

# A Rant About Hacking Labs

By Thomas Wilhelm, ISSMP, CISSP, SCSECA, SCNA

One of the more frequent questions I see on EH-Net pertains to creating pentest labs. Individuals new to the topic of hacking often have a limited understanding of what type of equipment is required, or how to go about setting up a lab to practice all of the cool attacks they have watched on YouTube. Details on how to get started using a single system and virtual machines are numerous &ndash; including some I have done. However, I think there is one question not being asked enough when discussing hacking labs&hellip; &ldquo;Why do you want a lab?&rdquo;

Most people create a lab containing a single host system and include virtual images of various Operating Systems. Unknowingly they have just restricted themselves to a very finite portion of real-world hacking &ndash; system attacks. I&rsquo;m not even sure I can classify these &ldquo;system attacks&rdquo; as internal (within the corporate network) or external (Internet-facing services), due to a lack of support systems typically found in corporate networks. Absent are the routers, firewalls, IDS/IPSeS, windows networks, switches, etc. Without these, we don&rsquo;t really have a good example of what someone might face during a real pentest, nor do we create an effective learning environment.

Discuss in Forums {mos\_smf\_discuss:Opinions}

## Asking the Right Question

“Why?” If we start with that simple question, we can better define exactly what we need to do when building a lab. Even the answer “to learn” is insufficient to truly understand how we need to build out our lab. For example, if we start with the idea of external pentesting, we have to discuss firewall rules. If we jump into internal pentesting, we need to talk about ARP spoofing. Neither of these is valid or available in a single host system using virtual images.

So let’s approach this topic a little differently (but I should warn you first; I’m putting on my “professor’s cap” soon). Instead of answering questions such as “what do I need in order to create a pentest lab,” let’s help newcomers rephrase the question to “what do I need to create an (external|internal) pentest lab.” If we start with the concept of an external pentest, we need to inject some sort of firewall into our configuration; whether it is software or hardware-based is immaterial at this point. Also, to add realism, we should create a DMZ subnet and place our target there. If we start with the concept of an internal pentest, we need to include subnets and windows domains, switches and routers; and we need to configure them just like a real-world corporation would.

Sounds like a lot of work and cost? Well, it certainly can be, but doesn’t have to be. Ok, so let me put on my “professor’s cap” and espouse the virtues of setting up a proper lab by asking a few rhetorical questions:

- First Question – What if you could reconfigure the packet routing within the entire network (think “private community strings”); would that be a “game over” in a pentest?

- Second Question – What if you could simply sniff usernames and passwords on the subnet, which gave you access to a financial system with no known exploitable vulnerabilities? Again, game over?

- Last Question – What’s more likely to be exploitable on an internal subnet, a system or ARP caches? Based on experience, it’s definitely ARP.

If we don’t include (at a minimum) these types of scenarios in our lab, we are missing out on the opportunity to learn how to exploit systems and grab sensitive data using trivial attacks. In addition, we are doing the heavy lifting first, by targeting the systems instead of the infrastructure that supports those systems. We also don’t practice or preach to newcomers the use of a solid methodology either, when we tell them to jump directly into system attacks. It almost feels like we are making newcomers learn the harder stuff first.

NOTE: I have to admit, however, I am just as guilty of pushing system-based attacks first on students. It’s easier to teach how to run Nmap than it is to set up an ARP-spoofing scenario. It’s quicker to demonstrate a password brute force attack than it is to walk through reconfiguring a router using an SNMP private community string. Perhaps it’s not the right thing to do, but it is certainly easier. Maybe we should approach things differently.

## Answering the Right Question

So, how should we respond to the question, "What do I need to create an (external|internal) pentest lab?" Let's start with the foundation — the network. In Figure 1, we have the HackingDojo.com remote pentest lab. If we look at it closely, we see that the network closely simulates a real-world scenario along with some unrealistic scenarios. Realistically, we have a dedicated firewall (PIX), a couple screening routers, internal networks, and a DMZ network. Unrealistically, we also have a VMware server outside of any firewall. Functionally, this provides the students with multiple scenarios from which to learn.

### Figure 1 - HackingDojo.com Remote Lab

Additionally, this lab configuration gives students an opportunity to learn the details behind the different attack scenarios. Let me give an example.

In some classes at the Hacking Dojo, we talk about SNMP and private community strings, and how to use them to modify the exploitable router. This means that the students can (and should) modify the device however they desire (thankfully, we have configuration backups). In the higher level classes, students are given admin access to the routers, so they can see how to configure them. By understanding how to configure a network device, they can better understand how to exploit it. Thus they can better talk to their clients (when they move into a pentest job) on how to mitigate the vulnerabilities found on their network appliances.

So, should we push this type of configuration on those individuals new to pentesting? I would say "yes" for a couple reasons. First is what I mentioned before, there are a lot of trivial attacks that are not possible in a simple "host / VM" solution. Second is that it forces the student to make a serious effort into the realm of pentesting through a commitment of time, money, and training.

As many in the field can attest, pentesting is not a casual profession, and it isn't something that can be learned in a weekend. Professional pentesters are constantly exposed to new systems, applications, and vulnerabilities; and to properly understand how to exploit them takes time and effort. We also spend our own money on books, software, and courses, so that we can improve our skills and add value to ourselves and our clients. Arguments against spending the time and money to learn how to conduct a proper pentest in a proper lab, should be met with concern by professional penetration testers, since it risks increasing the misconceptions of what pentesting really is, and how a pentest project should be conducted.

