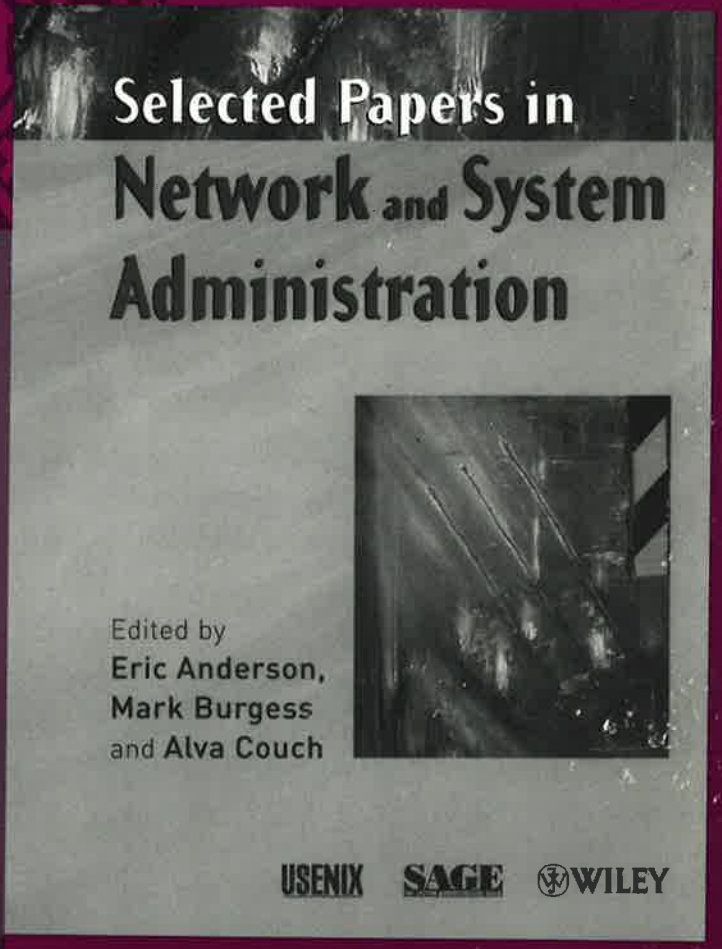
;login:

THE MAGAZINE OF USENIX & SAGE

February 2002 • Volume 27 • Number 1

inside:



USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

USENIX Upcoming Events

The 3rd International SANE Conference

Organized by NLUUG
Co-sponsored by USENIX and the NLnet Foundation.

MAY 27–31, 2002
Maastricht, The Netherlands
http://www.sane.nl

2002 USENIX Annual Technical Conference

JUNE 9–14, 2002
Monterey, California, USA
http://www.usenix.org/events/usenix02/

2nd Java™ Virtual Machine Research and Technology Symposium (JVM '02)

AUGUST 1–2, 2002
San Francisco, California, USA
http://www.usenix.org/events/jvm02
Notification of acceptance: March 12, 2002
Camera-ready final papers due: May 28, 2002
Registration materials available: April, 2002

11th USENIX Security Symposium

AUGUST 5–9, 2002
San Francisco, California, USA
http://www.usenix.org/events/sec02

16th Systems Administration Conference (LISA '02)

Sponsored by USENIX & SAGE

NOVEMBER 3–8, 2002
Philadelphia, Pennsylvania, USA
http://www.usenix.org/events/lisa02
Extended abstracts due: April 29, 2002

5th Smart Card Research and Advanced Application Conference (CARDIS '02)

NOVEMBER 20-22, 2002
San Jose, Califonia, USA
http://www,usenix.org/events/cardis02

2nd Workshop on Industrial Experiences with Systems Software (WIESS '02)

Sponsored by USENIX
Co-sponsored by ACM SIGOPS & IEEE TCOS

DECEMBER 8, 2002
Boston, Massachusetts, USA
http://www.usenix.org/events/wiess02
Submissions due: July 15, 2002

5th Symposium on Operating Systems Design and Implementation (OSDI '02)

Sponsored by USENIX
Co-sponsored by ACM SIGOPS & IEEE TCOS

DECEMBER 9–11, 2002
Boston, Massachusetts, USA
http://www,usenix.org/events/osdi02
Submissions due: May 17, 2002

4th USENIX Symposium on Internet Technologies and Systems (USITS '03)

MARCH 26-28, 2003
SEATTLE, WASHINGTON, USA
http://www,usenix.org/events/usits03

USENIX Annual Technical Conference (USENIX '03)

JUNE 9–14, 2003
San Antonio, Texas, USA

For a complete list of future USENIX events, see **http://www.usenix.org/events**

contents

;login: Vol. 27 #1, February 2002

;login: is the official magazine of the USENIX Association and SAGE.

;login: (ISSN 1044-6397) is published bimonthly, plus November, by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

$50 of each member's annual dues is for an annual subscription to *;login:*. Subscriptions for nonmembers are $60 per year.

Periodicals postage paid at Berkeley, CA, and additional offices.

POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Cover: Our latest book, see p. 67

motd

by Rob Kolstad

Dr. Rob Kolstad has long served as editor of *;login:*. He is also head coach of the USENIX-sponsored USA Computing Olympiad.

kolstad@usenix.org

Catching Up

Maybe it's one of those age-and-stage things, maybe it's coincidence, maybe it's a universal truth, but I'm noticing a lot of people around me in a sort of "catch up" mode. They are trying to execute long-postponed tasks, reduce their important-but-not-urgent task list, and generally shed some of the extra load that keeps them from being able to end time periods (days, weeks, months) with a feeling of completion and happiness instead of a feeling of burden and dread.

I'm doing this myself. Starting with the holidays, I've reduced my willingness to sign up for "just a day or two" of this or that volunteer work or projects. I've reduced my mailbox (which holds the majority of my small task reminders) by a factor of 3 and continue to reduce it each day. I've completed the re-launch of the taxonomy Web page (this would be formally "The Sysadmin Book of Knowledge" pages at *http://ace.delos.com/taxongate*). I've had the luxury of taking the time to automate many of those annoying little tasks that only requiring typing three (or four or five or . . .) commands to make something happen.

I'm just now beginning to feel un-buried enough to be able to see the big picture. I know this is happening mentally when I can embrace the phrase "Today begins a host of new opportunities to produce results" instead of the phrase "Today is yet another backbreaker that will probably put you farther behind than you were yesterday."

Of course, there's always a downside. Since my mind thinks I'm doing such a great job catching up, parts of me want to find a way to perform ever more tasks and re-commit to get right back in the old hole. I'm fighting it right now; we'll see how long that can last.

One of the interesting side results of the luxury of not having to squeeze tasks into impossible small time periods is the joy of being able to perform them with higher quality. The taxonomy pages, for example, needed a quality-lift for the re-launch. I spent three days trying to "get it closer to right." It was very strange because I no longer had the standard copout: "Well, I've put in all the time I have. This will have to be good enough." I know they're not good enough yet, but they continue to improve.

The same phenomenon has manifested itself with the programming contest automation software. I have a confession: I documented it. I know that the Brotherhood will now be out to get me, but I'm just sure it was the Right Thing to do. People can now run programming contests without my hand-holding and a requirement for continuous intervention. I'm not sure yet, but maybe this is a good idea for all sorts of software.

Upon reflection, it seems I am in one of these modes whenever I change jobs or have a "life crisis." Each time I do better in the long term of not overcommitting and being able to live many facets of my life in the allotted 24 hours/day.

I'd love to hear from you if you've caught up and managed to stay caught up. I think it must be a great thing.

apropos

The Mother of Invention

by Tina Darmohray

Tina Darmohray, co-editor of *;login:*, is a computer security and networking consultant. She was a founding member of SAGE.

tmd@usenix.org

For system administrators, using computer technology is a foregone conclusion. Many of us have been pushing that envelope for years. As such, it's easy to take for granted the impact of what we do has on society. However, recent history has shown that computer technology is increasingly turned to for solutions in our complex world.

The recent terror attacks highlight society's use of computer solutions. As the drama unfolded on September 11, the Web was used equally alongside TV and radio as a source of up-to-date information. The anthrax threat resulted in a number of newspapers and magazines requesting electronic communication over traditional paper mail, like this one from the *San Jose Mercury News*:

> Dear Reader,
> With the nation's increased attention on the safety of mail, we at the Mercury News have examined our procedures and made some changes. We need your help.
> We prefer that those who can contact us by e-mail or, secondarily, by fax. You can find those addresses and phone numbers on Page 4A.
> You can find reporters' e-mail addresses and phone numbers at the end of most stories. Thank you for your understanding.

And this one in the November 12, 2001 *Newsweek*:

> Readers who have recently sent submissions through the Postal Service should resend them as e-mail text.

It's a shame, in this case, that "necessity is the mother of invention," but it speaks to the reliance modern society has on the computer infrastructure we create and support.

Just as anthrax-laced mail prompted calls to use computer technology, recall the 2000 US election did the same. In fact, it was just last November that the US sat spellbound waiting for the results of a presidential election which, it appeared, couldn't be definitively declared. In that post-election climate, US citizens called out to statisticians, scientists, and engineers to explore computer technology and electronic voting solutions that would avert a future election snafu.

USENIX President Dan Geer has repeatedly challenged us to use our profession and our affiliation with USENIX and SAGE to challenge ourselves and continue to lead the computing world by excellence and example. Recent history shows that increased necessity, as well as our creativity, may be spurring on our new inventions.

;login:

letters to the editor

EARTHQUAKE ON THE EAST COAST?

from David J. Pfaltzgraff
davepfz@fred.net

Tina:

I enjoyed your first hand report of your experiences with the shutdown of the air system and how you coped with the inconveniences. However, living about 40 miles east of Hagerstown, I always thought it was in Maryland! (Maybe there was an earthquake that caused a major realignment of the state lines that I was not aware of, but I'll just write it off as a result of the exhaustion and confusion arising from your experiences.)

Thanks,

Dave

ANOTHER MIRACLE PRICING MILESTONE

from Jeff Polk
polk@delos.com

512MB PC133 registered ECC dimms are now $74.69 with free shipping. That's less than $150/gigabyte of fast reliable RAM. Can you imagine contemplating a gigabyte of RAM for every desktop just a few years ago?

Every silver lining has a cloud: now operating systems can feel free to use up all that RAM.

ADVANCED ENCRYPTION STANDARD (AES)

from Edgar Danyelian

You may want to inform *;login:* readers that the Advanced Encryption Standard (AES) has been approved and published as a US Federal standard FIPS-197. The standard is available at the following link: *http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf*

Now there are [almost] no reasons to use DES if AES is available . . . .

LETTERS ABOUT GOOD WORKS

In keeping with the USENIX commitment to promoting representation of women and under-represented groups in the computing professions, USENIX contributed $10,000 in support of the **Richard Tapia Celebration of Minorities in Computing Symposium**. Held in October 2001 in Houston, the conference focused on celebrating the technical contributions and career interests of diverse people in computing fields, providing access to diverse researchers in computing, expanding the computing community, and sharing of knowledge between the different disciplines of computing. USENIX funding was used for scholarships for students to attend the event. Several of the recipients forwarded their words of appreciation:

From Eric Brittain . . .

I recently learned you are my scholarship sponsor for the Richard Tapia Symposium later this month. I've decided to write this email to introduce myself and thank you for my sponsorship.

I am currently a PhD computer science graduate student at MIT. I am originally from Atlanta, Georgia and my undergraduate institution was Clark Atlanta University. While I've been at MIT, I've conducted research in the areas of computer graphics, human computer interaction, educational technology and more recently planning techniques. I am working with a research group at MIT that seeks to develop pervasive computing technologies. More specifically, I'm researching ways a person can seamlessly use desktop, handheld, and cell phone as modes of communication without interruption and minimum human interaction.

I sincerely thank you for my sponsorship.

more letters . . .

From Martin Warioba . . .

I am a junior at Louisiana State University majoring in Computer Science with a minor in Mathematics. I would like to take this opportunity to thank you for making possible my attendance at the 2001 Tapia Symposium. Without your sponsorship, all of this wouldn't have been possible and for that I want to say Thank You.

This will be my first computing symposium that I will be able to attend and I am looking forward to be part of it, learn a lot of things out of it and hopefully it will save as one of the rocks in building up my career.

From Rachel Elisabeth Vincent . . .

I am writing to thank you and USENIX for providing me with a scholarship to attend the 2001 Tapia Symposium. I am currently a fifth-year graduate student in the Department of Computational and Applied Mathematics (CAAM) at Rice University. My research interests include numerical linear algebra and computational biology.

the tclsh spot

by Clif Flynt

Clif Flynt is president of Noumena Corp., which offers training and consulting services for Tcl/Tk and Internet applications. He is the author of *Tcl/Tk for Real Programmers* and the *TclTutor* instruction package. He has been programming computers since 1970 and a Tcl advocate since 1994.

clif@cflynt.com

The previous Tclsh Spot article described a simple Tcl script to reformat the output from tcpdump into a more human-readable format. Once you can read the traffic between two systems, many things become easier. In this article I'll discuss one thing that can be done with the processed tcpdump output.

When I wrote the tcpdump reformatting program, I needed to view the traffic between a couple of systems that weren't quite talking. A month or two later I needed to perform regression testing on a set of CGI scripts I was reworking. Being able to read the interactions helped me see what was going on, but didn't quite fulfill my desire for a fully automated regression test.

I wanted to browse the pages using a normal HTML browser, fill in forms, etc., and use the reformatted tcpdump transcript of the session to automatically generate a Tcl script that would duplicate my actions. The next goal was to confirm that the results were what I expected and report the elapsed time, so I could be certain that my modifications actually improved performance. There is probably a package out there that would do what I wanted, but I figured that in the time it would take me to track it down, install, learn and customize it, I could write what I needed from scratch.

Between Tcl's string manipulation tools and HTTP support, this application is pretty simple. It consists of a main module with some utility procedures and a GUI to report progress and results; the tests are simply Tcl scripts loaded with the Tcl source command.

The Tcl http package was described in a few Tclsh Spot articles about a year ago. Briefly, the http package includes functions that enable a Tcl script to interact with a Web server. These functions will send GET, POST, or HEAD requests, as a single execution stream, or with multiple operations in process simultaneously. You can configure the calls to work directly connected to the Net, or through a firewall.

Tcl implements http support in the http:: namespace. Like a Java or C++ class, the Tcl namespace command hides implementation details from the application developer.

This application uses the following two http commands:

http::geturl *url* downloads data from a URL and returns a token to use to access this data.

http::data *token* returns the data associated with a token.

The automatic test building script converts HTTP GET commands that resemble this:

```
GET /cgi_bin/search.tcl?search=books&name=flynt HTTP/1.0
```

to a geturl command like this:

```
set token10 [http::geturl \
    http://$Test(httpServer)/cgi_bin/search.tcl?search=books&name=flynt ]
```

These lines instruct the server to invoke the cgi_bin/search.tcl script, passing search=books&name=flynt as the query string. The CGI script is responsible for unpacking the string into keyword and value pairs.

The HTTP protocol supports two data retrieval operations, GET and POST. The GET command is the simplest and most common command. It can simply request a static HTML document, or can be used to pass additional data to be used by a CGI-type

script by adding a set of keyword/value pairs to the end of the script URL. The data being passed is separated from the main URL with a question mark, and individual keyword value pairs are represented as keyword=value and are separated by ampersands.

There is a limit to the number of characters you can transmit on a single line, and some CGI submissions can exceed this length (for instance, if you are filing a software bug report).

The POST command solves this problem by treating the keyword/value pairs as the body of an HTML message. A dump of the search query done as a POST command would resemble:

```
POST /cgi_bin/search.tcl HTTP/1.0
Content-type: application/x-www-form-urlencoded
Content-length: 23

search=books&name=flynt
```

The first line is the POST command and the next two lines are the HTTP header.

HTTP messages are formatted like email messages. There is a header in which each line consists of a keyword followed by a colon, followed by data, terminated with a new line. The header is terminated with a blank line, and the body of the message follows that.

By default, the http::geturl command will generate a GET command. If your script uses the -query option, a POST is generated. This script would generate the POST command.

```
set token10 [http::geturl \
      http://$Test(httpServer)/cgi_bin/search.tcl \
      -query "search=books&name=flynt"
```

Since the HTTP protocol is stateless, most Web servers use a cookie to link a user to some state information that is being maintained on the server (e.g., items in a shopping cart). The cookie value (and other information) is passed in the HTTP header block.

The http package generates a simple header to declare that this HTTP message was generated by the Tcl http package, etc. If you want to pass other parameters in the HTTP header, you can do this with the -headers option.

The -headers option accepts a list of keyword and value pairs that it will reformat as a MIME-style HTTP header.

For example, this code would add the line

```
Cookie: chocolatechip
```

to the header:

```
http::geturl http://$Test(httpServer)/cgi_bin/search.tcl\
      -query search=books&name=flynt \
      -headers {Cookie chocolatechip}
```

By default, the http::geturl procedure will block until the URL has been retrieved. For many Web robots this is a good technique, but for an interactive test application, you can't freeze the test platform GUI while the Web server is busy thinking.

The -command option will register a Tcl script to be evaluated when the URL has been retrieved. When the callback script is evaluated, a token to identify the data will be appended to the callback script. You can use this token as an argument to a procedure to retrieve the HTML page to process.

This code would request an HTML page and process the page within the checkPage procedure when it becomes available:

```
proc checkPage {identifier token} {
    global correctPages
    set data [http::data $token]
    set dataLength [string length $data]
    set correctLength [string length $correctPages($identifier)]
    if {([string first $correctPages($identifier) $data] == 0) &&
        ($dataLength == $correctLength)} {
        # Report OK
    } else {
        # Report mismatch
    }

}

http::geturl -command "checkPage searchPage" \
    http://$Test(httpServer)/cgi_bin/search.tcl?search=books&name=flynt
```

We can also use the -command option to start several simultaneous searches running on a server with code like:

```
set tclAuthors {ousterhout welch flynt harris mclennan smith nelson}
foreach author $tclAuthors {
    http::geturl -command "checkPage $author" \
        http://$Test(httpServer)/cgi_bin/search.tcl?search=books&name=$author
}
```

As each search completes, Tcl will invoke the checkPage procedure with the author's name as an identifier, and a token to use to access the page retrieved from the Web server.

For this application, I didn't want multiple geturl commands active at once. I wanted just one HTTP interaction active at a time, but I still needed to allow the GUI to update (and a cancel button to interrupt the test). This means I needed to pause the script execution after issuing each http::geturl request and wait until the page was retrieved.

The vwait command causes the interpreter to enter the event loop and process events until the registered variable is assigned a new value.

Syntax: vwait *varName*

varName The variable name to watch. The script following the vwait command will be evaluated after the variable's value is modified.

A script like this will initiate an http::geturl interaction, return control to the Tcl event loop, and wait for the HTML page to be retrieved before going on to the next command.

```
proc checkPage {pageFile token} {
    global doneFlag
```

```
    set newPage [http::data $token]
    # Compare newPage to pageFile

    set doneFlag 1
}

http::geturl $site -command {checkPage fileName}
set doneFlag 0
vwait doneFlag
```

The next problem is validating the page that was just retrieved.

One simple solution to this problem is to create a good set of pages, and compare the new page to a known good page. That's the reason for the pageFile argument to the checkPage procedure. It's the name of a file to compare to the new page.

The Tcl gets command will read a single line of data from a channel. For this application we want to read an entire file, so it's simpler to use the read command which will read the file in a single action.

Syntax: read *channelID ?numBytes?*
The *numBytes* parameter is optional. If it's provided, the Tcl interpreter will read that many bytes (or up to the EOF). If there is no *numBytes*, the read command will read data until it reaches the EOF.

The code to compare pages looks like this:

```
set newPage [http::data $token]

set if [open $pageFile]
set oldPage [read $if]

if {([string first $oldPage $newPage] != 0) ||
    ([string length $oldPage] != [string length $oldPage])} {
    # fail
} else {
# success
}
```

My first thought for comparing the two pages was to use the string match command. However, the string match command will match a glob-style pattern to a string, rather than comparing two exact strings. The string match command will work for most simple tests, but will break when the real data includes metacharacters like *, ? or square braces.

The string first and string last commands compare characters with no wildcards, so these commands can be used to compare strings that may have magic characters in them. If the strings are identical, the first character where they match will be the first character of the string: position 0, since Tcl uses 0-based strings and arrays.

The string length command returns the number of characters in a string. This is used to compare the lengths of the two pages, to be certain that there is no trailing data to worry about.

The last item on my want list was for my test harness to report how long each HTTP interaction took. The Tcl clock command will report or format a time in seconds. The clock command can also report time in the smallest unit that the platform will support

(usually milliseconds), but won't reformat that value directly into a human-readable string.

Syntax: clock *subcommand args*

subcommand The clock command supports several subcommands including:

seconds Returns current time and date in seconds since a system defined epoch.

clicks Returns current time and date as a system dependant integer, usually milliseconds since the last clock rollover.

format Converts a time in seconds to a human-readable format. There are many formatting commands to fine-tune the output.

For this application, seconds were adequate, and the elapsed time can be calculated with a simple expr command like:

```
set startTime [clock seconds]
# Do stuff
set elapsedSeconds [expr [clock seconds] - $startTime]
```

Since the checkPage procedure can compare pages and calculate elapsed time, we might as well make the return value a human-readable string for a final report.

The format command is ideal for generating reports with columns of data.

Syntax: format *formatString value1 ?value2?...*

The formatString resembles a C language printf string, with %d to substitute an integer, or %s to substitute a string at a location, etc.

Like the C language printf command, the format command format string can have a number between the percent symbol and the format identifier to define how many characters wide the field should be, and whether to make the string flush to the left or right margin.

So, a final version of checkPage with page checking, time calculation, and formatted output resembles:

```
proc checkPage {pageFile startTime identifier token} {
    global resultString
    set elapsedSeconds [expr [clock seconds] - $startTime]

    set newPage [http::data $token]

    set if [open $pageFile]
    set oldPage [read $if]

    if {([string first $oldPage $newPage] != 0) ||
        ([string length $oldPage] != [string length $oldPage])} {
        set result "error"
    } else {
        set result "ok"
    }
    set formatString {%-10s %-30s %8s seconds}
    set resultString [format $formatString $result $identifier $elapsedTime]
}
```

When this is invoked with a command like:

```
http::geturl $site -query $query -headers $headerList\
      -command "checkPage page1 [clock seconds] $name"
vwait resultString
```

it will assign resultString a string like this:

```
fail      /booksearch.tcl?author=flynt        23 seconds
```

which can be displayed in a text widget.

A simple GUI for this harness would be a label to show which CGI script is being tested, an exit button, and a simple text widget to display the results.

```
label .l -textvar statusLabelVar
grid .l -row 0 -column 0

button .b  -text exit -command exit
grid .b -row 0 -column 1

text .t -height 23 -width 80 -font {courier 12}
grid .t -row 1 -column 0 -columnspan 2
```

By default, a text widget is created using a proportional font. This makes a nice, easy-to-read display, but is difficult to use for columnar output, since different characters have different widths. The courier font is a fixed-width font that's supported on all platforms.

The test scripts are generated by stepping through the readable tcpdump text looking for GET and POST commands sent from the browser to the Web server. When one of these commands is found, the script will extract the URL, header information, and message body and generate a Tcl script to duplicate the browser action and display the report lines in the text widget.

Each HTTP interaction in the test script resembles this:

```
http::geturl testsite.com:/booksearch.tcl?author=flynt \\
      -command "checkPage  page50 [clock seconds] 50 booksearch.tcl?author=flynt \\"
-headers {Cookie {cookieVal=chocolateChip; mode=Frames}}

set statusLabelVar /booksearch.tcl?author=flynt

set resultString 0
vwait resultString
.t insert end "$resultString\n"
```

The script that converts the HTTP conversation to a Tcl script needs to substitute some values when the test script is generated (like the search parameters), while other substitutions are done when the test script is evaluated. For instance, if the square bracket substitution around the clock seconds command were done during test generation, the test script would measure the time from when the script was generated until the new page was returned, instead of the time from when the HTTP request was submitted until the new page was returned.

Controlling when substitutions will occur when generating new Tcl commands within a Tcl script can get tricky, especially if you want the script you are generating to perform some Tcl substitutions when you are generating the script, and others at runtime.

You can handle this by escaping the Tcl control symbols ($, [and]) with backslashes, but that gets confusing and hard to read very quickly.

For example, to generate this code:

```
set tmpVariable [expr $variable1 + 2]
puts $tmpVariable
set tmpVariable [expr $variable2 + 3]
puts $tmpVariable
```

the commands using backslash escapes and brackets might resemble this:

```
for {set i 1} {$i < 2} {incr i} {
    puts "set tmpVariable \[expr \$variable$i + [expr $i + 1]\]"
    puts {puts $tmpVariable}
}
```

A better solution is to generate the new lines using the Tcl format command.

When the *formatString* is placed in curly braces, the normal Tcl substitutions phase is disabled. This allows us to use otherwise special characters inside the *formatString* and merge in substituted values with %s and %d. Separating the characters you want to output as literals from those you wish to be substituted makes the code easier to maintain.

This script would generate the code above using format commands instead of backslash escapes:

```
for {set i 1} {$i < 2} {incr i} {
    puts [format {set tmpVariable [expr $variable%s + %d]} $i [expr $i+ 1]]
    puts {puts $tmpVariable}
}
```

The *formatString* can be hardcoded, as shown above, or saved in a Tcl variable like this:

```
set id 10
...
set fmt {set %s [http::geturl }
puts [format $fmt token$id]\\
```

There's no advantage to using variables instead of hardcoded strings in the format command, except that using the variable makes the code fit better on these pages.

This procedure will generate a test script from values extracted from the tcpdump output. I only used the format command for output that needed runtime substitutions.

```
set State(id) 0

proc writeCmd {} {
    global State

        set lastSlash [string last / $State(url)]
        set identifier [string range $State(url) $lastSlash end]

        puts "http::geturl $State(site)/$State(url) \\"
        if {[string match $State(type) post]} {
            puts "    -query [string trim $State(body)] \\"
        }
        set fmt {    -command "checkPage  %s [clock seconds] %s %s \\"}
        puts [format $fmt page$State(id) $State(id) $identifier]
```

```
        puts "    -headers {$State(headers)}"
        puts ""

        set lastSlash [string last / $State(url)]
        puts "set statusLabelVar $identifier"
        puts ""

        puts "set resultString 0"
        puts "vwait resultString"
        set fmt {.t insert end "$%s\n"}
        puts [format $fmt resultString]
        puts "\n"

        foreach index {url headers body} {
            catch {unset State($index)}
        }
        incr State(id)
}
```

Extracting the appropriate values from the tcpdump output can be done with a simple state engine. Anyone interested in that part of the code can find it at *http://www.noucorp.com*.

new preprocessor features in C9X

by Glen McCluskey

Glen McCluskey is a consultant with 20 years of experience and has focused on programming languages since 1988. He specializes in Java and C++ performance, testing, and technical documentation areas.

glenm@glenmccl.com

We've been looking at some of the new features added to C9X, the standards update to C. In this column we'll look at how the preprocessor has changed for C9X.

Line-Oriented Comments

C++ and Java have always had //-style comments, where a comment goes from // to the end of the line. C9X now has this comment style as well. Such comments are straightforward to use, and you only need to remember that comments neither nest nor apply within string literals.

For example, in this code:

```
/*//*/ int main()
{
}
```

the // does not introduce a comment, and the code is a valid C program.

And in this example:

```
#include <stdio.h>

int main()
{
      char* s = "//testing";
      printf("%s\n", s);

      int a = 20;
      int b = 5;
      int c = a /**// b;
      printf("%d\n"" c);
}
```

the output is:

```
//testing
4
```

The // within the literal is not a comment, and /**// is equivalent to /.

Alternative Spellings and Digraphs

When you write C programs, especially if you live in the United States, you assume that the full ASCII character set is available to write your code. But this is a tricky area. ISO/IEC 646, the international equivalent of ASCII, allows local variations in some of the characters that C uses. These characters have the appropriate numeric value, but they won't necessarily display as the characters you might expect from looking in a C handbook like Kernighan & Ritchie.

Earlier versions of C defined trigraphs, sequences of three characters used to represent a character: for example, ??< to replace {. So if the ASCII value for { has some local variant meaning, you can use ??< instead.

But trigraphs are not the most readable, and in C9X, another attempt is made to tackle this readability problem by using digraphs, two-character sequences instead — for example, <% to replace {. To judge for yourself whether digraphs are successful, here's a short program written the usual way:

```
#include <stdio.h>
#define N 5

int vec[N];

int main()
{
    for (int i = 0; i < N; i++)
        vec[i] = i + 1;
    for (int i = 0; i < N - 1; i++) {
        for (int j = i + 1; j < N; j++) {
            if (vec[i] < vec[j]) {
                int t = vec[i];
                vec[i] = vec[j];
                vec[j] = t;
            }
        }
    }

    for (int i = 0; i < N; i++)
        printf("%d\n", vec[i]);
}
```

And here it is using digraphs:

```
%:include <stdio.h>
%:define N 5

int vec<:N:>;

int main()
<%
    for (int i = 0; i < N; i++)
        vec<:i:> = i + 1;
    for (int i = 0; i < N - 1; i++) <%
        for (int j = i + 1; j < N; j++) <%
            if (vec<:i:> < vec<:j:>) <%
                int t = vec<:i:>;
                vec<:i:> = vec<:j:>;
                vec<:j:> = t;
            %>
        %>
    %>

    for (int i = 0; i < N; i++)
        printf("%d\n", vec<:i:>);
%>
```

The second program is still readable, though its syntax is more cluttered. It's more portable in the display sense, that is, there is less likelihood that some of the characters will display oddly in a particular local environment.

Note that trigraphs are expanded within string literals, but digraphs are not.

Another C9X feature that aids readability and handling of local character-set variations is the <iso646.h> header, which defines macros for some common operators. For example, you can use or_eq instead of ||=. Here's a short example, showing two ways of writing the same assignment expression:

```
#include <stdio.h>
#include <iso646.h>

int main()
{
      int a = 37;
      int b = 47;
      int c;

      c = ~(a ^ b);
      printf("%x\n", c);

      c = compl(a xor b);
      printf("%x\n", c);
 }
```

Macro Expansion

C has long had a mechanism to define functions with a variable number of arguments, but no corresponding feature for macros. C9X adds this feature, and here's what it looks like:

```
#include <stdio.h>

#define f(a, b, ...) #__VA_ARGS__

int main()
{
      printf("%s\n", f(37, 47, 57, 67));
}
```

The output of this program is:

```
57, 67
```

__VA_ARGS__ is a special preprocessor identifier, and it is replaced with the tokens of the variable arguments to the macro, in this example, 57 and 67. Prepending # to __VA_ARGS__ turns it into a string.

This feature is quite useful, for example, in writing debugging macros:

```
#include <stdio.h>

#define debug(...) fprintf(stderr, __VA_ARGS__)

void f(int x)
{
      debug("x = %d\n", x);
 }

int main()
{
      f(37);
}
```

debug() uses a variable argument list and passes it to the fprintf() function, which also takes a variable number of arguments.

Another new feature is the ability to omit an argument. For example, if you run this program:

```
#include <stdio.h>

#define f(a, b) a##b

int main()
{
      printf("%d\n", f(37,47));
      printf("%d\n", f(,47));
}
```

the result is:

```
3747

47
```

Preprocessor Arithmetic

intmax_t and uintmax_t are new integer types defined in <stdint.h>. They are typedefs that specify the maximum-width integer type available on your local system and are at least 64 bits.

C9X requires that preprocessor arithmetic for #if and #else be done using these types. Here's an example of what this requirement means in practice:

```
#include <stdio.h>
#include <stdint.h>

#if UINTMAX_MAX > 0xffffffffUL

char* s = "UINTMAX_MAX greater than 32 bits";

#else

char* s = "UINTMAX_MAX not greater than 32 bits";

#endif

int main()
{
      printf("%s\n", s);
}
```

If preprocessor arithmetic is not performed using maximum-width types, then the #if comparison in this example cannot be made accurately.

The features we've described above are all useful in writing more expressive and portable programs and in debugging applications that you've written.

musings

by Rik Farrow

Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security* and *System Administrator's Guide to System V*.

rik@spirit.com

Not so very long ago, I took two weeks off. I traveled down the Colorado River, through the Grand Canyon, by raft. No email, no phones, no pagers, and the only news we heard were football scores.

