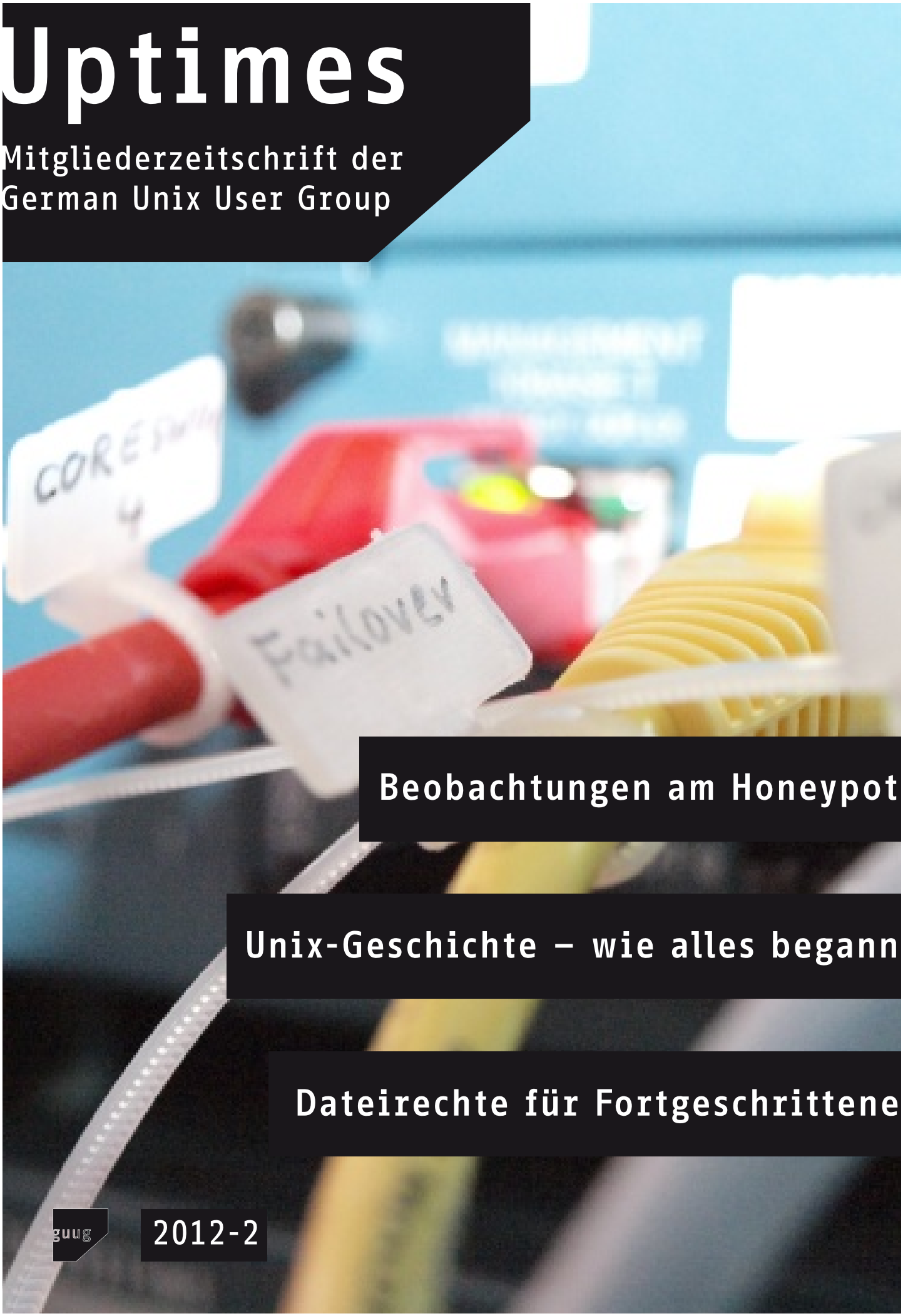
Uptimes

Mitgliederzeitschrift der
German Unix User Group



Beobachtungen am Honeypot

Unix-Geschichte – wie alles begann

Dateirechte für Fortgeschrittene

guug

2012-2

Inhaltsverzeichnis

Tag zusammen!
von Anika Kehrer **3**

Liebe Mitglieder!
von Wolfgang Stief, Vorsitzender des Vorstands **5**

sage@berlin
in Berlin macht das Jens Link **7**

Versammlungswesen
von Anika Kehrer **8**

Offene Sicherheit
von Sven Uebelacker **10**

Stille Beobachter
von Andreas Bunten und Torsten Voss **13**

Alles, was Recht ist
von Nils Magnus **18**

Geschichtsstunde
von Jürgen Plate **22**

T-Shirts ...
von Hella Breitkopf **27**

Nachlese
von Anika Kehrer **28**

Call for Presentations
vom Planungsteam **32**

Legenden eines alternden Beagles
von Snoopy **34**

Autorenrichtlinien **38**

Über die GUUG **40**

Impressum **41**

GUUG-Mitgliedsantrag **42**

Tag zusammen!
UpTimes mit neuer Redaktion

Auf GUUG-Listen und in Gesprächen brandete wiederholt die Diskussion auf, dass man doch gern wieder eine „richtige“ UpTimes hätte. Was „richtig ist“, weiß ich nicht. Doch hier ist die neue UpTimes.

von **Anika Kehrer**

Wohin gehen wir?
Frag mich was Leichteres!
Immer nach Hause.

Frei nach Novalis

Auf der neu belebten Redaktionsmailingliste ging es in den letzten sechs Wochen heiß her. Das ist zum Beispiel dem neuen Layout geschuldet: Robin Schröder hat aus LaTeX so einiges herausgekitzelt. Immerhin entstammt der erste Layout-Entwurf einem konventionellen Office-Programm. Wie würden die darin so leicht von der Hand gehenden Fancy-Elemente wie farbige Aufmacherkästen oder Kopf- und Fußzeilen aussehen, umgesetzt im reduzierten LaTeX-Stil? Ich finde: Richtig elegant. Ich bin echt begeistert. Doch ich hoffe vor allem, dass es Euch gefällt.

In der neuen UpTimes ergänzt die neue Rubrik *Vereinsleben* die bisherigen Fachbeiträge, zum Beispiel mit Portraits der Regionalgruppen. Den Anfang macht Berlin. Ein Artikel zeigt, was sich in den Vorstandssitzungen so getan hat. Am liebsten wird die Vereinsrubrik mit eigenen Artikeln von Euch der Platz für eine publizistische Informations- und Meinungsplattform in der GUUG, neben den Fachartikeln.

Für Fachartikel gibt es die Themenrubrik. Zum Thema *Geschichte* holt Jürgen Plate in dieser Ausgabe die lange, tolle Unix-Geschichte ins Rampenlicht. In Sachen *Security* zeigen Andreas Bunten und Torsten Voss, was alles um ihre Honeypots herumschwirrte, und Sven Uebelacker stellt OpenDNSSEC vor. Als *Sysadmin* hält Nils Magnus mit Feintuning von Dateirechten ein Plädoyer für diese grundlegende Unix-Technik. Aber das ist noch nicht alles. Für Humoristisches und Literarisches gibt es ab sofort die Rubrik *DownTimes*. Snoopy ist mit neuen Legenden dabei! Bei den *Events* sprechen Collagen von den letzten FFG in Bildern, für die nächsten ruft der CfP. Im ganzen Heft gibt es mit Geek and Poke jetzt immer wieder mal zwischen den Artikeln eigene Comics in der UpTimes. Kurz: Wehe, jemand findet diese Ausgabe doof. :-)

Und wenn doch? Der Entstehungsprozess der UpTimes ist mit Mailingliste und Wiki-Dokumentation transparent. Jeder kann sich einbringen: Mit Redaktion, Schlussredaktion, LaTeX-Gestaltung und Bilderrecherche erobern sich redaktionell interessierte GUUG-Mitglieder einen Platz im Impressum. Als Autor bieten sich mit Veranstaltungsberichten und Buchrezensionen – Empfehlungen wie Kritiken – überschaubare Textmengen. Wer für ein ganzes Thema brennt, schöpft mit einem Fachartikel aus den Vollen. In der UpTimes steht Euch die ganze Unix-Welt offen, von Hardware bis Warehouse. Und bei 700 GUUG-Mitgliedern liest das ein kompetentes Fachpublikum. Ihr.

Am wichtigsten ist in der UpTimes neben der fachlichen Diskussion das Vereinsleben. Daher schickt Themen, Texte, Erörterungen, Howtos, Bilder, Links, Rätsel, Witze. Schickt, was Ihr den anderen in der UpTimes zeigen wollt. In Eurer UpTimes. Denn für niemanden sonst ist dieses Heft gemacht.

Ihr erreicht die Redaktion unter <redaktion@uptimes.de>. Schickt Inhalte und Feedback an mich – <kehrer@guug.de> – oder gleich direkt an die Liste. Abonniert die Liste, wenn Ihr Euch wirklich einbringen wollt. Tippt, malt, lötet, stanzt. Egal wie: Lasst von Euch hören!

Über Anika

Anika Kehrer hat Literaturwissenschaft, Geschichte und Politik studiert und wohnt in der Heimat Berlin sowie in der Jobheimat München. Sie beschäftigt sich als freie Journalistin mit Informationstechnik, bis hin zur Elektrotechnik, und mit Open-Source-Technologie, von Community Culture bis Versionsverwaltungssoftware.

Liebe Mitglieder!
Grußwort vom Vorstand

von **Wolfgang Stief**, Vorsitzender des Vorstands

Wie? Was? Warum? ist Stentors Redekreis.
Gruß, Bitte, Rat, Erzählung, Wünsche, Klagen,
Vorwürfe, Schmeichleien, sind alles bei ihm Fragen;
Und wenn er euch nichts mehr zu fragen weiß,
Fragt er: Was sollt' ich sie doch fragen?

Friedrich Wilhelm Gotter

„Hurra! Endlich wieder eine UpTimes!", werden viele von euch begeistert ausrufen. Mich freut das auch sehr. Nachdem etliche Versuche eines Neustarts in den vergangenen Jahren meistens aus Zeitgründen gescheitert sind, haben wir für diese Ausgabe ein bisschen Geld in die Hand genommen und eine Chefredakteurin bezahlt, die den Laden zusammenhält und alle Mitwirkenden vorantreibt. Danke Anika!

Langjährige Mitglieder und UpTimes-Leser sehen sofort, dass das Layout ganz anders ist als früher. Der Vorstand hat in 2010/2011 die Überarbeitung des äußeren Erscheinungsbildes in Auftrag gegeben, an das sich die UpTimes jetzt sinnvollerweise anpasst. Und irgendwann wird schließlich auch der Webauftritt entsprechend gestaltet.

Eine weitere Neuerung ist, dass die UpTimes jetzt als PDF erscheint. Wir denken, das ist zeitgemäß. Unter den Mitgliedern gibt es beide Lager: Solche, die seit Jahren ohnehin nur noch am Bildschirm lesen (und begeisterte E-Book-Leser sind), und die Fraktion, die auf dem morgendlichen Weg ins Büro lieber ein Stück Papier zum Lesen in der Hand hält.

Die Verbreitung als Bytes ist jedoch für die GUUG logistisch um vieles einfacher, außerdem mit deutlich weniger Kosten verbunden. Die wir andererseits wieder in eine bezahlte Chefredakteurin investieren können. :-) Ein der Papierfraktion geschuldeter Kompromiss ist das Layout im A4-Format, das auch gedruckt ordentlich aussieht. Zum Lesen am Bildschirm würde sich ja etwas in 16:9 oder 4:3 besser eignen... Von einer Veröffentlichung als PDF versprechen wir uns außerdem eine weitere Verbreitung, als das bisher mit gedruckten Ausgaben möglich war.

Die UpTimes entsteht auch weiterhin in LaTeX. Prinzipiell haben wir damit die Möglichkeit, zusätzliche E-Book-Formate (ePub u. a.) anzubieten. Unter den derzeit Engagierten ist keiner, der sich in diesen Formaten zuhause fühlt, weshalb es aktuell keine Ambitionen in diese Richtung gibt. Wenn jemand diese Lücke füllen möchte, sind wir sehr dankbar.

Ab sofort soll die UpTimes wieder mehrfach im Jahr erscheinen, an einen Plan wagen wir uns im Moment aber noch nicht. Da halten wir es so, wie die Debianer: „It's done, when it's done." Wer Ideen oder Beiträge hat, meldet sich bitte per Mail an <redaktion@uptimes.de> oder <uptimes@guug.de>. Zum Ende dieser Ausgabe gibt es auch Autorenhinweise. Für Fachartikel können wir sogar Honorar bezahlen. Insbesondere möchte ich Mitglieder aufrufen, die auch Fachbücher schreiben, uns das mitzuteilen. Wir stellen die Bücher gerne in der UpTimes vor.

Entgegen der früheren Ausrichtung mit überwiegend Fachartikeln soll in Zukunft auch die Vorstandsarbeit verstärkt in der UpTimes Platz finden. Damit soll der Verein transparenter werden.

Zum Abschluss dieses Grußworts möchte ich dem Team hinter dieser Ausgabe danken für wirklich beeindruckendes Engagement: Allen voran Anika, die mit fordernden Mails ein beachtliches Tempo vorgelegt hat. Danke an Robin, der die ganzen Layout-Ideen nach LaTeX überführt hat. Wer LaTeX kennt, weiß, dass das nur bedingt geeignet ist, um eine Zeitschrit zu layouten. Großer Dank auch an Hella, die sich auch dieses Mal um das Titelbild und die Titelgestaltung gekümmer hat. Und schließlich noch Danke an Oliver Widder, der uns seine Comics auf Anfrage von Anika unter CC-BY-SA für zukünftige Ausgaben zur Verfügung stellt.

Nun wünsche ich euch viel Spaß mit dieser Ausgabe und freue mich schon auf die Arbeit am nächsten Heft, das noch in diesem Jahr kurz vor Weihnachten erscheinen soll.

Über Wolfgang

Wolfgang Stief hat Linux während des E-Technik-Studiums kennen und schätzen gelernt (Kernel 0.99). Seit 1998 verdient er sein Geld überwiegend mit Solaris, Storage-Systemen und verteilten Dateisystemen. Wenn er gerade mal nicht beruflich oder für die GUUG unterwegs ist, züchtet er Gemüse, treibt sich in sozialen Netzwerken herum oder macht Musik.

sage@berlin
Die Lokalgruppe der Hauptstadt

In mehreren Städten treffen sich GUUG-Mitglieder und andere Unix-Interessierte regelmäßig in Person. Dafür muss einer die Organisation übernehmen.

in Berlin macht das **Jens Link**

Ick sitze da und esse Klops.
Uff eenmal kloppt's.
Ick sitze, staune wundre mir,
uff eenmal jeht se uff, de Tür.
Ick jehe hin und kieke,
und wer steht draußen? Icke.

Berliner Spruch

Wie lange gibt es die Berliner Regionalgruppe?
Jens: Die Idee entstand im Herbst 2005, das erste Treffen war im Januar 2006.
Wie setzen sich die Berliner (Abbildung 1) zusammen?
Jens: Hauptsächlich Admins, hängt aber auch zum Teil von den Vorträgen ab.

Abbildung 1: sage@guug Berlin traf sich am 07. Juni 2012 zum Vortrag DNSSEC von Lutz Donnerhacke in den Telekom Innovation Laboratories (T-Labs). Anwesend waren mehr als 20 Personen. Schnappschuss: Florian Streibelt.

Wo in Berlin trefft Ihr Euch, und warum?
Jens: Vorträge finden meistens an der Technischen Universität Berlin statt, wir waren aber auch schon zu Vorträgen bei Cisco oder IBM.
Sind Eure Treffen eher Stammtische, oder gibt es bei Euch auch Vorträge oder Workshops?
Jens: Wir versuchen, jeden Monat einen Vortrag anzubieten. Termin ist der erste Donnerstag im Monat. In den meisten Fällen klappt das auch. Nach dem Vortrag gehen wir dann zum gemütlichen Teil in einem Restaurant über. Fällt der Termin auf einen Feiertag, verzichten wir auf den Vortrag. Außerdem treffen sich immer noch ein paar Leute zum Sysadminday und zu einem spontanen Treffen vor Weihnachten.
Habt Ihr Lieblingsthemen oder Themen, die häufiger vorkommen?
Jens: Wir versuchen, die Themen möglichst weit zu streuen. Es kommt aber immer auf die Vortragenden an. Wenn Wünsche der Teilnehmer kommen, versuche ich auch, einen passenden Sprecher zu finden.
Wieviele Teilnehmer kommen in der Regel zum Berliner Treffen?
Jens: Zum Vortrag so zwischen zehn und dreißig. Zum Bier danach dann immer so zwischen zehn und zwanzig.
Wie kamst Du dazu, die Berliner Lokalgruppe unter Deine Fittiche zu nehmen?
Jens: Da heißt die Antwort leider: Weil es sonst keiner macht. Wobei ich im Monat so zehn bis 15 Minuten für die Orga brauche. Ein fester Ort für Vorträge und ein großer Pool von Vortragenden erleichtern die Sache ungemein.

Über Jens

Jens Link ist freiberuflicher Computer-Geek, der sich auf Netzwerke hauptsächlich von Cisco spezialisiert hat. Zum Monitoren nutzt er freie Software wie Nagios oder Cacti. Besonders interessiert er sich für IPv6-Security.

