Corelan Team

:: Knowledge is not an object, it's a flow ::

How to become a pentester

Corelan Team (corelanc0d3r) · Tuesday, October 13th, 2015

Intro

I receive a lot of emails. (Please don't make it worse, thanks!) Unfortunately I don't have as much spare time as I used to, or would like to, so I often have no other choice than to redirect questions to our forums or our IRC channel (#corelan on freenode), hoping that other members of the community will jump in and help me answer those questions.

One of the most frequently asked question is "how do I become a penetration tester".

Depending on whom you ask this question, you may get different results or may be told to take a specific approach. With this post, I am trying to formulate my views on this question (with a focus on the process and not so much on the technical aspect), in an attempt to hopefully provide a good starting point for those that find themselves in a similar situation.

For the record, I am not a penetration tester… but I try to apply common sense (seasoned with a touch of plain logic) to challenges and pretty much all situations in life. Don't hesitate to provide feedback, suggest changes or tell me to STFU and GTFO. Any motivated additions or changes to this post are more than welcome, and I'll update this page as needed.

Where to start … or ?

There are a couple of approaches to getting started with information security. Approaches change as time flies by, technology changes, new platforms are designed & implemented, etc… All of this influences what is supposed to be the latest "hot" topic to dive into. Before doing so, regardless of the approach you take, there are 2 fundamental questions you should ask yourself:

1. How much effort (time, …) am I willing to put into this?

I have been working on exploit development for many years. Truth is that I don't have an advanced background in systems programming, not been trained in latest technologies. All it took was a lot of time & dedication, a strong will to learn and absorb new things. You can learn as fast as your brain is able to process and remember, and practise is able to lock down into your mind, converting the knowledge into experience. Some people learn fast, others need more time. Nothing wrong with both approaches, but being self-aware, self-conscious about your abilities and being realistic about the time you're able & willing to invest into supporting the learning process is important. Another factor to take into consideration is your balance between the various powers that play in life. If you have a family, make sure to talk with your family members and find a good balance between spending time with them, and spending time on this learning experience. Even if you're alone, get out from time to time. Don't rush into things, but try to dose and apply a healthy time-consumption model. Let your brain process, think, and take your time.

This "time" factor brings us to the second point:

2. What is my goal?

People set goals all the time. Goals can be far away in terms of knowledge & experience needed, they can be even so far away that they look more like a dream than anything else. The good news is that it is OK to be ambitious and to have dreams. However, from my experience, it will be easier to achieve your ambition by breaking the dream into smaller objectives, into smaller goals. So, my (quite limited) definition of a goal or target is something realistic, something you can achieve, using one or more steps (actions). We can discuss about semantics, words and definitions, but that's not the point of my statement. Let's apply the dream/goals/targets logic to the topic of this post. Let's assume it is your dream to become a pentester. The concept of being a pentester is quite exciting, but it's quite vague at the same time. What exactly do you want to do? What type of pens .. errr.. targets do you want to test. Why do you want to do it ?

If you don't have the answer yet, it might be useful to talk to pentesters and ask what type of work they do, and see if you are interested. Let's say your ideal scenario involves testing the overall security level of corporate networks, perform audits against web applications and do something with "mobile devices" because that's what people told you. Perhaps it's social engineering. It doesn't matter what you select, those are your goals. They are part of the "pentester" definition, but you've just broken down the dream into goals & targets.

Why am I being so philosophical about it? Well, becoming a pentester who is specialized in all types of audits may not be realistic after all. Technologies change so fast that it may not be possible to become an expert at everything, right away. Trying to understand & master everything would not be a realistic goal. It might still be a dream though, and you might eventually get there. It all depends on how much effort you're willing to invest. Taking this one step further, don't be discouraged nor too enthousiastic by what other people say. Timings are personal, there is no good or bad. The god news is : you can do whatever you want, some things will take more time than others. It's not about "IF" you can do it, it's about "WHEN", and how realistic this "WHEN" is.

Being a pentester does not mean being good at using tools either. It's about being able to understand how things work, how things are configured, what mistakes people make and how to find those weaknesses by being creative. Being a pentester is not about launching Metasploit against the internet.

A couple of years ago, I got interested in photography. After taking many pictures using my smartphone and being encouraged by family members that kept repeating how great my pictures were, I decided to buy a DSLR. Guess what. Buying a better camera or lens doesn't make me a better photographer. In fact, it made things worse because I didn't understand how light works, how a camera can be tuned to deal with the light and how we can influence light to get better pictures. Smartphones are designed so you wouldn't have to think about it. Beginners mistake. The reality is that learning how things work is time consuming, frustrating… but it will be rewarding in the end. I'm still not a good photographer, but I don't mind admitting it either. I consider this to be a journey and at least I'm determined to understand the fundamentals first; to try and to make mistakes a lot; and not to be afraid to ask for help.

So, this brings me back to the original question: "Where to start". It should be clear by now that perhaps you should try to answer "where do you want to end" first, as this will tell you where to start. Don't worry, even if you make mistakes, even if you find out that you picked the wrong ("less exciting") targets, you still win. Any knowledge you gain is valuable to a certain extent and can be helpful along the way.

There is a second way to look at the "goal". You can also define your goal as "your ability to generate an income". Let's assume, for the sake of this post, that you would like to make money as a pentester. This means that you may have to select certain technical objectives (knowledge) that will provide economical value. This could be driven by popularity of certain types of technology (web apps, for example); or relatively new areas (Internet Of Things, Mobile, …). So, even if you want to do many things (and should be looking at a broader perspective), there is a big chance that you will have to specialize in specific areas.

Before continuing, I'd like to @thelightcosine quoting HD Moore: "If you don't think you're a n00b, you're not trying hard enough". Challenge yourself. Try to learn more about something, try to gently push your limits, but do it in a realistic way. Never give up. It's a painful, long, but very rewarding journey.

