die datenschleuder.

das wissenschaftliche fachblatt für datenreisende
ein organ des chaos computer club

Quelle: Titelbild "radikal", Sozialistische Zeitung für Westberlin, Ausgabe 1 vom 18.6.1976.

ISSN 0930-1054 • 2005
Der Preis beträgt diesmal 2,50 EURo.
Postvertriebsstück C11301F

#86

Erfa-Kreise / Chaostreffs

Bielefeld im AJZ, Heeper Str. 132, mittwochs ab 20 Uhr *http://bielefeld.ccc.de/ :: info@bielefeld.ccc.de*
Berlin, CCCB e.V. (Club Discordia) Marienstr. 11, (Briefe: CCCB, Postfach 64 02 36, D-10048 Berlin),
donnerstags ab 17 Uhr *http://berlin.ccc.de/ :: mail@berlin.ccc.de*
Düsseldorf, CCCD/Chaosdorf e.V. Fürstenwall 232,
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, Leitstelle511 Kulturcafé, Schaufelder Str. 30, Hannover
2. Mittwoch 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. Chaoslabor, Vogelsanger Str. 286
Letzter Donnerstag im Monat ab 19:30 Uhr *http://koeln.ccc.de/ :: mail@koeln.ccc.de*
München, muCCC e.V. Kellerräume in der Blutenburgstr. 17
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/*

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, Bad Waldsee, Basel, Bochum, Brugg, Darmstadt, Dortmund, Dresden, Frankfurt am Main, Freiburg im Breisgau, Gießen/Marburg, Hanau, Heidelberg, Ilmenau, Kiel, Mainz, Mülheim an der Ruhr, Münster/Osnabrück, Offenbach am Main, Paderborn, Regensburg, Stuttgart, Trier, Weimar, Wetzlar, Wuppertal, Würzburg.

Friends & Family

Zur näheren Chaosfamilie zählen wir (und sie sich) die Häcksen (*http://www.haecksen.org/*), den/der "Verein zur Förderung des öffentlichen bewegten und unbewegten Datenverkehrs e.V." - FoeBuD (*http://www.foebud.de/*), den Netzladen e.V. in Bonn (*http://www.netzladen.org/*) und die c-base Berlin (*http://www.c-base.org/*).

Die Datenschleuder Nr. 86

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 und Produktion
Tom Lazar, *<tom@tomster.org>*

Layout
John-Paul Bader (hukl), Dirk Engling, Tom Lazar

Chefredaktion
Dirk Engling <erdgeist> und Tom Lazar <tomster>

Redaktion dieser Ausgabe
AmP, Markus Beckedahl, Alexander Bernauer, Michael Christen, Simone Demmel, Zapf Dingbatz, Ben Fuhrmannek, Peter Glaser, Rop Gonggrijp, Constanze Kurz, Angelo Laub, Frank Rosengart, Markus Schaber, starbug, Rüdiger Weis, Barry Wels, Ansgar Wiechers.

Die Welt ist voll geworden mit Dingen, die man nicht direkt sehen kann, die aber einen entscheidenden Einfluß auf unser Leben nehmen.

Früher mußte man sich allein vor (biologischen) Viren in Acht nehmen, alle anderen Risiken kamen in Form von Wildschweinen, Raubrittern und Feuersbrünsten recht sichtbar daher.

Heute geht es nun nicht mehr um die physikalische Unversehrtheit. Die haben die mitteleuropäischen Nationalstaaten durch Gesetze, das Ordnungsamt, Revierförstereien und den TÜV halbwegs unter Kontrolle bekommen. Einzig die Viren haben sich aus grauer Vorzeit bis dato herübergerettet.

Heute geht es eher um die Unversehrtheit des Lebenslaufs, der Personalakte, des Führungszeugnisses, der Datenbankeinträge bei Schufa, ACNielsen und in Flensburg. Was einem dort nicht alles in die Quere kommen kann...

Einmal unbedacht Fettflecken am Glas gelassen, durch Zufall in den falschen Kiez gezogen, ein WLAN eingerichtet. Überall lauert die Falle des Unsichtbaren, das Spezialisten sichtbar machen und manipulieren können. Und allem voran drohen die von ihrer eigenen Hetze getriebenen Nationalstaaten, all diese Fallen zum Schutze der körperlichen Unversehrtheit auszuweiden.

Die Existenz von Unsichtbarem weist man durch Probekörper (Abstraktion: Meßgeräte) nach. Schon allein der Betrieb eines Computers setzt Dinge in Gang, die wir uns nur noch mit Hilfe einiger weniger Meßgeräte, nämlich Bildschirme und Drucker, veranschaulichen können. Daß nun auch gerade das Vertrauen in Drucker, das quasi-letzte Interface zwischen digitaler und analoger Papier-Form, von den Herstellern mißbraucht wird (Seite 19ff), könnte uns mit einem Gefühl der Hilflosigkeit zurücklassen.

Es war noch ein Spaß, als die Naivität des Durchschnittsbürgers Wardriving ermöglichte: wir kannten „unsere“ Meßgeräte und konnten damit gerade den entscheidenden Tick besser umgehen, um das Gefühl der Kontrolle zu erfahren.

Aber der Wissensvorsprung ist nur noch gering und basiert auf Erfahrungen, wiederkehrenden Mustern: ein massives Schloß öffnet man einzig und allein mit Hilfe des dazugehörigen Schlüssels (siehe auch Seiten 12ff, 36f und 42ff). Mein Windows clickt nur genau die Buttons, auf die meine Maus zeigt und Firewalls schützen mich vor „Hackerangriffen“ (oder? Seite 27ff.) Wenn ich einen Webserver kompromittiere, zeigt mir mein Meß-Browser, daß sich dessen Interna meinen Wünschen gemäß angepaßt haben (über moralische Aspekte solcher Fernsoftware-Updates Seite 10).

Es wird schwieriger, sowohl mechanische, elektronische, optische, biologische und kryptographische Grundlagen zu überschauen und dabei nicht den Blick für versteckte soziale Implikationen und Bedrohungen zu verlieren.

Kreativer Umgang mit unserer Alltagstechnik kann uns aber aus den gewöhnlichsten Haushaltsmitteln die nützlichsten Meßgeräte zaubern. Ans Werk! *<erdgeist>*

Inhalt

Die üblichen Verdächtigen

Peter Glaser

Peter Glaser (*1957 in Graz, Österreich als Bleistift geboren) ist ein Schriftsteller und Journalist. 1957 als Bleistift in Graz geboren, wo die hochwertigen Schriftsteller für den Export hergestellt werden. Lebt als Schreibprogramm in Berlin. CCC-Urtier (alter Datenadel). Er war Co-Redakteur der Datenschleuder. 2002 gewann er den Ingeborg-Bachmann-Preis für seine Erzählung "Die Geschichte von Nichts".

Casablanca and beyond

Liebes Mitchaos, Liebe FreundInnen und liebe Freunde, Casablanca and beyond "Verhaften Sie die üblichen Verdächtigen" – von dem Zitat aus dem Film "Casablanca" könnte man gleich überleiten in die Gegenwart und zu den Hackern als einer artverwandten Truppe üblicher Verdächtiger. Es lohnt sich aber, zuvor noch einen Augenblick auf der Zeitkoordinate des Films zu verweilen.

Im Frühsommer 1942, als "Casablanca" gedreht wurde, war Frankreich von deutschen Truppen besetzt. Die Gerhard Fieseler Werke GmbH in Kassel übernahmen gerade den Bau der Flugbombe FI 103, der Urahnin aller Marschflugkörper. Propagandaminister Goebbels gab der FI 103 in zynischer Umkehrung der Verhältnisse den Namen "Vergeltungswaffe 1". Der V1 folgte am 3. Oktober 1942 der erste erfolgreiche Start einer Aggregat 4-Rakete von einem Prüfstand der Heeresversuchsanstalt Peenemünde. Auch die A4 wurde umbenannt, sie hieß nun "Vergeltungswaffe 2".

Auf die Verkleidung der ersten V 2 war auf Höhe der Triebwerksbrennkammer eine nackte Frau in schwarzen Seidenstrümpfen gemalt, die sich auf einer Mondsichel räkelt – eine Anspielung auf einen Film, der 13 Jahre zuvor in Deutschland eine Raketeneuphorie ausgelöst hatte: "Die Frau im Mond" von Fritz Lang.

Es lohnt sich, ehe wir uns wieder den üblichen Verdächtigen der Gegenwart zuwenden, einen Umweg über die Vergangenheit zu machen und eine kleine Geschichte der Technikbegeisterung zu versuchen. Den Deutschen ist die Beschäftigung mit der Vergangenheit als Sühne auferlegt, deshalb hauen sie besonders gern in die Zukunft ab.

Da ist die große Frage, woher diese ungewöhnlichen Antriebskräfte kommen, die Technikbegeisterung erzeugen, und vor allem, wohin sie führen.

Das 20. Jahrhundert leuchtet nur so vor jungen Menschen, deren Begeisterung an technischem Neuerungen entflammt ist. Ein paar von ihnen haben das ganze Jahrhundert in Brand gesteckt. Jede neue Technik hat eine Welle der Euphorie ausgelöst. Lewis Mumford beispielsweise, der Autor des Buchs "Der Mythos der Maschine", ein Buch, das aus dieser Welt eine bessere Welt machen würde, wenn es ein Schulbuch wäre – Lewis Mumford also erzählt von der Zeit zu Anfang des 20. Jahrhunderts:

"In meiner Jugend las ich "Modern Electrics", und die neuen Mittel der drahtlosen Kommunikation nahmen meine Jünglingsphantasie gefangen. Nachdem ich meinen ersten Radioapparat zusammengebastelt hatte, war ich hocherfreut, als ich tatsächlich Botschaften von nahe gelegenen Stationen empfing, und ich fuhr fort, mit neuen Geräten und Anschlüssen zu experimentieren, um noch lautere Botschaften von

weiter entfernten Sendestationen zu empfangen. Aber ich machte mir nie die Mühe, das Morsealphabet zu lernen oder zu verstehen, was ich da hörte."

Hier ist bereits der klassische Kontrapunkt zu erkennen: auf der einen Seite eine Kristallisation von Vernunft und menschlicher Erfindungsgabe, ein technisches Gerät, und auf der anderen Seite eine Verheißung, eine Bezauberung, eine Trance, die mit Vernunft nicht das geringste zu tun hat und die tief in die stammesgeschichtliche Herkunft des Menschen hinabreicht. Die zu tun hat mit Magie und der phantastischen Empfindung, wenn ein äußerer Gegenstand, ein Gerät, auf geheimnisvolle Weise mit einem Gefühl, mit einer Gestalt im Inneren des Menschen übereinstimmt.

Berlin in den zwanziger Jahren

Unsere kleine Geschichte der Technikbegeisterung beginnt im Berlin der zwanziger Jahre. Sie führt durch die Abgründe des Dritten Reichs bis auf den Mond und findet sich schließlich in einer ehemaligen Nato-Raketenstation bei Düsseldorf und hier in Berlin wieder in der Gegenwart ein.

Der Film "Die Frau im Mond" war der Kassenschlager der Kinosaison 1929 auf 1930, Lang hatte den Film nach einem Roman seiner Frau Thea von Harbou gedreht. Während draußen die Weltwirtschaftskrise die Existenz von Millionen Menschen bedrohte, konnte man sich im Kino von der atemberaubenden Illusion eines deutschen Raumschiffs einfangen lassen, das zum ersten Mal zum Mond fliegt.

Zwei Jahre zuvor, 1927, hatte Fritz Lang "Metropolis" gedreht. In dem Film gibt es einen Erfinder, der C. A. Rotwang heißt, ein sprechender Name und nur scheinbar ein Gegenbild zum bleichen Nerd; die roten Wangen verraten die fiebrige Hitze, die zu den untrüglichen Symptomen des Fasziniertseins gehört.

Zu den ergriffensten Besuchern der "Frau im Mond" gehörten die Mitglieder des privaten "Vereins für Raumschiffahrt" in Berlin, darunter Max Valier, der alles, was fahrbar war, mit einem Raketenantrieb versah und der von Fritz von Opel gesponsert wurde. Die Vision war: Der erste Deutsche in einer Rakete von Opel im All. Ebenfalls zu den Raketenfreunden gehörte Professor Hermann Oberth, der mit seinem Buch "Die Rakete zu den Planetenräumen" ein Pionierwerk der Raketenidee verfaßt hatte, außerdem der junge Wernher von Braun.

Es gab ein Hochgefühl in dem Raumschiffahrtsverein, als Avantgarde des technischen Fortschritts auf eine Welle der Modernität zu reiten. Die Ufa zahlte Hermann Oberth 17.500 Mark, um eine zwei Meter lange Rakete zu bauen, die anläßlich der Premiere der "Frau im Mond" zu Reklamezwecken aufsteigen sollte (was sie nicht tat). Inzwischen begann sich das Heereswaffenamt für Raketen zu interessieren. Die Reichswehr suchte nach Möglichkeiten der Wiederbewaffnung, mit denen sich die Beschränkungen der Versailler Verträge umgehen ließen. Artillerie-Aufrüstung war nicht gestattet, aber von Raketen stand nichts in den den Verträgen – als sie abgefaßt wurden, gab es die Technik noch gar nicht.

Mittelbau-Dora

Die Raketenfreunde, allen voran Wernher von Braun, ließen sich auf einen faustischen Pakt mit den Nationalsozialisten ein. Bei der Massenproduktion der V2 im Konzentrationslager Mittelbau-Dora bei Nordhausen starben bis Kriegsende über 10.000 Zwangsarbeiter an den unmenschlichen Lebens- und Arbeitsbedingungen in dem Stollensystem. Die V2 war die erste Waffe, bei deren Bau mehr Menschen ums Leben kamen als durch ihren Einsatz.

Wie später bei der bemannten Mondlandung handelt es sich bereits bei der Entwicklung der deutschen Raketenwaffen um Großtechnologien, deren Aberwitz niemanden zu stören schien.

"Die technische Faszination, herkömmliche Schranken zu durchbrechen und über noch nie dagewesene Entfernungen schießen zu können, ließ eine exakte Prüfung der voraussichtlichen Wirkung ... auf den Verlauf eines Krieges überhaupt nicht zu", schreibt Werner Eisfeld in "Mondsüchtig", einer kritischen Biografie Wernher von Brauns. Schon ein Vergleich zwischen einer Rakete und einem schweren Bomber hätte zu für die Raketentechnik äußerst unvorteilhaften Fragen geführt.

Die Mißerfolge der deutschen Militärs gegen die Engländer forcierte den Übergang vom Krieg zum Terror, wenn man auf diesen Unterschied noch Wert legt – den Übergang von der geregelten Unmenschlichkeit zur ungezügelten Unmenschlichkeit. Walter Dornberger, der Chef des deutschen Raketenwaffenprogramms und Förderer von Wernher von Braun, betonte bereits Mitte 1941, dass neben der, wie er es ausdrückte: "materiellen Wirkung" ein Raketenbeschuß "größte moralische Erfolge" erzielen würde.

Ernst Stuhlinger, später einer der deutschen Chefwissenschaftler bei der NASA, erinnert sich so: "Wir hatten nicht das Gefühl, dass wir eine Vergeltungswaffe entwickelten. ... Unser Ziel war eine leistungsstarke, steuerbare, hochpräzise Rakete."

Keine Waffe: eine Rakete. Präzise Technik. Leben und Denken in einer Schneekugel. Man sieht, was rund um einen herum vor sich geht, aber da ist diese gläserne Wand... Auch wenn er es nach dem Krieg immer bestritten hat: Wernher von Braun wußte sehr wohl, unter welchen Umständen die Arbeitssklaven im KZ Mittelbau-Dora die von ihm und seinen Ingenieuren entwickelten Raketen zusammenbauen mußten.

"Die Wissenschaft hat keine moralische Dimension", so Wernher von Braun. "Sie ist wie ein Messer. Wenn man es einem Chirurgen und einem Mörder gibt, gebraucht es jeder auf seine Weise."

"Laßt mich in Ruhe mit euren Gewissensbissen", sagte im Frühsommer 1945 der Kernforscher Enrico Fermi auf alle Einwände von Kol-

legen gegen den Bau der Atombombe: "Das ist doch so schöne Physik."

Es ist der klassische Ansatz des Technokraten, um Verantwortung zu vermeiden. Die Technik wird zum Selbstzweck ernannt.

Joseph Weizenbaum beschreibt eindrucksvoll die Verflechtungen der Militärmaschinerie und der aufkommenden Computertechnik in den sechziger Jahren. Alarmierend ist die zunehmende Diffusion von Verantwortung. Die Luftaufnahmen aus Aufklärungsflugzeugen wurden elektronisch ausgewertet, das Ergebnis der Softwareoperationen waren (und sind heute mehr denn je) Kästchen auf einer Landkarte, sogenannte "kill boxes", in denen Piloten das Recht hatten auf alles zu schießen, was sich bewegt. Wer ist verantwortlich für Zivilisten, die getötet werden? Die Maschine.

Die Verbindung von Technik und Moral nicht zu kappen sondern zu stärken und immer wieder kritisch zu überprüfen, gehört zu den Grundlagen des Hackens. Die Erkenntnis, dass technischer und moralischer Fortschritt nicht gleichzusetzen sind, haben Hacker weit eher gefördert als Männer wie Fermi oder von Braun.

Gut und Böse: Umwertung der Werte

Der CCC hat in Deutschland mit seiner Politik des Öffnen und der Offenheit einen Freiraum geschaffen, in dem neue Techniken eingehend und manchmal unkonventionell untersucht werden können, ohne dass es sofort Legalitätskonflikte gibt. Einen Freiraum, den weder der akademische Betrieb, noch ein Unternehmen oder eine Behörde bieten können.

Es ging von Anfang an um Teilnahme. Wir wollten am Netz teilnehmen und die elitäre Abgeschlossenheit der Wissenschaftlergemeinde knacken. Jeder sollte am Netz teilnehmen können. Wahrscheinlich war es auch kein Zufall, dass Ausgangspunkt des ersten international beachteten Netzhacks die Rechner im Goddard Space Flight Center der NASA waren, und dass es wieder Deutsche waren, die sich da drin herumtrieben. Der Nasa-Hack zog für den Chaos Computer Club und sein Umfeld Hausdurchsuchungen und Verhaftungen nach sich, Karl Koch hat sich das Leben geänommen. In den Jahren danach fand eine vollständige Umwertung der Werte statt. Waren Hacker erst als kriminelle Eindringlinge beschrieben worden, so stellte sich jetzt mit der explosionsartigen Ausbreitung des Internet in der Öffentlichkeit heraus, dass die Teilnahme am Netz eigentlich erste Bürgerpflicht sei und auch die Blinden, die Lahmen und die Tauben noch auf Tragbahren ins Netz geschafft werden müßten.

Hacker: Kampf um die Definitionshoheit

Hacker setzen die Tradition der Aufklärung und des kritischen Denkens fort, vor allem pragmatisch, nicht unbedingt als Theoretiker. Wenn man nach Freiräumen sucht, in denen ein gesellschaftlicher Fortschritt stattfindet: Hier ist einer. Anders als die Achtundsechziger, deren bevorzugte Strategie die Verweigerung war, haben Hacker zahlreiche konstruktive Methoden der gesellschaftlichen Teilnahme erprobt und entwickelt.

Inzwischen haben die Medien allerdings den Begriff Hacker für ihre Zwecke umgefärbt. Fürs Fernsehen und Zeitungen sind Hacker nur interessant, wenn sie für einen Thrill sorgen, eine Negativmeldung also.

Für 90 Prozent der Schadprogramme, die im Internet ihr Unwesen treiben, seien organisierte Kriminelle verantwortlich, berichtet beispielsweise die Computerwoche am 10. Dezember, das habe eine aktuelle Bestandaufnahme der Moskauer Security-Firma Kaspersky Labs ergeben. "Lediglich zehn Prozent des bösartigen Codes gehen auf das Konto von Teenagern", sagt Eugene Kaspersky, der Gründer und Leiter des Unternehmens.

Kaspersky und die Computerwoche lassen im Dunklen, wie viel gutartiger Code von Teenagern verfaßt wird.

Lieber wird über alberne Defacements so berichtet, als sei das nun das Größte seit der Erfindung

des tiefen Tellers. Hier zeigt sich wieder die fatale Gewaltenteilung der medialen Berichterstattung, die weniger mit dem Wunsch nach Wirklichkeit zu tun hat, als vielmehr mit dem Wunsch nach Unterhaltung: Auf der einen Seite ist der Journalismus zuständig für Nachrichten, und das heißt nach wie vor: für schlechte Nachrichten; und auf der anderen Seite haben wir die Werbung, die die notorisch guten Nachrichten anschleppt.

Irgendwo dazwischen klemmt die Realität. Wir dürfen nie aufhören die Frage zu stellen, ob man sie nicht vielleicht freihacken kann.

Im fröhlichen Angedenken an unseren verstorbenen Freund und Chaospionier Wau Holland möchte ich auch daran erinnern, dass sich das Hacken neben dem Umgang mit hochgradigen Komplexitäten, Millionen Zeilen Quellcode und den Wissenskaskaden der Wikipedia immer auch auf einfache Prinzipien oder Ideen zurückführen läßt. Nicht im Sinne von Albert Einstein, der mal gesagt hat: " Für jedes Problem gibt es eine einfache Lösung. Sie ist immer falsch." Sondern im Sinne von Wau, der sagte, dass schon jemand, der mit einer Kaffeemaschine heißes Wasser für eine Suppe macht, ein Hacker ist – jemand, der Technik einer unvorhergesehenen, kreativen Nutzung zuführt.

Wenn man die beiden masseführenden Drahtenden eines Stromkabels vorn und hinten in ein Würstchen steckt, fungiert das Würstchen als Widerstand und wird heiß, mit anderen Worten: das Würstchen wird zwar nicht unbedingt delikat, dafür aber extrem schnell gegrillt. Das ist Hacken. Diese Haltung hat von Anfang an unseren Forschergeist beflügelt, und jeder konnte und kann seinen Beitrag dazu leisten.

Koschere Maschinen

Dass unorthodoxe Techniknutzung auch orthodox sein kann, beweist die folgende Geschichte.

Von Freitagabend nach Sonnenuntergang bis Samstagabend, wenn die ersten drei Sterne am Himmel zu sehen sind – an Sabbat also – ist es orthodoxen Juden nicht erlaubt, zu arbeiten, zu schreiben oder Feuer beziehungsweise dessen moderne elektrische Entsprechungen anzuzünden.

Im Haushalt bedeutet das: kein Herd, kein Backofen und kein Lichtschalter, der betätigt werden darf. Die jüdische Küche ist durch das Feuer-Verbot am Sabbat beeinfluß und kennt zahlreiche Speisen, die vor Sabbat-Beginn auf kleiner Flamme aufgesetzt werden und dann ganz langsam garen. Die Beschränkungen in der Küche wußten findige Gläubige seit langem elegant zu umschiffen. Manche klebten den Kontakt in der Kühlschranktür ab, der beim Öffnen das Licht im Inneren einschalten würde (was verboten ist); andere schraubten, ehe am Freitag die Sonne unterging, das Glühbirnchen aus der Fassung.

Manche Herde waren früher mit einer Sicherheitsautomatik ausgestattet, die das Gerät nach 12 Stunden von selbst abschaltete. Ein am Freitagabend eingeschalteter Ofen war also vor dem Abendessen am Samstag wieder kalt. Einige Hersteller bekamen mit, dass dadurch für man-

che Juden die Vorbereitung des Nachtmahls kompliziert wurde. Sie ließen die Sicherheitsautomatik überarbeiten. Der "Sabbat-Modus" war geboren.

Mit dem Einzug von High-Tech in die Küche wurde es aber zunehmend schwieriger, sie mit herkömmlichen Methoden zu überlisten. Sensoren in Kühlschränken lassen sich nicht mehr so einfach ruhig stellen wie die alten mechanischen Fühler. Inzwischen lassen sich die Entwickler von Hausgeräten aber dabei beraten, wie man Maschinen Sabbat-kompatibel machen kann.

Ist an einem Ofen oder einem Kühlschrank der Sabbat-Modus aktiviert, bleibt die Innenbeleuchtung aus, die Anzeigenfelder erlöschen, Töne und Lüfter sind stillgelegt. Hightech-Geräte verwandeln sich für 24 Stunden wieder in schlichte Nachkriegstechnik. Damit kein Benutzer unabsichtlich eine vermeintliche Betriebsstörung auslöst, ist der Sabbat-Modus meist in einem abgelegenen Seitenzweig der Benutzerführung untergebracht.

Nicht immer ist es einfach, der Technik ein Schnippchen zu schlagen. In Kühlschränken von General Electric beispielsweise wurde eine automatische Temperaturanpassung ausgelöst, wenn man ein paarmal hintereinander die Tür öffnete. Von Menschen ausgelöste Temperaturregelung aber ist an Sabbat verboten. Gelöst wurde das Problem, indem die Kühlschranksoftware nun auch emulieren kann, ein Modell aus den neunziger Jahren zu sein – mit statischer Temperaturanpassung.

Rose Bowl Hack 1.0 und 2.0

1961 bekam das Massachussetts Institute of Technology einen PDP-1. Der MIT-Modelleisenbahnbastlerclub umgab die Maschine mit einer eigenen Kultur, dem Urschlamm der Hackerkultur.

Im selben Jahr hackten Studenten des Caltech in Pasadena das "Rose Bowl"-Footballspiel. Einer von ihnen gab sich als Reporter aus und "interviewte" den Verantwortlichen für die sogenannten "card stunts" – das sind die Bilder, die erzeugt werden, wenn die Leute im Publikum farbige Karten hochhalten. Anschliessend manipulierten sie die Ablaufpläne so, dass anstelle eines Huskies – des Maskottchens der gegnerischen Mannschaft – ein Biber zu sehen war, das Maskottchen des Caltech, der "Ingenieur der Natur".

Vor knapp einem Monat zeigte sich, dass auch die Tradition der Analog-Hacks fortgeführt wird (ein Jahr zu spät, wie ich finde, 2003 wären es 42 Jahre gewesen).

Die Rivalität zwischen den Universitäten Harvard und Yale wird seit 121 Jahren sorgsam gepflegt. Am 1. Dezember 2004 verteilten 20 Yale-Studenten, die sich als Harvard-Anheizer ausgaben, an über 1.800 Ehemalige ("Alumnis") rote und weiße Karten, die auf Kommando

hochzuhalten waren, um den Aufruf "GO HARVARD" sichtbar werden zu lassen.

Kurz vor Ende der ersten Spielhälfte wurde das Kommando gegeben, und über die ganze Tribünenfläche war das Selbsteingeständnis der Harvard-Leute zu lesen: "WE SUCK".

Eigentum ist Diebstahl?

Es gibt auch sonderbare, weniger lustige Effekte, die die Moderne nach sich zieht. Wenn ich über den neu bebauten Potsdamer Platz spaziere und die vanillefarbene Einheitsarchitektur der Daimler-City sehe, habe ich den Eindruck, sowas wie den Sieg der Plattenbauweise mit den Mitteln des Kapitalismus vorgeführt zu bekommen.

Ein ähnliches Gefühl entsteht aus dem sonderbaren Widerspruch zu den Ankündigungen der Industrie, den Datenstrom frei fließen zu lassen, und andererseits die immer restriktiveren Einkapselungen der Inhalte durch Digitales Rechte-Management (DRM), wobei es sich dabei ja wohl eher um ein Entrechtungs-Management handelt.

Geht es nach dem Willen der DRM-Falken, wird es ein Eigentumsrecht an digitalem Gut überhaupt nicht mehr geben. Das vollständige Verfügungsrecht – das, was früher Besitz oder Eigentum hieß – gibt es möglicherweise bald nur noch fragmentarisch oder temporär oder gar nicht mehr. Das Motto "Eigentum ist Diebstahl" erhält damit eine ganz neue, zeitgemäße Bedeutung. Die Mittel der freien Marktwirtschaft sind so erfolgreich, dass mit ihrer Hilfe nun offenbar auch die radikalen Grundideen des Kommunismus endlich durchgesetzt werden.

Wir werden, wie gesagt, nie aufhören die Frage zu stellen, ob man die Realität, zu der auch Dinge wie der Gemeinsinn gehören, nicht vielleicht freihacken kann.

Was Hacker zu weit mehr als einem Haufen Technik-Freaks macht, ist das Konzept der Hackerethik und von Utopien wie der eines Menschenrechts auf Information. Von den Anfängen der Raumfahrt bis in die Ära von Atomkraftwerken und Mainframes wurden immer nur Fragen der Machbarkeit erörtert. Es waren Hacker und ihre zum Teil sehr riskanten Unternehmungen, die einen Mainstream von moralischen und gesellschaftlich-politischen Fragen und Forderungen im Zusammenhang mit neuen Technologien initiiert haben.

Karl Kraus schreibt: "Es gibt nur eine Möglichkeit, sich vor der Maschine zu retten. Das ist, sie zu benützen." Und je länger wir mit der neuen Technologie umgehen, desto mehr entdecken wir, was sie nicht kann. Sie vermittelt uns ein lebendiges Gefühl von Souveränität.

