Wargame 5 | COMP6447 20T2

The following wargames will provide you with exercises where you will be required to:

1. hack stuff

2. Defeat ASLR in a stack canary challenge

3. Find bugs in source

You can download the challenges here: https://cloudstor.aarnet.edu.au/plus/s/3itfqT6ira5Y5OX

These challenges are a zip file with the password: Spl0itwarzWasFun??

There are 3 **exploitation challenges** this week, **1 reverse engineering challenge** and 1 **source code auditing challenges** !

Try to solve the **exploitation** challenges locally first, then connect to our servers to obtain the flags. To get full marks you must get the flag from our servers.

Challenge	IP:PORT	type
shellcrack	plsdonthaq.me:5001	pwn
stack-dump2	plsdonthaq.me:5002	pwn
image-viewer (**source provided**)	plsdonthaq.me:5003	pwn
source.c	none	src
re	none	re

Each **exploitation** challenge has a flag to submit. The flag is in the format FLAG{XXX}. To get full marks in this wargame, you need to submit all flags.

Source and Reversing challenges

There is **one** reversing challenge this week. You need to submit the **most simplified version** of C that would compile into the supplied assembly

simply putting each instruction into it's C counterpart is not correct. You must notice patterns and simplify the C code to understand what it is doing!!!

```
int modulo_14(int a) {
    int c = a + 0x10;
    int d = c & 0x813910412;
    int e = c + a - b;
    int f = d * 0xde000 + e;
    printf("%d\n", f);
}
```

vs

```
int modulo_14(int a, int b) {
    printf("%d\n", a % 14);
}
```

The first won't get any marks

For the source code auditing challenges, you must submit a list of **memory corruption vulnerablities** with the supplied source.

We don't care about syntax issues, missing imports, typos, bugs that aren't vulnerable/exploitable. We want to see bugs that could allow an attacker to gain leverage over the system. Submitting too many incorrect

bugs will result in lower marks for this section.

Each vuln you find, please note down the line number of the vuln.

Be detailed about why the vulnerability you disclosed is vulnerable. if you don't you won't get the marks

Submission Instructions

A markdown document (.md) containing the following for each challenge:

We are interested in proof that you understood the challenge, the vulnerabilities and how to exploit them. This is not intended as a formal bug report.

````
chal1
==========================
Flag: FLAG{hi}
General overview of problems faced
-------------------------------------
Had to hack the program
Script/Command used
------------------
```
print "hello_world"
```
src challenge
=================
General overview of problems faced
-------------------------------------
lines: Bug
1: The developer used python
18: Integer overflow on variable x because y is user controlled, and y is passed to fgets as size param
re challenge
=============
General overview of problems faced
-------------------------------------
needed to use man page to find arguments for atoi
```C
int main(int argc, char** argv) {
    int a = atoi(argv[1][0]);
    if (a > 20) {
        printf("%d\n", a + b);
    }
}
``
````

Please submit the document as a markdown file on give. You may submit as many times as you like. Only your most recent submission will be marked.

Submission

```
give cs6447 war5 war5.md
```

Marking scheme

Each week's wargames are worth 4 marks in total.

Due date

The wargames are due **17:59 Tuesday 14th July (Sydney time).** This is in Week 7.

Late Penalty

Late submissions will have marks deducted from the maximum achievable mark at the rate of 1 mark *per day* that they are late.
```

Resource created 2 months ago, last modified 8 days ago.