented transformations ready to use. Well engineered pieces of software, with well thought-out interfaces and functionality. Ideally, these will be easily shareable and augmentable. If you find that the transformation you need is missing, then you're either out of luck or you implement what you need (and then share it). I'm fairly sure there are already frameworks out there which are precisely that: an extensible collection of transformations for excavating information. I have seen at least one research project taking a stab at this [1].



Output generated by Vision Factory, a software for creating real-time interactive visual animations by the programming of reusable and sharable plugins, see http://www.v3ga.net/visionfactory/

Another requirement is already implicitly contained in the previous one: The need for a unified input and exchange mechanism for the individual transformations. As hesitant as I am to advocate XML, I'm afraid it would serve quite a good purpose there. A heap of information is just so much easier to fuse with another heap in some meaningful way if you can learn, in an automated fashion, about the semantics of that data heap. I'm using "semantics" in a very broad sense here, so don't get all wound up. What I'm referring to is this: Imagine you have a transformation that takes "Location"-data, the format of which is defined as an XML structure, and renders it into a map. Imagine then you encounter our example paper, in digitalized form, with citation entries given as XML data. In this setting, it is apparently a piece of cake to generate our map, and it will be done in no time at all and almost without effort. In this setting, generating a map would actually be less of an effort than reading the citation entries and determining the answer to our locality problem manually - provided, of course, that the data already exists in digitalized form.

If such information excavation tools would be available, this would, by the way, also call for yet another skill: Knowing which representation is best suited for a given task. This was obvious in our example. However, for more complex problems, this might be less trivial. Which representations are best suited to bring out certain features in certain data is in fact a subject of ongoing research, especially in the field of visualization. Intuitively knowing which representation to use and how to massage data most effectively will then gain you at least an advantage in speed.

Say what you will, but I find this terribly fascinating.

Oh, and yes. One name comes to mind here: Google. Quite a few of the services they are offering go into this direction. Very smart people over there.

[1] See the Visage system developed at Carnegie Mellon University (Visage at SIGCHI, paper introducing Visage)

Das anonyme Preisausschreiben "Keine Macht Für Niemand"

Constanze Kurz <46halbe@weltregierung.de>

Bei diesem anonymen Preisausschreiben gewinnt derjenige Leser oder diejenige Leserin, der/die alle Antworten korrekt beantwortet und als Mail an ds@ccc.de schickt. Bei mehreren richtigen Einsendungen gewinnt, wer die kürzeren, aber dennoch gut verständlichen Antworten gibt. Bei gleichen Einsendungen entscheidet nicht das Los, sondern irgendeiner hat dann plagiiert. Sowas kann nicht mit Preisen belohnt werden.

Die beste Zuschrift werden wir natürlich in der nächsten Ausgabe veröffentlichen!

1.) Du bist in einem offenen Funknetz. Nenne drei praktische Wege, wie jemand dich tracken könnte!

2.) Du verwendest deinen Browser für http mit TOR. Nenne drei praktische Wege, wie deine IP dennoch aufgedeckt werden kann! Dabei ist der Angreifer nicht die NSA.

3.) Manchmal nervt dich, daß TOR zu langsam ist. Daher schaltest du TOR zuweilen ab oder du benutzt einen Filter für bestimmte Webadressen, die du weiterhin nur mit TOR besuchst. Erkläre in maximal vier Sätzen, die jeweils nicht mehr als 18 Wörter haben, warum du damit deine Anonymität gefährdest!

4.) Du hast Javascript in deinem Browser aktiviert. Nenne zwei Möglichkeiten, wie dich jemand dadurch tracken kann!

5.) Erkläre in maximal zwei Sätzen, die jeweils nicht mehr als 18 Wörter haben, wie du auf einfachem Wege verhinderst, daß dich jemand mittels Javascript tracken kann, ohne daß du global die Benutzung von Javascript verbietest!

6.) Wie kannst du wirkungsvoll verhindern, daß bestimmte Applikationen deine Proxy-Einstellungen umgehen?

7.) Nenne vier lokale Services, die deine Identität im local network oder beim VPN-Endpunkt aufdecken können!

8.) Die Wikipedia blockiert das Editieren von Artikeln mit TOR. Nenne mindestens zwei Möglichkeiten, wie du diese Blockierung umgehen kannst, ohne deine Anonymität zu verlieren!

9.) Du möchtest einen anonymen Mailaccount haben. Welche Provider sollte man hier

nicht verwenden? Begründe in maximal vier Sätzen, die jeweils nicht mehr als 18 Wörter haben, worin jeweils die Gefahr besteht, wenn man diese Provider benutzt!

10.) Erkläre in maximal drei Sätzen, die jeweils nicht mehr als 18 Wörter haben, welche Gefahren generell bestehen, unabhängig vom Provider, wenn man einen anonymen Mailaccount klickt!

11.) Du möchtest einen öffentlichen IRC-Server anonym verwenden. Welche Einstellungen deines Client können hierbei deine Anonymität bedrohen?

12.) Nenne mindestens zwei anonyme Zahlungswege!

13.) Du möchtest Google Groups verwenden, um im Usenet zu posten. Erkläre in maximal drei Sätzen, die jeweils nicht mehr als 18 Wörter haben, was du vor und nach dem Zugriff beachten mußt?

14.) Du sitzt hinter einer Firmen-Firewall, die jeden Internet-Zugriff protokolliert und außerdem bestimmte Ports sowie Programme blockiert. Erkläre in maximal zwei Sätzen, die jeweils nicht mehr als 18 Wörter haben, wie du trotzdem machen kannst, was du eben so machen willst!

15.) Du sitzt immernoch hinter besagter Firewall. Nenne drei Dinge, die dein Antrieb sind, die Firewall zu umgehen!

16.) Du bist weiterhin in der Firma mit der vordergründig restriktiven Firewall. Du weißt, daß der dortige Netzwerk- und Firewallverantwortliche mittels eines Keyloggers auch deine Tastatureingaben überwacht. Du mußt aber ganz dringend deiner Mama tausend Euro überweisen. Erkläre in maximal drei Sätzen, die jeweils nicht mehr als 18 Wörter haben, wie du das Paßwort für dein Online-Banking so eingibst, daß dieser Schnüffler leer ausgeht!

17.) Erkläre in maximal sieben Sätzen, die jeweils nicht mehr als neun Wörter haben, wie man auf einfachem generischen Weg ein verstecktes Rootkit auf einem Rechner aufdeckt!

18.) Gib zwei Beispiele, warum der Online-Kauf von Büchern gefährlich sein kann!

19.) Du möchtest trotzdem ein Buch online kaufen. Erkläre in maximal sieben Sätzen, die jeweils nicht mehr als neun Wörter haben, wie du vorgehst, wenn du die eben genannten Gefahren ausschließen willst!

20.) Wie kannst du – solltest du das Quiz gewinnen – sicherstellen, daß du den Preis von uns bekommst, ohne deine Anonymität aufzugeben? Zeige mindestens drei Wege auf, unter der Bedingung, daß in keinem der Szenarien die physische Integrität eines Mensches gefährdet wird!

Anonymisierungsdienst TOR: Wenn die Polizei 2x klingelt

"Herr Weber" <ds@ccc.de>

"The Onion Router", kurz TOR, ist eine mittlerweile sehr populäre Methode, um anonym im Internet zu surfen. Mit Hilfe eines so genannten Exit-Nodes lassen sich TCP-Verbindungen aufbauen und so Daten im Web abrufen und bereitstellen, ohne daß der Nutzer mittels IP-Adresse auffindbar wäre.

Das TOR-Overlay-Netzwerk wurde an einer US-Uni entwickelt und zeitweise sogar von der dortigen Regierung mitfinanziert. Inzwischen wird es vor allem von der Netzbürgerrechtsorganisation Electronic Frontier Foundation (EFF) vorangetrieben; mit „JAP" gibt es ein technisch etwas anders aufgebautes deutsches Pendant von der TU Dresden.

Was an Universitäten als Forschungsprojekt gefördert und von Datenschützern und Bürgerrechtlern weltweit begrüßt wird, kann einen Otto-Normal-Nutzer jedoch in eine äußerst prekäre Situation bringen, wie Benedikt Weber (Name geändert) schmerzlich erfahren mußte. Der Grund: Die deutschen Polizeibehörden haben ein Auge auf die Technik geworfen.

Weber, ein 24-jähriger Jura-Student und freiberuflicher Web-Entwickler, ist in der Open-Source- und Free-Speech-Szene engagiert – kein Wunder, daß ihn daher technische Entwicklungen zum Schutz persönlicher Daten sehr interessieren. Aus diesem Grund entschloß er sich auch, mit TOR zu experimentieren – und zwar ursprünglich nur zu Testzwecken. Und so brummte ein Exit-Node – also der Rechner, der am Schluß der anonymen TOR-Kette steht und die Daten letztlich anfordert – über mehrere Monate friedlich nebenher auf einem von Webers Servern in einem Rechenzentrum.

Im August 2006 dann der Schock: Plötzlich standen zwei Ermittler der Kriminalpolizei vor Webers Tür und wedelten mit einem Durchsuchungsbefehl. Grund der Aktion: Ein „Ermittlungsverfahren wegen Verbreitung kinderpornographischer Schriften" – so ziemlich die unangenehmste Straftat, derer man sich neben Mord und Totschlag bezichtigt fühlen kann.

Weber wußte zunächst nicht, ob er lachen oder weinen sollte. Ihm und auch seiner Mutter, die während der Durchsuchung anwesend war, war klar, daß er eine solche Straftat niemals begangen hat. Doch der 24-jährige ist sich der Schwere des Vorwurfs bewußt und möchte den gegen ihn erhobenen Verdacht verständlicherweise schnell aus der Welt räumen.

Im Gespräch mit den beiden Beamten kam schnell heraus, daß die zum Ermittlungsverfahren gehörende Straftat über das Internet verübt wurde. Benedikt Weber bat die beiden Beamten daher um Nennung der betroffenen IP-Adresse. Sofort wurde ihm klar, daß diese zu dem von ihm angemieteten Server gehört.

Dann dämmerte es ihm: Die Sache hängt wohl mit seinem TOR-Exit-Node zusammen. Das heißt: Die Beamten suchten also gar nicht Weber als Betreiber, sondern einen TOR-Nutzer, der über seinen Exit-Node anonym ins Netz ging.

Das erklärte Weber auch den Polizisten, die offensichtlich keine IT-Experten waren. Viel half das allerdings nicht: Benedikt Weber wird darüber informiert, daß die Beamten bereits vom Betrieb des TOR-Servers wussten. Selbst

wenn ihm nicht nachzuweisen wäre, daß er selbst auf die Daten zugegriffen habe, könne ihm immer noch vorgeworfen werden, eine Straftat unterstützt zu haben, lassen die Herren verlauten.

Eine Beihilfe im Sinne des deutschen Strafrechts laut § 27 Abs. 1 StGB liegt allerdings nur dann vor, wenn jemand (der so genannte Gehilfe) vorsätzlich einen Täter bei der Begehung einer Straftat (erfolgreich) unterstützt. Weber müßte also vorsätzlich jemanden bei der Verbreitung kinderpornographischer Schriften unterstützt haben. Der Vorsatz ist nach dem Umkehrschluß aus § 16 Abs. 1 StGB das „Wissen und Wollen sämtlicher Tatbestandsmerkmale". Der Vorsatz muß dabei die „wesentlichen Elemente des eingetretenen Kausalverlaufs" umfassen, zumindest „in bedingter Form". Da Weber die Nutzer seines TOR Exit-Nodes aber allein schon aus technischen Gründen gar nicht kennen kann (TOR anonymisiert sie ja eben), kann ihm eigentlich auch kein Vorsatz nachgewiesen werden, diese (bei was auch immer) zu unterstützen.

Wie funktioniert TOR?

Jedem Rechner im Internet ist eine IP-Adresse zugeordnet, mit deren Hilfe der Nutzer identifiziert werden kann. Besucht beispielsweise ein Kunde des Providers XYZ die Seite www.nsa.gov, wird die ihm vom Provider zugeteilte IP-Adresse in den Log-Dateien des Servers www.nsa.gov gespeichert.

Benutzt der Kunde hingegen das TOR-System, um die Seite zu besuchen, wird keine direkte Verbindung zwischen seinem Computer und dem Server hergestellt. Statt dessen wählt die TOR-Software auf seinem Rechner zufällig drei TOR-Server aus, einen Einstiegspunkt (den sogenannten „Entry-Node"), eine Zwischenstation (den sogenannten „Middle-Man") und einen Ausstiegspunkt (den sogenannten „Exit-Node").

Die Anfrage mit dem Ziel www.nsa.gov wird dann in mehreren Schichten (daher The Onion Router, „Onion" ist das englische Wort für Zwiebel) verschlüsselt. Die Verschlüsselung funktioniert so, daß jeder TOR-Server jeweils nur die für ihn bestimmte „Schale" entschlüsseln kann. Mit dem Entry-Node wird noch eine direkte Verbindung aufgebaut, dieser kann aber nur die erste Schale entschlüsseln, unter der er lediglich die Anweisung findet, das Paket an den Middle-Man weiterzuleiten.

Der Middle-Man baut lediglich eine Verbindung mit dem Entry-Node auf, bereits ihm ist also die IP-Adresse des Nutzers unbekannt. Auch der Middle-Man entschlüsselt wieder seine Schale und findet darin die Anweisung, die Anfrage an den Exit-Node weiterzuleiten, der seinerseits die letzte Schale entschlüsselt und die Verbindung mit dem Zielserver herstellt.

Der Exit-Node kennt weder den Entry-Node, geschweige denn den Nutzer. Da einzig der Exit-Node eine direkte Verbindung mit dem Zielserver aufbaut, erscheint auch lediglich seine IP-Adresse in dessen Log-Dateien. Der Nutzer ist also anonym – auch eine spätere Aufdeckung seiner Identität ist ausgeschlossen, da TOR-Server keine Log-Dateien anlegen, wenn man sie korrekt konfiguriert.

Immerhin dient ein Anonymisierungsnetzwerk wie TOR vor allem auch einer ganzen Reihe ehrbarer Ziele – von Free Speech-Aspekten bis hin zur Möglichkeit, sich werbefrei im Netz zu bewegen.

Ähnliche Motivationen treiben auch Weber um: Er hat seinen TOR Exit-Node installiert, weil er damit seinen Teil zur Sicherstellung der ungehinderten Kommunikation der Bürger untereinander und damit letztlich auch das (in Deutschland sogar durch das Teledienstedatenschutzgesetz (TDDG) garantierte) Recht auf Anonymität unterstützt. „Gerade für freiheitliche Gesellschaften ist es doch kennzeichnend, daß die Bürger untereinander frei und ohne staatliche Gängelung kommunizieren können", sagte er auch den Beamten.

Doch auch diese Argumente konnten die Vertreter des Gesetzes nur wenig nachvollziehen und vertraten offensiv die Meinung, daß derartige rechtsfreie Räume geschlossen werden müßten. Bei der anschließend durchgeführten Durchsuchung fanden die Polizisten selbstverständlich keinerlei belastendes Material, legten Benedikt Weber aber dringend nahe, sich bei ihnen mit einer Aussage zu melden.

Nachdem sich Weber etwas beruhigt hatte, ging er an die Recherche. Vor allem die Erkenntnis, daß ein Strafverfahren absolut berufsfeindlich für einen Jurastudenten sein kann, verdeutlichte ihm schnell, daß dringend ein adäquater Rechtsanwalt von Nöten war. Ähnliche Fälle waren hingegen leider nur sehr wenige aufzutreiben, einzig das Vorgehen des Bundeskriminalamtes gegen das deutsche Anonymisierungs-

projekt JAP der TU Dresden im Jahre 2003 fand sich beim Fallstudium. Immerhin: Dort konnte sich das JAP-Projekt erfolgreich gegen die Maßnahmen wehren – allerdings betrieb das JAP-Projekt alle Knotenpunkte selbst, somit war die Situation dort eine andere.

Ein Anonymisierungsserver könnte rechtlich auch als Mediendienst im Sinne des § 3 Abs. 2 des Mediendienstestaatsvertrages (MDStV) zu sehen sein. Damit könnte sich der Betreiber eines Anonymisierungsservers auch auf § 7 des MDStV berufen, wonach ein Diensteanbieter nicht für fremde Informationen verantwortlich ist, soweit er die Übermittlung nicht selbst veranlaßt hat. Der Betreiber eines TOR Servers kann auch unmöglich wissen, wer Informationen durch seinen Server leitet, auch veranlaßt er die Durchleitung nicht selbst.

Glücklicherweise fand Weber mit dem Anwalt Wilhelm Achelpöhler einen Rechtsbeistand, der seine Interessen vertreten wollte. Zwar ist RA Achelpöhler kein spezieller IT-Anwalt, überzeugte aber dadurch, daß er die Funktionsweise eines Overlay-Netzwerkes sehr schnell verstand, generell den Eindruck machte, sich für das Recht der informationellen Selbstbestimmung einzusetzen und nach Erklärung der Funktionsweise von TOR direkt erkannte, daß Dinge wie die umstrittene Vorratsdatenspeicherung damit für den „emanzipierten IT-Nutzer“ zu umgehen sind.

