Vallejo Fernández Rafael Alejandro

Funciones vulnerables

Vulnerable	Segura	Acción
strcpy(char *dest, const char *src) Copia los caracteres de src a destino hasta encontrar NULL (carácter de terminación) por lo que no comprueba el número de caracteres a copiar.	**strncpy(char *dest, const char *src, size_t n)** Copia los n bytes de la cadena src en la cadena dest.	Copia la cadena src en la cadena dest.
strcat(char *dest, const char *src) Concatena los bytes de src a dest sin comprobar el tamaño de “dest”.	**strncat(char *dest, const char *src, size_t n)** Concatena los n bytes de src a la cadena dest para comprobar el tamaño de “dest”.	Concatena la cadena src a la cadena dest
gets(char *s) No se sabe cuantos caracteres leerá y continuará almacenando los caracteres despuúes del fin de buffer.	**fgets(char *s, int size, FILE *stream)** Leerá los n bytes de size del archivo que se indique por stream (STDIN u otro archivo) y los guardará en la cadena s.	Lee caracteres de la entrada estándar o archivo (fgets) y lo guarda en “s”.
scanf(const char *format, …) Lee la entrada y no se tiene en cuenta el número de caracteres leídos por lo que puede escribir en otras direcciones de memoria.	**Utilizar snprintf()**	Lee de la entrada estándar.
sprintf(char *destination, const char *format, …) No verifica el límite de “destination” por lo que puede sobreescribirse con más caracteres de los posibles.	**snprintf(char *destination, size_t n, const char *format, …)** Solamente escribirá los n bytes de la cadena format en la cadena destination.	Envía una salida con formato a la cadena “destination”.

Referencias:

Allain, A. Writing Secure Code. https://www.cprogramming.com/tutorial/secure.html

Buffer overflow attack. https://www.owasp.org/index.php/Buffer_overflow_attack