Middle East Technical University
Department of Computer Engineering

CENG 489

Spring 2019-2020

Homework 2 - Viruses and Pwning

Deadline: 05.06.2020, Friday, 23:59

Overview

In this assignment we will put on our black/?gray? hats for a little while and do some naughty things (not really but you get the idea). You should not use the skills we practice here in the real world without an explicit permission from the parties that will be affected. Our main focus will be on computer viruses and hacking and controlling a system (i.e. Privilege escalation or pwning in leetspeak).

In the first part of the homework we will create a computer virus. In the second part we will gain a root shell in a remote system.

Part 1: Viruses

Computer viruses are programs which carry (not always) malicious code and when run, infects other files by replicating and injecting itself. In turn when the infected files are executed and run infects other files and systems.

Every virus has three basic parts: an infection vector, a trigger and a payload. Infection vector is the means the virus spreads itself. Trigger is the part of the virus which determines whether the payload should be executed or not. And the payload is the actual malicious part of the virus, which can wreak havoc or can be really tame. Viruses should also be able to remain undetected by other programs. There are many ways a virus can hide itself but most of these methods rely on the virus modifying itself and changing itself slightly with each infection. Things we need to do for each part are explained futher below.

Coronavirus Virus: We are sick of people breaking the quarantine and going out for fun and spreading Coronavirus. We want to teach them a lesson about the seriousness of the matter. We come up with the idea of a computer virus which displays the countries with highest coronavirus infections. That should teach them that their actions have consequences.

For simplicity and ease of coding we choose to write our virus in Python and we will only infect Python files. And our trigger will be just running the infected file. When run, the virus will infect all the other python scripts **in the current directory and all the subdirectories**[1] and our payload will be to display the Coronavirus infection counts. We also want to make our virus resillient to antivirus programs so that our virus could spread and "educate" as may people as it can. For this our virus will encrypt itself while infecting files with a one time pad and decrpyt itself when run. This will cause the virus to be modified at each infection and make it harder to detect. Moreover, our virus shouldn't infect a file which has been infected previously.

[1] On a typical virus at this stage we would discover network devices and send them the virus or create and send emails to all the contacts available in the machine etc. to spread the virus

Step by step our virus should perform the following:

- **Copy Itself:** Our virus should create an encrypted copy of itself with a one time pad. The encrypted version should also contain the key in order to decrpyt itself later.
- **Infect Files:** Our virus should search for uninfected files in the current directory and all of its subdirectories and infect all the python script files it finds. It should also somehow mark the files it infected so that they won't be infected a second time.
- **Execute payload:** When an infected file is executed it should decrpyt itself and run the payload and perform above steps in order.

That's it!

And some tips that will help you:

- Python's `exec()` function. [to execute decrypted virus]
- Base64 encoding and decoding. [Not necessary, but it may come in handy after encryption]
- Cryptographic hash algorithms. [to identify previously infected files]
- There are many repositories on GitHub which provide an API for Coronavirus tracking. You can use any of them. For example

Submission and Evaluation

Evaluation of your viruses wiil be a mix of whitebox and blackbox testing so please don't forget to comment your code. Your viruses will be run on a VM with several python scripts scattered throughout the system and will be checked for infections after your submission is run.
Your submissions will be through Piazza.

- To submit your homework click on the "New Post" on the upper left corner. Then, make sure that you choose "Individual Student(s)/Instructor(s)" option for the "Post to" section and address it to me (Hakan Bostan). Also select the "hw2" folder.
- On the summary line put "`[HW2p1 Submission eXXXXXXX]`" where `XXXXXXX` is your 7-digit student number.
- Upload your submission via the "Insert" menu at the top of the editor (if you wish you can add explanatory comments in the message body).
- Finally make sure the name of your `.py` file is in the form of `eXXXXXXX.py` again `XXXXXXX` is your 7-digit student number.

Part 2: Privilege Escalation

In this part we will exploit a program running on a remote machine to gain root access to the machine. If you have taken the Computer Organization course, what we will do here is similar to that. We will find a bug in the code and exploit it to have a shell with root privileges on the target system. You will be provided with the unstripped binary of the program running at the remote system to make your job easier.

Background on setuid

When you look at the permissions of some executable files you might see a `s` on the execute permission for root user. For example `ping` has this s in its permissions (`-rwsr-xr-x`). This s means that it is a `setuid` binary. Well what is a setuid? It is a libc function which allows users to run an executable with the permissions of the executable's owner. This is often used to allow users to run programs with temporarily elevated privileges. For example `ping` needs to create ICMP packages to send out to network. This requires `ping` to work with lower layers of network stack and you cannot do this unless you are `root`. However since `ping` has its setuid bit set and it is owned by root, when a less privileged user executes `ping` it runs with the privileges of the `root` user. This page should come in handy when understanding setuid exploits.

This allows us to gain higher privileges by exploiting these binaries. The file we are exploiting in this part also has its setuid bit set. Our eventual purpose is to spawn a root shell by providing it with a maliciously crafted input.

Once you have a root shell at the remote machine all you have to do is create a folder with your student id and place the script you created to exploit the system in that folder.

This is it for this part!

Remote machine: `hbostann.com (188.166.14.126)`
Port: `1337`

Some notes about this part:

- Source code is compiled with `gcc -m32 -o overwrite overwrite.c`. That is, it has all the default exploit protections such as stack canaries, non executable stack etc. This means that the attack you will perform doesn't try to change the return pointer, or inject shellcode.
- Develop your exploit locally first and then use it to exploit the remote machine. Do not try to bruteforce your way in, any such attempt will be classified as a DoS and will be punished.
- Source code of the program will be provided 10 days before the deadline of the homework to make your job even easier. If you submit your solution for this part before the source code is released you will get bonus points.

Some tips that may help you:

- While looking for exploits try to identify what you can input into the program and what it affects.
- `gdb` and `objdump` are your best friends.
- You can use `nc 188.166.14.126 1337` to connect to the remote machine.
- You can use `(./your_exploit_script; cat)|nc 188.166.14.126 1337` to keep your connection open to the remote server to have an interactive shell.
- Read the man pages!

Submission and Evaluation

Your exploit scripts will be run again to see if they actually work, so please also submit your exploit script for this parth through piazza:

- To submit your homework click on the "New Post" on the upper left corner. Then, make sure that you choose "Individual Student(s)/Instructor(s)" option for the "Post to" section and address it to me (Hakan Bostan). Also select the "hw2" folder.
- On the summary line put "`[HW2p2 Submission eXXXXXXX]`" where `XXXXXXX` is your 7-digit student number.
- Upload your submission via the "Insert" menu at the top of the editor (if you wish you can add explanatory comments in the message body).
- Finally make sure the name of your `.py` file is in the form of `eXXXXXXX.py` again `XXXXXXX` is your 7-digit student number.