Ok, got it. So, where to start ?

Assuming you know where you want to end, and you have a realistic plan that involves dedicating time and efforts; what should you do with your time?

Before talking about possible roads, I'd like to briefly mention something that will become the most important part of your journey. It's YOU & your attitude & mindset. You'll be the one doing the work. You're the one that sets goals and wants to start working. You're the one that will make this work. But it requires a specific attitude to do so. It's the so-called "hacker" attitude. There are many definitions of the word hacker; but most of them boil down to this: A desire to understand how things really work, so you can optimize/change the behaviour, or apply the understanding to bend the rules of the game. Hackers tend to break stuff; penetration testers tend to break stuff too. The goal should be to break stuff in order to come up with solutions on how to improve it. If the purpose is to break stuff so you can prove you can break stuff, and systems/people have flaws... Newsflash: we already know that. You're wasting your time. You're a breaker, not a hacker. If you truly want to be a hacker, break stuff because you want to fix it, make it better. The word hacker can be applied to many disciplines. It doesn't need to be tied to computers, it can be applied to science in general as well. In fact, without hackers, we would not have medicines, or technological evolutions.

Be critical about what you see. Try to understand what you see. Ask questions and don't accept the "I don't know, that's how it works, that's what someone told me, just accept it and move on". Ask yourself the question "what would I do if I had to design X or Y". Putting your thought process into the mindset of someone else will help you understand why things work the way they work, how they were designed, and how people use them the way they use them. Using empathy & understanding that other people have different views will broaden your understanding of things, which in return will help you to discover strengths and weaknesses.

Being a hacker is not technical. It's a mindset, it's psychology. It's beautiful. It's very powerful. (Sidenote: I am truly blessed to be able to spend time with extremely intelligent hackers all over the world. Each of us has the potential to change something, to improve something. We can even change the world; if we would organize ourselves in a better way. Maybe it's time for selecting a new dream, a dream that involves hacking the world. Anyways...)

Ok Ok, cut the crap, where to start?

Hold on. We're almost there. Before giving you some hints on how to approach your journey, I'd like to share some thoughts on asking questions. In fact, unless you're born with all the answers already, you'll probably end up asking questions. Even if you know what your ultimate goal is, you may not know how to get there, or what is needed to get there. The only way to figure out is by asking questions. Interestingly enough, the way you ask a question and the type of questions you ask, will determine whether you get the answer you need or not.

I often hang out in various channels on IRC and I've been subscribed to a bunch of mailing lists for a long time. I see people asking questions and other people trying to answer questions on a daily basis. You would think that asking or answering questions is a trivial thing, but interesting enough, people get yelled at, kick-banned from IRC channels, or humiliated in public just because they were trying to find the answer to something they don't know. Lots of people end up frustrated because they failed at getting a satisfactory answer, and other people get frustrated because they felt they were wasting their time.

What exactly causes this conflict and how can both parties be more effective at asking and answering questions, and hopefully avoiding painful situations? Most of the items below are based on cases where direct interaction is possible (IRC, Instant Messaging, and so on), but they can be generalized very easily and are applicable to any form of communication (email, support form, forum).

Asking questions

Asking questions is very easy. Asking a good question seems to be far from easy. What does it take for a question to elicit a valuable answer and how can you avoid that people will start throwing tables, bicycles and elevators at you because you just wanted to get an answer? I've tried to gather some ideas on how to be more efficient at asking questions and increasing your chance on not only getting an answer, but also getting a helpful answer. A few years ago, I did a small survey on Twitter to discover what people believe are the ingredients of a "bad" question. The results included:

- No indicators that the person asking the question did any of their own research or attempted to find an answer, Googling Bing or Binging Google;
- When the question is ambiguous;
- When you forget to ask the question;
- Massive preamble to get to the question.

There are a couple of things you can do to avoid common pitfalls and getting your question labeled as "bad". For starters, I don't think bad questions exist. There's always a reason for a question, or logic behind a question. It just may not be clear what it is exactly, because of poor communication or other reasons, but that doesn't make the question bad. I'm listing some ideas below, in no particular order.

Avoid the obvious answer.

Think about your question. How easy would it be to find the answer online, on Wikipedia, via a simple Google search or by reading product documentation? If you're lazy, don't expect people to show appreciation for that. Trust me, being honest about your laziness, won't help either. If you get kickbanned from IRC because you are lazy and you advertise or admit it, you probably deserve it.

Show that you deserve an answer and anticipate.

Do your homework. See what you can find about your problem on the Internet, try a few things yourself and document what you did. Be prepared to show what you did. Be honest and accurate. People are more likely going to help if you show that you have tried and willing to try more. As soon as people sense that you just want to be spoon-fed, your support channel is going to blow up in your face. People might ask you to reproduce the steps you took to end up in your current situation, so you can anticipate to that. Put your documentation and procedure on Pastebin or Pastie before asking the question and be prepared to provide a link to your documentation when needed.

Don't leave out vital information or be embarrassed about something you did even if you think you shouldn't have done it. It might very well be part of the problem and if you want a solution, you'd better be honest. Be as factual as possible in describing your problem and don't let your thought process take over. Describing the symptoms and the exact steps needed to reproduce the symptoms will work better than explaining what you think the problem is. You may have missed something obvious and if you don't share all the facts, people may not be able to discover what really went wrong.

