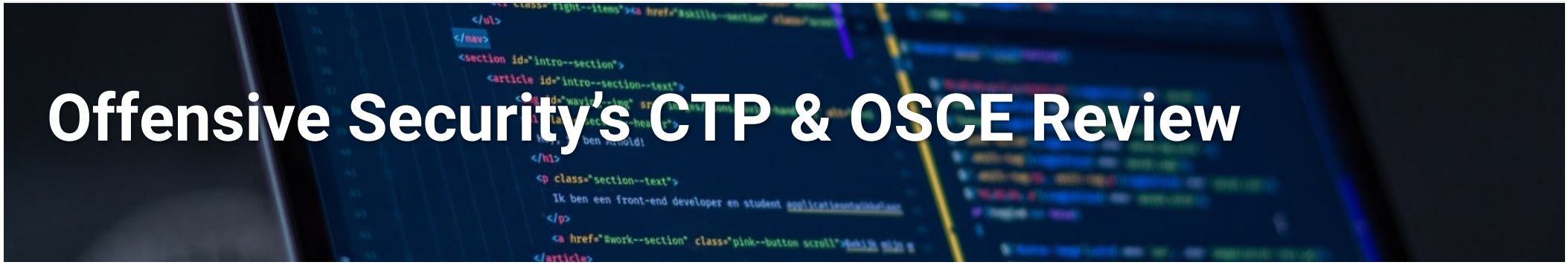


Offensive Security's CTP & OSCE Review

Jack Halon
I like to break into things; both physically and virtually.

Follow

On August 22, 2019 I received yet another one of the most desired emails by aspiring Offensive Security enthusiasts and professionals…

> *Dear Jack,*
>
> *We are happy to inform you that you have successfully completed the Cracking the Perimeter certification exam and have obtained your Offensive Security Certified Expert (OSCE) certification.*

It was finally over! I accomplished what I believed to be, as of yet, the hardest certification exam I have attempted! After a grueling year of training after my OSCP, followed by a month in the lab, and two 48 hour exam retakes, it all paid off at the end - I was finally an OSCE!

Now when people told me that the OSCE was a monster all on its own, I really didn't believe them. Well, that was until I failed my first exam attempt and got a taste for myself. Failing the exam led me to explore new technique and tactics, and took me down a pretty interesting rabbit hole that actually taught me a great deal of new things!

So as I write this post, I want to share my thoughts, experiences, and some tips for those who are aiming to achieve the OSCE. Because trust me when I say… you'll need them!

Background & Experience

Before I delve into the CTP Course and the OSCE, I want to provide you with some information on my background and experience. At time of writing this post I have been in the InfoSec Industry for ~5 years now. I completed my OSCP back in 2017 and detailed my previous background, and experience in my Offensive Security's PWK & OSCP Review blog post. Since then I have learned a great deal of new things and currently work as a Security Consultant and Red Team Operator at NCC Group.

Much of the learning I did to prepare me for the OSCE was done outside of work by reading books, practicing in HackTheBox, and competing in CTF's such as the 2018 Google CTF. A big reason I perused the OSCE was not to learn exploit development but to gain new skills that would make me a better red teamer in terms of being able to develop new tools, bypass anti-virus and EDR, to even learning how to fuzz and build more complex exploits if the need was to arise.

Now, do you need to hold the same experience to pass the OCSE? Absolutely not! I firmly believe that if you passed your OSCP, and took the time to learn more about web application vulnerabilities, x86 assembly and some windows internals, then you would be more than ready to attempt this course! I will delve a bit deeper on the specific studies you need to succeed a little later on, but for now, let's get into the meat of the review!

The CTP Course & Lab

Unlike the OSCP, the OSCE doesn't have a dedicated practice lab. In fact, the CTP course and lab are tied in together - making it more of a walkthrough and "follow along" then a self-taught course, which is then followed by the OSCE exam. You have an option to register for either 30 or 60 days of lab time. Once registered, on your assigned course start date you'll be provided access to download all your course materials. The materials include the ~4-hour Offensive Security CTP course videos, the 145-page CTP PDF course, and your VPN lab access.

When I started my OSCE journey I opted for 30 days as I thought that this would be a decent amount of time to cover the material, and spend some time practicing and honing the techniques taught to me. I don't recommend opting in for 60 days as I believe that you won't get much benefit from the additional days due to the fact that you can pretty much cover the whole course and more in the 30 day time span.

