

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[] =
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x7f"
"\x90\x03\x00\x20"
"\x92\x00\x20\x10"
"\xc0\x22\x20\x08"
"\xd0\x22\x20\x10"
"\xc0\x22\x20\x10"
"\x82\x10\x20\x08"
"\x91\x00\x20\x08"
"/bin/ksh" ;
char jump[] =
"\x90\x10\x00\x0e"
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

**CODITO, ERGO SUM**

got root?

## Know your Enemy

An introduction to the mind of a hacker

By Jonathan Levin  
([JL@hisown.com](mailto:JL@hisown.com))

CompuTAX 2(K+1)

Jonathan Levin  $\pi$  יונתן לוין  
Training & Consulting עוץ והדרכה

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[] =
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x7f"
"\x90\x03\x00\x20"
"\x92\x00\x20\x10"
"\xc0\x22\x20\x08"
"\xd0\x22\x20\x10"
"\xc0\x22\x20\x10"
"\x82\x10\x20\x08"
"\x91\x00\x20\x08"
"/bin/ksh" ;
char jump[] =
"\x90\x10\x00\x0e"
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

## Dramatis Personae



- ◆ “Hackers” - classified into three categories:
  - “Script Kiddies”:
    - ◆ Making up 80%. Potential: Low, Damage: High
  - “Amateur/Experts”:
    - ◆ ~20%. Potential: Considerable, Damage: Low
  - “Crack Experts”:
    - ◆ Under 1%. Potential: High, Damage: High

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
```

## Dramatis Personae

```
char shellcode[...]=
  "\x20\xbf\xff\xff"
  "\x20\xbf\xff\xff"
  "\x7f\xff\xff\xff"
  "\x90\x03"
  "\x92\x03"
  "\xc0\x22"
  "\xd0\x22"
  "\xc0\x22"
  "\x82\x14"
  "\x91\xd0"
  "/bin/ksh"
char jump[...]=

static char nop[...]= "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

- ◆ **Alignment classifications:**
  - **“White Hats”** – The “Chaotic Good” hackers, security experts and non-harmful types.
  - **“Black Hats”** – The “Chaotic Evil” hackers – malevolent, damage inflicting
  - **“Grey Hats”** – The “Chaotic Neutral” hackers – may be white or black, depending on their morning mood.

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
```

## The Modus Operandi

```
char shellcode[...]=
  "\x20\xbf\xff\xff"
  "\x20\xbf\xff\xff"
  "\x7f\xff\xff\xff"
  "\x90\x03"
  "\x92\x03"
  "\xc0\x22"
  "\xd0\x22"
  "\xc0\x22"
  "\x82\x14"
  "\x91\xd0"
  "/bin/ksh"
char jump[...]= "\x81\xc3\xe0\x08"

static char nop[...]= "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

- **Phase I: Reconnaissance/Intelligence**  
Gather the basic information required.
- **Phase II: Infiltration:**  
Social engineering, physical infiltration, exploits or trojans.
- **Phase III: Expansion**  
Strengthen hold, probe, and infiltrate more systems. By sniffing, keystroke logging, or using mass exploits.

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADENUM 600
#define ...
```

## The Modus Operandi

The hacker "algorithm" – Phase I

I) Target acquisition:

In this phase, the hacker chooses the target for penetration. This may either be done:

- Randomly: as a result of a network scan, or by chance password retrieval
- Directed: the hacker chooses the target due to political or other reasons.

```
char shellcode[] = ...
char jump[] = ...
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADENUM 600
#define ...
```

## The Modus Operandi

The hacker "algorithm" – Phase I

II) OS Identification:

The hacker attempts to identify the target operating system and service portfolio, by using known protocols (e.g. HTTP, FTP, SMTP)

At this stage, the hacker may also resort to port scanning, route tracing, or using NMap style tools.

```
char shellcode[] = ...
char jump[] = ...
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

## The Modus Operandi

The hacker "algorithm" – Phase I

### II) Intelligence:

Using publicly available protocols (e.g. DNS, ICMP) the hacker discovers the local "neighborhood" of the target, which may serve in finding footholds for a more effective attack.