One afternoon, one of the guides asked me what I thought was the most pivotal event in the history of computing. I thought about this briefly, then answered, "Solitaire. If Microsoft hadn't added Solitaire to Windows 3.1, Windows would never have caught on." People learned how to use the mouse and a bit about the "desktop metaphor" by playing Solitaire. Without this game, a crummy windowing interface over an unimpressive operating system would never have survived.

Rather than wander off into yet another Windows-bashing column (how boring!), I'd like to imagine what the world might be like if things had worked out differently. Imagine desktops and servers that could be patched simply by updating a single server. All the patches would include digital signatures, so the clients could check the patches for authenticity before applying them. User accounts would be centrally managed, and a user could log into any desktop system and find his or her home directory and environment waiting to be used.

Of course, people who used the Apollo Domain OS got all this – along with some pretty quirky stuff (see *http://www.citi.umich.edu/apollo-archive/* for a picture of an Apollo workstation). Apollo was not at all open source, and its windowing system had the odd feature that your input always appeared at the bottom of the window where the focus was, even during "full screen editing." Sun killed off Apollo, with its much better security features. Note that you could get X Windows for the Apollo, so you could actually see your input where you thought it should be.

To be honest, I don't really know if the Apollo Domain OS security was as good as it seemed. And the software update/patch system worked because all the hardware came from the same vendor. My only experience working on an Apollo involved getting it to work on a UNIX network with TCP/IP and NFS, something that the Apollo support person was not at all happy about. Why did I want to use NFS when their network file serving was more advanced and much more secure?

In retrospect, Apollo Domain OS looks pretty good. Another thing that still looks interesting is Inferno, a successor to Plan 9 (*http://www.vitanuova.com/inferno/index.html*). I liked several things about Plan 9, especially the notion of having file servers and diskless desktops. Take away those pesky disks from desktop users and do away with viruses, trojans, and other malware. Have some real control over your users (no games at work!), as well as having central file stores that are easy to back up. Plan 9 and Inferno do have improved security, as well as a programming language, Limbo, that, like Java, does not allow buffer overflows and is portable across CPU architectures. The old failing of Inferno was that it was not open source, but that has changed as well.

Inferno uses the Plan 9 concept of a hierarchical namespace to provide security. Basically, if some resource is not part of an application's namespace, the application will not have access to that resource. Public key cryptography is used for authentication as well as for setting up signed and/or encrypted links between networked systems.

Might Inferno work as a desktop OS? Perhaps, but with one very big problem that I'll get to later. It appears to me that one could safely read email without having to worry about the mail tool covertly installing or executing software, or setting up a network

connection to some remote system (Inferno considers the network stack to be a resource).

Linux and BSD have a different control mechanism that might also work. Type enforcement takes traditional ACLs, like those used in Windows 2000, to a finer degree of control. Instead of using an ACL to govern which users can have a particular type of access to an object, type enforcement considers the application used in making the access control decision. For example, user joe may write to /etc/shadow while using the passwd command. Any other attempt to modify /etc/shadow will fail for user joe. A mail tool could be prevented from writing anywhere on the hard drive except for mail folders. And anything stored by the mail tool could also be prevented from having any additional access, so that an executable delivered by email would not be allowed to do any damage.

UNIX and Linux systems do have a capability, called change root, for running a process with limited view of the local file systems. There have been exploits for breaking out of a change root environment, making this technique, while better, still not what I have in mind. FreeBSD (*http://www.freebsd.org*) includes an enhanced change root, called a jail, that provides even more security. Within a change-rooted jail, the superuser cannot use raw sockets, change the IP address given to the jail, or make device files. And the FreeBSD jail call prevents the call from succeeding if any file descriptors pointing to directories outside the new root are open. But instructions for setting up the jail (*http://docs.freebsd.org/44doc/papers/jail/jail-4.html*) call for a complete installation under the change root. Essentially, the jail is a virtual machine with a few features disabled – not what I am looking for either.

There is a really big problem with using Inferno, Plan 9, Linux, or BSD applications. Right now, people are accustomed to using Microsoft Office products as well as Internet Explorer. These applications do not run under the OSes I am interested in. When they run under Windows 2000 or XP, they are apt to misbehave, install trojans, set up back channels, etc. But as long as people insist on using these familiar applications, we won't get anywhere.

Can Microsoft create an operating system where these applications could run securely? Microsoft could graft in type enforcement, for example, but they would still be left with an operating system of ungainly size. They could whittle this down by migrating non-operating system libraries to user space, and then perhaps have a provably secure operating system. At that point, people could use the user-level libraries, properly licensed of course, to run their favorite Microsoft application, under the operating system of their choice (provided it runs on an Intel processor and supplies the same basic set of system calls).

Some people might argue that Windows 2000/XP already has sufficient security. Really? Remember Code Red as an interesting example of a failure to correctly control what IIS, the Web server, could do. A better example appeared recently, when people discovered that if they installed a popular Microsoft game, they could only play it if they were logged in as a user in the administrators group. Ooops.

And will people accept type enforcement as part of their Windows operating system? Unlikely, as it will surely cause slower performance when someone is playing a game.

I'm rambling, and I apologize. But when I hear Richard Clark, the new "cyberterrorism" czar (what cyberterrorism?) say that it is important to our nation's security that

. . . people discovered that if they installed a popular Microsoft game, they could only play it if they were logged in as a user in the administrators group. Ooops.

cable modem and DSL users install personal firewalls, I just want to scream. Why not have them install a simple and secure operating system, one that can play games, run basic business apps, browse the Web, and read email? What good will it do to provide a kludge (the personal firewall) in an attempt to fix something that is badly broken. Why not fix what is broken?

I heard Richard Clark speak at a dinner paid for by Microsoft. I was invited to Microsoft's "Trusted Computing Conference" and attended in the vain hope of participating in a discussion about vulnerability disclosure. Scott Culp had published "It's Time to End Information Anarchy" (*http://www.microsoft.com/technet/treeview/default.asp?url=/technet/columns/security/noarch.asp*) a couple of weeks previously, a treatise that lays the blame for Microsoft's security woes on people who disclose vulnerabilities. Culp did not speak during the conference. Instead, Chris Klaus, of ISS, took the hard-line approach. Essentially, only vendors and select insiders would ever get detailed information about any vulnerability. This should sound familiar, as it was how things worked until about five years ago.

Chris Wysopal (*cwysopal@atstake.com*) took the middle road. Wysopal explained that while working with Steven Christie (*cooley@linus.mitre.org*) on adding new entries to the Common Vulnerabilities and Exposure (CVE) database (*http://www.cve.mitre.org/news/inthenews99.html*), they had both realized that there were no true standards for the handling of vulnerability disclosure. True, there are several example policies for doing this, such as RFPolicy (*http://www.wiretrip.net/rfp/policy.html*), but no actual standards. As vulnerabilities often affect the Internet, they came with the idea of writing an RFC, an Internet standard document, covering vulnerability disclosure.

Actually, what Wysopal suggested was writing two RFCs. One RFC would cover the behavior of both the disclosing organization or individual and the vendor: for example, the timelines for responsibly disclosing to the public information about a vulnerability, for a vendor to respond to information about a bug or problem, or for the vendor to produce a patch. The RFC would also suggest conventions that vendors should follow, such as an email address of "security" as a contact point, as well as a Web page (*http://www.microsoft.com/security* is a good example). The second RFC would discuss standards for the content of a vulnerability disclosure statement and would actually be the more difficult of the two RFCs to write. When I wrote this column, a draft of a single RFC appeared at *http://jis.mit.edu/pipermail/saag/2001q4/000358.html*.

The thing to remember about vulnerability disclosure is that back when there was no public vulnerability disclosure (just the selected insiders, and, of course, all the hackers who exchange this information as well), many vendors pretended not to have security problems. If a vendor did patch a security hole, the patch was rolled into a much larger patch without announcement. And, quite likely, without many people installing the patch either. I don't think we want to go back there.

While I would like to turn back the clock and see something quite unlike Windows take over the desktop market, we have to work with what we have. Someone is sure to remind me of StarOffice, and that there are alternatives to Internet Explorer. Sure there are, but why are my Apache logs full of hits from IE and almost devoid of visits from other Web browsers?

The Microsoft settlement might include sharing some of Microsoft's proprietary information. If Microsoft exposed the Windows libraries, so that other operating systems could compete on an equal basis, this would at least provide a bridge that just might result in people being willing to use other, potentially much more secure, operating systems.

And in that case, I guess they could keep their games on their desktops.

USENIX Needs You

People often ask how they can contribute to the USENIX organization. Here is a list of needs for which USENIX hopes to find volunteers (some contributions reap not only the rewards of fame and the good feeling of having helped but also a slight honorarium). Each issue we hope to have a list of openings and opportunities.

- The ;login: staff seeks good writers (and readers!) who would like to write reviews of books on topics of interest to our membership. Write to peter@matrix.net.
- The *;login:* editors seek interesting individuals for interviews. Please submit your ideas to *login@usenix.org*.
- *;login:* is seeking attendees of non-USENIX conferences who can write lucid conference summaries. Contact Tina Darmohray, <*tmd@usenix.org*> for eligibility and remuneration info. Conferences of interest include (but are not limited to): Interop, Internet World, Comdex, CES, SOSP, Ottawa Linux Symposium, O'Reilly Open Source Conference, Blackhat (multiple venues), SANS, and IEEE networking conferences. Financial assistance to cover expenses may be available. Contact *login@usenix.org*.
- *;login:* always needs conference summarizers for USENIX conferences too! Contact Alain Hénon *ah@usenix.org* if you'd like to help.
- The *;login:* staff seeks columnists for:
  - Large site issues (Giga-LISA),
  - Hardware technology (e.g., the future of rotating storage)
  - General technology (e.g., the new triple-wide plasma screens, quantum computing, printing, portable computing)
  - Paradigms that work for you (PDAs, RCS vs. CVS, using laptops during commutes, how you store voluminous mail, file organization, policies of all sorts)

Contact *login@usenix.org.*

GVSAGE visits the local technology show

by Steven M. Tylock

Steven Tylock has been managing infrastructures for the past 15 years in the Rochester, NY, area. He helped organize GVSAGE as a local SAGE for the region,and has promoted GVSAGE talks at the ITEC technology show.

stylock@gvsage.org

Need a boost for your local SAGE group? Try rounding up some local talent and using a regional technology show to gain support. While it isn't a miracle magnet that will draw all of the local sysadmins out of the woodwork, it will boost the support of existing members, and has a good chance at bringing in some new members.

GVSAGE is a local SAGE group based in the Genesee Valley region of upstate New York (i.e., Rochester and surroundings). We have existed for a little more than three years and are constantly looking to keep our core members interested and involved with the group, as well as trying to attract new members.

For the past three years, we have had a presence at the "Rochester ITEC" (Information Technology Expositions & Conferences) by presenting talks, and this year ITEC was gracious in offering the group booth space. Some of the benefits:

- Visibility to the local sysadmin crowd that comes to the show.
- Opportunity to give back to the local IT culture through educational seminars.
- Chance to improve presentation skills.
- Free publicity for sponsoring organizations.
- Highlight talents of local groups' members.

A Little Visibility Helps Membership

Having fulfilled my obligation to staff the booth and make the pitch to show attendees, I got the strange feeling that nobody knew who we were. If you can catch the badges of attendees to see their occupation, you can properly address them – my eyesight isn't that good (and badges are notoriously small) – so I made some educated guesses and offered: "You don't look like a system administrator, but I bet you know one" (which invariably has to be answered in the affirmative). I was then free to launch into the pitch that GVSAGE is a professional society for system administrators, that, no, we are not selling anything, and if they would pass my information on to the sysadmin back at the office, I would appreciate it.

For bona fide sysadmins, I continued by representing the goals and activities of both the local and national SAGE organizations. I described the technical aspects of the organization as well as the social and networking aspects – my one night out each month with peers is something I enjoy.

This visibility translates into membership in the local SAGE group – GVSAGE counts 60 members on our mailing list and we have grown by 5–10 members each year from the interest generated at the shows.

Don't Forget the Show-Trinkets

It's not what your profession can do for you . . .

Without starting a debate on philanthropy, I'd like you to accept for the moment that it is important to give back to our profession (and society in general). We accept the benefits of belonging without blinking, but sometimes close our eyes when it is our turn to step forward. This giving back takes many forms – distributing software under an open license, helping others, writing/presenting papers, and sending some of our wealth to a cause. (There are *many* ways in which people do this, and no slight is intended by neglecting to mention a specific one here.)

By partnering with the show and providing speakers, GVSAGE offers the knowledge and experiences of the members to show attendees. We have presented talks on security, Web servers, disaster planning, Linux, Perl, email abuse, and careers in system administration. The six presentations in the ITEC 2000 show drew an average of 20 people to the off-the-floor room provided, and the six presentations in the ITEC 2001 show drew an average of 30 people to the on-the-show-floor theater provided.

In addition to the knowledge offered in the presentations, we have tried to offer the slides and speaker contact information for those who would like to follow up on any of the material. We think this is an important service to our local IT community.

Setting Up the "GVSAGE Theater"

Call It the Junior Toastmaster's Club

Lots of organizations create "personal development plans" with their employees at the beginning of each year. Yours might say "present a paper at a conference in the next 12 months" or "learn to speak in public." This aspect of our partnership offers GVSAGE speakers a relaxed atmosphere to dip their toes in the water. (Never once have any attendees thrown objects at our speakers or booed them.) The entrance requirements are not difficult – come up with a topic and a catchy title phrase, commit to following through with a 20- to 25-slide talk that will take 45 minutes, and potentially gain permission to speak, if needed, from an employer.

GVSAGE has not had problems with an over-subscription of potential speakers. In both 2000 and 2001 we offered six talks on the first of two show days. In preparation for 2002, we are looking to provide five talks on each of the two days of the show.

Presentation preparation is up to the individual members, and we offer to read through and comment on slides presentations if prepared far enough in advance. One of the members brings their laptop to the show, and another vendor at the show supplies a screen and projector.

This year we had some new speakers. I'm certain the uneasiness they felt beforehand was much worse than the actual feeling of presenting. Afterward they had the glow of "whew, that's done," as well as the smile of accomplishment. When asked about their participation, all said they would do it again. If you need a boost to polish some speaking skills, consider this as a step on your path.

GVSAGE has been able to have a presence at the local show thanks to a number of organizations. SAGE has sponsored us with membership literature, sample publications, and show gadgets. The names of GVSAGE members' employers appear on presentations, and the one donor required got acknowledgment as well.

GVSAGE was able to staff and outfit a 10' x 10' booth on a minimal budget combined with sysadmin resourcefulness. The show organizers donated the booth itself as well as publicity in the show guide and marketing material. The tables and stools were borrowed. One table draping was donated by a member's company that had changed logos (the old logo was on the reverse side). Another table draping was a pair of "Coleman green" sheets donated by a member (they looked exactly right). SAGE sent us a sign and literature. A member borrowed a company backdrop, and another printed a large-format sign. The single expense was $80 for an 8' x 10' rug. We accepted an $80 donation from one member's consulting company for this expense and displayed a sign thanking them for their support.

The GVSAGE Booth

A couple of the "extras" were the GVSAGE pins done in "shrink-plastic" by a member's wife, and the donation of the funds to make an embroidered logo and sample shirts for supporters to wear and members to order at a later time.

You Never Know What You'll Get in a Box of Chocolates...

One aspect of GVSAGE's participation in the local tech show that didn't seem obvious beforehand is the increased dedication of participants. Members who have given a talk stay members, and if we need to get assistance for something in the future, they either volunteer or don't say no when asked nicely. In a very real sense, by helping the organization, they are drawn to it, and place it higher on their list of "worthwhile groups to be involved with."

You get to know the people who regularly come to monthly meetings, but you don't get to see what motivates them. After a member takes the time to write up a set of slides to talk about for the better part of an hour, you understand them a little better. I can't say that I am "shocked" by any of the abilities that I have seen, but I am heartily encouraged by the care, dedication, knowledge, and experience of those around me in the community.

Give It a Try, and Good Luck!

Whether you are just forming a local SAGE group or want to revitalize a local group that has been coasting for a while, consider taking your XYZ-SAGE to a local technology show – it's a win-all-around proposition. I'd be happy to help make any introductions into the ITEC line of shows (*http://www.goitec.com*), read your anecdotes, and hear any feedback you have. I can be reached at *tylock@gvsage.com*.

if computers had blood, we'd be called doctors

SAs versus MDs, Part II

by Steven M. Tylock

stylock@gvsage.org

Part one of this series drew parallels between the fields of medicine and system administration. Part two looks at the growth of the American Medical Association and considers how SAGE might benefit by observation.

The medical profession struggled to create a training methodology that combined the vestiges of an apprentice/artisan guild background with a scientific/scholarly approach. In addition, the profession needed a mechanism to protect the public from charlatans, snake-oil salesmen, and other forms of "quackery." It evolved through the use of laws and licensing, but also through public demand for quality services.

History of Organization and Membership

Efforts to organize an association of medical professionals remained unsuccessful during the 1800s:

"True, through the end of the nineteenth century, the regular societies were relatively weak, poorly organized, and not particularly representative bodies. As late as 1900, the American Medical Association had only 8,000 members, less than 7 percent of the nation's physicians. Fewer than a third even belonged to a state or local society."[1]

Efforts to organize started, stalled, and started again throughout the century. The AMA ultimately built an association based on county, state, and national representation. This organization attempted to pull in voluntary membership and prevent the establishment of competing organizations.

The organization also attempted to reform the profession through both political processes and the effective training of new doctors. Progress was difficult due to the conflicting nature of the membership.

"The voluntary societies were thus trapped in a dilemma," writes J.L. Kett. "Either they kept their membership requirements loose, in which case they could hardly claim to have purified their ranks, or they tightened requirements and lost any chance of presenting a unified front....If a professional society included only a quarter of the practitioners in a given state, its president could not very well say that the remaining three quarters were outside the profession."[2]

If the society had no rules, it could not claim improvement, yet if the society created rules, it would be unable to sustain the membership. This dilemma simmered as the country grew and scientific progress was made. Some attempts at enacting legislation concerning the practice of medicine were partially successful, but Kett goes on to say:

"A subtle change in popular attitudes which produced a demand for medical advisors versed in scientific terminology brought about a tightening of requirements for entering the profession where innumerable laws and organizations had failed."

Resolution

As American society began to use more standard medical regimens, the need and desire to better regulate that supply took on more weight. Through a growing professional society, improvements in the field of medicine, and public demand followed by

REFERENCES

1. J. H. Cassedy, *Medicine in America: A Short History*, p90(Baltimore: The John Hopkins University Press, 1991).

2. J. L. Kett, *The Formation of the American Medical Profession, p. 177* (New Haven: Yale University Press, 1968).

the enactment of laws, the current system of mandatory licensing and AMA membership came to be.

While opinions may vary on the tradeoff between the current medical profession's unwieldy bureaucracy versus the protections it offers, the alternative of an unlicensed profession must be considered even less appealing.

The medical professional tendency to develop toward specialization (e.g., obstetrics, dentistry, ophthalmology, anesthesiology, neurology, surgery, etc.) has limited the overall power of the "general practitioner" while producing highly evolved forms in each field. The generalist who can see across the specialties to view the interdependencies remains vital, however. System administration is in the early stages of this same specialization process.

Despite their differing stages of evolution, however, both the medical profession and system administration exist today in a state of flux in which there are no easy solutions to the dilemmas confronting each.

Conclusions

SAGE has started on a course that includes some perils that took the medical profession decades to work out. We should be cognizant of what happened in their history to help us along with ours, recognizing that some solutions may be out of our hands for the time being. While we do not generally deal with people's lives, one can guess that if computer operations start threatening people's lives on a daily basis, the bricks of a formal licensing model will start to be laid much faster.

These articles don't attempt to solve today's training, certification, membership, or organization issues through the promise of a radical treatment program, but they do attempt to begin the process of education, opening our eyes a little wider and helping us along on our path.

consulting reflections

Why I Did, How You Can, and a Few Notes from the Field, Part Two

by Strata R. Chalup

President, VirtualNet; Starting as a Unisys 68K admin in 1983, Strata Chalup is now an IT project manager but allegedly has retained human qualities. Her mixed home network (Linux, Solaris, Windows) provides endless opportunties to stay current with hands-on tech.

strata@virtual.net

Last month we examined some basic setup issues around consulting, such as whether or not consulting is a good match for your work style, pros and cons of incorporation, fictitious name statements, and retirement planning. Let's continue on to some of the other big issues around running a consulting practice, namely setting rates and billing and dealing with various kinds of insurance.

After these business essentials, we will discuss how to create and maintain professional visibility, styles of networking, and some time management tips. The latter are definitely in the category of a practice, rather than a skill. They need to be constantly maintained. Even after 18 years of work, most of them consulting, I still find time management a challenge.

A Few Words on Rates and Billing

Charge about $25 - $50 more per hour for 1099 than for W2 work, and always consider your 1099 rate as your "normal" rate. It will cost you the difference in taxes and paperwork, so you are just making sure that you are compensated for the extra tasks you are taking on with 1099 work. Your portion of the social security tax will be between seven and eight percent for your 1099 income, and you will need to do quarterly estimated payments.

With the extra costs of self-employment factored into your normal rate, you can then safely discount your rate for hourly W2 contracts. The excellent article by Steve Simmons [*see references below – ed.*] on setting your bottom line gives you all the info you will need to set an initial rate and make sure you don't discount yourself into the red. Be sure to adjust for the changes in tax rates – at the time Steve's article was written, using a top tax bracket of 28% was reasonable. That top bracket is now 39.6%, a big difference!

Another significant cost of doing business is invoicing and collecting on jobs. I set a discount for longterm work, since it is usually as much (or sometimes more!) work to bill and collect for a 40 hour job as for a 400 hour job. You may think that the cost of invoicing is included in the rate you set as "normal," but in fact if you are doing very short gigs, you'll end up with uncompensated time due to outright theft of services. You're more likely to collect on the 400 hour job, since the folks who committed to that contract are comfortable with spending money. I had endless problems when starting out, accepting one and two week gigs where folks agreed on rates and tasks but then tried to weasel out later on once their stuff was working and they weren't feeling the pain anymore. Think hard about accepting anything less than a 150-hour/4 week committment.

When constructing a Statement of Work (SoW), I always make a point of using 42- or 45- hour weeks, and naming both a number of hours and an expected duration, with the hours multiplied by the rate being the binding figure. I find that 2-3 hours/week over the life of a standard contract tends to be correct for all the little extras that you do for goodwill but which aren't worth amending your Statement of Work for. I state up front in the SoW that we will bill only for work performed, and that while all the

Even though your skills are valuable, don't let your pride keep you from working when there are bills to pay!

hours are allocated, they may or may not be used. Remember that fixed-price contracts are generally a huge headache, and you may not wish to get into them.

The Paint Job is Lovely, But It's Still A Corner

Let's expand on the idea of setting up Statements of Work to supplement the standard contract agreement that you will either provide or accept from your client.

I set up my SoW very carefully, listing the tasks to be performed by VirtualNet and including tasks which need to be performed by the client or client partners in order for work to progress. If there are specific deadlines, I present them attached to tasks and subprojects, rather than to the entire project, and state that they are subject to change if their outside (from the client) dependencies change.

I also state quite explicitly what constitutes deliverables. Are they reports? Are they a service having a certain percent uptime over a certain number of days, barring outside factors like a third party doing a controlled downtime? Scripts? Presentations? An upgraded software package? A signed sheet from my client manager? Some combination of the above?

The most important part of the agreement is a clause saying that all changes to the tasks, deliverables, timetable, or the agreement itself shall be mutually agreed upon and commmitted in writing. If it's not in writing, it's not binding. You may, as I do, consider an email sufficient confirmation, depending on the client and the situation. It's a good idea to track changes on the activity summaries which accompany your invoices, listing them as specific attachments. Since your invoice has a signature line for your client manager (or ought to!), he or she is then signing off on the changes. This covers the case where you have such a good rapport with the client manager that you are willing to accept verbal change agreements – you simply write them up and note them "as per verbal communication of so-and-so, meeting date" and include them. We could spend many, many pages talking about SoW's, activity reporting sheets, invoicing practices, and the like, so we'll stop now, and cover some of those in future articles.

Dealing with the Downturn

In the current economic climate, you may find that the traditional formula for constructing rates will leave you in a position that is much less competitive when bidding for contracts. Even though your skills are valuable, don't let your pride keep you from working when there are bills to pay! Just as the rising tide of the last few years lifted all boats, the ebb tide of today's economy is putting some experienced skippers on the sandbar. Consider that your paycheck is only one component, albeit an important one, of every contract. There are also the professional relationships you make during the contract, the "foot in the door" advantage of having done business with the organization, and the avoidance of significant gaps in your resume or curriculum vitae.

It is up to you to decide what your real bottom line rate is. I don't encourage anyone to work at a rate where they actually lose money after taxes and expenses. This would be an extreme case, however, as there is usually quite a bit of room to maneuver between the hourly rate equivalent to a salaried position and the rate which offsets the higher risk involved in consulting. If you do not have several months of float set aside to cover your normal cash flow outlay, you may decide that it is better to work at a rate closer to what you would make on salary than to be idle, and that is a very responsible decision.

That said, you may wonder how to handle this with a client who knows your regular rates and has indicated that they cannot afford to work with you right now. Or perhaps you wish to preserve the option of raising your rates when the economy starts to move upward again, or when you have more options for work. My advice is to be very candid and open with your potential clients. Express your understanding that times right now are difficult for most organizations, and that you understand that they may not have as much flexibility about hiring contractors as they did in the past. Indicate the you have a choice about whether to work or not right now, and that you would prefer to be working, and are willing to do what it takes to come to a mutual agreement. Come right out and ask, "What could I do to make this work out for you?" A good manager will respect you for being upfront and for your realistic attitude.

My advice is to be very candid and open with your potential clients.

I encourage you to use any involuntary "vacation" time to catch up on professional and personal issues that often go neglected during the press of work. How's your home network doing? When was the last time you updated your Web server, put up a list of white papers, dug out that log rotation script you always meant to polish up and post to net, etc? The shoemaker's kids always go barefoot, and while you're forced to take a little slack time, how about making them some slippers? Another good use of this time is to work on some promotional material for your business. Update your business cards, or create a professional-looking trifold sheet or flyer about your services for conferences, user group meetings, and other networking events. Develop a talk or seminar to present at a future event, so that you can confidently volunteer for an open slot without budgeting time to write the talk.

This is also a great time to schedule all the irksome personal appointments you've been putting off due to your work schedule. Get your teeth cleaned, take the car in for a tune-up, change the filters on your furnace, and so on. Do something unusual and nice for your spouse or SO or a relative or friend. Get your holiday cards addressed, or maybe finally get around to making that mail-merge of your address book print out labels for the cards this year.

Health Insurance

Lack of insurance is a real problem in the United States, and the costs of self-insurance can make or break the economic viability of consulting for many people, especially those who have families or are planning to start a family.

If you are not already consulting, make sure you're eligible for COBRA coverage before leaving your salaried position. COBRA is, technically, the Consolidated OmniBus Reconciliation Act of 1985, but it's commonly used as the name for this special insurance. Substantial changes were introduced with the Health Insurance Portability and Accountability Act of 1996 (HIPAA), including automatic coverage for children born to or adopted by a covered employee.

COBRA is basically an 18-month extension of your current coverage, with you paying the costs that your company formerly subsidized. The eligibility rules can be complex, but the primary feature is that you must already be covered by the medical plan when you leave. Get the details from your HR department or via your internal Web site, or by browsing the insurance rules for your state. Resident aliens in the US are eligible for COBRA coverage in most cases if they are working in the US and receiving a salary. US Citizens working for US companies outside the US are also eligible. One good link for info is the BenefitsLink site's COBRA area: *http://www.benefitslink.com/cobra/summary.shtml*

The reality is that consulting is more of a full-time job than an employee position.

COBRA is not ideal, since you usually lose your prescription plan and various other goodies that depend on you being part of the company's group. The rates may be pretty lousy, since they are heavily subsidized when you are an employee, but it is better than not having insurance, or having to wait many months to have "pre-existing" health issues covered by new insurance. Look into purchasing a supplemental insurance plan before your insurance status changes. The best supplemental plans are just that, supplemental, and you cannot purchase them when you do not have existing insurance coverage. Most of the ones I have seen do not require you to notify them of changes in other coverage, so your supplemental plan could become your fallback plan if you leave an employer before the six-month COBRA eligibility window.

Do not, of course, neglect the option of using one's salaried spouse's insurance! That's what we do, and it's a great deal. Also consider, whether incorporated or not, joining an organization such as IEEE, ICCA (International Computer Consultants Association), the Better Business Bureau, or other similar organizations which often have group insurance deals for their members. The US Small Business Administration may be of help as well. One colleague specifically praised the IEEE health plan for members, and offered a very negative opinion of the ICCA plan. I recommend that you do your own research, as everyone's insurance needs are different. As always, be very careful about pre-existing conditions, and whether coverage is dependent on being with a "network" of providers. Those who travel frequently on extended assignments may find this to be a critical issue for them. Fortunately there are many resources available to help evaluate insurance plans. Consumer Reports has several features available online that are very helpful, and a quick Google or AltaVista search will turn up many more.

Presence and Visibility

If no one knows you're consulting, your phone isn't going to ring and your inbox isn't going to have business proposals dropping into it – except for the usual offers to help move money out of small countries and other illegal and/or dubious "business" deals. So, let's look at ways in which you can hang your shingle out where folks can see it and, more importantly, get a feel for what you do and why they'd want to call on you for work. You don't need to be, and shouldn't be, the kind of consultant who is always nagging his or her friends to try to find work. On the other hand, the reality is that consulting is more of a full-time job than an employee position. To succeed, you need to be constantly presenting the best aspects of your skills, your abilities, and your self. This is a good time to swear off temper tantrums, barbed conversational styles, constant complaining, and similar personal habits, and to go the extra mile to lend folks a hand in technical forums and at user group meetings.

Online . . .

For most of us in technical consulting, the online presence is key. Your Web site should look very professional. Keep a clear separation between your personal homepage and your business homepage. There isn't even a link from *http://www.virtual.net/* to my personal homepage, and I keep the spiders out of it with a robots.txt file as well. I've gotten several contracts over the years that began with someone remembering only my name, and not any contact info, and doing a search on the name. I want my emails on technical lists, or my business Web site, to show up on the results page, rather than pictures of my cats or long diatribes about social issues! I did get a contract once because someone remembered my name (it's a bit unusual) and did a search, only to find a

long diatribe. This manager, who became an important mentor for me, was heard to say, "This proves she can write! Let's get her in here!". Generally, however, you want your professional side to be more visible than your personal life, especially in places where potential clients may look.

Your most important visibility is actually on technical mailing lists among your own demographic. What type of consultant are you? A project manager, a hands-on sysadmin, a storage area network architect, a Perl programmer? For your own professional development, you are already almost certainly on some mailing lists of people with similar interests and abilities. Do you participate, or just lurk? When you do post, are you respectful and careful in your arguments, or are you a flamer who dashes off a few high-intensity lines?

Not all your online presence is technical, of course. I am on some "social" lists of very hightech folks who are usually in the thick of doing interesting stuff. I make time to keep up, to some degree or other, with these lists and to post when there's something relevant and appropriate I can add to a very technical or architectural discussion (as well as tossing in friendly chit-chat). I get a lot of leads from folks on these lists of the form, "hey, if you've got some time, there's something I'd like you to look at." They're not any better than "regular" leads, i.e., perhaps one in five to seven or so leads ever turns into paying work, but they do trickle in.

The second important function of a list community like that is to sanity check dumb ideas one has, drum up info in a hurry, and (best of all) find folks to whom you can quickly sub out bits of things you are doing that you don't have time to ramp up to speed on. You need to make time to stay mutual with it, and even when you turn someone down, make sure you offer them something they can use – a referral to someone else, a tidbit of wisdom on the topic, etc.

Don't *ever* spam your list communities, social or technical, when you're looking for work! If you know for sure a list is a small or liberal community that's ok with that, you can send a note. Some lists, like former work alumni lists, are specifically used for that kind of networking. The folks on some developer or technology list are not going to be happy about the same kind of usage.

