Peyton Page
CYEN 301
Infection.txt Assignment
2020-03-20
/*

Code, compile, and execute the following code on a variety of operating systems (at the very least try a version of Linux and a version of Windows). Comment on your observations. Then comment on what you think the code is, what it does, how an attacker might use it, and what you might do to deal with such an attack.

*/

```
int main(int argc, char** argv)
{
	for(;;)
		system(argv[0]);
}
```

When I attempted to compile the code given in infection.txt (after decrypting the binary with my Binary Decoder) I was given an error that was solved by adding "#include <stdlib.h>" at the beginning of the code. Once this was solved, I was able to compile and run the given code. The code creates a loop that repeatedly runs argv[0], which restarts the code. This happens forever, and even Ctrl+C did not stop it. To stop it I had to close the terminal. I believe this attack is known as a fork bomb as it repeatedly forks the process until the system becomes unresponsive. An attacker may use this code to make a user unable to use their computer by using enough of the system resources and making the system slow and unresponsive. Perhaps they could set the code up to run immediately when the computer is booted, making the user unable to solve the issue by restarting the computer. To deal with the attack a user may need to simply close the terminal. If the system's resources are not adequate, which could cause the system to immediately slow, the user may not be able to close the terminal. One solution to preventing a fork bomb could be to limit the number of processes that a single user may have at once, preventing the fork bomb from opening too many children.