For each discovered neighbor, iteratively execute step II.

```

#define NOPNUM 256+16
#define ADPNUM 600
#define ...

char sherlocode[] =
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x7f\xff\xff\xff"
"\x90\x03"
"\x92\x02"
"\xc0\x22"
"\xd0\x22"
"\xc0\x22"
"\x82\x10"
"\x91\xd0"
"/bin/ksh"

char ...
static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {

```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

## The Modus Operandi

The hacker "algorithm" – Phase I

At this stage, the hacker has constructed a network topology map, and a matrix of the potential targets for attack:

| IP Addr.   | DNS Name | OS        | Version    | FTP        | SMTP           | HTTP                   | RPC/NetBIOS        | POP/IMAP      | Other        |
|------------|----------|-----------|------------|------------|----------------|------------------------|--------------------|---------------|--------------|
| 10.0.0.24  | Dgre     | Solaris   | 2.5.1      | None (FW?) | None (FW?)     | None (FW?)             | TTDB, NFS, sadmind | OPopper 2.5.3 | NFS Mounts   |
| 10.0.0.25  | Goblin   | Linux     | RedHat 6.2 | WU 2.6.0   | SendMail 8.9.3 | Apache 1.3.9/PHP 4.0.5 | None (FW?)         | None (FW?)    |              |
| 10.0.0.199 | Drc      | Win/NT    | 5.0        | None (FW?) | Exchange 5.0   | IIS/5.0 SP1(?)         | YES!               | None (FW?)    |              |
| 10.0.0.254 | Troll    | Cisco IOS | 11.2       | N/A        | N/A            | N/A                    | N/A                | N/A           | Telnet, SNMP |

```

#define NOPNUM 256+16
#define ADPNUM 600
#define ...

char sherlocode[] =
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x7f\xff\xff\xff"
"\x90\x03"
"\x92\x02"
"\xc0\x22"
"\xd0\x22"
"\xc0\x22"
"\x82\x10"
"\x91\xd0"
"/bin/ksh"

char ...
static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {

```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADNUM 600
#define ...
```

## The Modus Operandi

The hacker "algorithm" – Phase II

**IV) The Attack:**  
Consult network matrix to find vulnerable spots. Seek anonymous or normal user access by:  
- username/passwd attacks  
- exploits vs. OS and Services

```
char shellcode[] = "\x20\xbf\xff\xff" ...
char jump[] = "\x81\xc3\xe0\x08" ...
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADNUM 600
#define ...
```

## The Modus Operandi

The hacker "algorithm" – Phase II

**V) The Attack – advanced stages**  
After obtaining the primary foothold, go for the prize – root or admin privileges.  
This is usually done by exploiting system misconfiguration, or known vulnerabilities.

Note, that usually steps IV and V may be combined, to instantly gain admin/root

```
char shellcode[] = "\x20\xbf\xff\xff" ...
char jump[] = "\x81\xc3\xe0\x08" ...
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADNUM 600
#define ...
```

## The Modus Operandi

The hacker "algorithm" – Phase III

**VI) Cover your tracks**  
By surgically editing system logs (event log, syslog, utmp, wtmp, etc.), remove all evidence of the hack.  
(of course, an `rm -fR /` is equally effective..)

If you feel especially mischievous, plant false tracks to lead sysadmin on wild goose chase...

```
char shellcode[] =
"\x20\xbf\xff\xff" /* jmp     $shellcode-4 */
"\x7f\xff\xff\xff" /* call   $shellcode+4 */
"\x92\x02\x00\x00" /* mov     $0, %eax */
"\xc0\x22\x20\x10" /* st     %eax, %eax */
"\xd0\x22\x20\x10" /* st     %eax, %eax */
"\xc0\x22\x20\x14" /* st     %eax, %eax */
"\x91\xd0\x20\x08" /* sa     %eax, %eax */
"/bin/ksh" ;

char jump["\x90\x10\x00\x0e" /* jmp     $+10 */
"\x90\x10\x00\x0e" /* jmp     $+10 */

static char nop["\x90\x1c\x40\x11"];

main(int argc, char **argv) {
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADNUM 600
#define ...
```

## The Modus Operandi

The hacker "algorithm" – Phase III

**VII) Plant backdoors**  
Today's vulnerability may be fixed tomorrow, leaving you out of the system. (we can't have that, can we?)

Recompile binaries/executables.  
Add/Change users & preferences