Das Chaos lebt

Während des ersten Golfkriegs 1991 haben sich erstmals in der Militärgeschichte Soldaten einem unbemannten Aufklärungsfahrzeug ergeben: Ein Trupp Iraker folgte dem ferngelenkten amerikanischen Sondierwägelchen mit weißen Fahnen.

Die Künder der sogenannten Künstlichen Intelligenz, Leute wie Moravec oder Warwick, haben eine Vision entworfen, nach der die Menschen demnächst von überlegenen Apparaten interniert werden. Die Evolution wird von Maschinen fortgeführt, effizient und mit der ihnen eigenen insektenhaften Eleganz und Kälte.

Wird sich die Aufgabe des Menschen in Zukunft darin erschöpfen, nur noch als Gewürz eine Spur Unordnung in die Kabelsalate zu bringen? Liegt das eigentliche Talent des Menschen also darin, fehlerhaft zu sein?

Tatsächlich kriecht die größte Gefahr aus der Ordnung hervor. Glänzend feststellen kann man das beispielsweise, wenn man gerade frisch renoviert hat. Alles ist an seinem Platz; gesaugt, frisches Tischtuch. Dann kommt, was Philosophen oft als Ziel verkünden: Der Moment der Vollendung. Das einzige, was noch stört, ist man selber. Man ist das einzige, was noch Dreck macht. Mit der Idee der Vollendung ist eine große Falle aufgerichtet worden, und in einem solchen Moment schnappt sie zu.

Man denkt, dass man glücklich sein wird, aber von jeder Ascheflocke, die von der Zigarette auf den Teppich fällt, springt einen ein Mißempfinden an, als hätte ein Hund hingeschissen. Was für ein jämmerliches Selbstgefühl: Ich bin, also störe ich. Die Lebendigkeit kommt erst in den nächsten Tagen wieder in dem selben Tempo, mit dem sich der Glanz des Neuen verliert. Der liebe Dreck.

Wehret der Vollendung, liebe Freundinnen und Freunde: sie ist es, die euch unglücklich macht. Nur das Chaos lebt.

Our Germans

Zusammen mit Wernher von Braun hat sich zu Kriegsende eine Gruppe von Ingenieuren aus Peenemünde den Amerikanern ergeben; sie wurden in die USA gebracht. Erst sollte mutmaßlichen Kriegsverbrechern unter ihnen der Prozeß gemacht werden, aber im Zuge der Aktion "Paperclip" – benannt nach einem Büroklammer-Symbol auf den Personalakten – wurde der Vergangenheit der deutschen Technokraten keine weitere Beachtung m ehr geschenkt. Der Kalte Krieg hatte begonnen. Ein anderer Teil der Peenemünder war in die Sowjetunion gebracht worden und baute Raketen für die Russen.

In der Verfilmung von Tom Wolfes Geschichte der ersten amerikanischen Astronauten gibt es diese Szene, in der jemand im Oktober 1957 die Tür zu einem Beratungszimmer im Weißen Haus aufreißt und ausruft "Es heißt Sputnik!". Auf einem Film, den Agenten in einem russischen Raketenversuchsgelände gedreht haben, und der dann dem Präsidenten vorgeführt wird, sind auch deutsche Wissenschaftler zu sehen. Ein Geheimdienstmann versucht den Präsidenten zu beruhigen: "Our Germans are better than their Germans!"

Es war übrigens keineswegs so, dass Wernher von Braun in Amerika die Idee einer friedlichen Raumfahrt wieder aufgenommen hatte. Die Redstone, die seinen Ruf in den USA begründete, war die erste amerikanische Rakete mit einem Nuklearsprengkopf. 1968 wurde sie in der Bundesrepublik stationiert und 1973 durch die Pershing I abgelöst.

Im Dezember 1979 wurde der sogenannten NATO-Doppelbeschluss gefaßt, der die Stationierung von amerikanischen Pershing II-Mittelstreckenraketen als Gegenüber für die neue sowjetische Mittelstreckenrakete SS-20 vorsah. 1983 stimmte der deutsche Bundestag der Stationierung der Pershing II zu. Aus dem Widerstand gegen den Beschluß erwuchs die deutsche Friedensbewegung. Vier Jahre später wurde ein Abkommen zwischen den USA und der UdSSR unterzeichnet, das die Zerstörung sämtlicher Mittelstreckenraketen vorsah.

Als ich im September 2004 zu einer Lesung nach Düsseldorf eingeladen war, wunderte ich mich erst über den Namen des Veranstaltungsorts: Raketenstation.

Dann, dort in Hombroich am Rand von Düsseldorf, war ich zu Gast bei dem Lyriker Thomas Kling, ein grandioser Lyriker übrigens, und seiner Gefährtin, der Malerin Ute Langanky, einer wirklich großartigen Malerin, und als ich mir die Hände waschen wollte, stand ich im Bad vor drei Waschbecken nebeneinander. Es waren die alten Mannschaftswaschräume der ehemaligen NATO-Raketenstation.

Am Kontrollturm leuchtet jetzt das Efeu, und dann saßen wir im Garten und schauten über eine freundliche Pracht an Pflanzen und ich dachte: So rum kanns auch gehen, erst Rasen und Raketen, dann Oleander und Ginko und Dichter und Malerinnen, und zwei Filmemacherinnen aus Berlin waren auch mit dabei, und ich sah diesen schönen Ort und ich dachte: gehackt.

Jetzt sind wir hier in Berlin, und da ist wieder eine Rakete, die "Fairy Dust", und auch das ist außerordentlich zufriedenstellend, denn es ist eine Rakete, die dazu da ist, auf keinen Fall die Bodenhaftung zu verlieren.

So. Die Spiele sind eröffnet.

Viel Spaß am Gerät.

Die Hackerethik im neuen Jahrtausend

Zapf Dingbatz <ds@ccc.de>

In den letzten Monaten erreichten uns immer wieder Nachrichten über Hacks, die heftige Diskussionen über Verantwortung, Ethik und Moral provozierten. Wesentliche Aufhänger für hitzige Debatten waren die Publikation der Hack-a-Bike Dokumentation in der Datenschleuder, das Massen-Defacement von Kunden-Webseiten eines Berliner ISPs zur Congress-Zeit und zuletzt der in der Mainstream-Presse hinlänglich beleuchtete Angriff unbekannter Aktivisten gegen sächsische Nazi-Webseiten und -Foren.

Die anonymen Hack-a-Biker und die unbekannten Urheber des ISP-Defacements waren die Adressaten von teilweise herber Kritik. Zusammenfassen lassen sich die Vorwürfe vielleicht am besten mit den Fragen: "Denkt Ihr denn gar nicht nach, welche Folgen das für die betroffenen Unternehmen hat? War es wirklich nötig, so einen Schaden anzurichten, nur um zu zeigen, was für tolle Hechte Ihr seid?"

Im Kern geht es immer wieder um die gleichen Fragen: Welche Verantwortung hat man als Einzelner für sein Handeln und welche moralischen Leitlinien können dabei helfen, nicht zu tief ins Klo zu greifen?

Die beiden letzten, auf Wau Holland zurückgehenden Punkte der Hackerethik sollen genau diese kniffligen Fragen addressieren:
- Mülle nicht in den Daten anderer Leute
- Öffentliche Daten nützen, private Daten schützen

In dieser Absolutheit nach den desaströsen Folgen des KGB-Hacks aufgestellt, sind diese Regeln im Hacker-Alltag natürlich immer mal wieder weiter und enger interpretiert worden. Im Wesentlichen haben sie sich bisher bewährt. Aber sind sie in dieser Form noch zeitgemäß?

"Hacktivsm" ist grob verkürzt als "Hacken für einen politischen Zweck" definiert. Das kann, muß aber nicht, im Einklang mit der Hackerethik stehen. Das Bestreben, durch Nutzung der Technologie die Welt schöner und die Menschheit glücklicher zu machen, liegt beiden Konzepten zugrunde. Ist es also legitim, Nazi-Webforen zu hacken, weil das ein Beitrag zur Verhinderung einer ganz sicher nicht glücklich machenden totalitären Regierung ist? Legal ist es ganz sicher nicht, aber das ist eine andere Diskussion.

Wenn es prinzipiell moralisch vertretbar wäre, für einen politischen Zweck zu hacken, wie fände man dann heraus, für welchen Zweck genau dies gelten würde? Sicher nicht für jeden Zweck, oder?

Man könnte argumentieren, daß jeder Zweck, der mit dem allgemeinen Geist der Hackerethik konform geht, d.h. der das friedliche Zusammenleben, die weltumspannende Kommunikation und die Informationsfreiheit fördert, eine hinreichende moralische (nicht juristische!) Rechtfertigung darstellt. Es gibt viele Leser, die das so sehen. Es gibt aber auch etliche, die sagen: "Meine moralische Leitlinie ist im Zweifel das Strafgesetzbuch." Beide Positionen haben ihre starken Punkte.

Natürlich behauptet jede halbwegs clevere Ideologie, daß sie ausschließlich das Wohl der Menschheit im Sinn hat. Nur, wie unterscheidet man "Die Guten"™ von "Den Bösen"™? Wie vermeidet es der "Hacktivist", eine Ansicht zu

unterstützen, die sich vielleicht im Nachhinein als höchst fragwürdig erweist, weil sie zwar das Gute will, aber das Böse schafft?

Wir werden hier kaum zu brauchbaren Antworten kommen. Es gab auch Zeitgenossen, die die lange dauernden Debatten damit erklärten, daß "die Hacker" ja nur neidisch sind, weil "das elitäre Insiderwissen" mittlerweile so weitverbreitet ist, daß es auch politischen Aktivisten neue Handlungsmöglichkeiten eröffnet. Auch dies ist ein nicht von der Hand zu weisender Punkt. Wissen und Fähigkeiten sind nicht regulier- oder beschränkbar. Die Hackerethik ist nicht zuletzt ein Leitfaden für den Umgang mit Fähigkeiten, die im Kern Macht bedeuten. Die implizite Grundannahme ist, daß jemand, der schlau genug ist, hacken zu lernen, auch schlau genug ist, mit den Implikationen umzugehen. Ob das noch so ist, darf angesichts des rasenden Fortschritts bei Technologie allgemein und Angriffswerkzeugen im Besonderen angezweifelt werden.

Jeder muß im Zweifel selbst entscheiden, welche juristischen und ethisch-moralischen Risiken er in Kauf nimmt, um seinen Überzeugungen zu folgen. Der CCC als Institution bekennt sich selbstverständlich zum ausschließlich rechtskonformen Umgang mit Technologie. Das löst aber nicht zwingend die persönlichen Gewissensnöte einzelner Leser.

Die Redaktion möchte daher vielleicht folgende Anhaltspunkte mit auf den Weg geben:

- Die wichtigsten Fragen sind: „Und was passiert dann? Wie geht es hinterher weiter?"
- Es ist immer gut, nochmal darüber nachzudenken, ob diese oder jene Handlung wirklich ein Beitrag dazu ist, die Welt zu einem besseren Ort zu machen.
- Leuten die ein bischen älter sind, haben oft überraschende Einsichten in moralisch-ethische Probleme. Besonders, wenn sie schon gesehen haben, wie Pferde vor, hinter, neben, auf und in der Apotheke kotzen.
- Man kann konkrete Probleme auch prima hypothetisch formulieren und so Vertrauensrisiken vermindern.

Abschließend bleibt noch zu sagen:
Think for yourself, fool.

PS: Auf dem nächsten Congress wird es eine Beratungsstelle für verantwortungsvolles Handeln geben, die bei hypothetisch vorgetragenen ethisch-moralischen Problemen Überlegungshilfe bietet.

Hash Probleme

Rüdiger Weis <ruedi@cryptolabs.org>

In den letzten Monaten wurde das lange Zeit vernachlässigte Gebiet der kryptographischen Hash-Funktionen verstärkt unter Feuer genommen. Hierbei zerbröselte faktisch die gesamte MD4-Hash-Familie. Diese bildet unter anderem die Grundlage für praktisch alle in Anwendung befindlichen digitalen Signaturverfahren.

Besonders hart traf es die noch vielfach genutzten Hash-Funktionen MD4 und MD5, vor denen Kryptographen erst seit über 10 Jahren warnen. MD4 kann inzwischen als Übungsaufgabe auch mit Bleistift und Papier gebrochen werden. Auch SHA-0, die erste Version des Secure Hash Algorithm Standard, ist inzwischen mit einem grösseren Fakultätsrechnerpool an einem Wochenende brechbar. Die von der NSA ohne Veröffentlichung der Gründe modifizierte Version SHA-1 ist höchstwahrscheinlich ebenso einer Gruppe um eine chinesische Mathematikerin zum Opfer gefallen. Dadurch ist eine neue Sicherheitsbewertung faktisch aller digitaler Signaturen notwendig geworden.

Kryptographische Hashfunktionen

Hash-Funktionen erzeugen für Eingaben (praktisch) beliebiger Länge einen kurzen, typischerweise zwischen 128 und 512 Bit langen, „Fingerprint". Für viele kryptographische Anwendungen ist vor allem die Eigenschaft der **Kollisionsresistenz** von besonderer Bedeutung.

Kollisionsresistenz

Als **Kollision** bezeichnet man zwei verschiedene Nachrichten M und M', die auf den selben Hashwert $h(M)=h(M')$ abgebildet werden.

Da es (mehr oder weniger) unendlich viele Eingaben, aber nur endlich viele Hashwerte gibt, ist die Existenz von Kollisionen nicht zu verhindern. Eine Hash-Funktion bezeichnet man dennoch als **kollisionsresistent**, wenn der Rechenaufwand, eine Kollision zu finden, so riesig ist, dass man diesen als praktisch unmöglich ansehen kann.

Aktuell gelten bei vielen Anwendern 2^{80} Rechenoperationen als untere Schranke dieser praktischen Unmöglichkeit. Schon wegen eines trivialen „Geburtstagsangriffes" muss eine Hash-Funktion für dieses Sicherheitsniveau also mindestens einen 160 Bit Hashwert liefern. Allein aus diesem Grunde sollte von Hash-Funktionen wie MD4, MD5 und RIPEMD-128, welche lediglich 128 Bit liefern, abgeraten werden.

Eine Hash-Funktion, die nicht kollisionsresistent ist, sollte als **gebrochen** angesehen werden.

Integritätsprüfungen

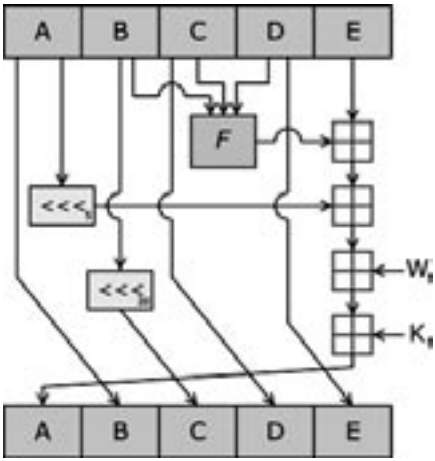
Kryptographische Hash-Funktionen werden in vielen Bereichen der IT-Sicherheit eingesetzt, zum Beispiel bei der Integritätsprüfung (etwa bei tripwire). Dabei wird von als „harmlos" und „nützlich" eingestuften Programmen ein Hashwert abgespeichert. Vor Ausführung des Programms wird dessen Hashwert mit dem Referenz-Hashwert verglichen. Stimmen die beiden nicht überein, wurde das Programm manipuliert. Stimmen Soll- und Ist-Hashwert dagegen überein, dann sollte es sich um das Originalprogramm handeln – allerdings nur, wenn die Hash-Funktion kollisionsresistent ist.

Digitale Signaturen

Besondere Bedeutung haben Hash-Funktionen auch für **digitale Signaturen**: In der Praxis unterschreibt man in der Regel nicht die Nachricht, sondern ihren Hashwert („Hash-Then-Sign Paradigma".

Gelingt es jedoch, zwei kollidierende Nachrichten M und M', M ≠ M' zu finden, gilt also *h(M)=h(M')*, dann ist eine gültige digitale Signatur für M auch gültig für M'.

Schwächen von Hash-Funktionen bezüglich der Kollisionsresistenz können die Rechtssicherheit von Verträgen in Frage stellen.



One iteration within the SHA-1 compression function. A, B, C, D and E are 32-bit words of the state; F is a nonlinear function that varies; <<< denotes left circular shift. K_t is a constant. Quelle: Wikipedia

MD4 und MD5

Ron Rivest veröffentliche 1990 das Design für die 128 Bit Hash-Funktion MD4 [Ri90]. MD4 ist eine iterierende, 3-ründige Hash-Funktion, welche nach dem Merkle-Damgard-Konstruktionsprinzip entwickelt worden ist. MD4 ist recht einfach implementierbar und besonders für die schnelle Verarbeitung auf 32-bit Prozessoren optimiert.

Da schon früh kryptographische Schwächen von MD4-Varianten aufgezeigt werden konnten, machte sich Ron Rivest bereits kurze Zeit später an die Entwicklung einer verstärkten Version.

1991 veröffentlichte Ron Rivest die Hash-Funktion MD5 [Ri91]. Neben einigen Modifikationen in der Kompressionsfunktion wurde eine zusätzliche 4. Runde spezifiziert. Hierdurch ist MD5 etwas langsamer als MD4. Wie MD4 liefert MD5 auch lediglich einen 128 Bit langen Message-Digest.

Bei einer 128 Bit Hash-Funktion benötigt der bereits erwähnte triviale Geburtstagsangriff nur einen Rechenaufwand von 2^{64} Aufrufen der Hash-Funktion, selbst wenn die Hash-Funktion keine spezifischen Schwächen aufweist.

MD4 mit Papier und Bleistift angreifbar

Schon 1996 zeigte der damalige BSI-Mitarbeiter Hans Dobbertin, wie man auf effiziente Weise eine Kollisionen für MD4 finden kann [Do96a]. Der Algorithmus von Dobbertin braucht nur kurze Zeit, um mit der Wahrscheinlichkeit 2^{-32} eine Kollision zu finden. Scheitert der Algorithmus, wird er wiederholt. Dobbertin benutzte für den Angriff einen einfachen PC.

Acht Jahre später wurde nicht einmal mehr ein PC gebraucht. Auf der Rump-Session der Crypto 2004 präsentierten die chinesischen Forscher Wang, Lai, Feng, Chen und Yu Angriffe auf MD4, die praktisch per Hand durchgeführt werden können.

Die neue Attacke findet eine Kollision mit einer Wahrscheinlichkeit von 2^{-6} bis 2^{-8} und einem überraschend niedrigem Aufwand von 2^8 MD4 Berechnungen.

Die Autoren geben auch eine Anwendung ihrer Techniken auf RIPEMD an, welche mit einer Wahrscheinlichkeit von 2^{-17} eine Kollision mit einer Angriffskomplexität von 2^{19} findet.

MD5 gebrochen

Am 17. August 2004 zeigten Xiaoyun Wang, Dengguo Feng, Xuejia Lai und Hongbo Yu Kollisionen für MD5 auf. Der praktische Angriff benötigte nach ihren Angaben auf einem Workstation Cluster nur zwischen 15 Minuten und einer Stunde .

Interessanterweise handelt sich es bei der vorgestellten Methode um einen speziellen differentiellen Angriff, welcher Differentiale bezüglich modularer Subtraktion verwendet.Diese Angriffsmethode findet nach Angaben der Autoren für MD4 eine Kollision mit einem Aufwand von 2^{23} MD4 Berechnungen, 2^{13} für HAVAL-128 [ZPS92] , 2^{33} für RIPEMD und 2^{62} für SHA-0.

Verwendungsstopp für MD4 und MD5

Angesichts der aktuellen Entwicklungen muss die dringende Empfehlung erteilt werden, MD5 und MD4 nicht mehr zu verwenden.

Im Bereich von digitalen Signaturen sollte auf die Verwendung von MD4 und MD5 sofort verzichtet werden. Darüber hinaus sei eine Analyse möglicher Auswirkungen bezüglich bereits geleisteter Signaturen angeraten.

Das US-amerikanische National Institute of Standards and Technology (NIST) William Burr, Manager security technology group, mahnte ebenfalls in einem Interview vom Februar 2005, insbesondere für die sensiblen Bereiche von Zertifikaten und digitalen Signaturen, einen sofortigen Verwendungsstop von MD5 an.

„If by some chance you are still using MD5 in certificates or for digital signatures, you should stop."[Olo5]

SHA-0 und SHA-1

Der SHA Algorithmus (inzwischen als SHA-0 bezeichnet) basiert auf einem von der „National Security Agency" (NSA) stammenden Design, dessen genaue Designprinzipien und Kriterien nicht veröffentlicht wurden.

Schon kurz nach seiner Veröffentlichung wurde dieser Standard mit dem Hinweis auf (unveröffentlichte) Sicherheitslücken modifiziert. Wörtlich heisst es im Standard [NI93]:

"SHA-1 is a technical revision of SHA (FIPS 180). A circular left shift operation has been added to the specifications in section 7, line b, page 9 of FIPS 180 and its equivalent in section 8, line c, page 10 of FIPS 180. This revision improves the security provided by this standard." [NI93]

SHA-0 und SHA-1 lehnen sich nahe an den MD4 Hash-Algorithmus an:

„The SHA-1 is based on principles similar to those used by Professor Ronald L. Rivest of MIT when designing the MD4 message digest algorithm ("The MD4 Message Digest Algorithm," Advances in Cryptology - CRYPTO '90 Proceedings, Springer-Verlag, 1991, pp. 303-311), and is closely modelled after that algorithm." [NI93]

SHA-0 gebrochen

Nach dem Rückzug von SHA(-0) dauerte es nicht lange, bis erste Schwächen von dieser Hash-Funktion auch von den öffentlich forschenden und frei publizierenden Forschern aufgedeckt wurden. Im Jahr 2004 präsentierten dann mehreren Gruppen unabhängig voneinander gravierende Angriffe auf SHA-0.

SHA-1 wahrscheinlich ebenfalls gebrochen

Trotz dieser wirklich klaren Signale zweifelten viele Industrievertreter, ob sich diese Angriffsmethoden auch auf SHA-1 übertragen lassen.

Mein Wettangebot an die Trusted Computing Group "*Ich persönlich würde eher auf die Jungfräulichkeit von Britney Spears wetten, als auf die Sicherheit von SHA-1*" vom Januar 2005 wurde allerdings nicht angenommen.

Schliesslich unterscheidet SHA-1 sich von SHA-0 ja lediglich in einer zusätzlichen, extrem

einfachen Operation: einer zyklischen Linksrotation eines 32-bit Wortes.

So liess der erfolgreiche Angriff nicht lange auf sich warten. Mitte Februar 2005 verbreiteteten die chinesischen Forscher Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu eine 3-seitige Notiz, dass sie Kollisionen für SHA-1 mit einem Zeitaufwand von 2^{69} Aufrufen der Hash-Funktion berechnen können [Sc05].

In ihrer Ankündigung veröffentlichen die Autoren weiterhin eine Kollision für SHA-0 und geben den Rechenaufwand hierfür mit 2^{39} an. Eine Kollision für SHA-1 mit einer auf 58 reduzierten Rundenzahl wurde mit einer Komplexität von 2^{33} gefunden.

Zum Redaktionsschluss dieses Artikels war die wissenschaftliche Diskussion über diese Arbeit allerdings noch nicht vollständig abgeschlossen.

Bezüglich Kosten für einen Angriff auf SHA-1 mittels für Privatleuten zugänglicher Hardware gab Bruce Schneier in seinem Newsletter „cryptogram“ vom 15. März 2005 recht beunruhigende Schätzungen an.

Basierend auf den Erfahrungen mit dem DES-Cracker der EFF von 1999 dürfte heute mit einem Einsatz von 250.000 Dollar eine ähnliche Maschine herstellbar sein, welche in ungefähr 39 Monaten die benötigten 2^{69} Berechnungen durchführen kann. Mit einen Einsatz von 25-38 Millionen Dollar wäre dies in 56 Stunden möglich. Angesichts der Bedeutung von Digitalen Signaturen sicher keine beruhigende Summe.

Merkle-Damgard-Design fragil

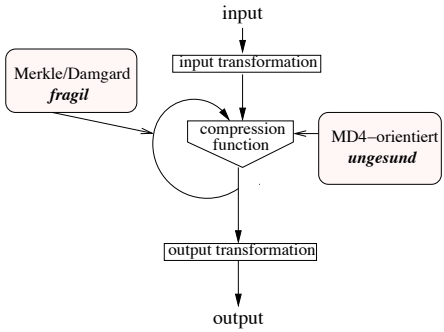
Faktisch alle in der Praxis eingesetzten Hash-Funktionen, wie MD4, MD5, SHA-0, SHA-1, SHA-XXX, RIPEMD, RIPEMD-160 und weitere, orientieren sich in ihrem Design an den theoretischen Grundlagenarbeiten von Merkle [Me89] und Damgard [Da89].

Das Merkle-Damgard-Prinzip besteht darin, eine Hash-Funktion für beliebig lange Eingaben dadurch zu realisieren, dass man eine Mini-Hash-Funktion (die so genannte “Kompressionsfunktion“ für Eingaben fester Länge iteriert. Merkle und Damgard wiesen nach, dass die iterierte Hash-Funktion kollisionsresistent ist, wenn die Kompressionsfunktion ihrerseits kollisionsresistent ist.

Auf der Crypto 04 Konferenz präsentierte Joux [Jo04] jedoch einen generischen Angriff, der eine Schwäche des Merkle-Damgard-Designprinzips aufdeckte.

Vereinfacht ausgedrückt zeigte Joux, dass, wenn man Kollisionen für die Hash-Funktion (bzw. die Kompressionsfunktion) H finden kann, es dann sogar leicht ist, k-fache Kollisionen für H zu berechnen. Der Aufwand hierfür ist nur logarithmisch.

Das Merkle-Damgard-Design für Hash-Funktionen hat sich somit ebenfalls als fragil erwiesen.



Stefan Lucks zeigte in [Lu04] Variationen des Merkle-Damgard-Designs, die gegen den Joux-Angriff immun sind, und für die man sogar formal beweisen kann, dass alle generische Angriffe zum Finden k-facher Kollisionen scheitern.

Interessanterweise verwenden einige Varianten der SHA-2 Familie ähnliche Techniken. Das macht die Geheimhaltung der genauen Designkriterien noch ärgerlicher.

SHA-1 und digitale Signaturen

SHA-1 ist bisher noch für qualifizierte Signaturen nach dem Signaturgesetz zugelassen. Alternativen, wie beispielsweise die Hash-Funktionen des SHA-2 Standards, werden von signaturfähigen Smartcards oft noch nicht unterstützt. Wie sicher sind also heute Signaturen auf Basis von SHA-1?

Zunächst die „guten" Nachichten:

• Der Angriff der chinesischen Forscher impliziert unmittelbar keine Gefahr für Unterschriften, die in der Vergangenheit geleistet wurden. Es sei denn, Angreifer kannten den Angriff schon länger und haben ihn in der Vergangenheit bereits benutzt.

• Die Aufgabe, zu einer bereits unterschriebenen Nachricht M eine Nachricht M' zu finden, zu der die Unterschrift wegen einer Kollision $H(M)=H(M')$ passt, ist schwieriger als die Aufgabe, eine beliebige Kollision zu finden.

• Der Rechenaufwand für den Angriff von Wang und den Co-Autoren ist mit 2^{69} Aufrufen der Hash-Funktion recht gross, und nur von gut finanzierten und motivierten Organisationen zu leisten.

Nun zu den schlechten Nachrichten:

• Der Angriff von Wang lässt es bedrohlich erscheinen, Nachrichten zu unterschreiben, die von einer anderen Partei vorgelegt wurden.

• Die Erfahrung aus der Vergangenheit ist, dass Angriffe immer besser werden. Es wäre also wenig überraschend, wenn die Analyse von SHA-1 in den nächsten Monaten und Jahren noch verbessert, und der Rechenaufwand für das Problem, Kollisionen für SHA-1 zu finden, noch weiter verringert werden würde.

• Die Hauptbedingung für den Angriff von Wang und Co auf SHA-0 besteht darin, dass die Nachrichten M und M' eine bestimmte Differenz aufweisen. Das lässt dem Angreifer große Freiheiten, die Wahl von M und M' einzuschränken, statt sie dem Zufall zu überlassen.

Aus ähnlichen Gründen ist es auch Dobbertin bei MD4 gelungen, gezielte Kollisionen zu finden. Konkret waren das zwei Fassungen eines Kaufvertrages mit sehr unterschiedlichen Kaufpreisen [Do96a].

„At the price of $176,495 Alf Blowfish sells his house to Ann Bonida."

"At the price of $276,495 Alf Blowfish sells his house to Ann Bonida."

Zusammenfassend sei gesagt, dass im Moment bei der Verwendung von SHA-1 für digitale Signaturen **kein Grund zur sofortigen Panik** besteht.

Unabhängig jedoch davon, ob sich die Angaben der chinesischen Forscher bestätigen, **ist es an der Zeit, die Hash-Funktion SHA-1 durch Alternativen zu ersetzen**.

Die Angriffe auf eng verwandte Hash-Funktionen, einschliesslich SHA-0, sind zu eindrucksvoll, als dass man SHA-1 noch lange vertrauen sollte.

