For hackers and freedom fighters engaged in illegal activity,you may want to consider the latter a bit. Once you make ingress and launch any manner of offensive action, you have escalated the legal ramifications of your trespass by multiple magnitudes.

Also remember that the probability of you getting caught and prosecuted is never 0.00%: you have to be prepared, you have to be careful, you have to be patient and you have to prepare contingencies.

I use a measurement/assessment of risk vs. reward to make each action within the network as efficient as possible; by percentages,losing a queen to take a rook is generally a loser's bet.

The best way I've learned to temper a careful approach is with an old sales slogan (" Always be closing the deal", which I modified to "Always be advancing your position(s)").

I try as much as possible to engage a target as a stalking, ambush predator: I move carefully and try to use the environment to hide myself as I seek to exploit the target/objectives lack of awareness.

I work to remain patient and identify/quantify as many of the variables of the current environment/situation as possible.

Sometimes the best decision you can make is to slow down or hold your current position for a bit; watching Tcpdump or Wireshark while thinking on a better move is still advancing your position.

To lower the probability of detection (whenever possible) I attempt to attack, enumerate or probe from an obfuscated position.

Configuring your attack host/node for the highest probability of situational anonymity (using tunneling, proxies, encapsulation ,etc.) is infinitely useful in pentesting, hacking and/or general security/privacy.

Mastering the manipulation of proxy, tunneling and encapsulation protocols (which involves a deep understanding of networking/TCP/UDP) almost lends you quasi-magical invisibility and teleportation powers when involved in network penetration.

Obfuscation itself is one of 10,000 reasons why experience/knowledge in the disciplines of networking, OS and programming combined with security research are such huge advantages (and another reason why if you take up this path you may never stop learning).

Learn to use every tool you can, but more importantly, learn why the tool works. If you work in/at exploitation long enough, the principles governing the tools will help you exploit a box someday,regardless of whether you use that particular tool to get the wanted/needed result…

Knowledge/experience over tool use is especially important today: regardless of what many sites say, you will not find many enterprise/corporate networks today (as a professional penetration tester at least) where there are gross configurations/deployments leading to an easy, out of the box (deploy tool== Meterpreter) exploitation.

When training for a fight, professional mixed martial artists put themselves in the worst possible positions so they react properly when the fight is underway.

Eventually, training/practicing your exploitation/research techniques the same way will be a huge boon in engagements, POCs (or in the wild). I especially like to round difficulty up during research; it is difficult for someone else to minimize your findings if you have added (and circumvented) greater security measures than the norm (rather than having reduced them).

Most of my exploitation of networks in the last couple years have been a process of discovering network misconfigurations and weaknesses (especially in Windows firewall, Programs and Features, LGPO/GPO policies and/or IE/Internet Options within Window Domains/Networks) or information leaks that I locate online or through DNS enumeration that ultimately leads to my gaining access to a host.

From there, remote exploitation (toward post exploitation/privilege escalation/pivoting) will often occur This is largely when knowledge of things such as Powershell (leveraged by itself or tools like Powersploit/CrackMapExec/PsExec/Empire) become invaluable (in Windows networks).

I have actually been finding easier remote exploits when attacking Linux/Unix boxes in enterprise networks (finding Solaris with Apache Tomcat during enumeration still springs hope eternal in my human breast).

Many (actually, maybe all) of these companies are/were new at deploying Unix/Linux boxes in their networks and were making some serious mistakes with deployment.

Enumeration is the most important part of an engagement to me. You should get used to enumeration without automated tools; I love Nmap, but many times it is not feasible to usewithin the customer’s network (network overhead issues, the chance of detection

by IIDS, the chance of breaking PLCs or other embedded devices, etc.).

In cases where you are on the customer’s network, tools like Wireshark, Tcpdump, knowledge of networking protocols/ports and banner grabbing are your friends.

For those engagements where you first need to gain access to the network, you definitely have more room for running some louder tools:

I love Fierce (and DNS enumeration in general) as it often presents my way in.

Google dorking is still also an incredible tool, as is Firefox with the right set of extensions (Hackbar, Tamperdata, Wappalyzer, BuiltWIth, Uppity, IP Address and DOmain Information, etc,.).

Who loves Dirbuster in these cirumstances? This carbon/caffeine based lifeform right here.

Whether you are pentesting, bughunting or hacking/freedom fighting, a paid Shodan subscription will($50) is worth every cent. The capacity to make exacting, accurate searches for greater than five pages has helped me in more engagements/bughunts than I can remember.

When I am explaining why a config/setting/LGPO /GPO (etc.) is a security risk to a client or my fellow employees, I like to explain that many of the advantages I look for in my environment are most often advantages that are needlessly provided to me.