Versammlungswesen
Aus MV und Vorstand 2010 - 2012

GUUG-Vereinsmitglieder diskutieren auf Mailinglisten und treffen sich in Regionalgruppen. Mitgliederversammlung und Vorstandssitzungen sind hingegen langweilig. Oder?

von **Anika Kehrer**

Zweck des Vereins ist,
Forschung, Entwicklung und Kommunikation
offener Computersysteme,
insbesondere Unix-artiger Betriebssysteme
und dazugehöriger Software zu fördern.

§ 2 GUUG-Satzung

Knapp 1.300 Subskribenden tummeln sich derzeit auf den <sage@guug.de>-Listen, dem virtuellen Salon Unix-geneigter Admins, Architekten und Programmierer aus dem deutschsprachigen Raum. Die Listen entstanden nach dem Vorbild der *System Administrators Guild* der Usenix vor etwa 12 Jahren als Projekt im Schoß der German Unix User Group: ein 1984 eingetragener Verein, in dem sich eine Gruppe Menschen mit obengenannten Attributen zum Zwecke gemeinsamer Handlungsfähigkeit zusammenschloss.

Die SAGE-Listen – genauso wie die regionalen Treffen der GUUG – setzen keine Vereinsmitgliedschaft voraus. „SAGE-Onlys“ erhalten jedoch keinen Einblick in Vereinsinterna wie Finanzen, und sie besitzen keine Stimme bei Wahlen oder Entscheidungen über GUUG-Aktivitäten. Welche diese sind, ist hier den Protokollen aus Mitgliederversammlung, Vorstandssitzungen und anderen Treffen entnommen, ausgewählt und anonymisiert.

Teilhabe in Zahlen

Von den derzeit rund 700 Vereinsmitgliedern subskribieren mit Stand von 2012 etwa die Hälfte die Vereins-Diskussionsliste <guug-members@guug.de>. Allerdings entstand diese Liste erst lange nach Gründung des Vereins. Inzwischen kommt jedes neue Mitglied auf diese Liste, nur die Altmitglieder wurden nie nachgezogen. Genau wegen dieser Divergenz hat der Vorstand neuerdings die Liste <mitglieder@guug.de> ins Leben gerufen, deren Abonnentenliste sich aus der Mitgliederdatenbank speist und somit wirklich jeden erreicht – doch dazu mehr unter dem Punkt ‚Kommunikation im Verein‘.

An der jährlichen Mitgliederversammlung nahmen 2010 rund 40 Personen teil, 2011 waren es 50. Etwa zwei Dutzend Vereinsmitglieder bringen sich aktiv in das Vereinsleben ein, zum Beispiel als Organisator von Regionaltreffen, als Rechnungsprüfer, in der Wahlkommission oder direkt in dem 9-köpfigen Vereinsvorstand.

Während sich die Mitglieder jährlich zusammenfinden – oft an demselben Ort wie das Frühjahrsfachgespräch – trifft sich der Vorstand mehrmals pro Jahr. Er führt die Feder bei Details des Vereinslebens, zum Beispiel IT-Infrastruktur, Event-Beteiligungen oder strategische Maßnahmen, und bereitet Beschlüsse der Mitgliederversammlung vor.

Seit rund zwei Jahren hat sich der Vorstand zum Ziel gesetzt, die Vereinsarbeit transparenter zu machen. Zweck ist, die Kommunikation zwischen Spitze und Basis zu beleben – und so für mehr aktive Teilhabe zu interessieren. Auf der Vorstandssitzung vom 29. Oktober 2011 in Köln und vom 7. Januar 2012 in Bochum fielen dafür einige wichtige Entscheidungen.

Öffnung des Vorstands

Die interne Kommunikation des Vorstands kann mit Vorstandsbeschluss von Oktober 2011 auch von GUUG-Mitgliedern geteilt werden, die nicht die Funktion eines gewählten Vorstandsmitglieds erfüllen. Wer möchte, bittet den Vorstand um Einbindung in die Kommunikation. Der Vorstand entscheidet dann über die Aufnahme in die Lese-/Schreibgruppe für Liste und Wiki.

Diese Methode der Einbindung auf Antrag halten die Vorständler für sinnvoller, als eine eigene neue Aktiven-Mailingliste der GUUG einzurichten. Mit dem Veto-Recht behält sich der Vorstand vor, seinen geschützten Kommunikationsraum nicht unkontrolliert nach außen zu öffnen. Und das Stimmrecht bei Abstimmungen bleibt den von den Mitgliedern gewählten Vorstandsmitgliedern vorbehalten. Der Vorstand betont aber:

> Mitgliedern soll damit ermöglicht werden, den Verein aktiv mitzugestalten und die Vorstandsarbeit transparenter zu erleben. Wir freuen uns explizit auf aktive Teilnahme an den Diskussionen.

Die GUUG-Spitze beschloss außerdem im genannten Meeting 2011, seine Protokolle ab sofort im internen Vereins-Wiki zu veröffentlichen, um eine weitere Kommunikationslücke zu den anderen Mitgliedern zu schließen. Genau wie bei der Öffnung der Kommunikationswege ist auch hierbei die Herausforderung, empfindliche Daten zu schützen, damit die Persönlichkeitsrechte vor dem dann breiteren Publikum gewahrt werden.

Diese Bearbeitung des Protokolls, das ja in seiner primären Funktion auch erst einmal geschrieben werden will, stellte sich als deutlicher Mehraufwand heraus, wie in der darauffolgenden Vorstandssitzung am 7. Januar 2012 in Bochum zur Sprache kam. Doch das Gremium bekräftigte aber noch einmal die Notwendigkeit der Umarbeitung – und hinkt daher in der Ausführung bislang etwas hinterher. Die entsprechende Wikiseite hat der GUUG-Vorstand bisher daher noch nicht sinnvoll bekannt geben können.

Kommunikation im Verein

Schon auf der Mitgliederversammlung von März 2011 wurde die GUUG-Präsenz bei Twitter, Identica und in der Blogosphäre offiziell. Anlässlich des Frühjahrsfachgespräches 2012 erhielt dann der neue GUUG-Blog einen Schwung vor-Ort-Berichte aus der Feder der GUUG-Presse-Mitarbeiterin Corina Pahrmann. Weitere GUUG-Mitglieder nutzen das Blog seitdem zum Beispiel für Ankündigungen von Regionaltreffen. Auch Facebook- und Google-Plus-Seiten vervollständigen inzwischen den Auftritt der GUUG im Web 2.0. (Anmerkung der Redaktion: Kann mal bitte jemand einen Artikel darüber schreiben, was das Web 3.0 oder gar 4.0 sein soll?)

Eine weitere Neuerung im Jahr 2012 ist der GUUG-Newsletter, den der Vorstand in den beiden schon genannten Sitzungen diskutiert und beschlossen hat. Demnach gibt es jetzt die neue Read-only-Mailingliste <mitglieder@guug.de>, die für die Kommunikation des Vorstandes an die Mitglieder gedacht ist: Der Newsletter soll quartalsweise über die Vereinsaktivitäten informieren. Damit niemand den Newsletter als Ärgernis empfindet, will der Vorstand für seine Relevanz und saubere Formatierung Sorge tragen. Seine erste Ausgabe ging im April 2012 über die Bühne, Verzeihung, Liste.

Auch die monatlichen Termine der Lokal- und Regionaltreffen gehen fortan über den Mitglieder-Verteiler und entfallen auf der Liste <guug-members@guug.de>. Im Gegensatz zu der neuen Ankündigungsliste, die nur als Leseliste gedacht ist, soll die GUUG-Member-Liste interaktiver Diskussion zwischen allen Vereinsmitgliedern vorbehalten sein.

Auf der Vorstandssitzung von Oktober 2011 wurde auch die bezahlte Chefredaktion der UpTimes beschlossen. :-) Da diese neue UpTimes-Redakteurin übrigens die Vereinsmailinglisten nicht lesen kann, sind die Vereinsmitglieder gefragt, hier in der UpTimes zu kommentieren, einzuordnen oder einfach zusammenzufassen, was ihnen auf den Listen bemerkenswert erscheint. Das gilt genauso für die SAGE-Listen bei fachlichen Themen. Wer will, kann etwas einfach runterschreiben, einschicken – und in der UpTimes einen neuen Rahmen geben.

Offene Sicherheit
OpenDNSSEC hilft, DNSSEC zu verwalten

OpenDNSSEC ist bei der Suche nach einer geeigneten DNSSEC-Lösung eine wertvolle Option. Die aktive Weiterentwicklung lässt auf viele neue Features in naher Zukunft hoffen.

von **Sven Uebelacker**

One Key to rule them all,
one Key to find them,
one Key to bring them all
and in the Resolver bind them.

modified from Lord of the Rings, Miek Gieben, NLnet Labs

Seit Sommer 2011 ist auch die .de-TLD-Zone von DENIC signiert, wie viele andere TLD-Domains. Netzwerkadministratoren können DNSSEC [1] daher für die eigenen Domänen ausrollen. Viele Betreiber von authoritativen Nameservern, besonders kleinere, stellt das vor die Herausforderung, Anpassungen an ihrem derzeitigen Workflow vorzunehmen und zusätzliche Ressourcen einzuplanen. OpenDNSSEC ist ein Hilfsmittel, DNSSEC fehlerfrei umsetzen, damit Kunden in den Genuss von überprüfbaren Resource Records kommen.

Sinn von DNSSEC

Dass DNS für immer mehr Dienste Verwendung findet, hat wohl niemand anfangs vermutet. Seine Ausfallsicherheit durch Dezentralität (zum Beispiel Anycast-DNS) und die weltweit eindeutige Hierarchie brachte und bringt viele in Versuchung. So präsentierte zum Beispiel Werner Koch auf dem Frühjahrsfachgespräch 2012 das Projekt STEED [2], wo er Informationen zu X.509-Zertifikaten und OpenPGP-Schlüsseln über DNS-Einträge verteilt. Damit will er einen neuen, benutzerfreundlicheren Ansatz von Ende-zu-Ende-Verschlüsselung etablieren. Denn das Vertrauensmodell vom OpenPGP-Web-of-Trust, so Koch, habe sich laut des ACM-Beitrages „Why Johnny can't encrypt: a usability evaluation of PGP 5.0" [3] nicht bis zum Endbenutzer durchgesetzt. Gerade wegen der weitreichenderen Einsatzzwecke von DNS ist eine Absicherung mit DNSSEC daher dringend geraten.

Überprüfbare und verlässliche DNS-Antworten dienen zunehmend auch der Absicherung von Infrastrukturen: Es dämmt bekannte DNS-Angriffe ein, wie zum Beispiel Dan Kaminskys *DNS Cache Poisoning* [4], was zu einem zusätzlichen Schub im Einsatz von DNSSEC führte. Es kann die viel diskutierten Begehrlichkeiten staatlicher Einrichtungen, das Internet kontrollieren oder gar zensieren zu wollen, minimieren, ebenso wie auch MITM-Angriffe, wenn es gelingt, die letzte Meile der DNSSEC-Validierung zum Benutzer zu überwinden. Denn das Vertrauen in DNS-Antworten sollte streng genommen nicht an den ISP delegiert werden. Bis eine Validierung auf Client/Benutzerebene stattfindet, wird aber leider wohl noch einige Zeit vergehen.

DNSSEC sichert nicht nur bisherige, auf DNS basierenden Dienste ab, sondern es kann auch zur Grundlage für darauf aufsetzende neue Entwicklungen werden, wie beispielsweise die IETF-Arbeitsgruppe *DNS-based Authentication of Named Entities* (DANE) [5] zeigt.

Deployment von DNSSEC

Große Unternehmen, die bereits Anycast-DNS-Server oder zumindest ein einwandfrei funktionierendes Konfigurationsmanagement betreiben, werden keine großen Schwierigkeiten haben, DNSSEC zu deployen. Wie bei geheimen Schlüsseln üblich, sollten diese auch hier in einer geschützten Umgebung liegen: Die Signierung sollte also auf einem dedizierten System passieren und nicht auf produktiven, weltweit erreichbaren authoritativen Nameservern. Falls ein Konfig-Management vorhanden ist, ist dies meist einfach für das Verteilen der signierten Zonen erweiterbar. Jedoch müssen sich kleine und mittlere Unternehmen sowie private DNS-Serverbetreiber ohne Konfigmanagement Gedanken machen, wie sie die Zonen ausrollen wollen – von einem einfachen Rsync von einem abgeschotteten (virtualisierten) Signiersystem bis zum Aufbau eines Konfigmanagements ist alles möglich.

Unbedingt zu verhindern ist das Ausliefern von falsch signierten Zone-Dateien. Denn fehlge-

schlagene Validierungen können einige Verwirrung stiften. Zusätzlich müssen die Zonen regelmäßig neu signiert werden, wobei ein kurzer Übergangszeitraum mit zwei öffentlichen Schlüsseln angeboten werden sollte. Andernfalls könnte auf noch nicht aktualisierten Resolvern oder in Caching-Nameservern die Situation eintreten, dass alte öffentliche Schlüssel nicht mit den neuen Signaturen validieren, oder umgekehrt. Diese hohe Anforderung an die Korrektheit der (Signatur-) Daten als auch die neue Komplexität ist manuell nicht mehr zu bewältigen.

Empfehlung: OpenDNSSEC

Neben einigen kommerziellen Lösungen wie Secure64 existieren auch FLOSS-Softwareprojekte für die Organisation des DNSSEC-Workflows. Ein beliebtes Tool ist OpenDNSSEC [6], das zum Beispiel Matthijs Mekking von NLnet Labs ebenfalls auf dem Frühjahrsfachgespräch 2012 vorgestellt hat. NLnet Labs [7] ist eingefleischten DNS-Admins durch *ldns* [8] oder dem darin enthaltenen *drill* (*dig* für DNSSEC) bekannt. OpenDNSSEC wird kontinuierlich weiterentwickelt. Derzeit wird es etwa von Surfnet.nl [9], dem niederländischen Forschungs- und Wissenschaftsnetz, und von der schwedischen .se-ccTLD-Verwaltungsstiftung [10] produktiv eingesetzt.

Das BSD-lizensierte OpenDNSSEC automatisiert den kompletten Signierprozess und überblickt alle DNSSEC-Schlüssel und -Zonen. Es springt in die Bresche, wo Standardwerkzeuge für DNSSEC nicht weiterhelfen können, wie sie etwa in *bind9* enthalten sind: Schlüsselverwaltung, Policy Handling oder Hardware Acceleration. Für Letzteres liefert das Projekt auch gleich die freie Software SoftHSM [11] mit, die ein HSM (Hardware Security Module) emuliert. Denn OpenDNSSEC nutzt eine PKCS#11-Schnittstelle zur ausgelagerten Schlüsselverwaltung. Dies ist für Entwicklungs- und Testumgebungen, aber auch für spätere Skalierbarkeit oder bei gewachsenen Sicherheitsanforderungen eine sinnvolle Herangehensweise.

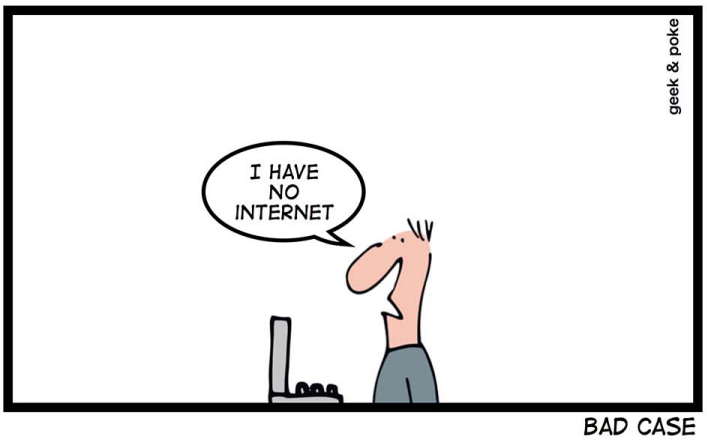
Die aktuelle, stabile Version ist derzeit 1.3.9. In der 1.2er Version ersetzten die Entwickler den ursprünglich in Python implementierten Signer durch einen in C geschriebenen. Version 1.3 steigerte die Performance beim Signieren um ein vielfaches. Im vierten Quartal 2012 wird Version 2.0 mit vielen neuen Features erwartet, etwa Unterstützung von unsignierten Zonen (pass-through) und verschiedene Rollover-Mechanismen (für Policys und Algorithmen). Die darauffolgende Version 2.1 soll die dynamischen Updates erweitern und einen Datenbankadapter schaffen. OpenDNSSEC und SoftHSM existieren für verschiedene GNU/Linux-Distributionen und BSD-Varianten. SoftHSM wurde auch nach Microsoft Windows portiert. Details zur weiteren Entwicklung enthält der Release Plan [12].