Noch bevor Achelpöhler einen Blick in die von ihm angeforderten Ermittlungsakten werfen konnte, überschlugen sich die Ereignisse in den Medien. Wie sich herausstellte, war Weber offenbar nicht der einzige Betroffene. Auf Grund eines Ermittlungsverfahrens der Staatsanwaltschaft Konstanz [1] wurden mehr als 10 TOR-Server von anderen Betreibern beschlagnahmt, mit dabei war auch ein JAP-Server des Unabhängigen Landeszentrums für Datenschutz Schleswig-Holsteins, das sich in einer Pressemittelung massiv über diese Maßnahme beschwerte [2]. Später äußerte sich auch *netzpolitik.org* zu diesem Fall mit einem kritischen Artikel zur langsamen staatlichen Vernichtung des Rechts auf Anonymität [3].

Warum TOR?

„Die Nutzung von Tor schützt gegen eine übliche Form der Internetüberwachung, die als Analyse des Netzverkehrs bekannt ist. Die Analyse kann dazu verwendet werden, Informationen abzuleiten, wer mit wem über ein öffentliches Netzwerk kommuniziert. Wenn jemand Quelle und Ziel deines Internetverkehrs kennt, kann er dein Verhalten und deine Vorlieben nachvollziehen.

Das kann sich auf deinen Geldbeutel auswirken, indem z.B. eine E-Commerce-Seite ihre Preise vom Herkunftsland und deiner Firma abhängig macht. Es kann sogar deinen Arbeitsplatz und körperliche Unversehrtheit bedrohen, wenn öffentlich wird, wer du bist und wo du wohnst.

Wenn du dich beispielsweise im Ausland auf Dienstreise befindest und dich mit dem Computer deines Arbeitgebers verbindest, kannst du ungewollt deine Nationalität und Ihren Arbeitgeber jedem gegenüber offenbaren, der das Netzwerk beobachtet, auch wenn die Verbindung verschlüsselt ist."

TOR-Übersicht, http://tor.eff.org/overview.html.de

Die schließlich eingetroffene Ermittlungsakte lieferte ein trauriges Bild ab. Schnell wurde klar, daß die durch die Medien geisternden Fälle mit dem von Benedikt Weber zusammenhingen. Der Ermittlungsbericht einer speziell eingerichteten „IT-SOKO" brachte Licht in den ganzen Fall. Demnach hatte der Betreiber eines Gratis-Hosting-Projektes auf seinem Webserver kinderpornographisches Material festgestellt und daraufhin die Polizei verständigt. Dort wurden dann schnell Beweismittel in Form von Log-Dateien und Bildmaterial gesichert. Schnell konnte durch die IP-Adresse aus den Log-Dateien festgestellt werden, wer die Daten dort bereitgestellt hatte. Bei einer Durchsuchung wurde bei dieser Person umfangreiches Beweismaterial sichergestellt. In einer anschließenden Vernehmung war diese Person laut Ermittlungsbericht auch geständig.

Eher beiläufig ist gegen Ende des Ermittlungsberichts folgendes zu lesen: „Im Zuge der Ermittlungen wurde festgestellt, daß auch von der IP-Adresse XYZ.XYZ.XYZ.XYZ auf die inkriminierten Daten zugegriffen wurde. [...] Der Tatverdacht gründet sich dabei zunächst jedoch nur darauf, daß es sich bei der ermittelten Person um den beim jeweiligen Provider registrierten Kunden handelt. Es ist daher nicht ausgeschlossen, daß es sich bei dem tatsächlichen Täter um ein Familienmitglied, bzw. eine sonst im Haushalt lebende Person handelt, die Zugriff auf den jeweiligen Internetzugang hatte. [...] Im vorliegenden Fall konnte der Tatverdacht nicht weiter erhärtet werden. Wie weitergehende Ermittlungen ergaben, gehört die IP-Adresse zu einem vom Beschuldigten betrieben Server. Auf diesem Server wird offensichtlich auch ein sogenannter Onion-Router des Anonymisierungsnetzwerkes TOR betrieben. Bei dem festgestellten Zugriff könnte es sich auch um einen über dieses Netzwerk geführten Zugriff gehandelt haben. In diesem Fall wäre nicht zu ermitteln, welche Person tatsächlich auf die fraglichen Daten zugegriffen hat."

Ergo: Die Staatsanwaltschaft wußte also von Anfang an, daß sie mit großer Wahrscheinlichkeit eine Hausdurchsuchung – und damit einen sehr schweren Eingriff in die Privatsphäre – bei einem Unschuldigen anordnet.

Interessanterweise scheint der für den Wohnort von Benedikt Weber zuständige Staatsanwalt die vorgenannten Ermittlungsergebnisse aber völlig anders interpretiert zu haben. In seiner Akte ist zu lesen: „Nach den bislang vorliegenden Erkenntnissen sind ausreichende Gründe dafür vorhanden, daß sich der Beschuldigte [...] Bilddateien mit kinderpornographischen Abbildungen verschafft hat und weiterhin besitzt. [...] Die Durchsuchung der Wohnung [ist] anzuordnen. Es ist zu vermuten, daß die Durchsuchung zur Auffindung von Beweismitteln führen wird." Auch der für den Vorgang zuständige Richter schien keinerlei Bedenken gehabt zu haben und unterschrieb den Durchsuchungsbeschluß ohne Einwände.

Benedikt Weber steht inzwischen mit vielen anderen betroffenen Betreibern von TOR-Servern in Kontakt. Erstaunlicherweise wurde in keinem anderen Fall eine Hausdurchsuchung durchgeführt. Lediglich die Server wurden beschlagnahmt – und selbst diese Beschlagnahmungen wurden in den Experten-Medien heftig kritisiert.

Auch in dem Ermittlungsbericht zur Akte Webers schrieben die Ermittler bereits, daß im

Falle eines Zugriffs über das TOR-Netzwerk keine Daten sicherstellbar sind. Eine Beschlagnahmung soll aber regelmäßig dem Auffinden von Beweismitteln dienen. Wenn von vorne herein klar ist, daß dieser Zweck gar nicht erreicht werden kann, ist die Rechtmäßigkeit mindestens fraglich.

Bei Weber wurde darüber hinaus eine Hausdurchsuchung angeordnet, obwohl bereits aus der Ermittlungsakte hervor ging, daß die Wahrscheinlichkeit einer Täterschaft äußerst gering ist. Man darf sich daher zumindest die Frage stellen, welcher Zweck hier tatsächlich verfolgt wurde.

Weber traf sich mit seinem Anwalt, um eine Stellungnahme/Gegenvorstellung an die zuständige Staatsanwaltschaft aufzusetzen, die im Wesentlichen auf eine Einstellung des Strafverfahrens gerichtet war.

Unter Berufung auf den Ermittlungsbericht heißt es darin, daß der Betrieb eines TOR-Servers eine durch den MDStV und das TDDG rechtlich völlig gedeckte Sache ist. Vor allem aber regte Anwalt Achelpöhler an, daß in der Einstellung des Strafverfahrens festgestellt wird, daß Weber unschuldig ist und kein Tatverdacht mehr besteht.

Zwar erfolgte nach einigen Monaten tatsächlich die Einstellung des Strafverfahrens, allerdings wurde nicht spezifisch festgestellt, daß Weber unschuldig ist und kein Tatverdacht mehr besteht. Falls nun eine Eintragung „zur Gefahrenabwehr“ z.B. im Verfahrenszentralregister erfolgen würde, hätte Weber also keinen Anspruch auf Löschung und damit möglicherweise erhebliche Nachteile im Beruf.

Die Staatsanwaltschaft beruft sich hingegen darauf, daß eine Feststellung der Unschuldigkeit Webers nur dann möglich sei, wenn die Person ermittelt worden wäre, die tatsächlich auf die Daten zugegriffen habe, und daß nur zu Gunsten Webers davon ausgegangen worden sei, daß tatsächlich jemand über das TOR-Netzwerk auf die inkriminierten Daten zugegriffen habe.

Fazit: Mit dieser Argumentation dürften zahlreiche Geschäftsführer von Internet- und Telekommunikationsunternehmen eine sehr lange Liste an eingestellten Verfahren erwarten – und zwar ohne daß die explizite Unschuldigkeit festgestellt wurde.

Weber fragt sich außerdem, warum es für ihn nun zum Nachteil werden soll, daß er eben keine Daten zur Auslieferung eines vermeintlichen Täters weitergeben kann, wo in Deutschland doch der Datenvermeidungsgrundsatz gilt, sprich: nur die Daten aufgezeichnet werden dürfen, die für den Betrieb eines Mediendienstes auch wirklich erforderlich sind. Bei einem kostenlosen Anonymisierungsdienst sind das konsequenterweise: Gar keine.

Zwar ist die Sache mittlerweile weitestgehend abgeschlossen, trotzdem macht sich Jurastudent Weber große Sorgen um den Zustand des Rechtsschutzes und damit der Liberalität in Deutschland.

Mitte 2007 macht Weber einen Einstellungstest bei der Lufthansa mit – sein eigentlicher Traumberuf ist die Fliegerei. „Die Chancen, den Einstellungstest als Pilot mit Erfolg zu durchlaufen, sind sehr gering, aber sollte es dennoch klappen, wäre es umso ärgerlicher, in der Sicherheitsüberprüfung durchzufallen, weil doch noch in irgendeiner Datenbank etwas über dieses Verfahren gespeichert ist“, sagt er.

Noch in diesem Frühjahr möchte Weber deshalb Anfragen an LKA, BKA, Verfassungsschutz und die Generalbundesanwaltschaft (die das Verfahrenszentralregister führt), schicken, um sicherzustellen, daß keine ihm nachteiligen Daten gespeichert wurden. Zur Generierung passender Formulare will er das Internet verwenden [4].

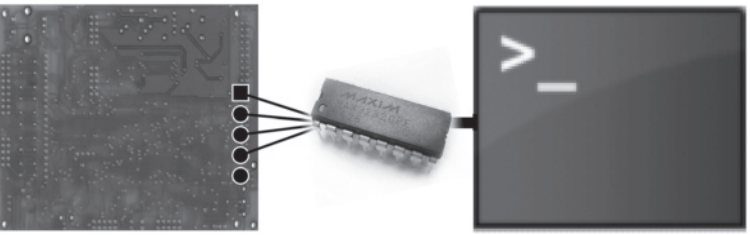
[1] *http://www.theregister.co.uk/2006/09/11/anon_servers_seized/*
[2] *http://www.datenschutz.de/news/detail/?nid=1933*
[3] *http://netzpolitik.org/2006/anonymer-internet-zugang-wird-kriminalisiert/*
[4] *http://www.argh-it.de/cgi-bin/auskunft*

Serielle Schnittstellen bei Embedded Devices

Steph <sla at stylepolice punkt de>

„Can we get instructions to access the serial port posted in here?

Er. The first post contains the instructions. or did you mean a quick course in soldering, basic electronic theory, TTL logic, and how to rx/tx via your serial ports?“



Viele Embedded Devices haben eine serielle Schnittstelle, mit der Bootloader, die Console oder Shell einfach anzusprechen sind. Meist ist ein embedded Linux das Betriebssystem und mit dem Zugang zu diesem eröffnen sich neue Möglichkeiten, das Gerät zu nutzen oder näher zu inspizieren.

Hat das Device eine zweite Schnittstelle, können sogar Geräte – wie z.B. ein serielles LCD – angeschlossen werden. Gewöhnlich sind nur TX und RX (Senden und Emfpangen) in Form von Pins vom UART (Universal Asynchronous Receiver/Transmitter), der den seriellen Datenstrom handhabt, auf das PCB nach außen geführt, was uns aber reichen soll.

Hier will ich nun kurz erklären, wie wir die relevanten Pins TX, RX, GND bzw. VCC auf dem PCB mit Hausmitteln finden können, eine Stiftleiste auflöten und mittels eines Levelshifters oder Mobilfunktelefondatenkabels die serielle Schnittstelle an unseren Computer anschließen und als Console nutzen. Die benötigten Elektronikbauteile gibt es bei z.B. Reichelt oder Segor.

Die Pinbelegung finden

Die Pins der seriellen Schnittstelle(n) liegen oft zusammen mit GND und Versorgungspannung VCC, manchmal mit weiteren Pins in Form einer oder zweier Lötaugen- oder Padreihe(n) auf dem PCB vor. Manchmal ist sogar schon eine Stiftleiste eingelötet, so daß wir uns diese Arbeit sparen können. Dies ist nicht weiter verwunderlich da die Entwickler der Geräte ebendiese seriellen Schnittstellen selbst für ihre Entwicklungsarbeit nutzen.

Von Anderer Arbeit Früchten profitieren

Oft haben schon andere die gesuchte Pinbelegung veröffentlicht, man suche z.B. nach: <Device name> serial pinout.

Das spart uns das Ermitteln der Belegung. Wir können aber auch ein wenig an einem bekannten Gerät üben, bevor wir es mit einer unbekannten Belegung probieren. An den Fundstellen im Netz ist oft auch mehr oder weniger ausführlich – zum Teil sogar bebildert – beschrieben, wie ein Levelshifter an welche Pins von $Gerät anzuschließen ist. Der berühmte Pin 1 ist übrigens markiert – mit einer 1, durch ein quadratisches Lötauge oder einer Markierung auf dem Bestückungsdruck.

Geeignete Pins selbst finden

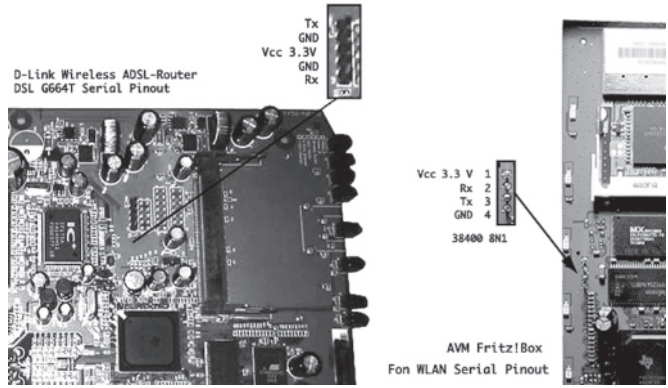
Guckt euch einfach das entspechende Board genauer an. Sobald wir eine geeignet erscheinende Pinreihe gefunden haben testen wir diese auf die gesuchten Pinouts durch.

Groundpin GND finden

Meist sind Groundpins mit der Groundplane verbunden, d.h. es führt keine Leiterbahn zum Lötauge sondern das Auge ist direkt mit der sie umgebenden Kupferfläche verbunden. Ein Groundpin läßt sich gut am ausgeschalteten Gerät durchpiepsen. Dazu ein Multimeter auf Durchgangsprüfung ⊣⊢ stellen und den schwarzen Meßfühler „COM“ an ein eindeutig mit GND verbundenes Bauteil z.B. den GND der Stromversorgungsbuchse am Gerät – **nicht** an der 230V Seite des Netzteils! – halten. Mit dem anderen Meßfühler die Pins nacheinander durchprobieren. Steht 0 auf dem Display, bzw. ist ein Piepston zu hören, ist ein GND Pin gefunden. Es kommt vor, daß mehrere Pins mit GND verbunden sind. Wir notieren uns alle Pins, die mit GND verbunden sind.

3,3V Pins finden

Wir suchen nun am eingeschalteten Gerät nach den Pins, die eine Spannung von ca. 3,3V (bzw. 5V oder 1,8V) aufweisen. Im Folgenden, da bei den meisten Geräten zur Zeit so vorhanden, beschreibe ich das Vorgehen stellvertretend mit 3,3V. Hat ein Pin 3,3V, könnte es ein Signalpin des UARTS sein, der TTL Spannung hat und auf HIGH gesetzt ist – also einer unserer gesuchten TX/RX Pins oder aber die Versorgungsspannung. Zum Finden der 3,3V Pins nehmen wir nun ein Oszilloskop, oder ein auf 20V Gleichstrom gestelltes Multimeter. Wir halten das Multimeter mit dem schwarzen Meßfühler an GND und probieren die Pins mit dem roten Meßfühler nacheinander durch. Wir notieren alle Pins, an denen wir ca. 3,3V vorfinden.



2 Beispiele für serielle Pinouts an Wireless Routern.

Versorgungsspannung VCC finden

Der Pin der Versorgungsspannung VCC kann ebenfalls durchgepiepst werden. Das Multimeter stellen wir dazu wieder auf Durchgangsprüfung und halten am ausgeschalteten Gerät den schwarzen Meßfühler des Multimeters an ein Bauteil mit 3,3V Versorgungsspannung (z.B. einen der Chips oder direkt hinter dem Spannungswandler). Mit dem roten Meßfühler probieren wir die Pins durch. Manchmal existieren mehrere Versorgungsspannungsleitungen auf einem PCB. Wenn es nicht piepst kann der Pin also trotzdem ein Versorgungsspannungspin sein. Es können auch mehrere Pins an die Versorgungsspannung angeschlossen sein. Wenn ihr keinen Pin findet, ist das nicht so schlimm, denn für die Mobilfunktelefon-Datenkabel-Lösung reichen TX/RX/GND. Die Versorgungsspannung des Levelshifters kann dabei woanders abgegriffen werden. Die gefundenen VCC Pins notieren wir.