Wir verweisen nochmals ausdrücklich auf die schon seit Jahren geäusserten Warnungen, SHA-1 nicht innerhalb der geplanten Trusted Computing Initiativen zu verwenden. [WeB02 ,WB03,WB04,We04,WL05,...]

Gibt es Alternativen?

Für die gestärkte RIPEMD Variante RIPEMD-160 [DBP96] wurden bisher keine praktischen Angriffe gefunden. Dennoch lassen die Angriffe auf die Vorversion RIPEMD und die Mitgliedschaft in der MD4 Familie zur Vorsicht raten. Eine 160 Bit Hash-Funktion sollte ohnehin nur als temporäre Lösung für wenige Jahre betrachtet werden.

Als Alternative bieten sich beispielsweise **algebraische Kombinationen von Funktionen mit unterschiedlichem Basisdesign** an (s.a.

[We00]). Interessante Ansätze stellen unserer Meinung hierfür beispielsweise Tiger [AB96] und Whirlpool [BR00] aus dem europäischen NESSIE Projekt dar.

Der neue **NIST Standard SHA-2** definiert eine ganze Reihe von Funktionen mit längerem Hashwert. Sie liefern zwischen 224 Bit (SHA-224) und 512 Bit (SHA-512) lange Hashwerte. Allerdings existieren für diese Funktionenfamilie erst sehr wenige Analysen [GH03].

Zudem gab es auch, im Gegensatz zum sehr erfolgreichen Wettbewerb um den Blockchiffre AES, im Falle der Hash-Funktionen leider keinen Wettbewerb – vielmehr hat das NIST, wie bei SHA-0 und SHA-1, auf ein von der NSA stammendes Design zurückgegriffen, dessen Designprinzipien und -kriterien nicht veröffentlicht wurden.

Henri Gilbert und Helena Handschuh zeigten, dass die Funktionen gegen eine Reihe von in SHA-1 und MD4 möglichen Angriffstechniken sehr widerstandsfähig sind. Beunruhigend ist allerdings, dass modifizierte Versionen gravierend schwach sind – auch bei nur geringen Vereinfachungen:

„However we show that slightly simplified versions of the hash functions are surprisingly weak: whenever symmetric constants and initialization values are used throughout the computations, and modular additions are replaced by exclusive or operations, symmetric messages hash to symmetric digests." [GH03]

Für Kryptographen sind die Hash-Algorithmen der SHA-2 Familie damit unangenehm fragil – sehr kleine Änderungen der Hash-Funktion sollten eigentlich nur moderate Auswirkungen auf die Sicherheit der Hash-Funktion haben.

Dennoch kann die Verwendung der Funktionen des SHA-2 Standards im Vergleich zu SHA-1 die bessere Alternative sein. Diese Einschätzung sehen wir durch die momentane Abwesenheit von veröffentlichten Angriffen, die längeren Hashwerte und die etwas aufwendigeren Berechnungen motiviert.

Einige moderne Sicherheitslösungen wie GnuPG bieten heute bereits SHA-256 als Option [GP05]. Cryptophone verwendet SHA-256 zur Erhöhung der Robustheit unterschiedlich parametrisiert hintereinander [Cr03]. Mir persönlich gefällt allerdings auch für die SHA-2 Familie das gesamte Design der Kompressionsfunktion nicht. Es ist mir recht schleierhaft, wieso nicht besser verstandene Konstruktionsmethoden – orientiert etwa am Design von Blockchiffrierern – verstärkt eingesetzt werden.

Zudem scheint es eine europäische Unart zu sein, Professoren-Stellen für Computersicherheit verstärkt mit Personen zu besetzen, welche nicht mal über kryptographische Grundkenntnisse verfügen. Es bleibt zu hoffen, dass hier langsam ein Umdenken einsetzt, bevor auch die letzten praktisch eingesetzten Verfahren zertrümmert sind.

Es sei daher eine verstärkte Förderung der kryptographischen Forschung und die Ausschreibung eines Wettbewerbe um eine Standard-Hash-Funktion, analog zum AES-Wettbewerb um eine Standard-Blockchiffre, dringend angeraten.

„Doppelt genäht"

Eine unelegante, aber kryptographisch robuste Methode, ohne grossen Aufwand anwendbar, ist die Verwendung mehrerer Hash-Funktionen.

Beispielsweise kann man einen Hash H durch das Aneinanderhängen des 128 Bit MD5-Hashes und des 160 Bit SHA-1 Hashes bilden: $H(x):=(MD5(x)||SHA\text{-}1(x))$.

Eine Kollision für H ist eine Kollision für **jede** der Hash-Funktionen, aus denen H zusammengesetzt wurde. Eine derartige „Kaskade" von Hash-Funktionen hat deshalb trivialerweise die Eigenschaft, dass der zusammengesetzte Hash mindestens so stark ist, wie der stärkste einzelne Hash.

Da bei RSA Signaturen ohnehin Schlüssellängen von 2048 oder mehr Bit verwendet werden, ist genügend Raum für derartige Konstruktio-

nen, ohne dass eine zweite modulare Exponation notwendig wäre. (Dies träfe sogar für 1024 Bit RSA Schlüssel zu, von deren Verwendung allerdings abgeraten werden sollte [WLB04].)

Eine derartige Randomisierung des Signaturinputs erhöht sogar die Robustheit des Signaturverfahrens im Vergleich zu den immer noch in Verwendung befindlichen trivialen Signaturpaddingverfahren mit nur einem Hashwert. Die Geschwindigkeitseinbussen erscheinen angesichts der deutlich aufwendigeren eigentlichen Signaturfunktion als vernachlässigbar.

Man kann sogar darauf hoffen, dass die Kaskade mehrerer Hash-Funktionens stärker ist, als jede einzelne Hash-Funktion **sollten** doch die Kollisionen einer Funktion keine Kollision für die andere Hash-Funktion darstellen.

Leider könnte diese Hoffnung auch trügerisch sein. Joux [J004] demonstrierte dies mit einer Anwendung seines Multi-Kollisionsangriffs auf Kaskaden. Ein weiteres Problem ist, dass praktisch alle in Anwendung befindlichen Hash-Funktionen auf den problematischen Bausteinen MD4 und Merkle-Damgard basieren.

Daher möchten wir die Verwendung von unterschiedlich konstruierten Funktionen innerhalb der Kaskade, beispielsweise *H'(x):=(SHA-256(x) || Whirlpool(x) || Tiger(x))* anraten [vgl .We00].

Literatur

[AB96] Anderson, R., Biham, E., "Tiger: A Fast New Hash Function", Fast Software Encryption - FSE'96, LNCS 1039, Springer-Verlag (1996), pp. 89-97.

[BR00] Barreto, P., Rijmen, V., „The Whirlpool Hashing Function", First open NESSIE Workshop, Leuven, Belgium, 13--14 November 2000.

[Cr03] Cryptophone FAQ
http://www.gsmk.de/html/faq_en.html

[DBP96] Dobbertin, H., Bosselaers, A., Preneel, B., „RIPEMD-160, a strengthened version of RIPEMD,'' Fast Software Encryption 1996, LNCS 1039, D. Gollmann, Ed., Springer-Verlag, 1996, pp. 71-82.

[Da89] Damgard, I. „A design principle for hash functions." Crypto 89, LNCS 435, pp. 416 27.

[Do96a] Dobbertin, H., „Cryptoanalysis of MD4 Fast Software Encryption", 1996.

[Do96b] Dobbertin, H., „The Status of MD5 After a Recent Attack'', CryptoBytes 2(2), 1996.

[D98] Dobbertin, H., „Cryptanalysis of MD4", Journal of Cryptology 11:4 (1998), pp. 253-271.

[GP05] GNU Privacy Guard, *http://www.gnupg.org/*

[Ol05] Olsen, F., IST moves to stronger hashing Federal Computer Week, Feb. 7, 2005

[Lu04] Lucks, S., Design Principles for Iterated Hash Functions, Cryptology ePrint Archive: Report 2004/253.

[NE03] NESSIE, "Portfolio of recoendatet cryptographic primitives",
https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/decision-final.pdf

[NI93] National Institute of Standards and Technology (NIST), FIPS180-1, SECURE HASH STANDARD,
http://www.itl.nist.gov/fipspubs/fip180-1.htm

[GH03] Henri Gilbert, H., Handschuh, H., "Security analysis of SHA-256 and sisters" in M. Matsui and R. Zuccherato, Eds., Selected Areas in Cryptography (SAC 2003), vol. 3006 of Lecture Notes in Computer Science, pp. 175-193, Springer-Verlag, 2004.

[J004] Joux, A., Multicollisions in iterated hash functions, application to cascaded constructions. Crypto 04, LNCS 3152, pp. 306--316.

[Me89] Merkle, R., One-way hash functions and DES. Crypto 89, LNCS 435, pp. 428--446.

[Sc05] Schneier, B., SHA-1 Broken, *http://www.schneier.com/blog/archives/2005/02/sha1_broken.html*

[Wea05] Wang, X., Lai, X., Feng, D., Chen, H., Yu, X., ‚Cryptoanalysis for Hash Functions MD4 and RIPEMD', preprint, 2005.

[We00] Weis, R., „Protocols and Algotithms" PhD Thesis, 2000.

[WB02] Weis, R., Bogk, A., Trusted Comuting, Chaos Communication Congress, Berlin, 2002.
http://www.cryptolabs.org/tcpa/WeisTCPAfutilePrinterfriendly.pdf

[WB03] Weis, R., Bogk, A., Trusted Computing, Chaos Communication Congress, Berlin, 2003.

[WB04] Weis, R., Bogk, A., Trusted Computing, Chaos Communication Congress, Berlin, 2004.

[We04] Weis, R., "Trusted Computing – Chancen und Risiken" DuD 11/04.

[WL05] Weis, R., Lucks, S., "Hash-Funktionen gebrochen" DuD 03/05, 2005.

[WLB04] Weis, R., Lucks, S., Bogk, A., „Sicherheit von 1024-bit RSA-Schlüsseln gefährdet" DuD, 2004.

[ZPS92] Y. Zheng, J. Pieprzyk, and J. Seberry, „HAVAL - a one-way hashing algorithm with variable length of output", Advances in Cryptology - Auscrypt'92, LNCS 718, Springer-Verlag (1993), pp. 83--104.

Datenspur Papier

Frank Rosengart <frank@rosengart.de>

Versteckte Seriennummern auf Farbkopien und Mustererkennung in Kopierern, Druckern und Scannern sollen die Herstellung von Blüten durch "Hobbyfälscher" eindämmen.

Geldfälscher, die noch echte Druckplatten für die Blüten anfertigen, werden selten. Moderne Farbkopierer bieten für „Hobbyfälscher" verlockende Möglichkeiten, ohne großen Aufwand sehr echt aussehende Falsifikate von Euro-Noten herzustellen. Um es den Gelegenheitsfälschern nicht allzu einfach zu machen, haben sich Kopiererhersteller einige technische Raffinessen ausgedacht.

Erkennung von Geldscheinen: EURion-Konstellation

Das ältere Verfahren zur Erkennung von Banknoten basiert auf kleinen, farbigen Kreisen, die auf Geldscheinen oder anderen Wertdokumenten aufgedruckt sein können. Diese Kreise haben einen Durchmesser von einem Millimeter, sind gelb, rot oder grün und im normalen Druckbild verteilt. Sie sind in sternförmiger Anordnung gruppiert. Kopiergeräte können so per einfacher Mustererkennung einen Geldschein erkennen. Dieses Verfahren wurde zum ersten Mal von Markus G. Kuhn im Jahr 2002 öffentlich beschrieben. Viele Farbkopierer bemerken diese Markierung und drucken dann entweder eine schwarze Seite oder verweigern vollständig den Dienst. Meist geht das einher mit einer kryptischen Fehlermeldung am Gerät, welche nur vom Servicetechniker zurückgesetzt werden kann. Copy-Shop-Mitarbeiter berichten von gelegentlichen Problemen, wenn jemand Landkarten kopieren möchte: Besonders im Ruhrgebiet kann sich eine solche Konstellation ergeben, wenn Städte mit einer bestimmten Einwohnerzahl als kleiner Kreis eingezeichnet sind.

Erkennung von Geldscheinen: Digimarc - Kennzeichnung

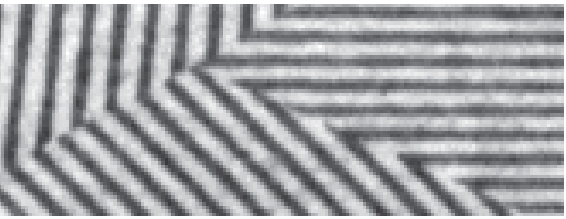
Ein neueres Verfahren wurde von der Central Banks Counterfeit Deterrence Group entwickelt und von der auf digitale Wasserzeichen spezialisierten Firma Digimarc implementiert. Es kann von Herstellern von Grafikprogrammen unentgeltlich verwendet werden, der Quellcode und der verwendete Algorithmus sind jedoch für

die Programmierer der Grafikprogramme nicht einsehbar und bleiben das Geheimnis von Digimarc. Auf Grund seiner Komplexität bei der Mustererkennung wird es hauptsächlich von aktuellen Grafikprogrammen wie Paintshop Pro oder Adobe Photoshop (ab Version 8) verwendet und ist bisher nicht in Farbkopierern zu finden.

Steven J. Murdoch und Ben Laurie, beide forschen in der Computer Security Group der University of Cambridge, haben dieses Verfahren untersucht, um Rückschlüsse auf die verwendeten Algorithmen zu finden. Auf dem 21. Chaos Communication Congress haben die beiden das Ergebnis ihrer bisherigen Analyse vorgestellt.

„Das Verfahren basiert auf einer bisher nicht vollständig identifizierten Frequenztransformation, vergleichbar mit einer Fast Fourier Transformation oder Discrete Cosinus Transformation, wie sie auch bei verlustbehafteter Audio- oder Bildkompression verwendet wird", so Murdoch. Mit verschiedenen Ausschnitten von Geldscheinen haben die beiden Forscher sich an die für die Mustererkennung relevanten Abschnitte auf den Scheinen herangetastet.

„Nach unseren Erkenntnissen werden bestimmte Kantenübergänge im Hintergrundmuster der Banknote ausgewertet", spekuliert Murdoch.



Wenn der Anwender einen Geldschein oder etwas, was der Computer dafür hält, in die Bildbearbeitung einlesen möchte, erscheint auf dem Bildschirm ein Hinweisfenster, dass dies nicht möglich sei. Dabei spielt es keine Rolle, welche Absichten der Nutzer hat oder ob nicht möglicherweise eine legale Kopie der Banknote angefertigt werden soll (Ausdruck 50% oder 200% der Originalgröße, sowie bestimmte Ausschnitte sind gesetzlich erlaubt).

Auch in Treibern von Tintenstrahldruckern scheint diese Mustererkennung eingebaut zu sein und – um direkt an der Quelle anzusetzen – möglicherweise sogar in Scanner-Treibern. Damit dürfte auch erklärt sein, warum einige Druckerhersteller, die bisher sehr freizügig Linux-Treiber im Quellcode mitgeliefert haben, jetzt wieder zu reinen Binär-Modulen übergehen. Wenn man versucht, auf einem aktuellen Tintenstrahldrucker von HP einen Geldschein auszudrucken, erscheint nach wenigen Druckzeilen auf dem Papier anstelle des Bildes die Webadresse *http://www.rulesforuse.org/*, wo erklärt wird, was gerade passiert ist. Die Logdateien des Servers dürften interessant sein, da in der Regel nur Besucher dorthin gelangen, die gerade versucht haben, einen Geldschein zu kopieren.

Eine gesetzliche Verpflichtung, solche Beschränkungen einzubauen, besteht bisher zumindest in Europa nicht. „Adobe übt sich hier in vorauseilendem Gehorsam, weil das Unternehmen Regierung und Behörden als gute Kunden hat und diese auch gern behalten möchte", mutmaßt Murdoch über die Beweggründe, diese Technologie freiwillig einzusetzen.

Kennzeichnung von Farbkopien

Wenn schon Wertdokumente oder andere kritische Papiere täuschend echt auf Farbkopierern vervielfältigt werden können, möchten Ermittlungsbehörden gern die Spur einer Kopie nachvollziehen können. Diesen Wunsch können die Hersteller von Farbkopierern und zukünftig auch Druckern befriedigen, indem auf jede Farbkopie für das menschliche Auge unsichtbar in einem Punktraster die Seriennummer des Gerätes aufgedruckt wird. Um diesen versteckten Code gab es viele Jahre Spekulationen in diversen Internet-Foren, eine öffentliche Dokumentation ist jedoch nicht zu finden. Im Jahr 2004 hat die Firma Canon Deutschland einen BigBrotherAward für die Kennzeichnung jeder Farbkopie bekommen. Die BigBrotherAward-Jury ist durch eine spezielle interne Service-Anleitung für Canon-Techniker auf die

Kennzeichnung aufmerksam geworden. Dort wird auch erklärt, welche Fehlercodes das Gerät anzeigt, wenn die Seriennummer an bestimmten Geräte-Bauteilen nicht mit der in der Elektronik gespeicherten übereinstimmt. So ist zum Beispiel unter der Glasplatte ein weiterer Aufkleber mit der Gerätenummer, der bei jedem Kopiervorgang eingescannt wird, um zu verhindern, dass die Seriennummer des Gerätes manipuliert wird.

Mit dem bloßen Auge ist der aufgedruckte Code nicht zu erkennen. Sichtbar wird ein regelmäßiges Punktraster in hellgelb, wenn eine Farbkopie unter UV-Licht gehalten wird oder wenn die zu untersuchende Kopie auf einem Scanner mit hoher Auflösung eingescannt wird. Bei etwa 30-facher Vergrößerung kann man die einzelnen Punkte des Rasters auswerten und in ein Binärmuster (1 = Punkt gesetzt, 0 = kein Punkt) überführen. Bei aktuellen Farbkopierern lässt sich eine Matrix von 32x16 Punkten = 512 Bit ablesen, was 64 Zeichen entspricht. Damit wäre es sogar möglich, jede Kopie einzeln zu kennzeichnen oder zumindest die aktuelle Uhrzeit/Datum auf der Kopie zu vermerken.

Über bestehende Serviceverträge, Ersatzteilkäufe oder Vergleichskopien kann die Polizei mit Hilfe des Herstellers einen gefälschten Geldschein oder jede beliebige andere Farbkopie zu dem Gerät zurückverfolgen, auf dem das Duplikat gedruckt wurde.

Gesetzliche Verpflichtungen für solche Kennzeichnungen oder Beschränkungen gibt es zumindest in Deutschland nicht. Umso fragwürdiger sind daher diese technischen Barrieren und die heimliche Spur. Die BigBrotherAward-Jury äußerte ihre Sorge in der Begründung für den Award: „Was für die Strafverfolgung noch gerechtfertigt erscheinen mag, ist eine Gefahr für die Informationsfreiheit: Wer wird sich noch trauen, Bestechungsskandale aufzudecken und entsprechende Beweise, z.B. an die Presse, weiterzureichen, wenn er weiß, dass die Anonymität einer Kopie nicht mehr gewährleistet ist, sondern dass z.B. sein Arbeitgeber als Besitzer des Gerätes ein Dokument bis in eine bestimmte Abteilung, ein bestimmtes Büro zurückverfolgen kann?“

Die Kennzeichnung von Farbkopien und eine Mustererkennung für Geldscheine ist aber erst der Anfang: Einigen Druckerherstellern zufolge soll in naher Zukunft jeder Tintenstrahldrucker und jeder Farblaserdrucker für Zuhause eine individuelle Markierung auf dem Papier hinterlassen. Und für den Kopierschutz in bestimmten Motiven dürfte sich die Musik- und Filmindustrie interessieren: Ein auf dem Farbdrucker ausgedrucktes DVD-Cover macht die Kopie erst so richtig perfekt. Auch Firmen können ihre vertraulichen Dokumente mit einem entsprechenden Kennzeichen versehen, um die Firmengeheimnisse nicht in die falschen Hände geraten zu lassen.

Analyse der BSI Studien BioP und BioFinger

starbug <starbug@berlin.ccc.de>

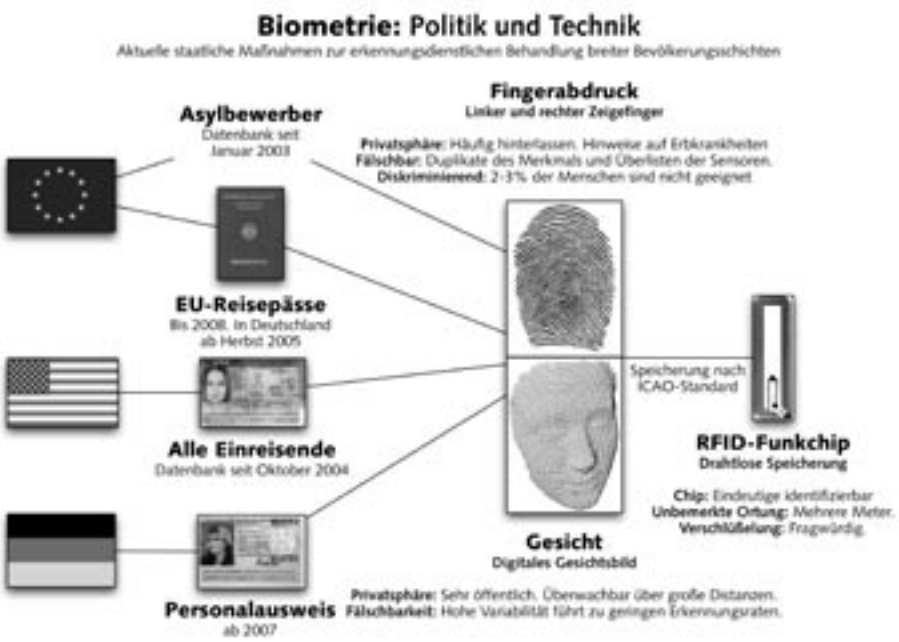
Nachdem in der Datenschleuderausgabe 83 versucht wurde, den inhaltlich sehr guten und umfassenden Bericht des Büros für Technikfolgeabschätzung des Bundestages #93 (https://ds.ccc.de/083/biometrieimausweis/) zusammenzufassen, beschäftigt sich dieser Artikel gleich mit zwei Test-Berichten des Bundesamtes für Sicherheit in der Informationstechnik (BSI): einer Einschätzung der Funktionsfähigkeit von Gesichtserkennugssytemen (BioP1) und von Fingerabdrucksystemen (BioFinger). Besondere Brisanz gewinnen diese Arbeiten durch die geplante Einführung von Gesichts- und Fingerabdrucksystemen zur Personenkontrolle an Grenzen und auf der Strasse. Dabei werden ab Herbst diesen Jahres digitale Bilder des Gesichts und ab 2007 zusätzlich Bilder beider Zeigefinger in alle neu ausgestellten Pässe aufgenommen. Ebenfalls ab 2007 ist für Deutschland ein neuer Personalausweis mit diesen biometrischen Merkmalen geplant. Vorrangetrieben werden diese Vorhaben in erster Linie durch unseren Bundesminister des Innern Herrn Schily. Und das trotz der teilweise erschreckenden Ergebnisse dieser beiden Tests, in Auftrag gegeben von seinem Ministerium.

BioP1

Ziel der Studie war es, die Bedingungen für den Einsatz eines Gesichtsbildes als biometrisches Merkmals auf neuen Ausweisdokumenten zu untersuchen. Im Speziellen ging es um die Bildqualität und die Form der Speicherung. Ferner sollten die möglichen Einflussfaktoren analysiert und deren Auswirkung auf die Erkennungsleistung bestimmt werden. Auch die Überwindungssicherheit fand Beachtung. Verglichen wurden dabei verschiedene Gesichtserkennungssysteme, mit unterschiedlichen Gesichtsfindungs- und Erkennungsalgorithmen sowie eine Reihe möglicher Referenzdaten (Templates). Diese sind im Einzelnen:

1. Bilddatei einer frontalen Bildaufnahme unkomprimiert (75kB)
2. Musterpersonalausweis mit Foto (aus 1.) und Enrollment vor jeder Authentifizierung
3. Foto auf EU-Visumaufkleber (aus 1.) (142kB)
4. komprimierte Bilddatei gemäß ICAO (aus 1.) (14kB)
5. Bilddatei einer Halbprofilaufnahme (75kB)
6. Foto eines aktuellen Personalausweises (65kB Graustufen)
7. System-Template durch Live Enrollment
8. wie 2. aber mit einmaligem Enrollment (65kB)

Die Tests wurden im Verifikationsmodus (1:1 Vergleich) unter einheitlichen Beleuchtungsbedingungen (schlagschatten- und blendfrei, 130Lux (entsprechend DIN 5035 Teil2 "Empfangsräume und Räume mit Publikumsverkehr")) durchgeführt. Es standen zwei Systeme und drei unterschiedliche Algorithmen zur Verfügung. Als Ausweisleser wurde ein Scanner (ca. 300dpi (472 x 620 Pixel) 8 Bit Graustufen) der Bundesdruckerei verwendet. Der siebenwöchige Testfand am BKA-Dienstsitz Wiesbaden



mit 241 Mitarbeitern statt, von denen 152 mehr als 50 Vergleiche tätigten. Insgesamt fanden über 10.000 Betätigungen statt.

Ergebnisse

Die Studie hat gezeigt, dass die Fotos auf den aktuellen Personalausweisen nicht für die biometrische Gesichtserkennung geeignet sind. Das liegt vor allem an der schlechten Bildqualität, insbesondere am fehlenden Kontrast und an der Aufnahme des Gesichts im Halbprofil. Der Musterausweis nach den ICAO-Normen zeigt, dass ein Vergleich mit Fotos auf Ausweisen generell funktioniert, auch wenn die Ergebnisse noch nicht zufriedenstellend sind.

Der direkte Vergleich der unterschiedlichen Referenztemplates liefert die beste Erkennungsleistung für die Templates 7, 1 und 4, wobei nur Nummer 7 akzeptable FRR (fälschliche Rückweisung einer berechtigten Person) lieferte. So wurden hier nur zwei Personen in mehr als 10% der Fälle zurückgewiesen, wohingegen die FRR für die Bilddatei nach ICAO-Standard bei ca. 10% der Nutzer höher als 30% lag (bei einer False Acceptance Rate (FAR) von 0,1%). Im realen Einsatz sind allerdings höhere FRRs zu erwarten. Dort würde das Template gegen mehrere Bilder verglichen, was für Unberechtigte eine deutlich bessere Erkennungsrate bedeute. Als Konsequenz müsste der Schwellwert des Systems angepasst werden, und so würde zwangsweise die FRR steigen.

Die Referenztemplates 5 und 6 schnitten am Schlechtesten ab. Im Test nahm die FRR nach einer Gewöhnungsphase von einigen Tagen ab. Denkt man aber an den Einsatz bei Grenzübertritten mit vll. 2 - 4 Benutzungen pro Jahr würde das eine "Eingewöhnungszeit" von mehreren Jahren bedeuten. Eine schwache Kompression der Templatedaten (75kB) hat fast keinen Einfluss auf die Erkennungsleistung des Systems. Bei starker Kompression (11kB) zeigt sich allerdings ein deutlicher Abfall. Die von der ICAO vorgeschlagene Größe von 14kB würde noch akzeptable Ergebnisse liefern. Eine Verringerung der Bildauflösung führt zu leicht schlechteren Erkennungsraten. Den Einfluss des Alters

der Referenz auf die Systemperformance konnte die Studie auf Grund des Mangels an Daten mit ausreichend hoher Bildqualität nicht klären.

Ein Trend, der die Abnahme der Erkennungsleistung mit zunehmenden Templatealter zeigt, war allerdings zu beobachten. Das größte Problem von Gesichtserkennungssytemen bleibt weiterhin der Lichteinfall. So führt Seitenlicht zu extremer Verschlechterung der Erkennungsraten. Hintergrundlicht hingegen spielt so gut wie keine Rolle. Auch die Software wartete mit teils gravierenden Mängeln auf. So lieferte sie fehlerhafte Verifikationsergebnisse und machte Aufnahmen von Teilgesichtern oder vollständig fehlenden Person.

Überwindungssicherheit

Durchgeführt wurden die Tests mit dem Template 7, welches zuvor die besten Ergebnisse gezeigt hatte. Eine Lebenderkennung wurde nicht gefordert. Die verwendeten Systeme konnten mit einfachsten Mitteln überwunden werden. Es wurden sowohl Farb- und Schwarzweißfoto als auch Videos als berechtigte Nutzer erkannt. Selbst augenscheinlich nicht ähnliche Personen lieferten reproduzierbar hohe Matchscores. Daher fordert der Bericht auf Seite 92, dass die "Verbesserung der Überwindungssicherheit [...] eine wesentliche Grundvorraussetzung" für den Einsatz ist.

Gleichzeitig kann man aber auf Seite 72 auch folgenden Satz lesen: „Die Verifikation biometrischer Merkmale wird eine Überwindung nie hundertprozentig verhindern können. Dies liegt unter anderem darin begründet, dass das Ziel einer akzeptablen Falschabweisungsrate zur Einstellung eines Schwellwertes führt, der von einem Angreifer zur Überwindung ausgenutzt werden kann."

