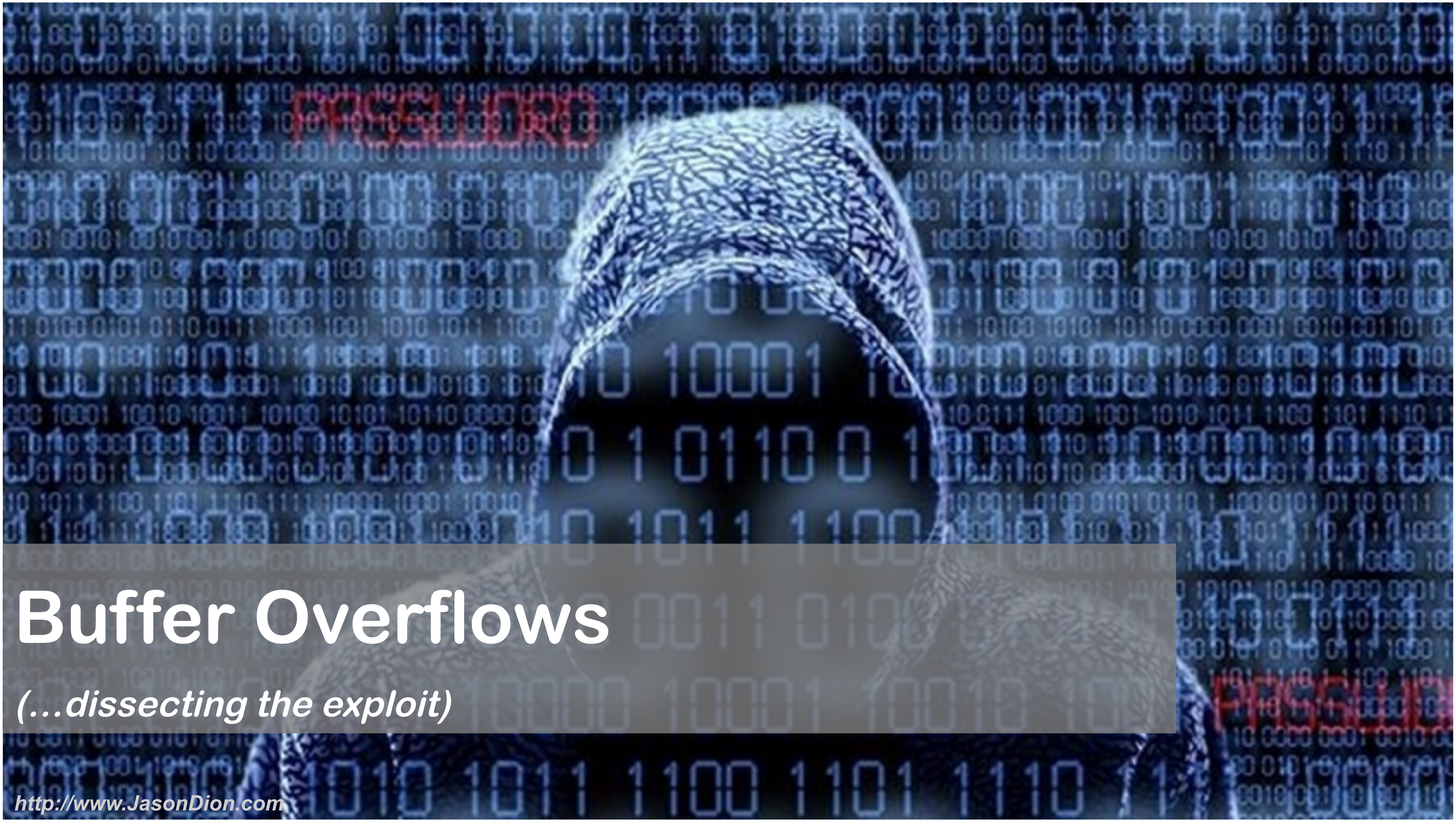
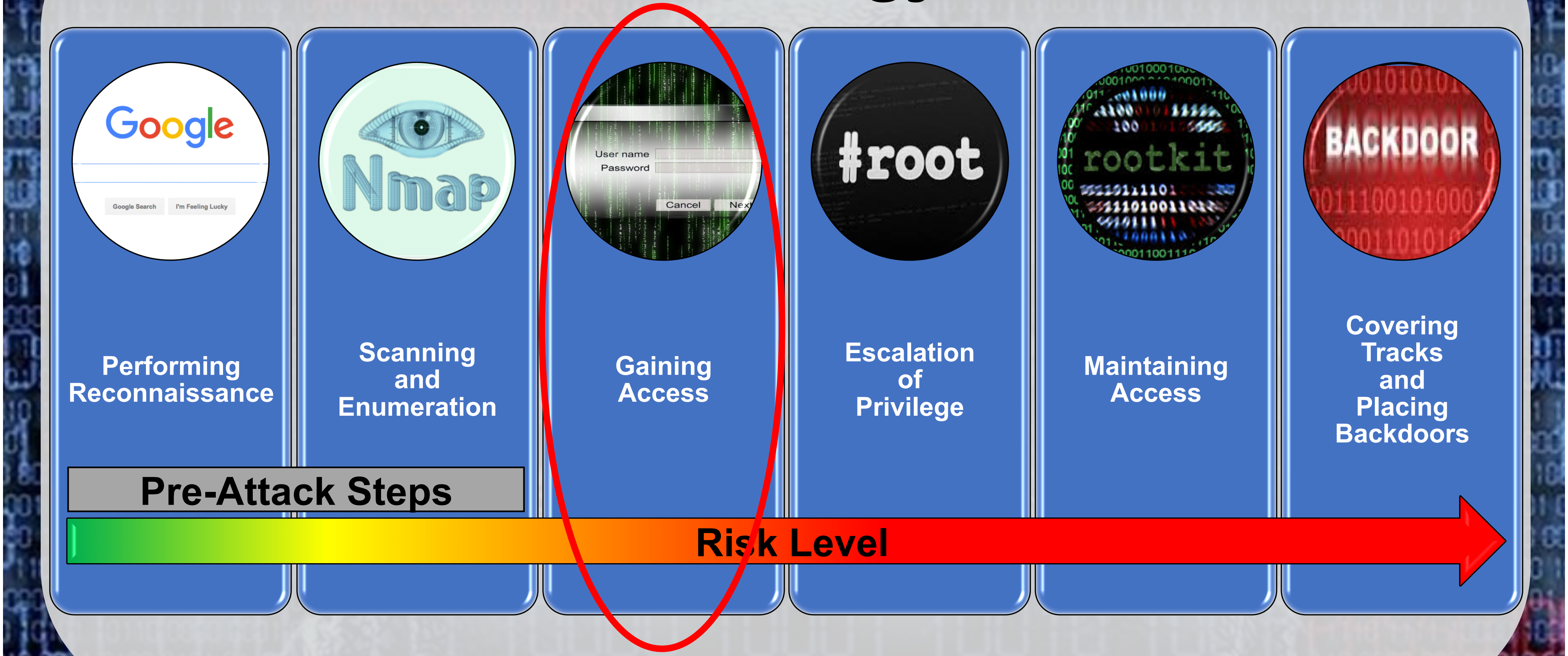


Buffer Overflows

(…dissecting the exploit)

http://www.JasonDion.com

Attacker’s Methodology



Performing Reconnaissance

Scanning and Enumeration

Gaining Access

Escalation of Privilege

Maintaining Access

Covering Tracks and Placing Backdoors

Pre-Attack Steps

Risk Level

What is a Buffer?

- A temporary storage area the program uses to store data

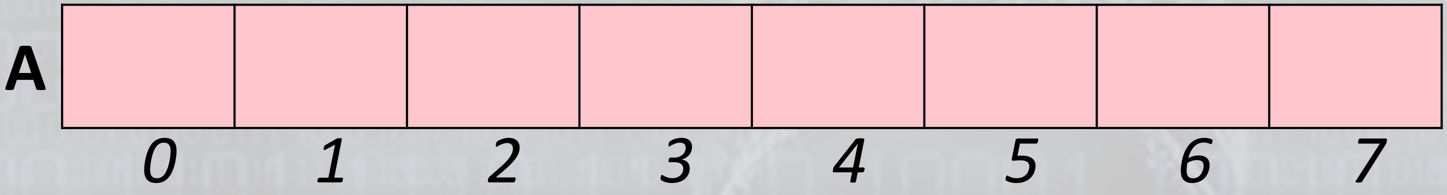
Buffer Usage

☎

Phone: 555-1234

Buffer Usage

Example of an 8-bit Buffer (A)



A

0 1 2 3 4 5 6 7

Phone: 555-1234

Buffer Usage

Example of an 8-bit Buffer (A)

A	5	5	5	-	1	2	3	4
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*

Phone: 555-1234

What is a Buffer Overflow?

- When a program puts more data into a buffer than the buffer can hold

Buffer Usage

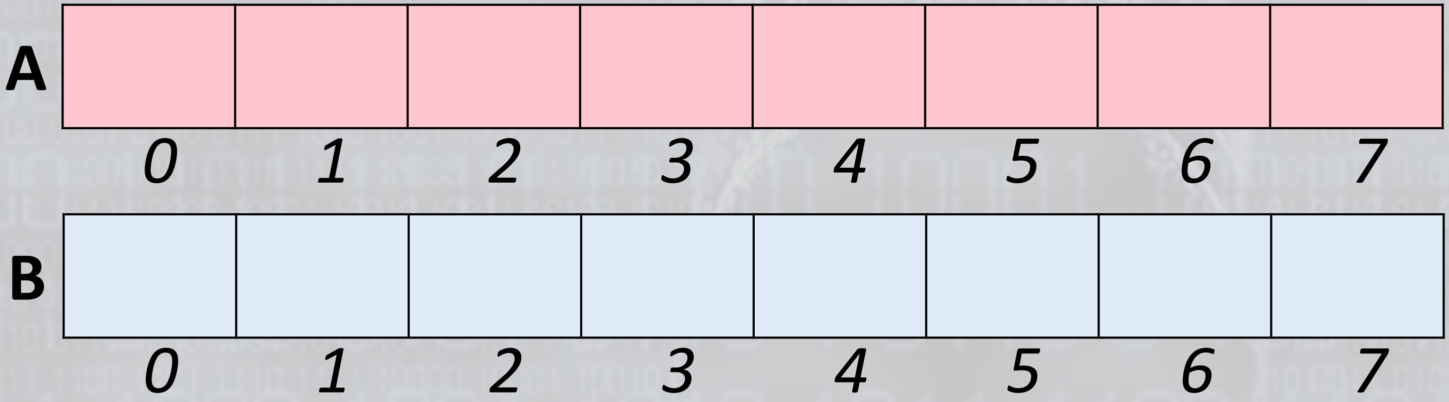
Example of an 8-bit Buffer (A)