Sendeleitung TX finden

Um die Sendeleitung (Transmit, TX) zu finden, machen wir uns den Umstand zunutze, daß die meisten Embedded Devices die Ausgaben ihrer Console auf die erste serielle Schnittstelle umleiten und daher beim Booten eine zeitlang Zeichen senden. Im Folgenden probieren wir nun alle Pins, die eindeutig nicht GND oder VCC sind und die 3,3V Spannung aufweisen, mit einer der beschriebenen Methoden durch.

• Mit dem angeschlossenen Oszilloskop können wir das Device mittels seiner Signalkurve nach dem Einschalten beim Booten beobachten, wenn wir den TX Pin gefunden haben – und auch gleich die Baudrate feststellen.

• Mit einer blauen LED (Durchlaßspannung ca. 3,5V) können wir ebenfalls beim Booten zusehen. Dazu halten wir das kürzere Beinchen der LED, die sogenannte Anode (–), an GND und das längere Beinchen, die Kathode (+), mit dem zu prüfenden Pin. Wenn die LED beim Einschalten des Geräts flackert, stehen unsere Chancen gut.

• Mit einem Piezopieper können wir das Gerät booten hören. Dazu sollten wir nur Pieper ohne Spule benutzen. Das schwarze Kabel verbinden wir mit GND, das Rote mit dem zu testenden Pin. Haben wir den TX Pin getroffen, „erklingt" unsere Serielle. Genau! Manche kennen diese Art Geräusch noch aus Modemzeiten.

• Ebenso können wir mit einem Levelshifter oder einem passenden Mobilfunktelefon-Datenkabel nach dem TX-Pin suchen. Die Überprüfung eines durch eine vorher beschriebene Methode gefundenen TX-Pins durch diese Methode ist ebenfalls sinnvoll. Dazu verbinden wir die GND- (und nur beim Levelshifter auch die VCC-) Leitung mit den bereits gefundenen korresspondierenden Pins auf dem PCB des Gerätesn. Die Leitungen können z.B. provisorisch angelötet werden.

Wir verbinden die TX-Leitung des Levelshifters/Mobilfunktelefon-Datenkabels mit dem zu testenden Pin und achten beim Levelshifter darauf, daß entweder ein Nullmodemkabel verwendet oder RX/TX hinter dem Levelshifter anderweitig auf dem Weg zum Computer gekreuzt werden. Andernfalls würde der Computer die Signale auf seiner Sendeleitung empfangen. Wir verbinden nun Levelshifter/Datenkabel mit dem Computer, starten und konfigurieren das Terminalprogramm (weitere Details dazu sind weiter hinten im Artikel beschrieben) und schalten das Gerät ein. Begrüßt uns Buchstabensalat, müssen wir die Baudrate ändern. Kommt hingegen nichts über die Leitung, kann es durchaus nicht schaden, mal die vermeintliche RX Leitung auszuprobieren – nur für den Fall, daß man sich da vertan hat.

Empfangsleitung RX finden

Um die Empfangsleitung (Recieve, RX) zu finden – so nicht sowieso nur noch ein Pin übriggeblieben ist – verbinden wir alle bisher ermittelten Pins wie beschrieben. Die RX-Leitung halten wir an den zu testenden Pin, verbinden Levelshifter/Datenkabel mit dem Computer, starten das Terminalprogramm mit der richtigen Baudrate und senden Zeichen mittels Tastatureingaben. Werden die Zeichen vom Embedded Device erkannt, haben wir die RX-Leitung des Gerätes gefunden.

Auf keinen Fall solltet ihr GND mit VCC kurzschließen oder die VCC des Gerätes mit dem GND eures Computers verbinden.

Einbau der Stiftleiste

Nachdem wir die Pins ermittelt haben, sollten wir – wenn nicht bereits schon eine vorhanden ist – eine Stiftleiste einlöten, damit wir den Levelshifter/das Datenkabel aufstecken können. Wer keine Lust dazu hat kann, die Leitungen auch direkt anlöten, ich bevorzuge allerdings die Steckvariante.

Entfernen des Lötzinns aus den Lötaugen

Sind die Lötaugen auf dem PCB mit Lötzinn verschlossen, entfernen wir dieses vorher, da die Stiftleiste sonst nicht eingelötet werden kann. Dazu kann wahlweise Entlötlitze, eine Absaugpumpe oder die Kanüle einer Spritze verwendet werden.

• Mit Entlötlitze: Wir drücken die Enlötlitze leicht(!) mit dem heißen Lötkolben auf das von Lötzinn zu befreiende Auge. Sobald das Lötzinn sich verflüssigt, saugt die Litze das Zinn auf. Eventuell kann hier durch vorheriges nochmaliges Verzinnen des Lötauges bzw. vorheriges Auftragen von etwas Flußmittel (Löthonig) ein besseres Ergebnis erzielt werden.

• Mit der Absaugpumpe: Wir spannen das PCB senkrecht ein und drücken den lotrecht dazu gehaltenen Lötkolben mit der Spitze leicht (!) auf das Lötauge. Von der anderen Seite halten wir die gespannte – ebenfalls lotrecht gehaltene – Absaugpumpe auf das gleiche Lötauge, nur eben von hinten. Wenn das Zinn flüssig ist, ziehen wir den Lötkolben weg und lösen im selben Moment die Absaugpumpe aus. Auch hier kann Flußmittel oder erneutes Verzinnen des Auges helfen.

• Mit der Kanüle: Wir placieren die Kanüle lotrecht mit der Spitze auf dem Lötauge. Mit dem Lötkolben erhitzen wir die Kanüle und das Auge seitlich im Winkel von ca. 45 Grad, etwa so, als wolle man ein Bauteil anlöten. Sobald das Lötzinn flüssig wird, schieben wir die Kanüle vorsichtig nach unten. Wir nehmen den Lötkolben weg und drehen beim Erkalten der Lötstelle die Kanüle eventuell vorsichtig, damit sie sich leichter aus dem Lötauge nehmen läßt. Bei dieser Methode wird das Lötzinn nicht entfernt, sondern nur aus dem Auge geschoben. Dies macht aber nichts, die Stiftleiste kann trotzdem eingelötet werden.

Besonders die Lötaugen, die mit der Groundplane verbunden sind, sind häufig etwas hartnäckiger, da die Wärme des Lötkolbens in die Groundplane abfließt. Hier hilft nur Geduld beim Erhitzen – bitte keine Gewalt oder zu hohe Hitze – denn barbarisch veranlagte Menschen bohren das Zinn lieber mit dem Minidrill aus dem Auge. Wer zuviel Druck oder zu hohe Hitze auf das PCB einwirken läßt, läuft Gefahr, das Lötauge zu zerstören und muß den Lötlack dann vorsichtig von der Leiterbahn entfernen und sich aus sehr dünnem Draht eine Brücke bauen – oder das Signal woanders vom Board abgreifen. Nach einer solchen Aktion empfiehlt sich übrigens das Eingießen des Desasters mit 5min Epoxidharz, um weitere Nervereien zu vermeiden. Dies sollte natürlich erst bei fertig eingelötete Stiftleiste nach erfolgreichem Test der Schnittstelle geschehen.

Laßt euch lieber von jemandem mit Löterfahrung und Lötstation helfen, wenn ihr Probleme haben solltet.

Das Einlöten der Stiftleiste

Wir kürzen zuerst die Stiftleiste auf die Anzahl der Pins. Auch hier lassen wir beim Löten höchste Vorsichtig walten, da das Plastik der Leiste durch die Hitze schnell weich und die Pins schräg eingelötet werden. Wir stecken eventuell eine Buchsenleiste oder ein paar Jumper als Fixierung auf die langen Enden auf – löten diese aber nicht mit fest ;)

Die Stiftleiste setzen wir nun von oben so ein, daß die kurzen Enden der Pins durch das PCB gesteckt sind. Wir heften zuerst 2 gegenüberliegende Pins von unten mit Lot fest und löten dann alle Kontakte, indem wir die Lötspitze im Winkel on ca. 45 Grad Winkel zwischen Lötauge und Pin legen, kurz warten und das Zinn zwischen Lötspitze und Pin zum Lötauge fließen lassen. Wenn wir Lötkolben und Zinn vorsichtig wegziehen, sollte ein schöner Kegel entsteht. Das Lötzinn sollte man immer erst richtig abkühlen, d.h. ohne Bewegung erstarren lassen, bevor wir uns der nächsten Lötstelle widmen. Sonst kann es zu einer sogenannten „kalte Lötstelle“ kommen, die keinen richtigen Kontakt hat.

Das Gerät mit dem Computer verbinden

Nachdem alle Lötstellen geprüft sind – Vorsichtige piepsen nochmal durch, ob es Kurzschlüsse gibt, das Ansehen der Lötstellen reicht allerdings meistens – wäre nun ein guter Moment zu testen, ob das Gerät noch funktioniert, bevor ihr es mit dem Computer verbindet.

Der Levelshifter

Was macht denn so ein Levelshifter nun und wieso kann ich die RX/TX-Pins nicht direkt an die serielle Schnittstelle meines Computers anschließen?

Nun, ein Levelshifter negiert die TTL-Level, die aus dem UART des Embedded Devices kommen und erhöht die Spannug auf ca. 12V. Dadurch wird es möglich, das Signal durch ein längeres Kabel zu leiten. Außerdem (genauer: deswegen)

verwendet die serielle Schnittstelle an unserem Computer solche 12V Signale. Der Computer würde also schlichtweg nicht empfangen oder interpretieren können, was direkt vom UART kommt. Hinter dem seriellen Port unseres Computers verbirgt sich übrigens auch ein Levelshifter, der die Signale wieder zurückgewandelt zum UART unseres Computers schickt. Wer in seinem Computer herumlöten will, könnte also die beiden UARTs direkt ohne Levelshifter miteinander verbinden, das Kabel muß dann jedoch recht kurz sein und die Logiken beider Geräte mit der selben Spannung arbeiten.

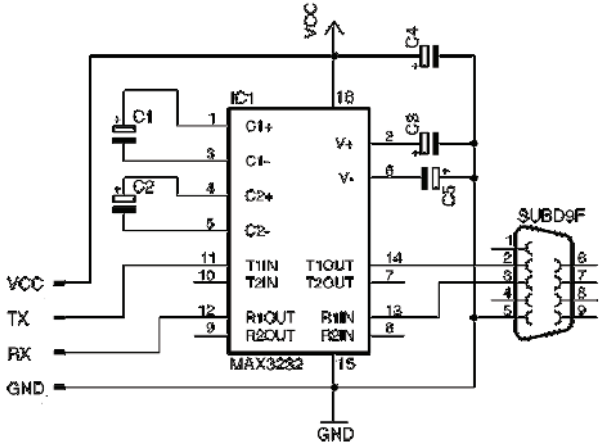
Der MAX3232 als Levelshifter

Ein oft verwendeteder IC ist der MAX232 (in der 3,3V-Variante MAX3232) oder baugleiche. Außer dem MAX benötigen wir nur noch 5 polarisierte Kondensatoren – Tantal oder Elko – je 0,1 uF beim 3232, eine 9-polige Sub-D Buchse, eine Buchsenleiste, um alles mit der Stiftleiste zu verbinden und ein bisschen Leitung. Das alles löten wir auf eine Rasterplatine – ich selbst habe mir dazu eine Leiterplatte gemacht. Es reicht hier, die Beispielschaltung aus dem Datenblatt zu verwenden, es ist lediglich zu beachten, daß die Sende- und Empfangsleitung zwischen Levelshifter und Computer gekreuzt werden – in der Schaltung oder mit einem Nullmodemkabel. Bei Segor gibt es auch hübsche kleine Nullmodemadapter in Größe eines Genderchangers.

Das Mobilfunktelefondatenkabel

Die andere Möglichkeit ist, ein Mobilfunktelefon-Datenkabel zu verwenden. In diesem ist meistens bereits ein USB- zu Seriellwandler eingebaut, der die UART-Signale als serielle Schnittstellenemulation über den USB-Port unseres Computers zu unserem Terminalprogramm schickt. Achtung! Nicht in jedem Kabel sind Chips verbaut, die wir gebrauchen können. Kabel mit dem „Prolific PL-2303“ funktionieren meistens, da es für diesen Chip ein USB2serial Treiber gibt (Linux, Mac, Win). Wenn ihr das Datenkabel an euren Computer anschließt, könnt ihr prüfen, welcher Chip sich auf dem USB meldet. Es gibt auch serielle Datenkabel mit 9poligem Sub-D Anschluss und integriertem Levelshifter, die wir an die Serielle unseres Computers anschließen können. Bei neueren Datenkabeln ist teils keine Logik mehr verbaut, die befindet sich dann im Telefon selbst. Also: vorher informieren und nachher nicht ärgern.

Um das Kabel zu nutzen, wird der Stecker der das Kabel mit dem Handy verbindet, abgeschnitten, das Kabel ein paar Zentimeter abgemantelt und ein halber Zentimeter der nun freiliegenden Adern GND, RX und TX abisoliert. VCC brauchen wir nicht, da die im Kabel verbaute Elektronik vom USB-/Seriellport mit Strom versorgt wird. Die abisolierten Kabelenden könnt ihr dann auf eine passende Buchsenleiste löten oder – wenn es von der Anordnung der Pins auf dem PCB paßt – recycelt ihr ein altes CD Laufwerksaudiokabel. Die übrigen Adern werden nicht benötigt und mit Schrumpfschlauch oder Isolierband vor Kurzschlüssen gesichert.



Beispielschaltung MAX3232 mit Sub-D Buchse und dort bereits gekreuzten TX/RX Leitungen. Wie zu sehen, nutzen wir hier nur T1/R1 des MAX, haben also für eine bei einigen Geräten vorhandene zweite serielle Schnittstelle noch T2/R2 übrig.

Die Pinbelegungen der Datenkabel lassen sich im Internet finden. Zur Not kann man diese an einer bekannten Seriellen herausfinden, vorher sollte man jedoch unbedingt GND mit dem Multimeter finden, indem man z.B. den Ground der USB/Seriell Buchse durchpiepst – und verbindt.

USB2serial Wandler selbstgebaut

Anstatt ein Datenkabel zu kannibalisieren, können wir uns einen USB2serial-Wandler auch selbst bauen. Ein sehr schöner Chip dafür ist der FTDI FT232R, für den es Treiber für Linux, Mac und Windows gibt. Auch hier reicht die Beispielschaltung aus dem Datenblatt.

Das Terminalprogramm

Um die gefundene Console zu nutzen, stellen wir unsere Parameter beim Terminalprogramm (z.B. Minicom) ein: die verwendete serielle Schnittstelle bzw. die durch USB emulierte Schnittstelle, die ermittelte Baudrate z.B. 9600, die Einstellung 8N1 ist meistens auch richtig, sowie xon/xoff. Hardwarehandshake gibt es nicht, also bitte deaktivieren, dafür Software Flow Control aktivieren.

Wir starten minicom mit `sudo minicom -s`, passen die Einstellungen an und speichern. Ist die Baudrate nicht bekannt, probieren wir nacheinander alle durch. Minicom ist manchmal etwas zickig und verschluckt sich gerne, wenn man damit herumspielt. Beim Suchen nach den Pins kann es also hilfreich sein, minicom vor jedem Versuch neu zu starten, insbesondere, wenn es vorher Buchstabensalat gegessen hat.

Bei Problemen können wir den Fehler durch Ausschluß von funktionierenden Komponenten weiter eingrenzen: Funktioniert das Terminalprogramm mit meiner seriellen Schnittstelle an einem bekannten Gerät? Funktionert der Levelshifter/das Datenkabel mit einem anderen bekannten Gerät? Haben wir vielleicht TX/RX vertauscht usw.

Wenn alles geklappt hat, sehen wir die Bootmeldungen und können uns vielleicht schon auf dem Gerät einloggen, da auf der Console meist eine Shell lauscht. Viel Spaß am Gerät! :)

The magick smoke has left my Device

Allgemeine Vorsichtmaßnahmen: Vor allem zuerst das eigene Gehirn einschalten. Lieber alles 2x prüfen, bevor wir irgendetwas einschaltet oder anschließen. Nur an der Niederspannung messen und löten! Keine unisolierten Leitungen, Levelschifter usw. herumliegen lassen. Beim Messen am eingeschalteten Gerät sehr vorsichtig sein, daß wir nicht mit den Meßfühlern abrutschen und darauf achten, daß Klemmen richtig sitzen. Wenns doch passiert ist: Nein, das ist kein Garantiefall. Manchmal hilft es, den betroffenen Smoke-Container durch ein baugleiches Teil zu ersetzen.

