López Fernández Servando Miguel

Funciones vulnerables en C

La mayoría de las vulnerabilidades en C están relacionadas con el desbordamiento del búfer y la manipulación de cadenas. En la mayoría de los casos, esto daría como resultado una falla de segmentación, pero los valores de entrada maliciosos especialmente diseñados, adaptados a la arquitectura y al entorno podrían dar lugar a la ejecución de código arbitrario.

A continuación se muestra una tabla en la que se listan funciones inseguras susceptibles a alguna de vulnerabilidades y su contraparte segura.

Función insegura	Función segura	Acción que realiza
gets insegura ya que no revisa el tamaño de búfer permitiendo sobreescribir secciones de memoria.	*fgets* segura ya que reserva memoria dinámica a partir del tamaño del búfer.	Lee de la entrada estándar un buffer de caracteres y lo copia a una variable.
strcpy es vulnerable ya que no revisa si el número de caracteres a copiar es mayor al espacio disponible donde se copiaran.	*strlcpy* segura ya que solo copia n cantidad de caracteres.	Copia el contenido de una variable dentro de otra, esto lo hace caracter a caracter.
strncpy es insegura ya que no mitiga el “\0” al final de la copia	*strlcpy* segura ya que solo copia n cantidad de caracteres.	Copia el contenido de una variable dentro de otra, esto lo hace caracter a caracter.
sprintf no comprueba los límites del búfer y es vulnerable a los desbordamientos.	*snprintf* tiene la doble ventaja de evitar los desbordamientos de búferes y devolver el tamaño mínimo de búfer necesario para que se ajuste a toda la cadena formateada.	Envía una salida formateada a una cadena.

printf y sus derivados Otra categoría de vulnerabilidad está relacionada con los ataques de string formatting attacks, que pueden causar fuga de información, sobrescritura de memoria. Este error puede ser explotado en cualquiera de las siguientes funciones: printf, fprintf, sprintf y snprintf, es decir, todas las funciones que tienen una "format string" como argumento.	*no existe una función en específico para mitigarla* es tan sencillo como siempre hardcodear el format string.	Funciones para mostrar contenido formateado a través de la salida estándar.

Fuentes

https://security.web.cern.ch/security/recommendations/en/codetools/c.shtml