A	**5**	**5**	**5**	**-**	**1**	**2**	**3**	**4**
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
B								
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*

Phone: 555-1234

Buffer Overflow

Example of an 8-bit Buffer (A)



A

0 1 2 3 4 5 6 7

B

0 1 2 3 4 5 6 7

Phone: 555-1234

210-555-1234

Buffer Overflow

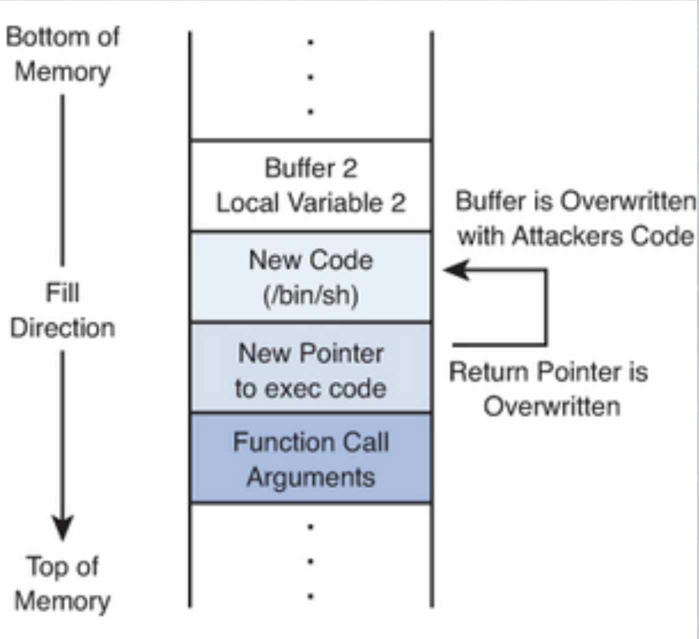
Example of an 8-bit Buffer (A)

A	**2**	**1**	**0**	**-**	**5**	**5**	**5**	**-**
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
B	**1**	**2**	**3**	**4**				
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*

Phone: 555-1234

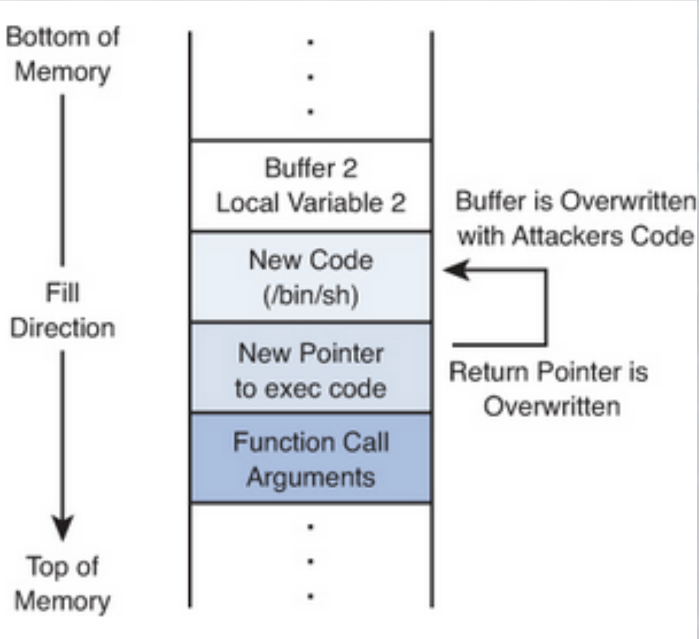
210-555-1234

How does the exploit work?



- Stack is a reserved area of memory where the program saves the return address when a call instruction is received

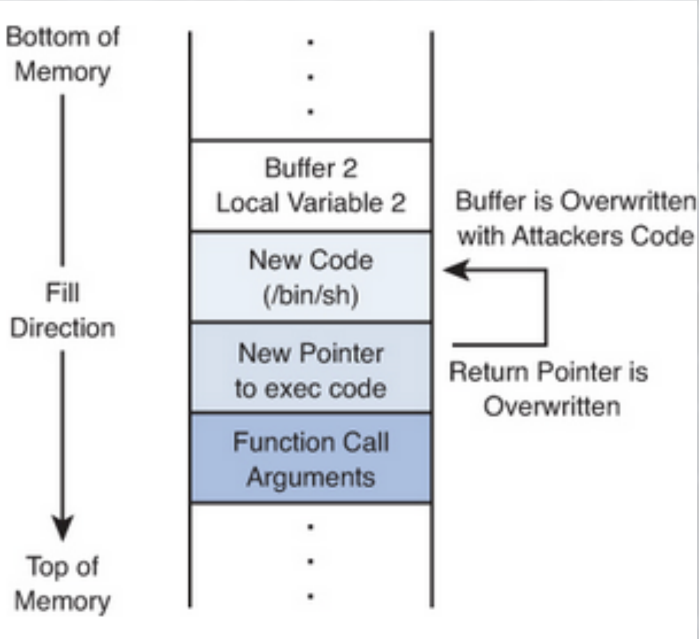
How does the exploit work?



- Stack is organized in FILO structure

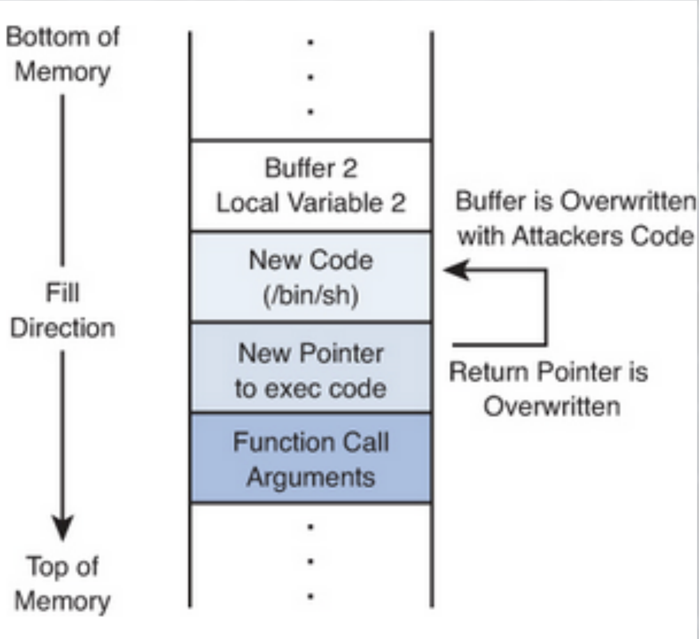
- First thing placed in the stack is the last thing removed

How does the exploit work?



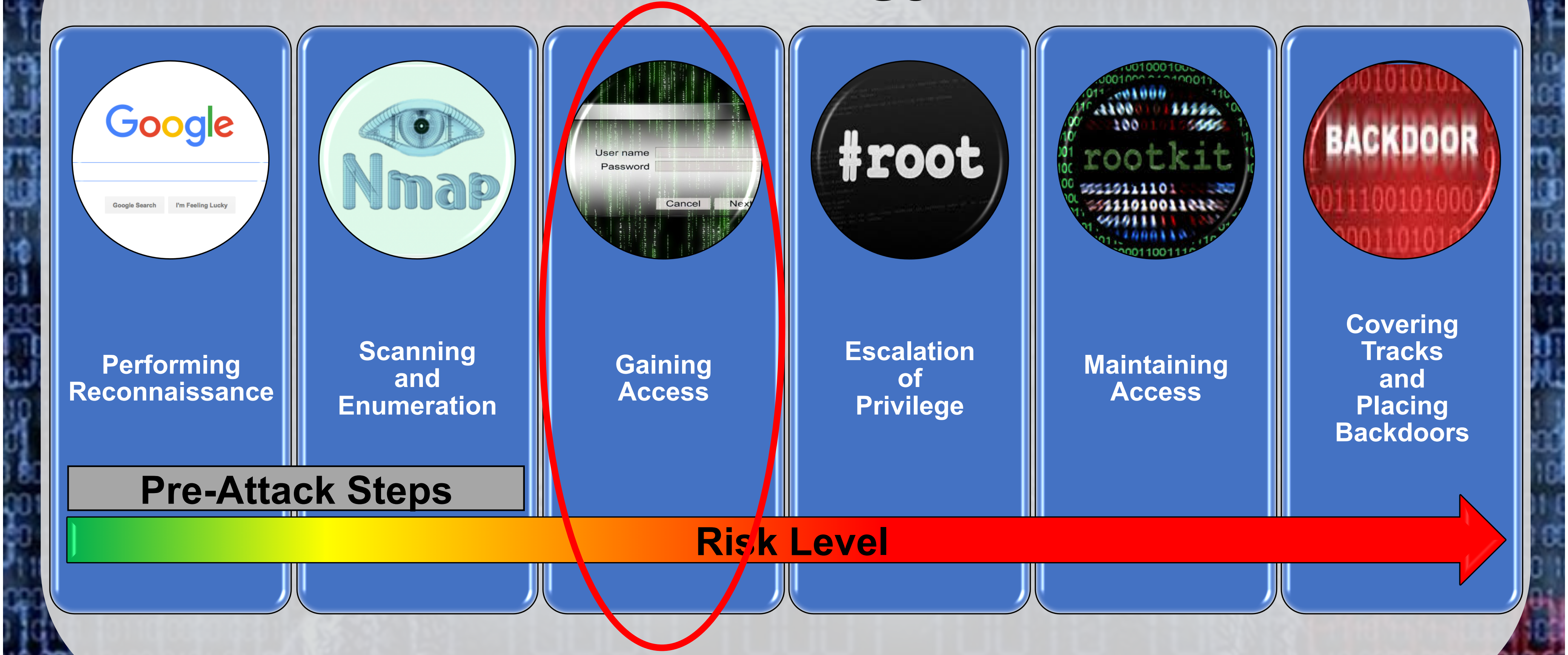
- Attacker can place too much information on the stack or change the value of the return pointer to carry out the attack

How does the exploit work?