ChipcardLab – das multifunktionale Chipkartenlabor

Dexter <dexter@berlin.ccc.de>

Chipkartenleser und Programmiergeräte gibt es viele. Diese Geräte helfen einem aber wenig, wenn es darum geht, eine eigene Chipkarte zu entwerfen und zu testen. Gleiches gilt für die Analyse von Chipkarten. Was wir brauchen, ist ein multifunktionales, beliebig anpaßbares, skalierbares Werkzeug, das uns bei unseren Experimenten unterstützt.

Das ChipcardLab besteht aus einer Grundplatine mit einer Kontaktiereinrichtung (Afnor und ISO-Karten) und einem 25-Pol-SUB-D-Stecker. Bei beiden sind die Kontakte herausgeführt und beschriftet. Auf diese Kontakte können verschiedene Module aufgesteckt oder schlicht mit Drähten eine Verkabelung zum Sub-D-Stecker hergestellt werden, so daß eine Chipkarte mit dem Parallelport eines PCs angesprochen werden kann. Auf Basis dieser einfachen Platine, auf der sich bisher nur rein mechanische Komponenten befinden, kann man das ChipcardLab für spezielle Zwecke erweitern.

Der Distribution liegen schon einige vorgefertigte Komponenten bei. Eine Kartenimitation für ISO Karten (Überlänge), eine Kartenimitation für SIM-Karten, Module zur Benutzung des ChipcardLab mit SmartLab [1] und zuletzt das letzte und wichtigste Modul: der ChipcardController.

Der ChipcardController ist ein mit einem Atmega16 bestücktes Board, an das die Kartenimitation direkt angeschlossen werden kann. Es verfügt über eine serielle Schnittstelle und einen Parallelport, über 4 LEDs und zu guter Letzt sind die ISO-Kontakte zur Kontaktiereinrichtung und die der Kartenimitation herausgeführt, so daß man hier noch zusätzlich Erweiterungsmodule und Drähte anbringen kann. Zudem ist der Quarz in einer Fassung, so daß er sich bequem austauschen läßt. Der Takt des Quarz und der Reset vom Atmega16 sind ebenfalls über einen Pin abgreifbar.

Auf dem Mikrocontroller läuft ein Mini-Betriebsystem, das am seriellen Port eine kleine Shell zur Verfügung stellt. Man benötigt auf dem angeschlossenen PC also lediglich ein Terminal-Programm. Implementiert sind bereits ein Programm zum Auslesen von Telefonkarten, ein Sniffer (funktioniert nicht so super ;-/), der ein Logikdiagramm der Kartenkommunikation erzeugt. Ein Programm, das es einem erlaubt, die Signale am Kartenleser manuell zu setzen. Damit kann z.B. das Verhalten einer Speicherkarte evaluiert werden. Und zuletzt noch ein Terminalprogramm, das es einem erlaubt, mit einer Smartcard zu reden. Man kann damit eine beliebige Folge von Bytes an eine Smartcard senden und sich die Response anzeigen lassen. Man sollte aber bei der Software keine Wunder erwarten, funktioniert aber quasi out of the box – einfach Controller flashen und fertig! Ein bereits assembliertes Binary liegt auch bei. Es ist auch zu erwarten, dass es in Zukunft noch Updates geben wird, da ich das ChipcardLab bei meinen eigenen Entwicklungen auch einsetze.

Die Platinenlayouts stehen unter CC-Lizenz und die Software unter GPL zur Verfügung und können unter [2] heruntergeladen werden. Ich setze das ChipcardLab bei meinen eigenen Entwicklungen ein. So ist zu erwarten, daß die Software in Zukunft weiter entwickelt wird. Ich würde mich freuen, wenn sich Leute fänden, die das ChipcardLab nachbauen und nutzen würden.

[1] http://gsho.thur.de/gsho/phonecard/index.htm
[2] http://www.runningserver.com/?page=runningservercontent.thelab.chipcardlab

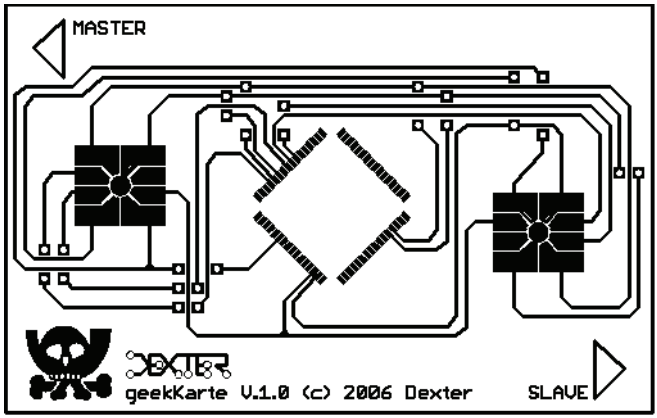
Die geekKarte – wie man Chipkarten selber baut

Philipp Fabian Benedikt Maier <dexter@berlin.ccc.de>

Es ist heute kein Problem mehr, Chipkarten mit allen erdenklichen Typen von Prozessoren und EEPROM-Speichern zu bekommen. Heutzutage führt jeder Satshop ein breit gefächertes Angebot an sog. Waverkarten. Im Einzelfall kann es jedoch vorkommen, dass die erhältlichen Waverkarten der angedachten Anwendung nicht genügen. Bei einer Waverkarte ist das Taktsignal und das Resetsignal des Prozessors direkt auf die dafür vorgesehenen ISO-7816-Kontaktflächen geführt. Das macht eine Waverkarte z.B. für die Emulation einer Speicherkarte (Stichwort: Telefonkarte, Krankenkassenkarte) mit SLE44xx unbrauchbar. Im Folgenden wird ein Verfahren erläutert, mit dem es möglich ist eine Chipkarte, die geekKarte im ISO-7816-Format herzustellen:

Die geekKarte

Die geekKarte ist eine Hybridchipkarte mit einer Master- und einer Slave-Kontaktfläche, als Prozessor kommt ein Atmega128 zum Einsatz. Die Master- und die Slave-Kontaktflächen sind bis auf die Taktleitung und die Resetleitung 1:1 miteinander verbunden. Taktsignal und Resetsignal sind bei der Master-Kontaktfläche direkt mit dem Prozessor verbunden – so wie wir es bereits von den Waverkarten gewohnt sind, während bei der Slave-Kontaktfläche die Takt- und Reset-Leitungen auf einem I/O-Pin (interruptfähig!) liegen. So kann mit dem Slave das Verhalten jeder beliebigen Speicherkarte (z.B.: Telefonkarte, Krankenkassenkarte etc.) nachgeahmt werden. Während der Master für den ganz normalen Betrieb als Smartcard ausgelegt ist und dementsprechende Einschränkungen aufweist, ist der Slave frei programmierbar. Die Karte ist pinkompatibel zur Funcard und kann mit jedem Smartcardprogrammer oder normalen In-System-Programmer (ISP) programmiert werden.



Herstellung der Platine

Um eine geekKarte herzustellen, muss zunächst das Layout gefertigt werden, dieses kann entweder mittels Isolationsfräsverfahren oder mit dem wohlbekannten Ätzverfahren hergestellt werden. Es ist sinnvoll, doppelseitige Platinen zu verwenden, auch wenn das Layout für einseitige Platinen ausgelegt ist; dann können nämlich auf der gegenüberliegenden Seite die Brücken mit einem scharfen Messer oder Schleifgerät von Hand aufgebracht werden. Zur Durchkontaktierung wird herkömmlicher

Draht verwendet, welcher dann auf beiden Seiten festgelötet wird. Die Lötstelle wird auf beiden Seiten mit einer kleinen Feile vorsichtig auf ein Minimum heruntergefeilt. Der Prozessor wird verkehrt herum in die Platine eingelassen, dazu wird vorsichtig eine Aussparung in die Platine gesägt und diese mit einer Feile so lange aufgeweitet, bis der Prozessor sich in das Loch fügt, ohne daß die Pins verbogen werden müssen. Abschließend wird der Prozessor mit einem SMD-Lötkolben festgelötet. Es reicht im Übrigen nicht, nur die tatsächlich kontaktierten Pins zu verlöten, aus Stabilitätsgründen müssen alle Pins verlötet werden. Wichtig: Das Layout darf jetzt noch nicht auf ISO-Größe zugeschnitten werden, dies ist der letzte Schritt: Wir brauchen die überstehenden Ränder noch.

Prozessor abschleifen

Als nächstes wird der Prozessor auf der Platinenrückseite mit Schmirgelpapier heruntergeschliffen. Es ist wichtig, hier etwas Feingefühl walten zu lassen und nicht zuviel abzuschleifen, da sonst die Bonddrähte beschädigt werden könnten. Es ist nicht schlimm, wenn die Karte später etwas dicker ist als ISO 7816 vorschreibt, da der überwiegende Teil der Lesegeräte tolerant gegenüber etwas dickeren Karten sind. In den allermeisten Fällen wird die Karte ohnehin nur knapp bis zur Hälfte eingeschoben. Auf jeden Fall muss der Prozessor, sowie die gesamte Platine nach dieser Prozedur mit einem Testprogramm elektrisch überprüft werden.

Versiegeln

Nun wird der Prozessor auf der Platinenrückseite mit Klebeband abgeklebt und die Platine auf einer Laborwärmeplatte fixiert. Die Kontaktflächen (Master und Slave) werden ebenfalls mit einem kleinen viereckigen Stück Klebeband abgeklebt.

Die Platine wird dann an den Rändern großzügig mit Klebeband auf die Wärmeplatte geklebt. Vor dem Abkleben sollte die Platine jedoch noch einmal gründlich mit Aceton gereinigt werden. Wenn die gesamte Platine durchgeheizt ist, wird auf der linken Seite der Platte auf dem Klebeband Epoxidharz (z.B.: Uhu Schnellfest) angerührt und mit einem Spachtel gleichmäßig über die Platine gezogen. Sofort danach wird mit einer Pinzette vorsichtig das Klebeband von den Kontaktflächen (Master und Slave) abgezogen. Das Ergebnis sollte eine hauchdünne Beschichtung mit Epoxidharz sein. Die Wärme der Platte lässt das Epoxidharz sehr schön dünnflüssig und streichfähig werden, zum anderen hat die Wärme noch einen härtenden Effekt. Nach einigen Minuten sollte das Harz ausgehärtet sein, und die Platine kann von der Platte gelöst werden.

Covern

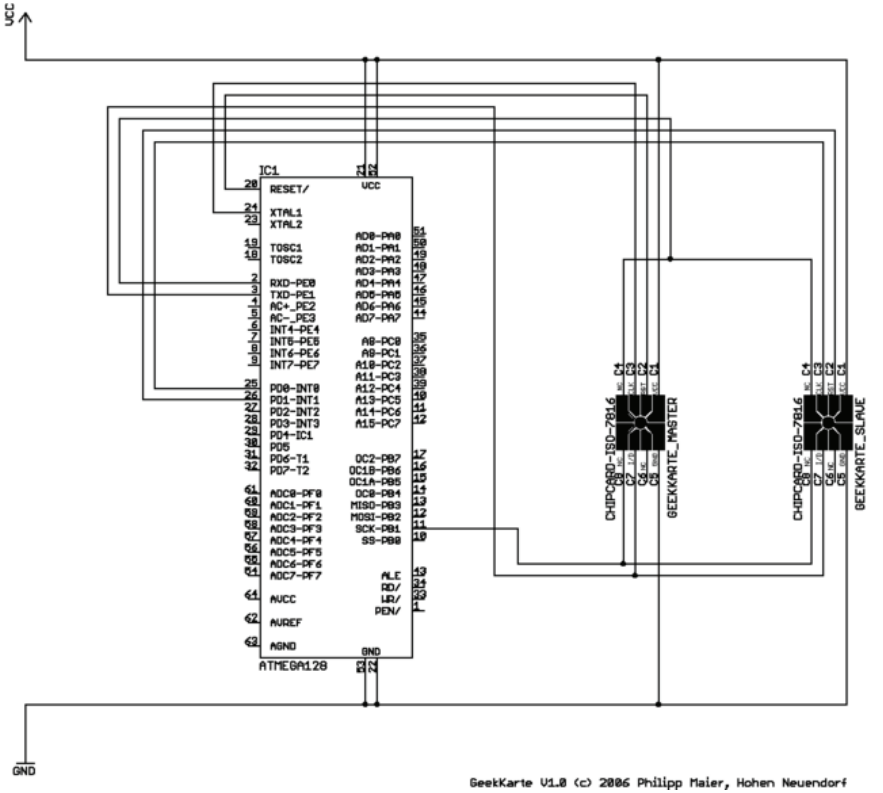
Nun wird mit einem Laserdrucker das Kartencover auf Papier gedruckt und auf die Platine ausgerichtet, damit man, wenn die Platine im nächsten Schritt umgedreht auf der Wärmeplatte liegt, das Cover auch richtig anbringen kann. Bevor wir die Platine jedoch wieder auf der Wärmeplatte fixieren, ziehen wir das Klebeband vom Prozessor ab. Schnell wird auch klar: Wenn wir den Prozessor nicht abgeklebt hätten, wäre das Epoxidharz an den Rändern des Prozessors durch die Platine gelaufen und hätte die

Karte mit der Wärmeplatte verklebt. Es ist ratsam, falls sich durch das Epoxidharz am Prozessor scharfe Kanten gebildet haben, diese mit Schmirgelpapier zu glätten. Im Übrigen ist es ratsam, die Platte zum Schutz vollständig mit Klebeband abzukleben. Wenn die Karte fixiert ist, tragen wir zwei Würste Epoxidharz links neben der Karte auf. Die erste Wurst ziehen wir wie schon vorhin mit einem Spachtel über die Platine. Danach wird das Cover auf die Platine gelegt und vorsichtig angedrückt. Zum Abschluss wird die zweite Wurst über das Cover gezogen und somit in die Karte einlaminiert. Wenn das Harz ausgehärtet ist, kann die Platine von der Wärmeplatte gelöst werden und auf ISO-Größe zurechtgeschnitten werden.

Conclusio

Wir haben nun eine vollwertige, freiprogrammierbare Chipkarte in ISO-Größe. Es sei hier noch erwähnt, dass man im Speicherkartenbetrieb den internen RC-Oszillator des Atmega128 aktivieren muss. Im Smartcardbetrieb ist es jedoch unerlässlich, den vom Terminal bereitgestellten Takt zu verwenden, da es sonst unweigerlich zu Timing-Konflikten kommt. Als Kartenbetriebssystem für die geekKarte kann entweder ein selbstgeschriebenes C- oder ASM-Programm verwendet werden. Es sei hier schonmal vorsichtig erwähnt, dass ich bereits an einem auf der geekKarte (und anderen Standardkarten) lauffähigen Chipkartenbetriebsystem arbeite. Aber ich kann nicht versprechen, dass etwas dabei herauskommt. Die Funktionen zum Senden und Empfangen von Bytes nach T=0 und das Ausgeben eines ATR-Strings sind jedenfalls schon fertig. Weitere Informationen (Schaltbilder, Layout ect.) können über meine Webseite bezogen werden:

http://www.runningserver.com/



Veränderung von Sendeanlagen kleiner Leistung

Christian Berger <casandro_lion@web.de>

Es gibt Gründe, kleine Sendeanlagen zu modifizieren. Sei es, damit man mit seinem Funkkopfhörer nicht die Geräte seiner Nachbarn stört, oder um bei einem Vortrag den Ton des Laptops über die Funkmikrophonanlage zu senden. Worauf ich hier nicht wirklich eingehe, sind Datenübertragungsverfahren. Allerdings können einige auf andere Verfahren zurückgeführt werden. Dieser Artikel soll nicht weit in die theoretischen Details gehen, sondern nur die Möglichkeit der Modifikation aufzeigen. Sender mit Zwischenfrequenzen werden hier nicht behandelt.

Am einfachsten kann man die Frequenz verändern. Jeder Sender benötigt eine Quelle für seine Frequenz. Der wohl einfachste Weg, dies zu erreichen ist es, einfach diese Frequenz aus einer externen Quelle zu beziehen. RFID-Tags arbeiten so. Hier muss man nur diese externe Quelle beeinflussen.

Diese Sender sind aber nur sehr schwach und somit für uns uninteressant. Die nächst aufwändigere Methode sind RC- und LC-Oszillatoren. Hier wird die Frequenz durch eine Kombination aus 2 Bauteilen bestimmt. RC-Oszillatoren sind im Funkbereich eher selten. Hier wird der Ladevorgang eines Kondensators dazu benutzt um den Entladevorgang zu starten. Dieses kippende Verhalten kann gefiltert werden, um ein brauchbares Sendesignal zu erzeugen.