If it does not break key functionality or seriously impede efficiency/development time, than it is in their best interest to deny me as many advantages as possible, even when the advantages appear as if they are minutia.

When dealing with a client or non-security fellow employees,you should work to create a relationship of mutual help and teamwork.

I am not there to rub their noses in there crap; I am there to help improve their security so the company can prosper. This is partially a customer service gig where solutions (remediation/counter measures) are more beneficial to the customer than the exploitation itself.

Whenever possible, I like to end the post-exploitation/penetration test conversation/meeting/presentation with the attitude that I am here to help fix these issues , how can WE best close these gaps? How can I help make your (or our) company safer, so that we can become
more prosperous?

I personally despise Microsoft (and many proprietary products/companies) on many levels, but when it comes to work, I am platform agnostic. Whatever tool is needed to complete the mission is the tool I am going to employ.

However, whenever possible without jeopardizing the mission, I am going to employ an Open Source/Unix/Linux-centric solution.

I work hard to show my company the value in Open Source. The way to show that value isn’t to be the super Unix/Linux/GPL neckbeard who constantly bemoans proprietary software./platforms.

The best way (for me), is to show how effective the strategy involving the Open Source tool is. Then, in my report, I explain the business hook of using Open Source (if the tool is free for commercial use).

I am sensitive to companies taking Open Source tools and turning them into something proprietary.

However, if I can make my company (which is both huge and almost universally recognized as ethical, which is rare) see the value in Open Source, I know they will eventually incorporate Open Source into the support packages for their products (which they have while keeping the tools ad the license in tact).

This than spreads the value of Open Source to smallercompanies who see it being trusted by a much larger company.

I have tens of thousands of dollars worth of licenses atmy disposal. However, I will never use tools like Nexpose, Nessus, Canvas orMetasploit Pro unless the project, client, or a governing body specificallyrequire them.

I believe these tools develop poor habits. Obviously, if a project such as evaluating an entire domain of IP/hosts for vulnerabilities is my task, I am going to use Nessus. However, (whenever a time/project permits, which they most often do) I am going to evaluate the findings (and search for other vulnerabilities) manually.

The ultimate goal should be reliance on nothing more than a Linux/Unix Terminal, some manner of network access and a programming language. One of my favorite exploitation tools is my Nexus 7 2013 flo tablet (running a modified version of Nethunter) and a Bluetooth folio keyboard (I got the idea from n-o-d-e, https://www.youtube.com/watch?v=hqG8ivP0RkQ) as the final product is a netbook that fits in a jacket pocket).

I have exploited some seriously huge clients with thislittle rig (for ingress and a quick root shell, WPS on network/enterpriseprinters and knowledge PCL/PJL/Postscript are often your friend).

I have also exploited other customers with a cheap UMX smartphone with 5 gigs of storage, 1 gb of memory and GNUroot Debian (Guest Wifi access from the parking lot or an onsite public restroom, human nature, and Responder.py analyze mode, followed by WPAD, LLMNR and NetBios poisoning with NTLMv1 and LM authorization downgradefor the win).

During (red team, onsite, etc.) engagements, even when the ultimate target of the engagement is located on a hardwired network with heavy segmentation/compartmentalization (such as the conduit/zone based layouts that are general best practice in Industrial sectors), it is always worthgaining a host/node with corporate WIFI access.

One thing WIFI access provides is reach: an Administrator's (or other privileged user's) dedicated workstation may be out of reach, but his other devices (if in scope) may be connected to Corp. WIFI for reasons such as saving data on a plan.

Also, WIFI allows me attacks of opportunity even when I am doing other things. Running Responder.py on a misconfigured network's WIFI while I am elsewise engaged is gaining me advantages (maybe clear text creds, maybe hashes, maybe NTLMv1 and LM hashes) at little cost to my time or attention.

When I employ this, I like to spoof the poisoning machines hostname/mac address to something familiar on the network. If you see a bunch of hosts named "Apple" during your recon, and all of those hosts are not online, spoof the hostname/MAC to match one of the Apple machines (this will not withstand close scrutiny, but will often suffice with a little work).

It always helps to watch and take note on the norms of the network traffic and protocols. Try to match this as much as possible (this will likely help you avoid IDS/IPS, firewall rules, etc.) and whatever traffic would seriously stand out, try to tunnel or encapsulate with normal network traffic/protocols.

This leads to two other points:

A) Be prepared for the majority of people within a company who do not care about, or will minimize security issues. Do not get frustrated; I find that showing the parties involved what they stand to lose as a company from a vuln to be more effective than focusing on the vuln itself.