If you're trying a procedure or a tool and you get an error message, there's a chance that other people have encountered the same situation. Google for the error message (leaving out specific parts such as IP addresses, and so on) and see what you can find yourself. Make sure to construct your question in a way that would make people believe you just want a gentle push in the right direction. Instead of asking "I don't understand how this works" or "I want to hack Gmail" you could also ask "What would you recommend I should learn to do this or that", or "Does anyone have any recommended sources about SQL injection against an Oracle database"? You're trying to achieve the same goal and you're pretty much asking for the same thing, but you're shifting the focus to the process of learning and finding a solution instead of drawing attention to the problem or goal itself. It's perfectly fine to ask for "some pointers in the right direction".There's a famous Chinese proverb that says "Give a man a fish and you feed him for a day. Teach a man to fish and you feed him for a lifetime". If you have a problem, you may choose the fastest and easiest path and get someone to fix it for you by giving you the solution right away. If you're taught how to troubleshoot a problem, you may be able to increase your insight and improve your ability to prevent and fix future issues yourself. The more you focus your question on what you can do and should do yourself, the easier it will be to convince someone to help you out. What also often helps is to explain your problem to a friend first. In some cases, explaining the problem and allowing a friend to try to understand the problem might also expose the solution. This has happened to me and helped me in numerous cases. In short, the more work you put into finding a solution yourself, the more precise your question will be and people will appreciate that.

Break things apart and be critical to yourself.

Before asking a question, break down your question into technical layers and components. Do you fully understand the other components or prerequisites needed to reproduce your problem? If you are asking a question about attacking a remote computer, make sure you understand the networking layer and have checked that everything is set up correctly between your device and the remote computer. If you don't know enough about networking, you shouldn't be attacking something that uses the network.

Don't forget or ignore that others had to go through the same learning experience as you did and had to work to for it. If your question suggests that you just want to skip learning fundamental knowledge, people may be offended because you're basically disrespecting the hard work hey have done in the past.

Consider there is a possibility that you still have a long way to go, and that "understanding" something doesn't necessarily mean you are applying it in the right way. Instead of asking why a certain technique for a certain case doesn't work, you may need to wonder whether you fully understand the

technique or not.

Don't start with an apology.

There's no need to apologize for not knowing something. It usually only makes people raise an eyebrow for a brief moment and move on. You should only apologize to yourself for not asking proper questions or for not being prepared to work or learn. Don't apologize for your lack of speaking or understanding English. People don't care. If it's bad, they'll notice it's bad. You have the opportunity to improve your English by reading documents, interacting with people, so you might as well do something about it. If you're unsure, prepare your question in advance and relax. If you're asking a good and well-prepared question, nobody will even notice. Of course, if support is available in your native language, that should be your first option. I'm not trying to say that apologizing is a bad thing. It's a token of maturity and respect, and can be very powerful in conflict situations and negotiations. I'm just not convinced it will help you getting an answer. Don't hide behind the fact that you don't speak a language very well in order to insult people. I've seen this happen before: somebody walks into an IRC channel, starts by apologizing for his bad English, and then blatantly insults everyone in the room. Bad idea. Excuses are not a magic patch for stupidity.

Be nice, polite and don't be impatient.

Even if your question is urgent, if you decide to rely on community support, you also have to realize and accept that people have lives too and may have other priorities than answering your question. Asking "why" you're not getting an answer after 10 minutes may trigger people to ignore you, so don't do that. Maybe you need to rethink your question instead, or find another source of information or support. Public forums and IRC channels are not private support channels. Don't expect the entire world population to care about your problem, so make sure not to flood the channel with your issues. Although IRC and Instant Messenger tools allow for direct communication, it doesn't guarantee that the communication will take place when you want it and at the speed you want it. Timezones are real and the people that have the answer may be asleep. Accept it.

Even if you think a tool is terribly broken, focus on what you may potentially have done wrong. Others will appreciate if you ask "what I have done wrong" or "what you should do different" even if it's a genuine bug in an application or tool. If you start by saying you believe a certain tool is broken or ask "why" a tool is broken, you're going to draw attention to yourself in a bad way. One bug in a tool doesn't make the tool bad, so don't disrespect the work of many people who may be reading your comments.You're the one to open a discussion, and you'll never get a second chance to make the first impression. The tone for the rest of the discussion is going to be set as soon as you initiate the communication, so be nice and respectful.

Only make jokes if you're sure others will understand and appreciate the joke, and won't be offended by it. You don't know who's behind the computer at the other side, so avoid anything that is potentially offensive, sexual oriented, inspired by religion. Understand that there are different cultures, different points of view, different people. None of them are good or bad, better or worse. A lot depends on how well you know the people you are addressing. (Sidenote: this is also why a "hacker culture" doesn't really exist. We're all individuals, with different backgrounds, cultures, beliefs. It doesn't matter what you look like, what language you speak, what clothes you wear. or what the color is of your skin. You're a hacker by the things you do, and why you do them.)

Jokes about yourself or your situation are an exception of course, and are often perceived well. Complaining about your crappy Internet connection and how it makes you consider sending faxes again, or make a comment about how the fact that your dog is so fat that it interferes the Wi-Fi signal in the house are just a few examples on how to help set a friendly tone.

Be creative, don't overdo and choose your timings carefully. Maybe you are impressed by the skills of the people you're going to address when asking your question. Don't start yelling how "l33t" or "pro" you think they are, and how easy it would be for those people to answer your question. This may set off some red flags and make people think you're a troll. Be yourself, act normal, be polite and you'll be fine.

Make it easy to answer.

If your question is too broad or contains too many elements that are open for interpretation, ambiguous, questionable or vague, don't expect anyone to take a few weeks of holidays to show you around.

Choose your audience.

What would be the best place to find an answer to your question, and who would be the right person to answer your question? Forums and IRC channels often focus on a certain topic, so try to pick the right medium and channel to ask your question. If your question is related with the use of a tool, it might be a good idea to find other users instead of sending your questions to the tool developers right away (or use the support mechanisms they made available).

Ask the question.