Die Frequenz ist hier proportional zum Produkt aus des Wertes des Widerstandes (R), sowie des Kondensators (C). LC-Oszillatoren schwingen, in dem sie das Ausgangssignal eines Verstärkers geschickt rückgekoppelt wird. Am Ausgang steht dann ein, mehr oder weniger sinusförmiges Signal zur Verfügung. In LC-Oszillatoren sind eine Spule (L), sowie ein Kondensator (C) in der Regel parallel geschaltet.

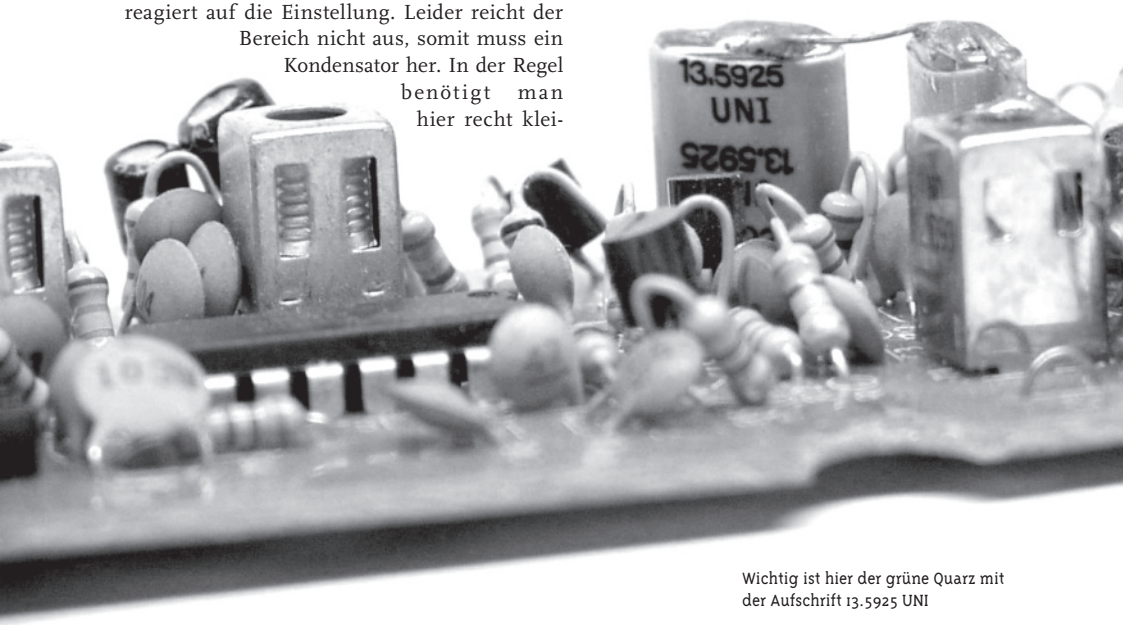
Die Frequenz ist proportional zu $\frac{1}{\sqrt{LC}}$. Häufig ist mindestens die Spule nachstellbar. Diese ist dann meistens eine kleine Metall"kiste" mit einem runden Loch oben, hinter dem sich ein Ferritkern befindet. Diesen kann man mit einem Schraubendreher hinein- und hinausdrehen. Dadurch verändert sich, in gewissen Grenzen, die Induktivität der Spule und somit die Frequenz des Senders. Reicht dies nicht, so kann man einen zusätzlichen Kondensator parallel schalten. Dies erhöht die Kapazität des Schwingkreises und verringert somit die Frequenz. So ein Kondensator kann im einfachsten Falle aus 2 isolierten Drähten bestehen, die miteinander verdrillt sind. Oder auch einem Stück Koaxial- oder Kopfhörerkabel. Besonders frequenzstabil sind quarzbasierte Sender. Diese nutzen ein kleines Piezokristall, das elektrisch zu mechanischen Schwingungen angeregt wird und dessen Schwingungen elektrisch abgegriffen werden. Die Frequenz eines Quarzes kann man (nur ein klein wenig) verändern. Aber, es ist möglich den Quarz auszuwechseln. Hat man einen anderen Oszillator mit einer passenden Frequenz, so kann man häufig auch den Quarz auslöten und das Signal an einen der Anschlüsse anlegen. Eine besonders interessante Klasse von Oszillatoren sind PLLs. Diese Oszillatoren haben einen Referenzoszillator (meistens Quarzoszillator) mit einer, in der Regel relativ niedrigen Frequenz (z.B. 1MHz). Ein zweiter Oszillator mit einer in der Regel höheren Frequenz ist abstimmbar. Der Takt des zweiten Oszillators wird geteilt und das Ergebnis wird mit dem ersten Oszillator verglichen. Mit dem Ergebnis

wird der 2. Oszillator abgeglichen. Diese Art von Oszillatoren ist sehr stabil. Häufig kann man das Verhältnis des Taktteilers einstellen, manchmal sogar per I²C. In jedem Falle kann man jedoch die Referenzfrequenz durch eine alternative Frequenz ersetzen.

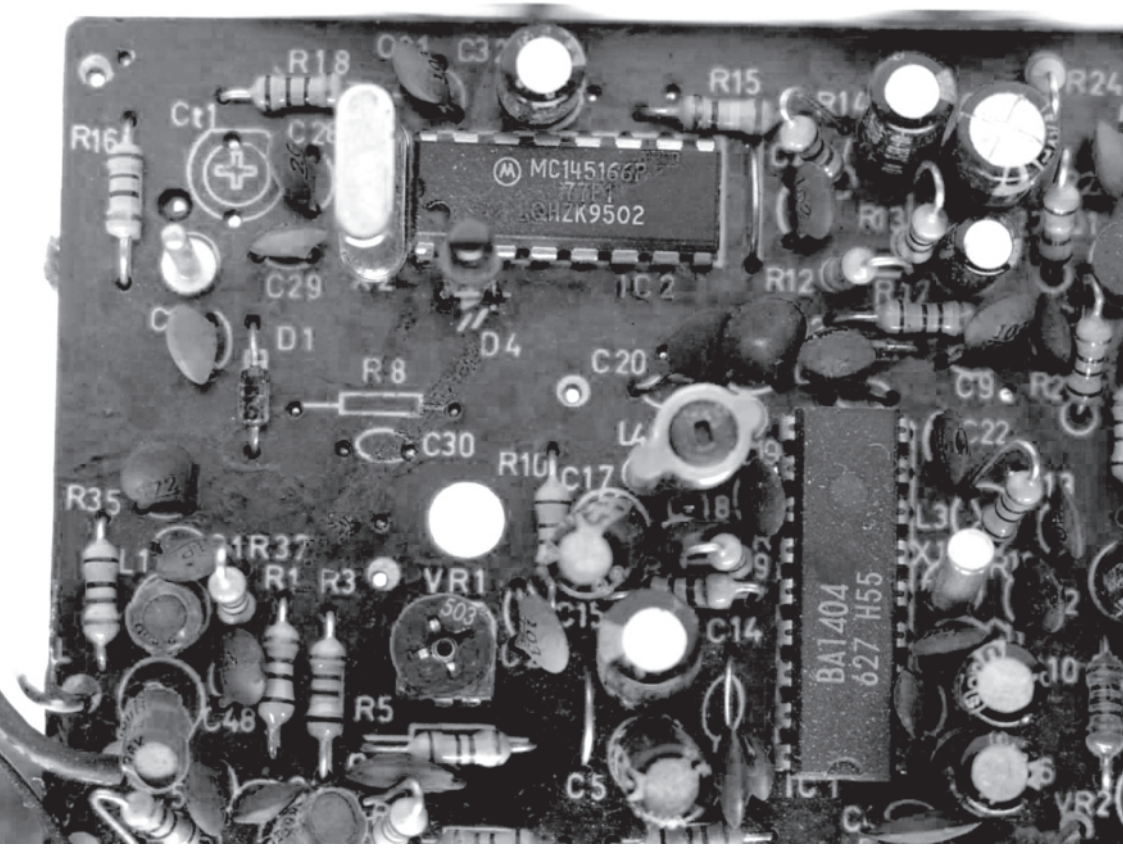
In jedem Falle sollte man die Frequenz nicht zu stark verändern, da nachfolgende Bauelemente auf bestimmte Frequenzen ausgelegt sind. Wie weit das wirklich geht kann man so pauschal nicht sagen. Was man nicht kann, ist die Modulationsart zu verändern. In der Praxis ist das aber nicht notwendig, da viele Geräte bereits den richtigen Modus haben, oder der Empfänger auch Signale mit einer falschen Modulationsart erträglich gut moduliert. Ein Schmalband-FM Empfänger kann zum Beispiel sehr leise Breitband-FM Sendungen empfangen und umgekehrt.

Ein paar praktische Beispiele: Ich möchte einen Sender für Funkkopfhörer, der etwa auf 42 MHz sendet auf die Frequenz von bestimmten Funkmikrophonen (ca. 39 MHz) einstellen. Der Sender hat eine einstellbare Spule und der Frequenzzähler, den man an die Antenne hängt reagiert auf die Einstellung. Leider reicht der Bereich nicht aus, somit muss ein Kondensator her. In der Regel benötigt man hier recht kleine Werte, häufig keramische Kondensatoren. Natürlich ist das hauptsächlich eine Frage der Experimentierfreude. Natürlich muss man dies in einem speziellen Bunker machen, aus dem keinerlei elektromagnetische Wellen entrinnen können. Sinnvollerweise überprüft man noch ob der Sender nicht noch auf anderen Frequenzen sendet, in dem man ganzzahlige Vielfache der Frequenz einstellt und hofft, nichts zu hören.

Billige CB-Funkgeräte senden in der Regel auf Kanal 19 (27,1850 MHz) im Modus Schmalband-FM. Diese Frequenz wird durch einen 13,5925 MHz Quarz erzeugt, der mit der 1. Oberwelle betreiben wird. Er läuft somit auf der doppelten Frequenz. Viele schwingfähige Systeme lassen sich auf Oberwellen betreiben, beispielsweise auch Saiteninstrumente. Ein Verstellen der Regler hilft hier nichts. Auch ein einfaches Parallelschalten von Kondensatoren ist zwecklos, da die Frequenz nur maßgeblich durch den Quarz bestimmt wird. Hier hilft es nur, den Quarz auszuwechseln. Einen anderen Oszillator anzuschließen hilft nichts, da die Frequenzmodulation durch eine Kapazitätsdiode im gleichem Kreis realisiert wird.



Wichtig ist hier der grüne Quarz mit der Aufschrift 13.5925 UNI



Wichtig sind hier IC2, der Chip oben und X2, das Quarz daneben.

Hier ist ein anderer Sender für Funkkopfhörer. Gut zu erkennen ist hier ein Chip mit der Bezeichnung MC145166 direkt neben einem Quarz. Der Quarz hat eine Frequenz von nur 7,6194 MHz, obwohl der Sender auf etwa 36 MHz sendet. Die Lösung steckt im Chip. Eine kurze Suche im Internet fördert das Datenblatt dieses Chips zu Tage. Es handelt sich hierbei um einen PLL, wobei der Oszillator sich auserhalb des Chips befindet. Dieser Chip wurde ursprünglich für US Schnurlostelefone entwickelt, kann aber mit einem anderen Quarz auch für diesen Zweck verwendet werden. 4 Pins wählen einen von 10 vorprogrammierten Teilerfaktoren aus. Lötet man den Chip aus, so kann man diese Pins kleine Schalter anbringen, die die nötigen Pegel anliegen lassen.

Ein paar Worte zum Umgang mit den Messgeräten. Frequenzzähler neigen dazu falsche Werte anzuzeigen, wenn die Empfindlichkeit falsch eingestellt ist. Dies rührt daher, dass das Eingangssignal verstärkt wird, und dann mit einem Vergleichswert verglichen wird. Ein Zähler zählt nun die Anzahl der Übergänge von „drunter“ nach „drüber“. Da die Amplitude des Signales schwankt, kann es passieren, dass nicht alle Perioden gezählt werden, und somit ein zu kleiner, und stark schwankender Wert angezeigt wird. Auch kann es sein, daß das Messgeräte Oberwellen oder seltsame Mischwerte anzeigt. Hier gilt es zu probieren.

ICMP3 – Die Freiheit nehm ich mir

Martin Haase <maha@berlin.ccc.de>

Was lockt 150 Nerds im August ins mittelfränkische Münchsteinach? Die (manchmal zu) frische Luft? Der klare Sternenhimmel? Die Nähe zur Quelle der Club-Mate [1], die hier produziert wird? Die schöne Landschaft? Die Abgeschiedenheit? Die Herausforderung, hier ins Internet zu kommen?

Sicher sind das alles gute Gründe. Der Hauptgrund dürfte aber darin bestehen, dass hier vom 3. bis 8. August 2006 die 3. ICMP [2] stattfand. Wofür die Abkürzung ICMP in diesem Zusammenhang steht, verschweigen die Veranstalter, der CCC-Erfa-Kreis Erlangen bits'n'bugs. Sagen wir der Einfachheit halber mal: Intergalaktische Club-Mate-Party, jedenfalls wurden so ungefähr 100 Kästen dieses Erfrischungsgetränks ausgetrunken, was sich natürlich belebend auf die Stimmung auswirkte. Allerdings konnten auch andere Produkte der Brauerei genossen werden, viele im Partnertarif, das heißt: wer zwei Getränke kauft, zahlt weniger.

Praktischerweise fand das Vortrags- und Workshop-Programm unter dem Motto „Die Freiheit nehm ich mir" überwiegend abends statt, sodaß man tagsüber den Sommer genießen konnte, soweit er zu genießen war, denn vor allem am Freitag und am Sonntag gab es Starkregen, der wohl an die „What the Hack" [3] erinnern sollte. Dann wurde es aber wieder sommerlich und die Wetterverhältnisse konnten sich doch nicht mit der „What the Hack" messen. Dafür gab es auch weniger Mücken, Bremsen oder Wespen, aber auch im Chaos Emergency Response Team (CERT) [4] kam wenig Langeweile auf.

Vor lauter Chillen und Grillen (unter anderem ein Spanferkel) bestand natürlich das Risiko, auch Highlights des Abendprogramms zu verpassen. Wer sich dennoch aufraffte, konnte bei einer Einführung in die Computer- und Netzwerkforensik eine Menge lernen, was dann gleich bei einem Hackerwettbewerb wieder unter Beweis zu stellen war. Überhaupt durchzogen Wettbewerbe das gesamte Programm: Es gab eine Schnitzeljagd (von Access Point zu Access Point), die aufgrund des wankelmütigen Wetters zwischendurch auch Züge einer Schlammschlacht hatte, zudem gab es ein richtiges Quiz, das die Blinkenarea-Leute organisiert hatten, und eben jeden Tag Aufgaben, mit denen die Teilnehmer Spaß am Gerät haben konnten. Übrigens nicht nur mit digitalem Gerät: Auch ein semiprofessionelles Teleskop war am Start, zahlreichen blinkende Lichter und Streß-Bälle, die immer mal wieder durch die Gegend flogen. Eine digitale Überraschung gab es auch: Diesmal funktionierten Netz und Internet praktisch einwandfrei! Und das auf dem Acker! Überhaupt war die gesamte Organisation sehr professionell.

Schon beim letzten Mal vor zwei Jahren dachte man, die Kapazitäten der ICMP seien mit 100 Teilnehmern erschöpft, allerdings fiel es kaum auf, daß diesmal 50 Teilnehmer mehr kamen. Dank sanitärer Einrichtungen, Duschen, Strom, Eventphone [5] und Internet gab es eigentlich jeden nur erdenklichen Komfort. Hinzu kam der eine oder andere kulinarische Höhepunkt zwischen indischem Dal und fränkischem Grillgut.

Insgesamt ist so eine ICMP genau die richtige Methode, eine entspannende Sommerwoche zu verbringen: Internet an frischer Luft und Entspannung mit Lerneffekt. Was will man mehr? Dazu kommt man dank der familiären Atmosphäre mit praktisch allen Teilnehmern ins Gespräch und fährt ungern wieder heim. Dafür kommt man beim nächsten Mal gern wieder.

[1] http://www.club-mate.de/
[2] http://www.icmp3.de/
[3] http://www.whatthehack.org/
[4] http://www.c-e-r-t.de/
[5] http://www.eventphone.de/

mrmcd101b „mission possible“

wonderer, gopher und andere

Der CCC ist in der Metropolregion RheinMain mehrfach mit verschiedenen Chaostreffs und Erfakreisen vertreten. Die Region Rhein-Main ist ein städtischer Ballungsraum im Süden Hessens sowie Teilen der angrenzenden Bundesländer Rheinland-Pfalz (Rheinhessen) und Bayern (Unterfranken). In der Regel treffen sich die Mitglieder dieser einzelnen Treffs einmal wöchentlich in ihren eigenen Reihen. Jeder dieser Treffs ist anders und hat seinen eigenen Charme. Der eine Treff nutzt z.B. die Räumlichkeiten eines Jugendzentrums, ein anderer trifft sich im Cafe „Club Voltaire“ und ein weiterer nutzt die Infrastruktur der Hochschule.