```
char jump["\x90\x10\x00\x0e" /* jmp     $+10 */
"\x90\x10\x00\x0e" /* jmp     $+10 */

static char nop["\x90\x1c\x40\x11"];

main(int argc, char **argv) {
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADRNUM 600
#define ALIGN 3

char shellcode[] =
"\x20\xbf\xff\xff" /* jmp $ -shellcode-4 */
"\x20\xff\xff\xff" /* jmp $ -shellcode */
"\x7f\xff\xff\xff" /* call -shellcode+4 */
"\x90\x03\xe0\x20" /* jmp $ -0x10 */
"\xc0\x22\x20\x08" /* or $,0x08 */
"\xd0\x22\x20\x10" /* or $,0x10 */
"\x82\x10\x20\x0b" /* mov $,0xb */
"\x91\xd0\x20\x08" /* sa $ */
"/bin/"

char jump[] = "\x81\xc3\xe0\x08" /* jmp $+0x8 */
              "\x90\x10\x00\x0e" /* mov $,0x0e */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

## The Modus Operandi

The hacker "algorithm" – Phase III

VIII) Expand  
Plant sniffers/keystroke loggers in system,  
to intercept valid user transactions.

By doing so, you are likely to  
recursively obtain access to other  
targets, dropping you in step IV!

```
#define NOPNUM 256+16
#define ADRNUM 600
#define ALIGN 3

char shellcode[] =
"\x20\xbf\xff\xff" /* jmp $ -shellcode-4 */
"\x20\xff\xff\xff" /* jmp $ -shellcode */
"\x7f\xff\xff\xff" /* call -shellcode+4 */
"\x90\x03\xe0\x20" /* jmp $ -0x10 */
"\xc0\x22\x20\x08" /* or $,0x08 */
"\xd0\x22\x20\x10" /* or $,0x10 */
"\x82\x10\x20\x0b" /* mov $,0xb */
"\x91\xd0\x20\x08" /* sa $ */
"/bin/"

char jump[] = "\x81\xc3\xe0\x08" /* jmp $+0x8 */
              "\x90\x10\x00\x0e" /* mov $,0x0e */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

## Tools of the Trade

- ◆ Social Engineering
- ◆ Virii & Worms
- ◆ Trojan Horses
- ◆ Exploits
- ◆ Physical Means

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[]=
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x77"
"\x90"
"\x92\x02\x20"
"\xc0"
"\xd0\x22\x20"
"\xc0"
"\x82"
"\x91"
"/bin/k"
char jump[] = "\x81\xc3\xe0\x08"
"\x90\x10\x00\x0e"
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

## Tools of the Trade - I

**Social Engineering** – the weakest link is the human one.

- Humans may be fooled.
- Email may be spoofed (and IS!)
- IDs can be faked. (technicians, consultants..)
- Phone calls can be impersonated...

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[]=
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x77"
"\x90"
"\x92\x02\x20"
"\xc0"
"\xd0\x22\x20"
"\xc0"
"\x82"
"\x91"
"/bin/k"
char jump[] = "\x81\xc3\xe0\x08"
"\x90\x10\x00\x0e"
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

## Tools of the Trade - I

**Example** – Spoofing Email

