Introduction to CTFs

Slides by @kathyra

ETHICS - LET’S *NOT* GET ARRESTED

- The skills you learn doing CTFs are similar in many ways to the skills you need to do real hacking/auditing.
- What you are learning can be used for malicious purposes.
- Neither I, the society, or the university are responsible for your actions.
- Never attempt to illegally gain access to a system you do not own.
- You’re all smart people. Use common sense!

SO WHAT *ARE* CTFS?

- Capture The Flag challenges
- A CTF will involve some sort of system (an executable, a web server, etc.) that has a vulnerability written in
- Your job is to figure out what the vulnerability is, and how to exploit it
- A successful exploitation will give you a ‘flag’ - often a text string

BUT HOW DO I *DO* CTFS?

- CTFs will normally hint at what the intended exploit is. Read any text that comes with the challenge carefully (sometimes even just the name of the challenge can give away what the vulnerability is).
- Scope out the system. If you're given an executable with code, you'll want to read the code carefully. If you're doing an easy web CTF, a good place to start with is checking the source code.
- Finding the vulnerability is half the battle. After that, you figure out how to exploit it to gain access to the system.

OK, FINE. *NOW WHAT?*

- The best way to understand how CTFs work is to just do them.
- On the next slides I’ll be listing some more or less beginner friendly CTFs with a brief overview of how to approach them.
- They’re not arranged in any order of difficulty. Feel free to start with whatever ones look more interesting to you.
- *Collaborate with the people around you!* Sometimes you might miss a really obvious solution, and getting another perspective is incredibly valuable. (also, the more you talk to others, the less work I need to do. thanks.)

PWNABLE.KR

- http://pwnable.kr/play.php
- http://taishi8117.github.io/2015/10/26/pwnable-easy1/ a coherent writeup for some of the challenges - don't read it unless you're SUPER lost.
- The challenges here were used for this years CySCA training.
- These are all binary/reversing challenges. I hope you like assembly.
- For the easier ones, you'll get source code. For the harder ones (I won't be covering these) you'll just get a binary file that you'll have to reverse using a program like IDA Pro.
- As far as beginner challenges go, the pwnable.kr ones are HARD. Don't feel bad if you have trouble with them. I've spent HOURS on each.

PWNABLE.KR - first challenge: “fd”

- You’ll need to pull up a terminal and ssh into the pwnable.kr server.
- ls around and take a look at the source code.
- Hint: when fd == 0, it will read from from stdin.

PWNABLE. KR - “collision”

- This one involves giving the program hex input. A little tricky.
- Hint: it takes in 5 values, and expects them to add up to something.
- Hint: learn how to run executables with hex input. You can use Python or Perl for this.

PWNABLE.KR - “flag”

- This is a reversing problem.
- Hint: UPX.

OverTheWire

- Much easier than pwnable.kr
- http://overthewire.org/wargames/natas/ - web challenges. These are in ascending order of difficulty.
- Writeup is here: https://infamoussyn.com/2014/02/05/overthewire-natas-level-0-16-writeup-updated/ (but again, don’t look unless you’re SUPER lost)

Backdoor

- https://backdoor.sdslabs.co/beginner
- Assortment of various challenges