Vor einigen Jahren verabredete man einen regelmäßigen Austausch unter den einzelnen Chaostreffs. Die verschiedenen Mailinglisten der Clubs laufen auf einem zentralen Serversystem zusammen und auch im IRC gibt es einen gemeinsamen Channel (#metarheinmain) im ircnet. Ein Austausch findet aber dennoch über die jeweils eigenen Channels der entsprechenden Chaostreffs statt. Um sich direkt auch mal treffen zu können, sich im Rhein-Main Gebiet austauschen zu können, weil die Wege hier recht kurz sind, wurde ein regelmäßiges Zusammenkommen vorgeschlagen und man nannte dies die Meta-Rhein-Main-Chaos-Days (mrmcd). Jeweils ein Chaostreff sollte im Wechsel diese mrmcd organisieren.

In Wiesbaden traf man sich zu den mrmcd100b Ende Juli und in Darmstadt traf man sich dann also am ersten Septemberwochenende zu den insgesamt fünften mrmcd, den mrmcd101b (die Durchnummerierung der mrmcd findet hierbei in binärer Schreibweise statt...).

Wie bereits im letzten Jahr erhielt der Chaostreff Darmstadt im Vorfeld der Veranstaltung die Ressourcen der Technischen Universität Darmstadt zugesichert. Aufgrund von Baumassnahmen stand dieses Jahr allerdings das Audimax (Veranstaltungsort der mrmcd11b im vergangenen Jahr) nicht zur Verfügung und auf intervenieren des ET-Dekans musste nach Absage des Gebäudes des Fachbereichs Elektrotechnik kurz vor Veranstaltungsbeginn eine weitere Alternative gesucht werden, die im neu renovierten Informatik-Gebäude letztend-

lich gefunden wurde. Dieses Gebäude erwies sich als Glücksgriff, ist das Gebäude doch mit den neuesten Hörsaaltechniken, sowie Netzwerk- und Infrastruktur ausgestattet. Der Organisationsaufwand für die Vernetzung des Hackcenters mit LAN und Strom fiel dadurch weg. Die Netzanbindung war über eine Leitung an das Regionale Hochgeschwindigkeitsnetzwerk der Wissenschaftseinrichtungen im Raum Darmstadt (MANDA – Metropolitan Area Network Darmstadt) gesichert.

Die in den Hörsälen integrierte Video- und Vorlesungsaufzeichnungsanlage erlaubte etliche Spielmöglichkeiten und das bereitstellen von einzelnen Vortagsvideos.

Mit den zugesicherten Ressourcen und dem verhältnismäßig geringem Planungsaufwand für die Infrastruktur konnte der ChaosTreff Darmstadt in dieser Veranstaltung deutlich mehr Akzente auf die Inhalte und das äußere Erscheinungsbild der Veranstaltung legen. Der Charakter der Veranstaltung sollte vielmehr einer Konferenz nachkommen, anstatt einen zweiten Kongress zu etablieren. Durch Nutzung der CCC Ressourcen wie das POC (PhoneOperationCenter), das CERT (ChaosEmergencyResponseTeam), dem Engelsystem für die Koordination der vielen Helfer via Infotresen, dem T-Shirt Verkaufsstand und Pentabarf zur Organisation und Virtualisierung der Vortragsthemen und Zeiten viel das dem einen oder anderen Besucher eher schwer dies zu unterscheiden. Aber der fast schon bei CCC-Veranstaltungen obligatorische OpenBSD-Stand von Wim Vanderputten und der T-Shirt Stand von Freddruck bei dem man sich personalisierte mrmcd101b T-Shirts mit Nick-Name drucken lassen konnte lockerten das Bild etwas auf.

Mit der Veranstaltung im Umfeld der Technischen Universität Darmstadt wurde der wissenschaftliche Hintergrund durch die spezialisierten Kenntnisse der Computerszene besonders ausgeleuchtet.

Anwendern, Entwicklern und Wissenschaftlern hatten dabei die Möglichkeit Erfahrungen, Informationen und Forschungsergebnisse auszutauschen. Durch diese offene Veranstaltungsstruktur ergibt sich eine besondere Chance gesellschaftskritische Aspekte von IT-Umsetzung darzustellen, ohne das Fachumfeld aus dem Blick zu verlieren. Aktuelle Großprojekte verlangen nach einer Plattform für eine Bewer-

tung solcher Vorhaben. Den Schwerpunkt der Konferenz bildeten die Themen Biometrie, Überwachung, Kryptographie und IT-Sicherheit.

Zahlreiche Vorträge zu diesen Themen wurden an den drei Konferenztagen angeboten. Nennenswert war gleich zu Beginn am Freitag eine Lesung von Rick Dakan, US Autor des Buches „Geek Mafia“, der mit einigen anderen Kollegen extra aus den USA angereist kam. Einige Kollegen u.a. des Darmstädter Chaostreffs hatten im Vorfeld bei ihren besuchen der Hope und der Defcon für den CCC in Deutschland geworben und somit internationales Publikum begeistern können. Darüberhinaus kamen außerdem noch Kollegen aus den Niederlanden und Italien angereist.

Neben den Vorträgen um die Themen Security, Kryptografie & Co, gab es Vorträge über Comparison of WAN IGP Protocols, Nintendo DS, Hacktivismus und die Möglichkeiten politischer Einflussnahme, Sicherheit und Angreifbarkeit heutiger Applikationen, Vorstellung einiger WLAN & Wardriving Fahrzeuge und Trusted Computing for Java. Ein weiteres Highlight waren dieses Jahr sicherlich der Spaßvortrag „Powerpoint Karaoke“, bei dem sich viele Anwesenden aktiv beteiligten (siehe auch Videos bei youtube.com), sowie die Lightning Talks. Traditionell gab es auch wieder die Keysigning Party (Cacert und GPG). Über eine Zeitspanne von geplanten 3 Stunden und am ende ca. 10 Stunden fand ein Antennenbauworkshop „Bau einer Helix Antenne“ statt und es gab für interessierte einen Amateurfunk-Crashkurs.

Ein weiteres Großprojekt neben den mrmcd im Rhein-Main Gebiet stellt das c-radar, das Chaos-Radio-Darmstadt dar. Normalerweise trifft man sich immer am 1. Donnerstag eines Monats in dem in Darmstadt ortsansässige Radiosender „Radio Darmstadt“ und sendet von dort aus Nachts zwischen 23:05 – 02:00 Uhr. Zum Auftakt der mrmcd101b wurde außer der Reihe am Donnerstag vor der Veranstaltung ebenfalls gesendet und damit der Startschuss für ein dreitägiges Dauerprogramm gelegt. Während der Veranstaltung war der Live Radio-Stream des Chaostreff eigenen Radiosenders c-radar jederzeit präsent und konnte als Livestream angehört werden und wurde dann zeitweise auch in das Liveprogramm von Radiodarmstadt eingespeist, und zur Nachtzeit sogar komplett übernommen.

Erstmals wurde auf den mrmcd in Darmstadt ein Partyabend veranstaltet und der Live-Auftritt der Gruppe „pornophonique“ erwies sich als absolutes Highlight. Pornophonique beschreibt sich auf ihrer Webseite am trefflichsten: „eine gitarre. ein gameboy. ein c64. zwei typen. space invaders samples. lagerfeuergeschrabbel. gameboygedudel. auch gesang mit dabei. lowtech. lo-fi. micro. porn.“. Wir können gespannt sein, wenn Pornophonique in der Dezemberausgabe das nächste mal live im Studio in der C-Radar Sendung auftreten wird. Die Party war danach aber lange noch nicht zu ende, denn drei live-DJs die zum Teil aus den eigenen CCC-Kreisen kamen legten bis in die frühen Morgenstunden Ihre DJ-Sets auf.

Einige Startschwierigkeiten hatte der Versuch von b9punks eine „US Bucket-Party“ zu etablieren. Mittels Drinks aus dem Bucket for Free, Spenden war natürlich erwünscht, sollte das Publikum aufgeheitert werden. Letztendlich war dies aber nach dem Auftritt von Pornophonique überhaupt nicht mehr nötig, und die Anwesenden erfreuten sich doch mehr an dem Fassbier und den Mate-Chunks an der Cocktailbar. Mit den Beitrag von b9punk haben wir einen Teil zur Völkerverständigung beitragen können: „Also our US guys and girls know how to party and now also know what a „Pfläumchen“ is, right?!“

Nach der langen Partynacht fiel leider die Ausrichtung des TY WEBB MEMORIAL INVITATIONAL GOLF DINGSBUMS aus, tagsüber fanden sich aber genug Nerds, die im angrenzenden Herrngarten ihre Golfschwünge verbesserten.

Während der Veranstaltung gab es ausreichend Verpflegung über das eigens organisierte Cateringteam für Fleischesser und Vegetarier mit fairen Preisen, die zum Schluss noch mit einem

Mate-Deal (Mate für die Heimreise) aufwarten konnten - insgesamt ging somit bei der Veranstaltung 120 Kästen Mate über den Tresen. Das räumliche Angebot des Caterings genügend Bierbänke am Übergang von mrmcd zum angrenzenden Herrngarten, einem schönen Park inmitten von Darmstadt, bereitzustellen und dank des klasse Wetters mit Sonnenschein und angenehmen Temperaturen machte das extra eingerichtete Hackcenter fast überflüssig. Die meisten hielten sich wenn nicht gerade in den Vorträgen in diesem Bereich auf und förderten somit auch den gemeinsame Unterhaltungswert einer solchen Veranstaltung. Mittels mobilen Geräten und WLAN konnte man auch von hier alle Vorzüge des Hackcenters genießen.

Die angestrebte Zielgruppe, der breite Kreis der Öffentlichkeit konnte mit der Veranstaltung nur zum Teil erreicht werden, bot sich mit dieser Veranstaltung doch die Gelegenheit wissenschaftliches Know-how von Universitäten und „Insidern“ direkt vermittelt zu bekommen. Die Veranstaltung verzeichnete zwar einen neuen Besucherrekord, blieb dabei aber hinter den eigenen Erwartungen zurück. Im Umfeld der TU Darmstadt, und insbesondere bei den Studenten der Informatik gab es nicht den erhofften Zuspruch an Tagesgästen. Letztendlich waren die CCC'ler wieder unter sich und man spürte nach dem Wochenende den Geist des MRMCD Gedankens wieder. Da sich die mrmcd in Darmstadt durch den Konferenz Charakter deutlicher von den anderen mrmcd unterscheiden, wurde zwischen den Chaostreffs am Regiotreff vereinbart, die mrmcd in Darmstadt auch im nächsten Jahr in dieser Form weiterführen zu wollen. Die Zusammenkunft der einzelnen Chaostreffs miteinander sollen in kleinerer Form, aber dafür mit verkürzten Abständen, unter der Bezeichnung MetaRheinMainTage (mrmt) weitergeführt werden. Bleibt abschließend natürlich noch allen Beteiligten, sei es Organisatoren, Helfern, CCC'lern und Besuchern zu danken. Auf das wir uns nächstes Jahr noch steigern können und eine noch bessere Party zustandekommt.

Die MetaRheinMain Chaosdays 110b finden übrigens vom 14-16.09.2007 in der Technischen Universität Darmstadt statt. Schwerpunkte der Veranstaltung sind elektronische Wahlmaschinen, Überwachung, Kryptographie und IT-Sicherheit. Drei Tage lang werden ein Hackcenter, Vorträge und Diskussionen mit wissenschaftlichen wie auch kommerziellen Anwendern und Entwicklern von Sicherheitstechnik geboten.

Links

mrmcd101b: *http://mrmcd101b.metarheinmain.de/*
POC *http://www.eventphone.de/*
c-radar: *http://www.c-radar.de/*
pornophonique: *http://www.pornophonique.de/*
Freddruck: *http://www.freddruck.de/*

Systeme mit Systrace härten

Stefan Schumacher <stefan@net-tex.de>

Systrace ermöglicht die Überwachung und Steuerung von Systemaufrufen. Dazu benutzt es Richtlinien, die für jedes verwendete Programm definiert werden. Anhand dieser Richtlinie werden Systemaufrufe erlaubt oder verboten. Ebenso kann man einzelne Systemaufrufe unter anderen Benutzerrechten ausführen. Somit ist es möglich, SETUID-Programme als unprivilegierter Benutzer zu starten und nur bestimmte Systemaufrufe mit Root-Rechten auszuführen. Systrace erlaubt daher die Implementierung einer feingranulierten Sicherheitsrichtlinie, die sogar Argumente von Systemaufrufen überprüfen kann.

Dieser Artikel beschreibt die Funktionsweise von Systrace, Aufbau und Erzeugung einer Richtlinie sowie den praktischen Einsatz anhand von zwei Beispielen auf NetBSD.

Systemaufrufe mit Systrace steuern

Wie inzwischen allgemein anerkannt ist, setzt eine erfolgreiche Verteidigung eines Rechner-Systems mehrere Verteidigungslinien voraus. Diese Linien müssen sich überlappen, ohne jedoch voneinander abzuhängen.

Dies können beispielsweise Paketfilter, Application-Level-Gateways oder biometrische Zugangskontrollen sein. Allerdings bietet jede Verteidigungslinie wiederum neue Angriffspunkte. So könnte beispielsweise ein Einbruchserkennungssystem mittels Speicherüberlauf übernommen und mit Root-Rechten mißbraucht werden. Daher ist es notwendig, die Schadmöglichkeiten von Programmen einzugrenzen.

Nahezu jedes heute eingesetzte Programm ist zu komplex und umfangreich, um sorgfältig auf Fehlerquellen und Sicherheitslücken überprüft zu werden. Selbst wenn es als quelloffenes Programm vorliegt, überprüft in der Regel niemand den Quellcode auf absichtliche oder fahrlässige Sicherheitsprobleme. Quelloffenheit ist zwar ein sehr gutes Kriterium für sicherheitsrelevante Programme, schützt aber definitiv nicht vor Lücken oder Hintertüren, wie (Thompson, 1984) beweist. Teilweise ist man aber auch auf den Einsatz geschlossener Software angewiesen und hat dann keinerlei Möglichkeit mehr, diese zu prüfen und muß ihr zwangsläufig trauen. Angriffe gegen Systeme konzentrieren sich in der Regel auf Systemaufrufe (auch System Calls oder Syscalls genannt), die unter anderem auch dazu verwendet werden können, um im Kernel privilegierte Operationen durchzuführen. Um derartigen Vorgehensweisen zu begegnen, wird die Angriffsfläche reduziert. Dazu wird ein Programm eingeführt, das die Ausführung von Systemaufrufen begrenzt und die Zugriffsrechte darauf feiner als bisher granuliert.

Hierzu benötigt man eine Richtlinie, die die Zugriffe reglementiert. Die Richtlinie muß alle möglichen Fälle abdecken und auch Kenntnisse von allen möglicherweise auftretenden Pfaden haben, was Dank symbolischer Links nicht besonders einfach ist.

Die Richtlinie beschreibt das Normalverhalten eines Prozesses und dient so als Vergleich zum laufenden System. Weicht ein Prozeß von der beschriebenen Richtlinie ab, wird dies als Einbruchsversuch erkannt und verhindert. Außerdem soll das überwachende Programm auch Überwachungsprotokolle der überwachten Programme erzeugen, um so Anomalien erkennen und analysieren zu können.

Ein derartiges System läßt sich auf verschiedene Arten implementieren. Einerseits kann es komplett im Kernel arbeiten, andererseits auch kom-

plett im User-Space. Im Kernel ist die Ausführung recht schnell, aber das System selbst ist äußerst komplex und nur sehr schwer auf andere Betriebssysteme zu portieren. Im User-Space hingegen ist das Programm portabel, aber sehr langsam und unsicher, da es zu Race-Conditions zwischen dem Zeitpunkt der Analyse und der Ausführung eines Systemaufrufes kommen kann.

Systrace, von seinem Entwickler Niels Provos in (Provos, 2006) und (Eriksen und Provos, 2003) beschrieben, verwendet einen hybriden Ansatz. Es wird ein kleiner Teil im Kernel implementiert und der größte Teil im User-Space. Der Kernel-Teil ermöglicht die sehr schnelle Behandlung von kontextinsensitiven Systemaufrufen, die bspw. stets abgelehnt oder erlaubt werden. Weiterhin ermöglicht dies, auszuführende Programme in einem Sandkasten zu kapseln und geforkte Prozesse mit der vererbten Richtlinie weiter zu kontrollieren.

Der Teil im User-Space überwacht die Systemaufrufe auf kontextsensitive Entscheidungen und trifft sie anhand der definierten Richtlinie. Während der Entscheidungsfindung blockiert der Kernel den fraglichen Prozeß. Weiterhin kann der User-Space-Dæmon über eine Schnittstelle Informationen wie Zustandsübergänge, PID-Änderungen oder Forks vom Kernel-Teil anfordern.

