Secure Programming Techniques — exam 26.05.2017

1. Imagine a simple Web Application Firewall that filters single quote, double quote and semicolon characters in query parameters. What attacks would it stop and what attacks would still work?

2. In which ways can sloppy web application authentication be broken in public passowrd-less WiFi network?

3. Find all potential vulnerabilities in this C function:

```
/* read specified number of 256-byte records from socket
   and validate them */
int populate_data(int fd)
{
    int i, nfields;
    char buf[16*256];
    unsigned char *dynbuf;
    int ret;

    read(fd, &nfields, sizeof(nfields));
    if (nfields > 16*256) {
        return 0;
    }
    dynbuf = malloc(nfields*256);
    for (i=0; i < nfields; i++) {
        if (read(fd, dynbuf+256*i, 256) < 0) {
          return ret;
        }
        memcpy(buf+i*256, dynbuf+i*256, 256);
    }
    if (validate_data(buf, nfields) ret = 1;
    return ret;
}
```

4. Find all potential vulnerabilities in this Python function:

```
def store_bypattern(myglob, secretdata):
  thefile = os.popen('echo ' + myglob).readline().strip('\n')
  if not os.path.exists(thefile):
    raise IOError('"%s" does not exist' % thefile)
  if os.access(thefile, os.X_OK):
    raise IOError('"%s" is executable, not writing' % thefile)
  if os.path.islink(thefile):
    raise IOError('"%s" is a symbolic link' % thefile)
  f = open(thefile,'w')
  f.write(secretdata)
  f.close()
```

5. What possible vulnerabilities could the following URL exploit? Try to find as many as possible:

```
http://a.com/a.php/b.html?sessid=10203040506070809&user=‘id‘&
 pass=.&name=x’%20union%20select%20’z&message=%3ci%4dg+s%52c
 %3d%22j%61v%61S%43r%69pt%3a%61l%65rt(%27%2c%27)%3b%22%3e
```

Send response by e-mail to mroos@ut.ee no later than 12.00. You can answer either in English or in Estonian.