■ Explorer Post 369 ■

ExpNews

VOLUME 4, NUMBER 6, June 1997

DES Code Broken!

Calendar of Events:

July 25	ExpNews Deadline
July 28-Aug5 1997	BSA Jamboree
August 22-24	Campout
August 22	ExpNews Deadline
September 12-14	Canoeing Campout
September 26	ExpNews Deadline
October 24	ExpNews Deadline
October 24-26	Campout
November 28	ExpNews Deadline
December 12	ExpNews Deadline
December 30	Leave for Australia
January 2-10, 1998	Australian Jamboree

Don't Trust Your Firewall

Daniel J. Gregor Jr.

Here is some food for thought: Most computer security professionals agree that about 20% of computer security incidents are caused by someone outside the organization, with the remaining 80% being caused by someone inside the organization. Does your budget for computer security take into account these percentages? Do you spend one-fifth of this budget on firewalls and remote access security, and the rest on security within the company?

What DESCHALL Means

C. Matt Curtain http://www.research.megasoft.com/deschall/what.html

With all of the press that DESCHALL has gotten, lots of people are asking "what does this mean?" What follows is an explanation, written for the nontechnical reader. This is important for all of us, because regardless of whether you like it, information about you is protected with DES.

What's a cryptosystem?

Cryptosystems are locks for data. By using mathematical functions (called algorithms), the data is turned into gibberish that can only be returned to a form that makes sense by applying the appropriate key. It's easy to understand how this works by envisioning a bicycle tumbler lock. In a bicycle lock, there are a number of tumblers, each with numbers on it, from 0 to 9. Because computers are binary -- only working with 0s and 1s -- at their most basic level, a cryptosystem is like a bicycle lock that has only two numbers: 0 and 1.

On such a lock, there are two possible choices: 0 and 1. By putting another tumbler with 0 and 1 on it, the number of possible combinations increases exponentially from 2 to 4. (In computers, we use "bits" instead of tumblers, but for our purposes here, it means the same thing.)

Possible Combinations for Small Binary Locks

TABLE 1.

Number of Tumblers	*Number of Possible Combinations*	*Possible Combinations*
1	2	0, 1
2	4	00, 01, 10, 11
3	8	000, 001, 010, 011, 100, 101, 110, 111

The Explorer Code

As an Explorer-

I believe that American's Strength lies in her trust in God and in the courage and strength of her people

I will, therefore, be faithful in my religious duties and will maintain a personal sense of honor in my own life.

I will treasure my American heritage and will do all I can to preserve and enrich it.

I will recognize the dignity and worth of my fellowmen and will use fair play and goodwill in dealing with them.

I will acquire the exploring attitude that seeks the truth in all things and adventure on the frontiers of our changing world.

■

So, not only does the lock have to be strong -- that is, function as it was designed, without allowing a "shortcut" to solve the problem (such as the application of boltcutters) -- it has to have enough possible combinations to make someone trying every single one infeasible.

A cryptosystem with only a 3-bit keyspace would have 8 possible keys, as shown in the chart above, or as we see mathematically: 23 = 8. This would not be very secure, since to decrypt the message, I would try to decrypt it using the key 000, then 001, then 010, etc., up until 111, until I am looking at the contents of that message.

Let's catch up with our chart, by moving ahead to as many tumblers as what some cryptosystems are using today:

Possible Combinations for Larger Binary Locks

TABLE 2.

Number of Tumblers	*Number of Possible Combinations*
40	1,099,511,627,776
56	72,057,594,037,927,936
64	18,446,744,073,709,551,616
128	340,282,366,920,938,463,463,374,607,431,768,211,456

For a cryptosystem to be secure, there have to be enough possible keys to make it practically impossible for someone to (on a regular basis) try every possible key until they find the right one that works.

DES is a 56-bit cryptosystem, first pronounced a standard in 1977. At that time, it was infeasible for someone to try each of more than 72 quadrillion possibilities and find the contents of the encrypted file. At that time, it was safe. We've proven that this is no longer the case.

What We Did

We took a message that had been encrypted with DES, and read the contents of the message, as well as figured out which key was used to encrypt the message in the first place. At the same time, we now have more data that shows the kind of horsepower we can assemble by having a bunch of people run a little piece of software (called a client) on their computer that has no noticeable effect on their systems' performance.