Nutzerbefragung

Die im Nachhinein durchgeführte Nutzerbefragung unter den Probanden des BKAs stellte als größten Mangel der Systeme eine hohe Störanfällig fest. Die Bewertung der Erkennungsgenauigkeit und Toleranz gegenüber Änderungen des Gesichts verschlechterte sich im Lauf des Versuches. So erhielten beide Systeme in der Kategorie Benutzerakzeptanz, Überwindungssicherheit und Lichteinfluss die Schulnote 5 (5,16 bzw. 4,93). Außerdem forderte die Mehrheit der Beamten, Gesichtserkennungssysteme nicht isoliert einzusetzen. Eine generelle Nützlichkeit wurde nur von einem Drittel der Befragten gesehen!

Kriterium	System A	System B
Erkennungsleistung[1]	3,11	3,33
Systemverhalten[2]	4,50	2,23
Weiterführende Untersuchungen[3]	5,16	4,93
Herstellerbeurteilung[4]	5,00	2,30

1) Besteht aus den Einzelkriterien FER, FAR, FRR, Standardabweichung Einzelbenutzerstatistik
2) Besteht aus den Einzelkriterien Systemfehler, Ausfallverhalten, Administrationsaufwand, benutzerbedingte Probleme, mittlere Bedienzeit
3) Besteht aus den Einzelkriterien Benutzerakzeptanz, Überwindungssicherheit, Einfluss Lichtbedingungen
4) Besteht aus Einzelkriterien Support/Service (inkl. Inbetriebnahme)

Die dunkelgrau gehaltenen Felder bedeuten „durchgefallen"

BioFinger

Das was die BioP-Studie für die Gesichtserkennung, analysiert BioFinger für den Fingerabdruck. Auch hier geht es um die mögliche Integration in deutsche Personaldokumente zur Verbesserung der Verifikation des Ausweisinhabers.

Im Einzelnen soll BioFinger die Fähigkeit zur Unterscheidung biometrischer Zwillinge, die Langzeitstabilität von Fingerabdrucken und den Einfluss des Alterungsprozess auf die Performance der Algorithmen untersuchen. Die beiden letzten Punkte sind der relativ langen Nutzungsdauer der Dokumente von zehn Jahre geschuldet. Zum Einsatz kamen elf Sensoren und sieben Algorithmen. Pro Sensor wurden 2160 Datensätze erzeugt, jeweils neun Aufnahmen von 8 Fingern von 30 Personen. Für eine qualifizierte statistische Aussage sind das natürlich deutlich zu wenig. Nochmals deutlich

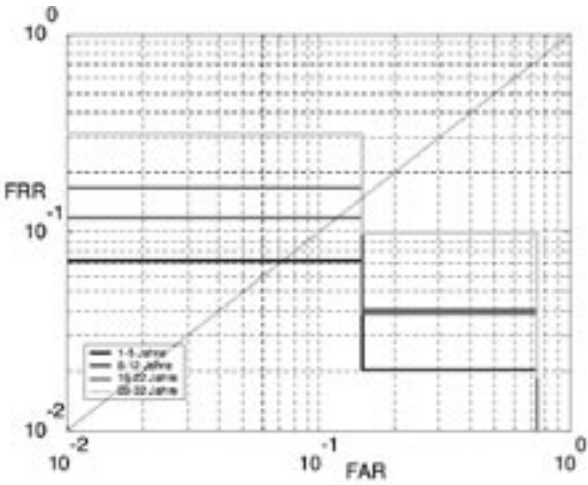
weniger Daten wurden bei der Untersuchung des Alterungseinflusses verwendet. Initial sind hier Fingerabdrücke von 183 Personen involviert. Für ein Alter von zehn Jahre sind es noch 65, für 30 Jahre sogar nur noch 26 Personen. Die Unterscheidbarkeit biometrischer Zwillinge wurde mit sieben Listen von BKA Datensätzen durchgeführt, in denen verschiedene Personen katalogisiert waren, bei denen das AFIS-System eine signifikante Ähnlichkeit festgestellt hat.

Ergebnisse

Als ein Qualitätsmerkmal von Sensoren gilt die Fähigkeit, Papillarleisten von Tälern zu unterscheiden. Für den so genannten RIP Count wird dazu deren Anzahl zwischen Core und Delta bzw. zwischen zwei Minutien ermittelt. Von den hier verwendeten Sensoren zeigten sieben eine mäßige, drei eine gute und einer eine sehr gute Trennung. Optische Sensoren schnitten hierbei besser ab, als kapazitive Sensoren. Als direktes Unterscheidungsmerkmal dienen die Fehler bei der Merkmalsaufnahme. Hier unterscheidet man in hardware- und in softwarebasierte. Der hardwareinduzierte FTA (failure to aquire) lag für sieben Systeme bei 0% und für die restlichen zwischen 0,1 und 1,2%. Ebenfalls bei 0% lag der softwareinduzierte FTE (failure to enroll) von vier der sieben Algorithmen. Die verbliebenen drei Systeme wiesen Werte von 0,05% bis 23% auf.

Eine Aussage über die Gesamtqualität der Systeme liefern die FAR (ein unberechtigter Nutzer wird zugelassen) und die FRR (ein berechtigter Nutzer wird abgewiesen). Beide hängen direkt miteinander zusammen. Je nach Anwendungsfall lässt man entweder höhere FARs oder FRRs zu. Für die Verifikation von Ausweisinhabern ist eine FAR von 0,1% durchaus akzeptabel. Basierend auf dieser Annahme weist die Hälfte der Systeme FRRs von kleiner 10% auf, 23% der Systeme kommen unter 3%. Das heißt dann aber immer noch, dass bei ca. jeder 30sten Überprüfung ein berechtigter Dokumenteninhaber einer genaueren Prüfung unterzogen werden würde. Die Tests zu Biometrische Zwillinge zeigten, dass deren Matchscores sich durchaus voneinander unterscheiden, wenn auch nur im niedrigen einstelligen Prozentbereich. (S.84) Angemerkt werden muss dazu aber, dass diese AFIS-Zwillinge nicht zwangsweise auch als Zwillinge für die hier verwendeten Systeme gelten müssen, da AFIS musterbasierte Algorithmen benutzt, die hier getesteten Algorithmen aber weitgehend minutienbasiert sind.

Auf Grund der sehr kleinen Anzahl von Datensätzen kann keine definitive Aussage zum Einfluss des Templatealters auf die Erkennung getroffen werden. Der Bericht geht von Verschlechterungen der FRR um Faktor 1,4 für eine Differenz von zehn Jahren zwischen der Templateerstellung und dem Vergleich aus. Bei 30 Jahren beträgt der Faktor dann ca. 2,5. Eine Verkleinerung der Sensorfläche von 780x780 auf 416x416 Pixel schien für alle getesteten Algorithmen keinen Einfluss auf die Erkennungsleistung zu haben. Das letzte Kapitel der Studie befasste sich noch mit Templatestandards. Hierzu sei nur ein Punkt angeführt. Änderungen in Templates führen auch immer zu Änderungen der Algorithmen. Ein Template, basie-



Quelle: BSI-Studie "BioFinger"
http://www.bsi.bund.de/literat/studien/BioFinger/BioFinger_I_I.pdf

rend auf dem kleinsten gemeinsamen Nenner, bringt zwar Interoperabilität aber gleichzeitig auch erhöhte Fehlerraten und geringere Performance.

Abschluss

Der BioFinger-Studie zufolge bietet die Technologie eine wirksame Verbesserung gegenüber dem Vergleich des Gesichts mit einem Passfoto durch einen Menschen. Die Studie selber war allerdings kaum das Papier wert, auf der sie ausgedruckt wurde. Viele Ergebnisse und Diagramme sind unkommentiert, die verwendeten Algorithmen wurden an drei Stellen in unterschiedlicher Reihenfolge und mit unterschiedlicher Bezeichnung verwendet. Alle Diagramme beginnen ab einer FAR von 0,1%. Kleinere Fehlerraten scheinen wohl nicht von Interesse zu sein.

Bei der Analyse zu biometrischen Zwillingen wurde erst von drei angepassten Algorithmen geschrieben dann tauchten aber sechs Diagramme auf, wobei das von Algorithmus 4 auch noch über einen weiten Bereich so konstante Ergebnisse aufwies, dass man einfach hätte stutzig werden müssen! Und zu allem Überfluss sind in den Diagrammen zum Einfluss der Templatealterung wohl auch noch die Kurvenfarben vertauscht worden. Zur Aussagekraft einer Studie mit solch einer Anzahl von Probanden kann sich jeder seine eigene Meinung bilden. Alles in allem eine ziemliche Zeit- und Geldverschwendung. Aber wir freuen uns einfach auf BioFinger II.

What The Hack is coming...

Rop Ginggrijp <rop@gonggri.jp>

As many of you probably know by now, 'What The Hack' is the name for this summer's edition of the congress / camping-trip / convention / festival / event that happens every four years in The Netherlands. Previous editions were called Hacking at the End of the Universe (1993), Hacking In Progress (1997) and Hackers At Large (2001).

We're calling upon everyone reading this to become involved, in one way or another. This is your chance to finally finish that really cool project you could show to everyone there. Or you could bring all your friends and be the initiator of a small 'village'. Organize things to happen there. Volunteer for some job, big or small, either in advance or when you're there. Send in an abstract of a talk or presentation you'd like to do. (And get in for free if the programme committee accepts it!).

This is likely to be a rather large event, and we'd like to show and experience the diversity of the various communities that make up the hacker world. We're trying to appeal to people that, simply put, become participants instead of 'The Audience'. There will be plenty of opportunities to find people that are into the same things you are. Talk, plan and maybe even get a whole new project on its feet. And then there's mass media attention for those that like it, as well as our own radio station for those that would rather roll their own. Add in some great conversations to be had and new friends to make. No reason to get paranoid or anything, but you are probably surrounded by people that first met at one of these events.

What The Hack happens from 28-31 of July 2005 near Den Bosch in The Netherlands. Tickets (when bought online before May 1st) are 120 Euros for 4 days of the event, but you can camp for over a week if you like (and help out a bit).

Much more on *http://www.whatthehack.org*

Windows Messages

Alexander Bernauer <alexander.bernauer@ulm.ccc.de>

Die meisten Hacker beschäftigen sich kaum mit Microsoft-Betriebssystemen. Gründe dafür gibt es genug, die meisten davon sind mehr als verständlich. Doch es gibt auf diesen Systemen viel zum Spielen, so dass es sich lohnen könnte, einen genaueren Blick darauf zu werfen. Ein Beispiel dafür ist das Design der grafischen Benutzeroberfläche. Mit den einfachsten Mitteln kann man hier Programme manipulieren, wobei erfahrungsgemäß die wenigsten Entwickler mit diesem Angriffsvektor rechnen.

Windows ist ein ereignisgesteuertes System. Ereignisse erzeugen Nachrichten, die vom System an das betreffende Fenster weitergeleitet werden. Unter Windows kann jeder Thread ein oder mehrere Fenster öffnen, wobei jedes Fenster einen systemweit eindeutigen `WindowHandle` besitzt. Ein Fenster kann der Vater eines zweiten Fensters sein, so dass die Menge aller Fenster einen Baum mit dem Desktop als Wurzel bilden. Jedes Fenster gehört zu einer Fensterklasse. Bestandteil einer Fensterklasse ist u.a. die Routine, die eingehende Nachrichten behandelt. Diese Routine wird üblicherweise `WindowProc()` genannt.

Wenn der Benutzer beispielsweie mit seiner Maus klickt, erzeugt der Maustreiber eine Fensternachricht und sendet diese an die so genannte „System Message Queue". Dieser FIFO wird zyklisch vom „System Dispatcher" ausgelesen. Er erkennt z.B. am Fokus, für welches Fenster die Nachricht bestimmt ist und sendet sie an die „Thread Message Queue" des für das Empfangsfenster zuständigen Threads. Ein in der Benutzerapplikation laufender „Thread Dispatcher" nimmt die Nachricht aus diesem FIFO entgegen und übergibt sie der passenden `WindowProc()` des Fensters. Zuvor wird eine Kopie der Nachricht an den „Translator" gegeben. Diese Zustandsmaschine erzeugt z.B. aus den Nachrichten „KeyDown" und „KeyUp" eine neue Nachricht „KeyPressed" und hängt sie hinten an die Thread Message Queue an. So kann eine oder mehrere Nachrichten ein Ereignis für eine neue Nachricht sein.

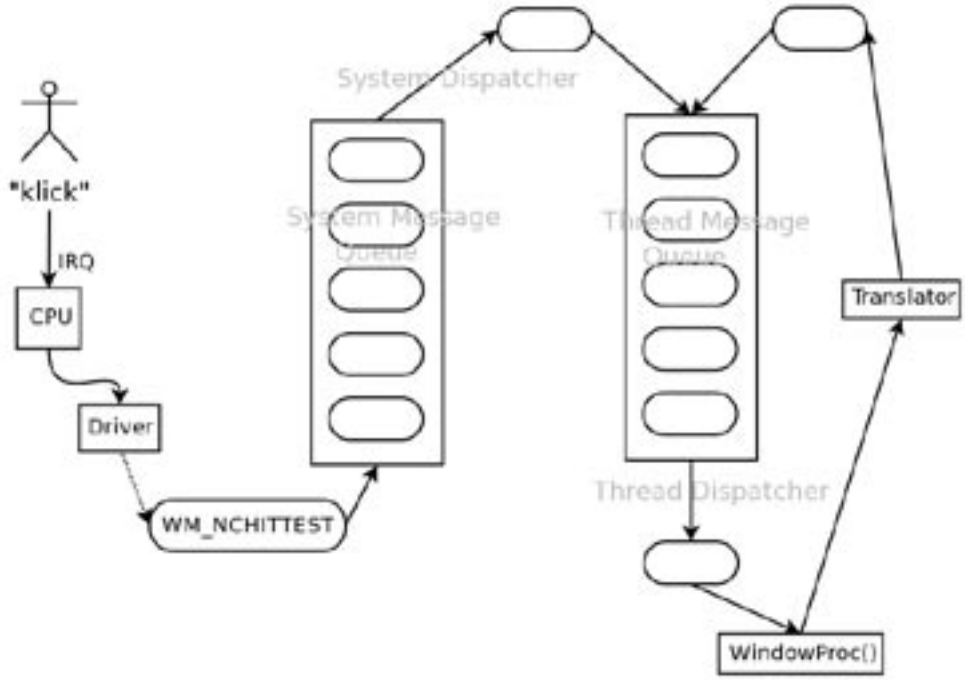
Es gibt eine ganze Reihe von Nachrichtenklassen, wobei zu jeder Nachrichtenklasse viele verschiedene Nachrichten gehören. So werden z.B. Nachrichten erzeugt, wenn ein Fenster verschoben, vergrößert, minimiert oder geschlossen wird. Eine andere Nachricht signalisiert einem Fenster, dass es sichtbar geworden ist und sich deshalb neu zeichnen soll. Wieder andere Nachrichten müssen beim Drücken von Buttons und Menüs oder beim Scrollen und Auswählen aus einer Liste behandelt werden. Damit sich kein Entwickler mit all diesen Details beschäftigen muss, gibt es eine sog. `DefaultWindowProc()` aus der Windows-Bibliothek, die alle Nachrichten (hoffentlich) richtig behandelt. Ein Entwickler behandelt in seiner `WindowProc()` nur noch die Nachrichten, die ihn gezielt interessieren, und gibt den Rest weiter an die `DefaultWindowProc()`. Deshalb reagieren viele Programme auf die meisten Nachrichten gleich.

Nachrichten können nicht nur vom System erzeugt werden, denn Fensternachrichten sind ein Teil des IPC-Konzepts von Windows. Beispielsweise kann ein Entwickler benutzerdefinierte Nachrichten erzeugen und versenden. Eine andere Anwendung sind komplexere Protokolle wie z.B. DDE und OLE, die zur Fernsteuerung von Programmen und Automation von Prozessen verwendet werden können.

Der Fehler an diesem Design ist, dass hier IPC im Push-Verfahren ohne Sicherheitssystem gemacht wird. Der Kommunikationspartner kann sich nicht gegen die ihm zugesandten Nachrichten wehren, vor allem, weil er nicht zuverlässig feststellen kann, wer der Absender ist. So kann jedes Programm Benutzerinteraktion simulieren, ohne dass das angegriffene Programm dies feststellen kann. Das heißt im Klartext: Wenn ein Angreifer es schafft, lokal seinen Code auszuführen, kann er damit alles tun, was der Benutzer auch tun kann, z.B. die Personal Firewall umkonfigurieren oder abschalten. Die wwwsh [1] nutzt das z.B., um einen Browser fernzusteuern und damit Informationen an einer Personal Firewall vorbei aus und in das System zu schleusen.

Diese Möglichkeiten genügen für die meisten Fälle bereits. So gibt es Programme, die einen OK-Button erst auf „enabled" setzen, wenn man den Registrierungsschlüssel eingegeben hat. Enabled heißt, dass ein Klick auf den Button eine Nachricht erzeugt, auf die die `WindowProc()` mit dem Aufruf des `OnClick()` Handlers reagiert. Generiert man diese Nachricht jedoch selbst und schickt sie dem Programm, so wird ebenfalls der `OnClick()`-Handler aufgerufen. Da dieser in den seltensten Fällen überprüft, ob der OK-Button überhaupt enabled ist, kann man so die Eingabe des Registrierungsschlüssels umgehen.

Für diejenigen, denen diese Möglichkeiten noch nicht genügen, bietet die Windows Bibliothek noch ein weiteres Feature: `MessageHooks`. Das sind Nachrichtenfilter, die man an beliebigen Stellen des Zustellsystems installieren kann. Damit kann man beliebige Nachrichten systemweit abfangen oder verfälschen. Lagert



man die Implementierung des Filters in eine DLL aus, kann man sogar die Thread Message Queue eines anderen Threads filtern ohne dafür besondere Rechte haben zu müssen. Damit existiert für jeden, der lokal Programme ausführen kann, ein mächtiger Angriffsvektor. Man ist der „Man in the Middle" in der Mensch-Maschine-Schnittstelle.

Ausnutzen kann man das z.B. beim Onlinebanking. Wenn der Benutzer die Kontonummer des Begünstigten eingibt, fängt der Nachrichtenfilter die entstehenden Nachrichten ab und ersetzt sie durch neue, so dass das Programm eine andere Kontonummer empfängt. Jetzt muss man nur noch verhindern, dass die eingeschleuste Kontonummer in der Eingabemaske der Applikation sichtbar wird. Doch glücklicherweise darf jedes Programm auf beliebige Stellen des Desktops malen ...

Normalerweise würde es einige Schwierigkeiten machen herauszufinden, welche Fenster eine Applikation öffnet und welche Nachrichten diese Fenster empfangen. Doch auch hier hilft Microsoft weiter und liefert u.a. mit seiner Entwicklungsumgebung Visual C++ den Spy++ mit. Mit diesem Programm kann man sich alle Prozesse, Threads und Fenster sowie die Beziehungen zwischen ihnen anzeigen lassen. Zusätzlich kann man alle Nachrichten für ein Fenster und dessen Kindfenster mitprotokollieren lassen.

In der MSDN-Dokumentation [2] findet man alle Details zum Windows-Nachrichtensystem. Es gibt verschiedene Nachrichtenklassen, wie z.B. die `General Window`-Klasse, deren Nachrichten mit dem Präfix „`WM_`" gekennzeichnet sind. So gibt es z.B. die Nachrichten `WM_SETTEXT` und `WM_GETTEXT` mit denen man den Inhalt eines Fensters schreiben und lesen kann. Zum Versenden einer Nachricht existieren die Funktionen `PostMessage()` und `SendMessage()`. `PostMessage()` hängt eine Nachricht an die Thread Message Queue an und kehrt zurück. `SendMessage()` ruft die `WindowProc()` direkt auf und blockiert, bis diese zurückkehrt. Mit beiden Funktionen kann man beliebige Fenster adressieren.

Es gibt verschiedene Filtertypen. Die Unterschiede liegen in der Stelle der Installation und dem zu filternden Nachrichtentyp. Zum Installieren von Nachrichtenfiltern gibt es die Funktion `SetWindowsHookEx()`. Falls mehrere Nachrichtenfilter des gleichen Typs an die gleiche Stelle des Zustellsystems installiert werden, bilden sie eine Filterkette. Jeder Filter kann die weitere Abarbeitung dieser Kette abbrechen lassen.

Fensternachrichten können sogar dazu dienen, eine Privilege-Escalation zu erreichen, wenn es einen priviligierten Prozess gibt, der ein Fenster öffnet. Dazu kann man z.B. die Fensternachricht `WM_TIMER` verwenden. Der Zweck dieser Nachricht ist es eigentlich, dass sich ein Programm diese regelmäßig vom System schicken lässt, um synchronisiert etwas zu tun. Deshalb besitzt die `WM_TIMER` als Parameter einen Zeiger auf die „Callback-Funktion", die das Programm beim Installieren des Timers angegeben hat. Wenn man jetzt eine `WM_TIMER` selbst erstellt und sie einem Programm schickt, dann stehen die Chancen gut, dass die `DefaultWindowProc()` richtig darauf reagiert und die Callback-Funktion anspringt. Wenn man jetzt zuvor mit Hilfe einer `WM_SETTEXT` in ein beliebiges Fenster der Applikation seinen Code hineinschreibt, kann man den Zeiger der `WM_TIMER` so setzen, dass dieser Code vom Programm ausgeführt wird. Das heißt, sobald irgendein priviligierter Prozess ein Fenster öffnet, kann jeder Angreifer beliebigen Code mit priviligierten Rechten ausführen. Der einzige Haken an diesem Angriff ist, dass man die Adresse des eingeschleusten Codes herausfinden muss um den Zeiger richtig setzen zu können. Dazu muss man das Programm debuggen und herumexperimentieren. Im Prinzip handelt es sich dabei um das selbe Problem, dass man bei der Ausnutzung eines Buffer Overflows hat.

Links

[1] *http://copton.net/vortraege/pfw/index.html*
[2] *http://msdn.microsoft.com/*

Personal Firewalls

Ansgar Wiechers <ansgar.wiechers@ulm.ccc.de>
Alexander Bernauer <alexander.bernauer@ulm.ccc.de>

Der CCC Ulm und der Chaostreff Bad Waldsee haben alle gängigen Personal Firewalls daraufhin getestet, ob das versprochene Mehr an Sicherheit erreicht wird. Das Ergebnis verwundert einen Experten nicht, ist aber für den Laien erstaunlich.

Die Arena

Personal Firewalls gehören zu den Mitteln, mit denen der moderne Windows-User den zahlreichen Gefahren des Internets (beispielsweise Würmer, Spyware oder trojanische Pferde) zu begegnen gedenkt. Es handelt sich dabei um Programmpakete, die eine Reihe von Funktionen bündeln, um dadurch sowohl eingehenden wie auch ausgehenden Datenverkehr gezielt zulassen oder unterbinden zu können. Die wesentlichen Funktionen sind:

- Filtern auf Adressen/Ports: Abhängig von IP-Adresse und Port werden Pakete zugelassen oder verworfen.

- Filtern auf Anwendungen: Abhängig von Pfad und/oder Prüfsumme (das wird bei den einzelnen Personal Firewalls unterschiedlich gehandhabt), dürfen Anwendungen Verbindungen nach außen aufbauen oder Verbindungen von außen annehmen.

Zusätzlich versuchen manche Personal Firewalls, das Starten von Programmen durch andere Programme zu kontrollieren. Dazu werden anscheinend Aufrufe von `CreateProcess()` bzw. `WinExec()` abgefangen. Daneben enthalten viele Personal Firewalls weitere Features, die den Benutzer schützen sollen. Dazu gehört beispielsweise das Droppen von ICMP-Paketen, um den Rechner vor Angreifern zu verstecken, oder das Sperren der Kommunikation mit einer IP-Adresse, wenn Pakete mit dieser Source-IP als Angriff erkannt wurden.

Die Kandidaten

Wir haben insgesamt sieben Personal Firewalls in Bezug auf ihre Funktionsfähigkeit und ihren Spaßfaktor für Angreifer untersucht. Die Ergebnisse wurden im Dezember bei einem ChaosSeminar in Ulm [1] gezeigt.

Getestet wurden:
- Kerio Personal Firewall 4.1.2
- Norman Personal Firewall 1.42
- Agnitum Outpost Firewall Pro 2.5
- Sygate Personal Firewall Pro 5.5
- Tiny Firewall 6.0
- Zone Labs ZoneAlarm Pro 5.5
- Symantec Norton Personal Firewall 2005

Zur Realisierung der genannten Funktionen haben die meisten Personal Firewalls einen dreigeteilten Aufbau:
- einen oder mehrere Kernel-Treiber zum Filtern im Kernelspace
- einen oder mehrere Dienste zur Steuerung der Treiber (laufen mit SYSTEM-Rechten)
- GUI zur Konfiguration (läuft mit Userrechten)

Das GUI lässt sich bei jeder Personal Firewall durch ein Passwort schützen, so dass nicht unbefugt Änderungen vorgenommen werden können. Leider ist dieser Schutz bei keinem der Programme voreingestellt. Die Anzahl der installierten Treiber und Dienste variiert sehr stark zwischen den Personal Firewalls. Um einen Überblick über die installierten Dateien, Treiber und Dienste zu erhalten, wurde die

Installation der einzelnen Personal Firewalls mit InCtrl5 [15] protokolliert.

	Kerio	Norman	Outpost	Sygate	Tiny	Zone Alarm	Norton
Reg-Schlüssel	477	64	315	277	1694	189	3556
Reg-Werte	705	246	1036	743	2283	499	5934
Verzeichnisse	12	23	15	8	32	8	34
Dateien	61	109	222	61	382	74	417
Treiber	1	2	2+n	5	8	1	8
Dienste	1	1	1	1	5	1	8

Der Eintrag „2+n" bei den Outpost-Treibern bedeutet, dass neben den Basistreibern eine variable Anzahl von Modulen für Filteraufgaben installiert wird.

Für alle Tests wurden die Default-Einstellungen verwendet, da wir bei Sicherheitssoftware (und solcher, die es werden will) voraussetzen, dass sie mit sicheren Defaults installiert wird.

Funktionalität mangelhaft: incoming

Was den Schutz nach Außen betrifft, geht eine Personal Firewall den Weg des „Paketfilterns". Der Angriffsvektor ist dabei immer ein Dienst. Wenn dieser anfällig gegen gezielte Protokollverletzungen (wie z.B. Buffer Overflows) ist, kann sich ein Angreifer damit Zugang zum System verschaffen. Die Personal Firewall soll das verhindern, in dem sie den eingehenden Verkehr scannt und u. A. Verbindungsversuche zu diesem Dienst unterbindet. Da jede Personal Firewall Software ist, und Software nun mal Fehler haben kann, weil sie von Menschen gemacht wird, hat man das Risiko damit effektiv nur verlagert. Jetzt ist es die Personal Firewall, die angegriffen werden kann. Wenn man den nicht benötigten Dienst aber einfach abschaltet, besteht dieser Angriffsvektor nicht mehr. Das Betriebssystem wirft die Pakete weg, weil auf dem angegebenen Port kein Dienst lauscht. Der einzige Angriffsvektor bleibt hierbei das Betriebssystem. Und wenn das einen Fehler im Protokoll Stack hat, dann hilft auch keine Personal Firewall mehr.

Eigentlich sollte Microsoft seine Betriebssysteme ohne aktivierte Dienste ausliefern. Aus unbekannten Gründen ist dies aber nicht der Fall, so dass sich Würmer (wie z.B. Sasser) verbreiten konnten. Man kann aber manuell nachhelfen. Hierzu gibt es entweder ein Batch-Skript von Torsten Mann [11] oder ein GUI-Programm von Volker Birk [12], die beide das selbe tun: Sie machen Änderungen in der Windows Registry, so dass die Dienste beim Booten nicht gestartet werden.

Die Funktionalität des Packetfilters für eingehenden Verkehr ist also für die meisten Anwender nutzlos, weil es bessere und einfachere Möglichkeiten gibt. Was die Kontrolle des ausgehenden Datenverkehrs betrifft, haben wir gezeigt, dass keine Personal Firewall ihr Versprechen halten kann.

Funktionalität mangelhaft: outgoing

Wenn ein Programm versucht, Daten ins Internet zu senden, kontrolliert der Applikationsfilter, ob es das darf. Wurde das noch nicht bestimmt, wird üblicherweise der Benutzer gefragt. Das Windows-Nachrichtensystem erlaubt es aber jedem Programm, beliebe Benutzeraktionen zu simulieren. Der Autoklicker [13] nützt das aus, in dem er im Hintergrund wartet, bis sich das Fenster der Personal Firewall öffnet, um dann schnell die nötigen „Sicherheitseinstellungen" vorzunehmen und auf OK zu „klicken". Effektiv bedeutet das, dass so bald der Benutzer gefragt wird, jedes beliebige Programm ins Internet verbinden darf. Es bedeutet auch, dass eine Personal Firewall effektiv nicht vorhanden ist, so bald sie der Benutzer ohne Passwort konfigurieren darf.

