**Cybersecurity Vulnerability Post-test**

Directions: Students will be given the set-up visually in front of the classroom. Students will record possible vulnerability AND possible solution on answer document. Scores will be graded based on attached rubric.

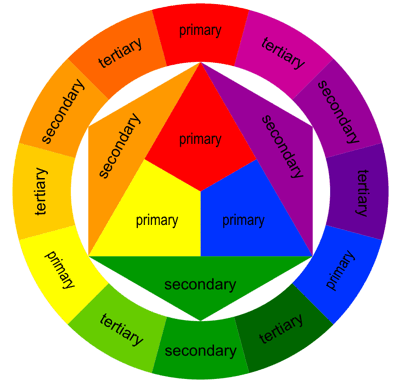
1. Reverse Function Attack
   1. Set-up:
      1. Server and Client are trying to make connection
      2. Both have a key (Blue)
      3. To establish link, Server sends Client a color (Green) that is the key mixed with another color
      4. Client decrypts Purple by reversing the mixing process (What color mixed with Blue gives me Green?)
      5. Client sends that (Yellow) back to Server
      6. Server confirms and connection is established
   2. Vulnerability:
      1. Attacker can intercept transmissions and figure out Blue by deciding which color mixed with Yellow gives Green
   3. Solution:
      1. Never send key without a hash function (something that can be worked backwards)
      2. Use timestamp
2. Time-stamp Attack
   1. Set-up:
      1. Server and Client are trying to make connection
      2. Both have a key (Blue)
      3. Client sends Server { Blue: Timestamp}
      4. Attacker can intercept the Key but doesn’t matter because they can reproduce the timestamp
   2. Vulnerability:
      1. Timestamp could be a window of times
         1. When two text messages send right after the other sometimes they say same time even though that is impossible (don’t include milliseconds)
      2. Attacker could set time back on server
      3. If multiple servers, attacker could use Red on different server avoiding the timestamp
   3. Solution:
      1. Require server to show all times
      2. Use key {Blue: Timestamp / Server}

\*\*Could also be used with hash function but server impersonation is still vulnerability

1. Man in the middle Attack
   1. Set-up:
      1. Server and Client are trying to make connection
      2. Both have a key (Blue) which is unknown to any outsider
      3. To establish link, Server sends Client Yellow
      4. Client accepts Yellow – mixes with Blue – sends back Green
      5. Server mixes Blue with Yellow as well and confirms Green
      6. Since both Client and Server get same color – connection is established
   2. Vulnerability:
      1. Attacker poses as Server and intercepts Green
      2. Attacker sends back to Client confirmation that Green was received
      3. Connection is now established between Client and Attacker and Server is left out of the loop
   3. Solution:
      1. Client and Server BOTH need to authenticate identities
      2. For example, Client needs to also send Red and make sure Server responds with Purple

\*\*Colors represent one-way function

1. Reflection Attack
   1. Set-up:
      1. Client and Server need to BOTH authenticate identity
      2. Both have a key (Blue) which is unknown to any outsider
      3. To establish link, Client sends Server Red
      4. Server mixes Blue with Red, responds Purple but also sends its own color (Yellow) to authenticate client
      5. Client confirms Purple, mixes Yellow with Blue and responds with Green
      6. Server confirms Green and connection is established
   2. Vulnerability:
      1. Attacker attempts to establish connection with Server, sends Red
      2. Server mixes Red with Blue, responds Purple but also sends its own color (Yellow) to authenticate client
      3. Attacker can’t responds because it doesn’t know what to mix with Yellow so instead pauses that connection and opens a NEW connection
      4. Attacker sends Yellow to Server, server mixes with Blue and responds with Green
      5. Attacker now knows what the correct response is to Yellow, goes back to paused connection and responds Green – Connection established
   3. Solution:
      1. Change direction of arrows – Force client to confirm identity first

**ANSWER KEY IN BLUE**