```
Johnny@Ogre (~)> telnet mailserver.of.some.com 25
220 Herald ESMTP SendMail 8.8.8 15:57:20 Oct 24, 2002 Ready
MAIL FROM: God@Heaven.Org
200 OK
RCPT TO: unsuspecting.user
200 OK
DATA
354 Enter Mail Data. End by <CR><CR>, or with '.' on a line by itself
From: The Almighty One <God@Heaven.org>
To: You <unsuspecting.user@some.com>
Subject: Repent, Sinner!
.
220 Message Accepted for delivery – xJrCCyL532
```

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADRNUM 600
#define ALIGN 3
char shellcode[] =
  "\x20\xbf\xff\xff"
  "\x20\xbf\xff\xff"
  "\x7f\xff\xff\xff"
  "\x90\x90\x90\x90"
  "\x92\x02\x20\x10"
  "\xc0\x22\x20\x10"
  "\xc0\x22\x20\x14"
  "\x82\x10\x20\x0b"
  "\x91\xff\xff\xff"
  "/bin/k"
char jump[] =
  "\x90\x10\x0b\x0e"

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

## Tools of the Trade - II

**Virii** – Self executing code. This code may execute on the fly, and may polymorph at will. Nowadays, losing ground to worms.

**Worms** – Piggyback code, that needs a “host” to execute. LoveBug (VBA), Sircam (IIS/Solaris), CodeRed (IIS), and nimDA (IIS/Explorer/Outlook)

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADRNUM 600
#define ALIGN 3
char shellcode[] =
  "\x20\xbf\xff\xff"
  "\x7f\xff\xff\xff"
  "\x90\x90\x90\x90"
  "\x92\x02\x20\x10"
  "\xc0\x22\x20\x10"
  "\xc0\x22\x20\x14"
  "\x82\x10\x20\x0b"
  "\x91\xff\xff\xff"
  "/bin/k"
char jump[] =
  "\x90\x10\x0b\x0e"

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

## Tools of the Trade - III

**Trojan Horses** – “Timeo danaos et dona gerentes..” ANY piece of software that runs on your systems is potentially a trojan. It may remain dormant, until a certain condition is met. It then activates, as a “logic bomb”.

**WHILE DORMANT, TROJANS ARE VIRTUALLY UNDETECTABLE!**

The “Aum Shinryko” is suspected to have used trojan horses. And maybe Al-Qaeda...?

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com



```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
```

## A stitch in time saves nine

Fortunately, the average hacker may be repelled relatively easily.

The following slides demonstrate but a few of the precautions that can save you a LOT of headache, if employed.

The list is by no means comprehensive, but is definitely a good start.

```
char shellcode[] =
"\x20\xbf\xff\xff" /* no. 1  shellcode-4 */
"\x20\xbf\xff\xff" /* no. 2  shellcode-4 */
"\x90\x03\xe0\x21" /* no. 3  shellcode-4 */
"\x92\x00\x30\x10" /* no. 4  shellcode-4 */
"\xc0\x22\x20\x08" /* no. 5  shellcode-4 */
"\xd0\x22\x20\x10" /* no. 6  shellcode-4 */
"\x82\x10\x20\x08" /* no. 7  shellcode-4 */
"\x91\xd0\x20\x08" /* no. 8  shellcode-4 */
"/bin/ksh"

char jump[] = "\x81\xc3\xe0\x08" /* jmp  offset */
"\x90\x10\x00\x0e" /* no. 9  shellcode-4 */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
CompuTAX - Know Your Enemy - Lecture Notes by JL@HisOwn.com
```

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
```

## Precautions

As a rule of thumb – any service may serve as a foothold for a hacker. Thus:

- Block **ALL** unnecessary services, including:
- SNMP (UDP 161)
- Telnet/Rlogin/etc on UNIX (23,513...)
- NetBIOS/RPC/DCOM on Windows (139!)

**UN\*X RPC**

```
char jump[] = "\x90\x10\x00\x0e" /* no. 9  shellcode-4 */
"\x90\x10\x00\x0e" /* no. 10 shellcode-4 */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
CompuTAX - Know Your Enemy - Lecture Notes by JL@HisOwn.com
```

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
```

## Precautions

```
char shellcode[] =
  "\x20\xbf\xff\xff" /* jmp esp; shellcode-4 */
  "\x30\xbf\xff\xff" /* jmp esp; shellcode */
  "\x90\x03\xe0\x20" /* jmp esp; nops */
  "\x92\x02\x20\x10" /* jmp esp; nops */
  "\xc0\x22\x20\x08" /* ar; esp; nops */
  "\xd0\x22\x20\x10" /* ar; esp; nops */
  "\xc0\x22\x20\x14" /* ar; esp; nops */
  "\x82\x10\x00" /* jmp esp; nops */
  "\x91\xd0\x20\x08" /* jmp esp; nops */
  "/bin/ksh" /* shellcode */

char jump[] = "\x81\xc3\xe0\x08" /* jmp esp; nops */
              "\x90\x10\x00\x0e" /* jmp esp; nops */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

Secure the existing protocols you DO need:

- Block DNS zone Xfers – (TCP 53)
- Make sure FTP is secured (TCP 21,20)
- Check HTTP server and content (TCP 80)
- Carefully filter ICMP
- Install antivirus SMTP gateway (TCP 25)
- Secure file sharing & NFS

And Strip version headers from all services!

