10.2.4 Client-Side and Network Attack Facts

This lesson covers the following topics:

- Application level hijacking
- Network level hijacking

Application Level Hijacking

At the application level, the session ID lets the server knows who they are communicating with. This permits the user to progress to a different page on a website without having to log in again. You can imagine it would be hard for a company to sell much of anything if a user had to log in every time they wanted to look at another product or another page. Session IDs can be found in various places. By reviewing a user's browsing history, you may be able to enter a previously used URL to gain access to an open session. If a user recently completed a form, you may be able to find a session ID in a hidden field in the HTTP POST command. The most notorious location of session IDs is in the HTTP cookies.

Several different methods are available for finding session IDs:

Method	Description
Session sniffing	Session sniffing is basically just an extension of sniffing efforts that we've discussed in the past, except now, we're specifically on the lookout for session IDs.
Predicting session tokens	The easiest way to predict session tokens is to collect several session IDs that have been used before and then analyze them to determine a pattern. Once you know the pattern or algorithm being used, you may be able to predict a future ID.
Man-in-the-middle attack	We'll talk about this more in the network hijacking presentation, but it's worth noting that man-in-the-middle is a viable method for obtaining a session ID.
Cross-site-scripting	Cross-site scripting attacks (XSS) involves the injection of malicious Java, Flash, or HTML script into web applications. This is usually done through user entered content that has not gone through any validation checks. A stored XSS attack is dangerous because it targets web applications that allow users to store data on the site for retrieval by other users.
Session fixation	Session fixation attacks target websites where session IDs are provided in the hyperlink. URLs are sent to a user with session IDs already embedded into them. When a user logs in using this URL, their user information becomes aligned with that session ID. An attacker following the same URL would have the same level of access as the targeted user.

Network Level Hijacking

There are also several methods for hijacking session IDs at the network level:

Method	Description
TCP/IP session hijacking	As the name suggests, TCP/IP session hijacking is an attack on a TCP session. The first phase in a TCP/IP hijack is to have a successful sniffing tool in place to capture traffic between two machines. Second, you'll want to monitor the existing traffic so you can predict the packet sequence numbers. Third, you'll want to carry out a denial-of-service attack on the target machine or manipulate their connection in some way that you're able to effectively take over the client role. Lastly, you'll begin injecting packets into the server as if you were the authenticated client.
UDP session hijacking	Unlike TCP, UDP is a connectionless protocol. In other words, there is not a verified connection between the server or host machine and the client. Because of this, you don't need to predict a packet sequence. Instead, you just need to convince the victim that you're the server. The best way to do this is to get a response back to the client before the actual server responds and take over the server's role. Given the high level of vulnerability and the low number of error recovery options of UDP, it's primarily used for DNS queries and network broadcast messages.
DNS spoofing	DNS spoofing, also known as DNS cache poisoning, targets Active Directory or other DNS-reliant networks. In DNS spoofing, an attacker alters the DNS server to redirect traffic to a malicious website that can gather sensitive information about a user or that can install malware onto the target machine.
Man-in-the-middle attacks	A man-in-the-middle attack is probably one of the most well-known attacks. This attack starts with the attacker sniffing traffic between the target machine and the server or the host machine. They will then use ARP poisoning to strategically redirect communication through their machine. At this point, the attacker can forward manipulated and potentially malicious traffic to either the victim or the host machine.