Doch es kommt noch schlimmer. Es war uns mit allen Personal Firewalls möglich, mit Hilfe der wwwsh [13] einen Rechner remote zu kontrollieren. Die wwwsh ist ein trojanisches Pferd, das mit Hilfe von Fensternachrichten einen Browser fernsteuert und somit Daten an der Personal

Firewall vorbei in und aus dem System schleusen kann. Die Proof-of-Concept-Implementierung verwendet ein sichtbares Browserfenster, damit man sehen kann, was passiert. Das Eintreffen der Nachricht zum Erneuern des Fensterinhalts beim Browserfenster kann aber mit Hilfe von MessageHooks verhindert werden. Es bleibt somit unsichtbar. Mehr dazu steht im Artikel „Windows Nachrichtensystem" in dieser Datenschleuder.

Wie oben bereits erwähnt, verhindern manche Personal Firewalls das Starten anderer Applikationen durch die Filterung von Bibliotheksaufrufen. Das tun sie, weil schon mehrere Methoden bekannt wurden, wie man sie mit Hilfe von Wirtsprogrammen umgehen kann. Die Filterung der Bibliotheksaufrufe bietet aber keinen Schutz. So verwendet die wwwsh den „Start->Ausführen" Dialog um dort „iexplore.exe" einzutragen und auf OK zu „klicken".

Die Informationen gehen base64 kodiert als Teil von URLs aus dem Zielsystem raus. Die Antwort kommt als HTML-Dokument mit einem Meta-Refresh auf eine neue URL, die wiederum base64 kodierte Informationen enthält. Der Browser surft die neue Seite an und zeigt die URL in der GUI, so dass die wwwsh sie auslesen kann und somit an die Informationen kommt. Das Funktionsprinzip der wwwsh ist es, regelmäßig eine URL zu pollen, bis sie im Antwortteil einen Befehl empfängt. Dieser Befehl wird lokal ausgeführt und dessen Standardausgabe in die URL kodiert und zurück gesendet. Damit umgeht man sämtliche Firewalls, weil eigentlich alle http auf Port 80 zu lassen.

Jeder Windows-Rechner, auf dem der lokale Benutzer mit einem Browser surfen darf, kann remote kontrolliert werden. Egal, ob eine Personal Firewall läuft, oder nicht. Alle Versprechen von Herstellern von Sicherheitssoftware sind falsch. Es bleibt dem Benutzer nichts anderes übrig, als aufzupassen, welche Software er lokal ausführt. Wie er das am besten macht, war Thema des letzten ChaosSeminars in Ulm [14].

Unsichtbar, unverwundbar?

Zu guter letzt bleiben noch Kleinigkeiten wie das Verstecken des Rechners durch ICMP Blocking. Jeder, sich mit TCP/IP auskennt weiß, dass es technisch nicht möglich ist, einen Teilnehmer in einem IP Netzwerk zu verstecken. Falls der Teilnehmer nicht existieren würde, müsste bei einem „ICMP Ping" der Router davor ein „ICMP Host unreachable" schicken. Wenn keine Antwort kommt, dann deutet das auf eine Firewall hin, die filtert. Damit weiß man auch, dass das Ziel existiert.

Keine der versprochenen Features von Personal Firewalls haben also einen Nutzen für den normalen Benutzer der Zielgruppe. Doch es kommt **noch** schlimmer. Manches Feature bie-

tet zusätzliche Angriffsvektoren. Es gibt also Angriffe, die sind nicht möglich, obwohl eine Personal Firewall installiert ist, sondern **weil** ein Personal Firewall installiert ist.

Zusätzliche Angriffsvektoren

Viele Personal Firewalls bieten sogenannte Intrusion Detection Systeme (IDS) an. Dabei handelt es sich um eine Heuristik, die versucht, Angriffe zu erkennen. Von unseren Testkandidaten boten Outpost, Sygate, ZoneAlarm und Norton dieses Feature. Die Art der Heuristik schwankt sehr von Hersteller zu Hersteller. Bei der Norton Personal Firewall beispielsweise zählt bereits ein IP-Paket, das auf einem bekanntermaßen von trojanischen Pferden verwendeten Port [2] ankommt, als „Hacker Angriff". Derartige „Angriffe" führen im Normalfall dazu, dass der gesamte ein- und ausgehende Datenverkehr zur Source-IP dieses Pakets temporär gesperrt wird. Die Sperrdauer ist bei allen oben genannten Programmen konfigurierbar, wobei der Defaultwert bis zu 30 Minuten (bei Norton) beträgt.

Da die Source-Address trivial zu spoofen ist und längst nicht jeder Router Pakete mit gespoofter Source-Address verwirft, bieten sich hier interessante Möglichkeiten, Anwender von Personal Firewalls zu Ärgern. Beispielsweise könnte man die IP-Adressen der Nameserver eines Netblock-Owners verwenden, um so einem User gezielt den Zugriff auf die Nameserver zu entziehen. So haben sie für eine gewisse Zeit nur noch per IP-Adresse Internet, was effektiv einem Denial Of Service für die meisten Benutzer gleich kommt.

Ein weiteres Feature, das manche Personal Firewalls bieten, ist die Überprüfung des Datenverkehrs auf persönliche Daten wie beispielsweise PINs, Passwörter oder Telefonnummern. Dazu hinterlegt man diese Daten in einem zentralen Formular der Personal Firewall, die daraufhin ausgehenden Datenverkehr auf diese Muster prüft und ggf. sperrt. Kerio, ZoneAlarm und Norton implementieren dieses Feature. Im Allgemeinen ist dieses Feature äußerst fragwürdig, denn z.B. bei https Verbindungen kann die Personal Firewall den Verkehr gar nicht scannen.

Es genügen aber auch einfache Kodierungen wie rot13 oder base64, um die Personal Firewall zu umgehen. Ein Angreifer wird also immer einen Weg finden, die persönlichen Daten rauszuschleusen. Doch leider ist das Feature nicht nur nutzlos, sondern bringt eine zusätzliche Gefahr: ein Angreifer hat eine zentrale Anlaufstelle, um Kreditkartennummern und Passwörter abzugreifen. Mit Hilfe von Fensternachrichten kann das Programm des Angreifers sich einfach zum Eingabeformular der Personal Firewall „durchklicken" und dort die Angaben auslesen. Das geht nur bei ZoneAlarm nicht, da dort die Daten nur als Sternchen erscheinen. Doch auch hier kann ein Angreifer z.B. die PIN herausfinden. Er lockt das Opfer auf eine Webseite, wo automatisiert URLs geladen werden, die alle vierstelligen Zahlen enthalten. Diejenige URL, die nicht angesurft wird, ist die mit der PIN des Benutzers.

Da alle Personal Firewalls Dienste installieren, die mit SYSTEM-Rechten laufen, bieten sie potenziell Angriffspunkte für Privilege Elevation Attacks. Dienste sollen gemäß den Vorgaben von Microsoft keine Fenster öffnen [3], aber manche Personal Firewall (Outpost, Sygate, Tiny) ignoriert diese Vorgabe. Welches Programm privilegierte Prozesse mit Fenstern startet, lässt sich z.B. mit dem Visual Studio-Tool Spy++ herausfinden, mit dem man sich die Eigenschaften von Prozessen und Fenstern anzeigen lassen kann.

Fenster ermöglichen unter Umständen sog. Shatter Attacks, bei denen man Code in das Fenster schreibt und diesen anschließend über eine Fensternachricht mit geeigneter Callback-Funktion zur Ausführung bringt. Die ursprüngliche, von Foon beschriebene, Shatter Attack [4] hat dazu per WM_PASTE den Shellcode in ein Edit Control geschrieben und diesen Code dann über eine WM_TIMER zur Ausführung gebracht. Zwischenzeitlich wurden noch weitere Methoden beschrieben, um auf diesem Weg Code zur Ausführung zu bringen [5]. Aus Zeitgründen konnten wir allerdings noch nicht weiter untersuchen, welche Personal Firewall tatsächlich für Shatter Attacks anfällig ist. Aber auch ohne Shatter Attacks lassen sich manche Personal Firewalls erfolgreich exploiten. Beispielsweise wurde im Dezember 2004 ein Sicherheitsproblem mit der LiveUpdate-Funktion von Symantec-Produkten gemeldet [6]. Über den Benachrichtigungs-Dialog, der im Fall neuer Updates angezeigt wird, konnte sich ein lokaler Nutzer erhöhte Rechte verschaffen. Aber auch andere Personal Firewalls haben mit der Rechtetrennung so ihre Probleme.

Outpost hat ein modulares Filterkonzept. Über einen Dialog kann man Module hinzufügen, die dann zusätzliche Filterfunktionen ausüben können. Anfang 2004 wurde entdeckt, dass dieser Dialog mit SYSTEM-Rechten läuft, so dass man über diesen Dialog eine cmd.exe mit SYSTEM-Rechten starten konnte [7]. Interessant auch die (nicht verifizierte) Stellungnahme von Agnitum zum Problem [8].

Dadurch, dass Personal Firewalls jedes eingehende Paket entgegennehmen und anschauen müssen, entstehen weitere potenzielle Sicherheitslücken, denn der Code, der die Pakete untersucht, kann natürlich Fehler enthalten. Dass dieses Bedrohungsszenario sehr real ist, zeigte sich im März 2004, als der Wurm W32/Witty.worm Systeme angriff, die durch Produkte des Herstellers Internet Security Systems (ISS) „geschützt" waren [9]. Der Wurm nutzte einen Buffer Overflow in einem Analyse-Modul für das ICQ-Protokoll aus. Da dieser Wurm ausschließlich Systeme befiel, die durch eine Personal Firewall „geschützt" waren, kann er als Proof-of-Concept für die Erhöhung der Angriffsfläche durch Personal Firewalls betrachtet werden.

Ärger mit Personal Firewalls

Dadurch, dass eine Personal Firewall ein- und ausgehenden Datenverkehr untersuchen muss, belastet sie natürlich auch noch die Ressourcen des Rechners. Downloadzeiten werden verlängert, die Auslastung des Systems steigt. Das führt so weit, dass beim Betrieb von ZoneAlarm in der Einstellung „unsicheres Netzwerk" ein 400 MHz Pentium II vollständig unbenutzbar wurde. Selbst auf einem 2,3 GHz Athlon kam es zu sekundenlangen Aussetzern, während derer keine Interaktion mit Anwendungen möglich war.

Außerdem nerven Personal Firewalls mit erfolgreichen Abwehrmaßnahmen, denen gar kein Angriff vorausging. Wenn man z.B. von seinem ISP bei der Einwahl eine dynamische Adresse bekommt und diese zuvor einem Teilnehmer gehörte, der einen eDonkey-Client betreibt, so bekommt man fälschlicherweise Benachrichtigungen über zur Verfügung stehende Shares. Die Personal Firewall wertet diesen Verbindungsaufbau als versuchten Angriff und meldet das dem Benutzer. Es gibt noch weitere Fälle, in denen Personal Firewalls fälschlich Alarm schlagen. Ein Auflistung gibt es z.B. hier [10].

Im Allgemeinen machen Personal Firewalls Probleme, weil sie „am Windows herum patchen" und somit sein Verhalten verändern. Da man unter Windows üblicherweise Nachrichten wie Ereignisse behandelt, haben es viele Programme mit einem Haufen Race Conditions zu tun. Man hofft einfach, dass die Nachrichten in der üblichen Reihenfolge und innerhalb der üblichen Zeitfenster ankommen. Das ist aber nicht garantiert. Veränderungen am System durch die Personal Firewall können deshalb dazu führen, dass manche Programme nicht mehr funktionieren. Und je mehr die Hersteller von Personal Firewalls versuchen, Windows mit Flicken dicht zu bekommen, desto mehr Programme steigen aus.

Der Meinungsteil

Zum Abschluss noch ein besonderer Leckerbissen: Norman. Dieses Stück Bitschrott ist derart broken, dass man es nur dann konfigurierbar installieren kann, wenn Windows auf einer einzigen Partition installiert ist. Hat man daran etwas geändert (z.B. %ProgramFiles% auf eine andere Partition gelegt), so werden Änderungen der Konfiguration mit einer Fehlermeldung quittiert. Bei einem Portscan öffnet Norman einen Dialog, den man mit vier Mausklicks bestätigen muss. Für jeden einzelnen Port.

Aber auch andere Personal Firewalls haben nette Eigenheiten. Verbietet man beispielsweise in Tiny dem Programm mozilla.exe die Kommunikation mit dem Internet, so braucht man es nur in ein anderes Verzeichnis zu kopieren, schon funktioniert die Kommunikation wieder. Darüber hinaus implementiert Tiny ein eigenes Rechtesystem für Zugriffe auf Dateien und Registry-Einträge, das komplett am regulären Rechtesystem vorbei funktioniert. Tiny ist die einzige getestete Personal Firewall, bei der in der Default-Einstellung Ports von außen erreichbar waren (u.a. 135/tcp und 445/tcp). Sehr nett ist auch ZoneAlarm, das seine Konfiguration in world-writable Dateien speichert, die es dadurch schützt, dass der Dienst diese Dateien beim Start öffnet und lockt.

[1] *http://ulm.ccc.de/chaos-seminar/personal-firewalls/index.html*

[2] *http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html*

[3] *http://support.microsoft.com/default.aspx?scid=kb;en-us;327618*

[4] *http://security.tombom.co.uk/shatter.html*

[5] *http://www.security-assessment.com/Papers/Shattering_By_Example-V1_03102003.pdf*

[6] *http://www.sarc.com/avcenter/security/Content/2004.12.13a.html*

[7] *http://www.heise.de/newsticker/meldung/43763*

[8] *http://groups.google.de/groups?hl=de&lr=&selm=bujt42%24io0%241%40newsreader2.netcologne.de*

[9] *http://www.heise.de/newsticker/meldung/45876*

[10] *http://www.dslreports.com/forum/remark,2169468*

[11] *http://www.ntsvcfg.de/*

[12] *http://www.dingens.org/*

[13] *http://copton.net/vortraege/pfw/index.html*

[14] *http://ulm.ccc.de/chaos-seminar/windows-security/recording.html*

[15] *http://www.pcmag.com/article2/0,4149,9882,00.asp*

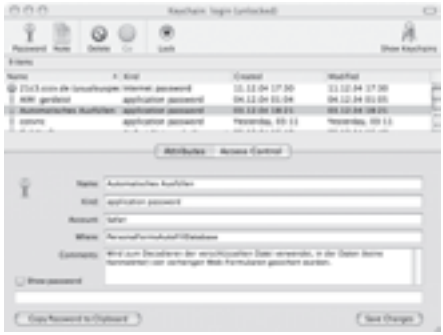
Mac OS X Keychain Hacking

Angelo Laub <al@rechenknecht.net>

Die Mac OS X Keychain bietet einen vermeintlich sicheren Weg, persönliche Passwörter verschlüsselt aufzubewahren. Allerdings kann man die Passwörter auch ohne Kenntnis des Master-Passwortes ohne weiteres auslesen.

Sicherheitskonzept der Keychain

Häufig wird im System und in Anwendungsprogrammen nach einer Möglichkeit gesucht, Passwörter sicher zwischenzuspeichern oder für den dauerhaften Gebrauch abzulegen. Damit nicht in jeder Anwendung eigene Routinen zum Speichern von Passwörtern implementiert werden müssen, bietet Apple zu diesem Zweck die Keychain. Apples Browser Safari legt sogar den Inhalt von Webforms mitsamt Passwörtern (z. B. Onlinebanking) hier ab. Bei der Benutzung von Apple Mail kommt man um die Nutzung der Keychain ebenfalls nicht herum, will man nicht bei jedem Emailabruf sein IMAP- oder POP-Passwort eintippen. Gilt eine Applikation für die Keychain erst einmal als vertrauenswürdig, so kann sie ohne weitere Abfrage auf alle ihr zugeschriebenen Passwörter zugreifen. Falls sich die Prüfsumme des Binarys einer vertrauenswürdigen Applikation ändert, merkt dies die Keychain, was einen Angriffsversuch durch direkte Manipulation am Binary vereitelt.



Exploitstrategie

Um dennoch an die Passwörter zu kommen, kann zur Laufzeit Exploitcode in eine vertrauenswürdige Applikation injeziert werden, was das Binary unverändert lässt. Dies geht bei Cocoa-Applikationen beispielsweise mit Hilfe des InputManagers [1] oder der Mach Injection-Technik [2]. Mach Injection erlaubt es, beliebigen Code zur Laufzeit in andere Prozesse zu injezieren und diesen in einem Thread auszuführen. Die Exploitstrategie dürfte nun klar sein. Man baut sich mit Mach Injection ein Objective-C-Bundle mit Code, der die Keychainpasswörter ausliest und injeziert ihn anschliessend in eine vertrauenswürdige Applikation. Dies kann man mit jeder Applikation machen, um an die ihr zugeordneten Passwörter zu kommen. Die dafür verwendbare Keychain-API ist in der Apple-Dokumentation [3] beschrieben.

Sicherer Einsatz der Keychain?

Derzeit ist es einem Angreifer möglich, alle Passwörter des momentan eingeloggten Users auszulesen, die einer vertrauenswürdigen Applikation zugeordnet sind. Erlangt der Angreifer Root-Rechte auf dem Mac (was z.B. durch den iSink Exploit [4] möglich ist), so kann er sogar auf die Passwörter aller User zugreifen. Es genügt dabei, eine Shell

auf dem System zu haben, man benötigt dafür keinen direkten Hardwarezugriff. Nach Einschätzung des Autors wird Apple diese Lücke nicht schliessen können, da sie tief im Grundkonzept der Keychain verwurzelt ist. Man müsste ausschliessen können, dass Code zur Laufzeit in das Programm nachgeladen wurde. Betrachtet man die Fülle an Injektionsmethoden, so erscheint dies fast unmöglich. Ausser InputManager und Mach Injection wären das beispielsweise noch der dynamische Linker mit DYLD_INSERT_LIBRARIES und pthread zu nennen. Die einzige Möglichkeit wäre, das Prinzip der vertrauenswürdigen Applikationen aufzugeben und bei jedem Zugriff auf die Keychain das Masterpasswort zu verlangen. Dies würde jedoch die sinnvolle Verwendung der Keychain in vielen Applikationen wie z. B. Safari unpraktikabel werden lassen. Einen Proof-of-Concept-Exploit, der Code in Safari injeziert um alle Webpasswörter auszulesen, hat der Autor zusammen mit einem Bugreport im Januar 2005 an Apple geschickt - bislang keine Reaktion.

Derzeit kann die Keychain nur für selbstangelegte Notizen, die nicht mit einer vertrauenswürdigen Applikation assoziiert sind, als sicher gelten.

Links

[1] *http://developer.apple.com/documentation/Cocoa/Conceptual/InputManager*

[2] *http://rentzsch.com/mach_inject/*

[3] *http://developer.apple.com/documentation/Security/Reference/keychainservices/*

[4] *http://www.k-otik.com/exploits/20050123.fm-iSink.c.php*

Von Elstern, Coalas und anderen Raubtieren

von Neko <neko@greenie.muc.de>

Hashimemashite, watashi wa neko desu... was so viel heisst wie: Ich darf mich vorstellen, mein Name ist Katze... und ich jage Vögel. Besonders, wenn sie mir zu nahe kommen oder gar aufdringlich werden.

Und genau das hat dieses Jahr eine Elster versucht. Elster steht für *El*ektronische *Ste*uer*r*abgabe (wie passend...), und unser geschätztes deutsches Finanzamt erwartet dies jetzt auch für Umsatzsteuervoranmeldungen: eine elektronische Abgabe. [1]

Bisher bin ich mit USt*V*A ganz gut gefahren. Ich hatte meine kleine Buchhaltung mit ihren 20 Buchungen pro Monat, schlicht erfasst in ein paar Textfiles. Dazu kam ein kleines Programm, welches am Monatsende alle Files ausgelesen, miteinander verrechnet und ein Postscriptfile ausgespuckt hat. Dieses habe ich dann an meinen Drucker verfüttert, in dem schon ein Vordruck der Steuerunterlagen lag. Noch unterschreiben, in einen Briefumschlag und ab damit zum Finanzamt.

Zugegeben, das Ergebnis war gerne mal krumm und schief, weil der Drucker das Papier nie ganz gerade eingezogen hat - aber es war lesbar, lief unter Linux, und ich konnte meine Buchhaltung halten, wie es mir passt.

Jetzt kommt die Elster daher und besteht darauf, dass ich mein Papier in Zukunft elektronisch abgeben soll. Fairerweise sei hier erwähnt, dass Elster eigentlich nur der Name des freien Windowsprogrammes ist, welches zu diesem Zweck vom Finanzamt zur Verfügung gestellt wird. Im allgemeinen Sprachgebrauch wird aber derzeit kaum ein Unterschied gemacht.

Das hübsche freie Programm vom Finanzamt hat allerdings einen kleinen Haken: Ich muss mir einen Windowsrechner zulegen, diesen ans Internet anschliessen (Sicherheit ade) und dort die Software installieren. Dazu kommt, dass meine Buchhaltung jetzt entweder auf die Windose wandern, oder ich per Hand Zahlen umkopieren muss. Oder andere Netzwerkspielchen anstehen.

Es tut mir schrecklich leid, aber für mich lohnt sich dieser Aufwand nicht bzw. ich sehe ihn nicht ein. Also bleibt nur noch eines: die Flucht nach vorn antreten und selber schreiben. Ich schreibe seit über 10 Jahren Programme, die Abläufe automatisieren, Daten auswerten, kontrollieren und irgendwo anders wieder ablegen. Da liegt dieser Ansatz nahe.

Die Formalitäten

So schwer kann das ja nicht sein - Daten in ein bestimmtes Format einpacken und per Internet an einen anderen Server liefern. Danach die Antwort einsammeln und auswerten.

Der Rest der Vorgeschichte ist schnell erzählt: Recherchen (*http://www.elster.de/* [2]) ergaben, dass man sich eine EntwicklerID holen kann. Zusätzlich bekommt man Zugriff auf einen EntwicklerKIT mit Dokumentation und einem Testserver. Wenn man dann etwas entwickelt und ausgiebig getestet hat, kann man für

dieses Teil eine HerstellerID beantragen. Diese ist an die Person und das Produkt geknüpft.

Die EntwicklerID war nach wenigen Tagen da. (Webformular ausfüllen und abwarten)

Die Bedingungen für UStVA lesen sich auf den ersten Blick einfach: etwas ZIP, ein wenig TripleDES und ein Hauch RSA. Dazu ein Schuss XML und ein Port 80 beim Finanzamt zum Einliefern. Dazu noch irgendwo etwas, was sich PKCS #7 nennt. Die freie, plattformunabhängige Entwicklungsschnittstelle heisst COALA.

Im EntwicklerKIT sind vorhanden: Einige PDFs und ein paar Java-Klassen (natürlich ohne Source-Code) sowie ein paar XML-Beispielvorlagen, Schemafiles und Stylesheets. Nur der Erwähnung wert: Ein RSA-Zertifikat.

Soweit auf den ersten Blick noch recht hübsch. In Anbetracht der Tatsache, dass unsere Ämter noch nicht viel Erfahrung mit freier Software-Entwicklung haben, fand ich das richtig lobenswert. Dass zu den Java-Klassen kein Code vorhanden war, war mir egal, ich hatte eh nicht vor, diese zu verwenden.

Beim genaueren Hinschauen fielen mir allerdings Mängel auf: Die Dokumentation ist an einigen Stellen etwas programmiererunfreundlich. Soll heissen: zuweilen wird lediglich auf die RFC verwiesen, statt kurz die verwendeten Teile des Verfahrens zu beschreiben. Besonders verhängnisvoll ist das im Fall von PKCS #7.

Allgemein gesprochen: Es werden alle Daten in ein XML-Format gepackt. Ein Teil dieses XML-Dokumentes wird dann per TripleDES verschlüsselt. Den Schlüssel hierfür darf man sich frei wählen. Danach wird der Schlüssel mit dem Public Key des Finanzamtes RSA-verschlüsselt und im noch unverschlüsselten Teil des XML-Dokuments abgelegt.

Dazu kommt noch eine ZIP-Kompression, aber die zählt nicht als Verschlüsselung. Das Ganze wird zwischendurch noch ein paar Mal in etwas base64 gewickelt.

Das Ergebnis ist ein XML-Dokument, in dem ein Teil, der seinerseits wieder XML enthält, verschlüsselt eingebettet ist. Ausgepackt und entschlüsselt ergibt sich wieder ein komplettes XML-Dokument.

Eine zusätzliche Signatur entfällt bei den Voranmeldungen. Das macht die Entwicklung einfacher.

Hindernisse

Leider entpuppte sich das bisschen ZIP und PKCS #7 als nicht ganz so einfach. Im Entwicklerforum gibt es eine Webseite mit Beispielen, wie ein Datensatz nach den einzelnen Schritten auszusehen hat. Leider konnte ich dies schon mit gzip nicht reproduzieren (da half auch kein base64), und alle PKCS #7 Versuche erstickten im Ansatz...

Deswegen beschränkte ich mich fürs Erste auf die mitgelieferten Java-Klassen. Schliesslich brauchte ich schnell eine lauffähige Lösung. Der nächste Stichtag für die Anmeldung kam langsam gefährlich nahe und da lauerte noch ein Hindernis bürokatischer Natur.

Das Verfahren für Voranmeldungen sieht grob wie folgt aus: Einreichung der Voranmeldung beim Finanzamt, das Finanzamt zieht den genannten Endbetrag vom Konto des Besitzers der Steuernummer ein... das Konto ist überzogen, die Bank lässt (im besten Fall) den Betrag zurückgehen, das Finanzamt springt im Dreieck... und der Besitzer darf wechselweise seiner Bank und dem Finanzamt erklären, dass er unschuldig ist.

Wer hindert mich jetzt daran, die UStVA für die ungeliebte Konkurrenz abzugeben? Oder vielleicht Siemens? BMW? Microsoft Deutschland? Alles, was ich benötige, ist eine Steuernummer. Die gehört ins Impressum (alternativ eine UstID) auf die Webseite. [3]

Das Finanzamt prüft solche Kleinigkeiten nicht. Schliesslich kann man ja eine berichtigte Voranmeldung abgeben.

Wo ist das Problem? Das Problem ist, dass man sich in keinster Weise authentisieren muss, um eine Voranmeldung abzugeben. Man kann dies mit jedem handelsüblichen Programm tun, welches UStVA unterstützt. Man kann jede Steuernummer verwenden, derer man habhaft wird - oder versuchen, ein paar zu erraten.

Hier hat das Finanzamt wohl am falschen Ende gespart. Wahrscheinlich - und das ist pure Spekulation von mir - hatte man keine Lust, zur Verwaltung der Steuernummern auch noch eine Schlüsselverwaltung hinzuzufügen. Ein normales Einmalpasswort-Verfahren, wie es Banken seit langem verwenden (Pin und TAN), wäre sicher schon hilfreich.

Man kann also nicht mehr verhindern, dass der Benutzer sich bei der Angabe seiner Steuernummer nicht aus Versehen vertippt.

Noch reicht das aber nicht: um seine eigene Anmeldung elektronisch einzureichen, fehlt eine offizielle HerstellerID. Um die zu erhalten, muss man ein Produkt erstellen – idealerweise etwas Vorzeigbares. Das habe ich getan (Oberfläche per Web, Demo Server aufgesetzt) und die URL mit meinem Antrag auf HerstellerID eingereicht. Dabei muss man einen Produktnamen angeben (also einen ausgesucht, bei dem Google nicht gleich 200 Treffer namhafter Steuersoftwarehersteller liefert).

Die Beantragung der HerstellerID sollte man frühzeitig angehen, da hier Wartezeiten von bis zu 4 Wochen versprochen werden.

In meinem Fall ging es dann innerhalb weniger Tage. Plötzlich konnte ich Bewegungen in den Logfiles zum Demo Server sehen und eine Stunde später war ein ok und eine HerstellerID per Email vorhanden.

Allerdings ist diese HerstellerID an Produkt und Hersteller gebunden. Einen Vermerk bezüglich GruppenID, insbesondere für Open Source Produkte, konnte ich nicht finden.

Auch hier sollte das Finanzamt wohl noch ein wenig feilen. Offensichtlich ist es nicht vorgesehen, dass Software erstellt wird und danach von völlig fremden Personen teilweise wieder zerlegt und neu zusammengesetzt - und das legal und mit Einverständnis des ersten Erstellers.

Fazit

Alles in allem ist unsere elektronische Steuerabgabe zumindest in Punkto Voranmeldungen ein sicherheitstechnisch mittlerer Alptraum.

Die Idee der Verschlüsselung wurde nicht zu Ende ausgeführt, weswegen jetzt zwar ein Sichtschutz für Aussenstehende bei der Datenübertragung besteht, aber kein Schutz vor anderen Netzteilnehmern, die auch einmal eine Steuervoranmeldung einreichen wollen oder sich bei der Angabe ihrer eigenen Nummer vertippt haben.

Es wurden keine weiteren Mechanismen verwendet, um eine Authentisierung des einreichenden Teilnehmers zu ermöglichen.

Die mitgelieferten Java-Klassen, die als Grundlagen zur Implementierung ausgegeben werden, sind im Paket nicht im Source Code vorhanden. Somit kann keiner prüfen, was innen drin wirklich geschieht.

Die Registrierung der Softwareentwickler (HerstellerID) ist in ihrer derzeitigen Anwendung für Entwickler von Open Source und freier Software nicht wirklich brauchbar.

