CTF-League

Systems and Security Group, Web Enthusiasts' Club

So.. What is a CTF?

- A Security Competition with challenges mimicking real-life security scenarios.

What are those real-life scenarios?

An Example!

Categories

1. Web
2. Crypto
3. Reverse Engineering / Software Security
4. Forensics

Lets get started!

RE / PWN

What is Engineering?

What is Engineering?

You design and develop a solution for a given problem.

Examples

1. Intel processors
2. Operating Systems (Linux, Windows, MAC-OS etc.,)
3. Compilers

Endless list!

What is Reverse Engineering?

What is Reverse Engineering?

An example!

Download the zip file from https://github.com/WebClub-NITK/CTF-League

What is Reverse Engineering?

- Literally the reverse of Engineering
- Given a solution(or a program), it is about finding out its functionality, what it does **when source is not given**.

Example: Windows OS

1. The most popular Desktop OS.
2. 85% of people use it.

Example: Windows OS

1. The most popular Desktop OS.
2. 85% of people use it.

Qn: Do we know how it works? What it’s code does? What algorithms are used? and so many more questions.

Why Reverse Engineering?

1. Understanding the complete functionality of a given program.
   a. What data structures does Windows use to perform process management?
   b. Are there any problems with their approach?
   c. Can they do better?

Why Reverse Engineering?

1. Understanding the complete functionality of a given program.
2. Make sure it doesn’t have any hidden functionality, like sending your information to Chinese hackers :P
    a. No backdoors, just a clean software.

Why Reverse Engineering?

1. Understanding the complete functionality of a given program.
2. Make sure it doesn’t have any hidden functionality, like sending sensitive information to Chinese hackers :P
3. Make sure the program is written in the right way.
   a. Let us see what this means.

Example time!

Why Reverse Engineering?

One more very important application of Reverse Engineering:

Vulnerabilities and Exploitation.

What is a bug?

- It is a flaw in a computer program.

What is a bug?

- It is a flaw in a computer program.

What if that flaw can be used to hack into that server?

What is a bug?

- It is a flaw in a computer program.

What if that flaw can be used to hack into that server?

That is a **vulnerability!**

Vulnerability!

Example time again!

What are we dealing with here?

You just exploited a buffer overflow vulnerability here!

What are we dealing with here?

- These type of vulnerabilities make up to 40% of total.
- They are **deadly af**.

Take a look!

What are we dealing with here?

- These type of vulnerabilities make up to 40% of total.
- They are **deadly af**.

Take a look!

1. https://www.facebook.com/security/advisories/cve-2019-3568
2. https://www.cvedetails.com/vulnerability-list/opov-1/overflow.html

What are we dealing with here?

- Deadly vulnerabilities
- Ways to **screw it / exploit** it and get access.
- Finding ways to defend these exploits.

What do we do in this category? : Summary

1. CTFs call this category as **Pwn**.
2. We will be dealing with **RE** and **Pwn** categories in a CTF.
3. Software Reverse Engineering and Security
4. Understanding Real-world vulnerabilities
5. Developing exploits for those vulnerabilities
6. Various types of security techniques to defend against these.
7. Operating Systems (Linux, Windows, MAC-OS), Programming Languages.

What do we do in this category? : Summary

- I want it to be more than just a CTF category.
- Lot of research opportunities.
- Break stuff or defend stuff - both are welcome :)
- It could be finding
    - A new vulnerability in a program
    - A new way to attack a security technique
    - New ways to exploit currently known vulnerabilities
    - and more!

* Finding processor vulnerabilities is an emerging research area.

How to get started?

1. Overthewire has a lot of good challenges in the form of games. You may start playing leviathon, Narnia.
2. Team bi0s's wiki on Reversing
3. Team bi0s's wiki on Pwning
4. Intro to RE and BE : Introduction to RE and BE.