Just as with the OSCP, it's recommended that you go through both the PDF and Videos as the videos sometimes have more details then the PDF. The course teaches some more advanced penetration testing skills and cover topics such as:

- Web Application Attacks (XSS/LFI to RCE)
- Backdooring PE Files
- Bypassing Antivirus Systems

- Bypassing ASLR on Windows Vista
- Crafting and Using Egghunters
- Fuzzing & 0Day Development
- Encoding Shellcode & Bypassing Restrictions
- Attacking Network Infrastructure

It took me about 2 weeks to get through all the materials in the course, not because it was long, but because some of the material and exercises were quite hard - and you really needed to put in some effort to make things work. Even though some of the material was hard, the learning experience was phenomenal. Fact, much of the material is a tad bit outdated - ranging back to the XP and Vista days - but even so the course did an excellent job on teaching you the basics of the exploit development life cycle and its associated techniques.

Now, I must give you a stern warning - just like the PWK course, the CTP will not and does not provide you with everything you need to know, but it does hint you on what you need to learn to pass the exam. So I highly suggest to spend time doing additional research and practice after the course.

Since the course and lab are tied together - I will briefly go over what you can expect. The lab itself only has a total of 4 virtual machines that contain all the tools and software you need to practice and complete the exercises. Now thankfully, unlike the OSCP, you don't have to write up a report for the exercises! =)

Within these four machines you'll practice the different topics stated above, and will be asked to mix and match what you have learned so far to create more complex exploit - such as bypassing a different antivirus, or using a 3-byte overwrite to execute your egg hunter. The exercises are pretty easily followed, but make no mistake, the devil is in the detail and if you

don’t pay attention or spend time doing additional research on the topics, and you’ll have a hard time understanding everything.

After my 30 days were up, I decided to do some more practice before scheduling my exam. I went to exploit-db and looked for simple vulnerabilities in applications that I could practice on. After 2 more weeks of practice and reading blogs, I decided to attempt my OSCE exam and locked in the time for May 26th at 12PM.

The OSCE Exam - Attempt #1

If you thought the OSCP was hard, then you’re in for a surprise. This soul crushing, gut wrenching, 48-hour exam is in all honesty the hardest I ever attempted - and by god did I love it. This exam really makes you demonstrate creative problem solving and your ability to think laterally while performing effectively under pressure to execute attacks in a controlled and focused manner.

For the exam, you are allocated 4 machines, and are allowed RDP/VNC access to 2 of them to build and debug your exploits for 3 of the 4 objectives. As with the OSCP, each machine has certain objectives that you need complete in order for your points to count. Along with that, automated tools like Brup Pro, Metasploit AutoPwn, etc. are restricted, but you can still use Metasploit. In order to pass you need to score 75/90 points. I highly suggest you read the OSCE Exam Guide for more details on what is and isn’t allowed during the exam.

01 x 12PM: Doesn’t Seem Hard!

Finally, May 26th came around. With some early morning breakfast in me and a coffee in my hand, I was sitting at my computer listening to some Monstercat when I received my exam information and VPN access from OffSec. “Let’s do this!” I thought to myself as I read the instructions, noted the objectives and began working on the first “easy” objective.

01 x 3PM: I Fued Up:**

Approximately 3 hours after my initial start time I came to realize something… and that something was that I messed up, badly. I misread the objective and went down a complete rabbit hole that didn't work. I only came to realize my error when I went back and slowly re-read the objective. I was fuming! I wasted three hours of my time to not only realize that I was doing it wrong, but that the solution for the challenge was going to be much more complex then originally thought. I knew what I had to do, but it was going to be time consuming… oh well.

01 x 5PM: It's not working…

After catching my mistake, I took a quick 5 minute breather, got something to drink and off I went working on my exploit. Two hours later I crafted my initial exploit and had it working on a local test machine I set up for myself - but for some reason the exploit would not work on the debugging machine! Why?! After a few more trial and error tests I decided to leave this objective for later and moved onto the next one.

01 x 7PM: You're kidding me, right?