- Attacker's code is placed in the buffer
- Code could be used to run commands or execute a series of instructions

Attacker’s Methodology



Performing Reconnaissance

Scanning and Enumeration

Gaining Access

Escalation of Privilege

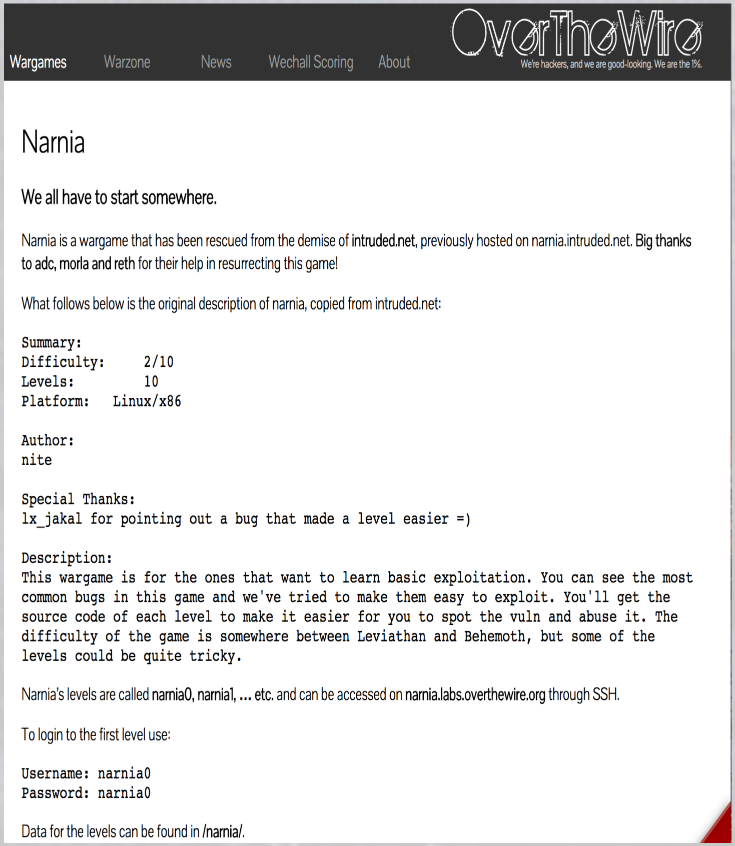
Maintaining Access

Covering Tracks and Placing Backdoors

Pre-Attack Steps

Risk Level

Practice: Over the Wire



Narnia

We all have to start somewhere.

Narnia is a wargame that has been rescued from the demise of **intruded.net**, previously hosted on narnia.intruded.net. **Big thanks to adc, morla and reth** for their help in resurrecting this game!

What follows below is the original description of narnia, copied from intruded.net:

```
Summary:
Difficulty:     2/10
Levels:         10
Platform:   Linux/x86

Author:
nite

Special Thanks:
lx_jakal for pointing out a bug that made a level easier =)

Description:
This wargame is for the ones that want to learn basic exploitation. You can see the most
common bugs in this game and we've tried to make them easy to exploit. You'll get the
source code of each level to make it easier for you to spot the vuln and abuse it. The
difficulty of the game is somewhere between Leviathan and Behemoth, but some of the
levels could be quite tricky.
```

Narnia's levels are called **narnia0, narnia1, ... etc.** and can be accessed on **narnia.labs.overthewire.org** through SSH.

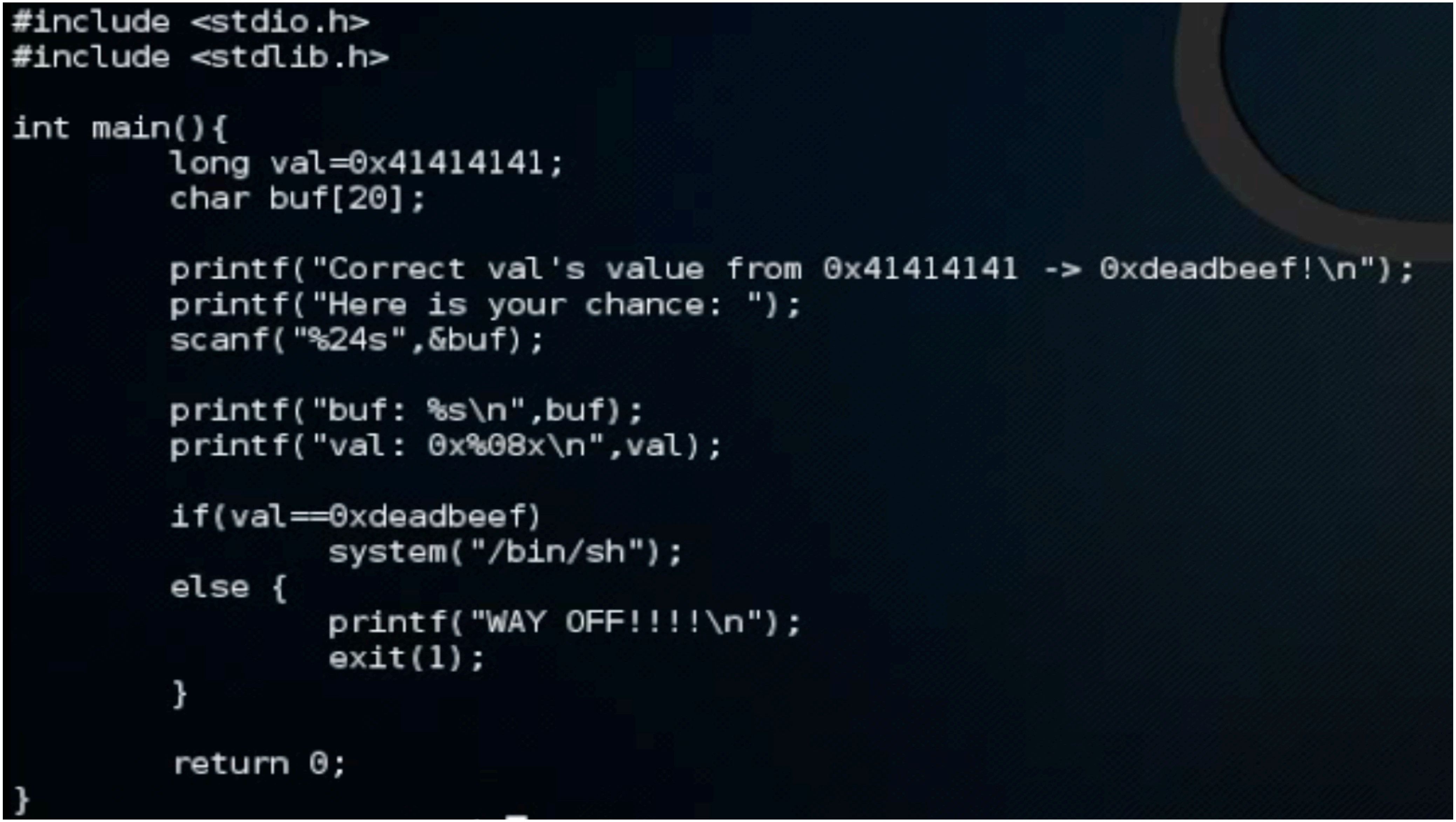
To login to the first level use:

```
Username: narnia0
Password: narnia0
```

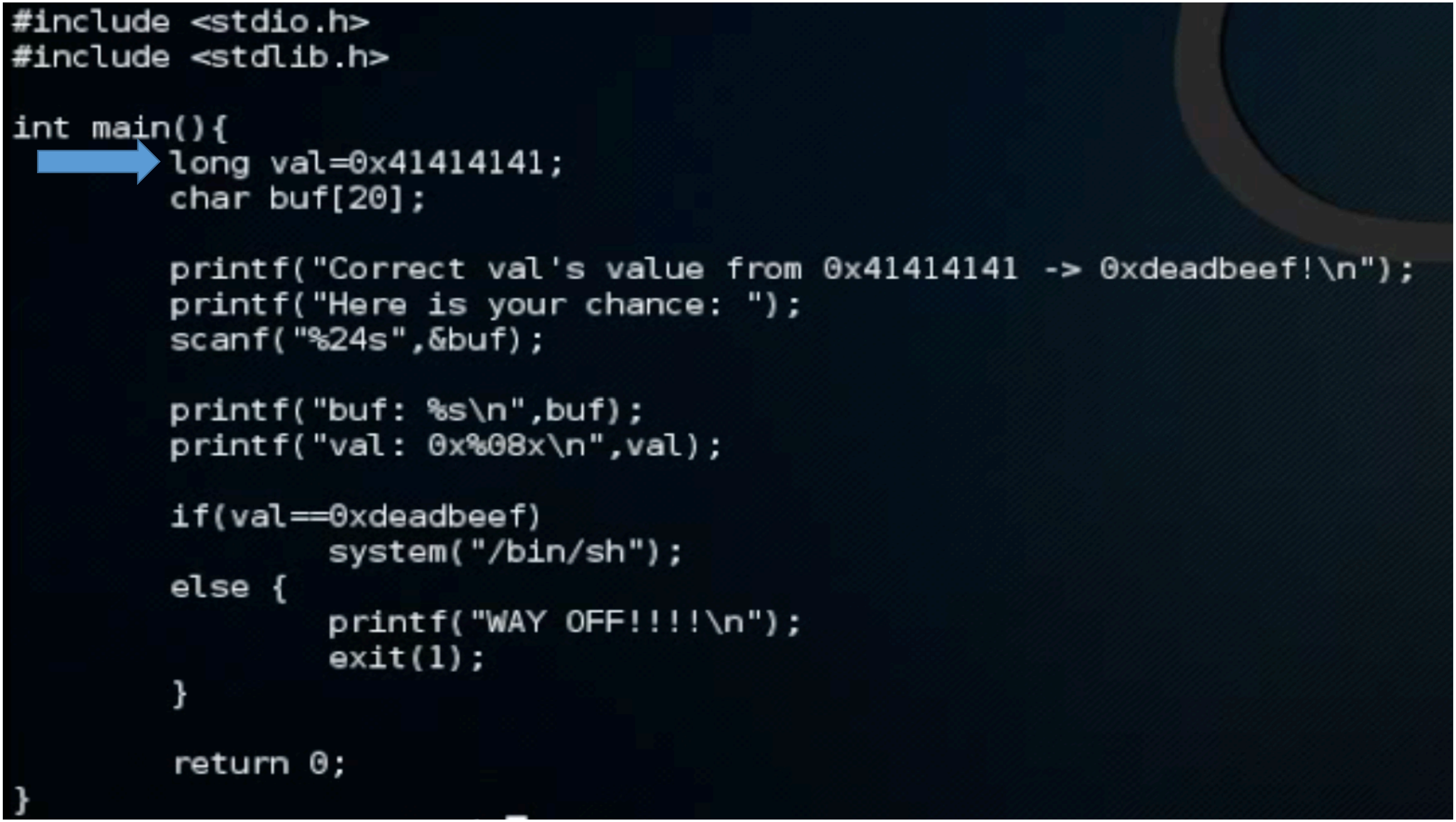
Data for the levels can be found in **/narnia/**.

