INCIDENTES DE CIBERSEGURIDAD

Unidad 1. Actividad 24

13 DE FEBRERO DE 2024

CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

Enunciado

This library implements the secure C11 Annex K1 functions on top of most libc implementations, which are missing from them.

The ISO TR24731 Bounds Checking Interface documents indicate that the key motivation for the new specification is to help mitigate the ever increasing security attacks, specifically the buffer overrun.2

The rationale document says "Buffer overrun attacks continue to be a security problem. Roughly 10% of vulnerability reports cataloged by CERT from 01/01/2005 to 07/01/2005 involved buffer overflows. Preventing buffer overruns is the primary, but not the only, motivation for this technical report

TRABAJO SOBRE SAFEC .

Descripción general

SafeC es un conjunto de extensiones del lenguaje C diseñado para mejorar la seguridad y la robustez del código C al abordar problemas comunes como los desbordamientos de búfer. SafeC se basa en el estándar C existente y proporciona funcionalidades adicionales para ayudar a prevenir errores comunes que pueden llevar a vulnerabilidades de seguridad.

El documento justificativo enumera los siguientes puntos clave para TR24731:

- Protegerse contra el desbordamiento de un buffer

- No produzca cadenas sin terminar

- No truncar cadenas inesperadamente

- Proporcionar una biblioteca útil para el código existente.

- Preservar el tipo de datos de cadena terminada en nulo

- Solo requiere ediciones locales de los programas

- Solución basada en biblioteca

- Admite verificación en tiempo de compilación

- Hacer obvios los fracasos

- Búferes cero, cadenas nulas

- Mecanismo de manejo de restricciones de tiempo de ejecución

- Admite código reentrante

- Esquema de nomenclatura coherente

- Tener un patrón uniforme para los parámetros de la función y el tipo de retorno.

- Deferencia a la tecnología existente

Y se puede agregar lo siguiente...

- proporcionar una biblioteca de funciones con comportamiento similar

- Proporcionar una biblioteca de funciones que promueven y aumentan la seguridad del código.

- Proporcionar una biblioteca de funciones que sean eficientes.

El estándar C11 adoptó muchos de estos puntos y agregó algunas variantes seguras en el Anexo K. La API segura de Microsoft Windows/MINGW hizo lo mismo, pero se desvió en algunas funciones del estándar. Además de Windows (con sus variantes msvcrt, ucrt, reactos msvcrt y wine msvcrt), hasta ahora solo el stlport no utilizado, Bionic de Android, Huawei Securec y Embarcadero implementaron esta API segura del Anexo K C11. Todavía faltan en glibc, musl, FreeBSD, darwin y DragonFly libc, OpenBSD libc, newlib, dietlibc, uClibc, minilibc.

Consideraciones de diseño

Esta biblioteca implementa desde 3.0 todas las funciones definidas en las especificaciones. En la biblioteca se incluyen extensiones de la especificación para proporcionar un conjunto complementario de funciones con comportamiento similar.

Esta biblioteca está destinada a usarse además de todas las libc existentes que carecen de las funciones seguras de C11. Por supuesto, sería mejor una integración más estrecha en el sistema libc, especialmente. con las funciones printf, scanf e IO.

Revisión del Grupo Austin de ISO/IEC WDTR 24731

Estándar C11 (ISO/IEC 9899:2011)

Estándar de codificación segura CERT C

Discusión de Stackoverflow:

Revisión de DrDobbs

Uso de errno

La especificación TR24731 dice que una implementación puede establecer errno para las funciones definidas en el informe técnico, pero no es obligatorio.

En la mayoría de los casos, los errores de ES* extendidos de safeclib no se configuran errno, solo cuando falla la llamada al sistema inseguro subyacente, se establece errno. La biblioteca

utiliza errnocódigos de retorno según lo requieren las API funcionales. Los códigos de error específicos de Safe C String y Safe C Memory se definen en el safe_errno.harchivo.

Restricciones de tiempo de ejecución

Según la especificación, la biblioteca verifica que el programa que llama no viole las restricciones de tiempo de ejecución de la función. Si se viola una restricción de tiempo de ejecución, la biblioteca llama al controlador de restricción de tiempo de ejecución actualmente registrado.