Um Funktionen des Kernels zu nutzen, können Anwenderprogramme Systemaufrufe verwenden. Mit diesen kann sich ein Programm bspw. an einen Port binden oder eine Logdatei öffnen, da solche Operationen System-Rechte erfordern. Damit ein Programm auf derartige Systemaufrufe zugreifen darf, muß es mit Root-Rechten gestartet werden – und eröffnet damit eine riesige Sicherheitslücke. Systrace von Niels Provos umgeht dieses Problem, indem die Rechtezuteilung nicht mehr auf Programmebene vorgenommen, sondern auf Ebene der Systemaufrufe heruntergebrochen wird.

Ein Programm kann also von einem normalen Benutzer gestartet werden und bekommt gemäß einer vorher erstellten Richtlinie von Systrace entsprechende Zugriffsrechte auf Systemaufrufe zugeteilt.

Die Grammatik der Richtlinie

Die Richtlinie für ein zu überwachendes Programm wird in einer recht einfachen Grammatik definiert, in der Ausdrücke aneinandergereiht werden. Ein Ausdruck besteht dabei aus einem booleschen Ausdruck und der auszuführenden Aktion. Für die Aktion gibt es folgende Kommandos: *ask* (frage), *deny* (verbiete) oder *permit* (erlaube) in Verbindung mit optionalen Argumenten. Ergibt der boolesche Ausdruck `wahr`, wird die definierte Aktion ausgeführt. Ist die Aktion als *ask* definiert, wird der Benutzer befragt, um die Aktion zu erlauben oder zu verbieten.

Der boolesche Ausdruck setzt sich aus verschiedenen Variablen und den Logik-Operatoren *and* (logisches Und), *or* (logisches Oder) und *not* (logisches Nicht) zusammen. Die Variablen bestehen aus den normalisierten Systemaufruf-Namen, den dem Systemaufruf übergebenen Argumenten und einem Logik-Operator, der beide Argumente verknüpft.

Die Filterausdrücke auf Argumente verwenden verschiedene Operatoren:

- `match` Wahr, wenn der Dateiname den Regeln in fnmatch(3) entspricht.
- `eq` Wahr, wenn das Argument des Systemaufrufes genau der Vorgabe entspricht.
- `neq` Die logische Negation des eq-Operators.
- `sub` Wahr, wenn die angegebene Teilzeichenkette im Argument des Systemaufrufes vorkommt.
- `nsub` Die logische Negation des sub-Operators.
- `inpath` Wahr, wenn das Argument des Systemaufrufes im Pfad der Vorgabe vorkommt.
- `re` Wahr, wenn das Argument des Systemaufrufes dem angegebenen Regulären Ausdruck entspricht.

Ein paar Ausdrücke zur Veranschaulichung der Möglichkeiten:

- netbsd-execve: `permit` – Erlaubt alle execve(2)-Aufrufe.
- netbsd-execve: `true then permit log` – Erlaubt alle execve(2)-Aufrufe und protokolliert sie mit `syslog`. Das »`true then`« ist nötig, damit »log« eingesetzt werden kann.
- netbsd-fsread: `filename eq ''/etc/passwd'' then permit` – Erlaubt alle Lese-Operationen auf */etc/passwd*.
- netbsd-fsread: `filename match ''/etc*'' then deny [eaccess] log` – Verbietet alle Lese-Operationen auf `/etc*` mit dem Fehlercode EACCESS und protokolliert sie mit `syslog`.
- netbsd-seteuid: `uid eq ''1007'' or uname eq ''systraced'' then permit` – Erlaubt das Setzen der effektiven UID, wenn die UID des aufrufenden Benutzer 1007 oder er der Benutzer »`systraced`« ist.
- netbsd-connect: `sockaddr re ''inet-.192\.168\.[0,4,8]\.[4-6].:22'' then permit log` – Erlaubt eine Socket-Verbindung, wenn die Zieladresse des Sockets im angegebenen Adressbereich liegt. Auch dieser Aufruf wird protokolliert.

Um über einen Systemaufruf zu entscheiden, traversiert Systrace alle Ausdrücke und bricht beim ersten Ausdruck ab, der zum Systemaufruf paßt. Dieser Ausdruck entscheidet dann, ob der Systemaufruf ausgeführt oder abgelehnt wird. Wird kein passender Ausdruck gefunden, wird die Entscheidung an den Benutzer delegiert. Wird ein Systemaufruf abgelehnt, kann Systrace an das aufrufende Programm einen spezifizierten Fehlercode zurückgeben.

Um die Richtlinie auf Benutzer- bzw. Gruppenebene granulieren zu können, werden Ausdrükke mit einem Prädikat versehen. Dieses Prädikat genügt der Form ”, if” {”user”, ”group”} {”=”, ”!=”, ”<”, ”>” } {Bernutzername, numerische UID}. Somit lassen sich Ausdrücke der Art netbsd-fsread: `filename eq ''/etc/master.passwd'' then deny[eperm], if group != wheel` erzeugen. Hier wird der Zugriff auf die Datei `/etc/master.passwd` mit dem Fehlercode EPERM abgelehnt, wenn der aufrufende Benutzer nicht Mitglied der Gruppe »wheel« ist.

Soll ein Systemaufruf unter anderen Benutzerrechten ausgeführt werden, kann an den Ausdruck die Direktive as user:group angehängt werden. Beispiele dazu werden später im Text aufgeführt.

An jeden Ausdruck einer Richtlinie kann die Log-Direktive log angehängt werden. Damit wird der Ausdruck und die durch ihn implizierte Entscheidung vom Betriebssystem geloggt. Mit dieser Option lassen sich Programme komplett überwachen und analysieren. Protokolliert man beispielsweise alle exec(3)-, execve(2)- und connect(2)-Aufrufe, erfährt man, welche Programme ein Benutzer ausgeführt und welche Sockets er geöffnet hat. Beachten Sie hierbei aber unbedingt datenschutzrechtliche Bestimmungen und andere Regelungen.

Erzeugung einer Richtlinie

Ziel der Richtlinie ist es, alle erlaubten Systemaufrufe einer Anwendung zu erfassen und zu erlauben. Nicht erfaßte Aufrufe sind als Angriff zu werten und zu verbieten. Somit läßt sich eine Richtlinie erstellen, indem ein spezielles Programm die zu erfassende Anwendung bei einem Probelauf überwacht und die abgesetzten Systemaufrufe mitschneidet. Diese werden dabei in die kanonische Form normalisiert und in Richtlinien-Ausdrücke übersetzt. Existiert in der bisherigen Richtlinie kein Ausdruck, der den aktuellen Systemaufruf behandelt, wird ein neuer Ausdruck angehängt, der den Aufruf erlaubt.

Manuelles Eingreifen in die erzeugte Richtlinie ist in der Regel nicht erforderlich, es sei denn, die überwachte Anwendung arbeitet mit Zufallsnamen für Dateien. Dann muß der entsprechende Ausdruck dahingehend abgeändert werden. Bei dieser automatisierten Richtlinienerstellung wird davon ausgegangen, daß das zu überwachende Programm per se sicher ist. Kann dies nicht gewährleistet werden, ist eine so erstellte Richtlinie nicht als sicher zu betrachten. Außerdem ist die Automatik eben-

falls nicht anwendbar, wenn es nicht möglich ist, alle auftretenden Code-Pfade durchzuexerzieren. Trotzdem dient eine so erstellte Richtlinie als Basis für eine händische Anpassung, oder als Basis für weitere Übungsläufe.

Bei diesen wird die Richtlinie nur dann erweitert, wenn neue Systemaufrufe erfolgen. Hier kann der Benutzer wieder die resultierende Aktion festlegen.

Nachdem eine Richtlinie fertiggestellt wurde, kann sie von Systrace gegenüber dem gewünschten Programm durchgesetzt werden.

Implementierung

Systrace kann in einem von drei verschiedenen Modi laufen:

- Initialisierungsmodus: Systrace überwacht ein Programm automatisch und generiert eine Richtlinie. Diese ist ein guter Startpunkt, um eine angepaßte und verfeinerte Richtlinie zu erzeugen.
- Nachfragemodus: Hier wird ebenfalls ein Programm überwacht und eine Richtlinie erzeugt, allerdings wird der Benutzer bei jedem Systemaufruf um Zustimmung gebeten. Dies ist sinnvoll, wenn dem zu überwachenden Programm nicht unbedingt von vornherein vertraut werden kann. Die Nachfrage erfolgt über das X-Programm `xsystrace(1)` oder im Textmodus ohne X.
- Überwachungsmodus: Systrace überwacht ein Programm und setzt die definierte Richtlinie durch. Nicht erlaubte Systemaufrufe werden abgelehnt und protokolliert.

Systrace verfügt über eine Reihe von Optionen:

- `-A` Initialisierungsmodus, erzeugt eine Richtlinie, in der alle Systemaufrufe erlaubt sind.
- `-a` Überwachungsmodus, setzt die definierte Richtlinie durch.
- `-c UID:GID` Spezifiziert eine numerische User- und Gruppen-ID. Unter diesen wird das zu überwachende Programm ausgeführt. Nur als Root machbar.
- `-d verzeichnis` Setzt ein Verzeichnis für die Richtliniendateien. Standard ist /.systrace.
- `-f datei` Systrace verwendet die Richtlinie, die in der Datei angegeben ist.
- `-g gui` Aktiviert eine alternative GUI.
- `-i` Vererbt die Richtlinie des Elternprozesses an die Kinder.
- `-p pid` Systrace bindet sich an den bereits laufenden Prozess mit der angegebenen PID. Der komplette Programmpfad muß ebenfalls angegeben werden.
- `-t` Textmodus, läuft auch ohne X.
- `-U` Benutzt nur globale Richtlinien (/etc/systrace) statt lokaler.
- `-u` Deaktiviert das Zusammenfassen von Systemaufrufen zu Aliasen.

Sicherheit des Systems

Ein System wie Systrace selbst hat auch mit einigen Sicherheitsproblemen zu kämpfen, z.B. Aliasen auf Systemressourcen, Dateinamen von Programmen, die in einem Chroot laufen oder der Verfolgung von Prozess-IDs.

Der Zugriff auf ein und dieselbe Datei ist unter Unix dank symbolischer Links und relativer Pfadnamen auf unendlich viele Arten möglich. Außerdem können Dateien von verschiedenen Diensten, wie z.B. Proxies, NFS oder CFS, bereitgestellt werden. Derartige Dienste sind für Systrace nicht sichtbar, müssen aber trotzdem korrekt funktionieren.

Weiterhin ist es möglich, eine Race-Condition zu produzieren, die eine Systrace-Sandkiste aushebelt. Systrace benötigt für die Überprüfung eines Systemaufrufes eine gewisse Zeitspanne. Während dieser Zeitspanne kann ein anderer Prozess den eigentlichen Systemaufruf des zu überwachenden Programmes ändern. Systrace erkennt diese Änderung nicht und gestattet die Ausführung des inzwischen geänderten Systemaufrufes.

Um diesen Problemen zu begegnen, verwendet Systrace nur normalisierte Dateinamen und Systemaufrufe. Alle Dateinamen und Parame-

```
1 # systrace -A apachectl start
2 /usr/pkg/sbin/apachectl start: httpd started
3 # apachectl stop
4 /usr/pkg/sbin/apachectl stop: httpd stopped
5 # ls /root/.systrace/
6 usr_pkg_sbin_apachectl usr_pkg_sbin_httpd
```

Abbildung 2: Automatisch eine Richtline für Apache erstellen

ter für Systemaufrufe werden von Systrace normalisiert, indem Dateinamen in absolute Form – also ohne Symlinks oder relative Pfadangaben – umgewandelt werden. Diese normalisierten Werte werden dem Betriebssystem wieder übergeben.

Ausgenommen hiervon sind nur einige Systemaufrufe, wie z.B. `readlink`. Weiterhin werden diese Werte auf einem nicht beschreibbaren Puffer zwischengespeichert, so daß kein anderer Prozeß die Werte manipulieren kann. Der Kernel verweigert die Ausführung von Systemaufrufen, die Symlinks als Argumente enthalten. Somit werden nur noch von Systrace normalisierte Aufrufe, die erlaubt sind, ausgeführt. Alle anderen Aufrufe werden abgelehnt.

Werden Systemaufrufe abgelehnt, müssen die überwachten Programme entsprechende Fehlermeldungen bekommen. Da nicht alle Programme eine funktionierende Fehlerbehandlung implementieren, kann in der Richtlinie ein bestimmter Fehlercode spezifiziert werden.

Ebenfalls zu betrachten ist ein Richtlinien-Wechsel und Prozeß- Beendigung. Wenn ein überwachter Prozeß einen neuen Prozeß startet, wird der alte Prozeß vom System aus dem Speicher entfernt. Der neue Prozeß wird stattdessen ausgeführt. Dieser neue Prozeß kann ein vertrauenswürdiges Programm sein, so daß keine weitere Überwachung notwendig ist. Es kann aber auch ein Prozeß sein, der mit einer anderen Richtlinie besser überwacht wird. Systrace überwacht Systemaufrufe auf Erfolg und kann so nach einem geglückten execve-Systemaufruf den neuen Prozeß mit einer anderen Richtlinie überwachen oder die Überwachung beenden.

Die gesamten Richtlinien werden in einzelnen Dateien gespeichert. Kann ein Einbrecher die Richtlinien-Dateien manipulieren, kann er Systrace aushebeln. Daher sind die Richtlinien-Dateien unbedingt zu schützen. Dazu kann man neben restriktiven Schreibrechten die NetBSD-Fileflags (`schg`) in Verbindung mit den Security-Leveln verwenden. Möchte man diese Methode nicht einsetzen, sollten die

```
1 [...]
2 netbsd-pread: permit
3 netbsd-fsread: filename eq „/etc/group“ then permit
4 netbsd-fsread: filename eq „/usr/pkg/etc/httpd/httpd.conf“ then permit
5 netbsd-fsread: filename eq „/usr/pkg“ then permit
6 netbsd-fsread: filename eq „/usr/pkg/etc/httpd/srm.conf“ then permit
7 netbsd-fsread: filename eq „/usr/pkg/etc/httpd/access.conf“ then permit
8 netbsd-gettimeofday: permit
9 netbsd-fsread: filename eq „/etc/etc.network/resolv.conf“ then permit
10 netbsd-fsread: filename eq „/etc/hosts“ then permit
11 netbsd-chmod: filename eq“/var/run/httpd.mm.2872.sem“
12 and mode eq „600“ then permit
13 netbsd-chown: filename eq“/var/run/httpd.mm.2872.sem“
14 and uid eq „1002“ and gid eq „-1“ then permit
15 netbsd-fswrite: filename eq „/var/log/httpd/error_log“ then permit
16 netbsd-dup2: permit
17 netbsd-select: permit
18 netbsd-fsread: filename eq „/usr/pkg/etc/httpd/mime.types“ then permit
19 netbsd-fsread: filename eq „/usr/pkg/etc/httpd/magic“ then permit
20 netbsd-fswrite: filename eq „/var/log/httpd/access_log“ then permit
21 netbsd-chdir: filename eq „/“ then permit
22 netbsd-fork: permit
23 netbsd-exit: permit
24 netbsd-setsid: permit
25 netbsd-fsread: filename eq „/dev/null“ then permit
26 netbsd-fswrite: filename eq „/dev/null“ then permit
27 netbsd-socket: sockdom eq „AF_INET“ and socktype eq „SOCK_STREAM“ then
permit
28 netbsd-setsockopt: permit
29 netbsd-bind: sockaddr eq „inet-[0.0.0.0]:80“ then permit
30 netbsd-listen: permit
31 netbsd-bind: sockaddr eq „inet-[192.168.0.5]:80“ then permit
32 netbsd-bind: sockaddr eq „inet-[127.0.0.1]:80“ then permit
33 netbsd-fsread: filename eq „/var/run/httpd.pid“ then permit
34 netbsd-umask: permit
35 netbsd-fswrite: filename eq „/var/run/httpd.pid“ then permit
36 netbsd-write: permit
37 netbsd-fswrite: filename eq“/var/run/httpd.lock.1827“ then permit
38 [...]
```

Abbildung 3: Automatisch erzeugte Richtlinie für Apache (Auszug)

Richtlinien zumindest regelmäßig mit einem Integritätsprüfer wie mtree(8), Aide oder Tripwire überprüft werden.