- http://overthewire.org/wargames/narnia/

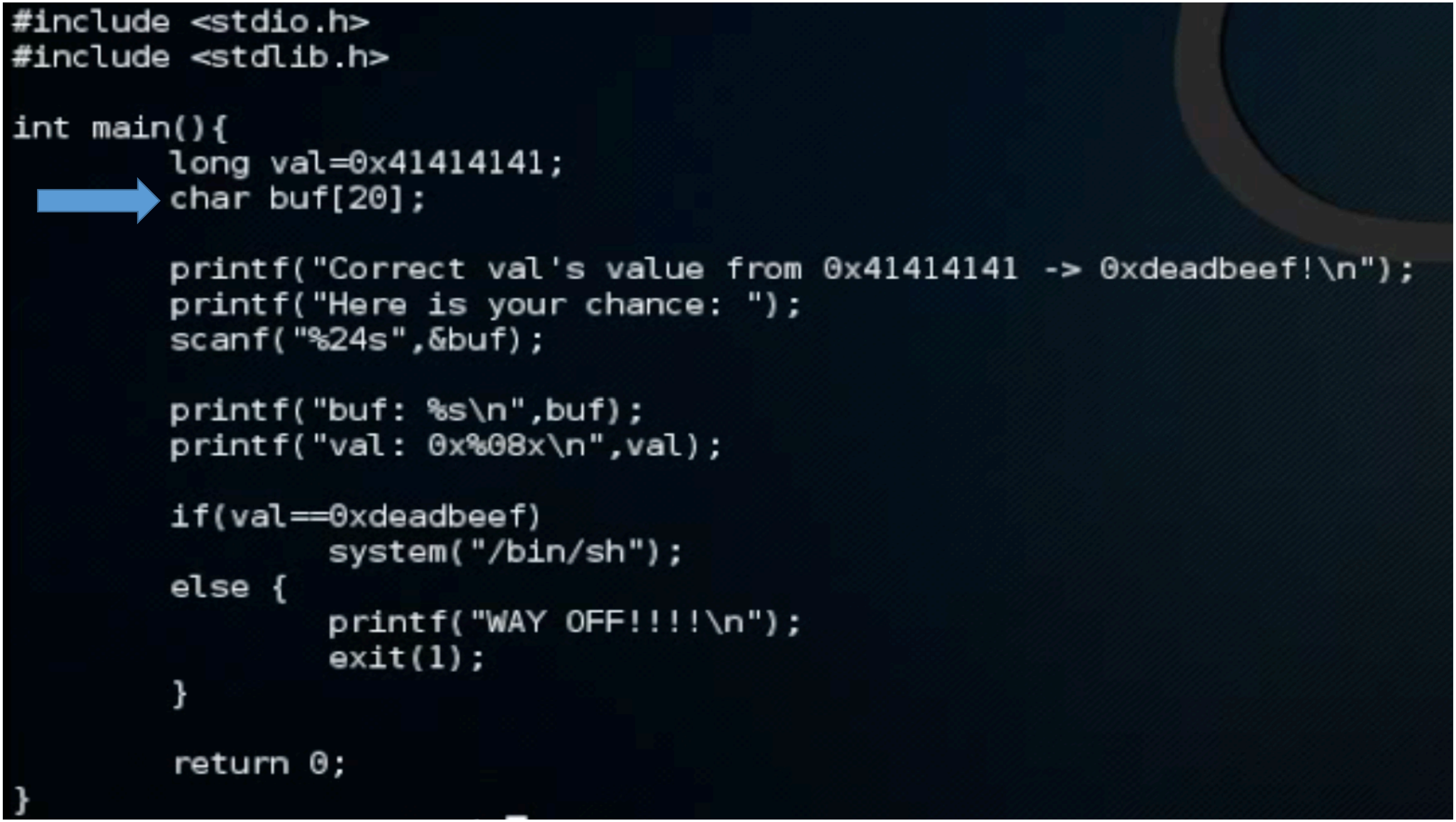
- A live environment you can connect to via SSH to attempt various binary exploitation challenges, including Buffer Overflows



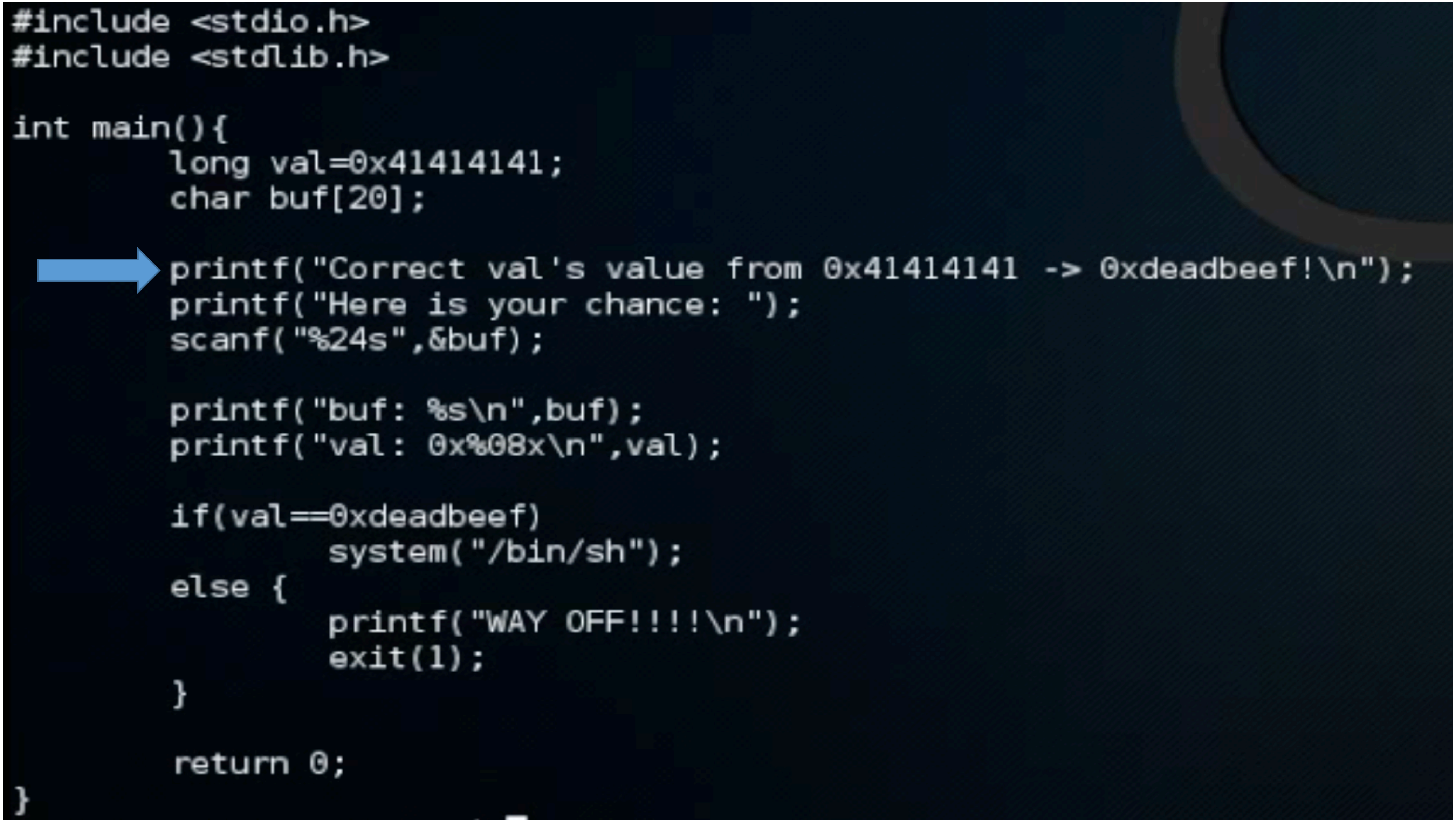
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



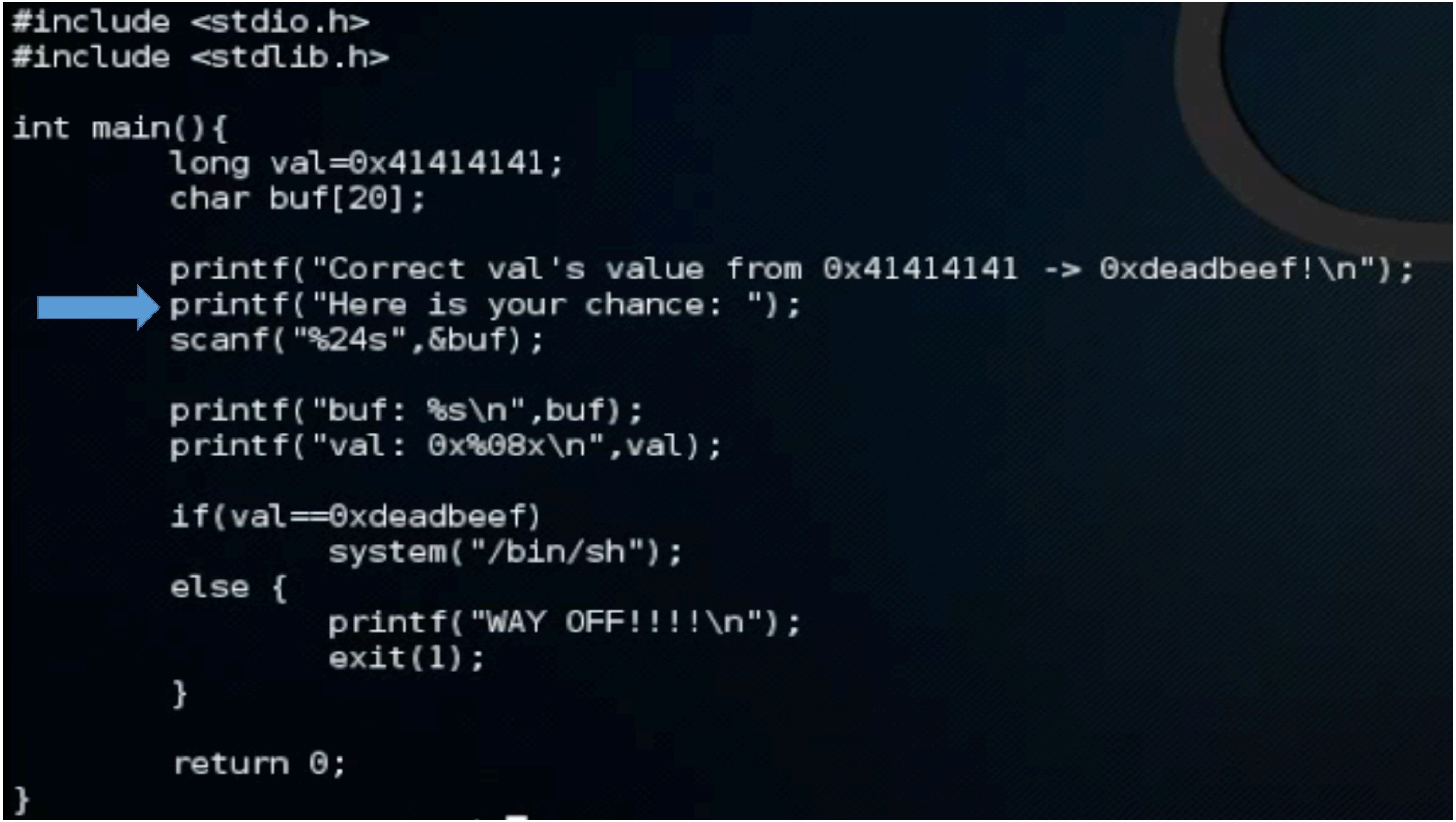
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



