SkyPenguin Labs

REplay Writeup - Introduction

Welcome to the official REplay writeup! This is the first document for understanding not only these documents but REplay. These document(s) will be acting as a guide on how to do something within REplay or answer questions, but will NOT go over the fundamental theory required. If you do want that, there are TONS of articles, I, the writer (Totally_Not_A_Haxxer) has released on my medium and that you can find by simply Googling my user!

The first thing I would like to mention right off the bat is going to be the CTF, what its for, what you will learn, and so on from there.

What Is REplay?

REplay pronounced 'replay' is a reverse engineering playground that is being developed for beginner reverse engineers all the way up to practitioners and experts in the RE space and exploit development space. With this, it is a playground also filled of many different levels (approximately 3 is whats in store) with many many different more unique and difficult questions (15 questions just in the first level) that can help you sharpen your reverse engineering skills!

Why Was REplay Made?

REplay was made because of the many people I mentor and teach actively on reverse engineering. I got tired of doing crackme challenges and I was also certain the students were not prepared for full fledged malware analysis! So before they decided to just get tossed into the water, I wanted to also examine their knowledge, the way they logically comprehend things and so on from there. So the idea came to me that if there was not actively a reverse engineering playground, I would make one surrounded with my experience as both a developer and a reverse engineer to help them and walk them all through new ideas and methodologies throughout the reverse

engineering world. When I created it, I felt that it would be only appropriate to publicize it and turn it into something from any kind of background no matter how advanced, can use this as a really good mind builder for specific skills even coming down to binary analysis and memory forensics!

How is REplay Designed?

REplay is designed to be a real world environment mimicking the world of black-hat software cracking. There are many crackme challenges, but a lot of times, you not only have to dig through the disassembly for days but have to document undocumented features or codes, document any developer panels or missing information, and so much more before you can make a valid crack. REplay was designed to mimic this experience, it was meant to be a game cheat that you want to crack and resell. So it has an entire GUI that was taken from my own CS2 cheat that I built a while back ago. For context, this application and CTF does NOT contain any code that can fully operate (systems that require the exploit code to work were ripped out) HOWEVER- bypasses for specific anti cheat systems are in there, there is some neat little shell code and a ton of things that are tossed in there for you to analyze.

REplay and Levels

REplay is a multi-level and multi-stage playground designed purely around specific skillsets. There are a total of 3 levels and each level gets and becomes harder than the next. Below are the levels in detail.

A. Level 1: Level 1 is the most easiest level, it is designed for beginners to explore the environment, get used to their tools and frameworks, and also get used to real world scenarios and understand larger environments. This level has about 14 questions and 15 objectives which you must complete to practice your skills. This level contains basic string encryption, basic anti-debugging systems and basic protection and logic.

B. Level 2: Level 2 is a intermediate level, there are many different systems including binary integrity systems, remote resources used, different functions and fun rabbit holes of binary auditing to go through. This one will also include the need to develop programs in C++20 working with the

native Windows API to gain access to the application and complete specific questions and tasks. This level is also for people who want to understand environments deeper.

_

C. Level 3: Level 3 is the hardest level by far. Not only does it include many different questions, sheets, information, binary analysis rabbit holes and general rabbit holes- it also includes exploit development from scratch using your own tools. This level will require the following skills - digital forensics, kernel development, development, exploit development, research & development, reporting, reverse engineering packers and cryptographic algorithms, defeating and bypassing complex systems and even being able to bypass memory locks and virtual randomization. This is a SUPER complex 30+ question level that will really cap off the level between exploit development and reverse engineering.

All 3 levels are designed to teach you a MASS amount of skills and a MASS amount of knowledge in reverse engineering and exploit development and was purely designed to make people get a HUGE interest in reverse engineering.

Why REplay

REplay was designed to solve a problem that the developer saw amongst the reverse engineering world- that there were not nearly enough beginner friendly playgrounds. Of course, there were MASSIVE CTFs by companies like Google but they were only single flags and they were either too beginner level or too expert based.

So I shot off to design something that was more than just a flag- that had rabbit holes, made you chase your tail and so on from there and mimicked a real world situation. REplay, due to that design, is one of the more stronger and well developed playgrounds out there that is guaranteed to teach you something about RE- especially if you are new!

You Should Know ... Before Using This CTF

There are many things I will need to make sure you understand before running and playing with this playground. They are listed below.

A. This playground IS MALWARE: While not specifically designed to be malware, it was designed to inject shell code into a specific process and was designed to read and write to a processes memory space by creating a process handle on the running process. This program also creates an overlay and uses that overlay in a fictitious manner. So please understand that when you run this, RUN THIS IN A VM WITH Microsoft DEFENDER OFF.

B. This playground is developed in: C++20, for Microsoft Windows and uses ImGui as a base for the GUI and implements it using the D3Dx11 graphic libraries to render and draw data onto a screen. There are also many mathematical

algorithms for calculations packed into this application. Do not be surprised when you come across a 3,000 line function purely for math LOL.

C. The playground requires files: This playground should be isolated onto a desktop before you run it. This is because when you run the exploit, the exploit will need to create files and then restart. The way it will do this is by running some commands to download files, it will then exit and you can finally execute it once again and start the challenge. PLEASE ISOLATE IN A SINGLE DIRECTORY THAT HAS NOTHING IN IT.

Questions & Submission

This CTF is designed to be done with a professional and someone who is a SME. A SME is a Subject Matter Expert, HOWEVER, you can also do this on your own as it was designed for that as well!

Someone asked me 'how do I know I have the right answers' usually you can find someone in the SkyPenguin Community to verify that, these are people tagged as reverse engineers or people such as myself who can help you out with the answers! Simply just create a sheet of the mapped questions and answers, then submit them to me to look over!

Other than that, there is no need to just stick to the questions only either, there are MANY things in the base CTF for you to find, I added many rabbit holes that are there to trap you and many different unique parts of the CTF that you can explore!

Conclusion && Summary

This document should have given you a good idea as to how REplay works, what it is, and what these documents are. If you did not, then these documents are going to act as writeup but will not provide the base theory- instead, it will tell you HOW to do it. For theory, go to the resources section in this document to get a list of resources for the theory and building blocks.

~ Thank you for playing REplay- we hope you enjoy :D

Resource Section

For this playground, there are many resources we (SkyPenguinLabs) have written and typed up for you to learn reverse engineering! We also recommend some other channels such as Crow, Ben Eater, Martian Defense, Low Level Learning, Guided Hacking, SkyPenguinLabs, Totally_Not_A_Haxxer, John Hammond!

From SkyPenguinLabs

- Getting around string encryption
- Cracking software
- Conceptual Logic & Theoretical Foundations of Reverse Engineering
- Dissecting Environments With Running Processes

- Utilizing Cheat Engine