It is noteworthy that this is the first time that a message encrypted with DES has been broken "in public." It's been widely believed that large governments' intelligence agencies (such as the NSA in the US) have been able to do this for quite a few years now.

That client took the encrypted message, and decrypted it using every possible key, until we found the one that resulted in some output that made sense.

We were trying about 7 billion keys per second at the time that the solution was found. If that computing power -- assembled using a bunch of PCs, Macs, workstations, and some servers -- was applied to the same cryptosystem of different key sizes, we can see how long it would take to decrypt a message by trying every possible key until the right one is found. This is the computer equivalent of turning the tumblers of a bicycle lock to

0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000

and pulling the chain to see if it unlocks. If not, we turn it to

0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000001

and try again. We do this until we find the right combination.

Which means that we tried 37,350,551,110,358,107 of 72,057,594,037,927,936 possible keys.

(Mathematicians, please forgive the following oversimplification. We can compensate for the extra work needed to compute RC5 by adding more computers to the project, so this comparison is possible.) Time to Crack Messages Encrypted with Various Key sizes

(On the average, at DESCHALL's final speed.)

TABLE 3.

Number of Tumblers	*Time to Crack the message*
40	78 seconds
48	5 hours
56	59 days
64	41 years
72	10,696 years
80	2,738,199 years
88	700,978,948 years
96	179,450,610,898 years
112	11,760,475,235,863,837 years
128	770,734,505,057,572,442,069 years

For comparison, Hawking[1] notes that the age of the universe is probably "only" 20,000,000,000 (or perhaps 10,000,000,000) years old.

So, while it's infeasible for DESCHALL to crack a 72 bit key, it seems that 64 might be within reach, by adding more machines. (We probably used between 15,000 and 20,000 machines.) Consider that the RC5-32/12/5 (40 bits) key crack took three and a half hours. The distributed computer we put together could do it in about 78 seconds.

The RC5-32/12/6 (48 bits) key crack took 13 days. A DESCHALL-sized effort could do it in 5 hours.

Given Moore's Law, which states that computing power will double every 18 months, and the fact that the brute-force searches done for RC5 and DES so far are written in software on general-purpose computers (a extremely slow method, by comparison to custom hardware), one can see how it's useful to get keys up over that 80 bits mark, especially if it's protecting data that has to be secret for any length of time. (US Census data, by law, must remain secret for 72 years[2].)

Imagine not only trying to figure out how much money it would take to build a machine to search a 96-bit keyspace today, but also in 99 years from now. Predicting what we'll be using next year is often tricky enough, much less 10 years, 40 years, or 100 years down the road. (See why we're paranoid?)

What's the Solution? (How Do We Address this Problem?)

It doesn't "cost" us much more, in terms of computer cycles to encrypt something with 128 bits, instead of 40 or 56. Yet, the level of security that we enjoy as a result of that extra step is amazing. We go from being able to trivially decrypt a message in seconds to requiring more time than the age of the universe many times over.

The solution, then, is a simple matter of replacing our DES "locks" with locks (cryptosystems) that use larger keys. Many such systems exist: Triple-DES, IDEA, Blowfish, and RC5, to name just a handful of the more well-known options. Data encrypted with DES must be re-encrypted using the new cryptosystems, and applications must be reprogrammed to stop using DES and begin using the new system for their encryption.

To be "safe," then, the data that's been encrypted should be worthless by the time someone is able to read it by using brute force. As an example, a credit card that will expire in a year or two from now might be OK to encrypt with a 64-bit cryptosystem. (Anyone who can raise enough money to build a machine that will crack 64-bit-encrypted messages would be able to find other means -- like bribing clerks -- to get the data he wants, so we'll only look at software-based attacks like DESCHALL.) However, it would be stupid to encrypt census data with a 64-bit algorithm, since DESCHALL could find the key in about 41 years (on the average). Factor in Moore's Law, and you're looking at something that can probably be read in a decade with little-to-no effort.

It's undesirable to change standards constantly, and be in a perpetual process of upgrading the cryptographic modules of our software. It's also unnecessary. Simply erring in favor of the paranoid and building systems NOW with keys that seem ridiculously long, and re-encrypting our small-key-encrypted data with these systems will adequately address the problem.