Die Dokumentation kann verbessert werden (ist kein Argument, irgendwer jammert immer), ist aber zumindest vorhanden und mit genug Aufwand verwendbar.

Es wurden ausschliesslich offene Verfahren verwendet. Das bedeutet: Es wurden keine proprietären Kryptoverfahren oder Datenformate neu erfunden. Das finde ich persönlich sehr positiv.

Die bisherige Unterstützung seitens der menschlichen Verantwortlichen waren sehr freundlich und zügig, kurz: positiv. (Es hat mich keiner gebissen, auch wenn ich das erstmal befürchtet hatte.) Ich persönlich sehe hier deutlich positive Bemühungen. Vielleicht müssen sie es einfach nur noch etwas üben?

Fussnoten

[1] Man kann sich von einer elektronischen Abgabe befreien lassen.

[2] *http://www.elster.de/*

[3] Anmerkung zur Steuernummer: Die steht bei grossen Unternehmen leider nicht auf der Rechnung (die verwenden eine UstID), sondern bestenfalls bei Kleinunternehmen.

Bumping locks

Barry Wels <barry@toool.nl>
Rop Gonggrijp <rop@toool.nl>

How to open Mul-T-Lock (pin-in-pin, interactive, 7x7), Assa (6000 Twin), DOM (ix, dimple with ball), LIPS (Octro dimple), Evva TSC, ISEO (dimple & standard), Corbin, Pfaffenhain and a variety of other expensive mechanical locks without substantial damage, usually in under 30 seconds, with little training and using only inexpensive tools.

In this paper we describe an underestimated lock-opening technique by which a large variety of mechanical locks can be opened quickly and without damage by a relatively untrained attacker. Among other things we examine how this works, why it works better on some locks than on others, whether one could detect that this technique was used against a lock and what the lock-industry could do to protect new locks against this technique. Understanding the threat of this new method of manipulating locks is of added importance because we have found that this method actually works better on the more expensive mechanical locks generally considered to be most resistant to manipulation.

Preface – Why publish this?

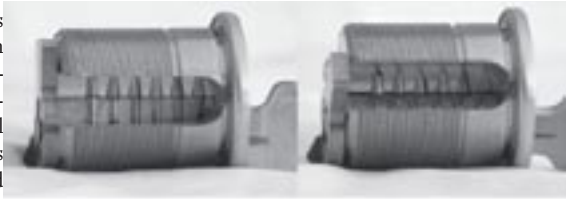
We decided to publish what we know about this method because we feel those that depend on the security of locks (or any other piece of technology for that matter) need to be able to continuously re-evaluate their security having full knowledge of any threats. This vulnerability is simply too generic: it affects many locks and cannot be 'fixed' by a single lock manufacturer working in secrecy until a new and better lock can be released.

Although we have further refined the method we were originally shown, we did originally learn about it through a public appearance by Klaus Noch. And we noticed yet other people knew how to make it work even better too. In other words, this knowledge is 'out there', the cat is out of the bag. Given these circumstances, rather than allowing knowledge of this method to spread slowly amongst those that could attack unknowing victims, we decided to publish so that facility managers can re-evaluate their security and see whether additional security measures need to be taken at some locations.

If you disagree, we encourage you to read [1] and [2] for a more thorough understanding of the discussion on whether or not to publish information describing security flaws before engaging in any heated debate.

Introduction to locks and lock security, How locks work

Photos courtesy of Matt Blaze



Pin tumbler locks, from the cheapest to the most expensive all work in roughly the same way. The key slides down the keyway in the inner cylinder of the lock. As it moves, the cuts in the key move stacks of two or more pins, moving in holes drilled through the outer and inner cylinder. Small springs behind these pins push the pins back after a high point on the key has passed. When the correct key is all the way in and the 'shoulder' of the key rests against the inner cylinder, all the gaps between the pins inside

the lock align on the 'sheer line', and the inner cylinder is free to turn.

The picture above shows a 'cut-away' version of a simple pin tumbler lock with the correct key inserted. For a much more thorough introduction to the inner workings of locks, please refer to [3].

Picking locks

A Lock can be 'picked'. A skilled operator can use tools to feel and move individual pins in the lock. Lockpicking allows one to open a lock by exploiting the fact that the pin stacks are never perfectly aligned. This causes some pins to be stuck between the inner and outer cylinder before others. Because of this, one can feel that certain pins are correctly aligned before all the pins are aligned. And because the outer pins that would jam before others will remain on the outside of the inner cylinder after the lock is turned slightly, one can successively place the pins in the correct position and open the lock.

Lockpicking takes quite a bit of practice. Apart from intelligence professionals, criminals and locksmiths practicing it, lockpicking has become a regular sport, complete with official clubs and championships1. Lock manufacturers have defended new locks against picking by inserting so-called 'mushroom pins', by making keyways narrower (providing less space for tools) and by lowering the mechanical tolerances of the lock manufacturing process. (See picture of EVVA lock on page 7)

Going over the details of locks and lock picking would be outside of the scope of this paper. Please refer to the "MIT Guide to Lock Picking" [3] if any of the above is unclear.

The snapper pick, lockpick gun and vibrating tools

Another means of opening locks without the key is by using a snapper pick, lockpick gun or vibrating tool. These devices all exploit Newton's law that says that for every action there is an equal and opposite reaction. Most people are familiar with Newton's cradle, a device which is often used to demonstrate this law.

If a ball all the way on the left or right side is lifted up and let loose to collide with the row of suspended balls, this ball will transfer all its energy to the next ball and so forth, until the ball on the other end moves to swing away from the other balls. When it swings back, the process is reversed and the original ball swings up. The same principle can be observed during a game of billiards: one ball hits another one, and this ball continues onward whereas the first ball now lies still.

This principle can be used to open locks: if impulse energy is transferred to the first pin, it will tend to stay in place and the second pin tends to move away from the first one, until the spring stops it and pushes it back to touch the first pin.

A 'lockpick gun' such as the one shown below will, when the trigger is pulled, tension a spring and then when the trigger is pulled all the way use the force of that spring to snap the needle up for a short distance, but with a very sharp and powerful motion. By positioning this needle into the lock, just touching the pins, and then pulling the trigger, one tries to hit all the pins simultaneously. By then making the lock turn in the split-second before all the upper pins are pushed back by the springs in the lock, one can

open the lock. The amount of turning force and the timing with which to apply it require some training.

Vibrating picks use the same principle except many times a second, requiring less training on the part of the operator. A snapper pick is the simpler version of a pick gun.

The lock industry has created locks that are more resistant to this technique. More resistant locks have narrower keyways, preventing tools from being inserted in the first place, and making it harder to transfer the impulse energy to the pins. More resistant locks also have smaller tolerances, creating less space for the pins to bounce around.

Lockpicking gun [2]

Snapper pick [3]

Bumping locks – History

Bumping, sometimes also called 'Rapping', has been a known technique for at least the past 50 years. A bump key is described in Marc Tobias's reference work "Locks, Safes, and Security" [4] on page 603. Few people use the technique, and the method does not seem successful against a large number of locks unless the 'minimal motion method' described below is used. Once correctly used, we found this technique to be immensely powerful, allowing a large variety of locks to be opened. We did not invent this technique, and others probably thought of some of the same refinements we did. We do feel bumping is underestimated, and this paper exists to point to its effectiveness.

Principle & Bump keys – 999keys

So we have a basic trick to open a lock by making the second pin jump away from the first, but no efficient means to apply this energy to the bottom pin. As it turns out, the best way to transfer energy to the pins is using a key. First of all, we need a 'bump key' for the lock in question. A bump key is a key in which all the cuts are at maximum depth. The picture below shows bump keys for various locks. Bump keys are sometimes called '999' keys because all cuts are at maximum (9) depth.

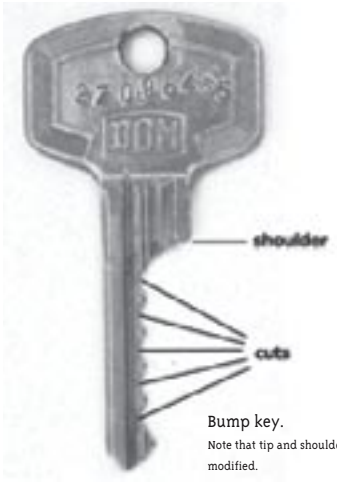
As you can see you can cut bump keys for both regular pin tumbler locks as well as for 'dimple locks', whether 'pin-in-pin' or not. Just remember to take away all the material that could be taken away by the deepest combination for that position.

There are machines that will cut a key based on the numbers that represent the depth at each position. Having access to such a machine speeds up the process of creating a bump key that has the cuts in the exact right position, although one can also use a file and a steady hand to create one. Bump keys, once cut, can be copied on regular key-cutting equipment. You do not necessarily need to have an uncut key (called 'blank') to make a bump key: because all the cuts of a bump key are at maximum depth, any used key for a given lock can be converted into a bump key.

The pull-back method

Now there are different methods for using such a bump key to transfer force to the pins inside the lock. When we first learned of the method, we were told to first insert the key all the way, and then pull it back one pin. Then, hit the back of the key (the part where you normally hold on to it) with a solid object such as a hammer, and then turn the key a split-second later. We found the exact timing for the turning of the lock to be critical, requiring quite a bit of practice. While this method worked on some locks, it did not work on a great many others. Among other problems: when keys had very deep cuts, the trick tended to not work either because the pins would still be pushed in too far by the parts of the bump key between the deepest points.

The minimal-movement method



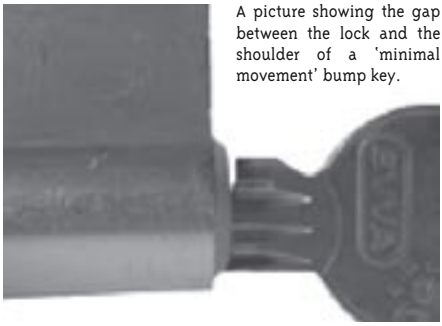
Bump key.
Note that tip and shoulder are not yet modified.

Normally, if you insert a key all the way into the lock, the pins inside the lock touch the deepest point of the cut in the key at the point where the shoulder of the key makes contact with the inner cylinder of the lock. By filing a tiny bit of metal off both the tip and the shoulder of the key, we can create a bump key that can go just a little bit deeper into the lock. When such a bump key is inserted all the way into the lock, it will be pushed out again a tiny bit by the force of the springs inside the lock, until the pins again rest on the deepest point in the key cuts. We found filing off between 0.25 and 0.5 mm works best, but you may wish to experiment for the best results.

We found it is very easy to take off too much. All you need to do is make sure that when the key is in all the way, the pins touch the sides of the cuts instead of the bottoms. Seeing the key be pushed back a fraction of a millimeter by the springs in the lock means you have filed away enough material from the shoulder.

Now that we have our bump key, we need to hit the back of the key with something that applies the right amount of impulse power, without having so much weight that it would damage the bump key or the lock. We use a special bumping tool built by Kurt Zühlke called the Tomahawk, but anything with not too much weight and preferably also some swing, such as a dull breadknife held by the blade or the handle of a hammer could also work.

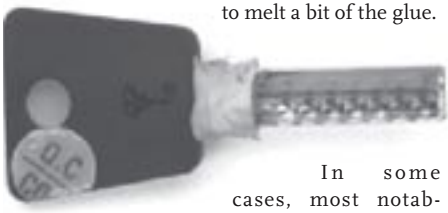
The bumping of a lock using the 'Tomahawk'.



A picture showing the gap between the lock and the shoulder of a 'minimal movement' bump key.

Some keys do not have a shoulder: such is the case with the Mul-T-Lock keys. In this case, the depth of the keyway determines how far one can insert the key. To create a bump key, one would theoretically only need to cut off a bit from the end of the key. However, the end of the key and the insides of the lock were found to be too fragile to withstand the repeated hammering while we were trying to open the lock.

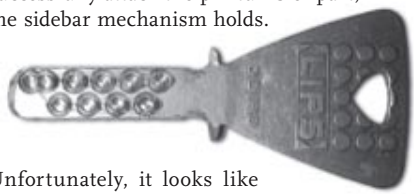
Oliver Diederichsen came up with an innovative way of making sure the key wouldn't go too deep. Take off a piece at the end to allow for the key to go further in, and then cut a glue stick in two, and glue the two half-round pieces to the key after heating them enough to melt a bit of the glue.



In some cases, most notably with some dimple key locks, the force needed is small enough that one can hold the bump key back between one's fingers: no need for glue or anything else.

Multi-principle locks

Some locks employ two different principles, such as the Assa Twin 6000. This is a very secure lock, and one of our former favorites. One part of the lock is a regular pin tumbler mechanism, while another part is a sidebar mechanism. Although bumping will successfully attack the pin tumbler part, the sidebar mechanism holds.



Unfortunately, it looks like most Twin 6000's sold in a certain region have the same sidebar, to allow for locksmiths to store pre-cut sidebar blanks for copying. If this is the case, one could simply cut a bump key out of any key with the correct sidebar for a region.[4]

Problems

It is very easy to damage the lock and/or the bump key using any bumping method. The force needed to transfer enough impact energy to the pins can cause the shoulder of the key to make a dent on the front of the inner cylinder, as shown below. The photo of the LIPS OCTRO bump key shown to the right shows the result of the forces on the shoulder, and also shows damage to the dimples from repeated bumping.

More seriously, bumping can cause minor deformations, causing the bump key to get stuck in the lock. Usually extra force when pulling it out helps to remove the bump key, but in some cases it can get very stuck.

The bump key should be made out of the hardest metal available: softer metals will quickly deform at the shoulder, causing the bump key to go too deep and not work. We predict a large market for hardened steel bump keys for the popular high-security cylinders. Also, some locks where the inner cylinder is made out of softer metals can be damaged quite easily by the shoulder of the bump key.

Refinements & Ideas

One could envision a way to clamp the key between two pieces of metal, possibly attached to a small rubber block that touches the lock. This way one could hit the key without impacting the lock in the same damaging way the shoulder of the bump key does. The rubber would be chosen such that it would deform only by the fraction of a millimeter needed for bumping to work. Shown is Jort Knaap's solution for a Mul T Lock Interactive, built out of two nylon washers and a piece of hard rubber which we have found to completely prevent denting.

For the time being, putting either a thick rubber band such as used in the postal system or a tie-wrap between the shoulder and the lock seems to prevent or diminish denting. (White tie-wraps seem to be the toughest, and one can tie it though the key-ring hole on the key to keep it in place.)

Expensive locks

We've noticed during our experiments that the more expensive a lock was, the better this method worked. Bumping works on some high-end locks we never thought could be manipulated easily, and can be very hard or impossible to get to work on very inexpensive locks. There are a number of reasons for this. First of all the more expensive locks are made out of harder metal, causing less deformations on impact. Then expensive locks also have tighter tolerances, allowing for smoother motion of the parts inside.

The fact that some of these locks have narrower keyways that block normal tools doesn't bother us: our bump key doesn't need more room than the normal key. In fact: the smoother everything is, the less of the impact force is wasted.

So it looks like everything that used to make a lock 'good' works in favor of this method, but we suspect a large number of less expensive pin tumbler locks to also be vulnerable. We have either bumped or personally witnessed the bumping of the following locks (in no particular order):

- Assa Twin 6000
- Mul-T-Lock pin-in-pin
- Mul-T-Lock interactive
- Mul-T-Lock 7x7
- LIPS Octro
- LIPS Keso
- DOM IX KG
- DOM 5-pin
- EVVA TSC
- Zeiss IKON 5-pin
- Corbin 5-pin
- ICEO dimple
- D.L.C. 5-pin
- Lince dimple
- ABUS 5-pin
- Pfaffenhain
- GEGE AP3000

Important disclaimer: please note that the above list does not mean to imply that every cylinder of a named brand and type will open readily using bumping. Locks are expensive and we are not a commercial testing lab, so we have had only a very limited number of testing locks available to us. The presence of a lock in the above list just means bumping worked at least once on a cylinder that we had access to. To us this means that type of lock is at least suspect, and further research is needed. Also, it is very probable that a great many locks not on this list are vulnerable too. Also note that we have seen locks that bump open quite easily a number of times, and then for some reason become very hard to bump, even though the regular key still works.

Forensics

Lock forensics is, among other things, the science behind knowing whether a lock was opened using manipulation. Lockpicking, for instance, often leaves tiny scratches on the pins in places where the regular key would not scratch. The first sign that a lock was bumped is the dent made on the outside of the inner cylinder by the shoulder of the bump key. But as previously discussed, there are ways to make sure this denting doesn't occur, and in some cases, such as the Mul-T-Lock bump key we've shown, no dents will be made on the outside. Also beware that both older and softer (cheaper) locks will have a dent there even if they were never bumped.

Looking at the pins on the inside of a bumped lock compared to pins from a lock that wasn't bumped showed no differences that could be detected by the naked eye or by using a magnifying glass. It could well be that differences can be found under a microscope. We lack the basic metallurgic knowledge, the forensic experience and the necessary equipment to say anything conclusive about the pins we examined.

Given that the insertion of a bump key isn't much different from inserting a regular key, we'd suspect no special scratch marks would be found other than maybe some miniature dents and deformations caused by the impacts. Until more is known, we think it is diligent to assume that any lock that can be bumped can also, with some care, be bumped without leaving any telltale traces.

Conclusions

The perfect lock does not exist. With enough training, tools and time, almost any lock can be manipulated. Practical security is almost always a trade-off between the cost of the lock and the time and effort needed for an attacker to open the lock. However: in terms of mechanical lock security, we believe that this vulnerability exposes a fundamental flaw in a large number of existing mechanical lock designs. Resistance against this attack will have to be incorporated in all future high-end locks, and judging by their own design criteria a large number of high-end locks seen today must be considered flawed.

Re-evaluating facility security

If your present security depends on one or more mechanical locks presently thought to be very resistant to manipulation, you should at least investigate whether these locks can be bumped. Manufacturer claims as to how manipulation-resistant a certain lock is should be considered worthless unless the claim specifically mentions resistance to bumping.

If you employ a type of lock that can be bumped and your security criteria do not allow for a lock that can be opened by unskilled attackers in 30 seconds then you should replace the locks in question.

In instances where security is of the utmost importance, you may wish to implement extra security measures assuming even high-end mechanical locks can be opened in much less time than previously assumed. Employing a number of different high-end locks for a given entry may add additional security.

The fact that a lock has a keyway-shape for which blanks are not generally available offers little protection: devices exist that can create a blank when given a key, or even a picture of the outside of the lock. Also note that one does not need a blank to cut a bump key: any key will do.

This may be a good time to consider deploying electronic locks and electro-mechanical opening mechanisms.

Locks that resist bumping

There are mechanisms that do not allow for the two pins to separate except when slid sideways, such as used in the Emhart interlocking lock (which is not being produced anymore). As far as we can see, such a mechanism would successfully foil the bumping attack. Also some mechanisms which have a one-piece locking mechanism (such as a 'sidebar') may resist bumping5.

Locks that involve rotating discs (such as Abloy Protec) or magnets (such as Evva MCS and Anker) are also not susceptible to this attack6.

Klaus Noch sells modified standard Euro profile locks which lock up (i.e. 'broken but closed') upon most attempted manipulations, including bumping. [5]

Acknowledgements

The authors wish to thank Walter Belgers, Matt Blaze, Manfred Bölker, Kim Bohnet, Paul Boven, Django Bijlsma, Paul Crouwel, Oliver Diederichsen, Han Fey, Julian Hardt, Jiemme, Jord Knaap, Klaus Noch, Marcel van der Peijl, Marc Tobias, Rob Zomer and Kurt Zühlke and all the other people from Toool and Ssdev for their input on this topic and/or for energizing discussion on the security of locks in general. In addition, the authors wish to thank Matt Blaze, Paul Boven and Marc Tobias for permission to use illustrations.

References

[1] Matt Blaze, On the discussion of security vulnerabilities, *http://www.crypto.com/hobbs.html*
[2] Paul Clark, Full Disclosure Debate Bibliography, *http://www.wildernesscoast.org/bib/disclosure-by-date.html*
[3] Theodore T. Tool, MIT Guide to Lock Picking, 1991, *http://www.toool.nl/mit.pdf*
[4] M.W. Tobias, Locks, safes and security (second edition), 2000, *ISBN 0-398-07079-2*
[5] Klaus Noch, *http://semtechnologie.de/technik.htm*

1 Ssdev (Sportsfreunde der Sperrtechnik Deutschland eV) in Germany and Toool (The Open Organization Of Lockpickers) in The Netherlands.

2 In this case a special gun, made by Kurt Zuhlke. The head on this gun can be reversed to snap either up or down, allowing picking of 'European style' locks where the pins are pushed up by the springs.

3 Image taken with permission from "Locks, safes, and security" [4], page 578

4 By the way: did we mention we collect sidebar profiles of the Assa Twin? If you have a key, please mail a detailed picture of it, complete with the region where you bought the key, to barry@toool.nl

5 Unless the sidebar combination is known, such as is the case with the Assa Twin 6000 where the same sidebar seems used for many locks sold in a certain region.

6 Bumping could still be used to attack a pin tumbler portion of a multi-principle lock.

GIS – Geo Informations Systeme

Markus Schaber <markus.schaber@ulm.ccc.de>

Daten mit räumlicher Zuordnung – vom Routenplaner bis zur Baumbestands-Datenbank – sind nicht nur ein zukunftsträchtiges Feld in der Informatik, sondern bringen auch ihre eigenen Problemstellungen mit sich. Dieser Artikel versucht, einen kurzen Überblick über das Gebiet zu geben, und einige Schwerpunkte exemplarisch zu beleuchten.

Was ist GIS?

Für GIS existieren mehrere Definitionen [1]. Viele der Definitionen bevorzugen die Teilaspekte von GIS, die im Projekt der jeweiligen Definierer die Hauptrolle spielten. Im Rahmen meiner Arbeit hat sich folgender Kompromiss herauskristallisiert:

GIS ist ein rechnerbasiertes System zur Manipulation, Analyse und Präsentation von Daten mit Ortsangaben.

Das Hauptaugenmerk bei GIS liegt herbei auf den Ortsangaben, also den geographischen Komponenten des Datenbestandes. GIS-Systeme helfen, aus Daten Informationen und letztendlich Wissen zu gewinnen. Hierbei sind – wie so oft – viele „**W**“ interessant:

- **W**as ist **W**ann **W**o?
- **W**arum ist es dort?
- **W**o hätte es sein sollen?
- **W**ohin soll es?
- **W**ieso interessiert es?

Die ersten drei „**W**“, die erste Frage, repräsentieren den reinen geographischen Datenbestand. Die nächsten drei „**W**“ sind zusätzliche Informationen, die helfen, den geographischen Datenbestand zu interpretieren. Und das letzte „**W**“ schliesslich ist der Grund dafür, dass wir uns die ganze Arbeit überhaupt machen.

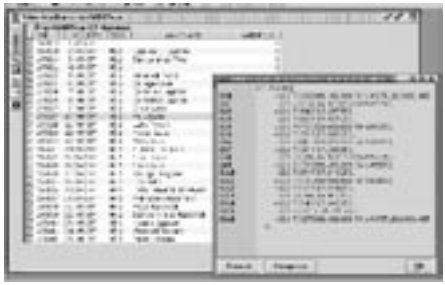
Eine CD mit Kartenmaterial, ein Programm, eine Landkarte oder ein GPS-Empfänger sind für sich jeweils noch kein GIS-System. Wenn man die Teile jedoch entsprechend kombiniert, erhält man ein GIS-System. Beispiele für GIS-Systeme sind:

- Stau-Info
- Grundwasserdatenbank
- Routenplaner
- Erdölsuchdatenbank

Räumliche Daten

Nach Schätzungen von Wissenschaftlern haben 80% aller Daten eine räumliche Komponente [2]. Daten aus fast allen Wissenschaften lassen sich räumlich analysieren. Beispiele hierfür sind Wettersimulationen, Kernspintomographien, Vogelzug-Verfolgung, Telefonbücher und sogar die Aufstellung der Spieler auf dem Fußballfeld. GIS-Systeme beschäftigen sich mit einer Teilmenge dieser räumlichen Daten, meist in der Größenordnung von Metern bis einigen dutzend Megametern [3].

Allerdings sind diese räumlichen Daten in der rechnerlesbaren Form nicht einfach zu verstehen, hier ein Beispiel:



Hier gilt, wie so oft, die Regel: „Ein Bild sagt mehr, als tausend Worte / Zahlen“ [4].

Die Erstellung und Verarbeitung solcher Bilder ist ein wichtige Bestandteil von GIS-Systemen.

Räumliche Daten können in Raster- und Vektorform vorliegen. Rasterdaten sind z. B. Satellitenphotos, Luftaufnahmen, Radar-Bodenuntersuchungen oder Scans von Papierkarten. Sie haben eine einfache Struktur und Verarbeitung, sind jedoch für den Rechner schwer zu interpretieren. Als Dateiformat ist hier vor allem GeoTIFF zu nennen, das im Grunde das TIFF-Format um einen Header mit Informationen zu Position und Maßstab erweitert.

In Vektorform liegen die meisten im Rechner entstandenen Daten vor, wie beispielsweise Baupläne aus einem CAD-System oder aufgezeichnete GPS-Koordinaten. Allerdings werden auch viele ursprünglich als Rasterdaten vorliegende Daten vektorisiert, da dies eine einfachere Interpretation der Daten ermöglicht. Bestandteil der Vektorisierung ist die Aufteilung in Ebenen oder „Features“, die verschiedene Arten von Objekten trennt. In obiger Grafik sind z. B. die Features „Wasser“, „Straße“ und „County“ dargestellt. Professionelle Datenlieferanten wie NavTeq liefern je nach Kundenwunsch und Geldbeutel dutzende von Features aus, neben Ländergrenzen, Strassen und Flüssen z. B. auch die Standorte von Tankstellen, Parkplätzen, Autowerkstätten, Hotels oder Fast-Food-Ketten [5].

Koordinaten und Projektionen

Aus praktischen und politischen Gründen haben sich historisch tausende von Referenzsystemen für Koordinatenangaben entwickelt. Allein die European Petrol Survey Group [6] hat in ihrer Datenbank über 3000 Einträge. So wird für Planungen auf Stadtniveau üblicherweise die Erdkrümmung vernachlässigt, und mit einer Ebene in Metern gearbeitet, so ist in Berlin hierfür das Soldner-System üblich.

Für weltumspannende Datenbestände hat sich in letzter Zeit WGS84 durchgesetzt, das z. B. auch von den GPS Satelliten verwendet wird. Es basiert auf Längen- und Breitengraden.

Wenn größere Bereiche der Erde auf einer zweidimensionalen Karte abgebildet werden sollen, dann treten durch die notwendige Projektion Verzerrungen auf. Dies kann jeder nachvollziehen, wenn er ein größeres Stück Orangenschale „plattdrücken“ will – es reißt.

Diese Probleme werden noch dadurch verschärft, dass die Erde keine ideale Kugel, sondern mehr eine Kartoffel darstellt. Sie ist zum einen etwas abgeplättet (durch die Fliehkraft der Erddrehung), und zum andern unterscheidet sich die Krümmung auch noch lokal (z. B. durch plattentektonische Vorgänge). Dies wird dadurch kompensiert, dass man für unterschiedliche Regionen und Zwecke unterschiedliche Projektionen und Referenzellipsoide [7] einsetzt. Zusätzlich ist z. B. die in Deutschland übliche Gauss-Krüger-Projektion in Zonen von drei Längengraden eingeteilt.

Das Hauptproblem bei den Projektionen ist, dass durch die Abbildung von drei auf zwei Dimensionen immer Informationen verloren gehen. Man muss je nach Anwendung entscheiden, ob man Wert auf die Längen, die Winkel oder die Flächen legt, und sich dann die entsprechende Projektion aussuchen.

Rechts sind drei Beispiele für sogenannte Zylinderprojektionen. Die erste ist eine sogenannte Mercatorprojektion. Diese ist Winkeltreu, d. h. alle Winkel auf der Karte stimmen mit den Winkeln in der Realität überein.

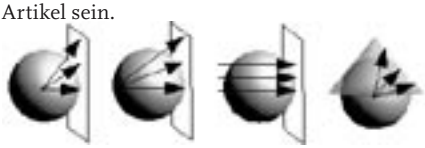
Die Gauss-Krüger-Projektion verwendet dasselbe Prinzip, allerdings wird hier der Zylinder

waagerecht über die Erde „gestülpt“. Der Kreis, an dem der Zylinder die Kugel berührt, ist der Längengrad in der Mitte der entsprechenden Zone.

Der zweite Zylinder zeigt eine orthographische Projektion, hier gehen die Strahlen parallel durch die Erde. Dadurch wird eine annähernde Flächentreue erreicht.

Im dritten Zylinder ist die stereographische Projektion abgebildet. Sie hat in Äquatornähe eine insgesamt relativ gute Treue von Winkeln und Längen.

Neben den Zylinderprojektionen gibt es noch die Azimutalen Projektionen und die Kegelprojektionen. Allerdings belasse ich es für diese bei obenstehenden Beispielbildern, schliesslich soll in dieser Schleuder auch noch Platz für andere Artikel sein.



