60 Cybersecurity Interview Questions [2019 Update]

Written By Daniel Miessler

Created/Updated: September 5, 2020

What follows is a list of techniques for vetting candidates in Information Security (InfoSec / Cybersecurity). The list and approach has evolved over the years, as I think it should, and I think it represents a good balance between technical content and the philosophy around desired answers.

How to Build a Successful Cybersecurity Career

Interview First Principles

Here are my First Principles of interviewing in general:

1. Humans are bad at interviewing because we are full of biases. You're bad at it. I'm bad at it. Everyone's bad at it. And the more you know this—and work to guard against it—the better (or at least less bad) you'll be.
2. The goal of interviewing should be to extract the best from the candidate, not to trick them, make them uncomfortable, or otherwise keep them from shining.
3. Maintain a mentality of *matching*, not of filtering. Imagine that both you and the candidate are both amazing, and the only thing you're doing is seeing if you are good fit for each other.

The worst manifestations violating these principles are things like:

- Puzzle questions that don't apply to the real world
- Pet questions that filter for people like the interviewer
- Pressure environments that melt people into 10% of what they are normally

Sadly this knowledge is not yet understood by most interviewers and HR departments, which are still doing these things like they're canon.

Some questions are borderline "gotcha", or core knowledge that can be googled, and these tend to function more like proxies for experience. Be willing to constantly evaluate your questions (including these below) to make sure they are not based on pet, gotcha, puzzle, or pressure.

So, onto the questions.

System Administration

How do you change your DNS settings in Linux/Windows?

Here you're looking for a quick comeback for any position that will involve system administration (see system security). If they don't know how to change their DNS server in the two most popular operating systems in the world, then you're likely working with someone very junior or otherwise highly abstracted from the real world.

What are your first three steps when securing a Linux server?

Their list isn't key here (unless it's bad); the key is to not get panic.

Encryption

Does TLS use symmetric or asymmetric encryption?

Both. Have them talk through how each are used. The key (sorry) is that they understand the initial exchange is done using asymmetric and that bulk data encryption requires speed and therefore symmetric algorithms.

What's the difference between symmetric and public-key cryptography?

Standard stuff here: make sure they know that symmetric uses a single key while public-key uses two.

In public-key cryptography you have a public and a private key, and you often perform both encryption and signing functions. Which key is used for which function?

You encrypt with the other person's public key, and you sign with your own private. If they confuse the two, don't put them in charge of your PKI project.

Describe the process of a TLS session being set up when someone visits a secure website.

Look for the standard responses, with the client sending `helo` with ciphers, server responding with a public key and picking a cipher, agreement on a shared key, etc. But then dive deeper into the questions below.

If someone steals the server's private key can they decrypt all previous content sent to that server?

We're looking for conversations about Perfect Secrecy here, threats to encrypted data, etc.

What are some common ways that TLS is attacked, and/or what are some ways it's been attacked in the past?

Look for a conversation about weak ciphers, vulnerabilities like Heartbleed, BEAST, etc. It's not necessarily crucial that they remember every themed vulnerability and the exact specifics, but they should know what the issue was, why it was a problem, and what the fix was.

Cryptographically speaking, what is the main method of building a shared secret over a public medium?

Diffie-Hellman. And if they get that right you can follow-up with the next one.

What’s the difference between Diffie-Hellman and RSA?

Diffie-Hellman is a key-exchange protocol, and RSA is an encryption/signing protocol. If they get that far, make sure they can elaborate on the actual difference, which is that one requires you to have key material beforehand (RSA), while the other does not (DH). Blank stares are undesirable.

What kind of attack is a standard Diffie-Hellman exchange vulnerable to?

Man-in-the-middle, as neither side is authenticated.

What is Forward Secrecy?

Forward Secrecy is a system that uses ephemeral session keys to do the actual encryption of TLS data so that even if the server’s private key were to be compromised, an attacker could not use it to decrypt captured data that had been sent to that server in the past.

What’s the difference between encoding, encryption, and hashing?

Encoding is designed to protect the integrity of data as it crosses networks and systems, i.e. to keep its original message upon arriving, and it isn’t primarily a security function. It is easily reversible because the system for encoding is almost necessarily and by definition in wide use. Encryption is designed purely for confidentiality and is reversible only if you have the appropriate key/keys. With hashing the operation is one-way (non-reversible), and the output is of a fixed length that is usually much smaller than the input.

What is an IV used for in encryption?