On the other hand, adding a one or two line postscript to a regular message is generally fine. Just say briefly that you've got some additional cycles and that you'd welcome hearing from people who have such-and-such kind of work going spare. Let your intuition and past list traffic be your guide. It may be big news to you that you're available for new gigs, but it's probably low on everybody else's priority scale. Fair or not, as a consultant you're "supposed," in the dictionary definition of "imagined," to be more self-sufficient than the Average Bear. Spamming the world with hire-me messages is a good way to reset folks' opinions of your professional skills downward a notch or two.

Don't ever spam your list communities, social or technical, when you're looking for work!

. . . and Offline

Get really nice business cards. It is worth a few extra dollars. Services like iPrint do a surprisingly nice job on full-color cards, and you can do the layout over the Web with any graphics you already have. Business cards do you no good if you don't carry them, so always tuck a few into your wallet and your computer bag, and even the pockets of nice blazers that you tend to wear at conferences and meetings. If you carry a PDA that allows beaming, create a business card entry for yourself so you can beam it to people. I recommend making a separate entry for friends, so that you aren't beaming potential

business contacts your personal homepage URL and your home phone number. It's the little things, like not fumbling around, that count in some cases.

You will be amazed (or perhaps not) at how quickly you forget who you met where and what you talked about. Always take notes afterward on the back of business cards you receive. It doesn't look very good to a potential client to have you email something like "I have your card, but I can't remember what we were talking about specifically – what were you looking for in a proposal?" If you weren't talking about a specific thing, but just got a card as part of an interesting chat, at least note where you met the person, such as LISA '00 or AMW '01. That can be a mnemonic jog for you to mention when you find that you want to contact the person. This is especially helpful if they are more of a prominent figure than you are, and thus less likely to remember you. By the way, some individuals and some cultures think highly of business cards. These folks will take it very negatively if they see you marking your notes on their card, whether it's on the front or the back! The best time to create those notes is between conversations, in private.

Most individual consultants will not find it economically worthwhile to do physical or online advertising of their business, beyond having a Web page and business cards. One important exception to this is local advertising within your professional, religious, or social community newsletters. Most newsletters will allow you to reproduce your business card for a minimal fee, and are targeted very specifically at people who probably already know you but may not know what you do: your church, your parent-teacher organization, a local charity newsletter, etc. People providing traditional services, such as medical, legal, or accounting services, are most likely to find customers by this type of advertising, but if you live in a high-tech community you may find it generates some valid leads. Joining local professional organizations at the higher corporate/business rate may also allow you to put your business card or logo in their newsletters or on their Web sites, and prove valuable.

The one notable exception to physical advertising is the creation of one-page trifold fliers to put on tables at conferences, user group meetings, and other similar events. These are easy to lay out with your choice of editing tool, and are more appealing to most businesspeople than the same info on one flat printed page. If you are providing end-user consulting services on any basis, from expert witnessing for trial lawyers to home internet debugging for individuals, a physical flyer may be one of your most important ways to reach people. Your potential client base will probably not think to look online to find you, and if they do, they will be confronted with a huge array of choices and no real idea how to choose. By leaving your flyers in places where your client base does business, with the permission of the business owner, you are prescreening yourself as someone who is trusted to some degree by a provider whom your client already trusts.

Depending on your target client market, you might want to leave fliers at your lawyer's office, veterinarian or medical clinic, neighborhood shops, on community college bulletin boards, and so on. If you are printing your own flyers in small batches, and one of the business owners or workers is a client, ask if they are willing to have an attributed sentence or two visible on the front of the flyer. Give them an especially good deal, or comp them some time or service as a way of thanks. One good way to handle this is to ask them while doing the invoicing for your work with them, and put a very visible discount or itemize something as "No charge this time, thank you for your support" on the invoice. If you provide services to businesses which are themselves service

providers, such as accounting firms, billing services, printing shops, graphic design firms, and similar, you may find that leaving flyers at these clients will bring inquiries from other businesses who use their services. Maybe the last billing-systems consultant was a total wash, and one of the HR staff mentions it to the folks at Acme Printing when ordering the latest batch of new-employee packets. Wouldn't it be great if they said, "Hey, we had someone really good in here a few months ago to straighten out our billing system, and they left us a few flyers. Want me to put one in with the packets when we deliver them?" Obviously this approach doesn't work for all types of consultants, but for the ones to whom it's applicable, it works wonderfully.

. . . and Internally

Telling your friends and colleagues about your availability, if done tactfully, is a good way to network. When you consult, you hate to turn down work but sometimes you are too busy to handle a new committment. Using the principle of "never turn folks away empty-handed," the best thing to do is to see if anyone who you trust might have some spare cycles, and perform an introduction if appropriate. I have both offered work contacts and been a recipient of offers by consulting peers. When I am open to new gigs, I will usually send a short individual note to four or five respected peers letting them know that I am available to refer work to, or to subcontract work with their clients. Since a client will often just use whomever they have in hand for additional work, it's important to be open to subcontracting via another consultant. This respects their history with the client and allows them to keep control of that relationship.

Telling your friends and colleagues about your availability, if done tactfully, is a good way to network.

Conversely, when a peer offers to hand a client off to you completely, it might be a good idea to inquire into the details. Perhaps the client is a difficult one, or has a history of friction with its consultants. Sometimes it's just a personality or work-style thing, and not something to worry about substantially – unless you have a similar personality or work-style! Usually anyone with whom you have a working relationship such that they are offering client contacts to you will not offer to hook you up with a client who won't work out. They want to preserve their networking too! It's always a good idea to at least lift the hood and hear the engine run, though. Go through your standard pre-client checklist and do your due diligence, just as you would with any "off the street" lead. The difference is that you can be much more sure that it's worth your time to do so, since it's a referral.

Sometimes you'll be competing with colleagues to bid a job, but you get used to this. A big part of a professional attitude is to not let your working relationship be affected by things like who ended up getting the job. You might also be approached to help someone put together a team bid on a job. Offer to help with the paperwork and footwork, but don't try to grab control of the bid process from your colleague, even if you think you'll do a better job. Word gets around about who is hard to work with and who can be counted on to stay professional.

Follow-through is also crucial. I have learned the hard way not to take on more than I can handle. It is not uncommon for it to take two to four years for someone to give you another chance if you flake on something – nor is it guaranteed that you will ever be given a second chance! Think about that every time you're tempted to lose your temper, make a fuss about something, commit to something, or leave something undone "until next week, when I have more time." Just like folks usually learn how much to drink (and not drink!) in college, you'll learn, generally the hard way, just what kind of a workload you can handle. At least you'll be thinking about it while lying

A detailed statement of work for a standard 250-400 hour contract may take 8-10 hours of solid work to put together the first time.

awake at night, rather than while on your knees on a cold tile floor! You'll also get a firsthand experience with the amount of overhead required to run a consulting business, a topic we'll cover more fully in the next chapter.

It's also worth thinking about what kinds of work you find stressful in and of themselves. I don't get out much, but I hate to think that I *can't* make plans to do so. Thus I find 7x24 ongoing operations work to be inherently more stressful than most other kinds of work, even those where I am on-call 7x24 during the build-out phase of a project. I know that responsibility is going to end, and that there won't be a drop-dead emergency involving thousands of customers and millions of dollars, ticking away every second that I haven't responded to my cellphone. Yeek! On the other hand, put me in multiple daylong planning meetings with corporate execs and high-level managers, and I won't turn a hair – I actually find that to be a lot of fun! Many folks I know work 180 degrees out of phase with that, and enjoy knowing that they'll be called on to save the day (or prevent themselves from ever being called, by their skill in managing the systems), but can't stand meetings or dealing with VIPs.

Time Off the Top

Suggestions for maintaining time for a personal life and hobbies are welcome! This is definitely a case of "those who can't do, teach." At least, that's how I feel most of the time, though in the course of a year I read dozens of books, do specialized digital photography (flower closeups), scuba dive, do the footwork for a weekly social dinner group, etc. My "secret" may simply be that we don't have kids, just cats, and that our ideas of housekeeping are somewhere closer to graduate student than to Martha Stewart. Practice is another factor. I have much more of a life than I did five years ago, and infinitely more of one than I did ten years ago. There is no right answer, merely an eternal juggling act. Why bring it up? Because when you start bidding out your time and giving binding hourly estimates, you need to factor in the time that you need to just keep up your skillset and to take care of critical business and personal issues.

I find that I spend 10–15 hours/week staying on top of mailing lists, Web sites, and general "whatever" tech over and above what I can bill for, just to keep myself "in the game" as it were. I am fortunate not to have any major medical problems (knock on wood), but I know people who have to spend several hours a week in transit to doctors' offices for allergy shots, RSI physical therapy, and similar things.

Then there are the non-business, non-personal items. I have responsibilities to two organizational boards, a local sysadmin group (BayLISA) and a national group (SAGE). I generally have at least one workday conference call or document review request per week, plus off-hours meetings twice a month which require some prep work. [*As of the time of publication, Strata Chalup had resigned from the SAGE Executive Committee – Ed.*]

Last, but by no means least, there is the overhead of the consulting itself – doing invoices, writing proposals and statements of work, etc. A detailed statement of work for a standard 250-400 hour contract may take 8-10 hours of solid work to put together the first time. The next one for a similar job could take three to six hours, depending on the similarities, how well you've set up a template, and how comfortable you are with generating text in that kind of mode. If you haven't seen and worked with similar documents much, it may take more time than you think.

If your particular line of consulting includes presenting tutorials at conferences in addition to more conventional work, you may find yourself putting in even more overhead than you bargained for, despite best estimates. Even if you have taught the material previously, you will need to update it annually and practice your presentation with the updated material. You may need to re-propose it for each conference at which it is being considered for inclusion. Make sure you are setting your rates and priorities to allow for the prep time and for responding to proposal deadlines – many of which may come during hectic times at your regular contract. It all adds up!

Crunch!

One solution I have used is to redline when I have a crunch and then catch up with everything else by billing less time when things are less urgent. For instance, this spring I spent a fairly grueling six weeks of preparation for a hard external deadline at a client, namely performance test benchmark results and a presentation for a large annual meeting. Following this endurance run, I only did 78 billable hours over the four weeks from Feb 18 to March 19, essentially cutting down to roughly half time for a month.

Was that a Bad Thing? It could have been, if I hadn't made the effort to prepare for this kind of situation in advance. The initial contract I had set up specified a number of hours for the project and set dates for specific milestones. Part of the reason for the crunch was that we were trying to accomplish about nine weeks worth of work in six weeks, due to delays in getting the contract signed by the company's CFO, who was on vacation over the December holidays. I was up-front with the client and checked in every few days to make sure that nothing was burning down. I kept up with the day to day responsibilities, but slowed progress on the next big projects. And, of course, I'd just saved the bacon in a big way over the previous several weeks. So it was a bit unusual but it was okay – because of the way it was handled.

If you are careful with your client, you should be able to arrange periods of slack time within longer engagements. This presupposes a contract longer than a few weeks, and clients who are not depending on your work as a critical path or gating item for some other project. Most clients will appreciate the reminder that you only bill for hours actually worked, and that your temporary focus of attention outside work will not negatively affect their bottom line.

"OK, I Have Time Now. Hey, Where Is Everybody?"

That takes care of your clients, but what about your family and friends? It's true that "they'll understand," but understanding isn't enough. After too many disappointments, understanding turns into resentment. Make time before the next stage, when resentment turns into a kind of cynical bitterness from which it can be difficult to win back the ground you've lost. "I keep trying to make up, but he/she just won't let it drop!" is a sad refrain one hears all too often. A good sysadmin solves problems that haven't happened yet. Practice "preventive maintainance" with your family as well as on the job! An ounce of prevention goes a long, long way.

We all have friendships that used to be close and are now "every now and then." In some cases, this is just what we wanted, and we may have breathed a sigh of relief! Most of the time, though, we miss hanging out with our friends, and wonder what we could do, or could have done, to stay in touch. Managing friendships is made even more difficult by the fact that your friends are usually your peers, and they are proba-

Practice "preventive maintainance" with your family as well as on the job!

BIBLIOGRAPHY AND RESOURCES

I've created a site which contains these links, as well as other resources which I've found useful: *http://www.virtual.net/Ref/resources.html.*

Steve Simmons, "36.8% Overhead or Money, the Bottom Line on Consulting". *http://www.usenix.org/sage/best.of/consultant/bottom-line.html*

Celeste Stokely, "Breadth of Vision – A Key to Successful Consulting". *http://www.usenix.org/sage/best.of/consultant/breadth.html*

Tina Darmohray, "SAGE Focus: Being a Consultant". *http://www.usenix.org/sage/best.of/consultant/editorial.html*

Mark K. Mellis, "Fifty-One Weeks" *http://www.usenix.org/sage/best.of/consultant/fifty.html*

Dave Clark, "Making the Jump: Moving from Permanent to Contract Employment" *http://www.usenix.org/sage/best.of/consultant/jump.html*

Shawn Instenes "My Own Boss" *http://www.usenix.org/sage/best.of/consultant/boss.html*

Janet Ruhl The Computer Consultant's Workbook, reviewed by Brent Chapman *http://www.usenix.org/sage/best.of/consultant/review.html*

Janet Ruhl's Computer Contractors' Site *http://www.realrates.com/* *http://www.realrates.com/links.htm* (some UK-specific links here)

Celeste Stokeley's Contracting Page *http://www.stokeley.com/*

Jeff Barr's Consulting Info Page *http://www.vertexdev.com/~jeff/consult.html*

Small Business Administration (USA) *http://www.sba.gov/*

IRS FAQs for Small Business/Self-Employed *http://www.irs.ustreas.gov/plain/tax_edu/faq/index.html#Cat12*

USEFUL SLASHDOT DISCUSSIONS ON CONTRACTING.

Many were initiated by folks doing contract programming, but the discussions are general enough to be useful to any consultants, covering insurance, dealing with clients, caveats on billing, etc.

Contractor's Cut of Billing Rate? *http://slashdot.org/askslashdot/01/03/19/1149243.shtml*

Finding Legal Coverage as a SubContractor? *http://slashdot.org/askslashdot/01/03/01/2016217.shtml*

bly following just as insane a schedule as you are. Even if you only see them at conferences or technical meetings, at least try to stay in the loop. Send email now and then, with news about your life and family, and ask about theirs. It's a small thing, but it keeps the connections open between you.

You're a techie, so don't be afraid to make technology work for you. Even if you've eschewed day planners and PDA's with syncable calendars, you can still set up cron jobs for birthdays, anniversaries, friendly "whuzzup" pings to your buddies, and so on. Is it bogus or fake somehow to have your computer tell you to get in touch with your friends? Or does it show you care enough about them to make sure you don't lose track of how long you've been out of touch? It's up to you, and nobody else's business what makes it onto your calendar versus what you spontaneously remember.

One important caveat, which is pure common sense but which I see folks disregard all the time: make time to unwind! In fact, you should plan to make separate catch-up time for yourself, and catch-up time for other people, even family. These two are generally distinct items, since your version of downtime might be to curl up with a book and your partner or friend's version might be to go out and do something. If you take care of only you, or only them, eventually something is going to blow up unpleasantly. I've seen a lot of relationships fail that really didn't need to, but were pushed over the edge by consistent bad time management on the part of one or both partners.

AM I MY OWN BEST CLIENT, OR MY OWN WORST CLIENT?

Back to "you" again – are you making time to take care of yourself, watching how you eat, getting any exercise at all? It's so easy to go down that slippery slope, and then not get back up again! If you think that you'll have more flexibility and more time as a consultant than you had as a regular employee, I have good news and bad news for you. The good news is that you generally will, but the bad news is that it won't happen for a few years. Oops! When you are working for yourself, you find that there seem to be even fewer hours in the day, and that you seem to have much higher stakes in your to-do items. When taking an hour off to swim or jog three times a week could mean missing a deadline at work, that's big and bad. When each client is likely to "fire" you at the end of the job (or at least, not re-hire you), it becomes huge and horrific. Don't take on as much as you think you can at first. If you truly do end up with spare cycles, you can write articles for *;login:*, go for an extra day of jogging a week, or download that new CPAN module you've been meaning to check out. Err on the side of caution so that you don't burn yourself out, and don't make your "normal" schedule one that never leaves time for self-maintainance.

Unfortunately, I'm a great example of this. I came to California in 1991, already overweight. I was in moderately decent shape, since I worked primarily on campus for a few years preceding that, and was always walking from building to building, down to Central square for lunch, off to Harvard Square to shop, etc. I came out here and found that if you're not in SF or Berkeley, you drive everywhere. Period. I picked up about 10 pounds a year, for five or six years, and didn't really think about how that would add up. Now it's much harder to deal with!

This leads us to another "great truth" of consulting, but one that works just as well for employees: Knowing what to do doesn't mean you will always do it. Part of the art of what we do professionally is knowing what corners we can cut, since there's almost never both the resources and time to do things perfectly. In an ironic quirk of fate, this field tends to draw perfectionists like flowers draw bees. Whoops! You can beat your-

self up over a mistake endlessly, but are you learning from it or just doing drama on yourself? There's an interesting lesson in the Jewish attitude toward this. The Hebrew word that is often loosely translated as "sin" actually is much closer to "mistake," and comes from an archaic word that means "missing the target." When you miss the mark, you turn (teshuvah) to re-aim yourself. The capstone is that your turning is not complete until you are presented with a similar opportunity and succeed at not making the same mistake! Perhaps a more succint phrase is offered by a button I purchased many years ago at a science-fiction convention: "Oh no, not another learning experience!"

In Summary

Before you start consulting, think about what kind of work you want to do and what kind of business structure makes sense for you. Get familiar with paperwork – employment contracts, statements of work, tax forms. Figure out how much to charge, how many hours you are able and willing to work. You are creating an image of yourself and your business whether you mean to or not, so create a good one. Make sure to practice give-and-take with your peer community and the larger community.

Finally, remember that one of your motivations for working so hard is probably to earn a lot of money so you'll be able to enjoy good things. Make sure that you still have some sanity and health with which to enjoy them, and family and friends who aren't too neglected to share in the fun. You *will* make mistakes – learn from them and move on!

I am always happy to meet other contractors, and encourage other people to get into contracting. Only you can tell whether or not contracting will be a good fit for your work habits, lifestyle, and financial needs. I hope that this article builds on the excellent work done by others and provides some additional tools with which to make that decision.

Is There Still a Market for Contract Programmers?
http://slashdot.org/askslashdot/01/01/18/2242246_F.shtml

Group Medical Insurance for Contract Programmers?
http://slashdot.org/askslashdot/01/01/09/0918238.shtml

Ask Slashdot: Contract Programming
http://slashdot.org/askslashdot/98/11/10/1242214.shtml

Ask Slashdot: Employees or Contractors?
http://slashdot.org/askslashdot/99/09/09/0022211.shtml

I know there's even more good information out there, so send me your favorite links or resources, and I'll update the Web area at *http://www.virtual.net/Ref/resources.html* Resources which are non-US specific are especially welcome.

what is your problem? that email was fine!

An Exploration into How Not to Write Flame Mail

by Christopher M. Russo

Chris Russo manages engineering at Genuity. His focus continues to be satisfying the customer – whether that be a Web developer, a system administrator, or an end user.

chris@thlogic.com

Most people in the technology industries read and send email every day. A significant number of these people read and send hundreds of such communications on a daily basis. Email has become a significant and powerful tool which has become critical in nearly every part of our society. It enables users to communicate a broad range of complex ideas, quickly, easily, and regardless of schedule or geography. It is immediate yet queued. It is also, however, arguably the most dangerous communications tool we have invented to date.

Most people have discovered that they have an extremely difficult time controlling how others perceive what they write. Where verbally we rely upon facial expressions, body language, and tone to express our feelings on a subject, in email we have none of these things. Therefore, personal catch phrases like "I don't think I like your attitude" (with a big smile and a playful glare) quickly turn into a personal offense and a gauntlet on the field.

Unfortunately, many other people have found that email can be a kind of shield. Since they don't actually have to look you in the face, they have the opportunity to write things that they would never dare say to any human being in a conversation. Where verbally this person might actually say nothing or possibly mumble a bit while walking away, they might actually say, "I don't think I like your attitude," and actually mean it. Worse, they might say, "It's not my fault that you don't know what you're doing," and it would be hard to say that and not mean anything other than what one would assume they meant.

There is a final group of people who simply feel that email is a tool and believe that the tool has a proper solitary and simple use. For example, what is a hammer, but an implement that is used to drive a nail into a piece of wood? This group often feels that people "need to relax" or should "stop reading between the lines." They feel that the words they place in their emails are factual and to the point, and there is little or no use for wasting countless hours putting flowery, soft words and phrases to coddle people; these people need to deal in the facts, not how the writer's statements "make them feel." The primary difference between this person and the shield person in the previous group is that these people might actually say everything they say in email to your face, and in fact will probably state with pride that they will do just that.

In writing this article, I don't intend to change you. What you do, who you are, and how you choose to write your emails to people is your business. What I do propose to do is educate; to give insight as to how emails function at the level beyond the presentation layer. Regardless of how you decide to write emails after this article, in better understanding how they work you may at least be able to deal more effectively with how people react to them. If you choose to embrace all, or even some, of the ideas presented here, I can guarantee that with practice and patience you will have positive results.

To begin, let's talk a bit about perception. All human beings function by taking what data they have available to them and filling in areas that they don't have in order to complete the picture. For example, if someone (let's call her Carol) is approaching me rapidly, she could be doing this for any number of reasons. While certainly I could not hope to mention all the possibilities of why Carol is racing towards me, let's propose that she could be doing this for one of the following reasons: she could be attacking me, rushing to me in alarm to tell me the building is on fire, happy and excited about something, or excited and mad about something.

How do I know which one it is? Well, first I might look at Carol's facial expression. Is she smiling or frowning? Is she, perhaps, crying? What does her body language say to me? Are her fists clenched? Perhaps she is reaching out to me? Does she look tense? What is she carrying? A fire extinguisher might be a clue, but then so might a knife. What kind of knife is that? Is it a birthday cake-cutting implement, or is it something you might see in the hands of a Dungeons & Dragons assassin? Finally, what is she saying? Or screaming?

In interacting with humans in person, we have a great many clues to indicate what a person may be thinking or feeling; in the example above, the clues that Carol is presenting us with are the same clues that we automatically use in our own lives to communicate our own feelings, whether we are trying to or not.

Consider a dog for a moment. If you have ever had or interacted with one, you may have realized that they seem to understand what your feelings are when you are talking to them. Regardless of whether you are happy, sad, or angry, they appear to understand and respond appropriately. Now, in the dog-mind-twisting spirit of Pavlov, have you ever tried opening your arms wide, looking really happy and excited and telling your dog at great length in a really excited tone how bad he was? Try it if you have not. The dog responds to that which he has available to him, and since by and large he does not understand a word you say, he will respond very happily because he perceives that you are happy and pleased with him. What do you think would happen if you deliberately made yourself look very angry and threatening and said in a gruff and loud tone how much you love the dog and think he's the most wonderful thing in the world?

Now pretend the dog is your coworker.

Seriously, think about that for a moment. How does the reaction of your coworker reading your email differ from your dog's reaction in this situation? Both your dog and your coworker are missing at least one key piece of your communication.

In the same way that your dog cannot understand the words you are saying, your coworker cannot see you or hear the tone of your voice. In fact, in most cases where you are not deliberately playing with the mind of Rover, the dog is clearly at a distinct advantage over your coworker. Because, regardless of the fact that Rover can't see or speak a word of English, he has a very strong understanding of what you are feeling because he is usually right that you're happy when you act happy, sad when you act sad, and angry when you act like a lunatic.

A coworker reading your email is denied the pleasure of your smiling face as you tell him what a waste of carbon he really is. If you were with him, you might have smiled and given him a playful shot in the arm, but you aren't . . . and you didn't. Since he knows that he is being severely insulted and has no other information to go on, he becomes insulted and proceeds to flame you back and carbon copy your boss.

A coworker reading your email is denied the pleasure of your smiling face as you tell him what a waste of carbon he really is.

Keep in mind that emoticons are really very unprofessional.

What is the key to irritating the heck out of everyone around us? The following sections illustrate some key areas that you can focus on to make your emails less offensive, or at least to understand why you're so good at making people so terribly angry with you.

Understand that Email Is a Written Medium

Anyone can print your email or forward it to anyone. You should consider refraining from writing anything that you don't want printed in the *New York Times*. Of course, sometimes you'll be referencing company confidential information or trade secrets, but the point is that you don't want to be the brunt of criticism if your written note is disclosed for any reason (i.e., security, anger, or something unanticipated).

Pick Up the Phone, Walk Down the Hall

Absolutely the simplest and best way to avoid e-flaming anyone is to avoid emails altogether. This may sound silly, but it's true. A large number of problems with bad email wars come out of the simple fact that the problem is way too hot to easily deal with "face-to-face," so people hide behind their electronic shield.

In these tough times, it is best to work with the person directly. Yes, it is harder, but at least each of you will be in the same room, see each other's facial expressions, and be less likely to lob insults and demands all over the place.

The more difficult the situation, the harder this will be, but this is clearly the most natural method of handling conflict aside from beating someone with a Tyrannosaur bone . . . which I do not personally recommend.

Consider Instant Messages

While still potentially dangerous, using "instant messages" can be helpful because they are more dialog oriented. In an email, you simply get up on your soap box and wax on about how badly people perform their job. In an instant messenger you go back and forth, often one line at a time. Frequently this type of "discussion" results in your getting a key piece of information that you were unaware of that diffuses the situation entirely.

Emoticons

It is a good idea to use an occasional emoticon ":-)" to show that you are happy and smiling as you write. A simple sentence like "I demand quality, and I won't stand for this infraction!" sounds pretty hostile. Add a simple ";-)" to the end, and suddenly you're just kidding around.

Certainly, don't add emoticons where they are inappropriate. If you are really trying to be serious and drive home a point, a happy emoticon is going to undermine that effort.

Also be certain not to use too many. If you find that you have to pepper your email with emoticons, then you're probably so angry that you should cool off and write this later, or possibly not at all.

Keep in mind that emoticons are really also very unprofessional. I shudder to even insert one in this article. In fact, I even loathe having to write the word "emoticon" itself. If you are writing a formal letter, something for a broad audience or your boss, avoid emoticons entirely. Sometimes using one in these situations would be a "bold

move, admired by all," but not usually. That's more of an advanced emoticon usage tactic. (I'll refrain from putting one more smiley face here, just picture me grinning.)

Be Self-Effacing

Oftentimes, half of the battle is keeping the person reading your email from taking a defensive posture. If you can do that, you can give yourself a margin of error with the person because they will simply disregard minor transgressions.

A really great way to do this is to make a self-effacing statement. This, however, is a dangerous art because you run the risk of (1) making yourself, and possibly even your team or group, look idiotic or (2) looking insincere or phony.

Some examples of self-effacing statements:

"I could be wrong, but . . ."

"I'm sorry, I'm a bit new to this process . . ."

"I think I'm confused . . ."

"You're far better at this kind of thing than I am . . ."

"If I've made any misstatements, please do let me know."

Statements such as these have a disarming effect because you are immediately saying to the reader that you are not necessarily certain, and especially that you are open to dialog and possibly even acknowledging that the reader may be more informed or capable than you.

Statements like these can turn a potentially hostile sentence like "You made a mistake and used the wrong template" into something much more approachable like "I'm sorry, I'm a bit new to this process, but it looks like you may have used the wrong template."

Ask for Help

This is another great method of putting yourself in a less authoritative position and ensuring the reader is not in a defensive one. The simple method of asking for help immediately says to the reader, "Hey, I need you and I acknowledge that you may have an understanding, information, or resources which I do not."

Some examples of asking for help:

"I was wondering if you could help . . ."

"I was wondering if you knew . . ."

"Thanks! I really appreciate your help!"

Employing this method can turn a single-sentence email of "Put those papers away" into something much less commanding like "I was wondering if you could help me out by putting those papers away? Thanks!"

Offer to Help

Just as powerful as asking for help is offering to help. Initially, one might think that this would put people on the defensive since you are clearly indicating that they might need assistance. However, what usually winds up happening is that the person feels that regardless of their power position, you are extending your hand in an effort to do

Just as powerful as asking for help is offering to help.

A great way to defuse a hostile situation is to simply throw in a couple of complimentary statements.

whatever you can to try to make things better for them. When offering help, you may even find that your assistance may actually aid in fixing that which was irritating you so much to begin with.

Needless to say, it's important that you follow through on these offers, as not doing so will eventually make you look insincere and predispose people to not trusting you.

Be Complimentary

A great way to defuse a hostile situation is to simply throw in a couple of complimentary statements.

Like many of the tactics described in this article, you need to be careful not to come across as fake or phony, for if you do you are likely to anger the person all the more. An easy way to avoid this pitfall is to simply compliment people on things that are genuinely true and relevant to the situation.

In other words, if you are discussing the next big project you are working on with the addressee, don't compliment them on their shoes, but certainly a statement of how great they were on the last project, or how you "couldn't have done it without them," will be appreciated and worth a lot in your efforts to keep them happy and calm.

Pretend it's Face-to-Face

When writing your email, make sure anything that you write is actually something you would be willing to say to the addressee if you were speaking to them face-to-face.

For example, do you really think you would say "I don't like your attitude" directly to someone if you were talking to them? Probably not, unless you were looking to get punched in the face. However, you might say something more along the lines of "I'm not really very comfortable with the way this is going. Maybe we should talk about this later."

Put it Aside; Let it Cool

If and when you write something that you are not entirely sure is going to come across the right way, it's often a good idea to put the email aside for now and come back to review it later. It's often best to wait a full 24 hours.

If, when you return, you read the mail and still feel that it is okay to send, then go ahead. Usually, people find that a day later they are not nearly as angry as they were when they first began writing the email, and are less inclined to send it the way they wrote it, or even send it at all.

Don't Write While Angry

If you're angry when you are writing an email, it will almost invariably show up in the way your mail reads to the recipient. You could even be angry about something completely unrelated to the content of the communication or the addressee, and it will still seem to the person reading it that you are angry at them.

For some people, this can be true even if you are mildly irritated. Basically, if you are in a bad mood, overly tired, angry, bitter, or any other negative emotion, it's best to avoid sending people email if you can.

Needless to say, the better you are at mastering some of the other more active email softening techniques, this becomes less of a problem as you are more aware of what it is that you are putting down on "paper."

Beware Brevity

Brevity in an email can be very dangerous. You will notice that in most of the examples shown in this article that we add something to the sentences to soften or defuse them, rather than just commanding someone to "Put those papers away."

Telling someone to put those papers away is clearly to the point, but it comes across as a very rude and abrupt command. The longer, "I was wondering if you could help me out by putting those papers away? Thanks!" clearly gets your thoughts across but avoids making you look like you're trying to be Lord Jerk, King of the Office.

Keep in mind that there is an unfortunate flip-side to this. Sometimes if your email is too long, you will come across as phony or just plain irritating. Some people simply cannot stand verbose email. In time, as you learn some of the other methods, you will actually be able to write fairly brief emails and get away with it.

Ask Someone Else to Review

It's always a good idea to let another, preferably impartial, person check your email before you send it. I like to pick a person who I know is not associated in any way with the situation, because they are far less likely to be upset about the same thing I am.

Keep in mind the mannerisms of the person you are picking, however. If you choose someone famous for flaming everyone within his or her reach, you are not likely to get a very useful review.

I also recommend having your boss review such emails. This allows your boss to know you are frustrated over something, while at the same time ensuring that she knows you are trying to do the right thing. Finally, if she says it's a good message and you get attacked for flaming someone after all, your boss is really the one responsible for giving you her approval prior to it being sent. It's certainly not their fault for the flame, but at least you won't be alone when the heat comes back in your direction.

Know Your Audience

In all cases, in all things, it's important to know your audience. Who will be reading this email? What kind of person are they? Are they sensitive? Or perhaps they have bark skin and a stone heart? Even if they are less sensitive, would they be particularly sensitive to this issue which you are raising?

The point is that no one method works for every person, every time. Carefully monitoring your audience and thinking of how they are likely to react to a given situation will be an enormous aid to you in writing email messages.