B) This is where the Nexus and cheap smartphone come into play: taking the client's domain with a laptop may scare up some results, but showing s customer that an attacker could cost them tens of millions with a $20 dollar smartphone or a $100 dollar tablet (from the parking lot) works wonders.

C) I have an interest in learning to exploit everything and anything. This has served me well during network penetration tests, as many targets will defend their DCs, file servers and hosts, but not pay much attention to the printers and IoT devices within the network.

D) To this end, learn to work with uncommon protocols. UPnP. NTLDNA and SSDP have been serving me well for the last couple years. Many file servers (and company smartphones/tablets when they are in scope) keep the UPnP door (and associated protocols) wide open. I once grabbed SNMP and other default network appliance creds from a fileserver through UPnP.

If you are going to pay for certs with your own cash, I recommend the OSCP. Yes, some of the machines/exploits are outdated. You won't find many of the SMB remote exploits used for the course in the wild very often anymore (unless an Admin leaves a test server up, which happens occasionally).

However, the overall experience, breakdown on enumeration methodology, self reliance and mindset the entire experience teaches you are invaluable.

I have seen some sites peddling garbage certs with no industry recognition. Save your money for the OSCP; its profile in the industry is high and growing. Certs are no replacement for experience, but starting out with a IT/CS related degree or some general IT experience (even Helpdesk work) along with the OSCP will get you hired somewhere.

For persistence, I prefer adding innocuous user accounts/Remote Desktop accounts.

If I am going to add some manner of privileged user account early to mid engagement, I usually try to add a more low profile account (if I have the option) such as Server Operator; these type of accounts allow privileged access you can build from, but generally are not watched with the scrutiny of an Administrator account.

When I do create Administrator accounts (I try to wait until I begin my endgame), I will try to match the naming convention to similar accounts in within the network. if a

For example, if the Administrator accounts within the network are named USsupervisor, I will name the added account something like USupervisor. If I know the clear text password of the account I have mimicked, I will use the same password.

Keep good notes during the engagement; too much information is better than to little information. Captured PCAPS of network traffic are great for examination during down time between engagements.

If you are a hacker, freedom fighter, or someone generally concerned about max privacy, this series of articles and configurations are for you:
https://www.ivpn.net/blog/privacy-guides/advanced-privacy-and-anonymity-part-1

My favorite distro is Backbox; it starts out with a solid set of tools ninus the obscure bloat (and so far I have been able to add anything Kali has to Backbox). You can use Backbox's "Anonymous" option for a full transparent Tor proxy, Macchanger and host name changer and set RAM to overwrite on exit.

I also keep Portable Virtualbox on a USB drive with a Kali Linux image…

You could follow some of the advice here: http://www.torforum.org/viewtopic.php?f=2&t=18320

And here: http://www.torforum.org/viewtopic.php?f=2&t=18320

The articles above could help you create an encrypted USB with a Whonix gateway and Kali Linux workstation (you could probably exchange Kali OS in the Whonix Workstation for any Debian/Debian like OS).

This configuration is disposable and concealable, and will run all of the Kali Workstation's (or other Debian/Debian like OS) through Tor. You could also create multiple other Vanilla Whonix Workstations/Gateways on the USB to create a type of local jumpbox sequencea to tunnel between/through SSH and/or VPN them before final Kali workstation.

(Note: This is just a gut feeling, but for your own OpSec/security/anonymity, you are probably best replacing the Kali workstation with another Debian/Debian like distro. I have tried Katoolin in the Whonix Workstation, but I find that Katoolin often breaks i).

A VPS with your pentest tools installed is a valuable commodity; I call mine DeathStar, and I can call down some thunder from my Nexus 7 2013 flo (and a prepaid Wireless hotspot) from pretty much anywhere.

There are some providers who do not give a damn about the traffic leaving your VM as long as you are using a VPN and a DMCA does not come their way.

For hackers and freedom fighters, get your VPS from a country outside 14 Eyes countries (providers in Eastern European/former Soviet Block countries can be both dirt cheap and extremely honorable; just do your research and have tolerance for the occasional technical issue).

You could pay with laundered/tumbled Bitcoin; even better are those providers who except gift cards (much like some VPN providers do)as payment.

Have another party buy the gift cards a good distance away from you; you can find some of these providers who take gift cards on Low End Box. The VPS can be a valuable addition to the encrypted USB above (as you now have a host/node to catch your reverse shells without sacrificing Tor) when combined with SSH or IPsec (such as Strongswan, which is in the Debian repos).

Again, this post was long because I am busy, and Iwanted to make the contribution I felt I owed this site since shortly after it began. If you have technical questions concerning (or any questions in general), please post them as comments and I will definitely get you back an answer.