An IV is used to initiate encryption by providing an addition (third) input in addition to the cleartext and the key. In general you want IVs that are random and unpredictable, which are used only once for each message. The goal is to ensure that two messages encrypted with the same key do not result in the same ciphertext.

What are block and stream ciphers?
What are the differences, and when would you use one vs. the other?

Block-based encryption algorithms work on a block of cleartext at a time, and are best used for situations where you know how large the message will be, e.g., for a file. Stream ciphers work on single units of cleartext, such as a bit or a byte, and they’re best used when you’re not sure how long the message will be.

What are some examples of symmetric encryption algorithms?

DES, RCx, Blowfish, Rijndael (AES)

What are some examples of asymmetric encryption algorithms?

Diffie Hellman, RSA, EC, El Gamal, DSAC

What are some common block cipher modes?

ECB and CBC.

What's the main difference in security between ECB and CBC?

ECB just does a one-to-one lookup for encryption, without using an IV, which makes it fairly easy to attack using a chosen-plaintext attack. CBC uses an IV for the first block and then propagates the XOR of the previous block onto subsequent ones. The difference in results can be remarkable.

What's more secure, SSL, TLS, or HTTPS?

Trick question here. And the goal is not to be cute. If you can't wait to smash someone with this then you should not be interviewing people. It's to identify people who've not been in the industry for any measure of time.

Look for a smile like they caught you in the cookie jar. If they're confused, then this role should be for an extremely junior position.

Network Security

What port does ping work over?

A trick question, to be sure, but an important one. If they start throwing out port numbers you may want to immediately move to the next candidate. Hint: ICMP is a layer 3 protocol (it doesn't work over a port) A good variation of this question is to ask whether ping uses TCP or UDP. An answer of either is a fail, as those are layer 4 protocols.

Do you prefer filtered ports or closed ports on your firewall?

Look for a discussion of security by obscurity and the pros and cons of being visible vs. not. Basically anything intelligent in terms of discussion. There can be many signs of maturity or immaturity in this answer.

How exactly does traceroute/tracert work at the protocol level?

This is a fairly technical question but it’s an important concept to understand. It’s not natively a “security” question really, but it shows you whether or not they like to understand how things work, which is crucial for an Infosec professional. If they get it right you can lighten up and offer extra credit for the difference between Linux and Windows versions.

The key point people usually miss is that each packet that’s sent out doesn’t go to a different place. Many people think that it first sends a packet to the first hop, gets a time. Then it sends a packet to the second hop, gets a time, and keeps going until it gets done. That’s incorrect. It actually keeps sending packets to the final destination; the only change is the TTL that’s used. The extra credit is the fact that Windows uses ICMP by default while Linux uses UDP.

What are your favorite security assessment tools?

You want to hear things like Metasploit, Responder, Empire, and maybe Kali, Nessus, etc. What you don’t want to hear is Kali by itself, with no specifics. As with most of these questions, the goal is to get them talking so you can expose their knowledge, passion, or lack thereof.

How does a buffer overflow work?

Hopefully they can describe the high-level, but ask them to go into as much detail as they can.

How can one defend against buffer overflows?

Look for answers around modern languages and frameworks, and built-in OS protections that exist in various operating systems.

What are Linux’s strengths and weaknesses vs. Windows?

Look for biases. Does he absolutely hate Windows and refuse to work with it? This is a sign of an immature hobbyist who will cause you problems in the future. Is he a Windows fanboy who hates Linux with a passion? If so just thank him for his time and show him out. Linux is everywhere in the security world.

Application Security

Describe the last program or script that you wrote.
What problem did it solve?

All we want to see here is if the color drains from the person’s face. If they panic then we not only know they’re not a programmer (not necessarily bad), but that he’s afraid of programming (bad). I know it’s controversial, but I think that any high-level security person needs at least some programming skills. They don’t need to be a God at it, but they need to understand the concepts and at least be able to muddle through some scripting when required.

How would you implement a secure login field on a high traffic website where performance is a consideration?

The answer you're looking for here is that TLS is a must for the entire site at this point, and that there are very few situations where you shouldn't insist on encryption.

Other good responses include those around using solid, dependable frameworks, and not building your own.

What are the various ways to handle account brute forcing?

Look for discussion of account lockouts, IP restrictions, fail2ban, commercial versions thereof, etc.

What is Cross-Site Request Forgery?