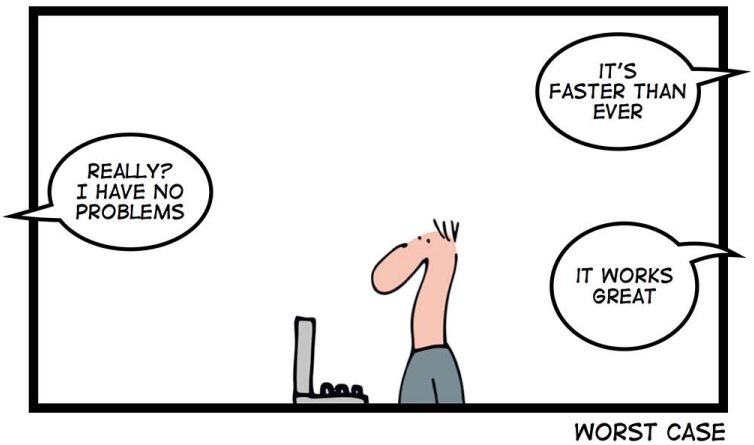
Literatur

[1] Website von DNSSEC: https://www.dnssec.net/
[2] Website von STEED: http://g10code.com/steed.html
[3] ACM-Beitrag zu OpenPGP: https://dl.acm.org/citation.cfm?id=1251421.1251435
[4] Dan Kaminskys DNS Cache Poisoning: http://www.kb.cert.org/vuls/id/800113
[5] IETF-Arbeitsgruppe DANE: https://datatracker.ietf.org/wg/dane/charter/
[6] Website von OpenDNSSEC: https://www.opendnssec.org/
[7] Website von den NLlabs: https://www.nlnetlabs.nl/
[8] Website von Ldns: https://www.nlnetlabs.nl/projects/ldns/
[9] Niederländisches Forschungs- und Wissenschaftsnetz Surfnet: https://dnssec.surfnet.nl/
[10] Schwedische .se-ccTLD-Verwaltungsstiftung: https://www.iis.se/
[11] Website von SoftHSM: https://www.opendnssec.org/softhsm/
[12] Release Plan OpenDNSSEC: https://www.opendnssec.org/about/release-plan/

Über Sven

Sven Uebelacker ‚arbytet' bei einem IT-Sicherheitsunternehmen in Hamburg. IRL tanzt er Lindy Hop und verbringt den Sommer gern chillend grillend auf seiner Tux-Terrasse. Seit 1995 arbeitet er mit Unix-/Linux-Betriebssystemen und dem Schwerpunkt IT-Sicherheit und Datenschutz. Zurzeit ist er Mitorganisator des Hamburger Software Freedom Days (Samstag, 15.09.2012, http://sfd-hamburg.de/).





Stille Beobachter
SSH-Angreifern mit Honeypots nachstellen

Wie sich Angreifer verhalten, liest man entweder in Büchern, oder man beobachtet sie bei frischer Tat. Das ermöglichen absichtlich schwache Lock-Systeme – sogenannte Honeypots.

von **Andreas Bunten** und **Torsten Voss**

Ein Honigtöpflein vom Netze hervor
war fröhlich gesprossen im löchrigen Flor.
Da kam ein Schlawiner und naschte fein –
die müssen wohl beide füreinander sein.

Frei nach J. W. von Goethe

Sieht man sich als Administrator die Logfiles eines SSH-Servers an, stößt man in der Regel auf viele Anmeldeversuche, die nicht von den eigenen Benutzern stammen. Es handelt sich in den meisten Fällen nicht um versehentliche Verbindungen, sondern um Angreifer, die gezielt versuchen, den Server zu übernehmen. Man nennt diese Angriffe *SSH Account Probes* oder auch Brute-Force-Angriffe.

Dahinter steht, dass Angreifer sich zum Opfer-System verbinden und eine Reihe von typischen Passwörtern durchprobieren. Grundsätzlich ist dies ein recht simpler Angriff. Bei der Aufklärung einer Vielzahl solcher Angriffe fiel den Autoren auf, dass immer wieder die gleichen Tools und Skripte zum Einsatz kamen. Weiterhin machte es den Anschein, dass der Großteil der Angriffe auf nur wenige Gruppen zurückzuführen sei. Diese Erkenntnis war für die Autoren der Ausgangspunkt, um SSH-Angriffe mit Hilfe von Honeypots näher zu untersuchen.

Honeypots

Lance Spitzner vom Honeynet-Projekt [1] definiert einen Honeypot als eine Ressource, deren Wert darin liegt, von einem Angreifer missbraucht zu werden. So allgemein diese Definition klingt, so unterschiedlich sind konkrete Honeypot-Implementierungen. Um eine Klassifikation durchzuführen, kann man unterschiedlich vorgehen (siehe [2] als ausführliche Referenz). Wir unterscheiden im Folgenden Honeypots einerseits nach ihren Möglichkeiten zur Interaktion, andererseits danach, wie leicht Angreifer den Honeypot als solchen erkennen.

Low Interaction Honeypots

Um erste Informationen über die SSH-Angreifer herauszufinden, setzten wir zunächst *Low Interaction Honeypots* ein. Sie beherbergten einen OpenSSH-Server mit leicht modifiziertem Quellcode: Eine erweiterte Protokollierung notiert das eingegebene Passwort und verhindert, dass ein Angreifer sich tatsächlich anmelden kann. Der modifizierte OpenSSH-Server läuft auf Systemen, die ansonsten nicht per SSH erreichbar wären. Zu den Systemen zählt eine Reihe von Cisco-NAS-Systemen (NSLU2) an privaten DSL-Anschlüssen. Die kleinen NAS-Systeme lassen sich unter Debian Linux betreiben und verbrauchen angenehm wenig Strom.

Rang	Konto	Zugriffe	Passwort	Versuche
1	root	193.134	123456	19.500
2	test	5.634	password	9.475
3	admin	5.074	1234	5.209
4	bin	3.729	12345	3.981
5	oracle	3.260	123	3.805
6	guest	2.329	qwerty	3.460
7	user	2.246	abc123	3.340
8	postgres	1.608	test	3.192
9	mysql	1.558	root	2.539
10	nagios	1.453	1q2w3e	2.342

Tabelle 1: Top 10 der angegriffenen Konten und verwendeten Passwörter

Wir registrierten im Frühling 2012 insgesamt 602.698 illegitime Anmeldeversuche. Aufgrund der Quell-IP und des zeitlichen Zusammenhangs lassen sie sich zu 1909 Angriffen zusammenfassen. Wir fanden so unter anderem heraus, welche Konten mit welchen Passwörtern angegriffen werden. Die jeweiligen Top 10 sind Tabelle 1 zu entnehmen. Sie überraschen nicht übermäßig. Angriffsziel ist vor allem das Konto *root* - dieses ist schließlich auf jedem Linux- und UNIX-System vorhanden. Die Passwörter sind typische 'Musst Du aber

sofort ändern'-Passwörter oder stellen Muster auf der Tastatur dar.

Die Anzahl der Anmeldeversuche pro Angriff ist sehr unterschiedlich. Wenige Angreifer probieren sehr viele Konten und Passwörter aus. Der Rekordhalter versuchte 30.689 Anmeldungen. Über 70 Prozent der Angriffe versuchten es allerdings weniger als 100 mal (Abbildung 1).

Manche Muster in den Daten zeigen, welche Passwörter zu meiden sind: Zum Beispiel entspricht bei einem Drittel aller Anmeldeversuche das versuchte Passwort dem Namen des Kontos. Ein Konto *webmaster* mit dem Passwort *webmaster* wird demnach sehr wahrscheinlich von einem Angreifer kompromittiert.

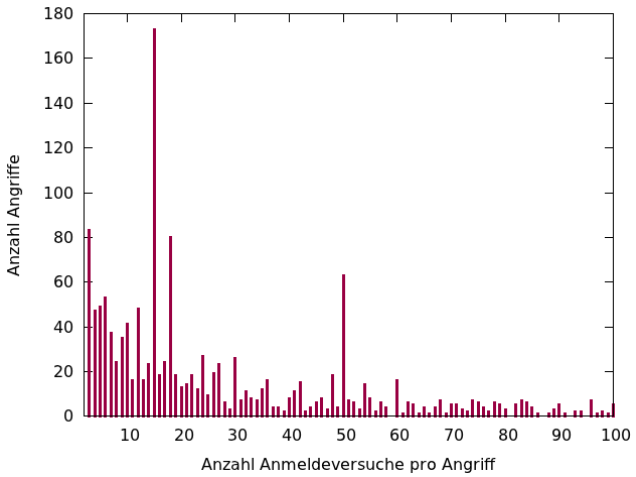


Abbildung 1: Histogramm zur Anzahl von Anmeldeversuchen pro Angriff bis 100 Anmeldeversuche.

Manche Funde in den Daten sind allerdings so verwunderlich, dass man sich fragt, ob es sich um Konfigurationsfehler auf Seiten der Angreifer handelt. Bei 177 Anmeldeversuchen wurden beispielsweise Passwörter verwendet, die länger als 30 Zeichen waren, obwohl die Chance, ein so langes Passwort zu erraten, extrem gering ist. Weiterhin zielten einzelne Angriffe auf die Konten *Terminator*, *Tolkien* oder *username*, die wahrscheinlich nicht sehr verbreitet sind.

Die IP-Adressen der Angreifer lassen sich Ländern zuordnen. Die zehn häufigsten Länder, aus denen die Verbindungen stammen, zeigt Tabelle 2.

Rang	Land	Anzahl
1	China	598
2	Korea	286
3	USA	213
4	Russland	79
5	Indien	57
6	Deutschland	54
7	Brasilien	50
8	Niederlande	43
9	Türkei	42
10	Großbritannien	41

Tabelle 2: Top 10 der Länder, aus denen die Account-Probe-Angriffe stammen

Virtuelle Honeypots

Wir wollten mehr darüber herausfinden, was die Angreifer nach erfolgreicher Anmeldung mit den Systemen vorhaben. So kamen wir zu virtuellen Honeypots. Wir verwendeten die Software *kippo* [3], um ein verwundbares Linux-System mit schwachem Root-Passwort zu simulieren. Es hilft, erfolgreiches Eindringen vorzutäuschen: In einer Shell-Session kann sich der Angreifer nun innerhalb des simulierten Dateisystems umsehen und Befehle ausführen, etwa um Downloads zu starten und zusätzliche Werkzeuge auf dem System zu installieren. Heruntergeladene Programme auszuführen ist jedoch nicht möglich, da sonst ein Ausbruch aus der simulierten Umgebung droht.

Kippo protokolliert, wann die Angreifer welche Eingaben tätigen, sodass wir den Angreifern über die Schulter sehen. Ein typisches Vorgehen besteht darin, zuerst den Download einer großen Datei anzustoßen, um die Online-Verbindung zu testen. In den meisten Fällen diente dafür das Service Pack 3 für Microsoft Windows 2000. Einige wenige Angreifer nutzten auch den Service von Cachefly.net, um die Anbindung zu testen. Nach diesen ersten Tests interessierten sich die Angreifer vor allem für Denial-of-Service-Tools und die Software psyBNC. PsyBNC leitet eine Verbindung per Internet Relay Chat (IRC) um und verschleiert so die Quelle der Verbindung. Vereinzelt wurde auch versucht, einen TeamSpeak-Server aufzusetzen, Webmin zu installieren oder einen Half-Life-Gameserver in Betrieb zu nehmen.

Als grundlegende Erkenntnis haben wir festgestellt, dass es zwei Arten von Verbindungen zum Honeypot gibt. Zum einen sind die bereits bekannten Account Probes zu sehen. Bei erfolgreicher Anmeldung beendet der Angreifer die Verbindung allerdings sofort wieder. Zum anderen erfolgt etwas später eine neue Verbindung von

einem anderen System. Dabei kommt sofort das richtige Konto mit dem korrekten Passwort zum Einsatz. Nun werden auch interaktiv Befehle eingegeben, und in der Regel landet weitere Software auf dem Honeypot.

Wir schließen daraus, dass die ersten Angriffe von bestimmten Systemen aus durchgeführt werden – wahrscheinlich von kompromittierten Systemen. Danach werden erfolgreich erratene Anmeldedaten geerntet und möglicherweise von dem eigenen System der Angreifer aus genutzt, um sich interaktiv mit dem neuen Opfer-System zu verbinden.

Rang	Land	Anzahl
1	Rumänien	1184
2	USA	336
3	Italien	311
4	EU	167
5	Spanien	149
6	Deutschland	77
7	Libanon	69
8	Großbritanien	60
9	Mazedonien	58
10	Frankreich	52

Tabelle 3: Top 10 der Länder, aus denen die interaktiven Sitzungen der Angreifer stammen

Tabelle 3 ordnet die IP-Adressen der interaktiven Sitzungen Ländern zu. Dies scheinen die eigenen Systeme der Angreifer zu sein, und das Ergebnis unterscheidet sich überraschend stark von Tabelle 2.

```
sales:/dev/shm# shutdown h
Try `shutdown --help' for more information.
sales:/dev/shm# shutdown  -h 99999999
Try `shutdown --help' for more information.
sales:/dev/shm# shutdown -h 99999999
Try `shutdown --help' for more information.
sales:/dev/shm# reebot
bash: reebot: command not found
sales:/dev/shm# reboot

Broadcast message from root@sales (pts/0) (Sat Feb 11 00:07:24 2012):

The system is going down for reboot NOW!
Connection to server closed.
localhost:~#
```

Abbildung 2: Ein Angreifer, der mit einem virtuellen Honeypot verbunden ist, versucht das System zu stoppen und hat die Verbindung vermeintlich getrennt.

Leider ist die Simulation des Opfer-Systems durch Kippo nicht vollständig. Die vom Angreifer auf das System gebrachte Software ist nicht ausführbar, und manche üblichen Befehle funktionieren nicht. Speziell gibt es keinen Editor in Kippo. Die Angreifer bemerken daher in der Regel bald, dass etwas nicht stimmt. In Abbildung 2 ist die Sitzung eines Angreifers zu sehen, der versucht, das System zu stoppen, um einen Neustart einzuleiten. Natürlich führt der Honeypot nicht wirklich einen Neustart durch und gibt nur eine Meldung aus, die den Angreifer vermuten lässt, dass die Verbindung getrennt wird. Tatsächlich bleibt er weiter mit dem Honeypot verbunden, und es wird jede weitere Eingabe protokolliert.

High Interaction Honeypots

Wir wollten herausfinden, was die Angreifer mit den auf das Opfer-System gebrachten Werkzeugen konkret vorhaben. Dazu mussten wir einen Schritt weitergehen und *High Interaction Honeypots* einsetzen. Wir überlassen den Angreifern damit die volle Kontrolle über ein echtes System.

Das System protokolliert ausführlich, was passiert, und ist aus Sicherheitsgründen nur mit einem Quarantäne-Netzwerk verbunden. Ein Gateway zum Internet überwacht alle Verbindungen. So ist es möglich, ausgehende Angriffe zu stoppen, bevor Dritte geschädigt werden. Zur Sicherheit wird der High Interaction Honeypot nur beaufsichtigt betrieben. Sobald ein Angreifer Zugriff erlangt, löst das Alarm aus, und das Vorgehen lässt sich live beobachten.

Der Betrieb eines High Interaction Honeypots ist mit erheblichem Aufwand verbunden, sodass den Autoren noch nicht viele Sitzungen der Angreifer vorliegen. Die Versuche dauern noch an. Die Technik ist allerdings sehr vielversprechend, da die Angreifer nun fast die gleichen Möglichkeiten besitzen, die ihnen ein echtes System bietet. Die Angreifer nehmen weitere Downloads vor, installieren Rootkits und fehlende Pakete über das Paket-System, und sie legen neue Benutzer an. Weiterhin ist nun der konkrete Einsatz von Angriffswerkzeugen zu beobachten (Abbildung 3). Eine beispielhafte Angreifer-Session zeigt ein Video unter [6].

```
.aly/ssh-scan
.aly/pscan2
.aly/s
.aly/a
root@fileserver:/var/www/httpd# cd .aly/
root@fileserver:/var/www/httpd/.aly# ls
a  gen-pass.sh  pass_file  pscan2  s  ss  ssh-scan
root@fileserver:/var/www/httpd/.aly# ./a 212

scanning network 212.*.*.*
usec: 0, burst packets 65535
using inteface eth0
using "(tcp[tcpflags]=0x12) and (src port 22) and (dst port 9061)" as pcap filte
r
my detected ip on eth0 is 192.168.1.13
capturing process started pid 6566
scanning 212.0.0.*
scanning 212.0.1.*
scanning 212.0.2.*
scanning 212.0.3.*
scanning 212.0.4.*
scanning 212.0.5.*
scanning 212.0.6.*
```

Abbildung 3: Die Angreifer starten auf dem High Interaction Honeypot eines ihrer Angriffs-Werkzeuge.

Exkurs: So vermeide ich, Opfer zu werden

Folgende Punkte helfen, Angreifer von den eigenen Servern fern zu halten:

Starke Passwörter durchsetzen

Verwenden alle Benutzer starke Passwörter, droht keine Gefahr durch SSH Account Probes. Admins setzen zum Beispiel mittels geeigneter PAM-Module eine Black List offensichtlich schlechter Passwörter, oder sie erzwingen eine gewisse Komplexität. So ist es möglich, ein Konto nach einer bestimmten Zahl von Fehlversuchen temporär zu sperren. Die Angreifer können dann allerdings das Root-Konto durch ständige Anmeldeversuche blockieren.