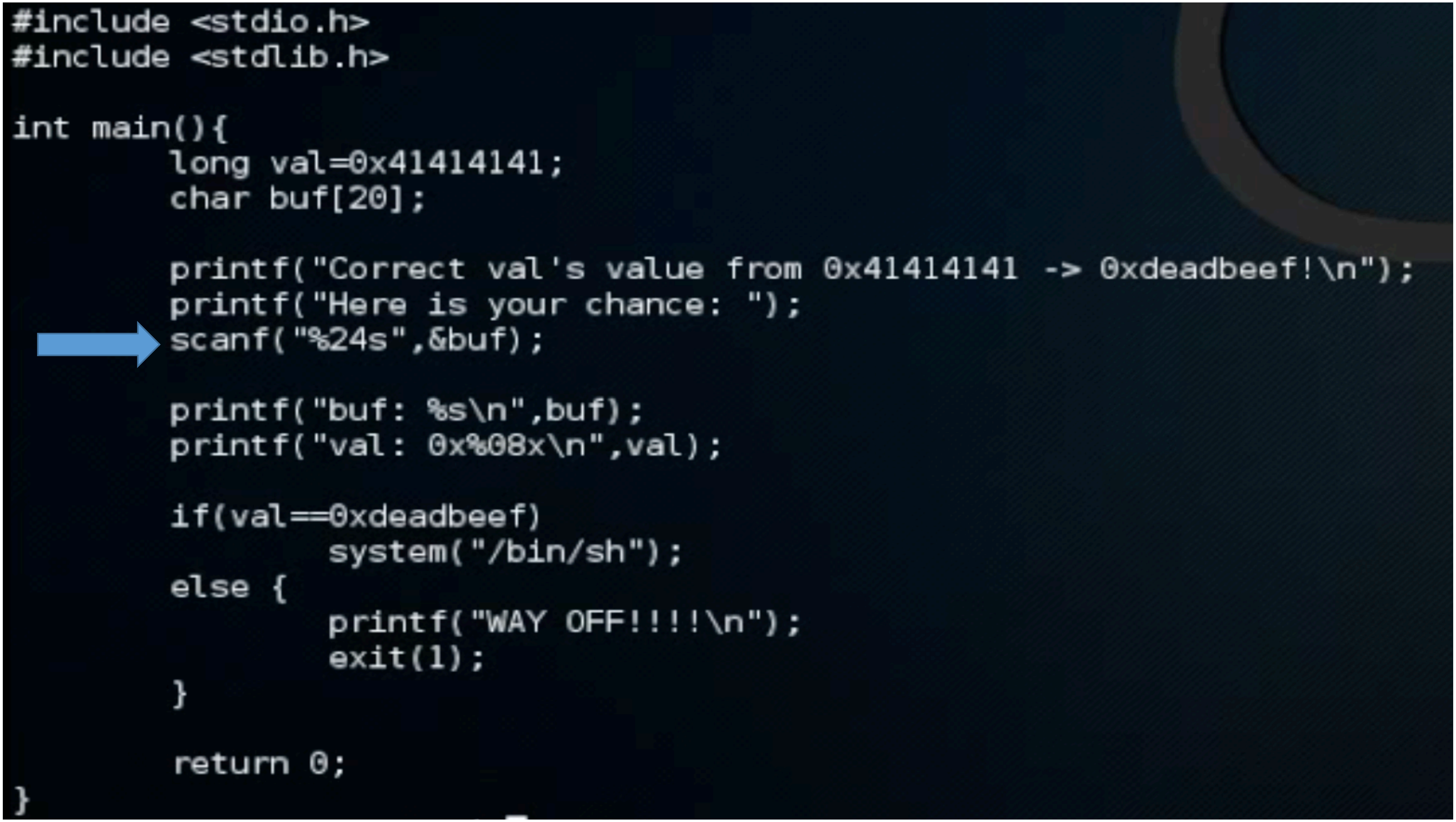
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



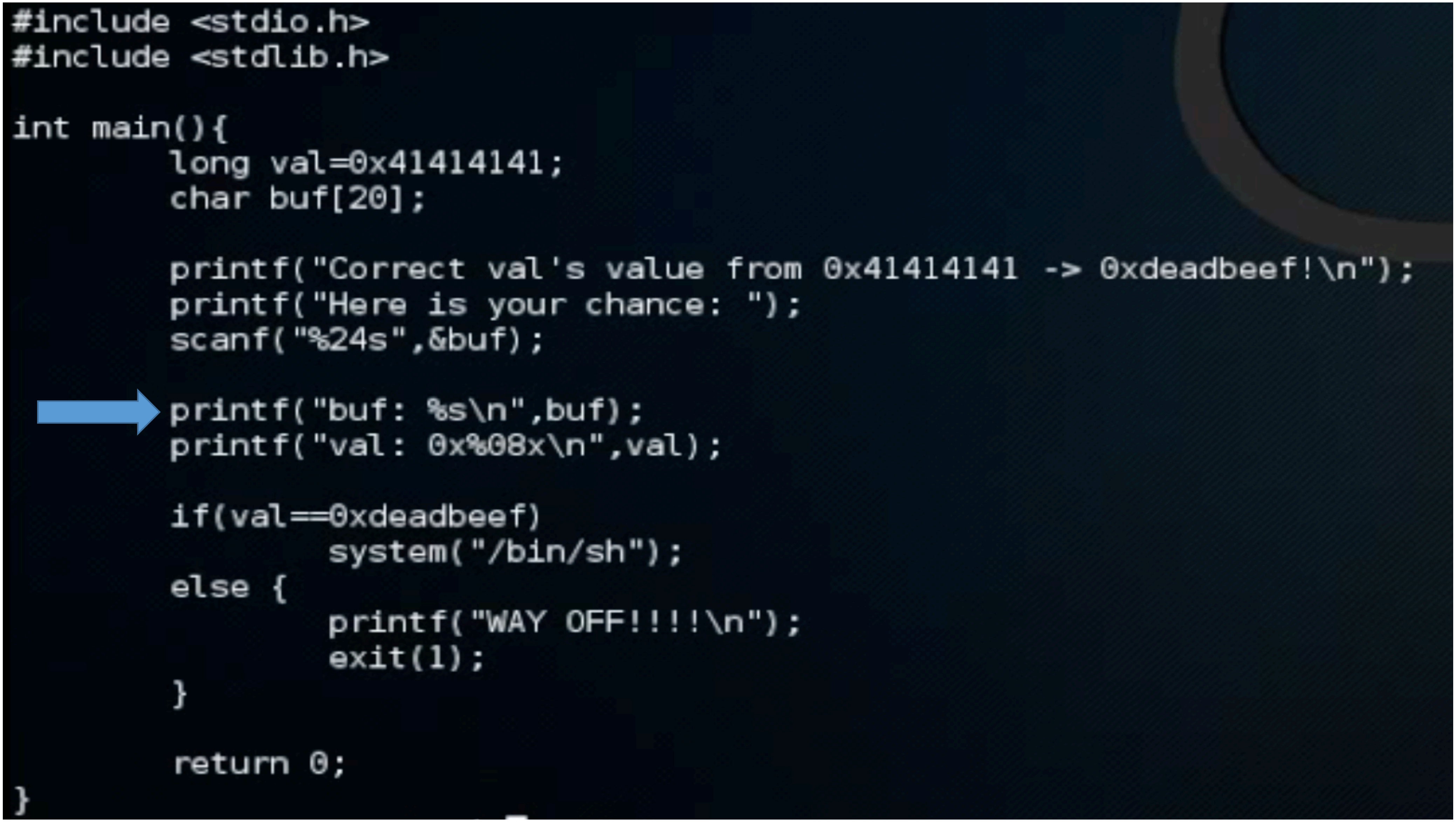
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