Not knowing this is more forgivable than not knowing what XSS is, but only for junior positions. Desired answer: when an attacker gets a victim's browser to make requests, ideally with their credentials included, without their knowing. A solid example of this is when an IMG tag points to a URL associated with an action, e.g. http://foo.com/logout/. A victim just loading that page could potentially get logged out from foo.com, and their browser would have made the action, not them (since browsers load all IMG tags automatically).

How does one defend against CSRF?

Nonces required by the server for each page or each request is an accepted, albeit not foolproof, method. Again, we're looking for recognition and basic understanding here—not a full, expert level dissertation on the subject. Adjust expectations according to the position you're hiring for.

If you were a site administrator looking for incoming CSRF attacks, what would you look for?

This is a fun one, as it requires them to set some ground rules. Desired answers are things like, "Did we already implement nonces?", or, "That depends on whether we already have controls in place…" Undesired answers are things like checking referrer headers, or wild panic.

What's the difference between HTTP and HTML?

Obviously the answer is that one is the networking/application protocol and the other is the markup language, but again, the main thing you're looking for is for them not to panic. The object here should be identifying absolute beginners and/or having fun with people who know how silly the question is.

How does HTTP handle state?

It doesn't, of course. Not natively. Good answers are things like "cookies", but the best answer is that cookies are a hack to make up for the fact that HTTP doesn't do it itself.

What exactly is Cross Site Scripting, and how would you explain it to a 10-year-old??

You'd be amazed at how many security people don't know even the basics of this immensely important topic. We're looking for them to say anything regarding an attacker getting a victim to run script content (usually JavaScript) within their browser.

What'ss the difference between stored and reflected XSS?

Stored is on a static page or pulled from a database and displayed to the user directly. Reflected comes from the user in the form of a request (usually constructed by an attacker), and then gets run in the victim's browser when the results are returned from the site.

What are the common defenses against XSS?

Input Validation and Output Sanitization, with focus on the latter.

Risk

What is the primary reason most companies haven't fixed their vulnerabilities?

I look for people to realize that companies don't actually care as much about security as they claim to—otherwise we'd have a very good remediation percentage. Instead, we have a ton of unfixed things and more tests being performed. A variation of this is something like:

It's not hurting enough yet.

Look for people who get this, and are ok with the challenge.

What's the goal of information security within an organization?

This is a big one. What I look for is one of two approaches; the first is the über-lockdown approach, i.e. "To control access to information as much as possible, sir!" While admirable, this again shows a bit of immaturity. Not really in a bad way, just not quite what I'm looking for. A much better answer in my view is something along the lines of, "To help the organization succeed."

This type of response shows that the individual understands that business is there to make money, and that we are there to help them do that. It is this sort of perspective that I think represents the highest level of security understanding—-a realization that security is there for the company and not the other way around.

What’s the difference between a threat, vulnerability, and a risk?

As weak as the CISSP is as a security certification it does teach some good concepts. Knowing basics like risk, vulnerability, threat, exposure, etc. (and being able to differentiate them) is important for a security professional. Ask as many of these as you’d like, but keep in mind that there are a few different schools on this. Just look for solid answers that are self-consistent.

If you were to start a job as head engineer or CSO at a Fortune 500 company due to the previous guy being fired for incompetence, what would your priorities be? Imagine you start on day one with no knowledge of the environment.

We don’t need a list here; we’re looking for the basics. Where is the important data? Who interacts with it? Network diagrams. Visibility touch points. Ingress and egress filtering. Previous vulnerability assessments. What’s being logged an audited? Etc. The key is to see that they could quickly prioritize, in just a few seconds, what would be the most important things to learn in an unknown situation.

As a corporate Information Security professional, what’s more important to focus on: threats or vulnerabilities?

This one is opinion-based, and we all have opinions. Focus on the quality of the argument put forth rather than whether or not they chose the same as you, necessarily. My answer to this is that vulnerabilities should usually be the main focus since we in the corporate world usually have little control over the threats.

Another way to take that, however, is to say that the threats (in terms of vectors) will always remain the same, and that the vulnerabilities we are fixing are only the known ones. Therefore we should be applying defense-in-depth based on threat modeling in addition to just keeping ourselves up to date.

Both are true, of course; the key is to hear what they have to say on the matter.

Industry Wisdom

Are open-source projects more or less secure than proprietary ones?