Según la especificación, múltiples violaciones de restricciones de tiempo de ejecución en la misma llamada a una función de biblioteca dan como resultado una sola llamada al controlador de restricciones de tiempo de ejecución. La primera infracción encontrada invoca el controlador de restricciones de tiempo de ejecución

Restricciones de tiempo de compilación

Con compiladores compatibles, las comprobaciones de desbordamiento de dmax y varias más se realizan en tiempo de compilación. Actualmente solo desde clang-5 con diagnose_ifsoporte. Esto verifica de manera similar a _FORTIFY_SOURCE=2si el __builtin_object_sizebúfer de destino es del mismo tamaño que dmax, y genera errores si dmax es demasiado grande.

Archivos de encabezado

La especificación establece que las diversas funciones se agregarían a los archivos de encabezado estándar C existentes: stdio.h, string.h, etc. Esta implementación separa las funciones relacionadas con la memoria en el safe_mem_lib.hencabezado, las funciones relacionadas con la cadena en el safe_str_lib.hencabezado y el resto en el encabezado. safe_lib.hencabezamiento.

Biblioteca del espacio de usuario

El sistema de compilación para la biblioteca del espacio de usuario es el conocido sistema de compilación GNU , también conocido como Autotools. Este sistema es bien comprendido y soportado por muchas plataformas y distribuciones diferentes, lo que debería permitir que esta biblioteca se construya en una amplia variedad de plataformas. Consulte la sección Plataformas probadas para obtener detalles sobre en qué plataformas se probó esta biblioteca durante su desarrollo.

Building

Para aquellos familiarizados con las herramientas automáticas, probablemente puedan omitir esta parte. Para aquellos que no lo son y quieren comenzar a crear el código, consulte a continuación. Y, para aquellos que necesiten información adicional, consulte el INSTALLarchivo en el mismo directorio.

Para construir haces lo siguiente:

```
./build-aux/autogen.sh
./configure
make
```

La instalación debe ser realizada por root, un Administratoren la mayoría de los sistemas. Lo siguiente se utiliza para instalar la biblioteca.

```
sudo make install
```

Módulo de kernel de Linux seguro

La compilación del módulo del kernel no se ha integrado en la infraestructura de compilación de autotools. En consecuencia, debe ejecutar un archivo MAKE diferente para compilar el módulo del kernel.

Building

Para construir haga lo siguiente:

```
./configure --disable-wchar
make -f Makefile.kernel
```

Esto supone que está compilando en una máquina Linux y que este archivo MAKE admite la infraestructura del sistema de compilación del kernel estándar documentada en: /usr/src/linux-kernel/Documentation/kbuild/modules.txt

Instalación

El módulo del kernel se encontrará en la raíz del árbol fuente llamado slkm.ko. El archivo testslkm.koson las pruebas unitarias que se ejecutan en la biblioteca del espacio de usuario, pero en forma de módulo del kernel de Linux para verificar la funcionalidad dentro del kernel.

Plataformas probadas

La biblioteca ha sido probada en los siguientes sistemas:

- Linux Fedora core 31 - 36 amd64/i386 glibc 2.28 - 2.36 (todos los gcc + clang)

- Mac OS X 10.6-12 con herramientas de desarrollo de Apple y macports (todos gcc + clang)

- Linux Debian 9 - 11 amd64/i386 glibc 2.24 - 2.28 (todos los gcc + clang)

- Linux centos 7 amd64

- Linux vacío amd64 musl-1.1.16

- x86_64-w64-mingw32 nativo y compilado de forma cruzada

- i686-w64-mingw32 nativo, compilado de forma cruzada y probado en Wine

- i386-mingw32 compilado cruzado

- cygwin32 gcc (nueva biblioteca)

- cygwin64 gcc -std=c99 (nuevalib)

- freebsd 10 - 13 amd64

- Imágenes de Linux Docker en qemu: i386/debian, x86_64/rhel, arm32v7/debian, aarch64: arm64v8/{debian,centos,rhel,fedora}, s390x/fedora (la única prueba big endian que pude encontrar), ppc64le/{debian ,ubuntu,fedora,centos,rhel}

- Modo de usuario Linux (UML), versión del kernel de Linux v3.5.3 con Debian Squeeze rootfs