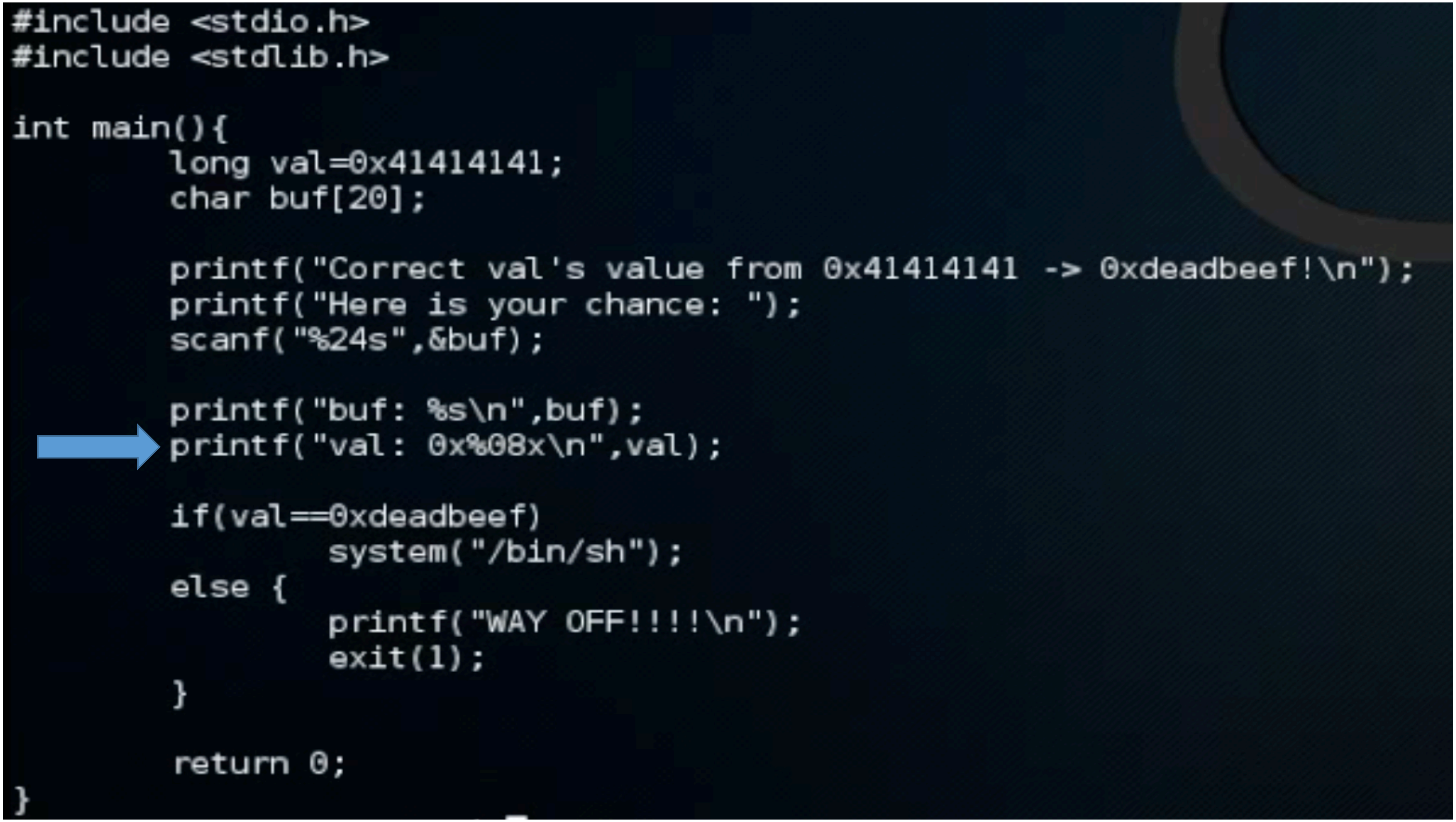
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



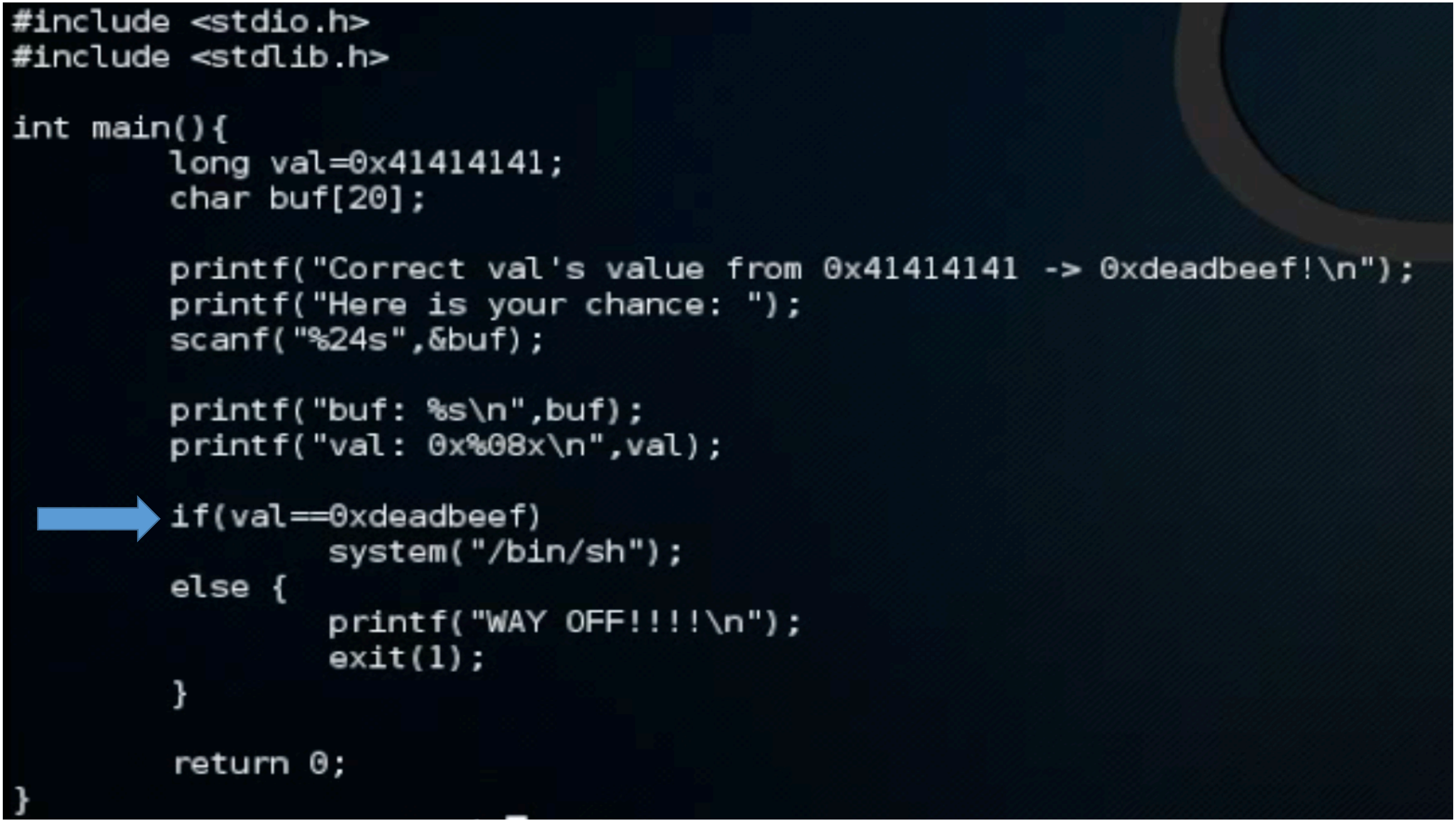
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



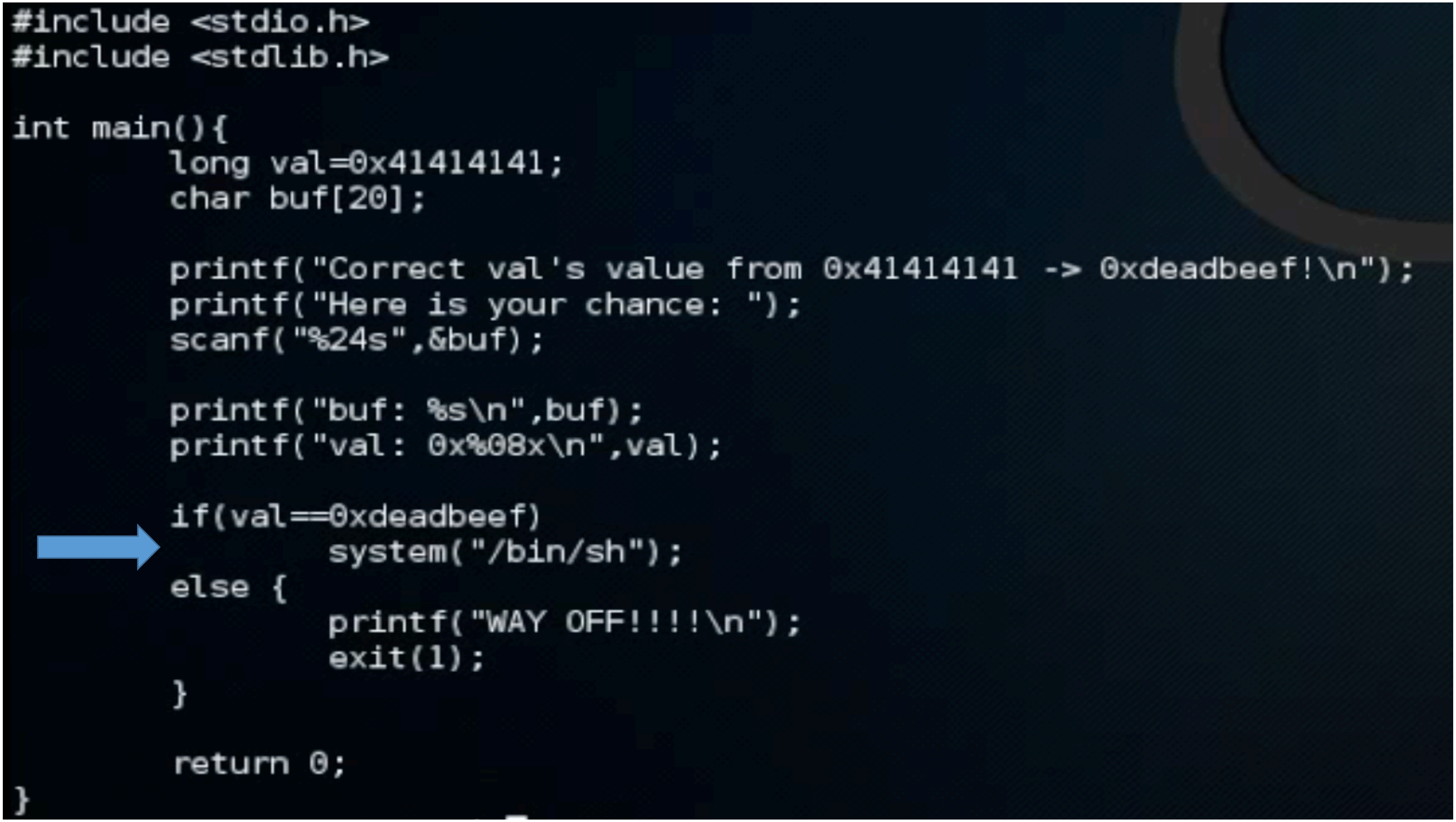
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