After wasting a lot of time on the first objective I decide to move onto the next "easy" objective. I was 7 hours into the exam and still haven't gotten a single point, but I told myself not to give up - I still had 41 hours left. I began working on my second objective using techniques that I learned from the course and from some blogs that I read. Everything was working as intended and doing what it was supposed to… but the final result was unsuccessful.

"You're kidding me, right?!" I yelled, as I knew that this technique had to work, but something was just not right! I found myself trying new techniques, googling, rebuilding my shellcode, googling some more, and updating my shellcode again, but even that wasn't enough. I couldn't finish the objective. Feeling down, I decided to come back to this later as well and moved onto the third objective.

01 x 8:30PM: Okay, I got this… Guess not.

At this point I opted to take a small break and eat some dinner to relax. I was in a bad spot. 9 hours in and I wasn't making progress but I had hope that I could still pass! After my short break I sat down at my computer, took a deep breath, and off I went attempting the third objective. Within an hour I made some progress, found the vulnerability, and was able to somewhat exploit it but not fully.

For the next 4 hours, I was at a roadblock. I knew how to exploit the vulnerability to get a shell, but I couldn't for the life of me find the exact location needed. I spent hours googling, researching, and testing techniques but nothing worked. At this point I found myself bouncing back and forth between the three challenges and decided that I need to sleep… I was exhausted!

02 x 1AM: Down but not out!

It was late. After 13 hours into the exam, and zero points under my belt I felt defeated. This exam was harder than I thought and at this point I started to come to terms that I might fail. Even though I got no points, I made some progress and the exam was not yet over! I still had a fighting change! With that mind set I went to bed hoping that the next day would be better.

02 x 10AM: Well did you RTFM!?

Day 2 arrived! With the sun shining through my window I woke up at 9AM, ate some breakfast, made some coffee and off I went to work on the exam. I focused my efforts on the first objective again and slowly re-read the objectives to make sure I wasn't doing anything wrong.

Oh… Oops! After re-reading the objectives again I noticed that I missed a critical piece of "additional" information. But surely that wasn't it, right? Oh boy was I wrong! I followed the additional information and within 50 minutes I got a working exploit! I jumped up and screamed for joy! I can't believe I failed to see this in the first place, boy did I felt stupid.

Oh well, with that successful exploit I attained 15 points! I felt ecstatic, filled with new energy I believed that I now had a fighting chance!

02 x 6PM: Two wrongs make a right, I guess…

After getting the first objective, I spent the next six hours jumping between the third and fourth objectives - some fuzzing here, some googling there, some CTP course magic here, the wrong google search here… wait what? Wrong google search? “This isn’t what I wanted! Oh, hey, wait a second…” I thought to myself as I came across a forum post while doing research for how to exploit the third objective. The post wasn’t really what I was looking for, but lo and behold it actually was! Using my new found information and after some trial an error, I was able to obtain my second shell, bringing me up to 30 points!

Yes! I celebrated with a victory lap around the house and thought that maybe I can still pass!

02 x 9PM: Hello Access Violation!

During the time I was working on the third objective, I was fuzzing the forth objective - but had no success. Being a little tired and hungry, I decided to write a script that would automate the fuzzing for me. With the script completed, I kicked it off and stepped away for an hour to eat and rest. Once my break was over I got back and was bestowed by the holy grail of exploit development - “Access Violation”.

I was thrilled! Within 15 minutes I was able to jump into my controlled buffer. I then set off to build a simple python proof of concept to use as my skeleton exploit. Now came the hard part, crafting a working exploit that would bestow me with a shell!

03 x 6AM: That’s impossible!

9 hours went by since I got control of my malicious buffer… 9 hours of hard work, testing shellcode and applying new techniques, but all of that was for nothing. Nothing seemed to work, this objective was literally impossible! There was one major road block that was killing me, the same thing that usually kills all exploit development, and it was right there staring me in the face with its evil little grin.

I didn't know what to do.

03 x 9AM: Defeated!

I was exhausted and only had 3 hours left before the exam was over. With only 30 points under my belt, an unfinished challenge, and an impossible exploit - I knew this was over, I won't pass. I choked up a little, and made up my mind to call it quits. I was defeated, it was over, my dream of becoming an OSCE… shattered.

I sulked my way over to my bed and fell asleep.

