Tutorial Session – 11

1. List some examples of injection attacks. What are the general circumstances in which injection attacks are found?
2. State the similarities and differences between command injection and SQL injection attacks with an example.
3. State a problem that can occur with input validation when the Unicode character set is used.
4. List several software security concerns associated writing safe program code.
5. Identify several concerns associated with the use of environment variables by shell scripts.
6. Identify several issues associated with the correct creation and use of a lockfile.
7. Identify several issues associated with the correct creation and use of a temporary file in a shared directory.
8. List some problems that may result from a program sending unvalidated input from one user to another user.
9. Investigate the meaning of all met characters used by the Linux/UNIX Bourne shell, which is commonly used by scripts running other commands on such systems. Compare this list to that used by other common shells such as BASH or CSH. What does this imply about input validation checks used to prevent command injection attacks?
10. You are asked to improve the security in the CGI handler script used to send comments to the Webmaster of your server. The current script in use is shown in Figure A, with the associated form shown in Figure B. Identify some security deficiencies present in this script. Detail what steps are needed to correct them, and design an improved version of this script.

```
#!/usr/bin/perl
# comment.cgi - send comment to webmaster
# specify recipient of comment email
$to = "webmaster";

use CGI;
use CGI::Carp qw(fatalsToBrowser);
$q = new CGI; #       create query object

# display HTML header
print $q->header,
$q->start_html('Comment Sent'),
$q->h1('Comment Sent');

# retrieve form field values and send comment to webmaster
$subject = $q->param("subject");
$from = $q->param("from");
$body = $q->param("body");

# generate and send comment email
system("export REPLYTO=\"$from\"; echo \"$body\" | mail -s \"$subject\"
$to");

# indicate to user that email was sent
print "Thank you for your comment on $subject.";
print "This has been sent to $to.";

# display HTML footer
print $q->end_html;
```

(a) Comment CGI script

```
<html><head><title>Send a Comment</title></head><body>
<h1> Send a Comment </h1>
<form method=post action="comment.cgi">
<b>Subject of this comment</b>: <input type=text name=subject
value="">
<b>Your Email Address</b>: <input type=text name=from value="">
<p>Please enter comments here:
<p><textarea name="body" rows=15 cols=50></textarea>
<p><input type=submit value="Send Comment">
<input type="reset" value="Clear Form">
</form></body></html>
```

(b) Web comment form

11. Investigate the functions available in PHP, or another suitable Web scripting language, to sanitize any data subsequently used in an SQL query.