The answer to this question is often very telling about a given candidate. It shows 1) whether or not they know what they're talking about in terms of development, and 2) it really illustrates the maturity of the individual (a common theme among my questions). My main goal here is to get them to show me pros and cons for each. If I just get the "many eyes" regurgitation then I'll know he's read Slashdot and not much else. And if I just get the "people in China can put anything in the kernel" routine then I'll know he's not so good at looking at the complete picture.

The ideal answer involves the size of the project, how many developers are working on it (and what their backgrounds are), and most importantly — quality control. In short, there's no way to tell the quality of a project simply by knowing that it's either open-source or proprietary. There are many examples of horribly insecure applications that came from both camps.

Can you describe rainbow tables?

Look for a thorough answer regarding overall password attacks and how rainbow tables make them faster.

What is salting, and why is it used?

You purposely want to give the question without context. If they know what salting is just by name, they've either studied well or have actually been exposed to this stuff for a while.

Who do you look up to within the field of Information Security? Why?

A standard question type. All we're looking for here is to see if they pay attention to the industry leaders, and to possibly glean some more insight into how they approach security. If they name a bunch of hackers/criminals that'll tell you one thing, and if they name a few of the pioneers that'll say another. If they don't know anyone in Security, we'll consider closely what position you're hiring them for. Hopefully it isn't a senior position.

Where do you get your security news from?

Here I'm looking to see how in tune they are with the security community. Answers I'm looking for include things like Team Cymru, Reddit, Twitter, etc. The exact sources don't really matter. What does matter is that they don't respond with, "I go to the CNET website.", or, "I wait until someone tells me about events.". It's these types of answers that will tell you they're likely not on top of things.

If you had to both encrypt and compress data during transmission, which would you do first, and why?

If they don't know the answer immediately it's ok. The key is how they react. Do they panic, or do they enjoy the challenge and think through it? I was asked this question during an interview at Cisco. I told the interviewer that I didn't know the answer but that I needed just a few seconds to figure it out. I thought out loud and within 10 seconds gave him my answer: "Compress then encrypt. If you encrypt first you'll have nothing but random data to work with, which will destroy any potential benefit from compression.

What kind of systems do you have at home or in the cloud to tinker with?

Good answers here are anything that shows you the person is a computer/technology/security enthusiast and not just someone looking for a paycheck. So if she's got multiple systems running multiple operating systems you're probably in good shape. What you don't want to hear is, "I get enough computers when I'm at work…" I've yet to meet a serious security guy who doesn't have a considerable home network—or at least access to one, even if it's not at home.

This used to say home lab, but now that extends to the cloud as well.

What are the advantages offered by bug bounty programs over normal testing practices?

You should hear coverage of many testers vs. one, incentivization, focus on rare bugs, etc.

How would you measure how well a security team is doing?

Here we're looking for them to ask us questions in return, such as, "What kind of team?" Answers that are bad include anything purely number-based like number of IDS events, or widget-thingies detected. We want to know how much experience they have tracking the things that matter vs. the things that don't.

What are your first three steps when securing a Windows server?

Their list isn't key here (unless it's bad); the key is to not get panic.

Who's more dangerous to an organization, insiders or outsiders?

Ideally you'll hear inquiry into what's meant by "dangerous". Does that mean more likely to attack you, or more dangerous when they do?

Why is DNS monitoring important?

If they’re familiar with infosec shops of any size, they’ll know that DNS requests are a treasure when it comes to malware indicators.

The Onion Model

The questions above are fairly straightforward. They are, generally, negative filters, i.e. they’re designed to excluded candidates for having glaring weaknesses. If you are dealing with a more advanced candidate then one approach I recommend taking is that of the onion model.

The Onion Model of interviewing starts at the surface level and then dives deeper and deeper—often to a point that the candidate cannot go. This is terrifically revealing, as it shows not only where a candidate’s knowledge stops, but also how they deal with not knowing something.

One component of this cannot be overstated: Using this method allows you to dive into the onion in different ways, so even candidates who have read this list, for example, will not have perfect answers even if you ask the same question.

An example of this would be starting with:

How does traceroute work?

They get this right, so you go to the next level.

- What protocol does it use?

This is a trick question, as it can use lots of options, depending on the tool. Then you move on.

Describe a Unix traceroute hitting google.com at all seven layers of the OSI model.

Etc. It’s deeper and deeper exploration of a single question. Here’s a similar option for the end-phase of such a question.

- If I’m on my laptop, here inside my company, and I have just plugged in my network cable. How many packets must leave my NIC in order to complete a traceroute to twitter.com?