Intermission - Back to the Basics

I was quite miserable after failing the exam, it was way harder than previously thought. At first I blamed OffSec for not teaching the required techniques needed to pass, but after talking to some friends, reading the forums, and going through the course again I came to realize something. OffSec did prepare me for the exam, everything that I needed to know and learn more in depth was provided in the course - maybe not directly stated, but it was there.

Here's the beautiful thing about Offensive Security. It's that they first hold your hand through the course and teach you the techniques needed to understand the basics, but after that they throw you into the deep end of the pool to learn on your own. While it might seem a bit brash at first, it really isn't because Offensive Security want's to teach you to be creative, think

laterally and to be detailed oriented. If you pay more attention to the course and exam you'll notice that the devil is in the details and that OffSec pretty much points you in the right direction or gives you hints for the challenges.

With a new found love for Offensive Security, a fresh mindset and the willingness to learn, I opted to take a 2 week break and returned to my studies shortly after.

I first started by going through the exam objectives again, spending about two hours reading them, taking notes and trying to read in between the lines. Unfortunately for me when I did this I noticed things I missed during the exam - which just goes to show that you really need to pay attention during the course and exam.

After taking notes, I did some OSINT (Open Source Intelligence Gathering) on Offensive Security's website for the CTP/OSCE and looked at the "**What competencies will you gain?**" section. A few points stood out to me, such as "**understanding of PE structures**", "**innovative ways of penetrating internal networks**", as well as the "**ability to work through encoding issues and space restrictions**".

Once my notes were completed, I went through the CTP course again and tried to focus on items that I knew would help me on the exam. Afterwards I went online and googled for the topics that I noted previously and to be honesty they really helped me understand the course and objectives in depth. This not only reflected back to the OSCE exam later but also helped me become a better red teamer as I learned new things such as PE Injections, some new internal network pivoting and attack techniques, and more!

While also reading and studying the new found material, I created a simple Windows XP lab that I used to practice on. I tried to craft the lab to resemble the OSCE exam as close as possible. This was fantastic for me as I got the opportunity to test techniques, learn the ins and outs of the OllyDbg debugger, and also got to play with shellcoding and the Windows API locally.

My additional study period lasted about two months and I believe that it was greatly beneficial for me. After all that I felt ready to go after my second OSCE attempt, and locked in the date of August 15th, 2019 at 2PM. Once the date was locked in I actually started working and drafting my OSCE report to save me time if I was to pass the next exam retake.

The OSCE Exam - Attempt #2

Finally, August 15th came around. I woke up at 10AM and followed my regular daily routine. The days leading up to my exam I spent away from the computer enjoying a small vacation, so I was relaxed and well rested. I decided to take a short walk outside as the weather was nice to relax and focus on what I had to do. By 1:30PM I was sitting at my computer reading through all my notes and making mental notes of what I needed to focus on.

I had some proof of concepts created from my previous attempt that I fine-tuned throughout my study period. I knew that these proof of concepts were the answers to objectives, all that was left was to implement them properly. Sure enough, at 2PM I got the email from OffSec with my exam information. After setting everything up, I took a deep breath and dived right into the first objective.

01 x 6PM: I'm on fire!

4 hours after my initial start time and I was on fire as I managed to take out 2.5 of the 4 objectives! This put me in a good position with 45 points under my belt. I did have some issues on the first objective such as my math being off and some stack alignment issues, but that was easily solved thanks to all the prep work I did. It was smooth sailing!

The second objective was taken out shortly after the first. Thanks to all the reading that I did, I knew why my previous attempts didn't work. A shell on the third objective followed directly after. I jumped for joy and did a lap around my house celebrating. Being ecstatic that everything was falling into place, I decided to take my energy and focus on the fourth

“impossible” objective.

01 x 8PM: Peekaboo! I see you!

There’s a reason the forth objective is considered “impossible”, and that’s because it literally forces you to think laterally and like an actual attacker to accomplish the objective. I was able to fuzz and crash the application with my proof of concept, I had control of my malicious buffer - but I still didn’t know how to exploit the vulnerability to get a shell.

So I did the only thing I could, and that was to put on my Red Team face on and play the role of a persistent attacker. After an hour of digging I found something very promising and luckily for me this topic was briefly presented in the course! So I built a quick proof of concept and was able to get a remote shell on the debugging machine. I just found the shell vector… but now the question was, how the heck do I craft this into my exploit?