For example, I presently work as a manager in the Engineering Department of my company. In Engineering, things are a bit on the formal side, and people tend to be fairly sensitive to the content and delivery of email. However, we also work very closely with the Operations group. They, too, can be sensitive to email, but usually only on certain subjects. What's more, their email is frequently offensive and derogatory, though usually through indirect means.

Since I know these things, I try very hard to ignore the seemingly nasty remarks in emails from Operations, but also don't work too hard to keep my email flame-free, unless I know I'll be hitting on something that is a sore spot for that group.

What's more interesting is that if I write an email to Operations that is more formal and filled with extra softening, as I often need to do when writing to Engineering, they

In all cases, in all things, it's important to know your audience.

It is always important to understand that any word can have multiple meanings or implications.

actually get more angry at me than they would if I had just rattled off a quick two-line message.

This type of scenario is common and makes the art of sending email infuriatingly complicated, because it means that while you can and will need to use many of the ideas contained within this article to keep people from showing up at your door with pitchforks and torches, sometimes using them with the wrong people will yield exactly the result you are trying to avoid.

Watch Big Lists

As part of knowing your audience, understand that big lists of addressees can be particularly dangerous. The more people you have on your list, the more likely any one of them is going to be annoyed by your email.

What's worse is that if someone is annoyed, there is a very good chance that when they flame you, they will flame the whole list, which is likely to make you pretty angry, like you have been publicly attacked.

I could easily write a whole article on list etiquette alone, but here are some basic tips:

First and foremost, do your best to keep the number of recipients to a minimum. Simply stated, the smaller the audience, the fewer the unexpected personality traits that are going to react negatively to your message. Sending an email to 100 people is like tossing a 20-pound block of sodium into the public pool during adult swim on Labor Day.

Remember that a good portion of avoiding a flame war is to keep the person on the receiving end of your email from being on the defensive. Carbon copying your boss, their boss, or anyone else of authority is likely to immediately make everyone uneasy. Don't include people like this, unless you absolutely need to.

When attacked, don't respond. Most of the time the person who lashes out at you looks far sillier to the people on the list. Let him hold his title and move on. If you must respond, deal only with the issues the person raised, not with their attack.

Consider sending blind carbon copies ("bcc") to people who you would like to see the message, but you don't want "on the list." Understand that bccs are dangerous because the act of using them is inherently deceitful, because you are hiding the list of true recipients. The people who receive the mail may then not realize that they were only blind copied and respond to all. When they do this, someone may realize your deception and become very angry. Sometimes you can send an email to the intended bcc recipients prior to sending the actual mail, warning them of the message they are about to get and not to do this. Sometimes, it's easier just to forward them a copy after you have already sent it.

Watch Implications

It is always important to understand that any word can have multiple meanings or implications. There are a great many words that can be used to say essentially the same thing; but some, depending on the context, have additional meanings which can be quite destructive in your efforts to communicate peacefully.

For example, I could refer to someone who does not react strongly to emotional stimuli as "insensitive." Insensitive certainly says this person can take some abuse without breaking down, but it also implies that the person is somehow unkind, uncaring, or callous. I could soften this up a bit and say that the person is, perhaps, "less sensitive."

This is pretty reasonable as it seems neither complimentary nor derogatory. However, if I called the person "stout" or "stalwart," this might assign the person a characteristic which most people find to be complimentary. Suddenly I've run the gamut from being offensive to complimentary, and I've been essentially saying the same thing.

It is obviously important to choose words that best fit what you truly mean, what you want to get across, or at least that will say roughly what you want without upsetting people.

Large Words and Formal Tone

The use of large words and formal tone can often come across as authoritarian and sometimes even patronizing. This immediately puts the person in a defensive posture and is likely to cause some pretty serious emotions to come up.

For example, phrases such as . . .

"It has come to my attention . . ."

"I would like to discuss the matter at hand . . ."

"Calibration of the units in question may cause significant problems with our schedule . . ."

These phrases are quite formal indeed. However, you can easily soften these up. For example:

"I just noticed something . . ."

"Hey, do you have a few minutes for us to talk today?"

"If we calibrate those, we may have some problems with our schedule."

The trick with this is to write to your email recipient in the same way you would talk to them face-to-face. If you happen to be the kind of person who says things like "It has come to my attention . . ." in everyday conversations, then you may have deeper issues to which the solutions are far outside of the scope of this article.

Capital Letters

Astoundingly enough, some people still do write email in all capital letters. It is pretty rare, but it does happen. For that reason, I feel I must take the time to issue this very rudimentary warning:

DO NOT USE CAPITAL LETTERS. CAPITAL LETTERS MEAN YOU ARE YELLING!

Whoops. Well, I assume you get the point.

Low-Grade Hostility

Low-grade hostility is a very difficult thing to describe. In essence, it is putting yourself in a position of authority by using certain key words and phrases in your sentences. Interestingly enough, this is almost as frequently a problem in verbal communications as it is in written ones.

The best bet is to avoid telling people what you think they "need" to do or be. Do your best to avoid direction and attempts to subtly point out their failings. If you feel the need to go after these areas, it's often best to ask leading questions that will allow the person to highlight their own issues.

The best bet is to avoid telling people what you think they "need" to do or be.

For example:

"Did you really want to put that there?"

"Hey, did you notice that you were standing under the bridge that you are planning to blow up?"

"I'm not sure if you know, but that's my arm under that knife."

Be careful when using questions as alternatives. As shown in the last example, they can quickly and easily become rather sarcastic.

Direct Hostility

Needless to say, tearing a person to pieces and saying all sorts of uncomplimentary things about his or her mother is not going to win you any friends. One would assume that this is pretty obvious, but the piece would not be complete without it.

Now, let's try an example. I happen to be fairly annoyed at a coworker of mine, so I will write an email that I would really enjoy writing to her, and I will follow that with a much more reasonable replacement that is less likely to get me fired . . . or beaten.

Flame Mail Example

Lucy-

You know, despite the fact that we've worked together for two years, you still insist upon assuming that my team is incapable and completely forgetting that I and my team have completely revamped a mess of a development process. This is clearly indicated by your attempting to micromanage my team, instead of dealing with your own...which, by the way, is falling apart.

I would really appreciate it if you would deal with your own issues, and if you feel compelled to bring up issues that you think my team is somehow falling down on, that you either tell your own staff so that they can bring them to my attention, or that you address them to me directly.

I would ask that you no longer engage my staff members directly. If I hear of this happening again, I will speak with your management.

-Chris

Fixed Flame Mail Example

Hi Lucy, how are you doing?

Hey, I was a little confused and was wondering if you could help me out with something. As you know, I've taken on a new team and have been given the responsibility of revamping it in a manner similar to what I've done with my existing team.

You and your team were a great help to me in my early days with that Engineering team, by the way. I know I've said this before, but I feel I can't really say it enough.

Anyway... some of my newer team members have been telling me that you've been stopping by and asking them to do things that are outside of our standards, asking them to set up meetings that they feel I need to be conducting, etc. They have pretty much all said that they would really like to do whatever we need to in order to do the job properly, but are confused as to why you and I seem to be out of alignment.

Short of making a visit to the chiropractor ;-), I was wondering if we could possibly work out a better way to get you what you need. I thought that maybe we could have a weekly or semi-weekly meeting between us to discuss your issues and make sure we are addressing them. I think this would be great, as it would give us a chance to work more closely on these issues and possibly also reduce any confusion between our two teams.

What do you think?

Thanks, Lucy- I really appreciate your time and help!

-Chris

Exercise

Now that you've read all the way through this article and studied the examples above, here are a few exercises that should help you to practice some of these techniques.

1. Go over the example flame mail above and write down a list of every bad thing that you think I may have done. Attempt to cross-reference them with some of the behaviors mentioned in the article. Consider for a moment how obvious these things are to you, and think about whether or not they would be so obvious if you were writing them yourself, when angry.

2. Go over the example fixed flame mail above and write down a list of every good thing that you think I may have done. Attempt to cross-reference them with some of the behaviors and tactics mentioned in this article. Now attempt to cross reference the points made from the flame to the fixed flame. Notice the differences in the wording and how it affected the tone. Is there anything that was missed?

3. Try to find some emails that you have written in the past that have been perceived as being harsh or abusive. Spend some time trying to rewrite these in a manner that is less offensive or hostile. Consider why the message was so hostile when you wrote it. Were there any simple behavioral things you could have done that would have toned it down a bit?

4. Lastly, create a quick checklist of things that you should consider when writing any email. Print this and post it on the wall next to where you are likely to spend time writing mail. Refer to this each and every time before you press send on a message that could come across as nasty.

There are a lot of lessons to learn, many subtleties and nuances to master and, for many people, many years of practice to get it right. If you truly want to accomplish this, you certainly can. Just be patient and keep practicing. In time, you will be able to write inoffensive email in the harshest of conditions without even trying. Good luck!

the law moves in

by Edgar Danielyan

Edgar Danielyan is a Cisco-certified network, security, and design professional, as well as a certified paralegal. He is the author of *Solaris 8 Security* as well as many articles on the Internet, UNIX, and security.He is currently a self-employed consultant and author.

edd@danielyan.com

The Convention on Cybercrime

The Internet has long been perceived either as lawless cyberspace inhabited by hackers and criminals or as a modern-day utopia, ideal environment for freedom, democracy, and libertarianism. As always, the truth lies somewhere between these polar viewpoints. While empowering individuals to share and disseminate information and ideas freely, it is also a powerful tool and medium for all kinds of crimes – ranging from simple theft to terrorism and espionage. Fortunately, it seems something is finally being done by the governments to combat international cybercrime. This article briefly introduces the flagship move on this front: the Convention on Cybercrime, drafted by the Council of Europe and now signed by thirty states, both members and non-members of the Council. While it is too early to say whether this legal instrument will actually improve the current state of affairs, it is nonetheless the first of its kind and deserves consideration.

The Convention on Cybercrime

On paper, the Convention on Cybercrime seems to be what was needed to combat international cybercrime. It defines as criminal offenses certain acts, such as illegal access to systems and data, and provides the legal and procedural framework for the investigation and prosecution of persons committing these crimes. However, it will be up to national legislatures and courts to enact and enforce the provisions of the Convention, and economic, legal, and administrative differences between signatory states will inevitably mean that international prosecution of cybercriminals remains a tough task. There is no doubt that the Convention will be ratified sooner or later, and that the legal landscape will change considerably after it comes into effect. Hopefully, it will also convince those of us who don't believe in legal action against cybercriminals to at least try to prosecute them. In the meantime, corporate legal counsel and law firms should inform themselves about the possibilities and procedures introduced by the Convention and stay tuned for the day it becomes law.

The Convention on Cybercrime was adopted by the Committee of Ministers of the Council of Europe on 8 November 2001 after years of consultations and work on the document. Shortly thereafter, on 23 November 2001, the Convention was signed at the Hungarian Parliament in Budapest by the following thirty states: Albania, Armenia, Austria, Belgium, Bulgaria, Croatia, Cyprus, Estonia, Finland, France, Germany, Greece, Hungary, Italy, Moldova, the Netherlands, Norway, Poland, Portugal, Romania, Spain, Sweden, Switzerland, Macedonia, Ukraine, United Kingdom, Canada, Japan, South Africa, and the United States. All these states, with the exception of Canada, Japan, South Africa and the United States, are members of the Council of Europe. The Convention itself, with its four chapters and forty-eight articles, is written in plain English and avoids legal language as much as possible. It will come into effect and become binding on the signatory states after ratification by national parliaments. As soon as five signatory states, at least three of which must be Council of Europe members, ratify the Convention, it will start its life as a working international legal instrument. In the preamble to the Convention, reference is made to the following international legal instruments which were taken into account by its drafters:

- European Convention for the Protection of Human Rights and Fundamental Freedoms (Council of Europe, 1950)
- International Covenant on Civil and Political Rights (United Nations, 1966)
- Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data (Council of Europe, 1981)
- Convention on the Rights of Child (United Nations, 1989)
- Worst Forms of Child Labour Convention (International Labour Organization, 1999)

The influence of the European Convention on Human Rights (ECHR) on the Convention on Cybercrime is especially profound and may be seen in Article 15, "Conditions and safeguards." This article stipulates that all provisions of the Convention are subject to the protection of human rights and fundamental freedoms guaranteed by the ECHR. However, since these guarantees are only applicable in the member states of the Council of Europe, Article 15 expressly states that application of the powers and procedures provided for in the Convention is subject to the principle of proportionality. The Convention also provides for judicial or other independent supervision, and limitation on the scope and duration of powers and procedures arising from the Convention.

The influence of the European Convention on Human Rights (ECHR) on the Convention on Cybercrime is especially profound.

Chapter I: Use of Terms

Appreciating the fact that computer terminology may be a potential source of confusion, many concepts are first defined before being used. Article 1 of the Convention, entitled "Definitions," defines terms such as "computer system" and "computer data":

> "computer system": any device or a group of inter-connected or related devices, one or more of which, pursuant to a program, performs automatic processing of data; "computer data": any representation of facts, information or concepts in a form suitable for processing in a computer system, including a program suitable to cause a computer system to perform a function.

Chapter II: Measures to Be Taken at the National Level

SECTION 1: SUBSTANTIVE CRIMINAL LAW

Title 1: Offenses against the confidentiality, integrity and availability of computer data and systems

Articles 2-6 establish the following actions as criminal offenses: illegal access, illegal interception, illegal data interference, illegal system interference, and misuse of devices.

The term "illegal" is defined as action without right; in some cases, signatory states may require that such actions be committed with dishonest intent.

Title 2: Computer-related offenses

Articles 7-8 establish the following actions as criminal offenses: computer-related forgery, computer-related fraud.

"Fraud" is defined in Article 8 as "any input, alteration, deletion, or suppression of computer data; any interference with the functioning of a computer system, with fraudulent or dishonest intent of procuring, without right, an economic benefit for oneself or for another."

Title 3: Content-related offenses

Article 9 defines as criminal any offenses related to child pornography. For the purposes of this article, "children" includes all persons under 18 years of age. By a reservation to the Convention, however, a party may require a lower age limit, which shall be not less than 16 years.

Title 4: Offenses related to infringements of copyright and related rights

Article 10 criminalizes infringement of copyright as defined by national law of member states pursuant to international obligations of member states.

Title 5: Ancillary liability and sanctions

Articles 11-13 establish as criminal offenses any attempt to commit, or to aid or abet the commission of, any of the offenses established by Articles 2-10; provide for corporate liability of legal persons; and provide for sanctions and measures intended to ensure that criminal offenses established by the Convention are punishable by effective, proportionate and dissuasive sanctions, which may include deprivation of liberty.

Section 2: Procedural law

Title 1: Common provisions

Articles 14-15 specify the general principles of procedural law and define the scope of applicability; specify the conditions and safeguards for the application of the Convention with reference to the norms enshrined in the European Convention on Human Rights.

Title 2: Expedited preservation of stored computer data

Articles 16-17 deal with expedited preservation of stored computer data and disclosure of traffic data.

Title 3: Production order

Article 18 covers interjurisdictional production orders.

Title 4: Search and seizure of stored computer data

Article 19 defines how computer data can be searched and seized.

Title 5: Real-time collection of computer data

Articles 20-21 cover real-time collection of traffic data and data interception.

Section 3: Jurisdiction

Article 22 deals with questions of jurisdiction of states in accordance with Articles 2-11 of the Convention.

Chapter III: International Cooperation

Section 1: General principles

Title 1: General principles relating to international cooperation

Article 23 stipulates that signatory states shall cooperate with each other to the extent provided for in their international agreements.

Title 2: Principles relating to extradition

Article 24 defines the procedures and requirements for extradition for criminal offenses established by Articles 2-11 of the Convention.

Title 3: General principles relating to mutual assistance

Articles 25-26 specify when and how signatory states should or may offer mutual assistance in matters covered by the Convention.

Title 4: Procedures pertaining to mutual assistance requests in the absence of applicable international agreements

Articles 27-28 define how parties may offer mutual assistance when there are no mutual legal assistance treaties between them. In particular it provides for requests for assistance to be made through the International Criminal Police Organization (INTERPOL).

Section 2: Specific provisions

Title 1: Mutual assistance regarding provisional measures

Articles 29-30 deal with expedited preservation of stored computer data and disclosure of preserved traffic data.

Title 2: Mutual assistance regarding investigative powers

Articles 31-34 provide for mutual assistance in international investigations under this Convention.

Title 3: 24/7 Network

Article 35 provides for the establishment of 24/7 points of contact in all signatory states which are to provide assistance defined in the Convention. It also requires all parties to ensure that trained and equipped personnel are available to satisfy the requirements of this article.

Chapter IV: Final Provisions

Articles 36-48 deal with legal and administrative procedures of the Convention.

Summary

On paper, the Convention on Cybercrime seems to be what was needed to combat international cybercrime. It defines as criminal offenses certain acts, such as illegal access to systems and data, and provides the legal and procedural framework for the investigation and prosecution of persons committing these crimes. However, it will be up to national legislatures and courts to enact and enforce the provisions of the Convention, and economic, legal, and administrative differences between signatory states will inevitably mean that international prosecution of cybercriminals remains a tough task.

The full text of the Convention on Cybercrime may be obtained from the Council of Europe at *http://conventions.coe.int* along with its Explanatory Memorandum.

ask mr. know-it-all

by Trey Harris

Trey Harris is Secretary of the SAGE Executive Committee. He lives in New York.

trey@sage.org

The "S" Debate

Dear Mr. Know-it-All,

My mommy's business card says she's a "system administrator." My daddy's says that he's a "systems administrator." I asked them at dinner tonight which one was right, and they sent me to my room without dessert! So I went to the SAGE Web site and it says that SAGE is the "System Administrators Guild." Does that mean that my mommy is right and my daddy is wrong? And by the way, didn't you leave something out of SAGE's name? Like an apostrophe?

Love,

Johnny
Sheboygan, WI

Dear Johnny,

First off, you shouldn't go rifling through your parents' wallets for their business cards. That's what probably got you sent to your room to begin with, not an argument over what a system administrator – or is it systems administrator? – is called.

That said, the first rule in linguistics is, if enough people say it, then it's "right," for some definition of "right." So by that yardstick, both are right. But Mr. Know-it-All would suggest that the way SAGE does it is the more linguistically sound.

Here's why: people like your father reason that since the person may have more than one system to administer, it should be a "systems administrator," because a "system administrator" would be someone who only has one system to run.

That would be fine if human language in general, and English in particular, ran by the logic of the real world. But it doesn't. It runs by its own, sometimes peculiar logic. And in the logic of the English language, the word "system" in "system administrator" is not actually a noun – remember, Johnny, a noun is a person, place, or thing, and it's a word that we can put an "s" on the end of to make it plural.

You see, a system might be a person, place, or thing – actually, it *is* a thing, or maybe it's a place, but it sure as heck isn't a person – anyway, it might be a noun, but when attached to another noun, "administrator," it becomes something else. An "adjunct," you might say, if you like fancy linguistic words.

In any case, that adjunct doesn't act like a noun, even though it still sounds like the noun it used to be. Instead, it acts like an adjective – you know, a descriptive word, like "blue," or "floccinaucinihilipilificatory." And adjectives in English don't get to have the plural marker "s".

You can see this in a lot of other cases in English where this happens – your teacher might call them "compound words," (not "compounds words!"), though linguists have much fancier names for them. For instance, a miner who mines diamonds is a "diamond miner," not a "diamonds miner," even though he'll mine more than one diamond during the course of his career. One hopes, anyway. And that pretty picture tube sitting on your desk is a "color monitor," not a "colors monitor," even though it can display more than just one color.

"Emergency management" (not "emergencies management"), "dog catcher" (not "dogs catcher"), "used-car salesman" (not "used-cars salesman"), "pastry chef" (not "pastries chef") . . . the list goes on.

(Just for accuracy's sake, the actual rule in English makes an occasional exception for words with irregular plurals. A person who hates mice might as easily be a "mice hater" as a "mouse hater," but someone who hates rats would always be a "rat hater," never a "rats hater.")

Not to say that there isn't room for disagreement. Libraries are rife with counterexamples, where there is a "periodicals department," presided over by a "periodicals librarian," and a "microforms department," with a "microforms librarian." (Though these two may be because of the unfortunate connotations of their adjectival counterparts – "periodical librarian" makes Mr. Know-it-All think that he or she may only occasionally show up for work, and "microform librarian" puts one in mind of a minuscule person squeaking, "shhh!" And even libraries can go the other way. Every "reference librarian" Mr. Know-it-All has ever met has had at least a dictionary *and* a thesaurus to call upon.)

In perhaps the most analogous counterexample, most companies and institutions that run their own campuses have a "facilities management" division. And the logic runs exactly the same. They manage more than one facility – for example, the water in addition to the electricity – so they are "facilities," not "facility," managers.

That said, these plural forms sound newfangled and contrived to Mr. Know-it-All's ears. As if it is hoped that adding an "s" will somehow make the profession sound more important. If that's the case, why not add two?

Oh, that brings up something else – it has been suggested that, while "system administrator" might be the correct term for your father, your parents together would be two "systems administrators." Put the "s" on both, you see, to make the plural. Mr. Know-it-All has even heard a rumor – don't laugh – that some people think it should be "two systems administrator," following from "two attorneys general" or "two mothers-in-law."

Pluralizing both is just plain wrong. In English, the process of agreement – that is, changing the ending of one word so it matches some property of another – only applies to verbs, as in "I *code*, but he, she, or it *codes*." In "system administrators," there's no verb, so no agreement. Other languages, like Spanish, get to have fancy noun-adjective agreement, but you can't force it into English, no matter how much you might want to.

How about "attorneys general?" It turns out that "attorneys general" acts just like "system administrators," only backwards (backwords?). You see, in English, adjectives almost always come before the nouns they modify – but in a very few cases, like "attorney general" and "mother-in-law," the adjectives – "general" and "in-law" – come *after* the noun. These are all very old terms that entered the language when the adjective-before-noun rule wasn't as strict as it is today, and they just got "grandfathered" into the language when the newer rules took effect. A new term, like "system administrator," has to follow today's rule – no grandfather clause applies. Besides, it would have to be "administrators system" if you wanted to do that – the adjective still gets no plural, whatever the order they're in.

So, the long and short of it – if you have one sysadmin, you have a "system administrator." If you have two sysadmins, you have two "system administrators." If you have two thousand sysadmins, you're at LISA.

If you have one sysadmin, you have a "system administrator." If you have two sysadmins, you have two "system administrators." If you have two thousand sysadmins, you're at LISA.

Oh, and about that apostrophe – any sysadmin will tell you that it's a bad idea to put a special character into a name.

Until next time,

Mr. Know-it-All

Dear Mr. Know-it-All,

I now understand that my uncle is a "network administrator," not a "networks administrator." But can you also tell me whether he works in a "Network Operation Center" or a "Network Operations Center"? I am sure it's not a "Networks Operations Center" (and I don't really care if it's Centre).

Eagerly awaiting a reply,

Johnny

Dear Johnny,

I believe the real problem may be that, because of the recent economic downturn in the technology sector, and the fact that your entire family works in technology, you may be unhappy with your lot in life and are thus fixating on nomenclatural trivia as a sort of escape.

But I'll take a stab at your question, because it hinges on another reason people say "systems administrator," one that involves much less rationalizing – and is more linguistically sound – than the one I discussed earlier. Please note, however, that if you're wishing to come away less confused than you started, you may do better to take up meditation rather than reading on. (Breathe in through your nose, and out through your mouth, and be mindful of your breathing. A mantra may help.)

One of the more confusing aspects of English to native speakers of many of the world's other languages is English's dualism of the possessive. Many languages have a "genitive case," that is, a marker on a noun to indicate its association with another noun: for example, the Latin amicus curiae, or "friend of the court." Many other languages have an "adposition" – a little word or sound that precedes, follows, or circumscribes another word – to do the same thing, such as the Spanish Madre de Dios or "God's mother."

English, confusingly, has both, though native speakers use them in subtly different situations, much too complex to go into here. English has both a genitive marker – a "z" sound typically written as "'s" or just "'" – and a possessive adposition, the preposition "of."

(Note that English is by no means singular in this dualism. It is merely more confusing in its dualism, as the difference between the two is usually more clear-cut in languages that have both.)

The reason this is relevant is that English has a rule whereby a cohesive *noun*-"of"-*noun phrase* combination can be transformed into a *noun phrase-noun* compound by simply moving the noun phrase to the front. (The previous sentence is actually provably *wrong* in English, so wrong that Mr. Know-it-All rewrote it dozens of times before giving up, but it's close enough without subjecting you to a couple of semesters' worth of generative syntax so that we can use precise language.) For example, "minister of foreign affairs" becomes "foreign affairs minister," "secretary of the interior" becomes

"interior secretary," "college of arts and sciences" becomes "arts and sciences college," and, yes, "center of network operations" becomes "network operations center."

However, those cases where we prefer the genitive *cannot* be transformed in the same way. So "my best friend Mike's wife" cannot become "my best friend Mike wife." (Note that even though the genitive can be coerced, with awkwardness, into "of" and vice versa, the important thing is that subtle rule, still not entirely understood, which prefers one over the other.)

"Network operations center" is almost certainly correct, since, if forced to judge, most of us would probably accept "center of network operations" as being semantically equivalent, but bristle at "network operations' center." Note that this is a bedevilingly difficult distinction to tease out, because "network operations' center" sounds the same as "network operations center," and native speakers make grammatical judgments unthinkingly based on sound, not on syntactic rules of which they are largely unaware. In any case, it doesn't really matter, as the important thing is that we are willing to accept "center of network operations" as being almost exactly equivalent.

So, in the same vein, is "systems administrator" simply shorthand for "administrator of systems?" If yes, then "systems administrator" should be correct. If, however, we only grudgingly accept as pedantically correct that long form, as, say, we'd grudgingly accept "miner of diamonds" for "diamond miner," then "system administrator" would be correct.

Mr. Know-it-All still sides with the no-ess form. "Miner of diamonds" sounds silly and contrived, like "catcher of dogs," "salesman of used cars," or "chef of pastries" (and unlike "center of network operations"), and to his ear, "administrator of systems" sounds silly and contrived too.

To muddy the waters a bit further, there is mounting evidence that competing grammatical rules are all tried at once by the brain, and the one that gets a good answer first wins, even if there might be one that will get a better answer later. It seems completely plausible that, for many people, a rule that would generate "systems administrator" simply beats out another competing rule that would generate "system administrator," regardless of which one is more "correct."

Some linguists turn that theory on its head and suggest that a number of likely candidate outputs (say, both "system administrator" and "systems administrator") are produced early by some estimation process, and then all the competing rules are run in parallel, and if a number of them begin to converge on one of the candidates, it is selected for utterance. (This theory has been used to explain how speech errors like Spoonerisms or so-called "Freudian slips" occur.) It would seem plausible that this process of "spreading activation" could select for "systems administrator," since so many (possibly incorrect) rules would lead to that end.

The point is, it isn't clear-cut that "system administrator" should be preferred. But when an organization like SAGE has to select one or the other, the evidence seems slightly stronger for the no-ess form.

Until next time,

Mr. Know-it-All

*Alexios Zavras (*zvr@pobox.com*) contributed the second letter from Johnny.*

judgment

by Steve Johnson

Steve Johnson has been a technical manager on and off for nearly two decades. At AT&T, he's best known for writing Yacc, Lint, and the Portable Compiler.

scj@mathworks.com

and Dusty White

Dusty White works as a management consultant in Silicon Valley, where she acts as a trainer, coach, and troubleshooter for technical companies.

dustywhite@earthlink.net

"Good judgment" is a quality highly prized by our culture. It implies that someone is able to look objectively at a situation and sum it up correctly, detecting the flaws and virtues. A significant part of a practical education in many professions is being able to "debug." That is, an object is studied in order to detect any flaws it might have – this often involves repeated trial-and-error testing of the object. Then the defects must be found, again often by a lengthy testing process, and finally corrected.

Since technical managers typically were good technical people, one thing that gets them promoted is their ability to look at something, quickly assess its strengths and weaknesses, and root out and correct any flaws it may have.

Unfortunately, this mind-set is one of the most damaging traits a manager can have, and nearly every technical person who starts to manage needs to "unlearn" it when dealing with people. Many non-managers need to suppress this when dealing with people, as well.

To get a sense of why this is so damaging, imagine that you are the "object" being studied. Of course, when we go to the doctor, we are. Doctors spend years developing a "bedside manner" that will compensate, to some degree, for the inevitable sense of invasion we feel when we are treated as objects that need to be debugged (even though, in the medical case, this might be literally true!). We have all probably experienced a doctor or nurse who was less skilled at this and who made us feel humiliated as we were poked and prodded, half naked in a cold room on a hard table, by someone who didn't seem to care if we were alive or dead.

In the same way, managers who are seen as waiting for people to screw up (so the manager can correct them!) can quickly dry up the creativity and morale of their groups. Such groups become focused on avoiding their manager's hot buttons rather than on customers or product.

There is another real minefield concerning judgment. Most of us had to learn, at an early age, to please parents or other authority figures who were judging our actions. When a two- or three-year-old child displeases a parent, it is a big emotional deal for the kid. Few of us have developed so much security in later life that we can be judged without some ghosts of two- or three-year-old emotions stirring. We have seen people fly into rages, quit the company, or burst into tears when they felt judged by their manager, and in some cases the manager was not even consciously being judgmental.

How does judgment work? When we judge something, we construct in our mind a model for how the thing should be. Then we take in data from the world, and compare it to our model. Typically, we focus on those places where the world fails to live up to the model. Usually, if your thoughts flow this way, you will take the next step and try to "fix" the world, or whatever part of it you are judging.

What's wrong with this? One problem is that, by focusing on what is wrong, you are encouraging "away from" behavior. In other words, you are focusing on what you don't want, rather than on what you do want. If you are judging a person, your attempt to "fix it" is likely to take the form of "Thou shalt not . . ." statements. Unfortunately, telling people what not to do is de-motivating, ambiguous, and often ineffective.

Another problem with judgment is that the person being judged tends to feel less powerful. They probably don't understand the thought and emotional processes that lead

to your judgment. If they are judged, it may well bring up those early childhood feelings of being a little kid – not a powerful, resourceful state for them. If you want to empower your group and encourage their creativity and initiative, avoid judging them. Keep their focus on pleasing the customer, completing the product, or whatever. You really don't want their focus to be on your judgment!

There is also a moral and ethical issue that needs to be raised. Even though you might believe that you have the right to judge other people as human beings, you are likely to find very few others who would agree! It is time to examine your own motivation here. The trainer Ron Luyet believes that "unsolicited advice is a form of attack." The same is true of judgment. So why are you attacking them? Even if you feel that the ends justify these means, or that your job calls on you to change their behavior, the same ends can probably be obtained without attacking the other person.

As some people begin to understand the damage their judgment causes, they may become overly nice and stop enforcing any standards in their group. They want to be "pals" to their group, and this is almost as big a danger as judgment. This is especially an issue with performance reviews, which trigger strong feelings of judging and being judged. For now, we urge you to keep a couple of things in mind.

Your job is not about good and evil. Your job as a manager is to achieve certain goals with your group, not to enter the Kingdom of Heaven (save that for your home life!). Your job is about meeting the standards required to achieve these goals. There are many reasons why people may fail to meet these standards, ranging from simple ignorance to carelessness, worry, overload, and a variety of psychological and mental conditions that they or their loved ones might be facing. You needn't care about the why. Ask how rather than why, and focus on fixing the process.

By distinguishing between an employee's worth as a human being and the specific behavior that needs correcting, you will probably learn much about the why as well as the how. Moreover, the employee will feel empowered by the experience, not diminished, and see you as a source of help rather than judgment.

Finally, if you work with a judger, nearly all judgers have a secret that they try hard to keep from you. Once you know it, it makes them much easier to work with. They judge themselves more harshly than they judge anyone else. They won't admit it, but it's universally true – it comes with the territory of judging – and when you know it's there it becomes easy to see. Being aware of this allows you to view their judging with a degree of compassion that offsets their often irritating mannerisms. You realize that their judging is an expression of their own inner conflicts, and isn't directed "personally" towards you. And you can meet their judgments and criticism with support and a positive outlook that will be a balm for their souls.

when do you plan to retire?

by Ray Swartz

Ray Swartz started his financial plan in 1988. He has been "voluntarily" unemployed since January 2001 and doesn't plan to work in 2002, either.

raybo@idiom.com

[Editor's Note: Ray will be writing occasional columns on financial planning. I think many of our members are beginning to consider such things. RK]

Retirement means different things to different people. To some, retirement is playing golf every day. To others, it is traveling to faraway places. Some people don't have an interest in ever retiring. In fact, every person's retirement ideal is unique to them and their circumstances.