Erreichbarkeit einschränken

Es kann sehr helfen, das eigene Netzwerk von außen zu untersuchen, um herauszufinden, wie viele SSH-Server man tatsächlich betreibt. Wer nicht alle SSH-Server aus dem Internet erreichen muss, sollte zu diesen auch keine Verbindungen zulassen.

Logs zentral auswerten

In Firewall-Logs fällt es auf, wenn jemand mehrfach in kurzer Zeit Verbindungen zu den SSH-Servern aufbaut. In den Logfiles der SSH-Server sind vielfache Fehlversuche verdächtig. Idealerweise laufen die Logs zentral zusammen und werden automatisch analysiert. So lassen sich Angreifer und eventuelle Angriffe von eigenen bereits kompromittierten Systemen erkennen und blockieren.

Key-Fingerprints kontrollieren

SSH-Keys stellen eine gute Alternative zu Passwörtern dar. Wurden diese sicher generiert, lassen sie sich nicht in der gleichen Weise erraten wie Passwörter. Es sollte aber unbedingt bedacht werden, dass Anfreifer auch SSH-Keys stehlen können. Die Schlüssel-Dateien sollten daher immer mit einem Passwort versehen werden. Um eine Anmeldung per SSH-Key zu ermöglichen, werden die Fingerprints der Keys in *authorized_keys*-Dateien hinterlegt. Es ist ratsam, die Fingerprints der verwendeten SSH-Keys zu kennen und gelegentlich zu prüfen. Andernfalls ist es einem Angreifer nach einem erfolgreichen Einbruch möglich, den Fingerprint seines eigenen SSH-Keys beispielsweise zu denen des Backup-Systems hinzu zu schmuggeln, um sich von nun an unbemerkt anzumelden [4]. Weiterhin lässt sich mit einer derartigen Kontrolle sicherstellen, dass Benutzer bereits gestohlene SSH-Keys nicht weiter verwenden.
Ist ein System kompromittiert, empfiehlt es sich, dort nicht einfach die Dateien der Angreifer zu löschen und zur Tagesordnung zurückzukehren. Es ist praktisch nie klar, welche Manipulation die Angreifer vornahmen und ob sie nicht sogar noch Zugriff haben. Ein kompromittiertes System sollte auf jeden Fall mittelfristig neu installiert werden [5].

Fazit

Die anfängliche Vermutung der Autoren, dass es nur wenige Gruppen von Angreifern gibt, haben die bisherigen Ergebnisse bestätigt. Vor allem der hohe Anteil an rumänischen Systemen stellt uns allerdings noch vor ein Rätsel. Wir haben festgestellt, dass die eigentlichen Account-Probe-Angriffe praktisch immer von kompromittierten Systemen ausgehen. Dies sollten die betroffenen Administratoren bedenken, wenn Kontakt zur vermeintlichen Quelle eines Angriffs aufgenommen wird.

Um die letzten offenen Fragen zu klären, geht der Einsatz der Honeypots weiter. Unser Ziel ist, die Daten der verschiedenen Klassen von Honeypots zu kombinieren und beispielsweise nur bestimmte Angreifer in den High Interaction Honeypot zu locken, die vorher mit den anderen Honeypots identifiziert wurden.

Neben den erlangten Informationen über die Angreifer und ihr Vorgehen hat der Einsatz von Honeypots weitere Vorteile. Zum einen ist es spannend, den Angreifern über die Schulter zu schauen und ihnen dadurch individuell zu begegnen. Zum anderen erlangen wir Zugriff auf die Werkzeuge der Angreifer: Denial-of-Service-Tools, Rootkits und Scan-Skripte. Durch Tests lassen sich die von den Werkzeugen hinterlassenen Spuren ermitteln, um besser auf echte Sicherheitsvorfälle reagieren zu können. So kann ein High Interaction Honeypot, zu dem der Angreifer gerade Zu-

gang hatte, gezielt auf Spuren des Angriffs untersucht werden, um das eigene Vorgehen bei einem Sicherheitsvorfall zu trainieren.

Literatur

[1] Homepage des Honeynet-Projekts: http://honeynet.org/
[2] Niels Provos und Thorsten Holz: Virtual Honeypots, From Botnet Tracking to Intrusion Detection. Addison-Wesley Longman, 2007
[3] Homepage der Honeypot-Software Kippo: http://code.google.com/p/kippo/
[4] Andreas Bunten: 99 Backdoors on my UNIX Host. Frühjahrsfachgespräch der GUUG, März 2012
[5] Andreas Bunten: Kompromiss nach Kompromittierung? S. 57ff. In: kes – Die Zeitschrift für Informations-Sicherheit, Ausgabe 4, August 2011
[6] Session eines SSH Angreifers: http://vimeo.com/39218661

Über Andreas und Torsten

Andreas Bunten (Bild) ist seit 1996 im Bereich Unix-Administration und Security tätig. Er war acht Jahre Mitglied des Emergency Response Teams des Deutschen Forschungsnetzes im DFN-CERT und hat dabei eine Vielzahl kompromittierter Systeme untersucht. Seit drei Jahren berät und unterstützt er die Kunden der Controlware GmbH im Bereich IT-Sicherheit.

Torsten Voss (ohne Bild) studierte technische Informatik und arbeitet seit 2006 beim DFN-CERT im Emergency Response Team des Deutschen Forschungsnetzes. Sein Schwerpunkt ist die Vorfallsbearbeitung und die Aufklärung von Kompromittierungen.

Alles, was Recht ist
Dateirechte im Unternehmenseinsatz

Das Prinzip von Dateirechten kennen Unix-Admins seit Jahrzehnten. Trotzdem reizen nur wenige ihre Möglichkeiten aus. Ein Plädoyer für eine unterschätzte Technik.

von **Nils Magnus**

Vielleicht können wir wollen,
sollen aber nicht?
Vielleicht müssen wir können,
dürfen aber nicht?
Gehe nicht verloren in
der Welt der Zugriffsrechte.

Frei nach Fred Fredson

Linux-Distributionen sind – zumindest was Benutzerrechte angeht – oft für den Desktop-Betrieb eingerichtet. Admins gehen dafür meist von einem oder nur wenigen Anwendern auf einer Maschine aus. Diese Anwender trennen sie einerseits untereinander und andererseits von den Systembenutzern. Die Bordmittel der Lese-, Schreib- und Ausführungsrechte erreichen bereits eine recht ordentliche Separierung von privaten Daten – im Hinblick auf Datenschutz und Privatsphäre eine nützliche Sache. Diese Rechte haben dazu beigetragen, dass Unix einen so guten Ruf im Hinblick auf Sicherheit und Betriebsstabilität genießt.

Das betrifft nicht nur die Funktionsweise der Rechte, die im Kern über Jahrzehnte unverändert blieb. Eine ebenso wichtige Rolle spielt, welche Dateien und Verzeichnisse welche Berechtigungen, Eigentümer und Gruppen erhalten. Hier hat sich in den letzten Jahren enorm viel getan. Das führt dazu, dass moderne Derivate dem Systemverwalter nur wenig Anlass bieten, Systemverwaltungsaufgaben unter Rootrechten durchzuführen.

Spezialfall Serverinstallation

Auf Server-Installationen liegen die Anforderungen jedoch anders. Hier gilt es weniger die Anwender untereinander zu trennen, sondern eher, Applikationen vor externen Angreifern zu schützen, die sich allein statistisch vorhandener Software-Schwachstellen bemächtigen.

Auf diese Frage hat die Betriebssystemzunft eine ganze Reihe von Antworten parat: Audit-Rahmenwerke wie SELinux oder App-Armor überwachen Zugriffe, erweiterte Attribute verwalten granular Rechte, Container- und Virtualisierungslösungen sperren ganze Anwendungen in besser handhabbare Schachteln ein. Trotz dieser Techniken, die alle ihre Berechtigung haben, bieten aber bereits die ganz konventionellen Unix-Zugriffsrechte einen guten Grundschutz.

In Umgebungen wie den Webanwendungen eines LAMP-Stack beispielsweise besteht das System oder die VM aus den Standardkomponenten des Betriebssystems sowie individueller Anwendungssoftware, etwa ein CMS, ein Blog oder ein fachspezifisches Programm. Was davon über das Paketmanagement aus den Repositories der Distributionen kommt, ist im Hinblick auf Rechte meist durchdacht. Das Ändern von Pfaden oder Berechtigungen führt fast immer zu Verschlimmbesserungen.

Andere Dateien erhalten ihre Rechte von Installern, individuellen Werkzeugen zur Software-Verteilung oder bestenfalls einem Konfigurationswerkzeug wie Puppet und Consorten. Gerade bei Individual-Software stellt sich die Frage, wie nicht nur Zugriffsrechte, sondern auch Eigentümerverhältnisse, Gruppen sowie die Gruppenmitgliedschaften von Usern zu vergeben sind. Hier ist der lokale Administrator gefragt, im Spagat zwischen Absicherung und Wartbarkeit zu balancieren.

User-Typen

Es lassen sich drei Arten von Benutzern unterscheiden. Erstens sind dort die *System-User*, die von Paketen automatisch bei der Installation angelegt werden. Beispielsweise bringt das als Online-Hilfe meist eingesetzte Paket Mandb seinen eigenen User *man* mit. Diesem User gehören dann gecachte Manpages unterhalb von `/var/cache/man`. Die System-User haben einen festen Zweck für einen Systemdienst und erlauben in der Regel kein interaktives Login. Falls notwendig, erhalten autorisierte persönliche Accounts die Möglichkeit, etwa mittels *sudo* oder entsprechenden Gruppenrechten für diese User zu agieren. User, die eine Mitgliedschaft in der Gruppe *admin*

besitzen, dürfen Sudo benutzen, um so einzelne Kommandos mit Root-Rechten auszuführen.

Zweitens repräsentieren die *persönlichen User* natürliche Personen. In der Regel sind sie die einzigen, denen ein interaktives Login gestattet ist. Auch Wartungsarbeiten sollten Administratoren unter persönlichen User-Accounts durchführen, um so per Auditing nachträglich nachzuvollziehen, wer was getan hat. Im Serverbetrieb sind den Accounts Fachaufgaben zugeordnet, sodass hier kaum personenbezogene Daten anfallen, die gesonderte Schutzmaßnahmen entsprechend §3 Abs. 1 BDSG (Bundesdatenschutzgesetz) nötig machen.

Für persönliche User ist ein Verzeichnisdienst mit Abstand die skalierbarste Variante. Mit einigen Abstrichen ist die Verwaltung auch durch kopierte und synchronisierte Benutzerdatenbanken möglich, etwa `/etc/passwd` und `/etc/shadow`. Einen Mittelweg stellt das Verteilen der Benutzerdaten durch das Konfigurationsmanagement dar.

Die dritte Form von Benutzeraccounts sind individuelle *Funktions-User*: Sie implementieren Zugriffsschutz zwischen Anwendungen und Diensten, sind aber nicht Teil eines vordefinierten Software-Pakets. Vom interaktiven Login unter diesen Funktions-Usern sehen erfahrene Administratoren möglichst ab: Sie erreichen das Gleiche durch Werkzeuge wie Sudo oder Gruppenmitgliedschaften. Gerade Letztere werden oft zu Gunsten von Eigentümerschaften unterschätzt, wie sich gleich zeigen wird. Denn bei der Vielzahl von kleinen Einstellungen gerät leicht aus dem Blick, welche Anforderungen die Rechte-Einstellungen eigentlich umsetzen sollen.

Du darfst, du nicht

Als erstes benötigt die Software zur Laufzeit meist Schreibrechte, etwa um Benutzereingaben zu speichern oder spätere Ausgaben zu cachen. Die Anwendung nutzt daher zu ihrer Laufzeit einen speziellen *Laufzeit-User* – einmal außen vor gelassen, dass noch feiner granulierte Szenarien während der Laufzeit die UID ändern.

Dieser Laufzeit-User wird bei verwundbarer Software zur Beute eines Einbrechers. Alles, was für den Laufzeit-User lesbar ist – und das sind fast alle Daten der angegriffenen Anwendung – kann dann auch der Angreifer einsehen. Noch schlimmer wäre, wenn er Daten und Code auch verändern könnte. Die zweite Anforderung heißt daher, einen *Software-User* einzusetzen, der in der Regel keine Prozesse startet, aber Eigentümer der Software ist. Seine Dateien kann der der Laufzeit-User nicht verändern. Das verhindert Manipulationen (Abbildung 1). Also kommen pro Anwendung zwei Funktions-User zum Einsatz: der Laufzeit-User und der Software-User.

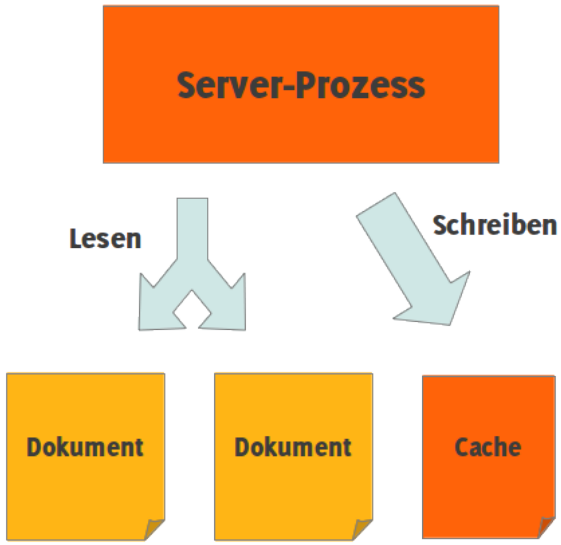


Abbildung 1: Die Aufteilung wirkt wie eine Schutzmauer vor externen Angreifern. Sollte der Server-Prozess einmal kompromittiert sein, kann er nur wenige Daten manipulieren (dunkel), denn die Software und ihre Dokumente (hell) kann er nur lesen.

Als dritte und vierte Anforderung kommen nun dazu, dass Betriebsmitarbeiter in zweierlei Weise auf die Anwendung einwirken: Um beispielsweise Caches zu löschen oder Indices aufzubauen, benötigen sie die Rechte, die denen des Laufzeit-Users entsprechen. Gelegentlich ist weiterhin ein Update oder ein neues Deployment der Software nötig. Dazu reichen dann die Laufzeit-Rechte nicht mehr aus – der damit betraute Admin muss die Dateien der Software selbst verändern. Dazu benötigt er die Rechte des Software-Users.

Gruppen-Typen

Um das zu erreichen, könnten sich Systemverwalter einfach unter dem gerade benötigten Account einloggen, oder sich diese Rechte per Sudo aneignen. Doch wenn sich jeder bei Bedarf als ein Funktions-User anmeldet, geht die Nachvollziehbarkeit verloren.

Dieses Vorgehen ist bei geschickter Planung gar nicht nötig. Damit sich bestimmte Objekte wie Dateien und Verzeichnisse von mehreren Subjekten manipulieren lassen, ist eine konzeptionelle Verschiebung der Rechte vom Eigentümer zur Eigentümergruppe notwendig. Das bedeutet aber, dass

solche Objekte durchgängig umfassende Gruppenrechte besitzen müssen, insbesondere Schreibrechte. Um diese Gewalt auf mehrere Benutzer aufzuteilen, legt der Systemverwalter also je eine *Laufzeit-Gruppe* und eine *Software-Gruppe* an. Sie sind auch die primären Gruppen der zugehörigen Funktions-User.

Nun gilt es, in der Benutzerverwaltung alle Mitarbeiter in die Laufzeit-Gruppe aufzunehmen, die Laufzeit-Aufgaben haben. Und wer verantwortlich für Updates und Deployments ist, erhält durch einen Eintrag in `/etc/group` eine Mitgliedschaft in der Software-Gruppe. Aus praktischen Erwägungen sind die Software-User zusätzlich Mitglieder der Laufzeit-Gruppe. Auf diese Weise können sie im Rahmen von Deployments auch Laufzeit-Daten (re-)initialieren. In kleinen Teams kann eine Person theoretisch auch mehrere Rollen einnehmen. Begrüßenswert ist jedoch, diese Aufgaben voneinander zu trennen.

Um trotz unterschiedlicher Gruppen zusammenzuarbeiten, hat sich meist eine *Team-Gruppe* bewährt, die auch gleichzeitig als primäre Gruppe aller persönlichen Accounts in einer Abteilung oder Teilorganisation herhält. Persönliche Gruppen, die gleichlautend mit dem Benutzeraccount sind, benötigt dann niemand mehr.

Folgendes Beispiel erläutert das Szenario (Abbildung 2).

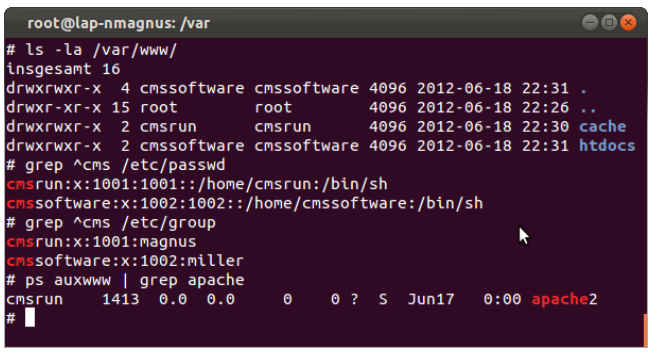


Abbildung 2: Der Systemverwalter teilt die Macht auf zwei User auf. Die Software und die Webdokumente unterhalb von `/var/www` gehören *cmssoftware*, bis auf das Verzeichnis `.../cache`: Dieses Verzeichnis schreibt der Serverprozess, der unter *cmsrun* läuft, zur Laufzeit.