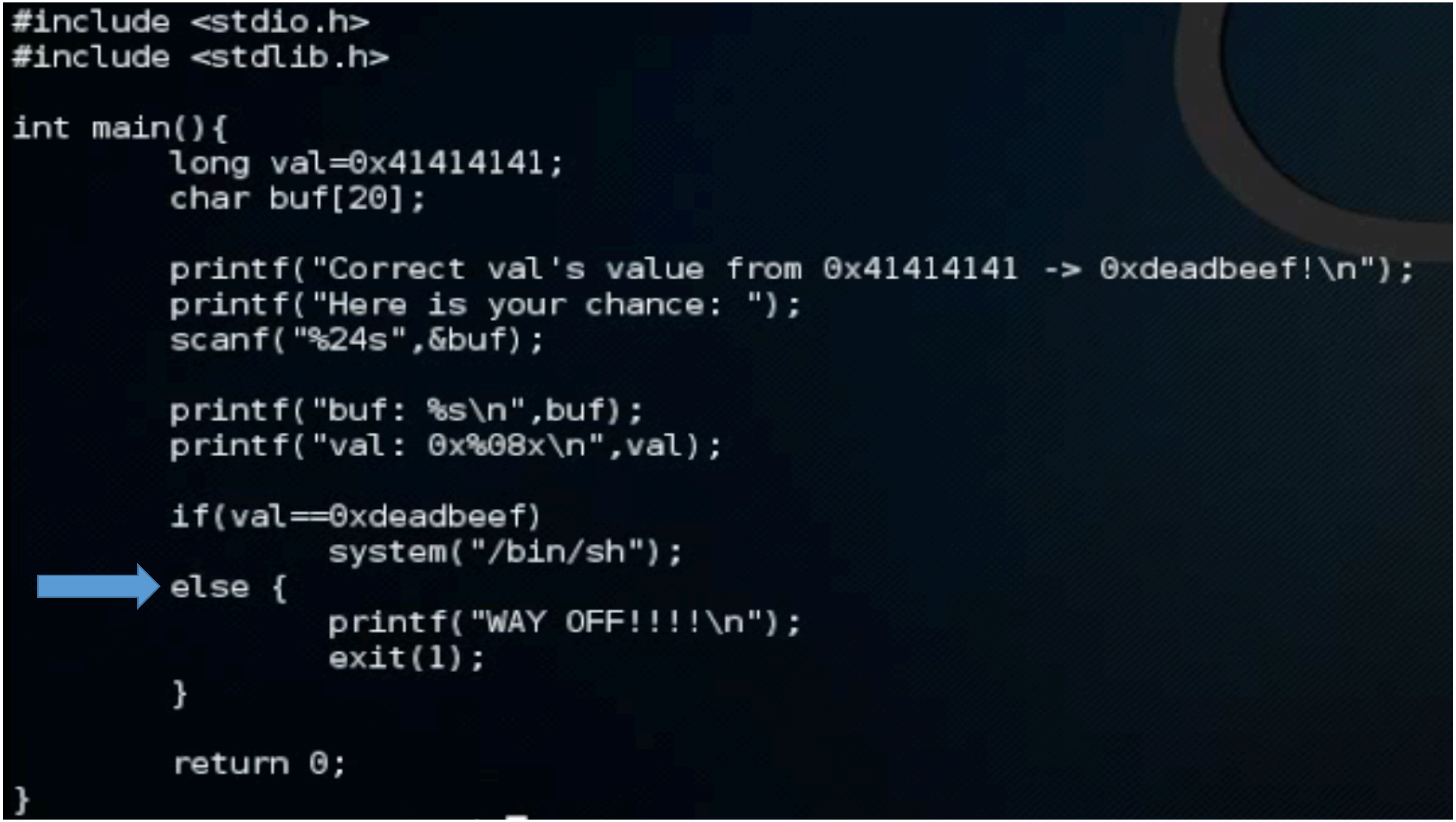
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



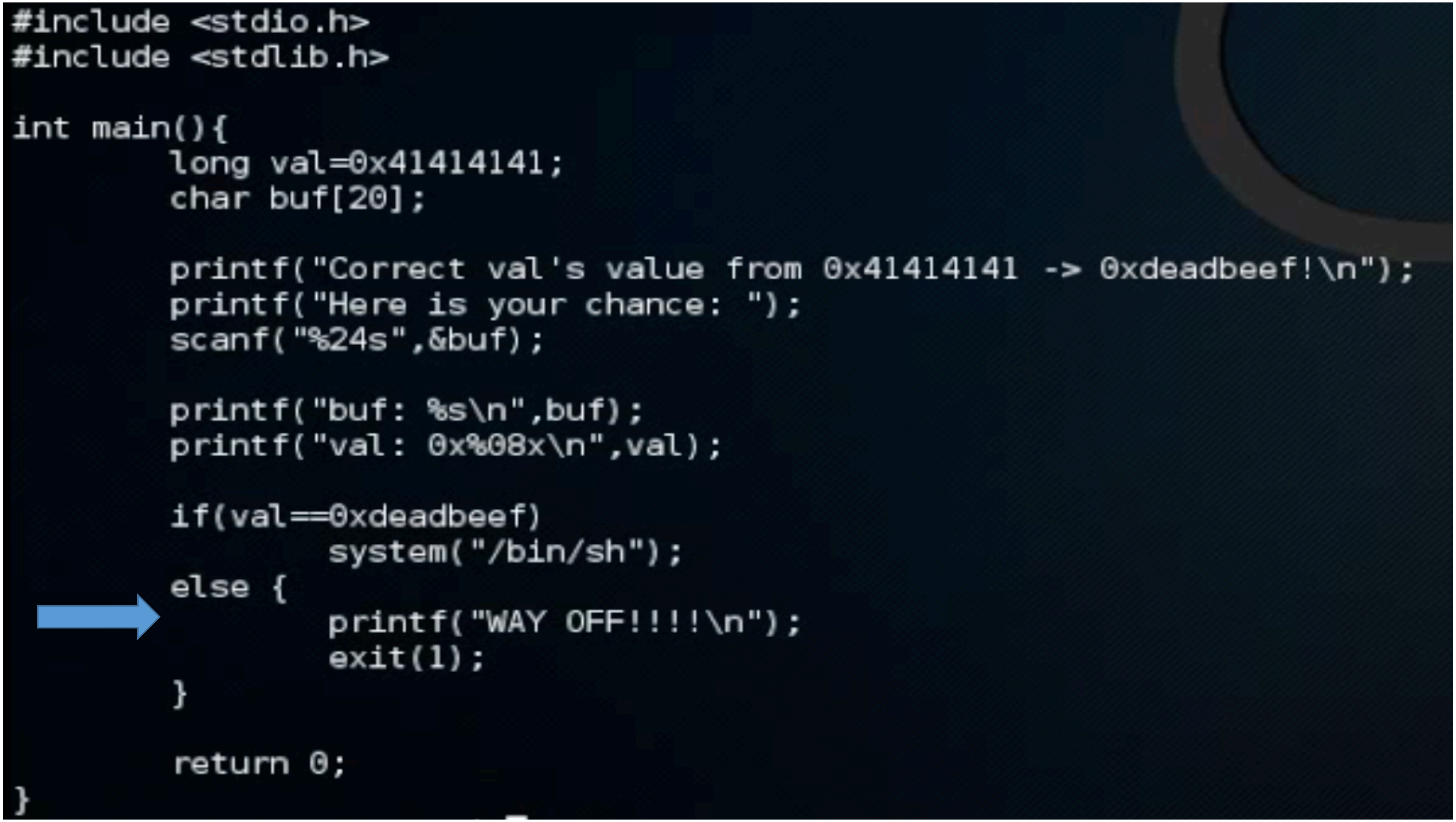
```c
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



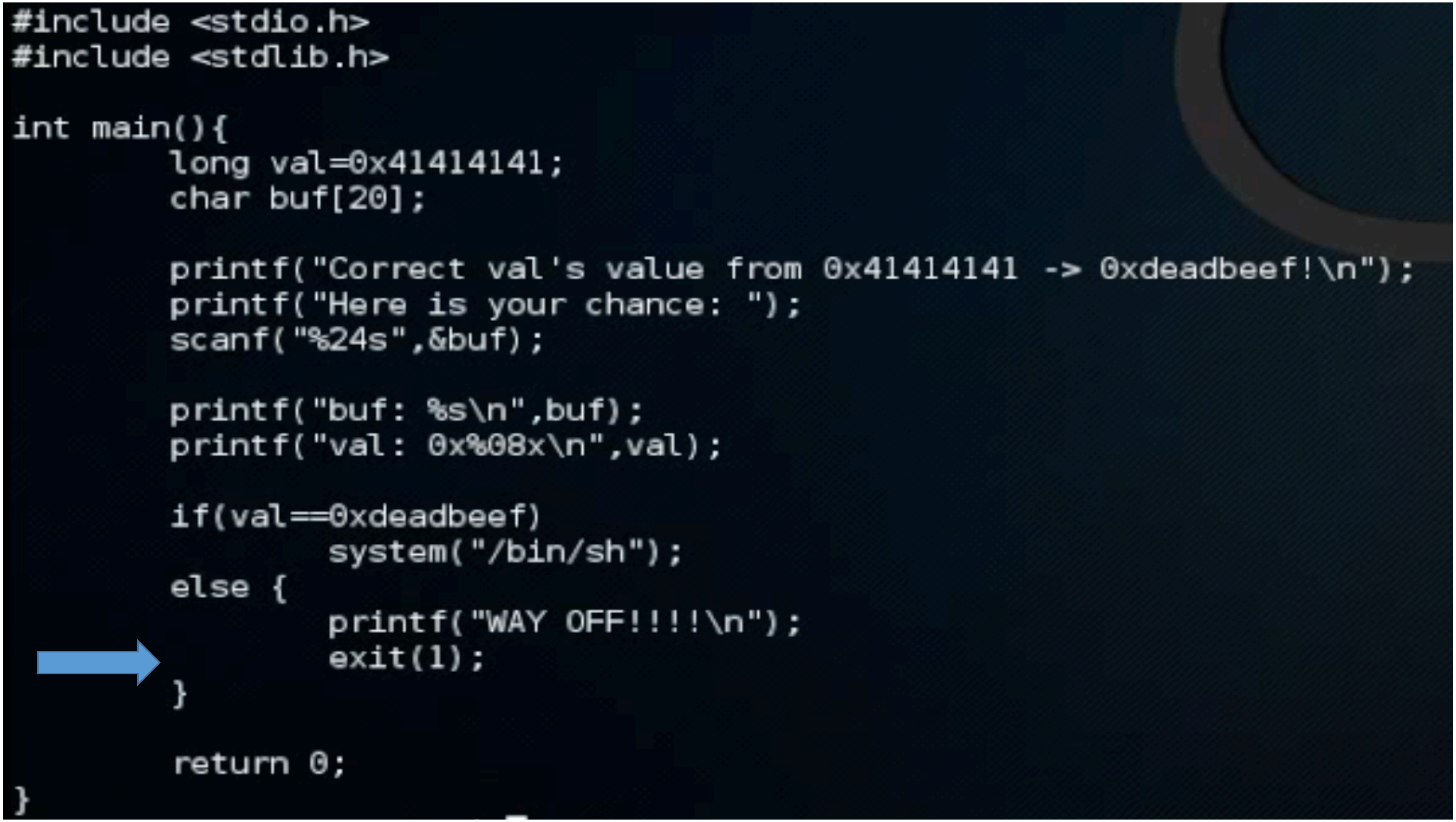
```c
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