01 x 11PM: Work dammit!

After finding the exploit vector to obtain a shell for the forth objective I decided to step away for an hour to eat and relax. I had to figure out a way to execute this exploit remotely, but how? This question stuck with me the whole time while I was eating. It wasn’t until me and my dad were talking about doing some custom work on our kitchen that it struck me… CUSTOM! Yes that’s it! How could have I missed this?! I needed to build custom shellcode to get past the restrictions, it was the only viable option!

By 10PM I was at my computer again doing some googling, and crafting custom shellcode to exploit the vulnerability. I crafted some shellcode on my local XP machine and verified that it worked; I finally found the solution to the objective! So I did the next best thing and tested it on the debugger machine hoping that it would work… but it didn’t. That little evil roadblock was still there, staring back at me with more than just a grin this time. I needed to find a way to optimize the shellcode, but attempt after attempt, I still couldn’t get it to work.

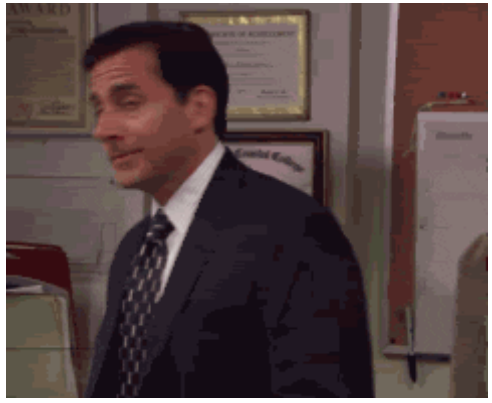
02 x 1AM: Let me sleep on it.

It was getting late. 11 hours into the exam and I knew that I was close to passing, but I kept hitting roadblocks. I was able to do some code optimization but I was still missing a critical piece of information to make the shellcode work properly. I decided to step away for the day and go to sleep. I knew I would pass, so might as well get a good night of sleep… right?

02 x 10AM: Ever heard of backups?

The 2nd day started off very well for me actually. I woke up at 9AM, followed my daily routine, and by 10AM I was at my computer again getting ready to tackle the final challenge! I started up my VM and was greeted by a "*Oh no! Something has gone wrong. A problem has occurred and the system can't recover*" message.

My reaction was something along the lines of…



I had to recover my data somehow! This is a VM right? So that means I have a snapshot somewhere, right? Wrong! I never backed up the data on my exam VM… lovely. Fortunately for me I was able to boot the VM into recovery mode and figured out that the error occurred in Xorgs. Thankfully that was easily fixed and I was able to log back into my system after an hour. Phew, crisis averted.

Maybe this a good time to remind you to back up your data or take snapshots of your VM's!

02 x 2PM: Not so fast cowboy!

With the data crisis averted, I got back to work trying hard to optimize my shellcode. I made some good progress, some CTP magic here, some googling here and I was really close! But yet again I was at a roadblock, I was missing something critical. What could it be? I started to lose hope after a few hours and even considered giving up - but I told myself that I will "**Try Harder**" and continued to push forward.

02 x 6PM: The devil IS in the details…

I was so close to passing that I could taste it! I only needed to tweak my shellcode a little and I would get my exploit to work, but I couldn't put my finger on what optimization I needed. At this point I decided to take a step back and walked to the kitchen to make some food and tea. As I sat in the kitchen my dad came and asked me how the exam was going, to which I responded with my current status. After listening to me he told me that I was probably over thinking it, and to relax, explaining that I will figure it out.

Overthinking? Impossible! I was thinking laterally… that's what OffSec wanted, right? Okay, I was in the same situation during my OSCP exam, so I returned back to my computer and decided to start from the basics. After some step by step debugging I noticed something simple that I previously overlooked. At first I didn't think much of it until I started to think about how I can use this in my shellcode. After some thinking and fiddling around, I made a small adjustment to my shellcode, held my breath, and kicked it off against the server.

Boom… shell! I jumped out of my chair and screamed! I got it, it worked! This shell bestowed me with an additional 30 points, brining me up to 75 points - enough to pass! I was so hyped but also disappointed with myself that I missed such a simple piece of information. Still I did it, I passed!