Alles, was man braucht

Insgesamt vier Admins verwalten die kleine Unternehmenswebsite, die als Vhost in einem Apache läuft. Alfred und Anton kümmern sich um den täglichen Betrieb und unterstützen die Redakteure mit Datenimporten. Daher sind sie Mitglieder der Laufzeit-Gruppe *cmsrun*. Ihre Kollegen Bert und Abteilungsleiter Ben halten den Kontakt zu den Entwicklern des CMS und die Software aktuell. Bert und Ben sind daher Mitglied der Software-Gruppe *cmssoftware*. Manchmal vertreten Bert und Ben auch ihre Kollegen, daher sind sie zusätzlich Mitglied der Gruppe *cmsrun*. Alle zusammen haben als primäre Gruppe *cmsteam*.

Eine ähnliche Struktur, aber andere Zuordnungen hat das Team, wenn es sich um die webbasierte Zeiterfassung kümmert. Dort kommt die Laufzeit-Gruppe *chronorun* und die Software-Gruppe *chronosoftware* zum Einsatz. Im Gegensatz zum klassischen Ansatz bestimmt damit nicht mehr in erster Linie der Besitzer, sondern die Gruppe den Zugriff auf Dateien, insbesondere für das Schreiben.

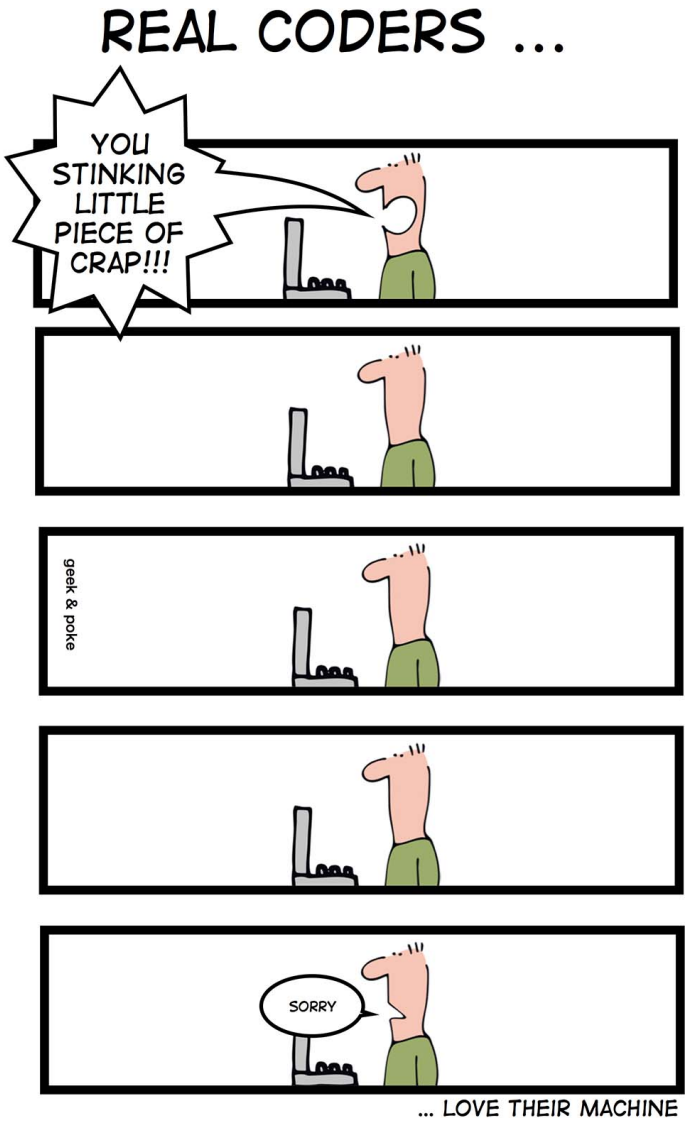
Diese Entscheidung ist essenziell und birgt einige Stolpersteine: Neu angelegte Dateien für das CMS müssen immer der Gruppe *cmssoftware* gehören, also Schreibrechte besitzen, sonst sind die Kollegen schnell außen vor und beklagen sich zu recht. Am besten hat sich dabei der Eintrag `umask 002` in einem Startskript wie der *.bashrc* erwiesen, sowie der Einsatz des Tools *newgrp*: Es öffnet eine neue Shell, deren primäre Gruppe der Anwender es durch ein Argument festlegt. Ein Cronjob, der unterhalb eines Verzeichnisses alle Schreibrechte und Gruppenzugehörigkeiten regelmäßig repariert, kann anfangs ebenso nützlich sein. Noch besser ist jedoch, die Verursacher aufzuspüren und für Abhilfe zu sorgen.

Verantwortung teilen

Mit der Aufteilung in Software- und Laufzeit-User und -Gruppen verbessern Admins einerseits die Sicherheit ihrer Anwendungen und sind andererseits in der Lage, fein zu steuern, wer welche Aufgaben im Tages- und Update-Geschäft übernimmt. Ein wenig Disziplin bei Einrichtung und Betrieb ist dazu allerdings notwendig.

Über Nils

Diplominformatiker, Perl-Bewunderer und Bash-Basher Nils Magnus hat nach acht Jahren Security-Beratung knapp fünf Jahre als Redakteur für das Linux-Magazin gearbeitet. Seit rund einem Jahr hat ihn das System- und Netzwerk-Handwerk wieder, das er als System Architect der inovex GmbH ausübt.



Geschichtsstunde
IT-Frühzeit von 1950 bis 1975

Was einigen als Schnee von gestern erscheint, gilt anderen als Fundament, moderne Informationstechnologie zu verstehen – das der Allgemeinbildung zu entgleiten droht.

von **Jürgen Plate**

Two of the most famous products of Berkeley are LSD and UNIX.

I do not think that this is a coincidence.

Anonymous

Etwa Mitte Mai bekam ich eine E-Mail von Anika Kehrer: „Ich würde in der UpTimes gern eine Rubrik Geschichtsstunde einführen. Mich persönlich fasziniert IT-Geschichte, und ich kann mir vorstellen, dass viele GUUG-Mitglieder dieses Interesse teilen." Dabei dachte sie nicht an eine dicke Abhandlung, sondern an eine regelmäßige Kolumne, die durchaus auch anekdotisch sein darf. Schließlich spielt UNIX in der IT-Geschichte eine zentrale Rolle. Auch in der GUUG sind nicht nur die alten Hasen vereint, die wissen, warum *awk* so heißt oder warum `if` mit `fi` beendet wird, aber `do` mit `done`.

Dass so eine Geschichtstunde sinnvoll ist, wird einem regelmäßig vor Augen geführt, wie die folgenden Beispiele zeigen: So stand von einiger Zeit in dem Magazin *Focus*, dass Konrad Zuse den ersten Digitalrechner der Welt gebaut habe – jedoch war tatsächlich Aikens Mark 1 ein Jahr früher dran. Meine Bauspar-Postille datiert die Geburt des Internet auf 1992 – jedoch frage ich mich, was ich dann die Jahre davor eigentlich für Mail und News benutzt habe? Auch der Kauf der renommierten Firma SCO durch Caldera, welch Letztere sich dann (weil der Ruf von Caldera ruiniert war) flugs in SCO umbenannte, ist schon fast vergessen – hängen blieb nur die Klage wegen des UNIX-Copyrights. Und in einem Blog wurde ein Fernschreiber (der *Teletype ASR 33*) zu einem Plotter – nur weil eine Papierrolle zu sehen war.

So werde ich versuchen, möglichst regelmäßig etwas über die nahe Vergangenheit zu bringen, was Newbies und alte Hasen gleichermaßen interessiert. Dabei werde ich das Thema nicht zu eng fassen und neben UNIX auch über die verwendete Hardware oder über das Netz der Netze schreiben. Thematische Anregungen aus dem Leserkreis sind durchaus willkommen.

Wie alles begann

Vor rund 60 Jahren entstanden die ersten, sehr einfachen Betriebssysteme. Sie erlaubten, Programme nacheinander auszuführen und auf die Peripherie des verwendeten Großrechners zuzugreifen. Dieses Prinzip der Stapelverarbeitung (*batch processing*) war bis in die 1960er Jahre hinein vorherrschend bei der Nutzung von Großrechnern. Es ermöglichte die unbeaufsichtigte und vereinfachte Benutzung des Rechners und erlaubte auch eine Umsortierung der „Rechenaufträge" nach Ressourcenbedarf und Priorität (*spooling*). Ein typischer Arbeitsablauf zur Bearbeitung eines Problems war folgender:

- Aufschreiben des Programms (etwa in Assembler oder FORTRAN) und der Eingabedaten auf Papier
- Eingabe des Programms am Lochkartenstanzer (offline!)
- Abgabe des Lochkartenstapels beim Operator
- Nach etlichen Stunden: Abholen der Ergebnisse als Ausdruck
- Sichten der Ergebnisse (inklusive Beseitigen der Syntaxfehler)

In den 1960er Jahren entstand dann eine Alternative zur Stapelverarbeitung: die interaktive Programmbedienung im Zeitmultiplexverfahren (*time-sharing*), das uns wohlbekannt ist – am PC gab es nie etwas anderes. Der Rechner führt jedes Programm nur für sehr kurze Zeit aus (im Millisekundenbereich), dann sichert er dessen Status, und es gelangt das nächste Programm zur Ausführung. So kann er mehrere Programme scheinbar gleichzeitig ausführen, wobei sich die reale Laufzeit eines jeden Programms verlängert. Anfangs war das Ganze aber keineswegs so komfortabel wie heute. Eines der ersten Timesharing-Systeme wurde am Massachusetts Institute of Technology

(MIT) entwickelt und 1961 vorgestellt: das Compatible Time-Sharing System (CTSS).

Eines der unter CTSS entwickelten Programme war das Textformatierungswerkzeug *runoff*. Das ist der Urahn einer Vielzahl ähnlicher Programme, unter anderem von *troff* und dessen Nachfolger *groff*, das noch heute zur Formatierung der UNIX Manual Pages dient.

Im Jahr 1964 begannen das MIT und die Bell Labs (das ist eine Kooperation von AT&T und General Electric) die Entwicklung eines Nachfolgers für CTSS. 1965 stellten sie eine Reihe Konzepte vor, die in das 1969 präsentierte Betriebssystem MULTICS einflossen (Multiplexed Information and Computing System). AT&T verließ das Projekt relativ früh, weil das Unternehmen unzufrieden mit Fortschritt und Ausrichtung des Systems war. Durch den Verkauf der Computersparte von General Electric gelangte MULTICS dann zu Honeywell und später zu Bull. Das letzte MULTICS lief bis zur Jahrtausendwende. Die kompletten Quelltexte von Multics stehen seit 2007 zur freien Verfügung.

Und dann kam UNIX

Nun endlich kommen die Väter von UNIX ins Spiel. Der an den Bell Labs beschäftigte Informatiker Ken Thompson hatte für MULTICS ein kleines Spiel namens Space Travel geschrieben, bei dem es um Planetenbewegungen und die Reise zu den Planeten ging. Um es weiterhin verwenden zu können, portierte er das Spiel auf GECOS, das Betriebssystem der Großrechner GE-645 von General Electric. Nachteil dabei war, dass eine Stunde Spielen ungefähr 75 Dollar kostete und die Anzeigemöglichkeiten dieser Maschine eher beschränkt waren – im Grunde handelte es sich nur um ein aufgebrezeltes Batch-Betriebssystem.

Eine Gruppe um Ken Thompson, unter anderem mit Dennis Ritchie und Joseph Ossanna, machte sich für die Anschaffung einer DEC-10 stark, auf der sie ein eigenes Betriebssystem implementieren wollten. Die DEC-10 war damals eine hochmoderne Maschine mit einem recht cleveren Timesharing-System. Leider war der Computer aber auch sehr teuer. So gab es trotz mehrfacher Anläufe keine DEC-10 – das Management hatte wohl keine Lust, sich nach den Erfahrungen mit MULTICS schon wieder auf die Entwicklung eines Betriebssystems einzulassen.

Rückblickend betrachtet war das eine glückliche Fügung. Denn Thompson stieß auf die heute noch berühmte „little-used PDP-7 sitting in a corner“. Space Travel auf die PDP zu portieren, war extrem nervig: Der Quellcode musste auf der GE 645 übersetzt, und dann der Binärcode per Lochstreifen in die PDP eingelesen werden. Das ging einigermaßen glatt, weil zur damaligen Zeit Fernschreiber (englisch *Teletypewriter*) als interaktive Terminals verwendet wurden (Abbildung 1).

Für die Jüngeren unter uns: Fernschreiber setzten Tastatureingaben in ein serielles Datensignal um, das ein parallel zum Telefonnetz existierendes Netz zum Empfänger übertrug, der es ausdruckte. Die Fernschreibmaschinen hatten auch angebaute Lochstreifenleser und -stanzer, damit man die Texte vorbereiten und mit maximaler Geschwindigkeit senden konnte. Das deutsche Fernschreibnetz namens Telex betrieb die Deutsche Telekom bis 2007.

Abbildung 1: Der amerikanischer Fernschreiber namens Teletype ASR-33.

Aus dieser Not heraus programmierten Ken Thompson und die anderen die nötigen Tools, die ihnen die Arbeit auf der PDP erleichtern und vor allem Unabhängigkeit von der GE 645 bringen sollten. Dazu gehört zum Beispiel ein Dateisystem für die DEC-Tapes (Magnetbandkassetten mit ähnlichem Aufbau wie Tonbandkassetten – man konnte sie hin und her spulen und blockweise davon lesen und schreiben). Schon hier tauchte die Verwaltung über i-Nodes auf. Außerdem gab es spezielle Dateitypen für Verzeichnisse und Gerätedateien.

Es folgten ein Assembler zum Übersetzen der Quelltexte sowie der erste Kommandointerpreter (*shell*). Zur Bearbeitung diente ein zeilenorientierter Editor (*ed*). Auch hatte das Mini-Betriebssystem schon Features, die wir von UNIX her kennen, beispielsweise den Fork-Mechanismus. 1970 schlug Brian Kernighan in Anlehnung an MULTICS den Namen UNICS (Uni-

plexed Information and Computing System) vor. Irgendwer hat es dann mit 'x' am Ende buchstabiert – und dabei blieb es dann. Daher gilt 1970 als das Geburtsjahr von UNIX.

Die kleine PDP-7 hatte einige Nachteile: Einerseits war sie nur geliehen, was nicht so schlimm gewesen wäre. Doch andererseits reichte ihre Rechenpower gerade so für die ersten Schritte. Das Team schlug deshalb vor, eine PDP-11 anzuschaffen, die mit rund 40.000 US-Dollar nur den Bruchteil einer DEC-10 kostete. Zum Glück zeichnete sich eine kommerzielle Anwendung dafür ab: Die Patentabteilung der Bell Labs suchte ein System zum Erstellen, Bearbeiten und Formatieren von Patentformularen (heute würde man so etwas Textverarbeitung nennen). So wurde tatsächlich eine PDP-11/20 angeschafft (Abbildung 2).

Abbildung 2: Das bekannte Pressefoto von 1971 mit Ken Thompson (sitzend) und Dennis Ritchie vor der PDP-11.

Mit dem auf die PDP-11 portierten und erweiterten Text-Prozessor *roff* (wie schon erwähnt, ist heute noch der Nachfolger *nroff* für die Formatierung der Manual-Pages zuständig) nimmt UNIX Mitte 1971 seinen ersten kommerziellen Dienst auf. Am 3. November 1971 ist schließlich das *UNIX Time-Sharing System, First Edition* fertig. Dieses UNIX besaß noch keinen Speicherschutzmechanismus und als Massenspeicher eine 500-KByte-Festplatte. Drei Benutzer konnten parallel mit der Textverarbeitung arbeiten. Und auch die Entwickler schrieben gleichzeitig Erweiterungen und neue Programme auf derselben Maschine. Die PDP-11 hatte 24 KByte Speicher, wovon das UNIX-System 16 belegte und 8 für Benutzerprogramme verfügbar waren. Verglichen mit heutigen Systemen ist das ein ausgesprochen schlankes Betriebssystem.

Traditionell werden die UNIX-Versionen auch nicht als Versionen veröffentlicht, sondern sie richteten sich nach den Editionen des Handbuchs. Version 5 ist also das Gleiche wie die *Fifth Edition*. Erst bei Linux ist man zu den profanen Versionen übergegangen, wobei die Distributionen von Debian und Ubuntu ihren Versionen auch noch Namen hinzufügen. Debian-Distributionen werden nach Darstellern aus Toy Story benannt, bei Ubuntu richten sich die wohlüberlegten Tiernamen nach den Buchstaben des Alphabets.

Nach B kommt C

Die zweite Edition von UNIX kam 1972 heraus und hatte nun Pipes sowie Ein- und Ausgabeumleitung, wobei beides schon von Anfang an konzipiert war. In dieser Edition legten die Programmierer das Augenmerk auf den Support und arbeiteten an höheren Programmiersprachen – bis zu diesem Zeitpunkt war das komplette UNIX-System in Assembler geschrieben. Ken Thomspn versuchte, den Kernel in der Sprache B zu schreiben, die er und Dennis Ritchie entwickelt hatten (genauer: in NB, einer modifizierten Version von B). B orientierte sich an einer existierenden Algol-60-ähnlichen Sprache namens BCPL (Basic Combined Programming Language), die 1967 in Cambridge entstand. B war wie BCPL eine Sprache mit nur einem einzigen Datentyp, dem Maschinenwort.