Don't start by asking if you can ask a question. There's no need to ask if someone is online or available to answer your question. Just ask the bloody question. If someone is online and if your question makes sense, you'll get an answer. Unless you are picking up where you left off from a previous discussion, don't target a specific person. Other people might ignore your question if you're suggesting that only a particular person can answer the question. If you want to get an answer, make sure it is perfectly clear what the question is. If you simply state "I have a problem with X or Y", "This tool doesn't work" or "My exploit got detected by Antivirus", you're technically just making a statement and not asking a question. Asking what you did wrong or what you can do to make something work (better) is more likely going to get you what you want. Even if you have to explain the context of your problem, try to keep it short, to the point and move to asking the question as soon as you can.

Listen, interact and seek clarification if needed.

Try to understand the answer and don't reply to it right away. If someone tells you to investigate something else, do it. Don't keep hammering away, ignoring advice that was given to you. If an answer is not clear, ask for clarification, but do it in a way that suggests you're trying to learn (process) and not trying to be spoon-fed (solution). As explained earlier, the better you are at setting the right tone and suggesting you only want some hints in the right direction, the higher the chance someone will help you. When asking for clarification, try to formulate it in a way that explains what you did and didn't understand. Rephrasing or summarizing certain parts often helps to show what part of the answer was clear, and what requires further clarification. Rephrase/Consider starting a new question with:

- If I understand you correctly,
- What you're saying is that...
- In other words,
- What if
- Does that mean...

In the event you didn't understand the answer at all, don't be afraid to say so. Ask if that person could rephrase, explain something in a different way or elaborate on a certain part of the answer, and do it in a polite manner.

Be grateful & give back when you can.

If someone tried to help you, tell that person you appreciate the help, even if it didn't fully answer your question. Realize that a lot of people have other things to do than answering questions. If they are trying to help, it's because they want to help, despite tight schedules, deadlines at work and other priorities. Credit them for taking the time. Do it in a short, clear and efficient way. Mention how the answer was helpful. If you have the impression the question you asked is a very common question, and your gut feeling says that the person who answered the question is actually getting sick and tired of having to address the issue over and over, you may want to consider lending him a hand. Document your question and the solution and put it online somewhere. It will help you to understand the cause and the solution, you can ask the person who helped you to verify your document to make sure it's accurate, and you can help other people by simply pointing them to your online document. It demonstrates you want to learn, you've listened and you want to give back. Don't wait to share until you have all the answers. Guess what, you'll always find another question.

Answering questions

Asking good questions is definitely an art that requires a bit of preparation. Answering questions, if you really want to help someone, is not trivial either. Although some cases may suggest the opposite, there's no such thing as dedicated askers and dedicated answerers (not sure if that is even a valid word). No matter how experienced you are, you might still find yourself at both sides of the story from time to time. When you're in the position to be able to answer a question, you really are in a unique situation. Think about it, you have the power to decide whether you want to answer the question or not, and on top of that, you can choose how to answer the question, which will have a direct impact on whether the answer will be valuable or not. If you decide to take the time to answer a question with the intent of helping somebody, you might as well do things right. Maybe some of the following guidelines may be of assistance:

Be nice.

There is a reason why the question was asked in a certain way. You should be able to sense the difference between somebody who's after a quick win and somebody who is genuine, who really wants to learn, but doesn't know how to communicate well. When you're not sure, grant the person the benefit of the doubt, you can still yell at him later. There's an easy way to help someone if they were too vague or didn't make themselves clear. Simply rephrase the question and ask if that is what they want to know, or just tell him his question didn't make any sense and ask that person to be more specific. It will make sure you properly understand the question and it will show the person how to properly phrase a question next time.

There's no reason to make fun of someone or make him/her feel bad. He or she already admitted being in the dark about something.

Think before you answer, ask for more info.

Do you really understand the question? Is your answer going to be helpful? Ask for clarification if the question is not clear. Rephrase; ask for an example. Try to reproduce the steps needed to come to a problem and ask for more details and documentation.

Don't answer because you have to.

Only answer a question because you want to help, and have the time to help. Although the first question may seem reasonable, it might get worse very easily. If you decide to step in to help someone, at least you'll have to try to get the asker onto the right path, and it's hard to estimate how much time you'll need for that upfront.If you do things right and understand the question well, it shouldn't be too difficult or time consuming to answer the question right away, or point the asker to the correct resources.

Reply with a question.

Tricky one. Some people enjoy doing this all the time, which can totally freak out people and destroy normal communication, so make sure to use this technique in specific cases only. There certainly is a lot of value in replying with a question, providing that the question suggests a solution, or aims at getting more information. Let's take a look at a quick example:

Question: "I ran an exploit against a target computer and the exploit says I was not able to get a reverse shell."

Many things could be wrong with this scenario, making it hard to answer the question in just a few words. Asking a few short questions might put the guy back at work, trying to get more details on why his procedure didn't work. You could for instance ask if both hosts are able to connect to each other. This suggests that there might be a network related issue. It shows that you understand the individual layers related with the act of exploiting a remote computer, and you help him using a structural approach to troubleshooting this kind of problems. Asking questions about the question itself might reveal underlying reasons and motives. Sometimes people are too embarrassed to admit something, because they can almost sense they are doing it wrong, or perhaps they know they are doing something illegal. By asking specific questions about why they want to do something, or suggest them to do things differently (in a way that wouldn't involve potential illegal activity), might give you some helpful information about that person and if his intentions are legit or not. If someone is having problems running an exploit against a machine on the Internet, you may want to suggest him to simulate the procedure in a private lab. If the person chooses to ignore your suggestion and insists he wants to do it over the Internet, you're almost positive he's up to no good.Try to discover what the person is trying to do. If someone asks if it would be possible to do a certain thing, ask him what he's trying to achieve. Ideally, it will force the person to explain and reveal any underlying motives.

