**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **05/August/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **Coursera** | **USN:** | **4al17ec074** |
| **Topic:** | **Network Security &**  **Database Vulnerabilities** | **Semester & Section:** | **6th b** |
| **GitHub Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **To convert numbers between binary, octal, decimal, and hexadecimal number systems, base 2, 8, 10, and 16 respectively This lesson is being presented by Ben Briggs, and is based upon a lecture series developed by Moises Monge. Let's talk about networking fundamentals, specifically IP addressing. This lesson will cover the basics of IPv4 addressing and touch on IPv6. I'm sure you've all been exposed to base 2 or binary numbers in school, but I'll give you a quick overview since that was a long time ago for some of us. We're used to thinking, talking, and calculating in base 10, or decimal number system.**  **Humans adopted the base 10 number system most likely because we have 10 fingers and 10 toes, so it's easy to speculate that early counting was done on our ancestors fingers. Computers, however, only know two different states, on and off or high-voltage, low-voltage, or open gate, closed gate, which we represent logically as one or zero. Computers can only differentiate between these two states, so this is how the decimal system works. We have digits for ones, tens and hundreds, thousands, ten thousands, etc.**  **Each digit being worth 10 times as much as the preceding digit. Each digit can hold 10 different values, ranging from 0-9. Remember that the word decimal means 10. In the binary system, on the other hand, each digit or placeholder holds a different value. There are placeholders for 1s, 2s, 4s, 8s, 16s, etc. Each digit being worth two times as much as the preceding digit. In binary, each digit can hold only two different values, zero or one.**  **Remember that binary means two. So converting from binary to decimal is easy, just add up all the values of the digits that contain a one and ignore those that contain zeros. In this example, we have an eight digit binary number, so there are eight placeholders. The decimal equivalent of this number would be 2 plus 8 plus 16 plus 64 plus 128, or 218. Converting from decimal to binary is a little more awkward, that might remind you of the long division you learned in grade school. Of course, there are plenty of calculators and online tools that will make these conversions for you, but here's how it's done.**  **In this example, we have to convert 235 from decimal to binary. Start by saying, what is the largest binary place holder you can subtract from your decimal number? Two hundred fifty-six is too large, but you can subtract the next smaller placeholder, 128. So put a 1 in the 128th place, and then subtract 128 from 235 and you're left with 107. Check to see that you can subtract 64 from 107. Yes, you can. So add 1 to the 64 place, and then subtract 64 from 127, and you're left with 43. Checking the next place holder down, can we subtract 32 from 43? Yes, we can. We add 1 to the 32 placeholder and subtract it from 43, leaving us with 11.**  **Check the next place holder down, can we subtract 16 from 11? No, we cannot, so we put a 0 in the 16 place and check the next place holder down. Can we subtract 8 from 11? Yes, we can, so put a 1 in the 8 place holder and subtract 8 from 11 to leave you with 3. As before, check the next place holder down. Can we subtract 4 from 3? No, so we put a 0 in the 4 placeholder and check the next place holder down. We can subtract 2 from 3, so we add a 1 to the 2 placeholder and do the subtraction. Now we have 1 left and the next placeholder down, it's the last 1, 1. We can subtract 1 from 1, so we put a 1 in the placeholder and do the subtraction, 1 minus 1 is 0, so we are done. This will be the representation of 235 in binary, 11101011. So that was a good introduction to converting between binary and decimal.**  **But unfortunately, binary and decimal are not the only number systems you will run into when working with computers. If you're already comfortable with binary, octal, decimal, and hexadecimal systems, please feel free to skip the last five minutes of this video. In binary base 2, each digit can have two possible values, 0 or 1. We'll skip over base 4 since it's unlikely you'll run into that, but the same principles apply anyway. Octal is base 8, it's less common than binary, but you'll see it from time to time. Octal can take on eight possible values ranging from 0-7. Decimal is base 10, and it can take on 10 possible values ranging from 0-9. Hexadecimal, called hex for short, is base 16, and it can take on 16 possible values ranging from zero to f. After we run out of numbers, we start using letters.**  **So 10 through 15 are represented by a, b, c, d, e, and f respectively. For any number system, not just these four, the way to calculate how many different numbers can be represented by any given number of digits is simply to raise the base you're working with to the power of the number of digits available. Let's say we have four digits available. In binary, we raised two, because it's base 2\_4. We get 16 different numbers that can be represented using these four digits. Remembering that zero counts as a possible number, that would give us a range of 0-1111 or 0-15.**  **In octal, raise the base 8\_4 and get 4,096 possible numbers. That would give us a range of 0-7777, that's 0-4,095 in decimal. In decimal, 10 raised to the fourth power is 10,000, giving us a range of 0-9999. Finally, in hex, our largest number would be 16 raised to the fourth power or 65,536 in decimal. That would give us a range of 0-ffff or 0-65,535 in decimal. It's easy to see that using a larger base makes for much more compact way of representing large numbers.**  **For example, three digits in hex fff, need 12 digits in binary, giving us 111111111111. Computer hardware and software at the machine code level is based on binary. So staying with a number system that's a power of two, makes it much easier to convert between these systems. Three binary digits can translate directly into one octal digit, and four binary digits can be directly represented by a single hex digit. In decimal , you see there is not this exact digit to digit conversion possible.**  **To convert from any base system to decimal, you can draw a table like the one we used previously, but substitute the right column header values for whichever base you're working with. For binary, that would be 1, 2, 4, 8, etc. For octal, that would be 1, 8, 64, 512, etc. For decimal, that would be 1, 10, 100, 1,000, etc, and for hex, it would be 1, 16, 256 and 4,096, etc. Notice that all of these column headers are the base raised to the power of 0, 1, 2, 3, etc. This works for any base you need to work with. I hope this helps. You may need to refer back to this section when covering the last video in this lesson, where hex is used for IPv6 addressing.** |