Die Migration von Assembler nach B erfolgte natürlich schrittweise. Dabei wurde schnell erkannt, dass Datentypen essenziell für die Programmierung sind. So entstand die Sprache C (als logischer Nachfolger von B). 1973 wurde C dann um *structures* und globale Variablen erweitert, woraufhin Thompson und Ritchie den Kernel in C neu schrieben. Auch die Shell migrierten sie von Assembler nach C. Inzwischen liefen 25 UNIX-Systeme. Zur damaligen Zeit war diese Zahl ganz ordentlich. Wenn man den Rechnerpreis bedenkt, stecken darin – grob über den Daumen gepeilt – Investitionen von einer Million Dollar.

1973 richtete Bell Labs auch die *UNIX Systems Group* ein, die den internen Support für das Betriebssystem leisten sollte. Mitte 1973 erscheint die UNIX-Version 4, fast vollständig in C geschrieben. Damit wurde das System sehr portabel. Inzwischen hatten etliche Universitäten bei den Entwicklern angefragt und Kopien der 4. Edition erhalten. Das lief damals fast so wie heute mit Linux: Man bekam das System ohne Support und Garantie, aber kostenfrei. Die ersten Magnetbänder mit

UNIX gingen an die Columbia University in New York.

1974 publizierten Ken und Dennis Ihre Entwicklung in dem Magazin *Communications of the ACM* unter dem Titel „The UNIX Time-Sharing System“. Später folgten Artikel über die Sprache C und die Weiterentwicklung von UNIX. Mit der fünften Edition wurde UNIX den Hochschulen offiziell für Ausbildungszwecke zur Verfügung gestellt. Der Preis war ebenfalls akademisch, er sollte lediglich die Kosten des Magnetbandversands und die Druckkosten der Handbücher decken.

Die beiden UNIX-Väter waren weiterhin in die Entwicklung von UNIX eingebunden, vermieden aber jegliches Supportversprechen. Ihre Gruppe bekam intern den Namen *Research* beziehungsweise *1127*. Auch ihre Maschine hieß *research*. Man konnte ihnen jedoch Fehlerreports schicken oder versuchen, einen von der Gruppe am Telefon zu erwischen. Meist kamen dann die Bugfixes einen Tag später.

Im Januar 1974 bekommt die Berkeley-Universität in Kalifornien eine UNIX-Kopie, was noch weitreichende Folgen haben wird.

Fortsetzung folgt – in der nächsten UpTimes.

Chronik

- **1969:** Ken Thompson legt im August 1969 die Grundlagen für UNIX auf einem PDP-7-Minicomputer. Das Betriebssystem ist anfangs namenlos.
- **1970:** Das Betriebssystem bekommt den Namen UNICS (Uniplexed Information and Computing Service). Später ändert sich der Name auf ungeklärte Weise in UNIX.
- **1971:** Im Februar des Jahres wird UNIX auf eine PDP-11 portiert. Thompson und Ritchie publizieren das *UNIX Programmers Manual*.
- **1972:** Ritchie und Thompson entwickeln die Programmiersprache C.
- **1973:** Pipes werden hinzugefügt. UNIX wird in C neu geschrieben.
- **1974:** Die Berkeley-Universität in Kalifornien erhält eine UNIX-Kopie. In den *Communications of the ACM* erscheint der Artikel „The UNIX Timesharing System“ von Dennis Ritchie und Ken Thompson.

Das Letzte

Sie suchen einen Wandschmuck für das Großraumbüro der Admins? Das kanadische Unternehmen Floating Point Digital Images (http://www.fpdimages.com/) vertreibt qualitativ hochwertige Ausdrucke des Diagramms der UNIX-Geschichte von Eric Levenez (http://www.levenez.com/). Das Diagramm enthält die zeitliche Abfolge und Verwandtschaft von über 1000 UNIX-Versionen. Das Angebot reicht vom sechs Meter langen Ausdruck auf normalem Papier für 70 kanadische Dollar bis zum 12-Meter-Transparent für 339 kanadische Dollar (rund 220 Euro). Es gibt das Diagramm auch zum freien Herunterladen und Selbstausdrucken als 25-seitige PDF-Datei.

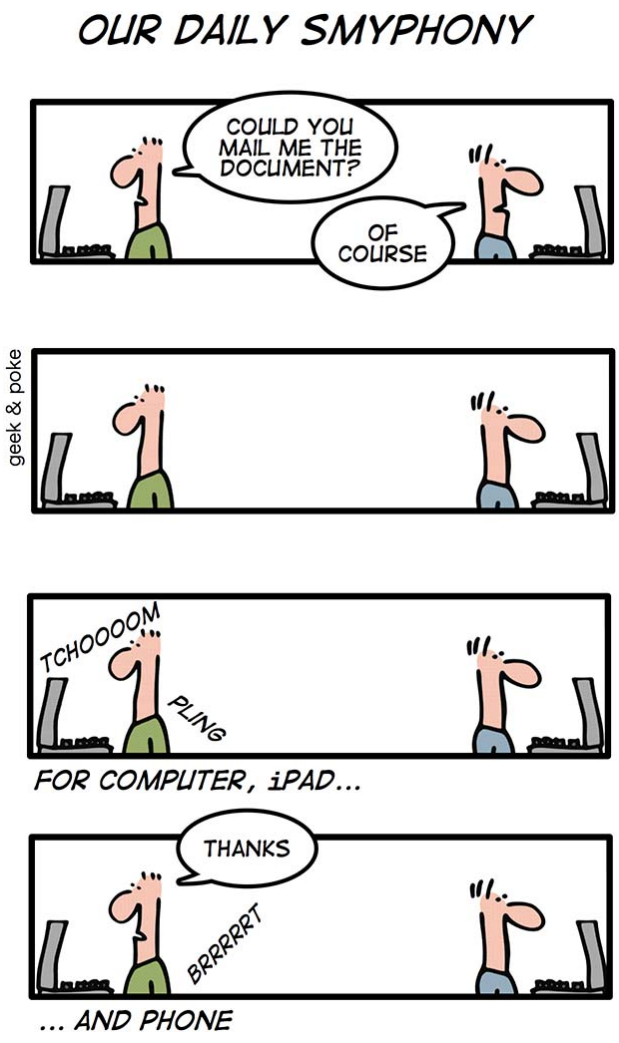
Abbildung 3: Das sechs Meter lange UNIX-Poster. Quelle: darcuncle, Flickr, CC-BY 2.0

Literatur

D. M. Ritchie and K. Thompson: The UNIX Time-Sharing System
http://www.cs.ucr.edu/~harsha/teaching/Winter2011/CS202/readings/UNIX-cacm.pdf
Dennis M. Ritchie: The Development of the C Language
http://heim.ifi.uio.no/~inf2270/programmer/historien-om-C.pdf
Dennis M. Ritchie: The Evolution of the UNIX Time-sharing System
http://www.cs.ucr.edu/~harsha/teaching/Winter2011/CS202/readings/UNIX-evolution.pdf
UNIX Haters Handbook: http://web.mit.edu/~simsong/www/ugh.pdf
The Bastard Operator from Hell: http://bofh.ntk.net/BOFH/index.php

Über Jürgen

Jürgen Plate ist Professor für Elektro- und Informationstechnik an der Hochschule München. Er beschäftigt sich seit 1980 mit Datenfernübertragung und war, bevor der Internetanschluss für Privatpersonen möglich wurde, in der Mailboxszene aktiv. Unter anderem hat er eine der ersten öffentlichen Mailboxen – TEDAS der mc-Redaktion – programmiert und 1984 in Betrieb genommen.



T-Shirts ... und mehr

Am letzten Freitag im Juli findet seit dem Jahr 2000 der Sysadminday statt. Guter Grund, einen Blick auf die GUUG-T-Shirts zu werfen.

von **Hella Breitkopf**

Wenn wir es recht überdenken,
so stecken wir doch alle
nackt in unsern Kleidern.

Heinrich Heine

Sysadminday, Grillabende, Arbeitsplatz, GUUG-Stammtisch, Konferenzen: Man kann ja eigentlich kaum genug Geek-Shirts haben. Damit man als GUUG-Mitglied auch zeigen kann, dass man dazugehört, gibt es unter http://guug.spreadshirt.de/ eigene GUUG-Shirts (Abbildung 1 und 3).

Abbildung 1: T-Shirt, Variante Unixversteher.

Also Farbe bekennen (schwarz, weiß oder mutig bunt – Deine Entscheidung)! Es ist für jeden Geek etwas dabei: weiblich, männlich, und auch für den Geeknachwuchs (die Größen reichen von ganz ganz klein bis 5XL).

Für Sparsame gibt es ein T-Shirt für knapp über zehn Euro zzgl. Versand, bei den anderen Shirts haben wir eine bessere Qualität ausgesucht. Nachweisbar wirksamen Schutz gegen die im Sommer doch manchmal bedenklich auftretenden kosmischen Strahlen gibt es auch (Abbildung 2).

Abbildung 2: Schutz vor kosmischen Strahlen, Variante Blackhat.

Wer sich privat oder beruflich eher nicht im T-Shirt sehen lässt und dem auch ein Poloshirt noch zu informell erscheint, kann auch mit Laptophülle, Button, Aufkleber oder Tasse zeigen, daß er oder sie der GUUG nahesteht.

Abbildung 3: T-Shirt, Variante Root.

Im Shop gibts übrigens keinen Aufschlag auf Material-, Herstellungs- und Versandkosten. Wir reichen die Preise von Spreadshirt so durch. Sinn der Sache ist, möglichst viele GUUG-Logos unter die Leute zu bringen, nicht daß wir Gewinn damit machen.

Über Hella

Sysadmin – *"Alles außer Windows"* – und Pythonfreundin Hella Breitkopf kümmert sich bei der GUUG als Amateur-Grafiker um das schöne*(?)* Äußere von Titelbildern, Anzeigen, Flugblättern und mit den T-Shirts auch von GUUG-Menschen. Werkzeuge der Wahl sind *Scribus*, *Inkscape* und *Gimp*.

Nachlese
Frühjahrsfachgespräch 2012

Wer nicht da war, hat etwas verpasst. Aber was eigentlich? Lassen wir Bilder sprechen.

von **Anika Kehrer**

Oben rechts: Mit Johannes Loxens Keynote fing es gut an. Streitbare Thesen und gewohnte Eloquenz (Foto: H. Breitkopf). Oben links: So sah das Ganze aus dem Zuschauerraum aus (Foto: C. Pahrmann). Links: Nils Magnus hat es gern diskursiv. Sehr gern. Emacs gegen Vi antreten zu lassen, zielt nicht wirklich auf eine Lösung – ist aber unterhaltsam (Foto: D. Wetter, CC-BY-NC-SA). Mitte rechts: Harald und Jens nahmen die Herausforderung an. Ob das irgend etwas am Lieblings-Editor ändert? Ach wo! (Foto: D. Wetter, CC-BY-NC-SA) Rechts unten: Code über Code über Code... „Wenn Ihr Strace beim Arbeiten zuseht, kann Euch schwindelig werden“, warnte Harald König. Stimmt. (Foto: C. Pahrmann)

Oben links: Sven Uebelacker im Gespräch mit Werner Koch (Foto: A. Kehrer). Oben rechts: Wer GnuPG-Schöpfer Werner Koch – hier in seinem Steed-Vortrag – nicht kennt, ist selbst schuld. Kleiner Insider (Foto: D. Wetter, CC-BY-NC-SA). Mitte links: Andri Holmgeirsson hat sich der Regionalgruppe Stuttgart angenommen. Mitte rechts und unten: Felix Pfefferkorn erklärt den Azubi-Raum von 1&1 zum Spielplatz für kreativen Bastelspaß. Noch nicht mal aufräumen muss man – das kommt an (Fotos: A. Kehrer).

Oben: Jens Link in seinem vergnüglichen Vortrag über Anycast und Multicast. Drunter links: Juhuu, das FFG-Plakat ist da! Doch erstmal ist kritisches Beäugen angesagt. Drunter rechts: Biz-Talk mit Corinna Pahrmann und Wolfgang Stief. Unten links: Kommunikationstrainer Thomas Rose klärte erheiternd über den Umgang mit Kritik auf. Unten rechts und ganz unten: Markus Niemitz hielt einen interessantenVortrag über Click-Hijacking (Fotos: A. Kehrer).

Links: Das Panoramabild kann man auch von oben nach unten machen. Klasse, oder? Oben rechts: Pause muss sein. Bernd Neubacher ermöglicht somit, in dieser Galerie auch mal das schöne FFG-Wetter zu zeigen. Auf seinem T-Shirt steht: „Ich bin Bernd.“ Mitte links: In den grübelnden Autorenseelen von Jan Kleinert (Linux-Magazin), Udo Seidel und André von Raison (iX) mag Folgendes vorgegangen sein: „Ist DAS *wirklich* Bernd?“ - „Also, wenn es draufsteht...“ - „... dann ist er’s nicht. So einfach *kann* das nicht sein.“ Mitte rechts: Andreas Bunten und Nils Magnus auf dem Weg zum Social Event in der Münchner Innenstadt. (Fotos: A. Kehrer) Unten links: So sieht es aus, wenn FFG-GUUGler zufrieden schlemmen (Foto: C. Pahrmann). Unten: Und zum Abschluss eine schöne Panorama-Aufnahme des Foyers (Foto: A. Kehrer).

Call for Presentations
Frühjahrsfachgespräch 2013

Das nächste Frühjahrsfachgespräch der GUUG findet vom 26. Februar bis 1. März 2013 an der Fachhochschule Frankfurt am Main statt.

vom **Planungsteam**

Ist doch – rufen sie vermessen –
nichts im Werke, nichts gethan!
Und das Große reift indessen
still heran.

Ernst Freiherr von Feuchtersleben

Das Frühjahrsfachgespräch (FFG) ist eine Veranstaltung von und für IT-Profis aus dem Unix- und Linux-Umfeld. Als Hauskongress der German Unix User Group (GUUG) e.V. ist das FFG gleichzeitig die ideale Gelegenheit zu einem persönlichen Treffen der GUUG-Mitglieder. Der Schwerpunkt des Kongresses liegt auf technischen und praxisorientierten Fragestellungen beim Betrieb von IT-Systemen. An Bedeutung gewinnen aber zunehmend auch rechtliche, ethische, soziale und ökonomische Aspekte der Informationstechnologie.

Die viertägige Konferenz teilt sich in zwei Tutoriums- und zwei Vortragstage. Die Tutorien können ein- oder zweitägig ausgelegt sein. Für die Vorträge sind auf der Konferenz 40-minütige Präsentationen in zwei parallelen Tracks vorgesehen. Hiervon sollten fünf bis zehn Minuten für die Diskussion eingeplant werden. Die Folien werden im Nachgang über unseren Webserver der Öffentlichkeit zur Verfügung gestellt, sofern sie frei von Veröffentlichungsrechten Dritter sind.

Wer als professioneller System- oder Netzwerkadministrator, Entwickler, Datenbankexperte, Systemarchitekt oder IT-Sicherheitsexperte über Erfahrungen berichten, Fragestellungen vortragen und diskutieren möchte, findet hier einen im deutschsprachigen Raum einzigartigen Teilnehmerkreis. Informationen zu den Programmen der bisherigen FFG befinden sich auf der Veranstaltungsseite der GUUG-Homepage http://www.guug.de/veranstaltungen/index.html.

Für Tutorien, Workshops und Vorträge suchen wir Referenten, die

- neue Entwicklungen vorstellen,
- interessante Fragestellungen aufwerfen und Lösungswege zeigen,
- einen Überblick über ein interessantes Themengebiet ermöglichen,
- aus ihrer Praxis oder ihren Projekten berichten.

Dem Fokus der Konferenz entsprechend sollten sich die eingereichten Beiträge in mindestens eine der folgenden Kategorien einordnen lassen:

- **Betriebssysteme und Applikationen:** Architekturen, Rechtekonzepte, neue Entwicklungen, Administration
- **Netzwerke:** Protokolle, Technologien
- **Virtualisierung, HA:** OS-Virtualisierung, Cluster, SAN, Dateidienste, (Hoch-)Verfügbarkeit, etc.
- **Informationsssicherheit:** Aspekte wie organisatorische, betriebliche und insbesondere technische
- **Infrastruktur des Rechenzentrums:** Klima, Energie und Überwachung
- **Betrieb:** Monitoring, Überwachung, Backup / Datensicherung
- **Nicht-Technische Themen:** Arbeitsorganisation, rechtliche Fragestellungen, Lizenzen und Patente

Wer sich nicht sicher ist, ob ein Thema für das FFG geeignet ist, wende sich bitte per E-Mail an <ffg2013@guug.de>. Wir freuen uns über alle Beiträge mit konkretem Praxisbezug. Wir begrüßen auch Erste-Hand-Informationen von Herstellern. Jedoch dient das FFG dem technischen Austausch und der Weiterbildung, es ist keine Sales- oder Marketingplattform. Reine Produktpräsentationen sind nicht erwünscht.

