



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.

#### 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’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’s more secure, SSL or HTTPS?

Trick question: these are not mutually exclusive. Look for a smile like they caught you in the cookie jar. If they’re confused, then this should be for an extremely junior position.

#### 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 junior 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 he doesn’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 he’s 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's the difference between symmetric and public-key cryptography

Standard stuff here: single key vs. two keys, etc, etc.

### 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.

### What kind of network do you have at home?

Good answers here are anything that shows you he's a computer/technology/security enthusiast and not just someone looking for a paycheck. So if he'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.

### 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.

### 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.

### 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.

## Network Security

### What port does ping work over?

A trick question, to be sure, but an important one. If he starts 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. 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 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.

### 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.

## 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 guy's face. If he panics then we not only know he's 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 guy needs 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?

We're looking for a basic understanding of the issue of wanting to serve the front page in HTTP, while needing to present the login form via HTTPS, and how they'd recommend doing that. A key piece of the answer should center around avoidance of the MiTM threat posed by pure HTTP. Blank stares here mean that they've never seen or heard of this problem, which means they're not likely to be anything near pro level.

### What are the various ways to handle account brute forcing?

Look for discussion of account lockouts, IP restrictions, fail2ban, 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 him not to panic.

### 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?

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's 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/Output Sanitization, with focus on the latter.

## Corporate/Risk

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

This is a bit of a pet question for me, and 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. 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 differing 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.

[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 fact 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?

[ NOTE: 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.

## 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.

[Daniel Miessler](#)