Be honest.

If you're not sure about the answer, just say so. There's nothing wrong with admitting you don't know something for sure. Guessing is acceptable, as long as you make clear you are guessing. It may suggest possible solutions and perhaps put the person on the right track already.

Stimulate, don't burn.

You can demonstrate your skills by providing a helpful answer, not by showing off, emphasizing how smart you are. Based on how specific the question is, and how it reflects the level of knowledge possessed by the inquirer, you can adjust the level of detail of your answer accordingly. If you need to explain that something is wrong or bad, don't forget to explain why it is wrong or bad and give pointers on how to avoid or fix the issue. You don't need to answer questions in detail, as if you're reading a tutorial to them. A gentle push in the right direction is often good enough to stimulate the learning process. It's ok to put someone on the right path and point him to the resources he should study if he wants to make progress in the future, but don't just throw URLs at him. If the other person understands why he needs to learn something, it will be easier to convince him to take the effort to do so. Of course, if the same person just continues to ask questions and doesn't want to take the time to learn things properly, your answers are obviously not going to help anymore, and that person probably doesn't want to be helped. He just wants someone to do the work for him. In that case, there's no value in trying. Wait until the person has figured out he needs to work for it, and ignore him until he proves it.

Language.

English is an important language in international IT or Infosec communities. However, that doesn't mean everybody is a native English speaker or even remotely close to that. The use of common and universal terminology is perfectly fine, but try to keep your sentences as simple as possible. We don't want to make the poor guy suffer more than necessary, do we? If you notice during the conversation that the other guy didn't really understand your answer, challenge him and verify that he understood what you said. Try to figure out if it's a language issue or knowledge issue. If it's the first time both of you are talking, it might be acceptable to just ask the asker if he understood what you said, so you can adjust your vocabulary if needed. See if you can give an example to clarify, or just ask a question about your explanation. If you're in a kind mood, you could say something that would suggest that it's ok to ask more questions if needed, which should break the ice if the inquirer is a bit shy.

Spot the troll.

Surely, there are people with too much time on their hands, without a real life, trying to waste everyone's time by asking a combination of stupid and intelligent questions, just for the fun of it. A small minority of these so-called trolls actually master the subtleties involved very well and might make it sound like they have a real question, and then continue to combine silly questions with good questions. If done well, these folks might actually keep you busy for a while. Luckily, most trolls have bad ninja skills and can be easily recognized. Wasting the time of brave volunteers and people who really want to help is not very nice. Getting kickbanned, they should.

Provide feedback.

If nothing worked and you have a few moments of time, explain why a certain question or remark didn't work. Maybe the asker said something disrespectful or suggested that he doesn't really want to learn things properly. Worst case, he'll ignore your advice and you can choose to ignore him too. Best case, he'll learn from your feedback and approach things differently next time.

Update: Check out http://xyproblem.info/

Sigh. Ok. Please, where to start ?

Horizontal or vertical ?

I don't really care whether you prefer to stand up, or to lay flat when learning new things. What I mean with the "horizontal or vertical" title is: should you focus on learning a broad variety of things first (horizontal), or should you dive directly into the area you're interested in (vertical)?

Good question. There are definitely pros and cons in both scenarios, there are more opinions than people. Yours truly has been blessed with opinions as well, so I'll share my personal view. Understanding the big picture first is useful. If your goal is to become a web application pentester, it would probably make sense to learn all layers involved, ranging from operating systems, networking, web server & application technologies, commonly used database platforms and common development languages. This is a big animal. The amount of information you're interested in, usually depends on what you need. At the same time, the better you understand how things work, the easier it will be to understand how to bend the rules. My recommendation is : try to understand as much as you can about the various layers first. Don't be impatient and dive into the nitty gritty details of finding bugs or exploiting right away. Especially the availability of tools will make your hands itchy and lowers the hurdles to start attacking systems right away. Always keep in mind that tools are not magic. They simply automate things. The better you understand what they do, the easier it will be to use them. Don't get me wrong, tools are useful. Just don't use them until you understand what they do, how to configure them, how to use them properly.

So, I believe there is a lot of value in trying to understand the system engineering aspect of systems. Understand how things communicate, how things are set up, secured, operate. Don't overdo either. You don't need to be an IP expert that understands all the RFC specification. You probably need more than what you need to abuse it. You'll need enough to use it and abuse it.

Furthermore, understand that you can take a phased approach. You don't need to be a BGP routing expert to perform web application testing. It doesn't hurt if you are, but you can still learn it when you're ready to expand your horizon and dive into other aspects of security audits. Be realistic in the goals you set, and try to accurately determine the prerequisites needed to get there. Ask multiple opinions if you're not sure and don't be afraid to learn too much rather than not enough.

How to learn?

There are many ways to learn new things, some of them are quite personal (= as in: they only work for some of you, and not for others). Some people are able to learn new things by reading a book or blog post. Some need to visualise things, and others need someone to explain things in a video or face-to-face setting. There are solutions for every methodology. You can buy books or read publications online. You can take classes (online or in real life), and you can find lots of online challenges to practise your new skills.

There is nothing wrong with any of these approaches, as long as you understand what works best for you, so you can adapt your strategy accordingly. The common aspect of all of these learning methodologies is to get practise. Trying out things for yourself (guided or non-guided) will make it easier to remember and to eventually transform the knowledge into understanding & experience.

In any case, having up a virtual lab environment can be extremely useful. Nowadays, Virtualization technology is now available for most common platforms, it's cheap/free and allows a great deal of flexibility. VirtualBox, VMWare, Parallels, Xen, Hyper-V are just a few examples.

Although this is not a catch-all advise, you'll get a long way by installing a Windows and a *Nix/Linux system. Of course, understanding how to manage & operate these systems is fundamentally important. You don't want to spend your time fighting the tool that are supposed to support your learning experience.