Vorschläge einreichen

Die Einreichung von Abstracts (maximal 2000 Worte) und einiger biographischer Angaben über den Autor erfolgt ausschließlich online über die Website http://www.guug.de/veranstaltungen/ffg2013/submission.html. Annahmeschluss für Abstracts ist der 3. November 2012.

Die Abstracts werden vom Programmkomitee einem Review unterzogen. Es informiert die einreichenden Autoren zeitnah über die Annahme bzw. Nichtannahme ihres Beitrags. Die GUUG ermuntert alle angenommenen Sprecher, ein Paper (8 bis 20 DIN-A4-Seiten) zur Aufnahme in den Workshopband zu verfassen. Damit erhalten Sie zusätzlich die Möglichkeit, eine voll zitierfähige Veröffentlichung vorzuweisen, denn die Konferenz-Proceedings der GUUG haben eine ISBN. Die Proceedings werden später auch über die Webseiten des GUUG e.V. erhältlich sein. Hierfür benötigen wir eine Zusicherung, dass die Paper keinerlei Rechte Dritter verletzen. Für Tutorien werden ausführliche schriftliche Unterlagen benötigt, welche ausschließlich den Teilnehmern der entsprechenden Tutorien zur Verfügung gestellt werden.

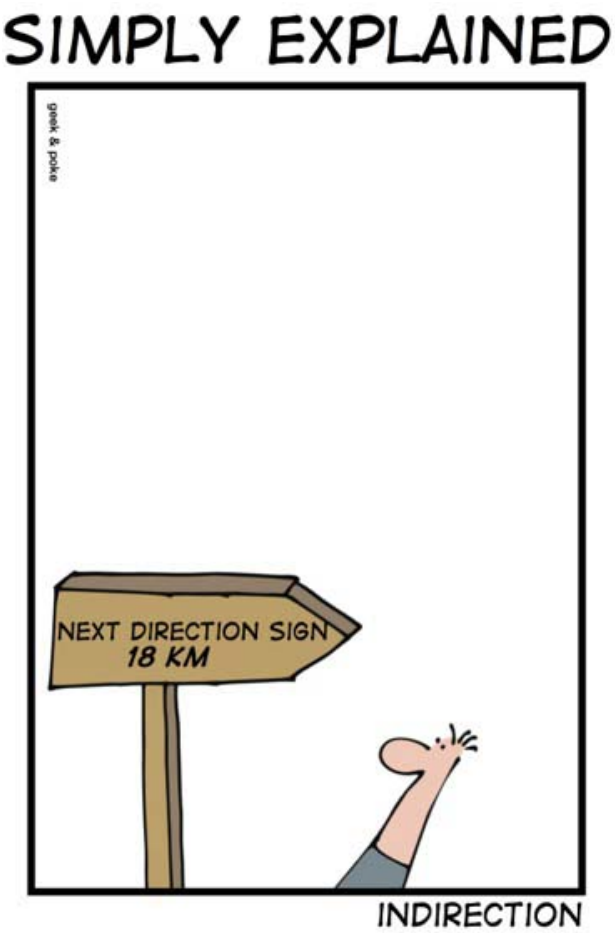
Alle Referenten sind zur Teilnahme an dem Frühjahrsfachgespräch und der Abendveranstaltung von der GUUG herzlich eingeladen. Reisekosten werden nicht von der GUUG übernommen. Sollte dies ein Problem darstellen, bitten wir darum, sich bei uns zu melden. Wir werden dann versuchen, einen Sponsor zu finden.

Programmkomitee

Programmverantwortlicher: Dirk Wetter
(Dr. Wetter IT-Consulting)
Michael Barabas (dpunkt Verlag)
Lars Marowsky-Brée (SUSE)
Hella Breitkopf (GUUG e.V)
Andreas Bunten (Controlware GmbH)
Erwin Hoffmann (GUUG e.V.)
Dr. Christian Paulsen (DFN-Cert GmbH)
Bernd Neubacher (GUUG e.V.)
André von Raison (iX)
Ralf Spenneberg
(OpenSource Training Ralf Spenneberg)
Dr. Christoph Wegener (wecon.it-consulting)
Ingo Wichmann (Linuxhotel)

Informationen zur Veranstaltung und Anmeldeformalitäten werden ab November 2012 auf der Website der GUUG zur Verfügung gestellt.

Wir freuen uns über Firmen, die Interesse an einem Sponsoring haben oder auf der begleitenden Ausstellung auftreten wollen. Interessenten mailen bitte an <ffg2013@guug.de>.



Legenden eines alternden Beagles

von **Snoopy**

```
#include disclaimer.h
```

Die GUUG hat nichts mit dem zu tun,
was ich denke oder schreibe.
Dies ist eine Glosse.
Dramaturgische Überhöhungen sind kein Zufall.

Moby DATEV-Pro, oder Der Wal

Eine Geschichte von Snerman Snelville, AD 2012. Mit Kommentaren der Cyborg-Historiker- und Positronik-Archäologen-Tagung vom 24. Januar 2678. Desweiteren sind Anmerkungen der Archivar-KIs von Memory-Alpha und Google-Prime enthalten. Alle sind klar mit [ ] markiert.

Kapitel Eins – Der Hirte

Nennt mich Ishmael. Vor einigen Jahren – egal wie lange her genau – hatte ich wenig Geld in meinem Beutel und wenig Interessantes, welches mich an das Land gefesselt hielt. So beschloss ich mein Glück zu suchen in der virtuellen Welt der Steuerberater, der elektronischen Abstraktion und anderer harter Arbeit.

[Der Autor ist wohl freischaffender IT-Berater, ein Beruf, der nach dem Cyber-Meltdown von 2103 zunehmend in Verruf geraten war. Entsprechend der damaligen Ersatzreligion der Natur-Verherrlichung und Mittelalter-Huldigung orientiert sich der Erzähler stark an alten Metaphern.]

Obwohl es Frühling war, als sich diese Geschichte zutrug, erfüllte eine graue Novemberstimmung meine Seele, und wenn ich sie erzähle, habe ich einen bitteren Geschmack in meinem Mund.

Der Auftrag des Kaufmanns

Es geschah, dass ein ehrenwerter Kaufmann seine elektronischen DATEV-Unterlagen in eine neue Ausgabe übertragen wollte. Für diese Arbeiten werde ich bezahlt. Das ist es, was den Unterschied macht, das wunderbare Gefühl bezahlt zu werden, während der Kunde doch nicht bezahlt wird, im Gegenteil, er muss zahlen, und so bin ich auch als einfacher Seemann noch gut dabei.

[DATEV ist wohl laut Memory Alpha eine sogenannte Software, welche auf Silizium-basierten Von-Neumann-Architekturen zum Ablauf gebracht wurde.]

Nun begab ich mich im Vorfeld auf eine Konferenz, ein Treffen mehr oder weniger gelehrter und weiser Menschen, welche ich da befragen konnte wegen des vorgetragenen Plans. In der Tat beschied mir ein menschliches Orakel, dass DATEV wie eine gefräßige Seeschlange aus der tiefen, dunklen See vor Sargasso einen massiven Appetit bewiesen hat, und das ausgiebige Füttern dieser Bestie selbst in der Grünen Kathedrale seiner Erschaffer zur schönen Stadt Nürenberge in Germanien zu heftigem Händel und Zank unter den dortigen Gelehrten führte...

[Fach-Konferenzen, wie die erwähnte, waren eine Weiterentwicklung von sogenannten Selbst-Hilfegruppen vgl. Watchers, Weight*, welche seinerzeit einen entscheidenden Beitrag zur Volksgesundheit leisteten. Hier wurde wohl der massive Ressourcen-Verbrauch der neuen Version DATEV-Pro thematisiert.]*

Somit beschloss mein Kunde, für diese neue DATEV-Pro-Ausgabe neue Rechenmaschinen mit mehr Weideplatz zu beschaffen, dabei auch eine große Dell-Weide als Mittelpunkt der Rechnungsarbeiten, um welche sich die anderen Hirten mit ihren Schafen scharen, um dem DATEV-Pro mehr Nahrung zuzuführen.

[Sogenannte Upgrades waren häufiger, chaotischer Teil der damaligen IT-Landschaft und trugen durch ihre Ineffizienz massgeblich dazu bei, dass traditionelle IT nicht mehr beherrschbar war. vgl. Cyber-Meltdown, 2103*]*

Der Beginn der Arbeiten

Schon bei der Einrichtung der wichtigen zentralen Weide ereilten uns böse Nachrichten: Wir hatten die Weide mit 48 Großen Büscheln Heu gewollt, aber der Große Grüne Hirte, landauf wie landab als Mikro der Weiche bekannt, erlaubte uns nur 32 Große Bündel! Als wir versuchten, mit ihm eine Erweiterung der Wiese zu verhandeln, meinte er, er könne sein Fenster aus dem Jahre des Herrn 2008 durchaus vergrößern. Allerdings würde der

zusätzliche Pachtzins der Dell-Weide dann 1.100 Taler mehr kosten, als die ursprünglich von meinem Kunden veranschlagten 600!

Entrüstet wir dann die zusätzlichen Büschel wieder von der Dell-Weide entfernt: Unsere Schafe kommen auch mit weniger Gras zurecht, auch die nimmersatten DATEV-Pro-Tiere.

[Ein damaliger, vorherrschender Software-Gigant namens Microsoft *wird hier wohl verklausuliert erwähnt, ebenso das damalige Software-Produkt Windows 2008R2, welches in der Standard-Lizenzierung nur 32 GByte RAM ansprach. Weiterer Ausbau des RAM nötigte die Kunden zum Erwerb einer sogenannten Enterprise-Lizenz. Heutzutage sind gemäß GMMA* Galaktischer Mass Memory Act *Speichereinheiten unter 1 Zettabyte nicht abrechenbar, da sie als Grundrecht definiert wurden.* Dell *war ein bekannter Hersteller sogenannter Hardware, ein Oberbegriff der verwendeten Rechner.]*

So weit, so gut, und unsere Transkription machte auch gute Fortschritte. Bis zu dem Punkt, wo es darum ging, die bereits geschriebenen Bücher sicher an einem anderen Ort zu verwahren, und wir die Mithilfe eines Acronis-Hirten auf der Dell-Weide wollten. Als wir ihm Zugang gewährten, beschwerte er sich lauthals, dass er erst mit der Erlaubnis seines Vorgängers Lehren austauschen wolle...

[Laut Archiv-KI/Google Prime: Acronis war ein untergeordneter Hersteller sogenannter Archivierungs- und Sicherungs-Software, welche auch das Erstellen von 'boot-fähigen' Sicherungen erlaubte. Die Erstellung solcher Sicherungen bedingte massive Eingriffe in das Boot-Verhalten der sogenannten Server durch Umschreiben von Boot-Sektoren, Registry-Einträgen und anderen Eingriffen, welche laut IT-Infrastruktur-Schutzgesetz aus dem Jahre 2125 verboten sind.]

Ich war zuerst mutlos, weil es garstig schwierig ist, sogar oftmals unmöglich, solche Gelehrten gefügig zu machen und zur Hilfe anzuhalten. Aber das Orakel der Allwissenden Müllhalde wusste Rat.

Der neue und der alte Hirte

Ich sollte den neuen Acronis-Hirten überlisten. Alle Acronis-Hirten haben eine Erlaubnis des großen Gottes der Archive, entsprechende Weiden zu benutzen und ihre Bücher zu kopieren. Daher wäre es möglich, dem neuen Acronis eine Kopie der Erlaubnis des alten Acronis-Hirten zu zeigen. Dieser chronische Segen überdauert viele Monde, viele Schafe und viele Weiden.

Wie konnte ich nur so dumm sein, so gutgläubig, so grenzenlos kurzsichtig? Das Orakel hat versagt, und der alte Acronis-Hirte hat nicht nur seine päpstliche Bulle gezeigt, sondern auch hinterhältig die Landkarten der Dell-Weide geändert!

So war es mir und anderen Schafen unmöglich, sich auf der Dell-Weide zurecht zu finden. Das Orakel hat nun gemeint, ich könnte mit gütiger Hilfe des BCDEDIT-Engels einige Wegweiser wieder korrigieren. Ich habe stundenlang probiert, aber es half nichts. Es war schlimmer als das Rudern auf Galeeren, und als alter Seemann weiß ich hier manche Geschichte zu erzählen. Nie war meine Lage so hoffnungslos wie jetzt!

Ich war also nach viel Haareraufen, Zähneknirschen und Zetern gezwungen, die gesamte Dell-Weide neu zu kultivieren und zu bevölkern. Sämtliche meiner früheren Arbeiten waren umsonst und vernichtet. Mein Kunde war erzürnt und fluchte in Worten, die ich mich hier nicht wiederzugeben getraue.

[Der genaue Ablauf dieser Katastrophe ist nicht klar, aber es erscheint am plausibelsten, dass Lizenz-Konflikte zwischen zwei Acronis-Versionen zu einem unbrauchbaren System führten, welches dann ultimativ neu installiert werden musste, samt sämtlicher Anwendungen, was erhebliche Aufwände nach sich zog. BCDEDIT ist laut Archiv „ein Utility zur Bearbeitung der Boot Configuration Data" und zurzeit Gegenstand dreier Dissertationen an den terrestrischen eBAYreuth- und TUMMLU-Universitäten.]

Nach weiteren drei Stunden, ich war ja in diesen Dingen nun geübt und schlauer, hatte ich den Zustand der Dell-Weide wieder hergerichtet. Ich erhielt die Nachricht, dass der alte Zustand der Weide unter dem Namen WEIDE.ALT beibehalten wurde.

Mit diesem Zustand wusste ich wenig anzufangen. Aber als ich dann die silberne Scheibe der DATEV-Pro-Sermone in die Dell-Weide einlegte, wußte der weise Engel aus der Grünen Kathedrale, wie er weiterarbeiten musste, um die Kopie anzufertigen. Meine Erleichterung war groß und das Gefühl der Beklemmung wich aus meiner Brust.

Weiden-Engel auf Irrwegen

Eifrig rannte er los und sah sich die neue Weide gut an. Aber dann stockte mir der Atem: Der Engel ließ sich nämlich von einem Dämon des Hirten Mikro blenden und täuschen. Er dachte, ALLE Weiden gehörten ihm, und er sah Alles und mehr. Dadurch wurde er sehr verwirrt und von Allmacht berauscht.

Er fand zu viele Schafe, zu viele Wege und Gabelungen und auch die Spuren, welche andere Engel vor ihm gegangen sind. So kürzte er einige wichtige Wege ab, weil er meinte, er sei sie schon gegangen!

Dadurch, dass er auf seiner Reise keine Erfahrungen gesammelt hatte, lernte er auch nicht die richtigen Worte und Dialekte, mit deren Hilfe er mit anderen Hirten und Schafen reden sollte!

Es war, als hätte man uns verflucht und verhext, und der Fluch der Hexe konnte nur dadurch gebrochen werden, dass ich nochmals alle Weiden vom Kot der Tiere und Hirten säuberte und abermals die mühselige Arbeit einer neuerlichen Kopie begann. Mein Kunde weinte, als er von mir Kunde nahm, dass seine Tiere einen weiteren Tag hungern müssten.

[Hierzu gibt es nur Theorien, aber die akzeptierteste ist wohl die, dass die DATEV-Installation die gesamte System-Festplatte scannt und dadurch Treiber etc. in der Registry der Alt-Installation gefunden wurden. Dadurch wurden dann im produktiven Windows diese Treiber nicht mehr eingebunden.]

Zufriedene Schafe

Nach all dieser Schufterei waren drei Tage verstrichen. Ich litt unter Skorbut und wurde von heftigen Fieberkrämpfen geschüttelt. Es dauerte dadurch noch fast eine Woche, bis sich auf der Weide alle Hirten und Tiere auskannten und wohlfühlten.

So gab es noch viel Verwirrung, als die Hirten der anderen Schafe mit dem DATEV-Orakel in der Grünen Kathedrale reden wollten, auf dass ihre Bitten erhört und der rechte Weg ihnen beschrieben werden. Die Schafe der Hirten mussten zuerst mit den Schafen der Dell-Weide im Chor blöken, bis sie erhört wurden.

Sie mussten erst über die Dell-Weide getrieben werden, um mit ihren Freunden zusammenzukommen. Aber dann waren sie so glücklich, wie nur Schafe es sein können, als ihre Unklarheiten verflogen waren und sie sich ihres Weges sicher fühlen durften.

[Vermutlich falsche DNS-Auflösungen der Arbeitsplätze. Laut aufgefundener Dokumentation darf bei DATEV nur der zentrale Server als DNS-Server konfiguriert sein, kein zweiter. Dies ist wohl unrichtig, war aber damals ein sogenannter „Industrie-Standard“, vgl. ITIL ISO9000*.]*

Nach getaner Arbeit war ich selbst immer noch geschwächt vom Skorbut und trieb mich durch die übelsten Spelunken. Mein Rausch wollte nicht enden, und ich torkelte arm wie eine Kirchenmaus durch die dunklen Gassen des Hafens.

Als ich eines Tages aus meiner Ohnmacht erwachte, blickte ich in die freundlichen Augen eines alten Mannes. Er hat mich im Straßengraben gefunden und gesund gepflegt. Ich wusste nicht wie mir geschah, aber ich war voller Dankbarkeit.