It's NOT about the Money&hellip;

So, am I saying newcomers need to break the bank getting set up with a lab? Absolutely not. Figure 2 is a picture of the

network devices I have in the Hacking Dojo lab. The total cost of these network devices was \$600 and was purchased second-hand. If there is a need to be more frugal, I could easily have spent half that amount and obtained (again, just the network devices) what is shown in Figure 1 in red, which would have been just as effective as a pentest lab. Besides eBay, there are numerous CCNA certification kits you can buy, such as the one at the following URL: <http://www.certificationkits.com/ccna-certification-kits/>. Throw in a low-end system or two, loaded with a virtual engine, and you can replicate what is seen in Figure 1. Again, if you want to keep it cheap, it's amazing what can be accomplished with only a few hundred dollars. I've used eeePCs as host systems, and see PCs at BestBuy for two to three hundred dollars frequently, especially on the "returns" table.

Figure 2 – Snapshot Taken During Deployment of Network Devices for the HackingDojo.com Lab

Is \$600 dollars for network devices and a couple systems too expensive? It depends. If the newcomer has a casual interest in hacking, then it probably is. If the newcomer is serious about being a professional, who requires a broader and deeper understanding of vulnerability exploitation, \$600 is just getting started. Just like most other activities, the equipment costs money. As a reality check, I spend more money on my daughter's swimming lessons in a year than I did on the lab in Figure 1. In short, cost should not be an issue.

It's NOT about the Time&hellip;

Once we get past the concept of cost, we are faced with the amount of time it takes to get everything setup in the lab. The network devices are not preconfigured, leaving us to configure it ourselves. Unless we have some sort of background in network administration, this alone could be a daunting task. However, like I mentioned earlier, how would you explain mitigation to a client, if you don't know how to configure the appliance in the first place?

Pentesting is not simply hacking and producing a report – we have to interact with clients. Oftentimes, the pentest engineer is the front-man when explaining how to conduct attacks and what options exist to secure the client's network. Without that ability to convey both the offensive and defensive side of security, you may leave your client's security posture in a weak state. This means, we need to set up an effective lab. I cannot count the number of times I heard a speaker at DefCon describe their attack by starting out with a description of what they did to set up their lab beforehand. Every organization I've been in had a lab as well. It's simply a requirement, and setting one up, along with all the additional functionality (SNMP, Dynamic Arp Inspection (DIA), LDAP server, DNS server, etc.) is an enormous training opportunity, despite the time it takes to learn them.

Even though it doesn't seem like hacking, learning how to deploy systems and devices is a necessary requirement. Just like it is necessary to know SQL database commands before you can conduct effective SQL injections on a web system, it is necessary to understand all the protocols and applications within the network to be effective pentesters even before a single attack. Otherwise we simply waste our time and the time of our clients.

It's ALL About the Training

One of my sayings I constantly convey to my students is "pentesting is 90% learning and 10% doing." This

is a stark contrast to when I was a system administrator, when the percentages were reversed. Every day I come across something new, and, in order to learn how to conduct an effective attack, I often need to do research. In some cases, I need to set up a system with a specific type of software to see what happens when I launch an exploit. Other times, I need to read white papers to see the purpose behind an implementation in a hardware device, so I understand whether or not it will cause problems during my attack or ways to avoid the device altogether. And I still ask questions of others with more experience in a topic than I have. It's all about learning new skills and becoming a more effective pentester. And to learn new skills, I need the right equipment and software; there just isn't any way around it.

## Conclusion

Hacking has become a profession, and as such we should treat it as one. When newcomers enter the field asking what they need with regards to a lab, we should be honest and explain the end-goal first, so they can better understand exactly what is required of them long-term. Starting with a single system loaded with a virtual engine is an option, but newcomers should be aware of the limitations of that configuration. As a community, we should be forthcoming of all that will be required in order to become a professional in this field and not be shy to say what the costs are.

We should also practice what we preach. There have been many threads on discussion boards talking about different attacks at an almost academic level, but few of them show details and specifics. If we are asked "how does one use a community string to attack a router," we should be able to provide screenshots or snippets of exactly how it's done via our own labs. Simply too many discussions are limited to system attacks, which are oftentimes the more difficult attack vectors during a pentest. We should also be sharing our configurations, especially those created on network devices. When I set up the Hacking Dojo lab, I simply could not find a single configuration example on how to do so, and had to create it from scratch.

Another milestone we need to reach as a community is creating "network hacking" scenarios with different configurations that can be dropped into devices. That will provide a challenge to those who are interested in learning new techniques which require network devices to be navigated or hacked. Similar to the De-ICE discs, we need to share network / system configurations whose hacking solutions are unknown and can be simply uploaded into a detailed, prescribed lab.

By treating hacking as a profession, practicing what we preach, and extending our knowledge to others, we can make serious advances in both our own skills as professional pentesters, and improve the skills of the community as a whole. When confronted with the question "what do I need to create a pentest lab," we should give a complete answer, provide some direction, and offer challenges that are realistic — not simply give them a myopic view of pentesting by telling them to create a virtual lab on a single system. If we give a complete answer, we all come out ahead.

## About the Author

Thomas Wilhelm has been involved in Information Security since 1990, where he served in the Army for eight years as a Signals Intelligence Analyst / Russian Linguist / Cryptanalyst, and is a Doctoral student who holds Masters Degrees in both Computer Science and Management. Thomas founded the HackingDojo.com hacker training program, and has

written numerous articles and books; the latest being "Ninja Hacking" published by Syngress. A new publication is in the works that will include downloadable network device configurations that can be used to practice network hacking techniques, similar to those mentioned in this article.