Spoon-feeding

Spoon-feeding sounds like something we do to babies, right? If you ask experienced people whether spoon-feeding is right or wrong, I bet most of them will tell you it's bad. I believe the answer is not black & white. It depends. First of all, we've all been spoon-fed. (Or at least most of us). This is what our parents did when we were not able to feed ourselves. This is what teachers do when you are entirely new to something. This is what we should be doing to put people on the right track. We've all been told certain things to allow us to practise, get better, and get to the next phase. There is a thin line between stimulating in a supportive way, and leaving people behind with no help whatsoever.

In "Leadership and the One Minute Manager – Increasing effectiveness through situational leadership II", Ken Blanchard explains 4 different "development levels". One of these levels is defined by a high commitment and low competence. This may be the place where you are right now. You're quite excited about learning something new, but you have no idea on where to start. For scenario's like this, some spoon-feeding can be useful. It doesn't mean that someone else will do all of the hard work for you, but simply being told to "go figure it out" without giving directed pointers or hints is not useful either. As soon as you learn more (and become more competent), you'll discover that there is much more to learn. At this point, you may find yourself becoming less committed, because you're starting to realise there is still a long road ahead (which can be quite demotivating). This is normal too. At this point, spoon-feeding won't help. In this case, coaching is more appropriate. Asking the right questions will force people to think, to apply the knowledge they already have, and look for answers. If they're stuck after all, and have no way to discover answers themselves, perhaps it's time to take one step back and get some detailed help after all. So – please be careful when being negative about spoon-feeding. The situation (development level) determines whether it's the right approach or not.

Anything else?

No, not really. Thanks for asking. Time to start drawing the tree that will become your journey.

1. Networking & operating systems

I would suggest to start by learning how systems work and communicate. Try to get a good understanding of TCP/IP, OSI layers, ephemeral & server ports, routing, port forwarding, NAT, firewalling, etc. You'll need it when trying to connect to targets, you'll need it to use tools, and you'll need it to configure your environment to allow your security audits to be successful.

You'll also need to be able to manage & operate common operating systems. Together with networking, this should be your primary starting point. Most of us are familiar with one operating system, but it "doesn't hurt" (=understatement) to understand how to use and configure both Windows and Linux/Unix. You should become fluent in setting up networking configurations, basic security features & implementations, using both command line utilities and GUI tools. Start to use these systems as your main desktop, use them on a daily basis in order to force you to become familiar with them.

I know, I know, you'd like to start attacking systems right away, without spending too much "overhead", right ? I fully understand that it sounds very exciting to start using portscanners or other tools right away, but what's the point in using the tools if you don't know what the output of the tools mean? Even worse, you could easily cause damage if you don't know what you're doing.

2. Multi-purpose resources

Next, try to get a broad understanding of the attack landscape. Maybe you already made up your mind about becoming a web application pentester, but it still doesn't hurt to understand what else is out there. There are many resources on this topic, but I decided to list the most important ones (at least the ones that cover a wide spectrum of skills):

- A hands-on introduction to hacking
- Grey Hat Hacking – The Ethical Hackers Handbook
- "Hacking Exposed" series
- Professional Penetration Testing

(If you feel an important resource is missing, let me know. Oh, and to the publishers/authors: if you would like to provide our readers with a discount coupon code, please contact me :-))

Aside from getting a better view on the landscape, you'll learn a few things about pentesting methodologies & approaches, including the difficult art of translating technical findings into something a customer or business can use and understand. Being a pentester does involve paperwork too. Just sayin'. Again, apply the true hacker mindset. Break stuff because you want to make it better, not because you want to break it. Without truly trying to "make things better" in reality, you're just a breaker. (So – don't complain about mistakes others made. Think & fix. Add value. Learn how to secure, harden and protect as well.)

3. Scripting & Tools

No matter how long you look at it, you'll end up using scripts and tools that automate certain things. You may even want to change existing tools or write your own to make your life easier. After all, that's what scripts are for. They are a tool, not a goal. Becoming familiar with scripting languages such as python and ruby is a must. You don't need to be an expert, you'll get better as you start to use them. Understanding some C / C++ can be useful too, as some people tend to write tools in lower-level languages (mostly for performance reasons). In any case, understanding what a tool does is more important than writing your own. Writing your own can be useful, because it proves that you understand what needs to be done.

This is probably a good time to start using a so-called "penetration distro", a pre-configured system that contains a large series of security assessment tools. Trying to create your own system from scratch can be helpful, it's also time consuming and probably not necessary until you fully master the ones that already exist. Kali Linux is one of the most commonly used/popular distributions. It has a large userbase and is well supported by most tool developers.

In addition to the more attacker-oriented tools, it's also a good idea to expand your lab environment and include local and online systems that are designed to be vulnerable, allowing to test your knowledge, using the tools available. If you're into web application security, a good place to start is https://www.pentesterlab.com/exercises or http://www.amanhardikar.com/mindmaps/PracticeUrls.html. You'll find more links on the websites listed below.

4. Dive deeper

Only when you're ready, pick the target or targets you want, and create a realistic action plan to achieve the goal. Some topics will take days, others will take weeks, months, maybe years to understand. Take your time, one step at a time. For each type of target, you'll find specific resources (books, online publications, classes, virtual labs, etc).

Some good resources include the websites listed here:

- https://code.google.com/p/pentest-bookmarks/wiki/BookmarksList
- http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html
- http://wiki.securityweekly.com/wiki/index.php/Penetration_Testing_Tips_&_Tricks
- https://github.com/enaqx/awesome-pentest
- https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Contents

- http://www.dfir.org/?q=node/8
- https://www.owasp.org/index.php/The_OWASP_Testing_Framework
- http://www.pentest-standard.org

(Don't forget to ask questions as you work your way through resources & materials.)

