Funciones Vulnerables a Buffer Overflow

Análisis de Vulnerabilidades

Pacheco Franco Jesús Enrique

jesus.pacheco@bec.seguridad.unam.mx

16/Abril/2019

Función Vulnerable	Función Segura	Descripción de la Función
char *gets(char *str);	char *fgets(char *str, int n, FILE *stream);	Lee una línea desde stdin y la guarda en la dirección de memoria a la que apunta *str. Para cuando encuentra un salto de línea o bien un EOF cualquiera que encuentre primero.
int sprintf (char * str, const char * format, ...);	char * strncpy (char * destination, const char * source, size_t num);	Forma una cadena usando la misma mecánica que printf solo que en vez de imprimir dicha cadena está es guardada en el buffer apuntado por *str.
char * strcat (char * destination, const char * source);	char * strncat (char * destination, const char * source, size_t num);	Junta una copia de la cadena fuente a la cadena destino. El carácter de fin de cadena en la cadena destino es sobrescrito por el primer carácter de la cadena fuente y un nuevo carácter de fin de cadena es escrito al final de la nueva cadena formada por la concatenación de ambas.
char * strcpy (char * destination, const char * source);	char * strncpy (char * destination, const char * source, size_t num);	Copia el contenido de la cadena fuente en el array apuntado por *destination e incluye un carácter nulo al final de la cadena.
int vsprintf(char *str, const char *format, va_list arg);	int vsnprintf (char * s, size_t n, const char * format, va_list arg);	Guarda una cadena con formato, formada a partir de una lista de valores que le son proporcionados, en un arreglo que se le especifique.

Conclusiones

- La única función que realmente es vulnerable por naturaleza es gets() ya que por la manera en que esta implementada esta puede recibir cualquier cosa y no hay manera de limitar esto. Para las otras funciones no sucede así ya que una parte de sus argumentos es el formato y este si que se puede controlar, sin embargo, se vuelven vulnerables cuando no se les usa correctamente.

- Las implementaciones seguras de dichas funciones corrigen sus fallas implementado el uso de un entero que es la cantidad de caracteres con las que se van a estar trabajando, de esta manera se puede asegurar que no se recibirán mas caracteres de la cuenta.

Referencias

- Stack Overflow. (2014). Unsafe C functions and the replacement. [online] Available at: https://stackoverflow.com/questions/26558197/unsafe-c-functions-and-the-replacement [Accessed 16 Apr. 2019].