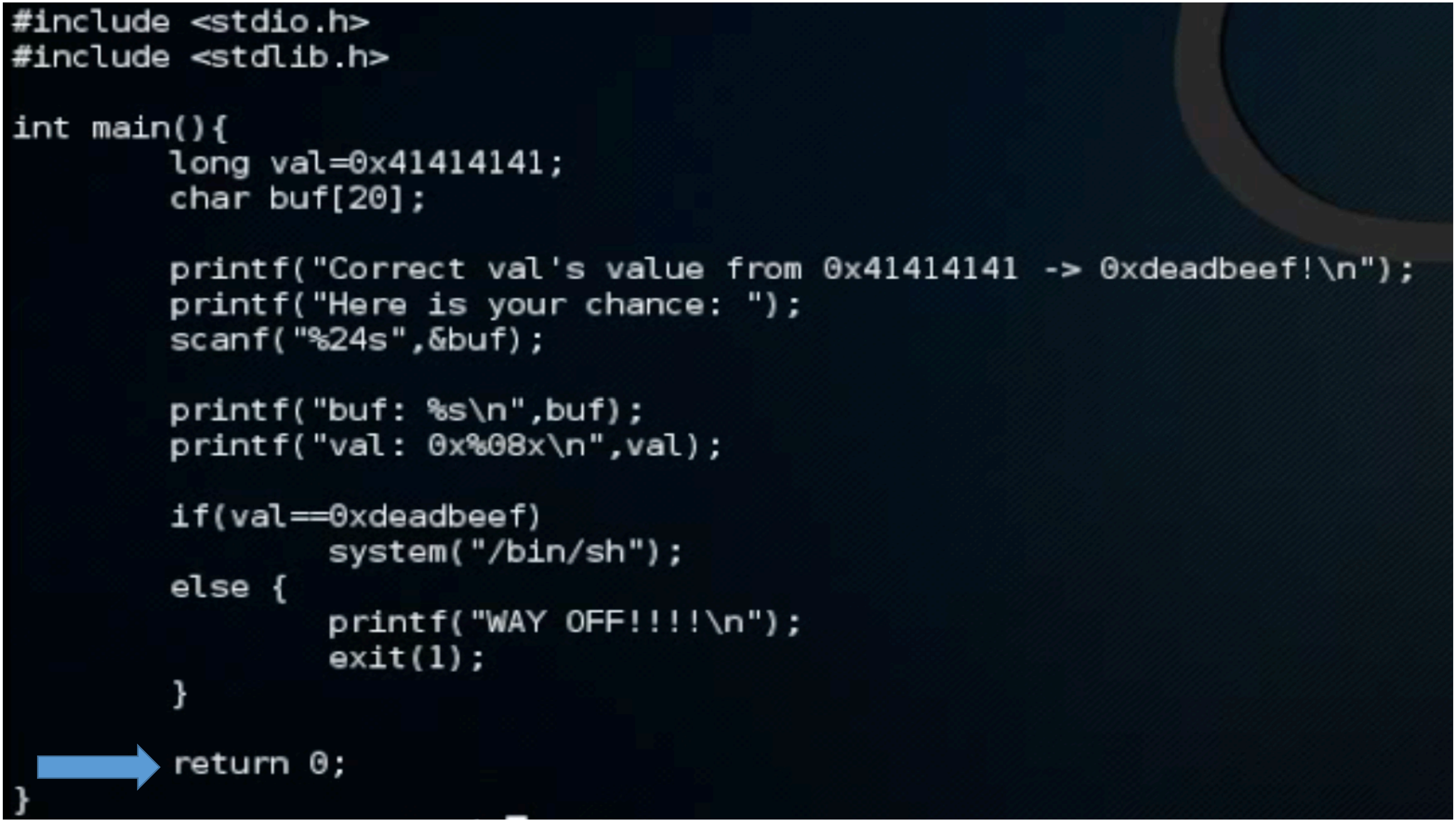
```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```



```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```

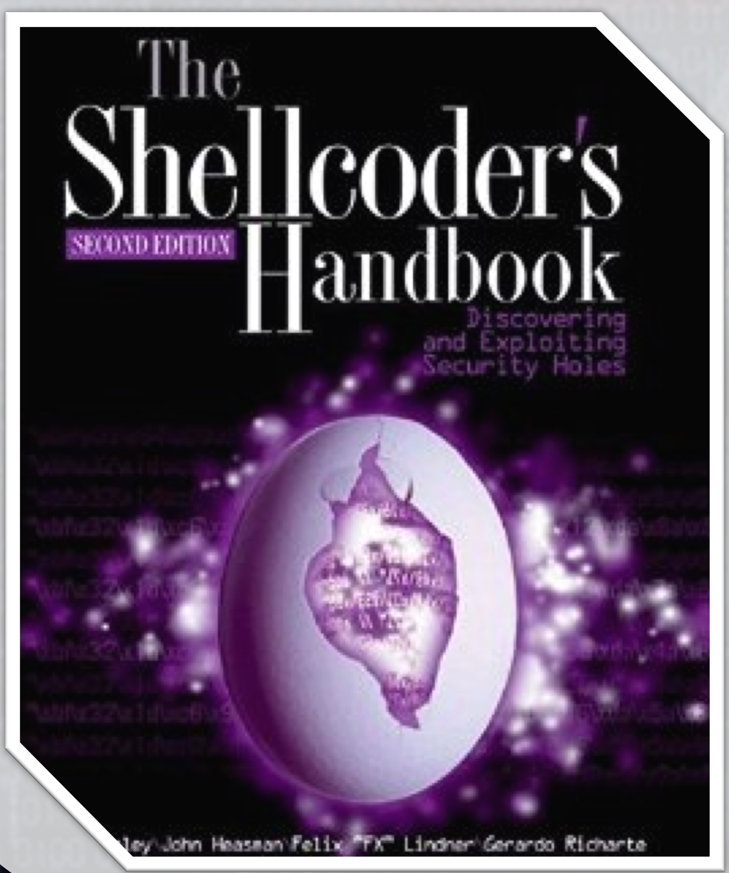


```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```

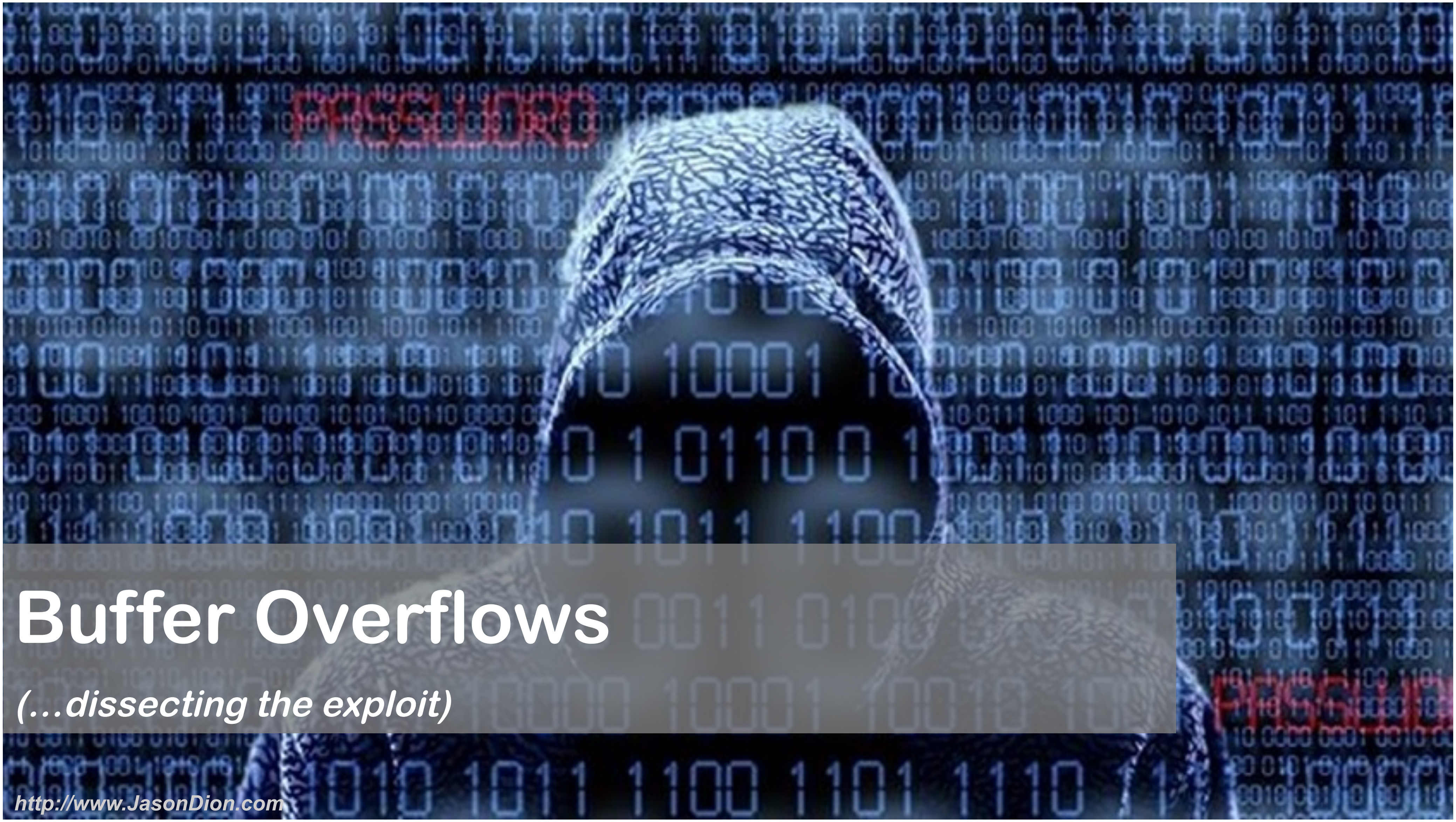


```
#include <stdio.h>
#include <stdlib.h>

int main(){
        long val=0x41414141;
        char buf[20];

        printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
        printf("Here is your chance: ");
        scanf("%24s",&buf);

        printf("buf: %s\n",buf);
        printf("val: 0x%08x\n",val);

        if(val==0xdeadbeef)
                system("/bin/sh");
        else {
                printf("WAY OFF!!!!\n");
                exit(1);
        }

        return 0;
}
```

Recommended Reading


The Shellcoder's Handbook
SECOND EDITION
Discovering and Exploiting Security Holes
ley John Heasman Felix "FX" Lindner Gerardo Richarte

- The Shellcoder’s Handbook
- Hacking: The Art of Exploitation
- The Hacker Playbook 2



Buffer Overflows

(…dissecting the exploit)

http://www.JasonDion.com