Although I encourage everyone to find their own area of expertise, if you are serious about becoming a professional pentester, you will have to learn a thing or two about web application security. After all, a lot of companies use web application services to serve applications to their employees, customers, partners, suppliers, etc. Since web applications need to be exposed to the outside world in a lot of cases, they are also an important target (and a way in for criminals). Understanding how HTTP works, how web applications are developed, secured, and how underlying database platforms work, will make up a big part of the journey. Your mission, if you choose to accept it, is to find the dependencies and prerequisites that are required to dive deeper into the area you would like to focus on, and to translate those into an action plan. Again, simply ask questions whenever needed.

5. Listen, engage, help

Use social media to follow influencers, people that have inspired others, and just more experienced people in general. Engage with people, be nice. Ask questions and help as you learn.

If you have the opportunity to attend Infosec conferences/seminars: please do so. It's a great way to meet more experienced people and talk with them. Ask them what they are working on. Share what you are doing and ask for tips. Ask them who they look up to, or were inspired by and check them out too. Become part of the community. (Oh btw – conferences are great places to find a new job too)

Open a website/blog, share your findings. Sure, you may not be the first person to go a particular road... but you won't be the last either. Environments & technology change, so as you apply your newly acquired knowledge, try to keep track of your progress & document how it applies to latest technology. In fact, you'll probably end up taking notes as you learn anyway. You might as well structure them and put them online for others to see. Potential employers may not be so interested in *what* you post, but rather focus on how you structure your notes, your thoughts, and your potential innovative approach to things. Make your work visible and teach it to others.

Don't be afraid to make mistakes. You'll get there. All it needs is time and efforts.

Good luck.

6. Don't be stupid

Unless you're attacking your own system, or you have obtained proper permission to do so, attacking a system (on a network, locally, physically, etc) is a crime. Don't be stupid.

What's next?

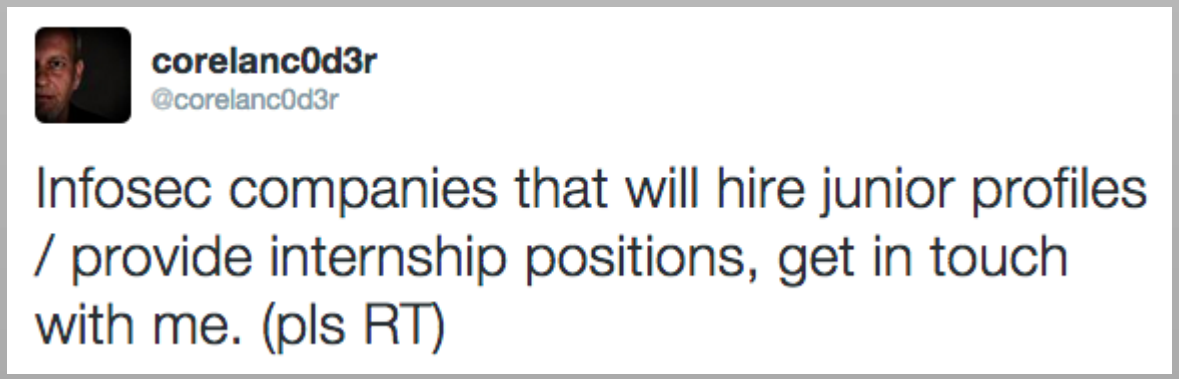
What to consider when trying to get a job as pentester

In all honesty, it may not be so easy to break into information security and get a job as a pentester. In fact, it's pretty hard to get into that area professionally (unless you have a desire and business plan that justifies becoming self employed). In general, companies tend to prefer hiring experienced pentesters. After all, most companies want to get "return on investment" as fast as possible, which means they don't really want to invest too much time in training you and becoming more experienced first before they can rely on you to take on assignments.

Not all is lost though. Some companies may offer (summer) internships or may give junior profiles a break... but nothing beats experience. Agreed, it sounds like a catch22 situation. I guess the key is to find a way to gain more "experience", or "credible ability".

You can gain experience by playing CTFs, by testing your skills in simulated environments and/or get certified. Becoming an "Offensive Security" certified penetration tester or passing SANS exams can be a good investment, as it is well regarded in the industry... and there are certainly other similar "titles" you can earn while you're at it. I agree, proven experience/knowledge is more important than a title (and some titles don't even guarantee knowledge), but unfortunately you may not be able to get a seat at the job interview table without a title in the first place.

So, to encourage companies to speak up, I decided to tweet this:



so... if you're reading this post and you work at a company that is willing to hire (relatively) inexperienced pentesters (at least, without a lot of professional experience), please let me know (i.e. get me a formal statement, a link to your website that contains more information) and I'll add the link to this post. If you have a (summer) internship program, please let me know too. Any help is much appreciated.

In fact, I strongly believe that companies tend to underestimate the true power of having a junior profile in the team. Benefits include a fresh view on challenges (fresh = less impacted by routine), pushing everyone to stretch their comfort zones, and encouraging senior profiles to share their knowledge and experience. Everybody wins.