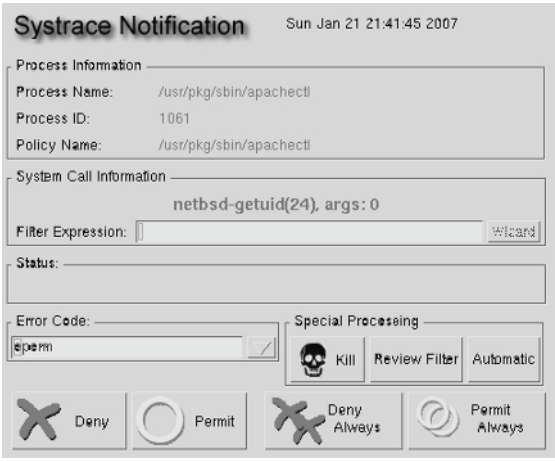
```
1 [...]
2 netbsd-chmod: filename match“/var/run/httpd.mm.*.sem“
3 and mode eq „600“ then permit
4 netbsd-chown: filename match“/var/run/httpd.mm.*.sem“
5 and uid eq „1002“ and gid eq „-1“ then permit
6 netbsd-fswrite: filename match“/var/log/httpd/error_log“ then permit
7 [...]
8 netbsd-fswrite: filename match“/var/run/httpd.lock.*“ then permit
9 [...]
```

Abbildung 4: Automatisch erzeugte Richtlinie korrigiert für Apache (Auszug)

Apache überwachen

Mit den Befehlen aus Abbildung 2 wird Apache gestartet. Dabei wird er von Systrace überwacht, so daß eine Richtlinie erstellt wird. Diese Richtlinie wird geschrieben, nachdem Apache wieder beendet wurde. Ein Teil der Richtlinie wird in Abbildung 3 gezeigt. In den Zeilen 11, 13 und 37 wird auf eine Datei im Verzeichnis `/var/run` zugegriffen. Wie man leicht erkennt, enthält der Dateiname die Prozeß-ID bzw. Semaphoren-Nummern.

Diese müssen durch den Joker »*« ersetzt werden. Außerdem muß der Operator »`eq`« durch »`match`« ersetzt werden, wie in Abbildung 4 gezeigt wird. In den Zeilen 31 und 32 bindet sich Apache an die angegebenen IP-Adressen und Port 80. In den restlichen sowie ausgelassenen Zeilen liest Apache diverse Konfigurationsdateien ein. Startet man nun Apache unter Systrace-Überwachung, kann kein Benutzer auf die Webseiten zugreifen, da die Richtlinie keinen Lese-Zugriff für `/home/www` beinhaltet. Die Verstöße werden in `/var/log/messages` protokolliert. Um dieses Problem möglichst komfortabel zu beheben, wird Apache wieder im Initialisierungsmodus von Systrace gestartet: systrace apachectl start. Wird nun in einem Browser die Adresse `http://127.0.0.1/` aufgerufen, meldet sich xsystrace (Ein Beispiel findet sich in Abbildung 1) zu Wort und verlangt vom Benutzer die Bestätigung oder Verweigerung der benötigten Systemaufrufe.



Da diese Prozedur für jede einzelne von Apache geladene Datei durchlaufen wird, brechen wir nach dem Laden der Index-Seite ab und beenden den Apache mit apachectl stop. Systrace hat die Richtlinie bereits um die Lesezugriffe auf `/home/www` erweitert. Allerdings für jede Datei einzeln, so daß hier wieder Reguläre Ausdrücke mit »match« und »*« eingesetzt werden. Abbildung 5 zeigt die automatisch generierte Richtlinie, die jede aufgerufene Datei einzeln behandelt. In Abbildung 6 wurde die Richtlinie mit regulären Ausdrücken verfeinert, so daß Zugriffe auf `/home/www/public/*` gestattet und auf `/home/www/institutsintern/*` verboten werden. Weiterhin wurden die Zugriffe auf CGI-Dateien erlaubt bzw. verboten. Zugriffe auf die verbotenen Dateien werden mit ENOENT abgelehnt, so daß für Apache die Dateien nicht existieren. Bis jetzt wurde die Richtlinie soweit konfiguriert, daß Apache unter Überwachung normal funktioniert.

Trotzdem muß Apache noch als `root` gestartet werden. Systrace kann mit der Option `-c` eine beliebige numerische Benutzer- und Gruppen-ID übernehmen, als die der zu überwachende Prozeß ausgeführt werden soll. Dazu muß allerdings noch die

Richtlinie angepaßt werden, da ein nicht privilegierter Benutzer keine privilegierten Operationen ausführen darf. Systrace kann jede Aktion in der Richtlinie als ein anderer Benutzer ausführen. Somit kann eine Anwendung als normaler Benutzer ausgeführt werden, und nur die benötigten Systemaufrufe werden als root ausgeführt. Dann müssen Systrace und das zu überwachende Programm aber als root gestartet werden. Startet man Apache nun mit systrace -c 1002:1001 apachectl start als Benutzer und Gruppe »www« mit der bisherigen Richtlinie, wird sich Systrace erneut zu Wort melden. Es werden alle Systemaufrufe angezeigt, die als Benutzer »www« nicht ausgeführt werden können. Am einfachsten editiert man daher vorher die Richtlinie mit vi(1) und sucht zuerst nach Aktionen, in denen Dateien mit »netbsd-fswrite« geschrieben werden. Das umfaßt in diesem Beispiel die Log-, PID- und Semaphoren-Dateien.

```
1 [...]
2 netbsd-fsread: filename eq „/home/www“ then permit
3 netbsd-fsread: filename eq „/home/www/.htaccess“ then permit
4 netbsd-fsread: filename eq „/home/www/index.html“ then permit
5 netbsd-fsread: filename eq „/home/www/index.pl“ then permit
6 netbsd-fsread: filename eq „/home/www/index.mhtml“ then permit
7 [...]
```

Abbildung 5: Automatisch generierte Richtlinie für Apache, die jede Datei einzeln aufführt.

Die nächsten offensichtlichen Kandidaten sind alle »netbsd-bind«-Aktionen, in denen sich Apache an die Netzwerkgeräte bindet. Die angepaßten Aktionen der Richtlinie für nicht-privilegierte Läufe finden Sie in Abbildung 7. Mit dieser Richtlinie kann Apache via Systrace als Benutzer »www« gestartet werden.

Systrace-überwachte Shell

Matthias Petermann beschreibt in (Peterman, 2005) die Einrichtung einer Shell, die von Systrace komplett überwacht wird.

Um eine Shell von Systrace überwachen zu lassen, müßte die Shell über Systrace in der Art systrace -a -i ksh gestartet werden. Da solch ein Befehl nicht in der /etc/master.passwd eingetragen werden kann, muß er in einem kleinen C-Programm (Abbildung 8) gekapselt werden. Diese Kapsel wird nach Erstellung der Richtlinie in /etc/shells und /etc/master.passwd für die jeweils zu überwachenden Benutzer als Shell eingetragen. In diesem Beispiel soll der Benutzer systraced eine eingeschränkte Korn-Shell erhalten.

Um die Richtlinie automatisch zu erstellen, loggen wir uns mit login systraced ein und starten eine Shell mit systrace -A /bin/ksh. Nach Beendigung der Shell haben wir wieder das Grundgerüst unserer Richtlinie in systraced/.systrace/bin_ksh vorliegen. Nun wird die Shell-Kapsel aus Abbildung 8 als Shell für den Benutzer aktiviert. Zudem wird sich wieder mit login(1) als systraced eingeloggt. Da nun die Systrace-Überwachung aktiv ist, werden Systemaufrufe überwacht und Fehler via syslog(3) protokolliert. Es ist in der Anfangsphase recht praktisch, in einem weiteren XTerminal tail -f /var/log/messages laufen zu lassen, um sofort die fehlgeschlagenen Systemaufrufe analysieren zu können. Die Richtlinien unter systraced/.systrace/ sind mit chown(8) root und wheel zuzuordnen und mit chmod(1) auf 644 oder gar 444 zu setzen. Selbiges gilt für das .systrace-Verzeichnis, allerdings mit den Rechten 755.

In der Beispiel-Richtlinie aus Abbildung 9 befinden sich vier Blöcke von Ausdrücken. Im ersten Block wird der Zugriff auf verschiedene Geräte und Konfigurationsdateien geregelt. Außerdem wird der Zugriff auf /tmp/, /var/tmp/ und das Heimatverzeichnis des Benutzers geregelt.

```
1 [...]
2 netbsd-fsread: filename match „/home/www/public/*“ then permit
3 netbsd-fsread: filename match „/home/www/institutsintern/*“
4 then deny [enoent]
5 netbsd-fsread: filename eq „/home/www/cgi-bin/cvsweb.cgi“ then permit
6 netbsd-fsread: filename eq „/home/www/cgi-bin/postgresql.cgi“
7 then deny [enoent]
8 [...]
```

Abbildung 6: Angepasste Apache-Richtline

```
1 # grep 'as root' usr_pkg_sbin_httpd
2 netbsd-fswrite: filename match „/var/run/httpd.mm.*.sem“ then permit as root
3 netbsd-fswrite: filename match „/var/run/httpd.mm.*.sem“ then permit as root
4 netbsd-fswrite: filename match „/var/run/httpd.mm.*.sem“ then permit as root
5 netbsd-chmod: filename match „/var/run/httpd.mm.*.sem“ and mode eq „600“
6 then permit as root
7 netbsd-chown: filename match „/var/run/httpd.mm.*.sem“ and uid eq „1002“
8 and gid eq „-1“ then permit as root
9 netbsd-fswrite: filename eq „/var/log/httpd/error_log“ then permit as root
10 netbsd-fswrite: filename eq „/var/log/httpd/access_log“ then permit as root
11 netbsd-bind: sockaddr eq „inet-[0.0.0.0]:80“ then permit as root
12 netbsd-bind: sockaddr eq „inet-[192.168.0.5]:80“ then permit as root
13 netbsd-bind: sockaddr eq „inet-[127.0.0.1]:80“ then permit as root
14 netbsd-fsread: filename match „/var/run/httpd.pid“ then permit as root
15 netbsd-fswrite: filename eq „/var/run/httpd.pid“ then permit as root
16 netbsd-fswrite: filename match „/var/run/httpd.lock.*“ then permit as root
```

Abbildung 7: Angepasste Apache-Richtline

Die Ausdrücke im zweiten Block regeln die Erstellung von Sockets. Dabei werden mit den regulären Ausdrücken die Zugriffe auf verschiedene IP-Adressen bzw. Adress-Bereiche unter Port 22 erlaubt. Alle derartigen Ausdrükke werden protokolliert.

Im dritten Block werden die `exec(3)`- und `execve(2)`-Aufrufe behandelt. Aufrufe von `/usr/bin/ftp` und `/usr/bin/telnet` sowie von Programmen, die in `/home/systraced/` liegen, werden verboten, alle anderen erlaubt. Der Benutzer kann somit auch keine Programme in seinem Benutzerverzeichnis ablegen und von dort aus starten. Er kann allerdings in dieser Richtlinie noch Programme in `/tmp/` oder `/var/tmp/` ausführen.

Im letzten Block finden sich alle Systemaufrufe, die ohne Argumente aufgeführt werden und von Systrace während des ersten Initialisierungslaufes geschrieben wurden.

Weiterer Nutzen

Systrace eignet sich nicht nur zur Absicherung eines Systems, sondern auch zur einfachen Überwachung und Analyse von Programmen. Indem man automatisch eine Richtlinie erstellen läßt, erfährt man, welche Systemaufrufe vom Programm durchgeführt werden. Diese Richtlinie kann man zu Testzwecken manipulieren und so überprüfen, wie sich ein Programm verhält, wenn es beispielsweise nicht mehr auf bestimmte Dateien zugreifen darf. Diese Vorgehensweise ist sehr nützlich bei großen Programmen, die nur als Binärversion verfügbar sind, wie Opera oder Acrobat Reader. Ebenso kann man damit selbst entwickelte Programme auf Fehlerbehandlungen hin überprüfen.

Fazit

Systrace ist ein umfangreiches Programm, das bei korrekter Konfiguration die Sicherheit eines Systems dramatisch erhöhen kann. Mit Systrace lassen sich Dienste als nicht-privilegierter Benutzer ausführen. Nur bestimmte Systemaufrufe müssen Root-Rechte erhalten. Damit reduziert sich das potentielle Schadensrisiko, das von einem SETUID-Programm ausgeht. Weiterhin ermöglicht Systrace eine feinere Granularisierung von Programmaufrufen oder Sockets. Man kann beispielsweise `ssh` nur auf bestimmte IP-Adressen erlauben und auf alle anderen verbieten.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4
5 int main()
6 {
7 puts(„Führe Systrace-überwachte Shell aus“);
8 execlp(„/bin/systrace“,“systrace“,“-a“,“-i“,“/bin/ksh“,NULL);
9 return 0;
10 }
```

Abbildung 8: Kapselung des Shell-Aufrufes

Die Konfiguration einer Systrace-Richtlinie setzt allerdings gute Kenntnisse des Betriebssystems – insbesondere der Systemaufrufe – und der Anwendung voraus. Außerdem ist Systrace eine technische Maßnahme, die ohne umfassende Sicherheitsrichtlinie allein zu kurz greift.

```
1 Policy: /bin/ksh, Emulation: netbsd
2 ## Zugriffe auf Konfigurationsdateien erlauben
3 netbsd-fsread: filename eq „/etc/man.conf“ then permit
4 netbsd-fsread: filename eq „/etc/passwd“ then permit
5 netbsd-fsread: filename match „/etc*“ then deny
6 netbsd-fsread: filename match „/home/systraced/*“ then permit
7 netbsd-fsread: filename match „/home*“ then deny
8 netbsd-fsread: filename match „/tmp/*“ then permit
9 netbsd-fsread: filename match „/var/tmp/*“ then permit
10 netbsd-fswrite: filename match „/dev/tty“ then permit
11 netbsd-fswrite: filename match „/tmp/*“ then permit
12 netbsd-fswrite: filename match „/var/tmp/*“ then permit
13 netbsd-fswrite: filename eq „/dev/crypto“ then permit
14
15 ## SSH auf bestimmte Adressen erlauben
16 netbsd-socket: sockdom eq „AF_INET“ and socktype match „*“ then permit
17 netbsd-connect: sockaddr eq „inet-[127.0.0.1]:22“ then permit log
18 netbsd-connect: sockaddr re „inet-.192˙168˙[0,4,8]˙[4-6].:22“ then permit log
19 netbsd-connect: sockaddr re „inet-.192˙168˙0˙5˙:22“ then permit log
20 netbsd-setuid: uid eq „1007“ and uname eq „systraced“ then permit
21 netbsd-seteuid: uid eq „1007“ and uname eq „systraced“ then permit
22 netbsd-recvfrom: permit
23 netbsd-setsockopt: permit
24
25 ## Ausführbare Dateien in $HOME und ftp + telnet verbieten, sonst erlauben
26 netbsd-exec: filename sub „/home/systraced“ then deny log
27 netbsd-execve: filename sub „/home/systraced“ then deny log
28 netbsd-execve: filename eq „/usr/bin/ftp“ then permit log
29 netbsd-execve: filename eq „/usr/bin/telnet“ then permit log
30 netbsd-exec: true then permit log
31 netbsd-execve: true then permit log
32
33 netbsd-mmap: permit
34 netbsd-fsread: permit
35 netbsd-__fstat13: permit
36 netbsd-close: permit
37
38 [...]
```

Abbildung 9: Kapselung des Shell-Aufrufes

Siehe auch:

Ioannidis, Bellovin und Smith, 2006
Goldberg, Wagner, Thomas und Brewer, 1996

Literatur

[Eriksen und Provos 2003] Eriksen, Marius A. ; Provos, Niels: Enges Korsett: Systrace setzt Regeln für erlaubte Systemaufrufe durch. In: Linux Magazin 2003 (2003), 1. Ausgabe.

[Goldberg u. a. 1996] Goldberg, Ian ; Wagner, David ; Thomas, Randi ; Brewer, Eric A.: A Secure Environment for Untrusted Helper Applications. In: Proceedings of the 6th Usenix Security Symposium (1996)

[Ioannidis u. a. 2006] Ioannidis, Sotiris ; Bellovin, Steven M. ; Smith, Jonathan M.: Sub-Operating Systems: A New Approach to Application Security. In: Proceedings of the SIGOPS European Workshop, SIGOPS@, 2006

[Peterman 2005] Peterman, Matthias: Systrace-Restricted Login-Shell mit NetBSD. (2005). – URL *http://wiki.bsd-crew.de/index.php/Systrace-Restricted_Login-Shell_mit_NetBSD*. – Zugriffsdatum: Jan. 2007

[Provos 2006] Provos, Niels: Improving Host Security with System Call Policies. (2006). – URL *http://www.citi.umich.edu/articles/reports/citi-tr-02-3.pdf*. – Zugriffsdatum: Nov. 2006

[Thompson 1984] Thompson, Ken: Reflections on Trusting Trust. In: Communication of the ACM, Association for Computing Machinery, Inc, 1984, S. 761–763. – URL *http://www.ece.cmu.edu/~ganger/712.fall02/papers/p761-thompson.pdf*. – Zugriffsdatum: Dez. 2006


1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D
3C COMMUNICATIONS
Pages Blanches
Gids
Pages d'Or
Gouden Gids
DOCKX
RENTAL
1207
1207


"In Fairy Dust We Trust!"
8 | 9 | 10 | 11 | 12. August 2007
Finowfurt, near Berlin
http://events.ccc.de/camp/2007
Chaos Computer Club presents:
Chaos Communication Camp 2007
The International Open Air Hacker Meeting