Wrapping it Up

With 75 points under my belt I decided to call it quits. I was tired and had a very busy weekend ahead of me, so I decided to finish up my report. I went back to gather all my screenshots, validate the exam requirements, and by 9PM I sent the report to OffSec which was about 98 pages long.

I received a response 5 days later from Offensive Security saying that I passed. It was finally over, I did it, I was an OSCE!

I'm honestly at a loss for words. This exam was very challenging and taught me a lot of new skills and techniques that I actually utilize day to day on my red team engagements. Sure, this course is more exploit development focused but it still teaches critical thinking and technical skills that can be utilized at your day to day job. I sincerely want to thank OffSec for this amazing experience and opportunity!

Tips & Recommendations

I know that many of you who will be reading this post will ask for tips/recommendations on either preparing to take the OSCE or on how/what to do during the exam. Well not to worry - in this section I will break down and include a lot of the materials I used to prepare for the OSCE as well as some tips/tricks to use for the exam.

Prerequisites:

In the CTP course, OffSec states that you need to understand the following fundamentals to take the course:

> *Cracking the Perimeter is an advanced course and requires prior knowledge of Windows exploitation techniques. You should be comfortable in OllyDbg and understand concepts such as shellcode encoding, use of the Metasploit Framework, and Linux at large.*

Honestly speaking, this is very broad and there are quite a few more skills that you need to have to pass this course. I suggest taking a look at the full syallbus to get a better idea of what you need to know.

These skills are actually tested when you register for the CTP course. You will be provided a link to a web application and will need to pass a two stage registration challenge to even complete the registration. If for any reason you are having trouble completing the challenge, then you need to take a step back and go learn some more basics because if you can't pass the registration challenge then you are not ready to attempt the course, nonetheless the OSCE exam.

If you are somewhat unfamiliar with x86 assembly, shellcoding, web application vulnerabilities, and basic exploitation, then here are some links to help you learn that required material:

- x86 Assembly Basics
  - Introductory Intel x86: Architecture, Assembly, Applications, & Alliteration
  - Hacking: The Art of Exploitation, 2nd Edition
  - NASM Tutorial
- Exploitation Basics
  - Exploit writing tutorial part 1 : Stack Based Overflows

    - Exploit writing tutorial part 2 : Stack Based Overflows – jumping to shellcode
    - Exploit writing tutorial part 3 : SEH Based Exploits
    - Introduction To Software Exploits
- Shellcoding Basics
    - The Shellcoder's Handbook: Discovering and Exploiting Security Holes
    - Exploit writing tutorial part 9 : Introduction to Win32 shellcoding

I highly suggest that you complete all of the material above before attempting to register for the CTP/OSCE, trust me - you will thank me later if you do!

Practice:

Now that you have a fundamental understanding of the basics, you need to practice… a lot! If you follow the material above, then you should be able to pass the registration challenge and start the CTP course. After the course, I suggest you take the time to read and study additional material before attempting the OSCE exam.

The following materials below will help you practice and expand your skills.

- Practice Binaries and Exploits
    - Solar FTP Server 2.1.1 - PASV Buffer Overflow
    - Easy File Management Web Server 5.3 - Remote Stack Buffer Overflow
    - Easy File Sharing FTP Server 3.5 - Remote Stack Buffer Overflow
    - freeFTPd 1.0.10 - 'PASS' Remote Buffer Overflow (SEH)

    - KenWard's Zipper 1.400 - Local Buffer Overflow (2)
    - QuickZip 4.60.019 (Windows XP SP3) - Local Stack Buffer Overflow
    - All Vulnserver Vulnerabilities
- Study Materials & Guides
    - Portable Executable File Format
    - Understanding PE Structure, The Layman's Way – Malware Analysis Part 2
    - Art of Anti Detection 2 – PE Backdoor Manufacturing
    - CVV #1: Local File Inclusion
    - Assembly Primer For Hackers: Part 1-4
    - Jumping with Bad Chars
    - QuickZip Stack BOF 0day: A box of Chocolates
    - QuickZip Stack BOF : A box of Chocolates – Part 2
    - FuzzySecurity - Windows Exploit Development Tutorial Series
    - Basics of Windows Shellcode Writing
    - Create a custom shellcode using System() function
    - Vulnserver KSTET WS2_32 Recv Function Re-Use

This might seem like a ton of material at first, but do know that these topics will overlap and you will have a better understanding of each after the course. Don't expect to know or learn this in one week! It will take you at least 2-3 months after your OSCP to be in a good position to go after your OSCE.

