Assignment 4: Individual - C code cleanup

Published

Purpose: This assignment, to be completed by each person in the class without help from team members, is to critically examine some C code for flaws that could lead to buffer overrun or even other exploits. Your job is to identify the vulnerabilites then provide fixes.

Sunny Sunheim likes writing C code. That's because when he writes it, the sun is always shining. If his code compiles, he won the battle and it's sunny. If his program runs properly with his input, he won the war and it's sunny *and* warm. Sunny smiles broadly when he wins the battle and the war, and then goes out into the sun for some serious R and R. Your job is to rain on Sunny's 'shining' C code and identify how it can be exploited (or that it can't be exploited -- maybe Sunny got lucky (the sun was shining on him)). Show Sunny that he spends too much time in the sun!

For each of the following letters, identify problems/vulnerabilities and suggest how to avoid them (first) inside the function itself or (second - assuming first is not possible) how you could modify the parameters to the function to help. Since you have the code, compile and run to confirm any suspicions. All problems assume a command line argument is specified when the program is run.

Submit a .pdf that contains your identified vulnerabilities and **specific** code fixes.

a.

```
int foo(char *arg, char *out)
{
  strcpy(out, arg);
  return 0;
}
 int main(int argc, char *argv[])
{
  char buf[64];
  if (argc != 2)
  {
    fprintf(stderr, "a: argc != 2\n");
    exit(EXIT_FAILURE);
  }
  foo(argv[1], buf);
  return 0;
}
```

b.

```
int foo(char *arg)
{
  char buf[128];
  int len, i;

  len = strlen(arg);
  if (len > 136)
    len = 136;

  for (i = 0; i <= len; i++)
    buf[i] = arg[i];
  return 0;
}
int main(int argc, char *argv[])
{
  if (argc != 2)
  {
    fprintf(stderr, "b: argc != 2\n");
    exit(EXIT_FAILURE);
  }
  foo(argv[1]);
  return 0;
}
```

c.

```
int bar(char *arg, char *targ, int ltarg)
{
  int len, i;
  len = strlen(arg);
  if (len > ltarg)
    len = ltarg;
  for (i = 0; i <= len; i++)
    targ[i] = arg[i];
  return 0;
}
int foo(char *arg)
{
  char buf[128];
  bar(arg, buf, 140);
  return 0;
}
int main(int argc, char *argv[])
{
  if (argc != 2)
  {
    fprintf(stderr, "c: argc != 2\n");
    exit(EXIT_FAILURE);
  }
  foo(argv[1]);
  return 0;
}
```

d.

```
int foo(char *arg, short arglen)
{
  char buf[1024];
  int i, maxlen = 1024;
  if (arglen < maxlen)
  {
    for (i = 0; i < strlen(arg); i++)
      buf[i] = arg[i];
  }
  return 0;
}
int main(int argc, char *argv[])
{
  if (argc != 2)
  {
    fprintf(stderr, "d: argc != 2\n");
    exit(EXIT_FAILURE);
  }
  foo(argv[1], strlen(argv[1]));
  return 0;
}
```

Points	40
Submitting	a file upload
File Types	pdf

Due	For	Available from	Until
Oct 12, 2018	Everyone	-	-