What is the same for all of us is the desire to have the money and time to do whatever it is we want to do when we stop working. For most people, this doesn't just "happen." It is something you have to plan for and arrange, usually over a period of years.

Many people view retirement planning as "saving as much as you can and putting it away until you need it." But this one-size-fits-all strategy may or may not work. It ignores these questions:

- How much do I need to retire?
- Where should the money I "put away" go?
- When can I stop working?
- Am I saving too much and sacrificing my current needs and wants?

While "saving as much as you can" is not a bad idea, it provides no real guidance about what to do today, nor is it based on any current need or desired outcome. A better approach is to put together a plan that takes your current situation and future needs into account and provides benchmarks for checking that your plan and results are on target. Without question, to have a nice retirement you have to plan to have a nice retirement!

A Desire to Retire

The first step in retirement planning is deciding what "retirement" means to you. Since retirement planning is about money, you should decide if you want to continue to work (part-time, project contracts, in a different field, etc.) as part of your retirement. That is, will you have to fund your retirement entirely from savings and investments or can you rely on some income?

You need to determine when you would like to retire. That is, in how many years do you plan to stop working? Often this will be determined by your company's pension arrangements or other commitments you might have. On the other hand, answering "as soon as possible" is perfectly acceptable, too.

Part of this process is trying to calculate how your retirement life will be different from your current one. Specifically, you need to estimate how your spending needs might change. Are you planning to go on cruises six months of the year? Do you want to move to Hawaii, where the cost of living is higher? What about increased health costs as you age?

As an example, suppose that you now spend your vacation time working around the house or driving to a friend's house in a nearby city. When you retire, you want to travel the world four months out of the year. In this case, your travel expenses will dramatically increase when you retire.

It may seem unrealistic to even guess what your expenses might be in future. You aren't looking for accuracy here. You only want to catch the obvious differences. There's a rule of thumb in the financial planning community that can provide guidance: absent unusual plans, the assumption is that retirement expenses will be 75% of what they

were pre-retirement. Note that some of the uncertainty may be removed using insurance and other investment options.

Getting There from Here

Once you rough out your retirement plans, you can start to put numbers to how much retirement will "cost" you. The next question is can you afford it? Put another way, what do you have to do today so that you can retire when and how you want?

The answer depends on your current financial situation. Given what you have invested, how much you make (and are likely to make in the future) and how much you spend, what, if anything, do you have to do to reach your retirement goals?

Your investments consist of retirement accounts, insurance policies, and any other monies you might have. Unless you plan to sell your house, don't count the equity in your house. After all, you have to live somewhere. The benefit of paying off your house (if you have one) is that you have no rent expenses.

In addition to salary, calculating what you make should include any rental properties, interest, and capital gains. This information is easily found on your yearly 1040 tax form. In my experience, people generally know how much money they have coming in.

The most challenging number you need to come up with is your yearly expenses. This can be difficult as most of us aren't accustomed to thinking in terms of annualized costs. For example, how much do you spend on clothes each year? How about house maintenance? Also, don't forget the interest you pay on credit cards or your mortgage.

The most challenging number you need to come up with is your yearly expenses.

The best way to determine your yearly expenses is to look through your checkbook. I like to put things in categories so I know where the money is going. However, simply knowing how much you spend is good enough for a first cut.

To make my life easier, I run all my expenses and ATM withdrawals through the same checking account. This makes collecting expense data much easier. If your situation is more complicated, you may find that you have to paw through several accounts to gather all this information.

Things that won't show up in your checkbook include car depreciation, taxes withheld from your paycheck, and expenses paid in cash. Car depreciation represents the amount of money you will have to spend when you replace your car. If you pay cash or take out a loan, you want to be sure and account for this future outlay of cash somehow. In order to fully understand the income and outgo of your financial system, you need to add-in taxes you pay. You will probably be shocked at how much it is! Also, be sure to capture how much cash you go through. I add up all my ATM withdrawals as a proxy for the amount of cash I spend.

The first time you try to determine these numbers may be a real challenge. However, keeping track of it after that is generally a breeze, once you know where everything is. Your reaction to even thinking about doing this might be one of shock and horror. But I urge you to stick with it. The information is invaluable to securing your financial future.

Constructing a Plan

After assembling a retirement scenario and calculating how much you have, make, and spend, you can determine where you stand. You may be surprised to discover that what you have been doing is adequate to meet your retirement goal. If not, you can find out

Good financial planners are like a good dentists: they may occasionally inflict pain, but in the end you will be better off for following their advice!

what you have to do to make your plans work. In some cases, the numbers may tell you that your dreams are a bit unrealistic and you need to scale them back or put them off a few years.

Even if you do nothing more than assemble this data, you will get a clear view of your current financial position. Knowing how much you spend in relation to how much you make is an important step toward achieving your life goals, whatever they are.

Creating a retirement plan involves combining your current resources with your income and expenses to create an *investment* plan that will deliver the amount of money you need to have when you want to retire. For most people, this will require the help of a professional financial planner and the will power to stick to the plan.

Also, retirement is not the only component of a financial plan. You also need to think about funding your children's education, making arrangements for your estate, and insuring or otherwise taking care of other problems that might arise in your special situation. A finance professional should know what is best for you after reviewing your specific particulars.

Aren't All Money People Slimy?

There are many kinds of financial planners: stockbrokers, insurance hawkers, annuity pushers, and assorted other creatures in the money jungle. Since they tend to get paid when you make investments they recommend, there is plenty of room for double-dealing and hidden self-interest.

Here are some basic guidelines to follow when choosing a financial advisor.

First, your finance person has to get paid. Find out how they get paid and make sure that you are comfortable with that. Ask questions about how the advisor gets paid. Don't even think about doing business with people who are vague about where their money comes from. Is what they recommend limited by what they make a fee on? What other charges might there be on your account?

Not all financial advisors are created equal. In theory, they all have passed the same tests given by the state where they are practicing. However, they all have their own biases and motivations. It is unlikely that choosing the first one you find (or who finds you) is the best approach. I suggest getting recommendations from people you trust, asking for referrals, and looking around. Avoid anyone who makes investments suggestions without first evaluating all the data discussed above. No one can advise you on what to do until they fully understand what you are trying to accomplish.

When you begin talking to people about finances, you will get a lot of unsolicited advice. People will brag about their financial wizardry, give you some hot stock tip, or provide other gems of financial hokum. Just nod your head and move on. Remember that you want to construct a future based on your unique needs. Any advice you get that doesn't take that into account should be ignored.

Lastly, not all financial advisors are greedy scumbags looking to get their hands into your pockets. Good financial planners are like good dentists: they may occasionally inflict pain, but in the end you will be better off for following their advice!

The most important point is this: it is up to you to create your own financial future; no one else will do it for you. Having and following a clear financial plan takes a lot of the uncertainty out of the future and gets you much closer to doing what you want when you want.

the bookworm

BOOKS REVIEWED IN THIS COLUMN

DOSSIER
RICH MORIN, ED.
www.ptf.com/dossier/

LINUX STANDARD BASE SPECIFICATION 1.0.0
www.opencontent.org/openpub/

BROADBAND INTERNET CONNECTIONS
RODERICK W. SMITH
Boston, MA: Addison-Wesley, 2002. Pp. 616. ISBN 0-201-73827-9.

CISCO IP ROUTING
ALEX ZININ
Boston, MA: Addison-Wesley, 2002. Pp. 635. ISBN 0-201-60473-6.

THE COMPUTER REVOLUTION IN CANADA
JOHN N. VARDALAS
Cambridge, MA: MIT Press, 2001. Pp. 409. ISBN 0-262-22064-4.

MECHANIZING PROOF
DONALD MACKENZIE
Cambridge, MA: MIT Press, 2001. ISBN 0-262-13393-8.

THE BOOK OF ZOPE
BEEHIVE ELECTRONIC MEDIA
San Francisco, CA: No Starch Press, 2001. Pp. 408. ISBN 1-886411-57-3.

LEARNING PERL, 3RD ED.
RANDAL L. SCHWARTZ AND TOM PHOENIX
Sebastopol, CA: O'Reilly & Associates, 2001. Pp. 316. ISBN 0-596-00132-0.

by Peter H. Salus

Peter H. Salus is a member of the ACM, the Early English Text Society, and the Trollope Society, and is a life member of the American Oriental Society. He is Editorial Director at Matrix.net. He owns neither a dog nor a cat.

peter@matrix.net

Too many books. Too many books.

Useful books. Interesting books.

But . . .

If you're reading this and don't have a copy of *The Root of All Evil* (O'Reilly's third User Friendly volume), stop reading and buy it. It will shed new light on your colleagues.

Docs

If you're like me, the first thing you do when you don't know/understand something is ask the person in the next office (or cubicle) or you phone someone you think might know. The next thing you do is look at the man pages. I keep my UNIX manuals right by my desk because I prefer paper to the screen. But there's a ton of stuff out there.

Rich Morin has begun a wonderful series, called DOSSIER (Documenting Open Source Software for Industry, Education, and Research). Rich has collected documents, placed them into a straightforward taxonomy, and printed them in handy 400-page volumes. I spent several hours reading in the *Email: Mail and Sendmail* volume. (There's also an *Exim* volume.) It's just great! There are *File Systems*, *Kernel*, and *Text* volumes, three *PostgreSQL* volumes, and two *Python* volumes. They run just over $40 each, including postage. A bargain.

Collect 'em all from: *www.ptf.com/ptf/dossier/*

Great idea, Rich.

A Note on Standards

Back in 1987, USENIX hosted a POSIX workshop at the Berkeley marina. I've been reading standards docs since then. So I wasn't startled when the Free Standards Group gave me a hardcopy of the *Linux Standard Base Specification 1.0.0* at the ALS in November. This is a brave attempt at defining a system interface for compiled applications. As we have seen the UNIX world fragment and subsequently attempt standardization, perhaps efforts like this can prevent something similar happening to Linux.

Internet Stuff

Roderick W. Smith has taken up a much tougher job: explaining broadband connections. I was taken by his early chapters and thought that I might try to actually connect a box at home following his instructions. I failed. The box I wanted to connect was a SPARC Ultra 5. Smith provides "detail for Windows, MacOS, and Linux." I was out of luck. So, if you're running FreeBSD, OpenBSD, NetBSD, BSDi, or Solaris, this book will not be your cup of tea. It's a nice book. B, but it wasn't helpful to me.

In fact, apropos of this, I'm really tired of reading stuff that assumes that I've been grafted to an Intel chip. While I recognize that much of the world creeps along Wintel Way, I recall only too well the various instruction set/RISC arguments (and a really good paper by Dan Klein). Hey, remember the Z80?

Alex Zinin's fat volume is not for the fainthearted. It's a first-rate, thorough work on the "packet forwarding and intra-domain routing protocols." Zinin actually describes just what's going on inside the router. It's not an easy read, but it is worth the time you'll put into it.

A Version of History

John N. Vardalas has produced a fascinating narrative of the development of computer technology in Canada from

book reviews

1945 through 1980. Unfortunately, it is a partial one. Vardalas has chosen to center his analysis on a few very large companies and on the military. The result is that he misses some of the more significant events, largely because they were either academic (or semi-academic) or because they concern software, not hardware.

Though the Canadian contributions to mail sorting and the airlines reservations system are mentioned, NewsWhole, the first real page-layout system (at the Toronto *Globe & Mail*), designed by David Tilbrook, has been missed. In fact, the incredible contributions of Ron Baecker and his students (Mike Tilson, Tom Duff, Rob Pike, etc.) are absent. HCR is missing. And some other notable Canadian software folk find no place here: Morwen Gentleman, Brian Kernighan, Heinz Lycklama, to pick a few. And while there are mentions of Toronto and British Columbia, I sought in vain for Waterloo or McMaster.

Ferranti of Canada, Sperry Gyroscope of Canada, and Control Data of Canada were important. But hardware gets overemphasized here. And the ARPANET/Internet doesn't even get an entry in the index, despite the fact that Atomic Energy of Canada in Chalk River, ON, was the first ARPANET site outside the US.

I learned a lot from Vardalas; I expect the expanded, second edition (in five years?) will be even better.

Prove It!

Donald MacKenzie has written a dense, difficult book on a fascinating topic: the history of proof, applied to what's done by traditional mathematicians and to what's required of formal, mechanized (computational) proof. This is a history of the interaction of mathematics and computation as viewed by a sociologist. MacKenzie's examination of the social influences on the development of automated proof is superb. He concludes that in pursuing dependable computer systems we don't eclipse the need for trust in human judgment. Tough stuff, but worthwhile.

Web Servers

Zope is the leading open source Web application server. Beehive Electronic Media is a German company that does Zope training and development. *The Book of Zope* is a straightforward introduction that appears to cover everything. Zope appears to be written, for the most part, in Python.

And a Final Perl

It's eight years since I got my first llama. Since then the book has become larger and better. Everything that Randal and Tom have learned in teaching Perl over the years seems to have gone into this new edition. If you don't have it, you need it. If you've been relying on the first or second editions, this one is far better.

WEB CACHING

DUANE WESSELS

Sebastopol, CA: O'Reilly & Associates, 2001. Pp. 318. ISBN: 1-56592-536-X

Reviewed by Alex Rousskov

The technical books I've read are usually of low quality. The factual material is buggy, analysis is shallow, and entertainment value is expectedly low. Most technical authors are unaccustomed to writing books and are too busy with their day-to-day routine to put much time into a book-length project. Most good editors are overloaded. Most publishers have to think of the bottom line first. Knuth-quality masterpieces are rare.

Duane Wessels' *Web Caching* would need thorough editing to meet my stringent quality requirements. However, as the first and only book devoted to an important subject, it is worth your consideration.

If your responsibilities or interests are related to the Internet and Web traffic, you must know about Web caching. Duane is a Web caching guru. He is the original and ongoing author of Squid, an open source free caching software that rivals commercial offerings in market share and features. Not familiar with Squid? Think Apache's httpd in the origin servers category. Duane also runs an international caching hierarchy that has been used by numerous researchers and practitioners worldwide to study and improve the Internet. Duane's name is on the Internet Cache Protocol (ICP), RFCs, and several landmark papers on Web caching. Duane has helped to make a series of international Web caching workshops a success. As an early industry player, Duane knows Web caching inside and out. He certainly has the expertise to write a book on Web caching. If only he could disable his email like Donald Knuth . . .

Web Caching starts with a two-chapter introduction to the topic that is meant to bring folks unfamiliar with HTTP and caching up-to-speed. Newbies may have a difficult time swallowing the dry cocktail of technical details, HTTP headers, and acronyms. In trying to build a foundation of knowledge for the rest of the book, the author often dives into unnecessary detail. If this is your first encounter with HTTP, brace yourself.

Chapter 3 talks about legal and ethical issues of Web caching and the politics that surrounds those matters. This is a rare case when politics belongs in a technical book. This may also be the only chapter that has a lot of original material not available in digestable form elsewhere on the Web. If you think caching is strictly a problem of placing content closer to the user, this chapter will open your eyes. If you are a cache maintainer

book reviews

advising company lawyers on technical matters or a lawyer advising technical folks, I strongly recommend reading this chapter before your users or content providers come after you or your company. Even caching experts are likely to find the discussion interesting and useful. The weak side of this chapter is mostly US-specific content and absence of clear-cut conclusions. Your company will still need a lawyer.

The next seven chapters talk mostly about configuring Web clients, Web caches, cache hierarchies, and origin servers to work with caches and/or to cache content efficiently. Generally useful, good technical stuff, though some topics would benefit from a more thorough treatment.. Many complex issues such as traffic interception or cache hierarchies are discussed with pros and cons of specific solutions compared. If you are deploying a cache on your network, you will find answers to many of your questions in these chapters. Some configuration details are likely to become outdated soon, but the book does a good job discussing general concepts and common pitfalls that are likely to remain relevant for at least a few years. Duane's open source software bias is especially evident in these chapters.

Chapter 11 is devoted to monitoring cache operation and would have been extremely useful to cache operators had it not been only nine pages long. I don't know why the author decided to hide his expertise and experience behind a few URLs pointing to available monitoring tools. The book would significantly benefit from more specific examples; perhaps a large-scale Web caching bibliography. The index is surprisingly incomplete considering currently available indexing tools.

Web Caching is definitely missing a chapter on content delivery networks (CDNs). The author claims that CDNs lack proven results. Perhaps total unavailability of all major news sites that were not using CDNs following the terrorist attacks in the US will convince him otherwise.

Web Caching's strongest points are breadth of coverage and fearless attempts to discuss controversial and even nontechnical issues. In the absence of competition, this is the best reference book on caching today. To stay on top, however, O'Reilly needs to do significantly better editing for the second edition as well as prompt Duane to provide in-depth treatment of key subjects.

Disclaimer: my objectivity in this review was in no significant way affected by the fact that Duane and I have been working together for the past three years and own the same company.

BUILDING SECURE SOFTWARE: HOW TO AVOID SECURITY PROBLEMS THE RIGHT WAY

GARY MCGRAW AND JOHN VIEGA

Boston, MA: Addison-Wesley Professional Computing Series, 2001.Pp. 528.
ISBN: 0-201-72152-X

Reviewed by Ray Schneider
ray@hackfoo.net

In the world of interconnected computers, whether on LANs or WANs, it has never been more important than now to have secure systems. The authors believe that it all starts with software source code; without source code written with security in mind, none of the rest matters.

Viega and McGraw break *Building Secure Software* down into two logical sections. The first section, consisting of several chapters, successfully introduces the reader to security by covering technologies, goals, as well as risk management.

Readers who make decisions about security in the development of software will find the first chapters enlightening.

There are many references to online sources of security information. The authors mention mailing lists such as BUGTRAQ and the RISKS Digest. They also cover things like "Penetrate and Patch" and the "Art of Engineering." The authors introduce the reader to ideas such as "The Common Criteria," which is an ISO standard that has its roots in the DOD and the Orange Book.

The authors continue initiating the reader into security by examining a few basic ideas: keep it simple, be reluctant to trust, and fail securely. Viega and McGraw also discuss such popular beliefs as the now infamous Eric Raymond quote, "given enough eyeballs, all bugs are shallow." The reader is also introduced to issues surrounding full disclosure and open source. This section of the book looks at these ideas with a critical eye, noting the potential problems that arise from following the popular mojo without considering the effects.

The later chapters of *Building Secure Software* get down to the nitty-gritty of it: source code. Readers will explore buffer and heap overflows, race conditions, cryptography, random number generation, and entropy. These topics are covered in detail. Source code is developed, examined, and improved. Viega and McGraw even supply working exploit code and demonstrate how it works. The majority of the source in the book is in C, but there are examples in Java and Perl as well.

As Bruce Schneier says in his foreword, "*Building Secure Software* is a critical tool in the understanding of secure software." I highly recommend this book to anyone responsible for the development of software in the sometimes hostile environment of interconnected systems.

SAGE news

From the SAGE President

by David Parter

President, SAGE STG Executive Committee

parter@sage.org

Welcome to 2002, and the first of a series of presidential messages to SAGE members.

My first topic is, obviously, SAGE, and the state of SAGE, and I would like to offer some explanation of some recent events affecting SAGE's governance. More up-to-date information should be available on the SAGE Web site – (*http://www.usenix.org/sage/*) – please look for it there.

I have learned from recent developments that the large disparity in speed between electronic and non-electronic media poses a challenge in trying to communicate news and timely information to all members.

By the time you read this, it may be old news for some of you. In mid-November, as part of the every-few-years "viability review" of SAGE, the USENIX Board of Directors took several specific actions with regards to SAGE. The STG Viability Review is on the Web at *http://www.usenix.org/sage/restructuring* (and also on the following page of this issue of *;login:*). As a result of the viability review, Peg Shafer and Barb Dijker graciously resigned from the SAGE Executive Committee, as requested by the USENIX Board. The Executive Committee accepted their resignations with regret and deep respect for the time, effort, passion, and commitment to SAGE that they have both shown over the years.

At our meeting at the start of the LISA conference in December, the Executive Committee unanimously adopted the following response to the viability review:

The SAGE Executive Committee considers that the process used in conducting the review was flawed.

Further, we believe that the removal of two elected Committee members by the USENIX Board of Directors was heavy-handed and damaging.

We do appreciate the constructive action that the USENIX Board has taken in providing additional resources for carrying out the SAGE mission.

We remain committed to working constructively with the USENIX Board of Directors and the SAGE community.

We also adopted a plan to appoint two members to fill the vacancies on the Executive Committee. This process will allow the Executive Committee to move forward in a thoughtful and timely manner. The full Executive Committee will have an in-person meeting in mid-February. More information about the appointments and the meeting can be found on the SAGE "What's New" page (*http://www.usenix.org/sage/whatsnew/whatsnew.html*).

We are working with the USENIX Board to correct the procedural and communication problems which led to this situation. In addition the Executive Committee is committed to improved communication with the membership. One aspect of that will be a regular column in *;login:* by the SAGE President, and more frequent contributions to *;login:* from other members of the Executive Committee. Other steps include more frequent "memos to members" with timely news and information, and making Executive Committee meeting minutes available (on the Web) to members.

We are also working with the staff to determine how best to utilize the additional staff resources we have been given, and how to best work with the staff and other volunteers to carry out SAGE programs.

An Executive Committee focus on effective delivery of our programs is only part of the picture. We also need your involvement. You can contribute in many ways – volunteer for one of our programs, contribute to *;login:*, submit a paper to LISA, or help with a SAGE local group. Please also contact us with ideas, comments, and suggestions.

SAGE STG Viability Review

USENIX Association policies regarding its Special Technical Groups (STGs) require a viability review for each STG every five years. The second viability review for SAGE was conducted on November 6, 2001. The attendees were Andrew Hume and Mike Jones (USENIX STG Committee minus Dan Geer, who was absent), Dave Parter and Trey Harris (SAGE Exec president and Secretary), and Ellie Young, Gale Berkowitz, and Jane-Ellen Long (USENIX Staff). Parter opened with a statement and a question and answer session followed. Parter and Harris were then thanked and excused, and recommendations were prepared for the USENIX Board.

The findings were

1. SAGE membership is active and growing in numbers.
2. SAGE has a number of achievements, including a certification

project, and a strategic plan for SAGE.

3. There is profound concern by the USENIX Board over the effectiveness of the current SAGE governance, a concern shared by both the SAGE Exec and the USENIX staff.
4. The SAGE Exec believes that SAGE programs would be better carried out by staff members than volunteers; the USENIX Board and Staff concur.
5. There has been an extraordinary amount of work expended around the issue of splitting SAGE from USENIX with very little to show for it.

The USENIX Board meeting was held on the next day, and during an Executive session, considered carefully the above findings and unaniminously adopted the following points:

a. the USENIX Board strongly believes in SAGE as an organization, in its programs, and in SAGE as the premier professional organization for systems administrators.
b. the USENIX Board agrees with the SAGE Exec that SAGE programs are best carried out by staff members rather than volunteers, and therefore explicitly changes the role of the SAGE Exec from both conceiving and carrying out SAGE programs to conceiving and overseeing SAGE programs as implemented by paid staff.
c. in light of b., the USENIX Board will fund the following additional staff resources for SAGE:

- A fulltime SAGE Projects Director position (to be filled by Gale Berkowitz)
- one new full-time position, devoted to SAGE-specific activities, e.g., mentoring activities, local group activities, Web content editing.

d. the USENIX Board has a lack of confidence in the effectiveness of the current SAGE governance. While significant time and energy has been spent considering restructuring alternatives, related action items intended to allow all parties to assess the viability of these alternatives have been repeatedly accepted by SAGE Exec members but not carried out.
e. in light of d., the USENIX Board has considered the SAGE Exec's motion of intent to split off from USENIX (dated 11/4/2001) and rejects it. To discourage wasting more time and energy on this issue, the USENIX Board will not consider any proposal to split SAGE from USENIX prior to 11/1/2003.
f. in the course of considering various restructuring plans over the last 18 months, the USENIX Board has repeatedly asked for a business plan covering splitting SAGE off from USENIX. Despite repeated assurances that it would be produced, this business plan still does not exist. Accordingly, the USENIX Board has requested the resignations of Barb Dijker and Peg Schafer, who held responsibility for these action items, as per Article 10.5.5 of the USENIX STG policy document.

These actions were unanimously adopted by the USENIX Board in the hope and judgment that they best use both SAGE Exec and USENIX resources in accomplishing SAGE's programs.

SAGE Certification Update

We are pleased to report that the first-level SAGE certification is nearly ready for release. Here is some important information about the SAGE Certification Project.

What is cSAGE?

cSAGE is the first level of certification offered by SAGE Certification. The program is geared for junior-level system administrators (as defined by SAGE).

cSAGE certification will first require a candidate to agree to a basic statement of a candidate's applicable system administration experience prior to taking the exam. cSAGE testing will then consist of two exams, a core plus a module, which should be completed in the same sitting. The core exam is neither platform- nor vendor-specific; the modules, however, are platform-specific exams (e.g., UNIX). cSAGE certification testing lasts approximately four hours.

Future certifications will be developed that include higher levels of testing. Descriptions of those levels will be announced by the time we go live in March 2002.

When will cSAGE be "live"?

cSAGE will be offered as a complete certification program on March 26, 2002, through VUE testing centers worldwide.

What is available before March 26, 2002?

Before that time, there are two opportunities to participate in the development process:

The core beta exam took place at the LISA conference in December. While this core beta was very successful, we are looking for more junior-level individu-

als to take the core exam to complete the demographic spread needed in the analysis process. Therefore, another session of this beta may be made available through VUE testing in late January. Please check the Web site for more information as it becomes available.

The UNIX beta exam is available through VUE testing worldwide from Monday, January 28, through Wednesday, February 13, 2002.

Both the core and UNIX beta exams WILL count towards the cSAGE certification. The betas include more questions than the actual live exams, and the results will not be available right away. We are using the betas to eliminate less appropriate questions from the exams, so scoring cannot take place until the proper item analysis is complete.

Who will benefit from SAGE certifications?

The SAGE Certification program is useful to individuals and corporations who wish to evaluate and promote system administration in real-life work situations. The program is well suited for:

- Junior-level system administrators seeking verifiable validation of their abilities
- Recent college graduates strengthening their marketability
- Channel partners, resellers, and authorized service provider programs seeking confirmation of their remote site capabilities
- Corporations who wish to assist their system administrators in furthering their career development

What is recommended for preparation for the cSAGE exam?

Current information about study materials can be located at *http://www.usenix.org/sage/cert/study.html*. This information will be expanded as we go live, so please check back monthly.

Who is involved in the certification development process?

The primary vehicle for the SAGE Certification Project is the Certification Interim Board, formerly known as the Policy Committee. The Board also makes appropriate use of professional certification consultants. SAGE has conducted an analysis of skill requirements and has evaluated testing methodologies and implementation logistics. To accomplish this in an objective manner, SAGE engaged the services of Human Resources Research Organization (HumRRO) and Galton Technologies to develop the exams using proven methodologies.

Where can I find out more about SAGE Certification?

The SAGE Certification Web site is located at *http://www.sagecert.org*. The site includes the cSAGE Candidate Handbook, which is particularly useful for obtaining more information about the program.

For further information, or if you wish to volunteer your time, please contact:

Stacy Gildenston at *stacy@sage.org.*

GUUG/SAGE Group Founded

On September 22 a new SAGE group was founded in Germany. The older German UNIX Users Group (GUUG, *http://www.guug.de*), members of which attended the meeting, will form the framework for the new organization to be named GUUG/SAGE.

The primary contacts for this new German SAGE group are Wolfgang Sachs (*mailto:sachs@atkins.swb.de*) and Martin Schulte (*mailto:schulte@guug.de*). Other volunteers, such as Jochen Topf, who set up the *http://sage.guug.de* Web site, are contributing to the new mailing list (*sage@guug.de*). A basic element of this initiative are regular meeting called Stammtische. Please consult our Web pages at *http://sage.guug.de* for details and schedules of meetings. We plan to offer links to tools useful to sysadmins, and to discuss them at meetings. We also want to have localizes "best practices" references.

The new group is, of course, in need of help from volunteers willing to help and share their knowledge. As we plan to have a SAGE track at GUUG conferences we particularly need speakers and people willing to represent us there. If you can help, visit our Web site or contact Wolfgang or Martin directly.

A Selection of Papers from LISA and Computing Systems Published

Selected Papers in Network and System Administration, a volume edited by Eric Anderson, Mark Burgess, and Alva Couch, has just been published by Wiley, in cooperation with USENIX and SAGE. The volume includes key contributions to the discipline of Network and System Administration originally presented at LISA and in *Computing Systems*. The papers are accompanied by a commentary reflecting on their larger significance.

USENIX and SAGE purchased a number of copies to be made available to its members. If you were lucky enough to attend LISA 2001 in San Diego, you were

given the opportunity to receive the volume for free. But members can order the volume from USENIX for $30 per copy, a considerable discount from the $55.95 list price. See *http://www.usenix.org/publications/ordering/*

2001 SAGE Outstanding Achievement Award

The 2001 SAGE Outstanding Achievement Award was presented to Hal Pomeranz at this year's LISA Conference in San Diego. This annual award goes to someone whose professional contributions to the system administration community over a number of years merit special recognition.

Hal Pomeranz was selected for his exemplary contributions as an educator of system administrators, through works such as the Perl Practicum series, and for his years of leadership in the system administration community.

Congratulations, Hal!

New SAGE Executive Committee Members

by Trey Harris
Secretary, SAGE Excecutive Committee
trey@sage.org

On January 8, 2002, Strata Chalup tendered her resignation from the SAGE Executive Committee, citing time commitments that she felt would keep her from fulfilling her responsibilities. The Executive Committee accepted her resignation with thanks for her outstanding service to SAGE.

The SAGE Executive Committee has appointed three new Executives to fill the vacancies on the Committee. They are:

Bryan C. Andregg, *andregg@sage.org*
Gabriel Krabbe, *gabe@sage.org*
Josh Simon, *jss@sage.org*

We wish to extend our congratulations to them and also our thanks to all the nominees for participating in this process. The next meeting of the SAGE Executive Committee will be February 23-24 in Monterey, CA.

The SAGE Executive Committee is now comprised of:

David Parter, President,
parter@sage.org
Geoff Halprin, Vice President
geoff@sage.org
Trey Harris, Secretary
trey@sage.org
Tim Gassaway, Executive
gassaway@sage.org
Bryan C. Andregg, Executive
andregg@sage.org
Gabriel Krabbe, Executive
gabe@sage.org
Josh Simon, Executive
jss@sage.org

USENIX news

If You Read Nothing Else, Read This

by Daniel Geer

President, USENIX Board of Directors

geer@usenix.org

A democracy, as Winston Churchill so eloquently noted, is the worst of all systems except for all the others. USENIX is a democracy, and in this *;login:* you will see that the USENIX election process is upon us. If you read nothing else, read this.

To be clear, I am not running. I would like to, but USENIX has term limits and therefore my role as Board Member and Officer comes to an end. Whether I have more to offer or not is irrelevant. Therefore I have at once the freedom and the duty to suggest what makes good leaders for USENIX, and what does not.

USENIX is "The Advanced Computing Systems Association." It has a purpose that is a given, namely

- problem-solving with a practical bias,
- fostering innovation and research that works,
- communicating rapidly the results of both research and innovation,
- providing a neutral forum for the exercise of critical thought and the airing of technical issues.

The Board of Directors is elected by the membership to discharge this responsibility. In a word, the above is the job description. Anyone who runs for office had better be prepared to make those goals their only goals. If they have other goals, they are running against the organization which, though a democratic right, is between a nuisance and a disgrace even as running for personal resumé enhancement is beneath contempt. We, the USENIX membership, are lucky to have almost entirely escaped the negatives of democracy, i.e., we have almost always elected Board members who endorsed the purpose of the Association and who come to the Board filtered by a proven history of service to the Association. If numbers are how you think, consider that Citeseer (*http://citeseer.nj.nec.com/impact.html*) recently started ranking publication venues by impact (citation rate), and 3 of the top 4 spots, out of 858, belong to USENIX conferences: OSDI #1, USITS #3, USENIX Annual #4. This is the standard every successive Board has to uphold if not improve upon.

