CMPS 122 Winter 2019

Homework #2

Due: 23:59 Thursday, January 31, 2019
Points: 50 (5% of final grade)
Submission: Upload pdf to Canvas

PROBLEM 1 (15PTS)

For each of the following code fragments, state one or more preconditions that together ensure memory safety. If no precondition exists, say why.

1.
```
void foo(char* a, char* b) {
   strcpy(a, b);
}
```

2.
```
void bar(char* a) {
   a[strlen(a)] = '\0';
}
```

3.
```
void baz(char* a) {
   char * b = malloc(sizeof(a));
   strcpy(b, a);
}
```

4.
```
void qux(char* a, size_t i) {
  char * b = malloc (i + 1);
  memcpy(b, a, i);
  b[i+1] = '\0';
}
```

5.
```
void quux(char* a) {
  memcpy((int *)0xffff49f0, a, strlen(a)/4);
}
```

PROBLEM 2 (15PTS)

Describe the vulnerability in the following code fragments, how (as specifically as possible) an attacker might exploit the vulnerability to execute arbitrary code, and what system or compiler mitigation technique(s) would be most effective at defending against it.

1.
```
char* foo(char * a) {
  char[128] buf;
  sprintf(buf, a);
  return buf;
}
```

2.
```
void bar(char * a) {
  for (int i = 0; i < strlen(a); i++) {
    if (a[i] == ';') {
      a[i] = '\\';
      a[i+1] = ';';
    }
  }
  system(a);
}
```

3.
```
void baz(char* a[], size_t n) {
  for (int i = 0; i < n; i++) {
      free(a[i]);
  }
}
```

PROBLEM 3 (10PTS)

Determine if the following Java code contains an exploitable SQL injection vulnerability. If it does, provide arguments to the procedure that would successfully execute an attack. If not, explain why.

```
ResultSet getMovieInfo(Connection c, String col, String title, Date date)
{
    String query = "SELECT " + col + " FROM Movies WHERE title = ? AND
       date <= ?";
    PreparedStatement p = c.prepareStatement(query);
    p.setString(1, title);
    p.setDate(2, date);
    return p.executeQuery();
}
```

PROBLEM 4 (10PTS)

A small but popular developer website, `jsdevsite.com`, has allowed their DNS registration to lapse. A devious friend of yours managed to snap up the registration and redirect it to their own server, but isn't sure what to do next. On a whim, you connect to your banking website `mybank.com` and capture the raw HTTP session below. After taking a look, you get an idea. Before telling it to your friend, you transfer all your money to a different bank.

Give detailed instructions to your friend that would allow them to steal money from the customers of `mybank.com`.

```
HTTP/1.1 200 OK
Date: Mon, 24 Jan 2019 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jan 2019 19:15:56 GMT
Content-Length: xxx
Content-Type: text/html
Set-Cookie: session=xf4330878444597bd3933d41; Domain=mybank.com ; Secure; HttpOnly
Connection: Closed


<html>
  <script src="http://jsdevsite.com/widget.js"/>
  <body>
  <form action="/transfer.php">
    <input type="text" name="sourceAcct" type="hidden" value="2325235"/><br>
    Destination account:<br>
    <input type="text" name="destAcct"/><br/>
    Amount:<br>
    <input type="text" name="amount"/><br/>
    <input type="submit" value="Submit">
  </form>
  </body>
</html>
```