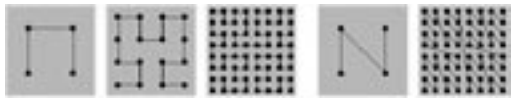
Räumliche Indices

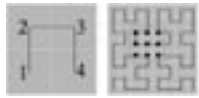
Ein weiteres Problemfeld für die GIS-Programmierer ist die Indizierung räumlicher Daten, um eine effiziente Suche zu ermöglichen. Die üblichen Index-Datenstrukturen sind lediglich eindimensional, ermöglichen also die Sortierung z. B. von Zahlen, Zeiten oder Namen. Wie soll man jedoch räumlich, also nach zwei oder drei Koordinaten „sortieren“, um darin effizient suchen zu können?

Hier haben sich zwei Hauptmethoden herausgebildet. Zum einen werden mehrdimensionale Bäume verwendet. Man kann also, vereinfacht gesagt, an einem Blattknoten nicht mehr nur nach links oder rechts „verzweigen“, sondern in alle räumlichen Richtungen. Dies geht relativ gut, solange sich die Objekte nicht überlappen (z. B. punktförmige GPS-Koordinaten). Dann kann man Zuordnungen wie „Objekt A ist nordwestlich von Objet B“ treffen, und damit einen wunderbaren Baum aufbauen. Jeder Knoten hat dann in einem zweidimensionalen Baum nicht mehr zwei „Zeiger“ für „links“ und „rechts“, sondern acht. (Nord, Nordwest, West, Südwest, Süd, Südost, Ost, Nordost.)

Sobald sich die Objekte überlappen oder sogar komplett ineinanderliegen, wird die Sache allerdings komplizierter. Dann müssen sich entweder die Bereiche der Blätter des Baumes überlappen, oder Objekte in mehreren Blättern eingetragen sein, was die Sache ineffizienter macht [8].

Die zweite Hauptmethode ist die Abbildung mittels sogenannter raumfüllender Kurven. Eigentlich sind dies Folgen von Kurven. in denen ich eine Kurve durch das regelgemäße Ersetzen der Teile der Vorgängerkurve durch kompliziertere Stücke erzeuge. Raumfüllende Kurven haben zusätzlich die Eigenschaft, dass sich für jeden Punkt im Raum eine Kurve in der Folge gibt, ab der dieser Punkt Bestandteil der Kurve ist. Unten sind links drei Iterationstiefen der sogenannten Hilbert-Kurve abgebildet, rechts die erste und die dritte Tiefe der Z-Kurven.





Um damit eine effiziente Suche zu ermöglichen, wird üblicherweise eine feste Iterationstiefe gewählt, und die Punkte in dieser Tiefe durchnummeriert. Jeder Punkt ist dann der Mittelpunkt eines Quadrates, dem alle innerhalb dieses Rechteck liegenden Punkte zugeordnet sind. Dann baue ich einen Index nach den Punktnummern auf, und alle innerhalb des Quadrates liegenden Objekte werden beim zugehörigen Punkt im Index eingetragen.

Wenn ich nun innerhalb eines Bereiches etwas Suche, dann bekomme ich Intervalle von Punktenummern, die die Teilkurven bezeichnen, die das Such-Rechteck schneiden. Mit diesen Intervallen sehe ich im Index nach, ob dort Objekte eingetragen sind. Das Geheimnis ist also, die Form der Kurve so zu wählen, dass möglichst wenig Intervalle herauskommen, da jedes Inter-

vall einen Zugriff auf den Objektindex bedeutet. Es gibt einen Rekord, die AR^2W^2 Kurven, deren Schöpfer mathematisch bewiesen haben, dass – bei festem Seitenverhältnis – jeder Suchraum auf maximal drei Intervalle abgebildet wird, und dass dies auch das bestmögliche Ergebnis ist [9].

Aufgrund der Abbildung von mehreren Dimensionen auf eine ist es nie möglich, dass alle räumlich nahe beieinander liegenden Punkte auch im Index (also dem Kurvenverlauf) nahe beieinander liegen. Allerdings läßt sich die Gegenaussage erreichen: Die Hilbert-Kurve garantiert, dass zwei Punkte, die auf der Kurve nahe beieinander liegen (also zwei aufeinanderfolgende Nummern haben), auch im Raum nahe beieinander liegen (nämlich genau eine Quadratlänge). Die oben rechts gezeigte Z-Kurve kann das nicht garantieren – die zwei Endpunkte der schrägen Linien sind im Index auch nur eine Nummer auseinander, aber in der Fläche deutlich weiter. Auch hier kann man wieder den Trick der überlappenden „Einzugsbereiche" der Kurven anwenden [10].

Ausklang

Um den Austausch von GIS-Formaten zwischen verschiedenen Anwendern und Applikationen zu vereinfachen, hat das Open Geospatial Consortium [11] eine Reihe von Standards entwickelt, wie geographische Daten repräsentiert und verarbeitet werden. Dazu gehören neben einem eindeutigen Geometrie-Modell auch deren Repräsentation, sowie auch eine Standardisierung für SQL-Datenbanken und den XML-Dialekt Geography Markup Language (GML 3.0) zur Darstellung von GIS-Daten.

PostGIS [12] ist eine zum OpenGIS-Standard kompatible Erweiterung der bekannten PostgreSQL Datenbank, und zudem unter den Open Source Datenbanken die vollständigste, wenn auch bei weitem noch nicht optimale, Implementierung. Die Jump Workbench [13], mit der auch der obige Screenshot mit Kartenausschnitt erstellt wurde, ist eine umfangreiche, erweiterbare Plattform für GIS-Applikationen, die sich ebenfalls am OpenGIS Standard orientiert, und zu PostGIS als Datenquelle kompatibel ist. Mit dieser Software und den Daten des oben genannten Tiger-Projektes kann man als Ausgangsbasis gut anfangen.

Eine schöne Applikation zum „rumklicken" ist auch *map24.de*, allerdings nur mit Internetanschluss und Java brauchbar.

Für die Zukunft ist zu erwarten, dass GIS-Applikationen zunehmende Bedeutung erlangen, und deshalb Informatiker mit den entsprechenden Techniken umgehen müssen. Eine Literaturliste zur weiterführenden Lesen findet sich auch unter *http://ulm.ccc.de/~schabi/gis/* am Ende der Vortragsfolien.

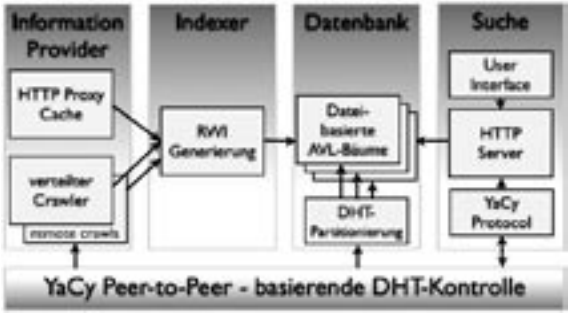
[1] Siehe z. B. *http://www.gis.com/whatisgis/* oder *http://www.e-isn.com/gis-software.htm*
[2] Quelle: *http://www.mapcruzin.com/what-is-gis.htm*
[3] Erdumfang = ca. 40.000km = ca. $3^1/_2$ Dutzend Megameter.
[4] Das Bild wurde mittels der Jump Workbench [13] aus den Daten des Tiger-Projektes *http://www.census.gov/geo/www/tiger/* erstellt, leider scheint es für Deutschland/Europa keine kostenlose Datenquelle in dieser Qualität zu geben.
[5] Wer will, kann sich mal auf http://*map24.de/* auf Strassenebene „durchklicken" und dabei auf die kleinen Symbole achten.
[6] *http://www.epsg.org/*
[7] Ein Referenzellipsoid ist eine etwas flachgedrückte Kugel, die (da mathematisch einfacher) näherungsweise als Ersatz für die zu komplizierte Erde herhalten muss.
[8] z.B. R-Bäume, siehe z. B. *http://www.dbnet.ece.ntua.gr/~mario/rtree/*
[9] *„Space-filling curves and their use in the design of geometric data structures"*, von Tetsuo Asano, Desh Ranjan, Thomas Roos, Emo Welzl & Peter Widmayer, siehe z. B. *http://www.ti.inf.ethz.ch/pw/publications/papers97.html* oder *http://dx.doi.org/10.1016/S0304-3975(96)00259-9*
[10] Siehe „XZ-Ordering: A Space-Filing Curve for Objects with Spatial Extension, Christian Böhm, Gerald Klump und Hans-Peter Kriegel, Uni München
[11] Ehemals OpenGIS-Consortium *http://www.opengeospatial.org/*
[12] *http://www.postgis.org/*
[13] *http://www.jump-project.org/*

YaCy – Peer-to-Peer Web-Suchmaschine

Michael Christen <mc@anomic.de>

Information ist im Web eine stark kontrollierte Ressource – Portale, Suchmaschinen und das DNS sind zentral in de-facto-Monopolen organisiert und bestimmen, welche Daten verfügbar sind. Mit YaCy wird die Kontrolle wieder an die Nutzer zurückgegeben.

Das YaCy-Projekt wurde Ende 2003 mit dem Ziel gestartet, eine freie, unabhängige und nicht zensierbare P2P-basierte Web-Suchmaschine zu erstellen. Zu diesem Zeitpunkt existierten viele gut funktionierende P2P-Filesharingtechniken und mehrere freie Implementationen von Crawlern/Indizierern, aber keine Technik, die P2P mit Suchmaschinentechnik verbindet. Wir stellen hier die über die Funktion einer Suchmaschine hinausgehenden Eigenschaften und die Architektur von YaCy vor.



Suchmaschine mit Mehrwert

YaCy ist nicht nur eine Suchmaschine mit Crawler und Suchfunktion, sondern auch ein Web-Server, ein caching http proxy mit optionalem prefetch, eine DNS-Erweiterung, ein Messagingsystem und ein Wiki. Warum das Ganze? Es gibt einen gemeinsamen Schlüssel für all diese Funktionen: Synergien zwischen Suchtechnik und der Loslösung von zentral-gesteuerten Diensten im Internet. Im Detail:

P2P-Suchmaschine und caching http proxy: Für die Existenz des Proxy in YaCy gibt es drei Gründe:

Eine P2P-basierte Software ist auf lange Onlinezeit angewiesen. YaCy soll nicht nur laufen, während der User eine Suche absendet. Synergie: Durch die Nutzung des Proxy als Mehrwert wird eine lange Onlinezeit erreicht.Der Proxy ist ein 'Information Provider' für den Indexer.

Synergie: die Suchmaschine erhält als Option 'kostenlos' Web-Seiten zum Indizieren ohne Crawling.

Der Proxy bietet mit seinen eingebauten Filtern einen persönlichen **Schutz vor ungewollten Webinhalten:** das ist der notwendige 'Gegenpol' zur Zensurfreiheit durch die Suchmaschine. Die Idee ist, dass jeder sich (seine Familie, sein Unternehmen, etc.) wieder soweit selbst zensieren kann, wie er es ggf. von einem Suchmaschinenbetreiber erwarten würde (wenn er denn diese Erwartung hätte).

Synergie: populäre Filter (bspw. Bannerblocker) können von Peer zu Peer importiert werden.

Crawlen und Prefetching: während Crawling eine typische Aufgabe einer Indizierungssoftware ist, liefert ein proxy prefetch ggf. schnellere Zugriffszeiten für den Benutzer des Proxys.

Synergie: beide Funktionen benutzen prinzipiell den gleichen Algorithmus.

Eingebauter Webserver, Filesharing, Wiki & Messaging: wer durch die dezentrale Struktur der YaCy-Suche Informationsfreiheit sicherstellen möchte, will ggf. auch ein Publikationsmedium nutzen, das in gleichem Maße keiner Zensur unterliegt. Webinhalte können zwar durch YaCy ad-hoc erfasst werden, aber selbsterstellte Daten könnten weiterhin auf fremden Servern gesperrt oder entfernt werden.

Synergie: unzensierte Suchergebnisse sind erst dann sinnvoll, wenn deren Ressource auch geladen werden kann. YaCy gibt dazu einige Basiswerkzeuge an die Hand. Da der dazugehörige Server dann dem Nutzer gehört, kann er nicht zensiert werden.

Webserver, Suchinterface und Proxy: die natürlichste Umgebung für eine Websuche ist eine Webseite. Daher besteht YaCys GUI aus einem integrierten Webserver mit Servlet-Engine.

Synergie: der Proxy, das GUI und die eigenen Webinhalte (siehe oben) können den gleichen eingebauten httpd benutzen.

DNS-Umgehung und Erweiterung um die Top-Level-Domain „yacy": Das DNS-System ist mit seiner zentral-hierarchischen Struktur ein einfacher Angriffspunkt für Webzensur. YaCy bietet jedem Peerbetreiber seine eigene „<peername>.yacy"-Domain, die automatisch durch den Proxy zum YaCy-Webserver des Peerbetreibers aufgelöst wird.

Synergie: die Nutzung des Proxies macht den Eingriff in die DNS-Auflösung möglich und die YaCy P2P-Verwaltung stellt ganz selbstverständlich eine Peer-zu-IP-Datenbank dar, was auch mit dynamisch zugewiesenen IPs funktioniert. Es existiert außerdem ein schlüssiges Konzept, um ohne zentrale Datenbank einen Namensdiebstahl der yacy-Domains unterbinden zu können.

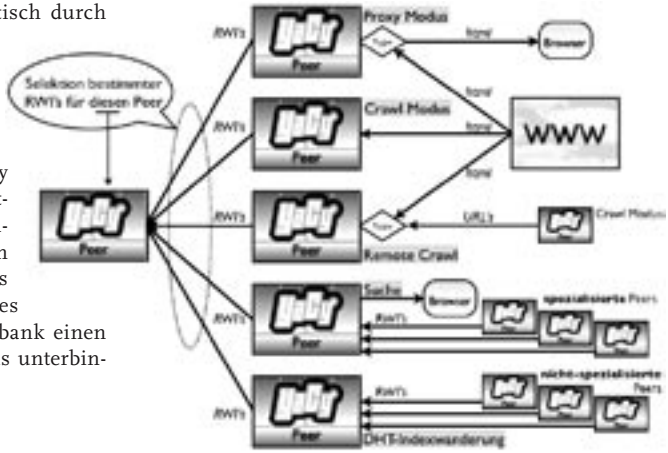
Zwar realisiert der eingebaute Proxy YaCys zentrale Konzepte, aber die Software kann auch betrieben werden, ohne den Proxy nutzen zu müssen. Dann dient YaCy „nur" als Suchinterface, Crawler, Webserver etc.

Indizierung und Peerarchitektur

YaCy besteht aus einem Crawler, einem Indexer, einer Datenbank, einem Suchinterface und der P2P-Organisation:

• Wir haben vom Crawler abstrahiert und sehen konzeptionell einen „Information Provider" vor. Als solcher kann ein Crawler eingesetzt werden. Es ist aber auch möglich, Seiten aus dem integrierten Proxy Cache als Input für den Indexer zu benutzen.

• Der Indexer erzeugt konzeptionell einen Reverse Word Index (RWI), d.h. zu jedem Wort eine Liste der URLs plus Rankinginformationen. In unserer Implementation des RWI werden jedoch keine Wörter im Klartext gespeichert, sondern lediglich Worthashes. Die Hashes können allerdings nicht wieder zurück in Wörter übersetzt werden. Das ist auch nicht notwendig und so können keine Klartextfragmente der ursprünglichen Webseiten auf den Rechnern der Peerbetreiber gefunden werden. Diese können daher auch nicht zur Verantwortung für die bei ihm/ihr gelagerten Wörter gezogen werden.



• Die Datenbank der RWIs ist eine hochspezialisierte Datenstruktur: eine AVL-Baumstrukur lässt ein geordnetes Aufzählen der URLs in RWIs zu und unterstützt damit effizientes Table Join, das für Kombinationssuche benötigt wird.

• Die Datenbank kann vom http-Webinterface des eingebauten http-Servers mit Servlet-Engine aus durchsucht werden. Der Webserver dient auch zur Administration des YaCy.

Das YaCy-P2P-Protokoll kontrolliert die Kooperation der Peers:

• RWIs werden in einer Distributed Hash Table (DHT) organisiert. Hierzu existiert ein komplexes Index-Wanderungskonzept zur Verteilung der Indizes zwischen den Peers.

• Peers können Teile eines Crawls an andere Peers abgeben. Es kann sichergestellt werden, dass URLs dabei nicht doppelt von verschiedenen Peers geladen werden.

• Jeder Peer publiziert seine eigenen Kontaktdaten regelmäßig an einen anderen Peer und pflegt eine Komplettliste aller Peers. Die Zuordnung von Peernamen zu IP wird unter anderem zur Auflösung von „<peer-name>.yacy“-Domains im Proxy genutzt.

Indexverteilung im YaCy-Netz

Es existieren verschiedene Möglichkeiten, wie ein YaCy-Peer seine Indexdatenbank erweitern kann. Im Überblick in der folgenden Grafik rechte Seite von oben nach unten:

Der Peer ist im Proxymodus und indiziert Seiten aus dem Proxycache. Der Peer führt einen lokal gestarteten Crawl aus.Der Peer erhält durch einen anderen Peer (der lokal einen Crawl ausführt) eine URL zum Indizieren zugewiesen und führt damit einen Remote Crawl aus.Der Peer bearbeitet eine lokal angestoßene Suchanfrage und fordert selektiv von anderen Peers RWI-Fragmente ein, die anschließend permanent in der lokalen Datenbank bleiben. Der Peer erhält von anderen Peers RWI-Fragmente zugewiesen, weil er für diese RWIs eine bessere Position entsprechend der DHT-Organisation hat. Alle Varianten der Indexgenerierung können simultan ablaufen.

Spezialisierung von Peers auf bestimmte RWI-Bereiche (linke Seite der Grafik): Simultan zur RWI-Gewinnung partitioniert jeder Peer seine RWI-Datenbanken in Teilmengen, die wiederum per DHT-Wanderung an einen anderen Peer zur permanenten Speicherung abgegeben werden. Dieser Zielpeer partitioniert ebenfalls seine RWIs, speichert die soeben zugewiesenen Indizes aber so lange, bis ein neuer Peer auftaucht, der eine ggf. noch bessere Position in der DHT besitzt. Dies geschieht beispielsweise, wenn das YaCy-Netz wächst und neue Peers hinzukommen.

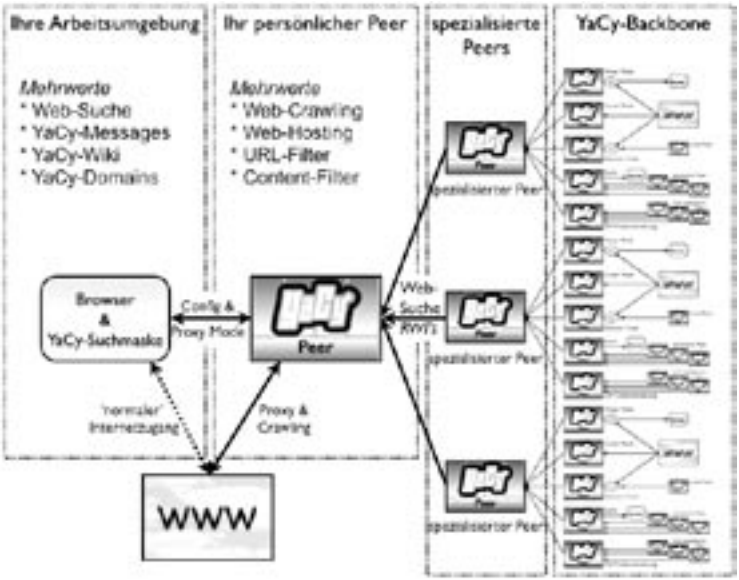
Die permanent ablaufende Indexwanderung sorgt für eine gute Durchmischung der RWIs, so dass recht schnell dafür gesorgt wird, dass weder feststellbar ist, ob ein Index, der an einem bestimmten Peer anzutreffen ist, auch von diesem erzeugt wurde, noch durch welche Methodik (Crawlen oder Proxy-Use) dies geschah. Dies ist sehr vorteilhaft für jeden Peerbetreiber, da dieser konzeptionell ausschließen kann, dass er für die Erzeugung einzelner Indizes verantwortlich ist.

Da die Indexverteilung ohne zentralen Server organisiert ist, gibt es keine technische Möglichkeit zur Zensur.

Verteiltes Crawlen und Suchen

Es ist eine Prämisse für das Projekt, Suchzeiten möglichst kurz zu halten. Suchanfragen werden nur zu Peers gesendet, die aufgrund der DHT-Konstruktion den entsprechenden RWI speichern sollen. Das funktioniert, weil RWIs ja im Vorfeld einer Suche bereits zu dem Peer, wo sie bei einer Suche erwartet werden, gewandert sind.

Mit steigender Peerzahl kann sich die DHT in dem wachsenden Netz weiter spezialisieren. Die Suchzeit für ein Wort steigt aber theoretisch nicht mit der Größe des Netzes, da der RWI ja immer nur an einer bestimmten Position



erwartet wird. Praktikabel ist aber ein gewisser Anstieg des Redundanzfaktors für die Anzahl der gleichzeitig abzusuchenden Peers, und eine Mischung mit Peers die einen besonders großen Index sowie Durchsatz besitzen.

Besondere Vorteile dieser dezentralen Suchtechnik

Der Benutzer kann selbst und ad-hoc bestimmen, welche Webseiten in den Index aufgenommen werden - Es können Seiten indiziert werden, auf die kein Link existiert und an die somit kein Crawler heranreicht.

Es bietet sich an, ein Ranking durch die Kooperation der Benutzer zu gestalten. Die Aufnahme in den Index geschieht ja durch die Aufmerksamkeit der YaCy-User auf bestimmte Webseiten. Dadurch wird die Indizierung auf populäre und für die Nutzer interessante Seiten fokussiert.

Bereits gefundene Webseiten werden vom suchenden Peer aus wieder per DHT-Wanderung verteilt. Dadurch verstärkt sich die Präsenz interessanter Suchergebnisse. Dies kann als implizite Moderation des globalen Index durch die Peeruser angesehen werden.

Inhalte können nicht global zensiert werden.

Es existieren Konzepte um die Peeranzahl stark zu vergrößern

YaCy könnte für folgende Bereiche interessant werden

Als Betriebssoftware für Internetcafes: Der caching Proxy ist nutzbringend zur Bandbreitenbegrenzung, und eine geplante Abrechnungsfunktion mit Clientkontrolle bringt einen zusätzlichen hohen Nutzwert. Einsatzgebiet wäre weniger Deutschland, eher weltweit. Die Internetabdeckung geschieht in vielen Regionen fast ausschließlich über Internetcafes und die Abrechung wird oft nur auf Papier ohne Software gemacht.

Als Browsererweiterung: YaCy könnte auch einen Browsercache auslesen, die Proxyfunktion ist dann hinfällig, aber alle anderen Vorteile – wie beispielsweise die Nutzung von .yacy-Domains – bleiben erhalten. Open-Source-Projekte wie Konqueror und Firefox könnten YaCy als Option in ihre Distribution aufnehmen. Dann hätte jeder Browser-Nutzer automatisch die komplette Kontrolle über Websuche und Inhalte des gemeinsamen Suchindex.

YaCy ist Open-Source (GPL-Lizenz), kompakt (rund 1 MB Download), portabel (Java), leicht zu installieren (nur auspacken, keine DB aufsetzten) und einfach zu betreiben. Wer mitmacht, der arbeitet aktiv an der Sicherstellung der Informationsfreiheit mit.

Links:

http://www.yacy.net/ – YaCy-Homepage
http://www.suma-lab.de/yacy – deutsche Doku

Softwarepatente update

Markus Beckedahl <markus@nnm-ev.de>

Die EU-Wirtschaftsminister haben in ihrer Sitzung am 7.3. die so genannte Software-Patentrichtlinie trotz massiver Kritik von Seiten der Parlamente, mittelständischer Unternehmen und der europäischen Zivilgesellschaft verabschiedet.

Bereits kurz vor Weihnachten hatte es einen Versuch gegeben, den umstrittenen Gesetzentwurf durch den EU-Rat zu bringen. Damals konnte ein Veto Polens das Durchwinken der Direktive im Landwirtschafts- und Fischereiausschuss verhindern. Diesmal setzte die Ratpräsidentschaft das Thema wieder als so genannten A-Punkt auf die Agenda der Wirtschaftsminister und machte damit den Weg für die Richtlinie frei. Als A-Punkte werden im EU-Jargon eigentlich diejenigen Sachverhalte bezeichnet, die als nicht weiter verhandlungsbedürftig und damit als abstimmungfähig gelten.

Wie wenig dies im Falle der so genannten „Richtlinie zu computerimplementierten Erfindungen“ zutrifft, dürfte angesichts der heftigen Debatte des letzten Jahres klar sein. Erst kürzlich hatte das EU-Parlament gefordert, den Gesetzgebungsprozess neu zu starten, da der vorliegende Entwurf der Direktive nach der EU-Erweiterung und der Neuwahl des EU-Parlaments nicht länger mehrheitsfähig ist. Noch am Vortag hatte das dänische Parlament ihrer Regierung den Auftrag gegeben, nicht abzustimmen. Dies wurde von der Luxemburgischen Ratspräsidentschaft nicht zugelassen, da die abzustimmende Position schon vor fast einem Jahr durch einen umstrittenen „Kompromiss” beschlossen worden sei. Die Abstimmung im EU-Rat kann deshalb nur als Nichtachtung des parlamentarischen Willens gewertet werden.

Verschiedene EU-Mitgliedsstaaten, darunter Polen und Dänemark, haben dem offiziellen Ratsbeschluss denn auch Zusatzerklärungen beigefügt. Darin wird insbesondere kritisiert, dass die Richtlinie in der derzeitigen Form die Patentierbarkeit von Computerprogrammen nur scheinbar verhindere und damit Wettbewerb im Softwarebereich nachhaltig behindere.

Der Begriff der Technizität gehört innerhalb der Software-Patente-Debatte zu den am heftigst debattierten Punkten. Auch der Deutsche Bundestag verwies darauf, dass die Patentierung von Computercode nur dann überhaupt legitim sein kann, wenn im einzelnen Fall der klare technische Beitrag mit einer naturwissenschaftlichen Wirkung erkennbar ist. Diese Abgrenzung ist, wenn überhaupt, dann nur sehr schwer nachvollziehbar und kann keinesfalls durch die nun verabschiedete Patent-Richtlinie geleistet werden.

Der Rats-Beschluss wird dem EU-Parlament nun in zweiter Lesung zugehen. Das Parlament hat zwar für einen Neustart gestimmt, allerdings ist es noch juristisch ungeklärt, wie und ob dieses Verfahren irgendeine Chance auf Realisierung hat, da die EU-Kommission diesem zustimmen müsste. Danach sieht es aber im Moment nicht aus. Sollte dies allerdings tatsächlich der Fall sein, kann der Beschluss des EU-Rates als Ausbau seiner Verhandlungsposition gesehen werden. Bei einem Neustart würde ein Vermittlungsverfahren zwischen den Positionen des EU-Parlaments und des EU-Rates greifen, um einen neuen Richtlinienentwurf zu schaffen. Allerdings stehen vor allem die „alten” EU-Mitgliedsländer einem Neustart sehr ablehnend gegenüber. Sie befürchten, dass ein Neustart von den „neuen” EU-Mitgliedsländern als Präzedenzfall genutzt werden könnte, um ande-

re Richtlinien auch neu zu diskutieren, da diese erst seit Mai vergangenen Jahres mitstimmen dürfen. So greifen mal wieder themenfremde politische und diplomatische Prozesse in einer Entscheidung, die massgeblich die Weichenstellungen für eine sich entwickelnde europäische Wissensgesellschaft schafft. Demokratie und Transparenz in reinster Form, die den Bürgern eigentlich kaum noch zu vermitteln ist...

Das EU-Parlament, welches sich mehrheitlich gegen die Richtlinien-Version des EU-Rates ausgesprochen hat, kann nun erstmal Zeit herausschlagen, indem der Parlamentspräsident eine gemeinsame Position zwischen Rat und Parlament verneint. Sollte das EU-Parlament wieder für seinen vor zwei Jahren in der „Ersten Lesung" und in anderer Besetzung beschlossene Position stimmen, würde auch ein längerer Vermittlungsprozess starten, der zumindest viel Zeit schinden würde.