So far, the following people/companies responded & allowed me to post a link here (or tweeted their policy on the matter):

Europe

1. KPN // the Netherlands
2. ERNW // Germany (internships) : ernw.de / info@ernw.de

3. Kyos // Switzerland (internships)
4. NetXP // Paris, France (junior/interns) : netxp.fr / recrutement@netxp.fr
5. Nettitude // UK & USA (interns)
6. Solucom // Paris, France (juniors/interns) : solucom.net/careers
7. EdgeScan // Ireland
8. Securify // the Netherlands (Juniors) : https://www.securify.nl/jobs
9. Toreon // Belgium : https://www.toreon.com/category/news/ – https://twitter.com/toreon_BE
10. ESET // The Netherlands (Sliedrecht) : donny@eset.nl
11. Cogiceo // Paris, France (internships): http://www.cogiceo.com/fr/carrieres/
12. 7Elements (UK)
13. TheSecurityFactory // Belgium : junior profiles – @securityfactory info@thesecurityfactory.be
14. CERT-XMCO // Paris, France (junior/intern) – recrutement@xmco.fr –https://www.xmco.fr/recrutement/
15. Fox-IT // the Netherlands (junior profiles) – vacatures@fox-it.com
16. Madison Gurkha // the Netherlands (junior profiles)
17. Intrinsec // Paris, France (junior & interns, pentesting/SOC) : https://www.intrinsec.com/fr/form/rejoignez-nous-securite-informatique.html
18. Kudelski Security // Switzerland (junior / interns) – @KudelskiSec
19. MDSec // UK (junior/interns) – https://www.mdsec.co.uk/
20. EY (Ernst & Young) // Belgium (junior profiles) – http://www.ey.com/BE/en/Careers – oana.butnariu@be.ey.com
21. DearBytes // The Netherlands (junior/intern) – personeelszaken@dearbytes.nl
22. TwelveSec // all over Europe (junior)
23. Stormshield // France (junior) – https://www.stormshield.eu/stormshield/carrieres/
24. NViso // Belgium (junior profiles) – https://nviso.be/hello/jobs
25. PwC // Belgium (junior profiles) – http://careers.pwc.be
26. Northwave // Netherlands (junior profiles) – https://www.northwave.nl/jobs
27. Infoguard.ch // Switzerland (juniors in infosec (SOC Operators, Security Analysts, Security Engineers, Pentesters)) – https://infoguard.ch/de/home/ – umberto.annino (at) infoguard.ch
28. www.scip.ch // Switzerland (junior profiles) – stfr (at) scip.ch
29. SBA Research // Austria (Junior Researchers)

USA

1. Milton Security // USA (junior positions for recent Veterans, internships)
2. NCC Group // USA (junior positions) : https://www.nccgroup.trust/us/about-us/careers/security-consulting-careers/ – careers-na@nccgroup.trust
3. Attifyme (Remote security internships on Mobile & IoT)
4. SalesForce // US (summer internships for BS/MS students) : http://salesforce.careermount.com/candidate/job_search/advanced/results/1?sort_dir=desc&industry=5571&sort_field=post_date
5. iSIGHT Partners //US (interns)
6. MWR Infosecurity (UK, SA, SG, UAE & US (soon)) – interns // careers
7. ISE (interns) : https://twitter.com/lisa_a_green
8. LinkedIn Internships
9. YearUp
10. NetSPI : https://www.netspi.com/about/careers/featured-current-openings
11. Facebook (Internships)
12. Facebook University (FBU)
13. Blackbird Technologies : mkaplan@blackbirdtech.com
14. Veracode (entry level people)
15. Motorola Solutions (entry level SOC analysts/interns)
16. GuidePoint Security (junior/intern)
17. Bishop Fox (internships)
18. Tenable (summer interns): http://www.tenable.com/careers
19. CyberPoint
20. WhiteHat Security (junior/intern)
21. root9B // USA (junior devs) – https://www.root9b.com/careers
22. Sentryo (junior/internship)
23. BlueCanopy (junior profiles) – http://www.bluecanopy.com/category/pages/careers

Russia

1. SakurityNetwork // Russia (Juniors) : http://sakurity.com/jobs

Various locations

1. IOActive // various locations (Juniors/Interns) : careers@ioactive.com
2. MWRLabs //various locations (juniors/interns)
3. Google // various locations (Juniors/ Interns)

(Check out the twitter thread, there may be some other companies that haven't agreed on posting a link yet, or just don't want me to post a link here).

Warning: Before getting too excited & sending messages to all of the above, think about it for a moment. These companies won't have unlimited seats. They're not looking forward to processing millions of applications either. Be creative. What will you do to make sure your profile will get the attention it deserves? What added value will you bring to the company? Put in some efforts, make sure your message stands out amongst the others.

Also, please keep in mind that the offers may be limited in time and number of people. On the other hand, there may be other companies out there that might want to give you a break. Use social media, use your network. Don't give up.

Finally, don't contact me to help you find a job. I am not a recruiter, and don't want to become a middle man either. I'd like to encourage companies to speak up and for you to take some initiatives too. You can do this. Be smart.

What to expect as part of your life as a pentester

Well... I don't know, I'm not a pentester :) ... but I wouldn't expect to be able to hide in a basement for years. Depending on your geographical location and your customer base, you'll probably end up having to travel to clients, have meetings (remote, on location), write reports, articulate technical findings into actionable information, present findings, work with clients to fix issues ...

Exciting times ahead ! :)

Reddit

Reddit has some threads related with infosec and hiring:

- https://www.reddit.com/r/netsec (Infosec related topics)
- https://www.reddit.com/r/netsec/comments/3n5qne/rnetsecs_q4_2015_information_security_hiring (quarterly hiring thread, link changes every quarter). Some companies will hire junior profiles from that thread.

Outro

I am certainly not the only person who would like to share a view on getting a job in infosec. In fact, hacks4pancakes posted an article which pretty much deals with the same topic (but maybe presents a different/new angle): http://tisiphone.net/2015/10/12/starting-an-infosec-career-the-megamix-chapters-1-3/ Go check it out, I believe our posts complement each other well.

This entry was posted
on Tuesday, October 13th, 2015 at 2:30 pm and is filed under 001_Security, Penetration testing, Web Application Security
You can follow any responses to this entry through the Comments (RSS) feed. You can leave a response, or trackback from your own site.