...at least until someone gets a quantum or DNA computer working. Then, all bets are off.:-)

References

1 S.W. Hawking, A Brief History of Time. p.108. Published in 1988, by Bantam.

2 U.S. Census Bureau. http://www.census.gov/genealogy/www/

Quote of the Month

Did you ever notice you are working harder than ever before so your employer can make a bigger income?

Web Stats

http://post369.columbus.oh.us

TABLE 4.

Pages	*Users*	*Button*
174	39	exploring
1025	239	ExpNews
890	94	links
3386	408	post
52	24	1997.calendar
1436	129	members
515	46	Adults
221	50	Toadies
318	46	Youth
369	55	program
20	12	project
204	47	scouting

When we installed "Access Watch" we reset the counters. Therefore, the statistics will not show much growth from the April issue of The ExpNews

Nobanner

D.J. Gregor

I just looked at how to setup the Solaris LP system so that a banner isn't printed. Just do the following:

Edit /etc/lp/interfaces/*<printer_name>*

Find the line `nobanner="no"`, and change the "no" to "yes". This will set the default to not print a banner.

If you want to REVERSE the '-o nobanner' option to lp(1) so that it prints a banner, go down a few lines in the specified file and change "yes" to "no" in this section of code:

```
case "${inlist}${i}" in
nobanner )
     nobanner="yes"
     #<Change "yes" to "no">
     ;;
```

The file /etc/lp/interfaces/*<printer_name>* is created whenever you make a printer, and it is derived from:

/usr/lib/lp/model/standard.

Up-an-Coming Post Expenses

12/01/97 Post Charter	$30.00
12/01/97 Post Insurance	$85.00
Monthly ExpNews	$75.00

Up-an-Coming Member Expenses

Registration 11/01/97	$15.00

Post Finances

Explorer Post 369 has	-$943.00
Floor Fund Need	$1,200.00
Floor Fund In Hand	$820.00
Pledges Outstanding FF,	$200.00
Room Fund Needed	$3,800.00
Room Fund	$0.00
Computer Fund Needed	$0.00
Computer Fund	$0.00

We are looking for a new sponsor for the ExpNews, Can you help?

Our Principals:

1) Honor before all else.
2) The difference between a winner and a looser is that the winner tried one more time.
3) K.I.S.M.I.F.

Our Creed:

Exploring: Enthusiasm, Energy, & Excellence.

Explorer Post 369:

Explorer Post 369 was chartered on December 31, 1994 to the Reformation Luthern Church.

Explorer Post 369 specializes in UNIX for Programmers while emphasizing a deep theme of Engineering Computer Information & Science

Membership in Explorer Post 369 is open to young men and women between the ages of 14 [and in high school] and not yet 20. Annual Membership fees are $15.00.

Our Web Page:

http://post369.columbus.oh.us

The views in this NewsLetter are strictly those of Explorer Post 369 and they do not necessarily represents the views or opinions of the Reformation Luthern Church or the Boy Scouts of America and/or the Simon Kenton Council.

Our E-Mail Addresses

Committee Member	
Herb Docken	Institutional Representative
Ralph Maurer(E)	Committee Chairman
Tom Niedzielski(E)	Committee Member
Steve Weller(E)	Committee Member
Adults Members:	
James D. Corder(E)	www.corder .com
Andy Drake	drake.73@osu.edu
Steve Potter	spp@psisa.com
Consultants:	
David J. Alden	
Honorary Members:	
Mark Bastian(Q)	mpb@icenet.com.au
Dan Jackson	
Lucas James(Q)	jj@ldjpc.apana.org.au
Alan Jones(Q)	alan@scoutnet.net.au
Sara Jones(Q)	sacubs@dove.net.au
Youth Members:	
DJ Gregor(E)	dgregor@gregor.com
Joe Harvey(E)	joharve4@mail.vt.edu
John Klapp(E)	klapp.2@osu.edu
Karl N. Matthias(E)	matthias.3@osu.edu
Jim Smith	smith.2407@osu.edu
Mike Turner	turner.319@osu.edu
Post-Toadies:	
Chris Gauger(1st)	Toady
Matt Groce(1st)	Toady
Allan Hamilton(S)	Page

(E)	Eagle Scout
(Q)	Queen Scout

BOY SCOUTS OF AMERICA

Explorer Post 369
P.O. Box 307218
Gahanna, Ohio 43230
United States of America