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
```

## Precautions

```
char shellcode[] =
  "\x20\xbf\xff\xff" /* jmp esp; shellcode-4 */
  "\x30\xbf\xff\xff" /* jmp esp; shellcode */
  "\x90\x03\xe0\x20" /* jmp esp; nops */
  "\x92\x02\x20\x10" /* jmp esp; nops */
  "\xc0\x22\x20\x08" /* ar; esp; nops */
  "\xd0\x22\x20\x10" /* ar; esp; nops */
  "\xc0\x22\x20\x14" /* ar; esp; nops */
  "\x82\x10\x00" /* jmp esp; nops */
  "\x91\xd0\x20\x08" /* jmp esp; nops */
  "/bin/ksh" /* shellcode */

char jump[] = "\x81\xc3\xe0\x08" /* jmp esp; nops */
              "\x90\x10\x00\x0e" /* jmp esp; nops */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

Verify system integrity:

- Stamp all executables using MD5
- Periodically check for new files
- Run periodic Anti-Virus checks

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[]=
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x90\x03\xe0\x20"
"\x92\x02\x20\x10"
"\xe0\x22\x20\x08"
"\xd0\x22\x20\x10"
"\xc0\x22\x20\x10"
"\x82\x10\x20\x0b"
"\x91\xcd"
"/bin/ks" ;
char jump[] = "\x81\xc3\xe0\x08"
"\x90\x10\x00\x0e"
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

## Precautions

- Never underestimate the value of logs!
- Backup logs on a daily basis
- GO OVER the logs before backing them up..
- If present, use enhanced auditing
- Consider setting a dedicated loghost

CompuTAX - Know Your Enemy - Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[]=
"\x20\xbf\xff\xff"
"\x20\xbf\xff\xff"
"\x90\x03\xe0\x20"
"\x92\x02\x20\x10"
"\xe0\x22\x20\x08"
"\xd0\x22\x20\x10"
"\xc0\x22\x20\x10"
"\x82\x10\x20\x0b"
"\x91\xcd"
"/bin/ks" ;
char jump[] = "\x81\xc3\xe0\x08"
"\x90\x10\x00\x0e"
static char nop[] = "\x90\x1c\x40\x11";
main(int argc, char **argv) {
```

## Precautions

- Don't rest on your laurels and be fooled into a false sense of security!
- Firewalls are merely the perimeter defense
- Perform periodic audit checks
- Consider external consultants

CompuTAX - Know Your Enemy - Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[]=
  "\x20\xbf\xff\xff"
  "\x20\xbf\xff\xff"
  "\x7f\xff\xff\xff"
  "\x90\x03\xe0\x20"
  "\x92\x02\x20\x10"
  "\xc0\x22\x20\x10"
  "\xc0\x22\x20\x14"
  "\x82\x11\x20\x10"
  "\x91\xd0\x20\x08"
  "/bin/ksh" ;

char jump[] = "\x90\x10\x00\x0e"

static char nop[]="\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

**Precautions**

**“Si vis pacem, Para Bellum”**

**Take the offensive – proactive security:**

- Use Short (or some other IDS) to monitor traffic and react in real time

**Consider setting up a honeypot.**

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
char shellcode[]=
  "\x20\xbf\xff\xff"
  "\x20\xbf\xff\xff"
  "\x7f\xff\xff\xff"
  "\x90\x03\xe0\x20"
  "\x92\x02\x20\x10"
  "\xc0\x22\x20\x10"
  "\xc0\x22\x20\x14"
  "\x82\x11\x20\x10"
  "\x91\xd0\x20\x08"
  "/bin/ksh" ;

char jump[] = "\x81\xc3\xe0\x08"

static char nop[]="\x90\x1c\x40\x11";

main(int argc, char **argv) {
```

**Precautions**

**Remain updated!**

- Subscribe to security mailing lists
- **INSTALL PATCHES,**
- **INSTALL PATCHES,**
- **INSTALL PATCHES, and**
- **INSTALL MORE PATCHES!**