When I say that we have largely elected Board members "filtered by a proven history of service to the Association" I mean something that is very, very important. At any given moment, half of the USENIX membership is new and therefore naturally has limited scope with which to assess that "proven history" factor. At any given moment, there is almost always a negative correlation between mere name-recognition popularity and the "proven history" factor since, as we all know, you get a lot more accomplished when you take no care as to who gets the credit. This is why USENIX, like every durable organization for which institutional memory and leadership capacity are closely and eternally correlated, entrusts itself so fully to a formal Nominating Committee. There is no more thankless yet essential task than that of a nominating committee; there are no more easily bruised egos than those who think they are *deserving*

of office rather than humbled by the *prospect* of office, and there is no more easy target for interference by the insolently self-satisfied than to undermine the work of a nominating committee. And, yes, as with democracy, nominating committees are the worst of all systems except for all the others.

I write to denounce the efforts of Greg Rose to pre-empt the governance of an organization from which, in high dudgeon, he peremptorily walked away. I write to thank, beyond words, the work of the USENIX Nominating Committee whose members deserve our reverence, our gratitude, and our trust. I write USENIX members to urge in the strongest way that you make your selections from amongst those who played the game on the field rather than behind the grandstand and who are, by my inspection and their affirmation, committed to the USENIX Association and to its purpose as laid out above. I write to remind us all that there is a vast gulf between a leader and a provocateur, and that as the Good Book says "By their works shall ye know them." It is by the works of the Nominating Committee that they are known. Even more, it is by their works that those nominated for office are known, and rightly. Any choice amongst these fine people below is a choice consistent with honor and with earned appreciation; you have my word on it.

for Board President: Kirk McKusick
for Board Vice President: Mike Jones
for Board Treasurer: Lois Bennett
for Board Secretary: Peter Honeyman
for Board At Large: Clem Cole
for Board At Large: Tina Darmohray
for Board At Large: John Gilmore
for Board At Large: Darrell Long
for Board At Large: Adam Moskowitz
for Board At Large: Avi Rubin
for Board At Large: Ted Ts'o

Report of the Nominating Committee for the Election of the USENIX Board

The Nominating Committee for the 2002 USENIX Board Election is Eric Allman, Andrew Hume (Chair), John Kohl, and Rob Kolstad. The purpose of the Nominating Committee is to ensure a slate of qualified candidates for the USENIX Board election. This report simply describes how we chose the slate of nominees and the list of nominees itself. Unlike previous years, we did not perform a detailed evaluation of the current board, nor are we providing any commentary on the nominees themselves.

Choosing the nominees is a tension between several, sometimes conflicting, guidelines. These include continuity between old and new boards, the need for the treasurer to be experienced in financial matters, and for the nominees' backgrounds to span several key constituencies within USENIX; for 2002, we took these to be researchers, academics, system administrators, and the fields of security and Freenix.

The nominees are

President: Marshall Kirk McKusick, Author and Consultant

Vice-President: Mike Jones, Microsoft Research

Treasurer: Lois Bennett, Harvard University

Secretary: Peter Honeyman, CITI, University of Michigan

Directors (4 positions):
Clem Cole, Paceline Systems
Tina Darmohray, Stanford University
John Gilmore
Darrell Long, Univ. of California, Santa Cruz
Adam Moskowitz, Menlo Computing
Avi Rubin, AT&T Labs - Research
Ted Ts'o, IBM

Vote for the 2002 Election for Board of Directors!

The biennial election for officers and directors of the Association is being held right now.

Ballots have been sent to all paid-up members on or about February 19. Members have until April 1st to cast their votes. The results of the election will be announced in *comp.org.usenix*, the USENIX Web site, and the June issue of *;login:*.

The Board is made up of eight directors, four of whom are "at large." The others are the President, Vice President, Secretary, and Treasurer. The balloting is preferential; those candidates with the largest number of votes are elected. Ties in elections for directors shall result in run-off elections, the results of which shall be determined by a majority of the votes cast. Newly elected directors will take office at the conclusion of the first regularly scheduled meeting following the election, or on July 1st, whichever is earlier.

Summary of the USENIX Board of Directors Actions

by Gale Berkowitz and Ellie Young

The following is a summary of some of the actions taken by the USENIX Board of Directors between September and December 2001.

ALS Conference Registration Fees

It was decided to eliminate the registration fees to the technical sessions at the Annual Linux Showcase which was held in November 2001.

Anti-Terrorism Act (ATA)

The Board agreed to alert USENIX members of its concern regarding provisions in the proposed Anti-Terrorism Act (ATA) that would treat all computer trespasses as terrorism, and was under consideration in Congress . A subcommittee was formed to draft a statement that was published on the Web site and sent to the membership.

SAGE Certification Effort

The Board adopted the resolution creating an Interim Certification Board and governance structure for the SAGE certification program. It also guaranteed up to $200,000 per year in funding, subject to review, for the SAGE Certification Program over the next two years.

SAGE Staffing

It was agreed to increase the number of staff dedicated to SAGE to include one full-time SAGE projects director and another position of an online system administration/content editor for the SAGE website.

Registration Fees and Member Dues

The Board voted to raise the conference registration fees beginning with the June 2002 conferences as follows:

Tutorials:

- One-Day tutorial: $600
- 2-Day tutorial: $1100
- 3-Day tutorial: $1500.

Technical sessions:

- Annual Technical Conference and LISA: $595
- 3-Day events: $645
- 2-Day events: : $600
- Student fees: $100

Member Dues

A modest raise in dues was approved to cover increased costs, as follows:

- Individual: $100
- Student: $30
- Supporting: $2500
- Educational: $220
- Corporate: $430

Distance Learning Program

The sum of $15,000 was budgeted in order for USENIX to offer Digitalthink courseware as a member benefit. A proposal was requested from the staff to fund four Web tutorial pilots in early 2002. Suggestions regarding alternatives for using non-proprietary distance learning software were discussed and will be explored.

Standards Activities

A proposal in the amount of $14,370 to continue standards work primarily in the area of Linux Standards Base Project was approved.

USACO

It was agreed to fund once again the USA Computing Olympiad team in 2002 in the amount of $29,300.

NordU Conference

A loan of $25,000 to EurOpen.SE for the support of the NordU 2002 conference was approved.

STG Viability Review

The report from the STG Review committee indicated that viability of SAGE was reaffirmed. There were, however, concerns about the effectiveness of the governance. See page 65 and *http://www.usenix.org/sage/restructuring/viabilityreview.html*

Nominating Committee

Andrew Hume was appointed chair of the Nominating Committee for the USENIX Board Elections in '02. The rest of the committee is Eric Allman, Rob Kolstad, and John Kohl. The election will be conducted by paper ballots due April 1st.

Linux International

USENIX will become an Affiliate member of Linux International.

Internet Measurement Workshop

It was agreed that USENIX will co-sponsor this workshop next year.

Next Meeting

The next meeting of the USENIX Board of Directors is scheduled for Friday, February 15, 2002, in conjunction with BSDCon in San Francisco, CA.

Fifteen Years Ago in USENIX

by Peter H. Salus
USENIX Historian
peter@matrix.net

In January 1982, Mike O'Brien (now the amanuensis of Mr. Protocol) hosted an annual meeting in Santa Monica. In January 1987, Rick Adams, then of the Center for Seismic Studies, was the host in Washington, DC.

USENIX met in DC in 1984 in January. The location was actually determined by /usr/group. The meeting was memorable for the weather. In 1987, Washington maintained its standing: we had the "second DC snowstorm," beginning around 11 a.m. on Thursday, January 22.

Thursday evening, with flights cancelled and people unexpectedly sharing accommodation, was true chaos. Friday was somewhat better, and I flew back to California on Saturday without a hitch.

But Wednesday was a full day of sessions to remember. Dave Tilbrook (now at RIM) and Debbie Scherrer (now at Transmeta) had set up seven speakers to discuss "What it is to be UNIX": John Mashey, Eric Allman, Steve Johnson, John R. Mullen, Peter Collinson, Dennis Ritchie, and Mike O'Dell.

Allman spoke about the data forms and Johnson about the language forms. Collinson spoke about UNIX as a cult. O'Dell gave us a "world view." Mashey began with "Leverage – Past, Present, Future." Ritchie gave us his view of "why the UNIX system has succeeded."[NOTE WHAT ABOUT MULLEN?]

It was a star-studded day. The presentations were really interesting. And they gave us something to talk about besides the snow.

Profile on Good Works

In keeping with its commitment to promoting representation of women and underrepresented groups in the computing professions, USENIX contributed $10,000 in support of the Richard Tapia Celebration of Diversity in Computing Symposium.USENIX funding was used for scholarships for students to attend the event. [*See letters of thanks from some of those students on page 4*]

Coalition to Diversify Computing Launches Tapia Celebration of Diversity in Computing Series

by Valerie Taylor

Northwestern University, CDC Co-chair

The Coalition to Diversify Computing (CDC) held the inaugural Richard Tapia Celebration of Diversity in Computing Symposium (*http://www.sdsc.edu/Tapia2001/*) on October 18-20, 2001, in Houston, Texas. The symposium was sponsored by the Association of Computing Machinery (ACM). Designed to celebrate the technical contributions and career interests of diverse people in computing fields, the symposium offered a mix of technical and nontechnical talks and panels, a poster session, and a unique awards banquet. The symposium program was designed around the theme of "Expanding Horizons," reflecting a focus on access to powerful knowledge from diverse researchers in computing, expanding the community of people in the field of computing, and sharing knowledge between the different disciplines of computing.

The celebration honored Dr. Richard A. Tapia, a mathematician and professor in the Computational and Applied Mathematics Department at Rice University in Houston. Dr. Tapia is a member of the National Academy of Engineering, the first recipient of the A. Nico Habermann Award from the Computing Research Association, and a member of the National Science Board; he is equally well known for his commitment to educational equity, mentoring, and student success. As the CDC embarked upon naming the symposium, only one name came to mind to all members, simultaneously – Richard Tapia. The members of CDC discussed Richard Tapia's significant impact on their careers, either as a role model, mentor, colleague, or very dear friend. His impact was felt throughout the symposium, at which hundreds of people shared their impressions of Tapia's influence on their lives; newcomers were able to get to know Richard Tapia and absorb his great energy and enthusiasm about increasing diversity in computing.

Tapia's keynote address, "Diversifying Computing: Its Contradictions, Challenges, and Successes," was a candid assessment of current educational policy and reform initiatives in which he urged attendees to close the education gap, engage minority youth in the world of science, and hold high the bar of academic excellence. Jackie McNab of KDH Science gave an invited plenary talk entitled "Breaking through Barriers: A Journey to Success." McNab's talk, which was targeted to the student attendees, provided examples of questions to ask and standards to use to "think outside the box," aim high, and capitalize on one's unique talents. She provided proof of her methods with her own successes and challenges.

Judging by conference evaluations, the symposium was a great success. More than 96% of the participants found the symposium intellectually stimulating. Further, 76% of the participants felt the symposium increased their desire to conduct research in the areas of science, mathematics, engineering, or technology, and 61% felt motivated to conduct interdisciplinary research as a result of the symposium. The symposium had a profound and positive effect on those who were there.

As with the Tapia Symposium 2001, the Tapia conference in 2003 will make an effort to involve as many students as possible. Scholarships will again be provided, and students will be asked to contribute ideas for speakers and conference activities.

The Tapia Symposium 2001 was sponsored by ACM, with additional support from the National Science Foundation; the Alliances for Graduate Education and the Professorate at Rice University; Argonne National Laboratory; the Computing Research Association; the Education, Outreach, and Training Program, Partnership for Advanced Computational Infrastructure; Microsoft Corporation; NASA; Rice University; and USENIX.

The mission of the Coalition to Diversify Computing (CDC), which planned the event, is to increase the visibility of people of color in computing research and to provide networking opportunities for minority researchers, faculty, and students. CDC is a joint committee of the Association of Computing Machinery (ACM), the Computing Research Association (CRA), and the Institute of Electrical and Electronic Engineering (IEEE) Computer Society. For more information on the CDC, see *http://www.npaci.edu/Outreach/CDC*.

USENIX Launches Distance Learning

by Catherine Vegher
Marketing Manager
catherine@usenix.org

This past June, the USENIX Board authorized a three part Distance Learning pilot program to test the viability of offering conference tutorials over the Web, offering Web-based training courseware at a discount to our members, and potential affiliations with established University Distance Learning programs. We feel these programs could represent a significant benefit to USENIX and SAGE members and could become an additional revenue source for the organization. A brief outline of the project plan follows.

Part 1. Virtual Classroom: Live interactive tutorials delivered over the Web

Conducted last Fall, the first phase of the pilot program included four, one and one half hour LIVE tutorials. Each tutorial was limited to 35 participants and 'sold out' within two hours of sending an email announcement to the USENIX database. Participants in tutorials rated the content and delivery methodology very high.

Based on the overwhelmingly positive feedback, the Board authorized an additional set of pilot programs to test the concept with full-length tutorials. Beginning in February 2002, this second installment of courses will be delivered in hour and a half sessions over three successive weeks. The registration fee for each course will be $395. Complete course descriptions and registration information can be found at: *http://www.usenix.org/events/elearning/*

After the completion of the pilot programs, USENIX will determine if online tutorials should be added to the educational programs offered at USENIX.

Part 2. Web Based Training

Basic online interactive courseware in a wide range of technical areas such as Linux, Java, XML, Microsoft, Cisco, and Oracle.

We explored industry standard Web-based training, in order to make additional courseware available to SAGE and USENIX members at a discount.

A committee of experts from our community, including Aeleen Frisch, John Sechrest, Steve Simmons and our consultant Richard Jaross of Global Training Solutions, examined the Web-based training courses currently available. The committee established criteria for evaluation and conducted a review of the three leading providers: Smartforce, NETG and DigitalThink.

DigitalThink was the unanimous choice of the committee due to the quality of their courseware, the ability to run it on any browser, and the live tutorial support. USENIX and SAGE members will now receive a 20% discount off the list price of any DigitalThink course.

Part 3. University Distance Learning Program

The third part of the pilot will explore relationships with leading University distance learning programs. We will be looking to make these programs available to members at a discount price. We will also explore the opportunity to have members be a resource for University Programs in a teaching or research capacity.

Thanks to our Volunteers

Ellie Young
Executive Director

USENIX's success would not be possible without the volunteers who lend the expertise and support to our conferences, publications, member services, SAGE, new projects, and philanthropic activities. While there are many who serve on program committeees, coordinate the various activities at the conferences, work on committees and contribute to this magazine, I would like to make special mention of the following individuals who made signficiant contributions in 2001:

The program chairs for our 2001 conferences:

- Yi-Min Wang and Rajendra Raj, 6th Conference on Object-Oriented Technologies & Systems
- Tom Anderson, 3rd USENIX Symposium on Internet Technologies & Systems
- Saul Wold, Java VM Research & Technology Symposium
- Yoonho Park, 2001 USENIX Annual Technical Conference
- Dan S. Wallach, 10th USENIX Security Symposium
- Ted Ts'o for organizing and moderating the Linux 2.5 Kernel Developer's Summit
- Bryan C. Andregg, 5th Annual Linux Showcase & Conference
- Keith Packard, XFree86 Technical Conference
- Mark Burgess, 15th LISA Conference
- Jan Sael for chairing the NordU2001 Conference.

The conferences' Invited Talk/Special Track Chairs:

- Doug Schmidt, Tutorial Program Chair for COOTS, and Murthy Devarakonda, Advanced Topics Workshop Chair for COOTS
- Clem Cole, 2001 Freenix Program Chair
- Matt Blaze and John Kohl for the invited talks at the USENIX Annual Tech Conference
- Lee Damon for organizing the "Guru is In" Sessions as USENIX Annual Tech and LISA
- Greg Rose for the invited talks at the 10th USENIX Security Symposium
- Jon "maddog" Hall for the invited talks at the 5th Annual Linux Showcase & Conference

All the volunteers from Atlanta Linux Showcase, Inc. who helped in organizing ALS in Oakland: Chris Farris, Marc Torres, Greg Hankins, Paul Manno, Blake Sorenson, Hunter Eidson, Vernard Martin, Levien de Braal, Ben Cooper, Danny Cox, Valerie Cox, Sam Davis, Steve DuChene, Robbie Honerkamp, Ray Knight, and Piotr Misztal.

Esther Filderman and Tom Limoncelli for the invited talks at the 15th LISA Conference

Cat Okita and Tom Perrine for coordinating the Network/Security Track at LISA

Pat Wilson for coordinating the workshops at LISA

Esther Filderman for her hard work in organizing the AFS workshops at recent USENIX conferences.

Peter Honeyman for his continued efforts in reaching out to international groups e.g., SANE and HAL Conferences, Smartcards/CARDIS, Middleware, and Stichting NLnet.

Aeleen Frisch for her hard work (with assistance from John Sechrest and Steve Simmons) in evaluating vendors for Web-based training, and selecting one (Digital Think) that will provide their courseware at a discount to our members.

Brad Johnson, Tina Bird, Jerry Carter, and Evan Marcus for their efforts in launching and evaluating the Web Tutorial Pilot program.

Rob Kolstad and Don Piele for making the USA Computing Olympiad, which USENIX sponsors, a success.

Andrew Hume for chairing the Nominating Committee for the USENIX Board Elections, and Eric Allman, Rob Kolstad, and John Kohl for serving on the committee.

Andrew Hume for serving as liaison to the Computing Research Association.

John Gilmore for serving as liaison/point person to the Electronic Frontier Foundation and its legal team during a particularly litigious year.

Darrell Long, Avi Rubin, Mary Baker, and Peter Honeyman for serving on the USENIX Scholastic Committee which oversees the USENIX scholars and student research grant programs.

The SAGE Executive Committee members for their contributions: David Parter, Trey Harris, Peg Schafer, Barb Dijker, Strata Rose Chalup, Tim Gassaway, and Geoff Halprin.

Andrew Hume and David Parter for serving as liaisons for USENIX Board and SAGE Exec committee.

The following folks who served on committtees that are launching the SAGE certification effort:

SAGE Certification Policy Committee:

Lois Bennett, Stephen Berry, Mark Burgess, Barb Dijker, Bradley Donison, Tim Gassaway, Trey Harris, Andrew Hume, Mark Langston, Eric Smith, Mark Stingley, John Stoffel, Leeland G. Artra, and Julie Thornton

SAGE Certification Exam Development Committee:

Susan Alderman, Joel Anacker, Tom McDonald, Lois B. Bennett, Jeremy O'Leary, Russell Biggs, David Parr Lyndon, M. Colvin, Eric Smith, Trey Harris, John Stevenson, Philip Kizer, Mark C. Langston, William Lovins, Christine Wanta, Leeland G. Artra, Thomas Wong, Phil Temples, Vidya Tirupathi, and Tom Treat.

USENIX is grateful to all!

conference reports

15th Systems Administration Conference (LISA 2001)

San Diego, California

December 2-7, 2001

KEYNOTE ADDRESS

Slime versus Silicon

Greg Bear

Summarized by Steven Levine

The keynote address at LISA 2001 was given by much-awarded science fiction author Greg Bear. Mr. Bear spoke about new paradigms in our understanding of biological systems that encourage us to view cells, particularly bacterial cells, as nodes in a network, independent supercomputers that cooperate, communicate and use transfer media to alter themselves and each other. Who is particularly suited to understand and talk about this paradigm? System administrators.

Well, system administrators and science fiction readers, who are also characteristically open to listening to visionary worldviews underscored with conspiracy theories. Mr. Bear noted right up front that the LISA crowd was indistinguishable from the crowd at the largest science fiction conventions. He later extended the comparison by noting that science fiction fans are like children in that they are "eternally curious and not interested in fashion." Sci-fi fans, like sysadmins, are below the radar level of most of society. Mr. Bear's understanding of system administrators, and particularly the self-image of system administrators, was stunning.

The real LISA, says Mr. Bear, exists in the acreage surrounding the Town and Country Resort Hotel. LISA is the Laterally Integrated Stochastic Anticipator, the bacterial computer network in the soil system. A bacterial network is a slime machine, a bacterial supercomputer: a cell has three billion base pairs, each a computational unit. Bacterial cells communicate and cooperate.

Greg Bear

If a cell is a supercomputer and a node in a network, who administers the cells and the network? Who is nature's sysadmin? Not DNA, administering from the top down, as we have been taught. Cells can absorb information and change proteins by means of viruses and retroviruses. The viruses are the methods of communication between the nodes in a network. They seem to be necessary, which is why they are so tough to get rid of.

In a biological system, everything is a kludge. Change, however, is a necessity. Mr. Bear spoke of Barbara McClintock's work with jumping genes, DNA that changes systematically. A genome is like an ecosystem, and a gene must cooperate with hundreds and thousands of other genes; we know this from the evidence of embryological cells not cooperating, and the resulting problems.

What we are looking at is evidence of "social biology" (social biology – not sociobiology). Biology is social from the genome on up. We are now finding the very language to describe how systems work. And who speaks this language already? System administrators.

The old paradigms of biology – randomness, DNA writing to RNA in a process that never reverses – are "dead wrong." Yet the paradigms are still being

This issue's reports focus on on the 15th System Administration Conference (LISA 2001) held in San Diego, California, December 2-7, 2001.

Our Thanks to the Summarizers:

Mark Burgess
Marguerite Curtis
Yolanda Flores-Salgado
Liliana Hernandez
Doug Hughes
Steven Levine
Mark Logan
Armando Rojas-Morin
Joel Sadler
Michael Sconzo
Josh Simon, Coordinator
Tim Smith
Crystal Stockton
Jeff Tyler
Jin-ping Wan
Jason Wertz
Garry Zecheiss

taught. We need, instead, to look at the way we put systems together to gain some understanding of the problems of biology.

So, Mr. Bear says, go forth and study biology. You will learn how to administer systems; slime has been doing it for billions of years. All biological systems are networks of users, and users all have different priorities. We must learn to be open, to think like children, in order to deal with networks of users.

REFEREED PAPERS

STIRRING THE MATRIX: ORGANIZATIONAL SYSTEM ADMINISTRATION

Summarized by Tim Smith

Defining the Role of Service Manager: Sanity through Organizational Evolution

Mark Roth, University of Illinois at Urbana-Champaign

The presentation began with Mr. Roth defining a service as a collection of tools that allow users to do their jobs. He then reviewed the evolution of services over the past decade. Services in the early nineties were homegrown tools where no distinction was really made between the system and the services provided. In the late nineties, client-server applications, where there was some distinction between the system and the service provided, took the place of in-house software. Services today are taking the form of black boxes where the service software is distinct from the system it runs on, and users are not aware of the type of system used to run the service. The thesis of the paper is that in this new environment the role of service manager should be entirely separate from that of system administrator.

As presented, the role of service manager includes several components: initial planning, production deployment, and ongoing maintenance of a service. Thereís too much work required by the components to be handled by either system administrators or developers. System administrators are too busy to begin with, and their core competency is system management not maintaining services. In addition, system administrators need to achieve economies of scale in their work, and this is not possible in service management. Service management should not be handled by developers due to their incompatible time requirements and core competency in programming.

In Mr. Roth's approach the service manager focuses on the users of the service and is responsible for ensuring that the services needed by users are available. This means giving requirements to developers for in-house software and delivering system requirements to system administrators so the hardware service will be available when needed. The advantages of this approach as seen at UIUC include improved communication with the service manager as the only communication channel between system administrators, developers, and users; increased staff retention; easier budgeting; and achievement of some economies of scale. Mr. Roth pointed out that the approach is not fully in place at UIUC but that its advantages were already being seen.

Additional information can be found on Mr. Roth's Web page at *http://www.uiuc.edu/ph/www/roth/*.

New Technologies for Small and Medium Businesses (SMB)

Degan Diklic, Venkatesh Velayutham, Steve Welch, and Roger Williams, IBM Almaden Research Center

Mr. Diklic's presentation addressed the remote outsourcing of services for multiple branch offices and small businesses. In this presentation a small business is defined as having fewer than 500 machines, and a medium-sized business is a business with fewer than 5,000 machines. Servicing branch offices and small businesses is not an attractive venture for service companies trying to make a profit, because the clients are on the other side of a firewall, dedicated lines to bypass the firewall are expensive, a full-time administrator for a small site is also expensive, and there is no generic service infrastructure in place across sites. One idea is to remotely manage small sites using VPN and remote management tools. Current solutions that implement this idea, such as OpenView or Cobalt Blue, are expensive.

Mr. Diklic's solution avoids the expense of the available solutions while still allowing sites to be serviced effectively. The solution places a communication server outside the firewalls of the remote site and the service provider. These communication servers are owned by a trusted company, IBM in this case, and are used to make a connection between usher servers on the local LANs. The usher server is used to resolve network locations of the remote machines so they can be administered as if they were part of the service company's network.

This architecture has been used in several projects in Mr. Diklic's research group. The first project was a disk expansion project that allowed additional disks to be used as an extension of an existing disk in the remote site. This allows remote sites to share disk storage and makes remote data storage possible. The second project is a backup utility for remote sites. The backup of the remote machines is performed over the network at night when the bandwidth is not being utilized. Data restoration of user data is performed by an application CD that connects to the remote backup facilities and allows the user to access the data. Authentication is also handled by the CD since it contains the customer's username and password in a secure form.

TECHNOLOGIES INDISTINGUISHABLE FROM MAGIC: ANALYTICAL SYSTEM ADMINISTRATION

Summarized by Marguerite Curtis

A PROBABILISTIC APPROACH TO ESTIMATING COMPUTER SYSTEM RELIABILITY

Robert Apthorpe, Excite@Home, Inc.

Apthorpe began by expounding on how the tutorial on probabilistic risk assessment is a technique for finding vulnerabilities. He then addressed the reasons why he wrote it, talking about the problems that system administrators face. For example, they generally have little background on systems engineering and therefore have little context for understanding formal risk assessment, or they don't know how to detect problems, or they simply make a bad decision, like putting a primary and secondary server on the same switch. After acknowledging the problems, he spoke on why risk assessment is so relevant. His list contained about five points. For example, analysis is cheaper than firefighting and a good design defends against known problems.

In the second half of his talk Robert gave us the overview of his method, which consists of eight steps: define your problems; define your system; build a logic model of system failure; decompose system information of most basic actions and events; find the minimal sequence of events that lead to failure; estimate probability of event sequence from observed or estimated data; generate measures of component importance; and sanity check the model and the results. He then showed us an event tree analysis and a sample event tree. The next topic was how to use the results and what the weaknesses of the system are. He addressed these points and why they exist. Concluding, he spoke of his future hopes and plans, other possible research topics, and other applications, such as security, capacity analysis, or insurance and risk management.This paper won the Best Theory Paper award this year.

SCHEDULING PARTIALLY ORDERED EVENTS IN A RANDOMIZED FRAMEWORK: EMPIRICAL RESULTS AND IMPLICATIONS FOR AUTOMATIC CONFIGURATION MANAGEMENT

Frode Sandnes, Oslo University College

Sandnes began by explaining his idea: the schedule would automatically maintain a system state to benefit all users and would be achieved by tools such as cfengine. It can be viewed as a mix of dynamic and static schedules where the dynamic tasks are triggered by actions and the static tasks have precedence. One of his main points is that the user can allocate any task to a particular schedule. He moved on to discuss randomized strategies and algorithms, as compared to the scheduled algorithm. His objectives consisted of finding out how the randomized schedule affects the efficiency, ability to intervene, and ability to identify the config model. He compared a deterministic management framework with a random one to determine efficiency. In the second half of his talk, he addressed malicious intervention and how it relates to his work and randomization. If the abuser wants to uncover the model, assuming the abuser can observe, randomized scheduling can make it more difficult. His idea is that the abuser will be unable to observe a sequence of events if there is only a random sequence to look at. Sandnes concluded that randomized scheduling lead to consistent performance, reduces the predictability of management abilities, and hides strategies, as well as noting that the framework is easy to implement.

THE MAELSTROM: NETWORK SERVICE DEBUGGING VIA "INEFFECTIVE PROCEDURES"

Alva Couch and Noah Daniels, Tufts University

Couch began by stating his target problem, which is to automate network troubleshooting. His dream was to create a quicker response to network response. He then began showing how it all is formed, starting with pre-declaring precedences, which must be done every time a script is added. This is a pain, so he moved on to discovering order between scripts without declaring, claiming that they will fail robustly when called at the wrong time, tell you when they fail, and won't undo each others actions. Moving further into discovering order in ineffective procedures, he talked about how it is easier to check whether a condition is present than to execute it. Trading extra executions for lack of precedence tables is cheaper and less work. The efficiency depends on the initial ordering.In the second half of his talk, Alva explained what is necessary for the commands and what he learned from it all. Each command requires awareness as to whether or not it failed (which is the easy part), must be homogeneous (which is the hard part), and must be convergent. There are no preconditions to engineering maelstroms, and they are safe to run in any sequence. He learned that causality is not a myth and cannot determine what will happen. You can determine what repaired a specific problem, not what caused it. It is not causal, but operational. He concluded with tasks he is working on and will be working on, such as a troubleshooting script.

MONTE LISA OVERDRIVE: EMPIRICAL SYSTEM ADMINISTRATION

Summarized by Joel Sadler

PERFORMANCE EVALUATION OF LINUX VIRTUAL SERVER

Patrick O'Rourke and Mike Keefe, Mission Critical Linux, Inc.

O'Rourke presented a performance comparison showing the relative merits of Linux Virtual Server (LVS) over hardware-based load balancing alternatives. Patrick began by explaining what LVS was and how it could be used to improve a Web site's performance. Their testing showed that not only is LVS quite capable of competing with hardware LB devices, it can be dramatically less expensive per request/second.

Measuring Real-World Data Availability

Larry Lancaster and Alan Rowe, Network Appliance, Inc.

If there is a holy grail in sysadmin today, it's the much-coveted five 9s (99.999%) of reliability. This somewhat eye-opening presentation showed a view of reality specifically with regard to NetApp filers. Using data gathered from customers via ONTap's Autosupport feature, Lancaster showed how they had categorized the failure data and then laid out the conclusions the data had shown. Quite surprisingly, their data showed that fewer "Operator" type errors occurred than power failures, even with clustered systems.

Simulation of User-Driven Computer Behavior

Harek Haugerud and Sigmund Straumsnes, Oslo University College

Haugerud talked about the challenges inherent in building a model to simulate user behavior on a given multi-user computer system. His presentation showed their design principles and explained some of the initial goals they had in setting out. He then presented a fairly technical breakdown of their testing methodology. In doing so, Harek showed how they had tested the simulation with a known user-action data set obtained from a third party. Their results were impressive; while they admit that this particular simulation is in its infancy, its usefulness is easily visible.

SEEING HOW THE LAN LIES: NETWORK MONITORING

Summarized by Liliana Hernandez

Specific Simple Network Management Tools

Jürgen Schönwälder, Technical University of Braunschweig

Schönwälder described the design and implementation of an SNMP management tool called scli, which provides an efficient-to-use command-line interface to display, modify, and monitor data retrieved from SNMP agents.The SNMP management tools available today fall into one of the following five categories.

- Generic low-level SNMP tools
- Generic low-level SNMP APIs
- Generic MIB browsers
- Generic monitoring tools
- Generic management platforms

But the author still often feels uncomfortable when trying to use them; for example, the generic tools often do not understand the relationships between MIB objects.The software design addresses five key requirements: extensibility, robustness, maintainability, efficiency, and portability.The package uses the glib library to archive portability and to reuse generic data structures such as list and dynamic strings. The SNMP engine gsnmp has been derived from the gxsnmp package and was subsequently modified to fix bugs and to improve stability. The SNMP engine itself uses glib. The interpreter core and some command implementations also use the libxml2 library to create and manipulate XML documents.The SNMP engine does not yet support SNMPv3 security. The code generator can be improved in many ways. The biggest limitation right now is the restriction that stubs can only operate on table rows or groups of scalars.

Gossips – System and Service Monitor

Victor Götsch, Albert Wuersch, Tobias Oetiker, Swiss Federal Institute of Technology

Gossips is a modular client-server-based system monitor. Gossips not only reports problems but suggests solutions to the problems by consulting a knowledge base. The monitor software is written in object-oriented Perl.The goal in this project was to address some of the problems found with existing solutions like SNMP, Big Brother, Swatch, Spong, and PIKT.