Exam Tips:

As with everything, there are always certain things that you should know and be doing during the OSCE Exam, these following tips should help you stay on focus and to stray away from rabbit holes.

1. Read the objectives on the OSCE exam slowly, and VERY carefully.
2. Pay attention to all the little details in the CTP course, the answers to parts of the exam are in there!
3. Pay attention to all your registers in the debugger. What do they store, where do they point, etc.
4. If you're ever doing any calculations on an x86 stack such as aligning pointers, make sure that the calculations are divisible by 4, otherwise you'll have stack alignment issues!
5. Learn some hexadecimal arithmetic. The OSCE forum has a good post explaining it!
6. Access Violation after your shellcode? Check to make sure you didn't overwrite any important data in your registers or on the stack!
7. Sometimes a direct approach isn't feasible. Can you chain attacks to get the final result?
8. Take frequent breaks. Opt for 15 minute break every 2 hours.
9. Eat and drink! Make time for Lunch, and Dinner. Your brain needs food to function.
10. Organize your notes, take screenshots, and document everything!
11. You have 48 hours for the exam. Make sure you sleep at least 8 hours. There's plenty of time to finish!
12. Don't give up to easily, and most importantly… "**Try Harder!**".

Updated: August 22, 2019

SHARE ON

Twitter Facebook G+ Google+ in LinkedIn

Previous Next

LEAVE A COMMENT

3 Comments jhalon 1 Login

Recommend Tweet Share Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Lasha T. • 2 months ago • edited

Wow, great review. I was reading it slowly and it seems you very much liked it. I'm thinking about CTP training course for about 1 year but didn't purchase it yet. Main reason why I'm waiting is that training material is outdated and I somewhat think that same result can be achieved by reading corelan/fuzzysec tutorials but after reading many reviews I think I'm wrong and that OSCE gives much more than just reading Corelan tutorials. I'm deeply motivated in exploit development and intending to take that career path. I have little experience in custom shellcoding,c/c++,x86 assembly. I've got OSCP cert last year and working in security for about 3 year. Know little bit about building ROP chains but on linux, little bit SEH, ASLR bypassing(but on linux). In summary I have basic background in those topics but don't have much practical experience.

Now, 2 questions :)

1) Do you still recommend this training course?
2) You didn't mention SLAE(Security Tube Assembly Expert) training course. In every review it's mentioned that it will be huge help in OSCE exam. But you didn't mentioned it. Is there any reason for that? :)

P.S Sorry for my English,it isn't my native language :)

Jack Halon Mod ➚ Lasha T. • 2 months ago

Hey Lasha,

First of all thanks for reading and thanks for the kind words! Second of all, yes I very much did enjoy the course as it not only explained some basic concepts, but it also took me down my own rabbit holes that forced me to learn many new things, which is great!

Now to answers your questions.

1) I do still recommend the OSCE. Yes, while it is a little out-dated, it still teaches the core concepts of windows exploitation which is different from Linux. While the assembly part stays the same, the function calls and shellcoding is completely different since it is Windows. Sure, you can read Corelan tutorials and other blogs - but this course forces you to be hands on and the Exam literally forces you to think laterally. It makes you see things from a different perspective on the many possibilities you can solve/hack something. If you want to focus on Exploit Dev, then certainly take the OSCE and follow up with the OSEE at Blackhat.

2. I didn't mention the SLAE since it's Linux focused, and the OSCE is Windows focused - so while the assembly training is beneficial, it's really not going to help you with the Windows side of things. I honestly believe that the "Art of Exploitation" and

"Shellcoding Handbook" was worth way more then the SLAE since the books touch on Windows a bit. At the same time, there are ton of training courses and blogs you can utilize (posted above) that directly work with Windows. I guess it just depends on where you want your training/knowledge to go. If you want to learn about Linux exploitation too and then hop into windows, then sure take

see more

1 ^ | v • Reply • Share ›

Lasha T. ➚ Jack Halon • a month ago

Thank you, I finally decided to buy CTP course, but before that I have to prepare well :)

^ | v • Reply • Share ›