**“The price of liberty is eternal vigilance”**

**(Wendell Phillips)**

CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
Web Resources
char shellcode[]=
"\x20\xbf\xff\xff" /* jmp     $shellcode-4 */
"\x20\xbf\xff\xff" /* jmp     $shellcode */
www.packetstormsecurity.com:
  Excellent resource for hackers & security experts alike
"\x92\x02\x20\x10" /* mov     $0,10 */
"\xe0\x22\x20\x08" /* or      $0,(10) */
www.sys-security.com:
  Ofir Arkin's EXCELLENT paper on ICMP
"\x82\x10\x20\x0b" /* mov     $0,10 */
"\x91\xe0\x20\x08" /* or      $0,10 */
www.insecure.org:
  Fyodor's NMAP.
char jump[] = "\x81\xc3\xe0\x08" /* jmp     $+10 */
              "\x90\x10\x00\x0e" /* mov     $0,(10) */

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv) {
CompuTAX - Know Your Enemy - Lecture Notes by JL@HisOwn.com
```

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALLIGN 3
Web Resources
char shellcode[]=
"\x20\xbf\xff\xff" /* jmp     $shellcode-4 */
"\x20\xbf\xff\xff" /* jmp     $shellcode */
www.securityfocus.com:
  BugTRAQ archive, vulnerability archive
"\x92\x02\x20\x10" /* mov     $0,10 */
"\xe0\x22\x20\x08" /* or      $0,(10) */
www.sans.org:
  "Security Alert Consensus" mailing list.
"\x82\x10\x20\x0b" /* mov     $0,10 */
"\x91\xe0\x20\x08" /* or      $0,10 */
www.attrition.org/mirror:
  Defaced web-pages mirror.
char jump[] = "\x81\xc3\xe0\x08" /* jmp     $+10 */
              "\x90\x10\x00\x0e" /* mov     $0,(10) */
www.honeynet.org:
  The "HoneyNet" Project - attempting to lure hackers and analyze their modus operandi
static char shellcode[] =
"\x20\xbf\xff\xff" /* jmp     $shellcode-4 */
"\x20\xbf\xff\xff" /* jmp     $shellcode */

main(int argc, char **argv) {
CompuTAX - Know Your Enemy - Lecture Notes by JL@HisOwn.com
```

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALIGN 3

char shellcode[]=
"\x20\xbf\xff\xff"  /* jmp     $shellcode+4 */
"\x20\xbf\xff\xff"  /* jmp     $shellcode+4 */
"“Know your Enemy”:"
"By the HoneyNet project"
"“Secrets & Lies”:"
"By Bruce Schneier"
"\x82\x10\x20\x0b"  /* mov     $0x20, %bl */
"\x91\xcd\x20\x08"  /* mov     $0, %eax
"/bin/ksh" ;

char jump[] = "\x81\xc3\xe0\x08" /* jmp     $+10 */
              "\x90\x10\x00\x0e" /* mov     $0, %eax

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv){
CompuTAX – Know Your Enemy – Lecture Notes by JL@HisOwn.com
```

```
#define NOPNUM 256+16
#define ADPNUM 600
#define ALIGN 3

char shellcode[]=
"\x20\xbf\xff\xff"  /* jmp     $shellcode+4 */
"\x20\xbf\xff\xff"  /* jmp     $shellcode+4 */
"\x7f\xff\xff\xff"  /* call    $shellcode+4 */
"\x90\x03\xe0\x20"  /* jmp     $+20 */
"\x9c\x22\x20\x08"  /* mov     $0x22, %eax
"\xcd"
"\xc1"
"\x8"
"\x9"
"/bi

char jump[] = "\x81\xc3\xe0\x08" /* jmp     $+10 */
              "\x90\x10\x00\x0e" /* mov     $0, %eax

static char nop[] = "\x90\x1c\x40\x11";

main(int argc, char **argv){
```

Questions? Comments?

Email: [JL@hisown.com](mailto:JL@hisown.com)

Web Site:  
<http://hisown.com/computax>

Jonathan Levin  $\pi$  יונתן לוין  
Training & Consulting עזר והדרכה