Es bleibt zu hoffen und zu erwarten, dass die Abgeordneten die Gelegenheit zu gravierenden Änderungen am Entwurf nutzen werden. Sollte das Parlament diese letzte Chance, Änderungen in den Gesetzesentwurf einzuarbeiten, ungenutzt verstreichen lassen, wäre eine gravierende Schwächung der Softwareindustrie und der Innovationskraft die unmittelbare Folge. Noch sprechen sich auch die grossen Fraktionen für weitgehende Änderungen aus. Allerdings haben andere Abstimmungen über Richtlinien, wie z.B. bei der Biometrie gezeigt, dass es relativ einfach ist, die Abgeordneten gegen ihren Willen denoch zu einer positiven Wahl durch Erpressung zu „motivieren". Deshalb ist es wichtig, durch persönliche und freundliche Briefe und Faxe die EU-Abgeordneten mit guten Argumenten weiter zu sensibilisieren und sie in einer Ablehnung zu bestärken. Auch die Online- und Offline-Proteste müssen weitergehen, um Öffentlichkeit auf das Thema zu lenken und ein grösseres Bewusstsein der Bürger und Politiker für diese entscheidende Fragestellung zu schaffen.

http://www.netzpolitik.org

Survive Technology

Constanze Kurz <46halbe@weltregierung.de>

Die von-Neumann-Architektur ist Grundlage jedes handelsüblichen Computers. Über den Mann, der ihr seinen Namen gab, wurden Bücher mit wundersamen Anekdoten über seine erstaunlichen mathematischen Fähigkeiten, aber auch seine politische Skrupellosigkeit geschrieben. Sein Lebenswerk ist aufgrund seiner aktiven Mitwirkung am Bau von Massenvernichtungswaffen jedoch nicht unumstritten. Doch gleichgültig, wen man fragt, ob Bewunderer oder Kritiker, Fakt bleibt, daß John von Neumanns Genialität der Computerentwicklung einen entscheidenden Impuls gegeben hat.

Der gebürtige Ungar galt bereits früh als Wunderkind. Als er die Mathematik zu verstehen begann, eröffnete sich ihm eine neue Welt. Besonders die Logik hatte es ihm angetan. Noch als Teenager schrieb er seine erste mathematische Veröffentlichung. Entsprechend seinen überragenden Talenten verlief auch seine universitäre Karriere. Zeitgleich mit seinem Abschluß als Chemieingenieur in Zürich erwarb er seinen Doktortitel in Mathematik an der Universität in Budapest. Schnell überflügelte von Neumann seine Kollegen. Einige Professoren fürchteten seine außerordentliche mathematische Begabung. Einer bemerkte: „Wenn ich während einer Vorlesung ein ungelöstes Problem erwähnte, standen die Chancen gut, daß er nach der Vorlesung mit einer vollständigen Lösung zu mir kam, die er auf einem Stück Papier hingekritzelt hatte."

John von Neumann

Nachdem er 1929 von der Universität in Princeton zu einer Vortragsreihe eingeladen worden war, beschloß von Neumann, seine Karriere in den USA fortzusetzen. Das Land und die Lebensart der Menschen gefielen ihm, die aufstrebenden amerikanischen Universitäten versprachen Wissenschaftlern seines Formats attraktive Bedingungen. Auch aufgrund der zunehmend antisemitischen Stimmungslage gab er seine akademischen Positionen in Deutschland auf. Was als Besuch geplant war, wurde zu einem lebenslangen Aufenthalt. 1933, noch vor seinem dreißigsten Geburtstag, wurde er Professor am renommierten Institute of Advanced Study in Princeton.

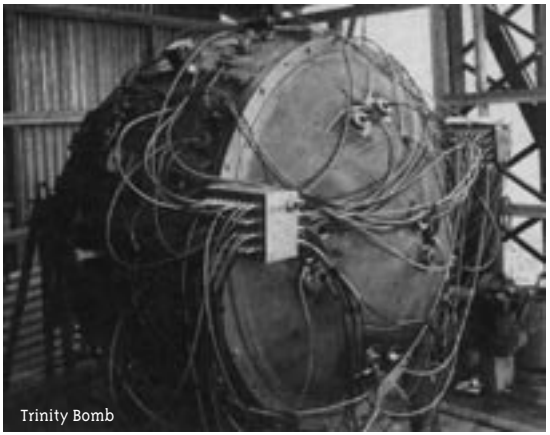
Der Krieg in Europa sollte der Karriere des jüdischen Mathematikers eine neue Richtung geben. 1941 hatte Präsident Roosevelt mit einem Budget von 2 Milliarden Dollar den Auftrag zum Bau einer nuklearen Waffe erteilt. Eingeladen von dem wissenschaftlichen Leiter des Manhattan-Projekts, Robert Oppenheimer, begann der ehrgeizige Akademiker ab September 1943, als Berater in Los Alamos zu arbeiten. Physiker forschten zu diesem Zeitpunkt bereits mehr als ein Jahr intensiv über die benötigte Menge rei-

nen Urans sowie über die mögliche Form und Größe der Bombe. Wie die meisten seiner Kollegen kannte von Neumann keine Skrupel, an dem Projekt mitzuarbeiten, das er als intellektuelle Herausforderung sah.

Die Berechnungen für die geplante Massenvernichtungswaffe erforderten komplexe mathematische Modelle. Raumfüllende Röhrenrechner halfen, die umfangreichen Kalkulationen auszuführen. Von Neumann nahm innerhalb des Projektes eine wichtige Position ein, denn er „besaß die wirklich bemerkenswerte Fähigkeit, blitzschnell Berechnungen im Kopf durchzuführen, besonders wenn er Größenordnungen grob überschlug." Seinen mathematischen Ergebnissen vertrauten die Physiker, Chemiker und Mathematiker in Los Alamos. Für Amerikas wichtigstes Militärprojekt sollte von Neumann entscheidende Bedeutung erlangen, denn er entwickelte in nur wenigen Monaten die Implosionsmethode als Zündungsmechanismus für die Plutoniumbombe, die Nagasaki zerstören sollte.

Von Neumann war in Los Alamos nicht nur ein führender Mathematiker, er setzte sich ebenso für seine politischen Überzeugungen ein. Er wirkte im Target Committee daran mit, strategisch und psychologisch geeignete Ziele für die Bomben auszuwählen, die Detonationshöhe mit dem größten Zerstörungspotential zu bestimmen und die erwarteten radiologischen

Effekte abzuschätzen. Das 13-köpfige Gremium beschloß im Mai 1945, welche japanischen Städte bombardiert werden würden. Als einer der wenigen Wissenschaftler setzte er sich selbst nach dem Abwurf der Atombombe für die Fortsetzung der nuklearen Waffenforschung und die Entwicklung der Wasserstoffbombe ein. Trotz der katastrophalen Konsequenzen in Japan sah von Neumann nukleare Abschreckung weiterhin als einzig wirkungsvolles politisches Mittel. Den ethischen Debatten über die Verantwortung von Wissenschaftlern angesichts des Ausmaßes der Zerstörung entzog er sich weitgehend.


Trinity Bomb

Seine überragenden intellektuellen Fähigkeiten und seine Erfahrungen im Atombombenbau ließen ihn schnell erkennen, daß die Entwicklung einer thermonuklearen Bombe ohne leistungsfähige Computer unmöglich sein würde. Im Gegensatz zu vielen Zeitgenossen verstand er die klobigen Geräte nicht nur als bloße Rechenwerkzeuge, er sah bereits ihr Potential als universelle Maschinen. „You insist that there is something a machine cannot do. If you tell me pre-

cisely what it is a machine cannot do, then I can always make a machine which will do just that."

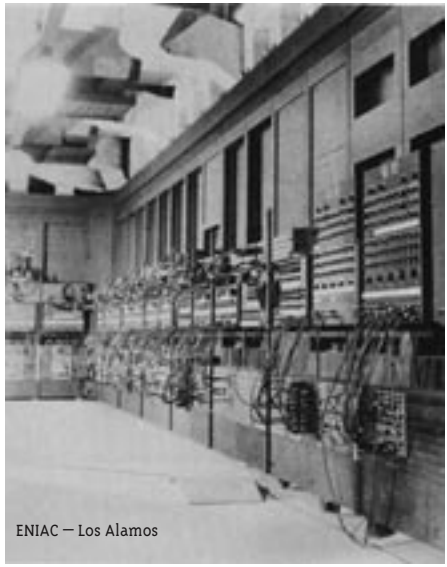
Mit seiner Arbeitsgruppe entwickelte er die fundamentale Idee des *stored program*. Er veröffentlichte 1945 den *First Draft of a Report on the EDVAC*, in dem er detailliert die mathematisch-logische Struktur der Maschine zeigte. Die von ihm vorgestellte sogenannte von-Neumann-Architektur wurde zum Synonym für die bis heute weitverbreitete Rechnerbauweise. Er begann, seinen politischen Einfluß geltend zu machen und Gelder für den Bau von Rechnern einzuwerben. Ab Ende 1945 förderte die US-Regierung insbesondere das ENIAC-Projekt in großem Umfang, denn der Computer sollte für die Berechnungen in Los Alamos genutzt werden. Die Fertigstellung des ENIAC verzögerte sich zwar um viele Monate, dennoch wurde die erste amerikanische Wasserstoffbombe gezündet, bevor russische Wissenschaftler eine thermonukleare Waffe bauen konnten.

Mit Beginn des Kalten Krieges war aus dem geachteten Akademiker, der sich selbst als *violently anti-Communist* bezeichnete, ein Regierungberater und Politiker geworden. Weltpolitische Entwicklungen beschäftigten ihn nun weit mehr als die wissenschaftliche Forschung. Er plante, die Entwicklung von nuklearen Raketenprogrammen und Interkontinentalraketen vorantreiben. Ab 1953 saß er dem sogenannten Teapot Committee vor, das die Aufgabe hatte, das technologische Potential der Sowjetunion zu evaluieren und Vorschläge zur Verteidigungspolitik der USA auszuarbeiten. Den politischen Höhepunkt seiner Karriere bildete 1955 die Berufung in die Atomic Energy Commission durch Präsident Eisenhower.

Die kurz darauf bei ihm diagnostizierte Krebserkrankung hinderte ihn, sein einflußreiches Amt lange ausüben zu können. Einer der brillantesten Köpfe seiner Zeit teilte sein Schicksal mit vielen Wissenschaftlern, die an der Atomwaffenforschung teilgenommen hatten. Der letzte öffentliche Auftritt des legendären Mathematikers John von Neumann fand 1956 im amerikanischen Machtzentrum in Washington, D.C., statt. Eisenhower verlieh ihm im Weißen Haus die Medal of Freedom für seine Verdienste bei der Entwicklung der Atom- und der Wasserstoffbombe. Im Februar des darauffolgenden Jahres starb der erst 53-Jährige in einem Militärkrankenhaus.

Quellen / Links

http://jvn.46halbe.org/



ENIAC — Los Alamos

Leichtes Spiel mit symboltables

Gerd Eist

Im Postfach der Datenschleuder häufen sich in letzter Zeit die Briefe lauter braver Enkel, die alle die Hilfe der Telefonbuch-CD in Anspruch nehmen wollen, um ihre Omis zu besuchen. Leider stellte sich schnell heraus, dass neuere Versionen der Telefonbuch-CD "Map'n'Route" einfach nicht mehr so, wie in der Datenschleuder 077 beschrieben, zu Rate zu ziehen sind. Nach reiflicher Überlegung beschloss die Redaktion Datenschleuder, den folgenden, anonym eingesendeten Bericht abzudrucken, um auch nachfolgende Generationen von Omis glücklich zu machen.

Versuchsprotokoll vom 14. April 2004, Protokollführer Gerd Eist.

Zutaten zum Nachvollziehen des Experiments:

- 0.2kg *„Telefonbuch Map&Route Herbst 2004"*
- 2.1kg Computer der Marke Apple
- 1.5827 * 10^{-23} kg `gdb-6.3.tar.gz`

Man führe die CD mit der Aufschrift "CD 1" in den dafür vorgesehenen Schlitz seines Rechners ein. Das Betriebssystem sollte diese nach `/Volumes/DasTelefonbuch` mounten.

Man begebe sich nun nach
`/Volumes/DasTelefonbuch/Das Telefonbuch.app/Contents/MacOS`,
gebe das Kommando
`nm Das\ Telefonbuch | grep Secret`
ein und staune über das Symbol
`000e4f58 t _SecretXorEncryption`.

Der `gdb` verrät einem an dieser Stelle, dass die Funktion ein Symbol namens `xorkey` referenziert. Dessen erste 29 Bytes werden mit denen eines Speicherblocks xored, der von der aufrufenden Funktion (namens `CTBMemory::UnCompress()`) hernach `inflate()` hingeworfen wird.

Der Inhalt des Symbols macht einen kurz stutzig, vor allem, wenn man gerade 10 Stunden durch PowerPC-Assembler ge-"singlestep"-t ist.

Aber da steht dann wirklich

`0x14ae10 "Just for Fun. Linus Torvalds."`

Der Rest ist nur noch ein Kinderspiel. Die grosse Datei™ heisst dieses Mal `/Volumes/DasTelefonbuch/atb/phonebook.db` und referenziert Strassenindizes in der Datei `/Volumes/DasTelefonbuch/atb/streets.tl`.

Wenn man nacheinander alle Chunks aus der `phonebook.db` extrahiert, erscheinen jeweils elf zusammengehörige Files mit je 3000 Einträgen. Die Strassen sind linear im File `streets.tl` gemappt und werden dezimal indiziert.

Fehleranalyse

Die komplette Symboltabelle in der MacOSX-Version der CD war ein gnädiges Geschenk. Die Wahl von MacOSX als Testumgebung war nicht ganz zufällig: in den Versionen für Linux und Windows war keine vorhanden.

In den neueren Ausgaben der CDs sind leider auch keine hausnummerngenauen Geokoordinaten im File `zip-streets-geo.tl` gespeichert, früher hiess dieses File noch `zip-streets-hnr-geo.tl`.

Der Autor freut sich schon auf das nächste Level der Herausforderungen seitens der Telekom.

Der westliche Brückenkopf des innovativen Technologieeinsatzes

AmP <mike@koeln.ccc.de>

Der C4 als Chaos Computer Club Colonia aka "Et kölsche Chaos" wurde im Jahre 1997 gegründet, nach dem Einschlafen des CAC aus den 80er Jahren. Seitdem ist viel passiert: Traf man sich anfangs noch in einer Eck-Kneipe, wurden später die Räume im aufstrebenden Köln-Ehrenfeld angemietet und über die Jahre immer wieder umgezogen und erweitert.

Treffen

Rund 42 Geeks sind derzeit Mitglied im C4 und treffen sich jeden Dienstag zum gemeinsamen Plenum. Aber nicht nur dienstags sind die Räume mit Leben gefüllt, donnerstags ist das selbstironisch genannte 'Freie Verpeilen' und auch am Wochenende ist das Chaos-Labor selten leer, dank der ausgezeichneten Infrastruktur von voll ausgestatterten Küche plus der Möglichkeit einer erfrischenden Dusche nach einer durchhackten Nacht. Diese Tage und Nächte werden genutzt, um an Projekten und der lokalen Infrastruktur zu hacken. Außerdem dienen die Donnerstage dazu, Nicht-CCCler einzuladen und ihnen die Aktivitäten des Club in seinen Räumen mal genauer zu zeigen.

OpenChaos

Jeden letzten Donnerstag im Monat sind besonders viele Leute im Club. Der Grund? OpenChaos! Das ist eine freie Vortragsreihe, in der ein Clubmitglied oder ein externer Referent über technische, politische oder gesellschaftliche Themen vorträgt. Der Eintritt ist frei und steht jedem offen. Unser OpenChaos ist somit die definierte Schnittstelle des C4 zur Aussenwelt.

Jugend

Eine andere Schnittstelle zu (jungen) Computerfreaks aus der Nicht-CCC-Welt ist das so genannte U23. Bisher wurde das "Hacker Jungend Forscht" zweimal veranstaltet, jeweils mit

durchschlagendem Erfolg. U23 richtet sich an Jugendliche unter 23, die sich für den innovativen Einsatz von Hard- und Software interessieren. So war das Thema des ersten U23 "Mini-Roboter bauen und programmieren". Im Jahr darauf wurden von den Nachwuchs-Hackern verschiedene Chat-Clients geschrieben, welche über ICMP-Pakte miteinander kommunizieren konnten. In diesem Projekt werden Teams gebildet und durch die freundliche Konkurrenz Lösungen selbst erstellt und nach wenigen Wochen der Öffentlichkeit vorgeführt.

Projekte

Auch wenn die Clubräume nicht zu einem Computer-Museum auswachsen sollen, so hegt und pflegt der C4 (Alt-)Hardware von fast allen großen Namen der IT-Branche, z.B. zahlreiche VAXen und VAX-Stations, eine IBM RS/6000, eine AS/400, zwei SGI Indys, eine NextStation und viele beschauliche große und kleine Absonderheiten der Computergeschichte mehr. Begleitet wird die Pflege durch Workshops der jeweiligen Crew. Hier wird auch gerne mal ein Lötkobeln gesehen.

Aber auch Software wird im C4 entwickelt, z.B. das Mail-Analyse Projekt "Schnucki" [1], welches ironischerweise es darauf anlegt, von Adress-Harvestern aufgefunden zu werden, um folgend Spam zu erhalten. Dieser Spam wird anschliessend analysiert.

Viel Erfahrung wurde auch beim Umgang mit Portscans aufgebaut. Das wirkt sich auch im Coden neuer Lösungen aus: "bannerscan" ist ein nmap-Frontend oder das distributed portscanning Tool "nippes", welches aus PORZ [2] hervorgegangen ist.

Gesellschaftspolitische Themen begleiten den C4 seit Jahren, wie z.B. Netzzensur oder die zunehmende Überwachung des öffentlichen Raumes durch Kameras (CCTV). Da der C4 schon früh mit Aktionen zu letztem Thema bekannt geworden ist, wird er im Rheinland als Know-how-Träger angesprochen und eingeladen.

Leben

In der letzten Zeit hat man das kölsche Chaos auch als "Chaos Comedy Club" kennengelernt. Einige Mitglieder mit schauspielerischen Fähigkeiten haben eine Serie geschaffen, die sich mit aktuellen politischen und gesellschaftlichen Themen auseinandersetzt, indem das jeweilige Geschehen durch Socken interpretiert wird. Die persiflierende Aufbereitung des bundesweiten Clublebens darf aber nicht darüber hinwegtäuschen, dass der C4 gerne auch auf anderen Ebenen Aufgaben übernimmt, ob es die von uns organisierte MV im letzten Jahr o.ä. sei oder der Aufbau unserer eigenen kleinen Bibliothek... In Köln passiert immer was.

Learned Lessons

- Socken sind toll
- Ein Besuch im Chaos-Labor lohnt
- Mehr Infos unter *https://www.koeln.ccc.de/*

[1] *http://koeln.ccc.de/schnucki/*
[2] *http://porz.org*

HACKtivitäten in London

BeF <bef@erlangen.ccc.de>

Wie manch einer vielleicht mitbekommen hat, geistere ich seit wenigen Monaten nicht mehr in Erlangen umher, sondern im schönen London. Nachdem es hier keinen CCC als Anlaufpunkt für einen gepflegten kognitiven Datenaustausch gibt, bin ich hier auf die eine und auch andere interessante Gruppierung gestoßen.

LONIX Meeting, zehn / nullvier

In London gibt es ungefähr fünf oder sechs Linux User Groups, die meisten davon sind irgendwelchen Himmelsrichtungen zugeordnet, wie z.b. die South London Linux User Group [1]. Recht zentral trifft sich immer eine nicht regional zugeordnete Linux User Group namens LONIX [2]. Das Treffen findet immer in Kneipen im Londoner Zentrum statt (siehe Bild auf der Webseite), wie auch letzten Oktober nach der Linux EXPO. Man trifft sich, trinkt - wer es mag - das ein oder andere Bier, unterhält sich nett über Gott und die Welt - oder auch über Linux. Aber DIE GEEKS verirren sich hier nur vereinzelt.

Freedom Hacklab, zehn / nullvier

Während des ersten LONIX Meetings drückte mir jemand einen Zettel mit einer Wegbeschreibung in die Hand. Also.. gleich mal nachsehen.. *such* und *find*, im ersten Stock eines Anarchy Bookshops, im Hintereingang in einer völlig unauffälligen Gasse, findet sich tatsächlich ein Raum. Ja, ein Raum, zunächst mal nichts weiter. Er ähnelt leicht dem mit alten, Computerkram vollgestopften EWerk-Raum (*), auch von der Größe. Personen, denen man auf den ersten Blick nachts lieber nicht begegnen möchte, finden sich hier tief in uralte Röhrenmonitore vertieft. Der leicht versiffte, aber durchaus gemütliche Raum hat, einschliesslich der an den Monitoren festklebenden Personen, meiner Meinung nach viel Potential zu vielem interessantem. ... und es gibt Internet.

LONIX Meeting, elf / nullvier

Wieder trifft man sich, dieses mal in Soho, an der Grenze zu China Town. Viele Gesichter kennt man noch vom letzten Treffen, man unterhält sich ganz nett, tauscht sich aus, und so geht der Abend irgendwann zu Ende. Das ist wirklich eine gemütliche Art, einen Abend zu verbringen. Nur zu empfehlen.

2600 Meeting, elf / nullvier

Das Trocadero ist ein Einkaufszentrum im Herzen von London, in der Nähe des Piccadilly Circus, wo sich am ersten Freitag im Monat diverse Leute treffen, denen schon ein gewisser Ruf vorauseilt. Überall auf der Welt gibt es solche Treffen an jedem ersten Freitag im Monat, nur in Birmingham - so wurde mir gesagt - findet das Treffen einen Tag später statt. Der Ort, ein Einkaufszentrum, wurde offenbar gewählt, um auch jüngeren Besuchern den Zugang zu dem Treffen nicht zu verbauen. Nach einer Stunde interessanter Diskussionen im stehen sind aber trotzdem alle Anwesenden in eine Kneipe umgezogen. Denn, man sollte es kaum glauben, hier sind alle süchtig nach Bier. Und trotzdem finden sich hier zum ersten mal ganz offensichtlich Leute, die von ihren gesnifften POP3 Accounts oder von irgendwelchen Schwachstellen in CISCO Routern reden. Ununterbrochen. Ein geniales Konglomerat von Wissen und Macht - für ein paar Stunden. Danach sind die meisten Anwesenden anonym geblieben, bis zum nächsten Freitag in vier oder fünf Wochen. Mindestens einer der überwiegend anonym gebliebenen wird sich - neben mir - auch auf

dem 21C3 zeigen. Wer sonst noch da war, und was das eigentlich soll, findet sich auf [3].

Freedom Hacklab – zwölf / nullvier

Jeden Samstag, manchmal auch öfter, ist dieser Raum für die Öffentlichkeit zugänglich - leider nur tagsüber. Nach dem Vorbild diverser Hacklabs in Spanien und Italien - siehe [4] - ist dies ein Anlaufpunkt für Wissbegierige, Open Source Fanatiker, Anarchisten, usw. . Unübersehbar ist auch eine gewisse Verknüpfung mit Indymedia - [5] - denen letztes Jahr in einer fadenscheinigen Polizeiaktion mehrere Server abhanden gekommen sind - [6]. Dieses eine Hacklab ist nur eines der "Projekte" des London Hacklab Collective (neuer Name seit wenigen Wochen). Weitere Projekte und Workshops präsentieren sich manchmal auch auf der ganz netten Webseite [7], wenn selbige nicht wieder wie Ende letzten Jahres gehackt wurde und sinnlos irgendwohin umleitet.

CCCongress in Berlin - zwölf / nullvier

Wer von euch auch auf dem letzten Kongress war, hat es vielleicht mitbekommen: Es wimmelte nur so von Internationalen Teilnehmern. Alleine bekannte Gesichter aus London zeigten sich schon mehrere fünf oder sechs, die vorher u.U. hier auf diversen geekischen Veranstaltungen gemütlich ein Bier tranken (**), manche davon waren sogar Vortragende.

2600 Meeting, nulleins / nullfünf

Eine sehr kurze Angelegenheit - ich war nur kurz da. das Trocadero füllte sich langsam mit Leuten, man unterhält sich kurz über dieses und jenes, auch über den Kongress. Für die Kneipentour war es dann nicht nur mir zu kalt - Januar und Winter und so.

LONIX Meeting, nulleins / nullfünf

Dieses mal findet das Treffen in einer riesigen, labyrinthartigen Kneipe in der Nähe des Covent Gardens statt, in der die Linux-Begeisterten

ungefähr ein Achtel für sich reserviert hatten. Wie immer ergaben sich nette Gespräche, dieses mal über OpenSource-Pilotprojekte, was Linux auf dem Mac macht und auch über die Qualität des Bieres. Besonders entgeisterte Blicke fanden sich, nachdem jemand Windows bootete und das auch noch entdeckt wurde. Ja, es ist zwar ein Kneipentreffen, aber die Notebooks fehlen auch hier nicht.

Freedom Media Hacklab – London Hacklab Collective, nulleins / nullfünf

Wer dieses eher versteckte Plätzchen in London findet und vorher schon davon gehört hat, der wird überrascht sein, dass sich jetzt fast jeden Tag Leute dort hin verirren. Nicht nur die Internetverbindung existiert noch, sondern auch diverse - überwiegend Linux rechnende - Computer überzuegen dort alle möglichen Besucher von Open Source Software. Gleich nebenan befindet sich, wie schon erwähnt, ein Buchladen für Anarchisten, die schon per Definition immer gerne viel wissen wollen und auch sehr an alternativen, nicht-kapitalistisch orientierten Computerthemen, wie Open Source, interessiert sind. Deshalb sind auch politische Diskussionen in dieser Atmosphäre von summenden Computern keine Seltenheit: “GPL supports that property is illegal...”

Greater London Linux User Group, nulleins / nullfünf

Wie mir jemand vom Hacklab berichtete, schwankt das Niveau der Diskussionskultur dieser netten Linux User Group stark von Treffen zu Treffen. Also besuchte ich eines Samstags, natürlich völlig vorurteilsfrei, den Keller der University of Westminster. Schilder warnten davor, dass Essen und Trinken in diesen Räumlichkeiten nicht erwünscht seien, es ist also ausnahmsweise mal eine Gruppe, die sich nicht treffen, um zu trinken. Die Webseite [8] der Greater London Linux User Group machte schon durch die Ankündigung eines Vortragsprogramms neugierig. Leider sagten fast alle Referenten ab, so dass hinterher alle Anwesenden dumm aus der Wäsche guckten, als jemand versuchte, sein Linux auf der Xbox zu präsentieren. Danach wandelte sich ein Spontanvortrag zu sudo noch in eine Diskussion über die Notwendigkeit des selbigen. Linux-Einsteigern wird in dieser Gesellschaft gerne weitergeholfen, Zeit für komplexere Themen ist ja immer noch bei weiteren Treffen.

2600, nullzwei / nullfünf

Jaja, ich weiß, schon wieder so eine Kneipengeschichte, aber man trifft ja auch nette Bekannte. Was man hier auch immer wieder antrifft, sind verirrte Seelen, die vergeblich versuchen herauszufinden, wie man ein Hacker wird. Ansonsten wird offensiv auf die nackte Realität aufmerksam gemacht, nämlich, dass in Großbritannien die Einführung von RFID-Ausweisen mit biometrischen Merkmalen eigentlich schon fast beschlossene Sache ist, dass in London fünfhunderttausend Kameras die Bevölkerung auf Schritt und Tritt überwachen, dass sowieso jeder schon mit Sendern - Telefon, RFID-Buskarte, ...- rumläuft, und dass auch Terroristen durch absurde über-Überwachung nicht mehr oder weniger enttarnt werden können.

Präkognition

Neben der günstigen Gegebenheit, dass im Zentrum von London alle paar Meter ein offenes WLAN nutzbar ist, wird es hier auch in Zukunft nicht langweilig. Interessante Projekte gibt es jedenfalls viele, z.B. zu offenen Funknetzen [9] oder Diskussionen rund um interessante Programmiersprachen [10].

(*) Clubraum des CCC Erlangen

(**) im UK wird nach dreiundzwanzig Uhr kein Alkohol mehr ausgeschenkt

[1] http://www.sl.lug.org.uk/
[2] http://www.lonix.org.uk/
[3] http://london2600.org.uk/
[4] http://www.hacklabs.org/index_en.html
[5] http://www.indymedia.org.uk/en/regions/london/
[6] http://www.heise.de/newsticker/meldung/51953
[7] http://hacklab.org.uk/
[8] http://gllug.org.uk/
[9] http://www.consume.net/
[10] http://python.meetup.com/41/

Bestellungen, Mitgliedsanträge und Adressänderungen bitte senden an:

CCC e.V., Lokstedter Weg 72, D-20251 Hamburg, Fax +49.40.401.801.41

Adressänderungen und Rückfragen auch per E-Mail an office@ccc.de

- Chaos CD Blue, alles zwischen 1982 und 1999 EUR 23 + EUR 3 Porto
- Alte Ausgaben der Datenschleuder auf Anfrage
- Datenschleuder-Abonnement, 8 Ausgaben
  Normalpreis EUR 32
  Ermäßigter Preis EUR 16
  Gewerblicher Preis EUR 50 (wir schicken eine Rechnung)
- Satzung und Mitgliedsantrag
  EUR 2,50 oder zum Selberausdrucken unter http://www.ccc.de/club/membership

Die Kohle

- liegt als Verrechnungsscheck bei
- wurde überwiesen am _____._____._____ an

  Chaos Computer Club e.V., Konto 59 90 90-201
  Postbank Hamburg, BLZ 200 100 20

Name:

Straße / Postfach:

PLZ, Ort

Tel.* / Fax*

E-Mail:

Ort, Datum:

Unterschrift

***freiwillig**

Etwa 20-fache Vergrößerung einer Farbkopie. Die gelben Punkte sind ein mit bloßem Auge unsichtbares Bitmuster, welches die eindeutige Kennzeichnung des Gerätes enthält. Näheres im Innern auf Seite 19.