The key here is that they need to factor in all layers: Ethernet, IP, DNS, ICMP/UDP, etc. And they need to consider round-trip times. What you’re looking for is a realization that this is the way to approach it, and an attempt to knock it out. A bad answer is the look of WTF on the face of the interviewee.

This could be asked as a final phase of a multi-step protocol question that perhaps starts with the famous, “What happens when I go to Google.com?”

How would you build the ultimate botnet?

Answers here can vary widely; you want to see them cover the basics: encryption, DNS rotation, the use of common protocols, obscuring the heartbeat, the mechanism for providing updates, etc. Again, poor answers are things like, “I don’t make them; I stop them.”

Role-Playing as an Alternative to the Onion Model

Another option for going to increasing depth, is to role-play with the candidate. You present them a problem, and they have to troubleshoot. I had one of these during an interview and it was quite valuable.

You would tell them, for example, that they’ve been called in to help a client who’s received a call from their ISP stating that one or more computers on their network have been compromised. And it’s their job to fix it. They are now at the client site and are free to talk to you as the client (interviewing them), or to ask you as the controller of the environment, e.g. “I sniff the external connection using tcpdump on port 80. Do I see any connections to IP 8.8.8.8.” And you can then say yes or no, etc.

From there they continue to troubleshooting/investigating until they solve the problem or you discontinue the exercise due to frustration or pity.

Innovation Questions

At the top tier of technical security roles you may want someone who is capable of designing as well as understanding. In these cases you can also ask questions about design flaws, how they would improve a given protocol, etc.

These questions separate good technical people from top technical people, and I imagine less than 1% of those in infosec would even attempt to answer any of these.

Here are a few examples:

- What are the primary design flaws in HTTP, and how would you improve it?
- If you could re-design TCP, what would you fix?
- What is the one feature you would add to DNS to improve it the most?
- What is likely to be the primary protocol used for the Internet of Things in 10 years?
- If you had to get rid of a layer of the OSI model, which would it be?

You can ask infinite variations of these, of course. Asking for three options instead of one, or asking them to rank the results, etc.

It’s important to note with these questions that you could have a superstar analyst who knows nothing about these matters while someone who is at this level would make a poor forensic expert. It’s all about matching skills to roles.

My Top Questions

So with all that being said, here are my current favorite questions to ask if I have limited time.

1. Tell me about a project you worked on in the past that you really enjoyed.
2. What was challenging about it?
3. Why did you choose to approach it the way you did vs. (list alternatives)
4. If you could have the perfect job and the perfect manager, what would that look like? What you do day to day, and what kind of projects would you have?
5. What is a skill you wish you had but don’t yet have?
6. How are you working to get that skill?
7. What do you think the most important technology is right now?
8. How are we going to secure it?

Conclusion

For more on hiring overall, I recommend doing a good amount of research. Most important to learn, as I talked about above, is the limitations of interviews. Use other data available to you whenever possible, and above everything else: Be extremely cautious of anyone who thinks they can spot “the one” because they’re good at it..

Bias is a major problem in interviewing, and it’s likely that someone with a steadfast belief in his or her interview brilliance is doing harm to your organization by introducing bad candidates. When possible, do what Google did: *explore the data*. Look at how candidates did in interviews relative to how they did on the job. Wherever you have mismatches you have a problem with your process.

Feel free to contact me if you have any comments on the questions, or if you have an ideas for additions.

Notes

1. Here is an article about Google revealing the ineffectiveness of their brainteaser questions.
2. As a hiring organization, be cautious of any interviewer that has an ego or attitude. The odds of you getting any good data from them is low. The name of the game is reducing bias, and that type has a lot of it.
3. Also beware that any interviewee who is extremely nervous is not performing their best. As an interviewer, your job should be to get them relaxed enough to perform the way they will at work, and to reduce any tension that’s preventing that from happening.
4. Always try to combine any interview with a work sample, and/or great reference data.

5. I have had these questions asked to me on numerous interviews. It’s quite humorous when they find out they’re reading from my website.
6. The June 2017 update was a rewrite based on an evolving view of technical interviews. Check out the Philosophy section above to learn about that evolution.
7. A key question you should be asking yourself with these types of questions is whether it’s something they should know off the top of their head, or if it’s something they should be able to research quickly and find out. If it’s the latter, then why are we asking them to recite it from memory? That’s the old style of interviewing, and it is not effective in predicting real-world success.