- Big Brother is good in design, scalability, and messaging. It is okay in configuration and is extensible.
- Swatch is very extensible. It is okay in configuration, design, scalability, and messaging, but it is not modular.
- Spong is good in design, scalability, and messaging. It is okay in configuration, extensibility, and modularity.
- PIKT is good in configuration, design, scalability, extensibility, and messaging, but it is missing modularity.
- gossips is good in configuration, design, scalability, extensibility, modularity, and messaging.

The distributed architecture of gossips builds a scalable monitoring system. Through its flexible and central configuration environment, together with its command-line module, gossips is easily maintainable.The object-oriented design of gossips builds a flexible and well-defined framework for developing new monitoring tasks. The concept of separating data acquisition and data analysis makes defined monitoring tasks reusable and provides the possibility to build combined tests. The knowledge base allows one to archive solutions to known problems in one place and to integrate the knowledge of the system manager. By including cfengine, gossips could be extended into an automated repair tool.

The CoralReef Software Suite as a Tool for System and Network Administrators

David Moore, Ken Keys, Ryan Koga, Edouard Lagache, kc claffy, CAIDA

CoralReef is a package of device drivers, libraries, classes, and applications and provides a suite of tools to aid network administrators in monitoring and diagnosing changes in network behavior. CoralReef offers a unified platform to a wide range of capture devices and a collection of tools that can be applied at

multiple network levels. Its components provide measurements on a wide range of real-world network traffic flow applications, including validation and monitoring of hardware performance for saturation and diagnosis of network-flow constraints. CoralReef can be used to produce stand-alone results or data for analysis by other programs. CoralReef reporting applications can output in text formats that can be easily manipulated with common UNIX data-reduction utilities, providing enormous flexibility for customization in an operational setting. CoralReef provides a balanced collection of features for network administrators seeking to monitor their network and diagnose trouble spots. By covering the range from raw packet capture to real-time HTML report generation, CoralReef provides a viable toolkit for a wide variety of network administration needs.

LEVEL 1 DIAGNOSTICS: SHORT TOPICS ON HOST MANAGEMENT

Summarized by Jeff Tyler

Global Impact Analysis of Dynamic Library Dependencies

Alva Couch and Yizhan Sun, Tufts University

SoWhat is a tool for analyzing and tracing library dependencies in a large distributed environment. ldd can tell you what libraries any given program will load, but how do you determine the total set of programs in a large environment that might require a given library? The simple answer is that you don't, and thus one can never delete a library in a complex environment without a significant chance that some program somewhere in the environment will then break. Therein lies the path to library rot. SoWhat attempts to address this problem by analyzing and cataloging all library dependencies in just such a large environment. SoWhat currently runs on Solaris 7/8, with a Linux version promised. It's written in Perl and requires MySQL. It is freely available at *http://www.eecs.tufts.edu/~couch/sowhat.*

Deriving Tools to Administer Domain and Type Enforcement

Phil Kearns and Serge Hallyn, College of William and Mary

In this context, Domain and Type Enforcement (DTE) means a mechanism to provide fine-grained mandatory access control beyond the level provided by a conventional UNIX kernel. DTE systems normally use a rather densely populated text file as a control and policy establishment tool, and simple typos in these files can have a catastrophic effect on the surety of the system. Phil and Serge have addressed this issue by producing two tools to aid in administration of DTE configuration files and to provide a graphic view of system objects that any controlled program might interact with. The tools are called DTEedit and DTEview and are available at *http://www.cs.wm.edu/~hallyn/dte.*

Solaris Bare-Metal Recovery from a Specialized CD and Your Enterprise Backup System

Lee Amatangelo, Collective Technologies, and Curtis Preston, The Storage Group

Building on the success of their popular CART tool (first presented at LISA 2000), Lee and Curtis have constructed BART, the Solaris Bare-Metal Recovery Tool. CART was a system-specific tool, but BART is a networked version that can deal with multiple machines using an enterprise backup system and a single CD. It currently works on Solaris only due to Jumpstart dependencies and will operate with both Legato and Veratis NetBackup, although there are some Veratis issues.

Accessing Files on Unmounted Filesystems

Willem A. (Vlakkies) Schreuder, University of Colorado

Now *this* is a very useful utility. It is used for recovering files from unmounted disks and general bunged-up disk spelunking. If you've ever spent any time in fsdb you'll appreciate what went into the construction of this tool. It works like cat and has both stand-alone (ruf) and callable library (libruf) versions; it can automatically determine the location of alternate superblocks and perform other useful disk tricks. It currently works on *BSD, Linux, Sun OS/Solaris, and HP-UX. It is available under the BSD license at *http://www.netperls.com/ruf.*

TO YOUR SCATTERED PCS GO! DISTRIBUTED CONFIGURATION MANAGEMENT

Summarized by Tim Smith

Automating Infrastructure Composition for Internet Services

Todd Poynor, HP Labs

The automatically configured data center is an environment where the efficient redeployment of resources in the data center is required in order to meet changing demand. In this environment federations of resources from autonomous compute systems work together to provide a service. Such an environment does not exist today, but the framework presented by Poynor is a result of research and industry activity.

Poynor's talk presented a framework for composing Internet services from component services. In this framework, system administrators issue instructions to the computing resources on what services to deploy. The services that can be deployed are grouped into contexts that allow services to cooperate to achieve a larger goal and automatically discover new members of the context. Information is also provided to the framework

about deployment changes in the context that allow services to adjust and reconfigure relationships so the proper services are still provided.

Changes to the service deployment must be specified by an administrator or automated process. The affected resources are notified of the change, which allows them to start and stop component services and possibly reboot machines into new environments. The services add and drop relationships based on the current environment. Once the instructions have been provided, Internet services are stopped and started without administrator intervention. Mr. Poynor gave an example of what would happen when adding a new machine into a Web server farm.

The framework will require a protocol suitable for all hardware and software. The current prototype used by Mr. Poynor's group is an extension of the IETF Service Location Protocol. The framework, implemented with the protocol, extends UNIX system startup scripts or the Windows Services applet to allow the machine to be dynamically configured.

The PowerPoint presentation of the talk can be found at *http://www.hpl.hp.com/personal/Todd_Poynor/*.

TemplateTree II: The Post-Installation Setup Tool

Tobias Oetiker, Swiss Federal Institute of Technology

TemplateTree II addresses the problem of adding new machines into an environment. Each new machine needs an operating system and software packages installed and any site-specific configuration changes. All of the modifications can be made to a base operating system install using cfengine.

Oetiker's presentation covered how TemplateTree II can be used to generate the cfengine.conf files necessary to apply the modifications to a base system and POD-style documentation of the components that make up the modifications. TemplateTree II uses tools to set up subsystems including network configuration, the AFS client, and SSH configuration. Metadata are added to each subsystem description, which also includes the component configuration files, so TemplateTree II knows what each subsystem does.

Configuration of a base system using TemplateTree II involves specifying which subsystems to apply to the system. The metadata of each subsystem are used to create the cfengine.conf file. Once cfengine and the generated configuration file are installed on the base system, cfengine will finish installation and configuration of the subsystems, and the system will be ready for use in the system administrator's environment.

More information on TemplateTree II can be found at *http://isg.ee.ethz.ch/tools/*.

The Arusha Project: A Framework for Collaborative UNIX System Administration

Matt Holgate, Glasgow University, and Will Partain, Arusha Project

The Arusha Project allows system administrators at modest-sized sites to collaborate with one another on a large scale using the Internet. The presentation focused on ARK, an XML-based configuration language that can be used to describe system administration objects. An object is anything an administrator interacts with, including software packages, systems, and teams of administrators.

Partain's presentation showed how a software package can be described by different administrators using the configuration language. Each administrator began with different description fields, which include the package name, any administrator comments, the options used to build the package, and many other fields. Partain's presentation showed how isolated system administrators can collaborate with a few other system administrators via the Internet to exchange their package descriptions. Each administrator can take the descriptions from others and plug useful fields into his or her own description. As the updated descriptions of the package are shared, the package description at each site is improved.

Partain's examples showed how packages can be parameterized and inherited by other packages. He also showed how macros are created in the configuration language and clean up the parameterization of an object.

The Arusha Project home page can be found at *http://ark.sourceforge.net/*.

HUMAN INTERFACE: TIMELY SOLUTIONS

Summarized by Jeff Tyler

Lexis: An Exam Invigilation System

Mike Wyer and Susan Eisenbach, Imperial College

This paper won the Best Applied Paper award.

Wyer and Eisenbach faced the problem of converting (temporarily) a large number of Linux workstations to a highly secured configuration to allow students to take programming examinations while still maintaining network connectivity to a central server to collect test data. After the exams are over, the workstations have to be reverted to a more normal state. This transition has to be done repeatedly over the course of a semester.

They solved this problem with a combination of local and remote lockdown tools and a secured client-server configuration built around SSH and ipchains. They took advantage of tricks like mounting the root file system without suid bits active and heavy use of custom run levels. To activate a Lexis client one

simply changes to run-level 4 and the rest is automatic. The Lexis server, on the other hand, is a dedicated box with more traditional system security and maintains "Lexis state" at all times.

Mike provided a lot of detail about development and testing of the system, building confidence with students and staff, and discussed how they overcame problems such as scaling and system reliability. The Lexis system is in use at Imperial College and may be obtained under GPL at *http://www.doc.ia.ac.uk/~mw/lexis/*.

Javamlm, A Customizable Mailing-List Manager

Ellen Spertus, Mills College; Robin Jeffries, Sun Microsystems

The authors attempted to tackle a problem with which all of us are familiar: the fact that a successful mailing list soon generates volume levels that overwhelm some users, who then drift away. They studied and rejected some traditional approaches such as static sublists and user filtering and implemented a dynamic sublist approach (threads). This approach presumes nothing on the part of the mail client (e.g., no filtering) and allows the user access via a Web interface to adjust subscriptions and preferences. Javamlm works with qmail to do the heavy lifting (e.g., thread distribution) behind the scenes.

This effort was strictly a prototype and the authors intend to fold their work into mailman, the GNU mailing list manager.

GEORDI: A Handheld Tool for Remote System Administration

Stephen J. Okay, Road Knight Labs, and Gale E. Pedowitz, Protura, Inc.

This was an interesting talk and a fascinating paper. The authors detail their efforts to build a useful (and relatively secure) sysadmin remote access tool on a Palm Pilot. Quoting directly from the paper, "At base, GEORDI is a forms based UI wrapper for an RSA/DSA ssh connection to a remote host running sudo." GEORDI also understands about 60 UNIX commands and can recover state from previous sessions (scripts and commands built with its command-builder tool).

I suffer from the same bias that I suspect most of us do – if I can't get to a shell, then I tend to suspect the utility of the tool. The authors, sysadmins themselves, have gone a long way toward addressing this concern and producing a useful tool, given the inherent limitations of the platform. If you need to perform highly mobile systems administration, then GEORDI may very well be useful to you.

GEORDI is available under GPL at *http://www.GEORDI.org*.

ADAPTING THE COLLECTIVE: SHORT TOPICS ON CONFIGURATION MANAGEMENT

Summarized by Tim Smith

Pelican DHCP Auto-Registration System: Distributed Registration and Centralized Management

Robin Garner, Tufts University

Garner presented the DHCP registration system implementation used at Tufts University. The university has a class-B address space and about 9,500 systems. Six DHCP servers are used to service these machines. After experimenting with DHCP registration systems from 1997 to 1999, Garner was involved in the development of Pelican. Pelican was developed to address scaling issues not handled by the other registration systems.

Before registration the host is given an IP address from an untrusted portion of the address space. Pelican works by deriving a hosts MAC address when they register with the Web utility. The MAC is put into the dhcp.conf file, and the DHCP service is restarted every fifteen minutes to pull in the new MACs. When the new machine is restarted after DHCP has its MAC address, it is able to obtain an address in the trusted address space and see the entire network and Internet. Pelican also has functions to add and purge leases from the database, and to purge old machine registrations from the database. Garner concluded with performance results of Pelican.

A Management System for Network-Shareable Locally Installed Software: Merging RPM and the Depot Scheme under Solaris

R. P. Channing Rodgers and Ziying Sherwin, US National Library of Medicine

Network-shareable software has several problems. Packages are not independent of one another, and there is a need to allow host-specific "custom" packages. While locally installed software allows for host-specific packages, it is a large burden to maintain.

After covering previous work in the area, Rodgers noted that depot was selected for the project because it is a simple format, RPM was selected because its format is open, and each format contains information that complements the other.

Rodgers' presentation concluded with future work for the project. The RPM and depot functionalities should be coupled. In order for the combined format to work well, the RPM database needs to be modified to allow for dependency checks across the network. The documentation in the two formats should also be merged. Other RPM enhancements that would require modifying the RPM code would also be useful.

File Distribution Efficiencies: cfengine Versus rsync

Andrew Mayhew, Logictier, Inc.

The native file transfer protocol in early versions of cfengine did not work. To overcome this problem Mayhew put rsync in place to transfer files between systems. When the cfengine file transfer was fixed, the possibility of comparing it against rsync motivated Mayhew to compare the two.

The experimental setup used two machines on the same segment and compared the performance of unencrypted cfengine file transfers, encrypted cfengine file transfers, and rsync. Files transferred in the experiments varied in size from 128 kilobytes to 2 megabytes.

In the experiments rsync performed better on large file transfers, while cfengine was better transferring smaller files.

CfAdmin: A User Interface for cfengine

Charles Beadnall, W. R. Hambrecht, and Andrew Mayhew, Logictier, Inc.

Mayhew presented CfAdmin, a user interface for cfengine designed to allow facilities staff, release engineers, system administrators, and network operators to preview and edit information regarding systems and the software for them. Each of the groups needed to use cfengine for their work, so a common interface was created.

CfAdmin uses cvs for version control, cfengine for host management, and netcool for network monitoring. Apache with secureid is used for the interface. The interface allows the different groups to perform host entry and software location entry (e.g., binary paths, etc.). The system administrator then uses the interface to configure a system before deployment using software information from the release engineer. Facilities are able to use the interface to install the machine in the proper location.

A cfengine.conf file is generated by CfAdmin and then pushed out to all hosts based on the information entered. The automatic generation and distribution of the configuration file eliminates human error while updating the configuration file and makes cfengine easier to use.

WORK-IN-PROGRESS REPORTS

Summarized by Jeff Tyler

Email Redo Logs for User-Initiated Restores

Rich Graves, Brandeis University

Email redo is more concept than code at the moment but in use nonetheless. In a nutshell, create two mail spools and declare one read only (at the user level). Allow the users to recover individual pieces of mail from the r/o spool after they zap them in error in the traditional r/w spool. Supports UW-IMAP, currently running on Linux. Roll the spools on a systematic basis and expire the r/o spool at a reasonable point. A very clever idea that only requires simple changes to the local mailer to write both spools on incoming mail, some pointers to allow users to recover their own files, an expire mechanism for the r/o spool, and a LOT of disk space.

Bringing Undo to System Administration: A New Paradigm for Recovery

Aron Brown, University of California, Berkeley

Very much at the concept level at the moment, undo would provide the ability to "recover" at will to almost any point in time. Requires a LOT of prior planning. The concept is based on the system-recover three Rs: rewind, repair, replay. It sounded very transactional but aimed at base OS, not databases. It will be interesting to see where this one goes.

Operational Failures in Large-Scale Internet Services

David L. Oppenheimer, University of California at Berkeley

This case study – the first in what is hoped will be a series of case studies involving large-scale Internet-based enterprises – involved an Internet-based network storage company. Among the findings of interest was the fact that most failures were due to human error, which accounted for 33% of the events studied. The next largest cause of failure was listed as software failure and this was charged against the fact that software used was mostly home grown and being operated without rigorous change control. Oppenheimer is actively seeking companies or organizations willing to participate in his study and guarantees that no one will ever learn your name if you sign up.

Verdad

Jeff Kellem and Jeff Allen, tellme.com

Verdad is a central configuration store. It understands inheritance, versioning, and is based upon MySQL and Perl. It controls software, DNS and DHCP data, and user ACLs. It has a r/w Web interface. Sounds useful.

Establishing an Associate of Applied Science in Computer Security Degree

Will Morse, North Harris Montgomery Community College

Morse is working on establishing this degree program in the Texas community college system. He has a core curriculum, two OS-based courses, and a "lore" segment. He's looking for ideas and wants to draw upon the experiences of others in this area. His goal is to produce an entry-level security person who can meet 80% to 90% of the security requirements that a small business might have.

Installing Linux in under Three Minutes

Paul Boven

The keywords for this talk are PXC, BIOS re-direction, serial console, and remote power control. Then stir in DHCP, TFTP, and NFS. Ever dig into the Sun OS or Dec remote workstation build procedure? Then you understand 90% of this WIP. The other 10% is mostly in getting PC hardware smart

enough to boot off the wire. Paul seems to have a very nice scalable version of this ever-popular hack. It used to take us longer than three minutes to do this in Athena, but disks and the wire were both slower then. Boven can be contacted at *p.boven@sara.nl.*

The Condor Cluster Tool

Erik Paulson

It's a bird, it's a plane, it's a big cycle stealer! Currently at 1000 CPUs and growing daily. It's been adapted to interactive use and taught some manners (from the standpoint of the people whose cycles you are stealing). Currently uses only advisory locks and has some issues with reservation timeouts (they don't). In the words of it's author, "It's not too secure . . .." V2.0 is under development and will address security with Kerberos, have resource reservation timeouts, and stronger-than-advisory locking. Don't let those spare cycles go to waste, folks.

A Portable Linux Cluster

Mitch Williams, Sandia National Labs

This was an extremely neat hardware hack. Visualize this: a 4-banger Linux cluster in a tiny (5.3" x 5.3" x 13") custom rack. Weighing 15 pounds, it has its own little packing case that fits in the overhead rack on a plane. Built around the PC104 card buss system. Each CPU has 128 Megs of memory, and the whole thing is driven by a 50W power supply. They built it in a month from scratch for about $5,000. Seems like Sandia needed a PORTABLE teaching and demo system that could do some serious parallel processing. You have to see this thing to appreciate it – it's a jewel-like creation. I asked Mitch what he'd change if he had to do it again and he said, "I might make the power supply a tad larger, like 75W or maybe even 100." Mitch asked that, in addition to his coworkers at Sandia, I give credit to the Parus and Advanced Digital Logic corporations for all their help. This was the winning WIP and a very well deserved win in my opinion.

INVITED TALKS

CNN.com: Facing a World Crisis

William LeFebvre, CNN Internet Technologies

Summarized by Joel Sadler

It was a presentation that few will forget. LeFebvre showed in great detail how CNN.com dealt with the traffic load created by the 9/11 tragedy. He opened the talk by presenting some introductory information about how the CNN.com operations actually run. The group that handles the hardware for CNN.com also performs the same function for quite a few other Turner Web sites, including WCW.com, SI.com, TBS.com, and CartoonNetwork.com. All of the Web serving hardware for the various sites is identical, which allows for very simple "swings" of hardware among the various sites when required for special events or other heavy traffic times.

Moving on, Bill presented a time line with a stunning array of data about their traffic load. He showed that their inbound HTTP requests doubled every 7 minutes, with a starting metric of 84,719 hits/minute at 08:45 (all times EDT). By 09:00, they were already up to 229,006 hits/minute! At this point, the traffic monitoring software was shut down for several hours to remove any and all unnecessary load from the network and servers.

Meanwhile, the staff was scrambling to borrow servers from other sites so that CNN.com could continue to serve the exponentially increasing load. Starting with 10 servers at 08:45, they were able to increase that number to 52 by 13:00, including an amazing swing of 20 servers in a half-hour period. In addition to the server moves, they were minimizing the contents of the home page in an attempt to meet the overwhelming demand. At their lightest point, there were only 1247 bytes of HTML, with one small logo and a small picture.

Other interesting statistics of note: CNN.com's previous high traffic record was on 11/8/2000, the day after the US presidential elections. It reached a peak of about 1.2 million hits/minute for a total of about 139 million page views. On 9/11/2001, CNN.com successfully served about 1.1 million hits/minute for a total of about 305 million page views, not including the several hours that monitoring was deactivated. Their best guess as to the actual peak was about 1.8 million hits/minute.

Security for E-Voting in Public Elections

Avi Rubin, AT&T Labs — Research

Summarized by Crystal K. Stockton

Rubin talked about his previous experience with developing a system of e-voting for a public election in Costa Rica, where the voters needed to vote in their home districts. The government has already provided public transportation for those not living in their home district and has made the voting day a national holiday to ensure that everyone has a chance to vote, yet e-voting would help those unable to travel.

Problems they encountered were that each district had a different ballot, adults had little experience with using a mouse, and the computers were limited to regular voting districts. The written software took into account the problem of several different types of ballots, and it was suggested to use touch screens or light pens to compensate for the mouse, but the government did not have the extra funds. Other problems during testing were that all votes were recorded correctly for the primary Republican and Democratic candidates but wiped out the votes for other candidates. A power surge switched all votes from one candidate to another, and without an audit trail it was impossible to redistribute the votes.

Rubin also outlined possible threats to the system, social apprehensions, and technical issues. What types of consequences would there be if an attack were successful? How motivated are these attacks? What type of voting coercion, sale, or solicitations will there be? Is this a secure platform to use? What will the availability of the network be? How can it verify that a living person is voting?

Zope

Michel Pelletier, Digital Creations

Summarized by Armando Rojas-Morin

Zope is an open source Python-inspired object-oriented Web environment. With Zope, Web sites are developed through the Web itself.

One of Zope's strengths is its security system. Users have roles assigned, and actions are protected by permissions. It provides a file-system-like structure with a root folder. Permissions can be asigned to each subfolder. It's a good way to delegate. Because everything is done via the Web, things are not actually files in the file system.

Zope is more than just a server (HTTP, FTP, pcgi). Zope's philosophy is that data, login, and presentation should each be a different layer. This keeps all of the designers and developers happy by never violating their layer.

Zope offers relational database integration and content objects for the data layer; Python, Perl, and SQL for the scripting layer; and page templates and DTML templates for dynamic presentation.

2001: A Communications Anniversary

Peter Salus, Matrix.net

Summarized by Joel Sadler

Peter Salus gave a wonderfully humane and informative talk. He discussed the technological distance covered, primarily in the 20th century, to get us to the current level of communications dexterity. Peter also made mention of some of the leading innovations necessary to such advancement, such as the mechanical calculator, the telephone, and transatlantic radio messaging. He linked seemingly unimportant items together and illustrated their relevance in further advancing communications capabilities.

High points included the first transatlantic radio message in 1901; the first transistor in 1951; Clarke's short story "The Sentinel" in 1951 and its movie adaptation *2001: A Space Odyssey* in 1968; the birth of UNIX and of Linus Torvalds in 1969; Lyons' explication of the UNIX kernel in 1976; and the release of both PGP and Tim Berners-Lee's first HTTP work in 1991.

If I Could Talk to the Animals: What Sysadmins Can Learn about Diagnostic Skills from Another Profession

David N. Blank-Edelman, Northeastern University

Summarized by Crystal K. Stockton

Blank-Edelman began by comparing his profession to that of mechanics and doctors and pointed out several reasons why their work is completely different than that of a system administrator. He went on to say that a more accurate comparison would be to the profession of veterinarians. The reason comparing a system administrator to a mechanic is not accurate is that mechanics can remove and replace parts in determining a problem and fixing it. Their world does not fluctuate much, and they have various instruments available to help with diagnostics. Blank-Edelman believes comparing system administrators with doctors doesn't work either, because doctors treat others of the same species as themselves. They have the luxury of communicating with their patients. Veterinarians have the most similar profession to systems administrators because they work with a large variety of species, cannot easily remove and replace parts, and collect diagnostic information from a third-party source.

Blank-Edelman then talked about different types of decision-making. One type described in depth was deductive logical thinking. This is a classical approach to decision-making, which is usually what people are taught at an early age. Another type of decision-making is naturalistic decision-making where the environment influences your decisions. Types of environmental variables are time, pressure, high stakes, and experience.

After explaining different approaches to decision-making and explaining the reasons for trying to mirror sysadmin and user relations, Blank-Edelman showed a clip from the movie *Doctor Dolittle.* While watching this clip, he explained how a sysadmin's job is similar to a veterinarian's. This was a fun and light way to compare the two professions.

To review Blank-Edelman's slides and learn more about his research, refer to the Web site *http://www.otterbook.com.*

The Problem with Developers

Geoff Halprin, e-smith, Inc.

Summarized by Yolanda Flores-Salgado

"There is a problem with developers. They don't develop maintainable, production-ready, manageable code."

Maintenance is 70% of the software development lifecycle, but most developers can't maintain their software. They suppose and assume a lot, don't consider changes, don't provide enough documentation, and so on. Developers need to be re-educated, but sysadmins can't re-educate developers.

The only way is not to accept the product if requirements are not complete. If an application doesn't satisfy our needs, don't accept it, don't use it. Sometimes this is not possible, but if we could do it, our lives would be better.

An application has a life cycle: install, configure, manage, monitor, build, update, and de-install.

Developers need to know configuration standards. We need standard configuration files and logs. We also need a partitioning of file types – not all files are equal, and to improve file access, each location should be specified separately and set by the administrator.

Applications should be separated from system areas, and developers should give us the choice whether to isolate the application in a simple hierarchy and to have separated data areas.

Configuration management is very important. Proper configuration gives us the choice to manage the application's behavior. For simplicity of use, everything should go in one file if possible.

Sysadmins need control over:

- How to stop and start the application
- Application requirements
- User Management

Sysadmins need monitoring applications. We need standard logs, and log-file management (files rotating, configuring logs, etc.). Sysadmins also need backups. How easy is it to backup and restore the application?

Error handling – does the application trap potential errors? Does it report errors in a consistent format?

Sysadmins need installation instructions. Halprin said, "It is a sad statement that most software installations are done by executing the vendor installation scripts without question."

An application should provide documentation to install, de-install, and upgrade it. We need to understand installation procedures, but, most importantly, we need to be able to control them.

PHP for System Administration

Shane Caraveo, ActiveState

Summarized by Yolanda Flores-Salgado

PHP is a Perl/C-like scripting language designed specifically for the Web. It can be used on almost all platforms (e.g., UNIX, Windows, and MacOS), and it can be used either as a stand-alone language or as embedding SGML, XML, ASP, or JavaScript. PHP provides a lot of extensions supporting all of the commonly used databases (postgres, MySQL, Generic database, oracle), system protocols, and distributing processing.

PHP is easy to use and learn for sysadmins, especially if they are familiarized with C or Perl, and because it is Web browser-oriented, it can run anywhere. Web interfaces are also easy with PHP.

PHP can be (but is rarely) used for system administration. Since GUI interfaces are easy in PHP, PHP can be useful in building interfaces to delegate some sysadmin duties to non-sysadmins.

Some interesting PHP-related sysadmin projects are:

- PhpMyAdmin (PHP and MySQL – provides full MySQL admin capabilities)
- LDAP Admin (LDAP support via OpenLDAP or Netscape SDK)
- PhpQLAdmin (supports qmail, LDAP)
- Mailing List Admin. (simple EZMLM interface, but it could be better using EZMLM and MySQL. PHP provides PHP classes; easy to use. Lacks most options for make/edit lists.)
- proBIND (kindly interface for BIND config)
- PhpCron (simple Web-based cron server in PHP, integrated with crond)

Some PHP resources are: *http://php.net*; *http://SourceForge.org*; and *http://aspn.ActiveState.com.*

Rules of Thumb of System Administration

Steve Simmons and Elizabeth Zwicky .

Summarized by Tim Smith

The presentation was a collection of the wit and wisdom of the system administration field. "The only thing more frightening than a programmer with a screwdriver or a hardware engineer with a program is a user with a pair of wire cutters and the root password" is representative of the slides used during the presentation. After each slide Simmons or Zwicky would tell a quick story related to the slide and would point out the underlying rule of thumb or great truth that could be used by system administrators in their jobs every day. The slides for this presentation are not available online. The material used by Simmons can be found in his sigfile collection at *http://www.nnaf.net/~scs/Fun/sigfiles.html.*

What Sysadmins Need to Know about the New Intellectual Property Laws

Lee Tien, EFF

Summarized by Josh Simon

The short answer to the title, according to the speaker, is "a lot." He provided a general overview of the issues, but when in doubt, always contact your own attorney.

The theme of the legislation of late has been to figure out who controls the technology. Copyright law provides the creator of a work or expression fixed in some tangible medium, including electronic media such as RAM and disk storage, the right to exclusively copy, sell, and distribute their work and the right to authorize others to do so. Copyright infringement is when someone does any of this without authorization. There are two kinds of infringement: direct and indirect. Direct infringements are those where you yourself are the violator. Indirect infrigements are when there is a direct infringement and you're involved

intermediately. There are two types of indirect copyright infringements. The first is contributory, where you condone or help the direct violator, have knowledge (which has been extended to mean both "you know" and "you have reason to know"), and materially contribute to the violation, which includes the control of the facilities or the systems. The second type is vicarious, where direct infringement affects the right to control and leads to a direct financial benefit for the vicarious infringer. The example is of a tenant/landlord relationship. Since financial benefits are typically not present for system administrators, vicarious infringement probably doesn't appply to us. However, knowledge or reason-to-know do not apply to vicarious infringements.

So what can we as system administrators do? In smaller environments, we can avoid infringements. Unfortunately, this doesn't scale well. There's the so-called Betamax defense, which says if something can be used for substantial non-contributory use it's okay – but the courts aren't buying this argument yet, because it's only been applied successfully thus far to contributory, not vicarious, infringement.

What about Napster? They should have known there was infringement going on, and they provided the software and hardware (servers), so they've got contributory infringement. They also performed direct violations, and affected the right to control (vicarious) and cost the copyright owners revenue (vicarious). Even if only contributory infringement is involved, you can't foist it off and say it's someone else's problem once you have knowledge of it. So the advice here is to take cease-and-desist letters very seriously.

What about new legislation? Some case law shows that some knowledge is essential. Title II of the Digital Millenium Copyright Act (DMCA) provides safe harbors for ISPs and other providers, though the safe harbors are very complicated. A safe harbor provides immunity for monetary damages only and is intended to limit the legal exposure of the provider. There are four of them defined: transitory network passage, where all you do is deliver bits from one place to another, as in the Usenet model; system and caching, where you provide the hardware and OS but no monitoring; user-stored files, where you provide the disk space; and search-and-retrieval tools, such as Yahoo! The definitions and requirements and exceptions are all very complex, written in legalese, and there's very little case law behind them. In general, though, you have to meet the specific criteria for a safe harbor: you must have an anti-infringement policy, accommodate and not interfere with standard technical measures to protect copyrighted works, and comply with notice and takedown requests. Unfortunately, some of these terms, such as "standard technical measures" and "anti-infringement policies," are legally ambiguous.

The big question becomes who controls the technology of the Internet? The RIAA and others want to control it because it can be used to copy and distribute works to which they own the copyrights. The DMCA, in the opinion of the speaker, is a strategy to control devices, and it doesn't provide exceptions like the Betamax rule; so it requires the right to control access and to make devices to circumvent access controls.

Hardening Windows 2000

Phil Cox, SystemExperts Corp.

Summarized by Jason Wertz

For those of you out there who have to deal with Windows 2000 servers, there is always the question of how to protect your servers. What can you do as an administrator to make sure someone else's system is more attractive to a hacker than yours is? Phil Cox (*phil.cox@SystemExperts.com*) had many of the answers in his presentation, breaking the topic down into several parts. First, determine the purpose of the server. Second, what types of physical security are necessary for the server. Third, what should be done as the OS is installed to promote tight security? Next, what can be done to tighten the security on the server after it has been installed? Finally, he suggested ways to test the servers after they are locked down to make sure they are as secured as necessary.

Before a new Windows 2000 server is set up, its purposes must be established. Determine what services will be offered, which ones won't be offered, which computers the server is allowed to talk to, what domains and workgroups the server is part of, and what protocols the server will be using. If these answers aren't known, the server should not be set up.

After the server's purpose is known, one should decide how to secure it physically. At a minimum, case locks should be used, EEPROM passwords should be activated, and the hard drive should be designated as the first boot device if removable media is usable, and the server is publicly accessible. If the system is critical or highly sensitive, cages should be used as well. Other methods of physical security are up to the administrator.