Begegnung mit dem Walfänger

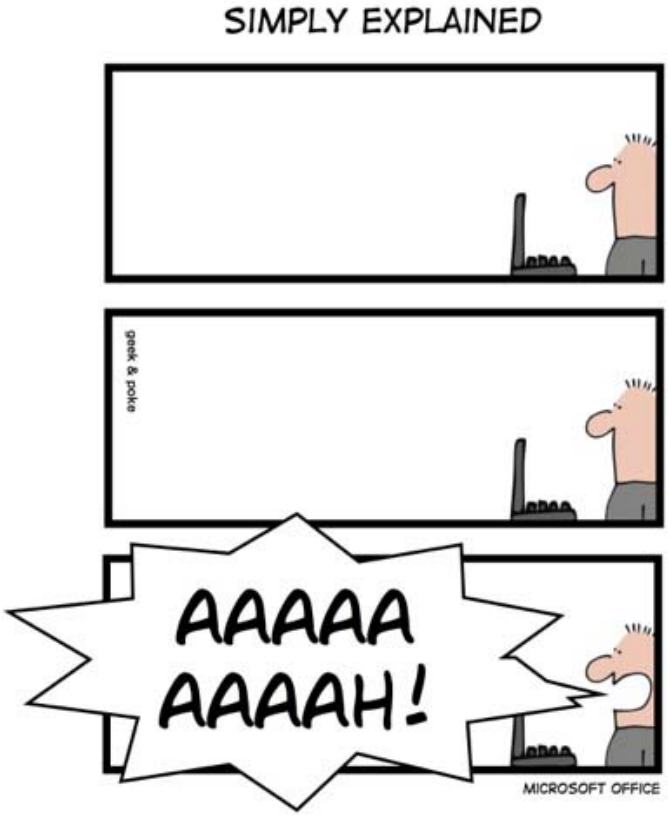
Als ich wieder bei Kräften war, wollte ich ihm danken und fragte, ob ich ihm helfen könne. Er meinte, er sucht eine Handvoll guter Männer für seinen Walfänger, und ich könnte ja gerne bei ihm anheuern.

Ich war begeistert und sagte auf der Stelle zu. Wir umarmten uns, er wandte sich ab und ging zur Tür, sein Holzbein machte auf dem Boden der Hütte merkwürdige schleifende Geräusche.

Und so begann meine eigentliche Geschichte. Fortsetzung folgt – in der nächsten UpTimes.

Über Snoopy

Snoopy hat vor langer Zeit mal in London Computer Science studiert, wo er erste Sysadmin-Erfahrungen mit Unix Version 6, `dcheck` und `icheck` sammelte. Seit seiner Rückkehr nach München hat er neben seinen Administratorentätigkeiten immer wieder versucht, mehr oder weniger lustige Glossen zu schreiben und Vorträge zu halten. Nach einem kleinen Abstecher in die Gastronomie (als Mitinhaber eines Computer-Gaming-Lokals) turnt er zurzeit als Selbstständiger durch die Kundennetze und wundert sich, wie viele Leute mit derart schlecht administrierter Infrastruktur es überhaupt schaffen, etwas zu tun, das vage an Arbeit erinnert.



Autorenrichtlinien
Hilfreiches für alle Beteiligten

Selbst etwas für die UpTimes schreiben? Aber ja. Jedes Thema ist willkommen, das ein GUUG-Mitglied interessiert. Was sonst noch zu beachten ist, steht in diesen Autorenrichtlinien.

Der Schriftsteller ragt zu den Sternen empor,
Mit ausgefranstem T-Shirt.
Er raunt seiner Zeit ihre Wonnen ins Ohr,
Mit ausgefranstem T-Shirt.

Frei nach Frank Wedekind

Wir sind an Beiträgen immer interessiert. Wir – das ist diejenige Gruppe innerhalb der GUUG, die dafür sorgt, dass die UpTimes entsteht. Dieser Prozess steht jedem GUUG-Mitglied offen. Der Ort dafür ist die Mailingliste <redaktion@uptimes.de>.

Themen und Zielgruppe

Die UpTimes richtet sich als Vereinszeitschrift der GUUG an Leser, die sich meistens beruflich mit Computernetzwerken, IT-Sicherheit, Unix-Systemadministration und artverwandten Themen auseinandersetzen. Technologische Diskussionen, Methodenbeschreibungen und Einführungen in neue Themen sind für dieses Zielpublikum interessant, Basiswissen im Stil von *Einführung in die Bourne Shell* hingegen eher nicht. Wer sich nicht sicher ist, ob sein Thema für die UpTimes von Interesse ist, kann uns gern eine E-Mail an <redaktion@uptimes.de> schicken.

Beitragsarten

Neben Fachbeiträgen sind Leserbriefe, Buchrezensionen, Konferenzberichte und Berichte aus dem Vereinsleben für uns immer interessant. Wer sich also nicht gleich traut, mehrseitige Artikel zu schreiben, darf sich gerne erstmal an kleineren Beiträgen versuchen.

Formate

LaTeX: Wir setzen die UpTimes damit. Wer es nicht kennt: Das ist ein Satzsystem, das aus dem technischen Buchdruck kommt. Daher ist es für uns am einfachsten, wenn Autoren uns TeX- oder LaTeX-Dateien zusenden. Weil wir – wie es sich beim Publizieren gehört – mehrspaltig setzen und ein homogenes Erscheinungsbild anstreben, verwenden wir für die UpTimes bestimmte Formatierungen. Eine Vorlage mit den von uns verwendeten Auszeichnungen etwa für Tabellen, Kästen und Abbildungen gibt es per Anfrage unter <redaktion@uptimes.de>. Im Zweifelsfall formatieren wir eingesendete Texte um. Für umfangreichere Artikel ist ein mit der *Article-Class* formatierter Text für uns am einfachsten zu handhaben.

ASCII, HTML: Alternativ können wir recht einfach blanke ASCII-Texte (ISO 8859-1 oder UTF-8) verwenden. *Einfaches* HTML (ohne CSS, Frames und ähnlichem) geht auch.

OpenOffice bzw. LibreOffice, Word, PDF: Alle drei haben eine Formatierung, die wir grundsätzlich nicht übernehmen können, nicht nur wegen des mehrspaltigen Satzes. OpenOffice bietet einen Export nach LaTeX an, der passabel funktioniert. PDF ist kein Format, um Texte weiter zu verarbeiten, sondern nur, um sie dem Empfänger zum Lesen zu geben. Daher bitten wir darum, um PDF eher einen Bogen zu machen, oder selbst in ein verarbeitbares Format zu konvertieren.

Bilder: Wir können alle gängigen Bildformate verwenden, soweit ImageMagick sie verdaut und sie einigermaßen hochauflösend sind. Am besten eignen sich aber PNG- oder PDF-Dateien (für Bilder, nicht für Texte – siehe oben).

Listings: Der mehrspaltige Druck erlaubt maximal ca. 40 Zeichen Breite für Code-Beispiele. Breitere Listings müssen wir als separate Abbildung drucken, was beim Schreiben entsprechend berücksichtigt werden sollte. Ohne entsprechenden Hinweis formatieren wir sonst um.

Manuskripte einreichen

Am einfachsten ist es, wenn wir ein Manuskript per E-Mail an <redaktion@uptimes.de> erhalten. Das ist jederzeit möglich, spätestens jedoch vier Wochen vor dem Erscheinen der nächsten UpTimes. Zum eigentlichen Manuskript ist eine kurze Info zum Autor wichtig, ein Bild ist wünschenswert.

Wir sind sehr dankbar, wenn der Text vor Einsendung durch eine Rechtschreibkorrektur gelaufen ist. `aspell`, `ispell` oder `flyspell` für Textdateien sowie die von OpenOffice bieten sich an. Wir redigieren zwar die Einsendungen, doch üblicherweise bekommt man erst in mehreren Anläufen einen fehlerfreien Text hin. Wir halten uns an die neue deutsche Rechtschreibung.

Redaktion und Satz

Nachdem ein Text eingegangen ist, wird er redigiert. Das bedeutet, dass Syntax und Satzanschlüsse besser lesbar werden, dass Passiva und Substantivierungen verschwinden oder auch Unklarheiten beseitigt werden (die zum Beispiel Fragen offen lassen oder aus Passivkonstruktionen resultieren, ohne dass der Schreibende das merkt). Das kann mit Nachfragen an den Autoren verbunden sein.

Die endgültige Textversion geht jedem Autoren am Ende zur Kontrolle zu. Das ist weniger dazu gedacht, alles noch einmal umzuwerfen oder Verschreiber herauszusuchen. Sondern es geht um die inhaltliche Kontrolle, ob sich durch den Redaktionsprozess Fehler eingeschlichen haben. Zwei bis vier Wochen vor dem geplanten Erscheinungstermin setzt die Redaktion die Artikel. Nach Redaktion und Satz wird die UpTimes dann zum geplanten Termin veröffentlicht.

Rechtliches

Die Inhalte der UpTimes sollen ab Veröffentlichung unter der CC-BY-SA-Lizenz stehen, damit jeder Leser die Artikel und Bilder bei Nennung der Quelle weiterverbreiten und auch weiterverarbeiten darf. Bei allen eingereichten Manuskripten gehen wir davon aus, dass der Autor sie selbst geschrieben hat und der UpTimes ein nichtexklusives, aber zeitlich und räumlich unbegrenztes Nutzungs- und Bearbeitungsrecht unter der CC-BY-SA einräumt.

Bei Fotos oder Abbildungen, die nicht selbst erstellt wurden, flutscht das gern mal durch: Es ist wichtig, dass der Autor sich bei dem Urheber die Erlaubnis zu dieser Nutzung einholt, und fragt, wie die Quelle genannt zu werden wünscht. Die Frage nach der CC-BY-SA ist hierbei besonders wichtig.

An Exklusivrechten, wie sie bei kommerziellen Fachzeitschriften üblich sind, hat die UpTimes kein Interesse. Es ist den Autoren freigestellt, ihre Artikel noch anderweitig nach Belieben zu veröffentlichen.

Finanzielles

Für Fachbeiträge, die einen eigenen inhaltlichen Anspruch erheben, zahlt die GUUG dem Autor nach Rechnungstellung durch den Autor pro Seite 50 € zuzüglich eventuell anfallender USt. Leserbriefe, Buchrezensionen und Artikel bezahlter Redakteure sind davon ausgenommen. Gleiches gilt für Paper, wenn die UpTimes die Proceedings der Konferenz beinhaltet.

Nächste Ausgabe: UpTimes 03-2012, Winterausgabe

- Redaktionsschluss: Sonntag, 4. November 2012.
- Erscheinung: Dezember 2012. :-)
- Gesuchte Inhalte: Fachbeiträge über Unix und verwandte Themen; Veranstaltungsberichte; eigene Buchvorstellungen, Rezensionen; Beiträge zum Vereinsleben.
- Manuskripte an: <kehrer@guug.de> oder direkt an die Mailingliste <redaktion@uptimes.de>

Über die GUUG

Vereinigung deutscher UNIX-Benutzer

German Unix User Group e.V.

Die Vereinigung Deutscher Unix-Benutzer hat gegenwärtig etwa 665 Mitglieder, davon 86 Firmen und Institutionen.

Im Mittelpunkt der Aktivitäten der GUUG stehen Konferenzen. Zwei große viertägige Events der GUUG haben eine besondere Tradition und fachliche Bedeutung: In der ersten Jahreshälfte treffen sich diejenigen, die ihren beruflichen Schwerpunkt im Bereich der IT-Sicherheit, der System- oder Netzwerkadministration haben, beim *GUUG-Frühjahrsfachgespräch*.

Dagegen hat sich im Herbst der internationale *Linux-Kongress* seit 1994 als Treffen der Kernel-Entwickler etabliert. Alle bedeutenden internationalen Entwickler haben dort schon über ihre aktuellen Projekte berichtet.

Seit 2004 findet jährlich der *Free Software/Open Source Telephony-Summit* statt, der neben einer Konferenz und Tutorien für Anwender insbesondere den Entwicklern die Möglichkeit zum direkten Gedankenaustausch gibt. Die ebenso internationale *OpenSolaris Developer Conference* fand 2007 zum ersten Mal in Berlin statt.

Ebenfalls seit 2004 hilft die GUUG bei der Durchführung des *LinuxTag*, Europas größter Veranstaltung zum Thema Freie Software.

2007 haben erstmalig zwei neue internationale Konferenzen der GUUG stattgefunden: Die *LDAP-con* und die *ECAI6* (European Conference on Applied IPv6).

Seit Oktober 2002 erscheint mit der *UpTimes* – die Sie gerade in den Händen halten – auch wieder eine Vereinszeitung. Daneben erhalten GUUG-Mitglieder zur Zeit die Zeitschrift *LANline* aus dem Konradin-Verlag kostenlos im Rahmen ihrer Mitgliedschaft.

Schließlich gibt es noch eine Reihe regionaler Treffen (http://www.guug.de/lokal): im Rhein-Ruhr- und im Rhein-Main-Gebiet sowie in Berlin, Hamburg, Karlsruhe und München.

Warum GUUG-Mitglied werden?

Die GUUG setzt sich für eine lebendige und professionelle Weiterentwicklung im Open Source-Bereich und für alle Belange der System-, Netzwerkadministration und IT-Sicherheit ein. Wir freuen uns besonders über diejenigen, die bereit sind, sich aktiv in der GUUG zu engagieren. Da die Mitgliedschaft mit jährlichen Kosten

Fördermitglied	350 €
persönliches Mitglied	90 €
in der Ausbildung	30 €

verbunden ist, stellt sich die Frage, welche Vorteile damit verbunden sind?

Neben der Unterstützung der erwähnten Ziele der GUUG profitieren Mitglieder auch finanziell davon, insbesondere durch die ermäßigten Gebühren bei den Konferenzen der GUUG und denen anderer europäischer UUGs. Mitglieder bekommen außerdem *c't* und *iX* zum reduzierten Abopreis.

Wie GUUG-Mitglied werden?

Füllen Sie einfach das umseitige Anmeldeformular aus und schicken Sie es per Fax oder Post an die unten auf dem Formular angegebene Adresse. Falls Sie die Seite nicht herausreißen wollen: Sie können den Mitgliedsantrag als PDF herunterladen, siehe URL auf dem Mitgliedsantrag.

Impressum

UpTimes – Mitgliederzeitschrift der
German Unix User Group (GUUG) e.V.
Herausgeber: GUUG e.V.
Bruno-Walter-Ring 30
D-81927 München
Tel.: +49-(89)-380 125 95 0
Fax: +49-(89)-380 125 95 9
E-Mail: <redaktion@uptimes.de>
Internet: http://www.guug.de/uptimes/

Autoren dieser Ausgabe: Snoopy, Jürgen Plate, Andreas Bunten, Torsten Voss, Sven Uebelacker, Nils Magnus, Hella Breitkopf, Wolfgang Stief, Anika Kehrer
V.i.S.d.P: Wolfgang Stief, Vorstandvorsitzender, Anschrift siehe Herausgeber
Chefredaktion: Anika Kehrer
LaTeX-Layout: Robin Schröder
Titelbild und -gestaltung: Hella Breitkopf
Verlag: Lehmanns Media GmbH, Hardenbergstraße 5, 10623 Berlin
ISSN: Aus vergabetechnischen Gründen in der zweiten Ausgabe

Wenn Sie Interesse an Anzeigen in der UpTimes haben,
wenden Sie sich bitte an <werbung@guug.de>.

German Unix User Group e.V.
Mitgliedsantrag

Name, Vorname: .....

Firma/Organisation: .....

.....

Straße: .....

PLZ, Ort: .....

E-Mail: .....

Ich möchte/wir möchten

☐ persönliches Mitglied (Jahresbeitrag 90 €)
  ☐ Ich bin in der Ausbildung (bitte Nachweis beifügen) und zahle nur 30 €
☐ Fördermitglied (Jahresbeitrag 350 €)

werden.

☐ Bitte richten Sie mir den Alias ..... @guug.de ein.

☐ Ich bin mit der Weitergabe meiner Adressdaten an Verlage, die GUUG-Mitglieder im Rahmen der Mitgliedschaft kostenlos mit Zeitschriften beliefern, einverstanden.

Ich bin damit einverstanden, dass die o. g. Daten durch die GUUG elektronisch gespeichert werden. Mir ist bekannt, dass die Kündigung der Mitgliedschaft nur mit einer Frist von drei Monaten zum Ende des Kalenderjahres möglich ist.
Die Satzung ist unter http://www.guug.de/verein/satzung/ einsehbar.

....
Ort, Datum Unterschrift

Ich ermächtige die GUUG bis auf Widerruf zum jährlichen Einzug des Mitgliedsbeitrags von meinem Konto:

Name des Kontoinhabers: .....

Kontonummer: .....

BLZ: .....

Bank: .....

....
Ort, Datum Unterschrift

Bitte faxen Sie diesen Antrag an +49-(89)-380 125 95 9 oder schicken ihn an:

GUUG e.V. * Postfach 25 01 23 * D-44739 Bochum

Dieses Formular ist auch unter http://www.guug.de/verein/mitglied/guug-anmeldeformular.pdf zu finden.