Once the methods of physical security have been established, the installation can be done. When installing, use NTFS as the file system, set a good admin password, and install only the required network services. Do *not* upgrade from older windows servers if possible.

After installation is complete the administrator needs to figure out what services are actually running on the server. For each service to be kept, startup options

need to be set. Unnecessary services should be disabled or deleted. Deletion is the preferred method since a deleted service can't be restarted by a hacker, though certain services are difficult or impossible to delete. System policies such as password policies, account lockouts, auditing policies, user rights, startup/shutdown policies, etc. should be set at this time. Directory permissions should also be checked. Networking must be looked at, and filtering methods should be used to protect various used and unused ports. Time synchronization should also be used. Next, the active directory must be secured. Finally, install service packs and hot fixes.

Once the system seems secure, it must be tested out for security holes. There are many commercial tools that can be used for this. Unless great care has been taken, these final tests will likely show that there are a few bases that still need to be covered. It is much better to find security holes at this stage rather than when a hacker breaks into the system and exposes them.

To download the white paper that is the basis for this presentation and contains all the details for each of these steps, go to: *http://www.systemsexperts.com/literature.html* and download "Hardening Windows 2000" (*tutors/HardenW2K101.pdf*). There should be a version 1.2 of this file soon.

SANs and NAS

W. Curtis Preston, Storage Designs

Summarized by Mark Logan

Preston focused on the differing technologies of SANs (Storage Area Networks) and NAS (Network Attached Storage), and the strengths and weaknesses of each. He concluded with a look at the future of the two technologies and how they will be affected by the advent of NFS v4 and NDMP (Network Data Management Protocol).

The first segment of the presentation discussed the actual architectures of SANs and NAS. SANs is a fiber-channel network of RAID devices attached to a host machine. NAS, on the other hand, is an appliance (often called a "filer," a term coined by Network Appliance) attached to a LAN. NAS typically uses either NFS or CIFS as the network file system. Preston mentioned that running SANs behind NAS is becoming more and more common.

Preston attributed several advantages to SANs, such as reliability due to the ability to design systems with no single points of failure between storage arrays, but he was also fairly critical of the technology, pointing out its very high cost and its difficulty to administer.

Throughout the talk, Preston was much more enthusiastic about NAS technology. He was the first to point out that last year, he categorically warned against trying to run an RDBMS on a NAS appliance. Now, he is a proponent of the practice, after seeing numerous success stories. The advantages he attributed to NAS included ease of administration, its generally superior speed measured against equally priced SAN and local disk solutions, and many of the "goodies" being included by NAS manufacturers, such as snapshots and truly advanced file systems.

He was quick to mention that a NAS system does suffer all the flaws of the network file system it implements, and that NAS presents some problems in the realm of backups. However, NDMP promises to fix many of these problems by allowing backup from filer to self, filer to filer, filer to backup server, and server to filer.

In conclusion, Preston declared that the choice between SANs and NAS was largely dependent on the particular situation. The overall message of the talk, however, seemed to be that NAS was really maturing and becoming more affordable, but SANs still offered more in the realm of high availability and performance.

NETWORK/SECURITY TRACK

Whither End-to-End: Placing Bandwidth and Trust at the Edge

Gordon Cook, The Cook Report

Summarized by Jin-ping Wan

Gordon Cook, the author of *The COOK Report on Internet*, a monthly newsletter on Internet infrastructure development, calls for customer-empowered network infrastructures, in which customers control bandwidth and other resources over telco-empowered network infrastructures. Today's Internet is dominated by a few supercarriers; in the interests of the public, this should change to a scenario dominated by customers' networks intersecting global and other local networks. This is analogous to computing, which was dominated by large mainframe computers 40 years ago, and changed to personal computing due to the proliferation of mini-computers in the 1970s followed by the PC. Gordon uses Canada's CA*net4 to illustrate an edge-controlled infrastructure that has customer-owned networks with fiber bandwidth to the users.

Crypto Blunders

Steve Burnett, RSA

Summarized by Mike Sconzo

Cryptology can be a powerful and secure way to send data, but it must be used properly. Algorithms can range from simple to complex, from almost unbreakable to easily broken. Whatever the algorithm, the use is the same: encrypt data and keep it safe from attackers. Five blunders were introduced in this presentation.

One of the newest blunders is to declare one's algorithm unbreakable. Several crypto schemes have done this and as a

result were quickly broken having garnered the attention of people wanting to be "the one who broke the unbreakable." The newest trend is using a security proof to prove mathematically that your algorithm is "perfect." One such case was the Atjai-Dwork crypto system; the algorithm was "proved" to be unbreakable in 1997 and broken in 1998.

The second pitfall is "worshiping at the altar of the one-time pad." Since the one-time pad crypto system was proved to be perfectly secure by Shannon, several companies and individuals have tried to use this to their advantage. The problem resulting from adopting the one-time pad to suit specific needs was demonstrated by Microsoft in 1998. Microsoft created a product that used the one-time pad as part of an algorithm, but the pad was used twice. This allowed people to figure out what the pad was and attack the traffic.

Another problem arises from not using the best available algorithm, which leads to the commonsense question, why should an algorithm be used when it is known to be insecure? This also leads into the next blunder: an incorrect implementation of an algorithm. This was illustrated by a story about a man who called with a complaint about the RSA he had implemented. No matter what he did, his message always encrypted to itself. When asked what he was using as his exponent he replied "1". In the formula $c = m^x \bmod n$, if 1 is chosen for x, then m will always equal c. Lesson learned.

Finally, blunders can stem from an incorrect implementation or just poor security policy. This happens when you "don't protect the key." If the private key is not protected, any crypto scheme becomes near trivial to break. Some famous instances of not protecting the key arose both from Microsoft and Netscape.

It is important to keep an eye on the crypto system that you are currently using. Make sure that it is not out of date, you have a good implementation, the private key is kept private, the algorithm is used correctly, and it is a good system to use. Finally, even if all those are present, the issue becomes one of trust. With third-party systems in place to verify identities, who has to worry about invalid certificates? Then again, maybe we *should* worry.

How Not to Configure Your Firewall

Avishai Wool, Lumeta Corp.

Summarized by Joel Sadler

Wool presented a fast-moving talk on firewall configs, mostly centered around Checkpoint Firewall-1. He opened with some overall policy auditing concepts before moving to common misconfigurations. Unsurprisingly, the most common errors that he's seen are allowance of all DNS traffic (both inbound and outbound), improperly controlled ICMP, and general problems with misuse of the "ALL" directive. Using example client data, he showed specific firewall configurations with serious problems. His strongest recommendation to firewall administrators was to keep their policies as simple as possible. His data showed that as firewalls grew in complexity, not only were new vulnerabilities introduced, but the danger of exploiting existing vulnerabilities increased.

The Future of Computer Security

Moderator: Marcus Ranum, Network Flight Recorder; Panelists: Tom Limoncelli, Paul Proctor, Anne Benninger, John Flowers, and Steve Atcheson

Summarized by Mike Sconzo

The question posed to each member of the panel was "where will we be in 3, 5, and 10 years?" One believed that computer security will get much worse before it gets any better. Although quite a few people think that the majority of the problems will be partially if not completely solved within 10 years. This is because we should have better tools and more knowledge about the problem(s) we are trying to solve.

It was pointed out that we are headed in the right direction, and with the increased realization by management that security is important, there will be more spending on security-related technology. The industry can then produce such true security products as a (nearly) self-correcting operating system and products that enhance security rather then just acting as burglar alarms. These ideas about security and security enhancements will eventually trickle down to IT people, and this will eventually help with the current state of security.

No matter how great the technology is, however, we will still have problems due to human error. This can be mitigated through education. Consolidation of products/ideas is also anticipated so that products will be easily scalable.

The need for a standardized system of evaluation was also brought to light. Some panelists agreed that the business sector (e.g., insurance companies) would influence this. It might eventually become possible to buy network security insurance. The insurance would be priced according to how secure the network was, and this would lead to a security rating system. Introduction of VISA compliance standards might be another way to achieve this.

Several other issues were discussed, such as Public Key Infrastructure, authentication, and risk measurement. Everybody seemed to agree that "management is bad, insurance and government standards are good, and PKI is dead."

GURU SESSIONS

AFS

Esther Filderman, PSC, and Garry Zacheiss, MIT

Summarized by Mark Logan

The AFS session consisted of questions about bugs in various AFS implementations, questions about AFS setup and administration, and commentary about the state of the AFS community. The discussion took place in a packed room containing everybody from longtime veterans of AFS to people who were just curious about AFS.

The first round of questions addressed bugs in certain setups that seemed to be caused by Jumbograms, which are on by default in AFS. Jumbograms can speed up AFS communication in some cases, but the consensus was that they should be the first thing to go anytime strange bugs start to show up.

The discussion then turned to issues of AFS performance, including caching to disk and/or memory, and how AFS performance compares to that of NFS. One attendee told of having dramatically improved workstation performance with a relatively small memory cache, while another shared his experience of running AFS with a 2GB memory cache on an E10K. Allegedly, it was rather sprightly.

Discussion of backup was bound to come up sooner or later, and it centered around alternatives to butc, the most common backup tool used with AFS. However, the only tool about which anybody had anything good to say was Tivoli Storage manager. A few folks who were unconcerned with preserving AFS ACLs reported success using tar and commercial backup products.

Toward the end of the session, the history and future of AFS came up, and Esther and Garry fielded questions about the Transarc implementation, the progress of OpenAFS (which is now reportedly quite stable), and the directions that AFS may take in the future. There was some speculation about the possibility of an AFS Foundation, but the gurus were not optimistic about such an organization actually being founded in the near future.

Infrastructure Architecture

Steve Traugott, TerraLuna, LLC

Summarized by Tim Smith

Topics suggested for discussion were large systems beyond credible manual reach; industry acceptance and understanding of infrastructure architecture; notations, semantics, and type checking in infrastructure architecture; organizational infrastructure; and turn-key management solutions. However, organizational infrastructure and notations, semantics, and type checking were not discussed before the sessions ended. Traugott began the discussion on large systems by saying that without automation and tools there is an infrastructure such that an infinite number of system administrators could not administrate it. The key to managing such large infrastructures is centralized management of the systems and network. The discussion then shifted to tools used to centrally managed network hardware. On the software and configuration side, how to avoid changing the infrastructure manually was discussed. Manual changes should never be made even though the pressure to immediately fix a problem is great. Instead, the changes should be made using the tools available. If this is done correctly, a machine can be reformatted and all of the customization after the base operating system installation can be recovered automatically.Industry acceptance was the next topic covered. The primary problem faced in this area is describing to recruiters and bosses what an infrastructure architect does. An infrastructure architect is concerned with reducing the cost of ownership of managed systems through careful planning of the infrastructure of the computing environment. They are typically a senior system administrator who can code well and whose intent is to build the view of a single enterprise architecture. Infrastructure architecture can be thought of as something to do after system administration. An infrastructure architect is needed to make the design decisions because if these decisions are not made by someone assigned specifically to the task then they are made during triage situations in emergencies, and this is clearly evident in the resulting infrastructure.Turn-key management solutions for infrastructure architecture do not exist at the moment. The best solutions would be written by vendors, but those would not be acceptable in heterogeneous environments. Any solution produced for a site tends to be site specific and not really sharable. The Arusha Project (*http://ark.sf.net*) was mentioned as an effort to make site-specific efforts sharable between sites using their XML-based configuration language. Standardizing GNU tools and infrastructure policies are other ways to allow sharing.More information can be found on *http://infrastructures.org*.

PKI/Cryptography

Greg Rose, Qualcomm, Inc.

Summarized by Mark Logan

The PKI guru session started off with the announcement that the AES has been approved. Rijndael, the winning cipher submission, was accepted with few modifications. Rose expressed his satisfaction with the committee's choice and cited Rijndael's propensity for encrypting very quickly as its biggest strength. He added that there was hardly any doubt that all of the ciphers under consideration were secure, so factors such as speed were given greater weight. Discussion focused for a while on problems with current PKI implementations. Top on the list of gripes was the diffi-

culty in issuing certificate revocations. Rose and a few attendees explained several different approaches to addressing this problem. The simplest one was to simply set certificates to expire relatively quickly. One of the more interesting proposals involved storing part of a key on a trusted server, so that to sign or encrypt documents, users would have to pass part of the computation to the server since they would hold only a portion of the key. Then, instead of revoking certificates, an administrator would simply remove the key material from the trusted server, and the relevant user would not be able to sign or encrypt documents. In the second half of the session, several attendees had questions regarding the state of export regulations. There was some trepidation in the room about what the US legislature could be expected to do with regards to cryptography export regulations in the wake of September 11th. Rose felt that regulations were still quite relaxed compared to those of several years ago and that there was little cause to worry for the time being. He justified his stance by arguing that export regulations were never meant to keep cryptography from leaking across US borders but, rather, were meant to keep any non-US business from using US cryptographic technology to do business.

Writing Papers for USENIX Refereed Track

Tom Limoncelli, Lumeta

Summarized by Josh Simon

Tom noted that publishing a paper was good since it helps the community and can change your career (allowing for both peer and management recognition, and providing ammunition when your boss needs to justify your next raise).

How does one start writing a paper? The advice here is to write what you know. Are you doing anything to make your life easier? Automating a task? Writing a cool tool? Working on a neat project? Providing a case study, whether positive ("Here's what we did, and it worked") or negative ("Here's what we did, how it broke, what we did to fix it, and what we should've done to begin with")? Asking yourself, "What have I done that nobody else has" is an excellent way to start. Then following up with the terms and concepts and a statement of the problem, its scope, and how you solved it provides a good basis for your paper.

Don't forget to survey the literature. With the publication of *Selected Papers in Network and System Administration*, or "The Best of LISA" as it's been called, there's a single place to go to find references. Add to that the resources available to all USENIX members on the *http://www.usenix.org/* Web site and you're definitely off to a good start.

Tom also discussed the evaluation process, based on his experience serving on or alongside several program committees. The readers consider whether the paper is enduring and whether it can result in a good presentation. Papers are evaluated on several criteria, including the technical quality of the work, the presentation of the paper, whether it advances the state of the art of system administration, and whether it's relevant for LISA or somewhere else.

If your paper is not accepted, don't consider that anything more than a momentary setback. Papers are usually returned with commentary that explains why it was not accepted and suggestions on where to submit it (if not to LISA next year), along with commentary on the paper's quality, presentation, and so on.

If your paper is accepted, meet your deadlines. Work with your shepherd, whose job it is not only to nag you to make deadlines but also to help you by providing constructive feedback on what is good and what isn't. The shepherd is a resource to help make your paper the best it can be. Remember that they have their own lives to live but that they are willing to help you out – just don't deliver a draft and expect same-day turnaround.

Some additional commentary:

- Both proofread and spell check your paper. Have someone else proofread and spell check your paper. You're too close to it by the final submission deadline, so another set of eyes can help a lot.
- Do your presentation beforehand. Practice in front of a mirror, or present it to your team, department, or company, or to your SAGE local group.
- Give away the ending early. You're not writing a mystery novel; it's a refereed paper. You should identify the problem you're trying to solve and how you solved it in the abstract, the introduction, or both. You should also spell that out early in the presentation.
- In your presentation, consider demonstrating the tool (if your paper is about a cool new tool). Also, consider what your audio/visual needs will be: laptops, transparency projector, microphones, any special needs. The AV team needs as much advance warning as possible.

Finally, we discussed some paper ideas and how best to present them for future conference paper tracks.

WORKSHOP SERIES

cfengine

Moderator: Mark Burgess

Summarized by Mark Burgess

The cfengine workshop, led by Mark Burgess, attracted 21 participants from all backgrounds to discuss the current and future developments in cfengine v2. Among the highlights: a discussion, prompted by Steve Traugott, as to whether it is best to determine a config-

uration as a sequence of known steps (Steve's view) or as the specification of a final state (Mark's view) – the two are not always equivalent; Paul Anderson talked about the need for even higher-level specification languages at the enterprise level; Andy Mayhew and Christian Pearce discussed how cfengine could be enhanced for the enterprise with ancillary tools like CVS and LDAP; and Martin Andrews talked about using cfengine on Windows NT/2k. The workshop produced a stimulating discussion, which is summarized at *http://www.cfengine.org*.

MetaLISA

Tom Limoncelli, Lumeta, and Cat Okita, Earthworks

Summarized by Josh Simon

The MetaLISA workshop about managing system administrators first discussed the question of how to provide motivation to help retain quality personnel. We decided that providing a good work environment without major stressors would be better than just throwing money (salary) at the problem, that authority and responsibility should both be well defined, and that resources have to be made available to handle problems.

Next we discussed the different types system administrators: the "work 9 to 5, get a check, leave work at work" type and the "computers are my life so I play on them at home, too" variety. One manager organized his group so the former type was given the trouble-ticket queue processing and the latter was given more of the infrastructure and hard install problems.

We then discussed career path issues. Several companies now have multiple paths and levels, such as team lead, project lead, and assistant manager, each with appropriate and well-defined levels of expectations, evaluation scores or results, experience, requirements, and so forth. Providing different levels of responsibilities, independence, authority, and even money (base pay increase) on a path for both technical and management types, junior to senior, seems to work well. Even better is when there are well-defined criteria for promotion and lateral transfers between tracks. Remember, however, to provide allowances for exceptions or case-by-case waivers in your written policies.

Next we covered professionalism. Some people lie on their resumes. Some people wear inappropriate clothing (suit or t-shirt) to an interview or to the job itself, and don't alter their clothing choices even when informed to do so. Some people don't understand the concept of punctuality. Some people don't know how to be tactful, to provide the right level of information, or even to say "I don't know" to the customer. One topic of discussion was how to educate these people to improve these skills. Information sharing – such as email lists, databases, and even IRC channels – helps teams to share knowledge, cross-train people, and provide a way to let everyone contribute. Admitting when you're wrong builds trust for when you're positive that you're right. Encouraging people to ask for help can work, but so can offering help and asking people if they need help. However, the insecure may not respond or take you up on the offer. For these people, it may help to present a situation as a "show me what you did so I can learn." Don't use killing statements (such as "you're wrong") but ask leading questions ("what if").

Next we looked at determining what information is important (to share) and what is not (to keep political fallout from the team)? One technique is to have a staff meeting and say, "Here's the important stuff" and then let folks leave if they don't care about the politics. It's necessary to get folks to realize "best" isn't always "right" and that politics can override the right technical basis. The team needs to be aware that there are politics even if they don't know the details. However, email is often not the best medium for this; in-person and telephone contact may be a better (or, at least, a good supplemental) way to contact and inform people. Also, giving people the framework to put the details in and answering their questions is good. Some people, though, just don't care about the political issues. Sometimes, having face-time in meetings with your people and the lord high political muckety-muck may be useful.

Many people can follow a checklist and don't have problem-solving skills. How do you teach them to acquire the skills? Problem-solving is linked to curiosity, background, and experience. Teaching people skills is important. Using child-raising techniques, such as brainstorming with a timer, may be helpful. Again, you have to be careful to ask leading questions and not use killing statements that make the other person defensive. People need to remember to look at the big picture in order to make informed big-picture decisions so questions of direction get addressed within the group. Also, the problem and scope need to be explicitly defined, because that sets some limits. Finally, the instruction or detail level of the recipient may be relevant; instructions to senior people may be much less detailed than instructions to juniors.

The next major discussion topic was the balancing act between technical and management responsibilities and tasks. Some of the tricks include allowing the people who report to you to assume you're still technical, knowing that the theory can be as good as the technical details, keeping yourself informed about major issues, and using one day a week as a technical day for working on small projects. Also, if you only rise to the point of comfort, whether that's team lead, division lead, project manager,

company head, or whatever, you may be better able to find your balancing point. One of the problems is trusting the people to whom you hand off your pet projects to will do the "right thing" with them.

Finally, we discussed moving from an ad-hoc group to a more procedurally based group, formalizing processes by documenting not only how things work, but also why the decisions were made. Pairing people, one to explain and one to write, can work well. Starting with something like script is a good start. Having a cheat-sheet or template can be very helpful. But you have to practice what you preach; document things yourself so your people will. Also, make documenting a requirement for the performance review.

Sysadmin Book of Knowledge Project

Moderators: Geoff Halprin and Rob Kolstad

Summarized by Rob Kolstad

Twelve people spent the day planning out the next phase of the System Administration Book of Knowledge (BoK) project that Halprin started a few years ago.

Halprin and Kolstad opened the workshop with over three hours of presentations that we have independently been delivering to audiences over the past year. The presentations introduced and motivate the Sysadmin BoK.

The Sysadmin BoK is intended to name all of the items that a system administrator might encounter in his or her work. It goes little further than naming the items – it is not a tutorial or "best practices" document. In its barest form, the BoK will end up being a list of 2,000 or so line items (e.g., "backups," "security policy for firewall," etc.).

When annotated with a paragraph or two for each of the line items, the BoK becomes:

- A weighty tome to impress those who wonder what sysadmins do for a living
- The basis for curricula – university/HS, training, advancement, individual career planning (the individual point of view)
- The basis for creating a benchmark for corporate IT/admin maturity (corporate/organizational point of view)
- The basis for creating best-practice documents (refining the knowledge)

In combination with the above, the BoK forms the foundation of system administration as a real profession. It is only the beginning, but as a strong foundation, we believe it is essential.

Currently, the BoK has about 1500 line items created by a core team of a dozen sysadmins and occasionally reviewed by a total of just over 100. The line items have been categorized into a 73 x 44 matrix whose rows are general topics of sysadmin and whose columns are specific properties of those topics (e.g., "security" and "mobility"). Printing the list in 9-point type, two columns, small margins yields 19 pages of line items.

Discussion over the remainder of the day yielded these action items:

- Combine some of the rows and columns to reduce the number of elements in the matrix.
- Add new rows and columns, as contributed over the last few months by reviewers.
- Fill in another 500 or so line items.
- "Factor out" common elements that appear throughout a column (e.g., common security elements).
- Write a paragraph or two for each line item.
- Find a way to present the matrix in linear form (i.e., in a book).

Then we can proceed with the subsequent BoK projects:

- Capability maturity model
- Encyclopedia

The project particularly needs reviewers with a strong background in administering sites with Windows, both on servers and desktops. If you would like to contribute, please contact *kolstad@delos.com*.

The project Web site is *http://ace.delos.com/taxongate*; trivial registration is required so that your contributions can be tracked.

AFS

Derrick "Dana" Brashear, CMU; Ted McCabe, MIT; OpenAFS Elders; and Esther Filderman, PSC.

Summarized by Garry Zecheiss

Twenty people interested in AFS, OpenAFS, and Arla participated in the AFS Workshop, either by giving a short presentation or by suggesting topics and contributing to their discussion.

Derrick Brashear gave an update on the status of OpenAFS. New ports are available, including MacOS X and Solaris 9. There are many new features, including some support for AFSDB resource records, dynroot support, and a new build system using autoconf, as well as some bug fixes, including better RX tuning. Long-awaited features, such as disconnected operations and Kerberos 5 support, are coming soon.

Love Hornquist-Astrand gave an update on Arla. New features include more support for MacOS X and FreeBSD and support for incremental file caching. A lot of RX work is being done, including GSSAPI/SPNEGO support and the removal of unwanted features. Love and Magnus Ahltorp have been working half-time on Arla but the funding for this stopped at the end of 2001.

General discussion covered topics such as Arla-AFS compatibility; cache man-

ager issues, including using memcache; backups – what are people using and how do they cope with the Transarc backup system; migration to Kerberos; AFS infrastructure tools; proactive AFS administration; performance tuning; and getting funding support and publicity for OpenAFS.

Details about the workshop are available from the AFS workshop Web site at *http://www.psc.edu/~ecf/afs-workshop/*.

Teaching System Administration

Moderators: John Sechrest, PEAK Inc., and Curt Freeland, University of Notre Dame

Summarized by Tim Smith

The third annual Teaching System Administration workshop consisted of four sections. The first section focused on learning objectives for a system administration course. These objectives ranged from basic system administration tasks such as creating and managing user accounts to more advanced issues such as needs assessment.

The second morning session focused on assignments for a system administration course. The discussion included how quizzes, tests, labs, and projects could be structured. During the discussion the participants who had already taught a system administration course related what they had done in their course and how it worked. The participants broke up into groups to design a lab for one of the learning objectives discussed in the first session. Presentations of the labs concluded the morning session. During the lunch break participants discussed how to evaluate success in a system administration course.

The first afternoon session began with a group summary of the different discussions held during the lunch break. This discussion lead into the topic of exam questions. Different types of exams, from essay to multiple-answer and true/false, were considered based on the ability to evaluate what a student has really learned and how easy it is to grade. Based on this discussion the participants broke up into their small groups from the morning session to come up with three exam questions. Most of the questions produced by the different groups were short answer, with a few multiple-answer questions and essays.

The final session of the workshop centered on large and real-life projects. The discussion was about how to bring real projects into the class without inconveniencing a business owner while still allowing students to apply what they were learning in class. The discussion shifted to the types of projects that had been assigned by the participants who had taught a system administration course or completed projects in a similar networking course.

The workshop concluded with a discussion of tools, primarily ones that would make grading easier and allow materials to be shared between individuals teaching system administration courses. Ongoing discussions about this issue continue on the sysadmin education mailing list at *sysadm-education@peak.org*.

Advanced Topics

Moderator: Adam Moskowitz

Summarized by Josh Simon

The Advanced Topics Workshop was ably hosted and moderated once again by Adam Moskowitz and co-piloted by Rob Kolstad. Introductions by the 26 attendees generated interesting questions and topics for discussions: random opinions, the Undo command for sysadmins, hot tools, and surprises from the past year.

Random Opinions

People are indeed using SANs and NAS, since they're well suited to specific problems (such as archiving, Fortune 1000 companies, and so on). However, they are not being used for general file services, mainly because the FibreChannel implementation is too expensive for general use.

We also discussed the centralization/decentralization pendulum, which seems to be moving back toward centralization. Perhaps condensing is a better term, since places are condensing locations for their hardware and personnel but still keeping some geographic distance between them. Centralizing administrative functions is different from actual physical centralization, since (to use the SAN/NAS model), users don't care if the disk is local or across the continent as long as the performance is unaffected.

We're moving toward more of an ASP model within a given environment, be it company or infrastructure. The ASP model works well between divisions within an organization but not as well between different organizations, primarily due to trust issues.

The events of September 11th caused a shift in the thinking of some of the tight-fisted financial staff. They now realize how integral computing is to business, so collocation and backups are now more important.

The next major topic was mobility. Without mobility today's commonplace high-speed network infrastructures and reliable file servers make a lot of system administration fairly easy; workstations can be built from images or automated installation processes, and all mutable data lives on centralized file servers, where it's easy to backup and manage.

But mobility changes all that. Mutable data has to be local to the endpoint (e.g., laptop); we can't expect network connectivity to be high-speed, and we have to be able to deal with connections over insecure networks. We have to deal with a host of security issues, find new ways

of ensuring data availability, and be able to provide the needed services of various levels of network quality.

Mobility is becoming increasingly important – there are now many organizations where most endpoints are mobile platforms. But IT infrastructures have not yet caught up to this changing reality. To deal with this we will have to abandon our traditional (and previously successful) modes of thinking and use technologies that involve disconnected operation, mobile IP, synchronization, transparent data encryption, and so on.

Wireless computing has changed our behavior; 70% of us in the ATW are on laptops. Our expectations seem to be that we're approaching ubiquitous computing; of those using laptops, about 2/3 use them to access remote services (mail, Web, files) and 1/3 use them as the centralized storage point. This leads to the intrusion of mini-environments into your own macro-environment. Laptops can move from administrative domain to administrative domain and pick up and distribute viruses and whatnot in the process. Managing and keeping them from screwing up your environment is a hard problem.

Recovery-Oriented Computing

Recovery-Oriented Computing (ROC) is targeted to services. A PowerPoint presentation is available, along with information at *http://www.cs.berkeley.edu/~pattrsn/*.

The goals are ACME – availability, change, maintainability, and evolutionary growth – instead of performance (which is what we've looked for in the past 15 years). We're not doing that well.

One of our needs is not just to get real data to improve reliability but to measure reliability and availability. Making the system administration tasks have an Undo function may be helpful. Think about the three Rs: rewind (go back in time), repair (fix error), and redo (move forward again). We're looking to recover at the service level, not just at the server (hardware or component) level.

- Predictability – Having predictable recovery would be a huge improvement even now. Most recovery plans (or even risk mitigation) is pure guesswork now, based on experience and trial and error. Change control and change management need to be more formal and actually predictive of detailed determination.
- Avoidability – Can you avoid the problem to reduce the recovery time? If you can avoid the problem then the need for recovery time is less. This is reasonably important and very hard.
- Repeatability – Making tasks easily repeatable will help reduce complexity and can lead to increased avoidability and thus increased reliability.
- Risk mitigation – A lot of the changes we make at one time – one change – affect multiple machines (such as servers, routers, switches, firewalls, and so on). Rollback within any one system is good, but we need to have rollback in all of them. The problem becomes system-specific; is it a GUI or CLI?
- Tools – They're trying to reduce the MTTR in the MTTR/MTTF equation. This project is more about building recovery-from-something-that-has-happened than making-the-problem-less-likely-to-occur.

Right now the thought is to build a sample (prototype) email system as a starting point.

What about security breaches (intrusion detection)? Something similar can be done; this kind of technology would be good. You could roll back to before the intrusion, install the filter or preventative mechanism, and then roll the good stuff back in again.

Simply changing (fixing, simplifying, etc.) the interface is insufficient. Work does need to be done on SA recovery interfaces but this is beyond the scope of the ROC project.

Hot Tools in Use Today or Coming Soon

Next we discussed the new tools, technologies, ideas, or paradigm we're investigating or using. The list included new IP telephony products; tricks for SSH and CVS; wireless networking; integration and aggregation of alarm, monitoring, and administrative functionality with automation; reducing information replication; load balancing; anomaly detection; miniaturization; mirroring network storage for high-speed failover; VMware; MacOS X; Java; and Perl 6. The list also included business problems as opposed to technology problems.

There was a side discussion about programming languages. Some people like Java, others like C#. Java is the new COBOL in that it's the new business language but not a system language. Some debate ensued, with no conclusion, about whether to teach C, C++, Java, or even Scheme first.

Surprises from the Past Year

Several people mentioned surprises they'd had in the past year. This list includes Cygwin, the PC Weasel, the dearth of middlemen in the DSL/POP/ISP markets, and the number of people running wireless networks without any security.

SANE 2002

System Administration and Networking Conference

The meeting point for the Unix and Open Systems community

Tutorial days
May 27-29, 2002

Conference
May 30-31, 2002

With talks and tutorials presented by widely acclaimed speakers such as:

Mark Burgess, Joe Greco, Arthur Donkers, Kirk McKusick, Peter Salus, Lance Spitzner, Tobias Oetiker, Guido van Rooij, Brad Knowles, Alexios Zavras

Covering timely topics like:

Black Hats, DNSSEC, Firewalls, Network Management, Honeypots, Building large clusters, Web server, Encryption and Security techniques

WWW.SANE.NL

Tutorials - Conference - Social event - Terminal Room - Exhibition - BoF sessions - WiP sessions - And more!


Tutorials & Lectures


Time Machine


Exhibition


Internet Access Room


Social Event

Pictures: SANE 2000

CONNECT WITH USENIX & SAGE

MEMBERSHIP, PUBLICATIONS, AND CONFERENCES

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Phone: +1 510 528 8649
FAX: +1 510 548 5738
Email: <office@usenix.org>
<login@usenix.org>
<conference@usenix.org>

WEB SITES

<http://www.usenix.org>

<http://www.sage.org>

EMAIL

<login@usenix.org>

COMMENTS? SUGGESTIONS?

Send email to <ah@usenix.org>

CONTRIBUTIONS SOLICITED

You are encouraged to contribute articles, book reviews, photographs, cartoons, and announcements to *;login:*. Send them via email to *<login@usenix.org>* or through the postal system to the Association office.

The Association reserves the right to edit submitted material. Any reproduction of this magazine in part or in its entirety requires the permission of the Association and the author(s).

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

12

;login:

PERIODICALS POSTAGE
PAID
AT BERKELEY, CALIFORNIA
AND ADDITIONAL OFFICES

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

35

POSTMASTER
Send address changes to *;login:*
2560 Ninth Street, Suite 215
Berkeley, CA 94710