Universitat Oberta de Catalunya

uoc.edu

Sandbox environment for IOT malware

Víctor Fernández Coderch
Máster universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Seguridad en el Internet de la Cosas

Carlos Hernández Gañán
Helena Rifà Pous

4 de Junio de 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Sandbox environment for IOT malware*
Nombre del autor:	*Víctor Fernández Coderch*
Nombre del consultor/a:	*Carlos Hernández Gañán*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	*Máster en seguridad de las tecnologías de la información y de las comunicaciones*
Área del Trabajo Final:	*Seguridad en el internet de las cosas*
Idioma del trabajo:	*Castellano*
Palabras clave	*IOT, malware, sandbox*

Resumen del Trabajo

En la actualidad, el crecimiento tan elevado del número de dispositivos interconectados ha establecido al internet de las cosas como un buen objetivo para los desarrolladores de malware. Sus características lo hacen ideal para ser objetivos de ataque, la baja seguridad, el largo tiempo que pasan conectados a Internet, interconectados a otros dispositivos, son el eslabón más débil contra el que iniciar un ataque resultando un punto de acceso a otros dispositivos.

Todo ello hace que se requiera poner énfasis en el análisis de malware estudiando específicamente el desarrollado para estos dispositivos. Para poder analizar malware es necesario disponer de herramientas que faciliten esta tarea. En la actualidad existen los sandbox para desempeñar esta tarea.

El objetivo de este trabajo es preparar un sandbox, que se define como un entorno aislado, seguro, donde los cambios realizados por el malware serán reversibles. Esto implica que el malware tendrá vía libre para realizar todas las modificaciones para las que esté programado dejando el respectivo rastro, pero permitiendo posteriormente dejar la máquina como estaba en su origen, previo a la ejecución del malware. Este entorno estará preparado y configurado para poder replicar la ejecución de malware dirigido a arquitecturas usadas comúnmente en el IoT como ARM o MIPS.

El resultado de cada análisis vendrá dado por la ejecución del malware dentro del sandbox. Esta ejecución devolverá un conjunto de información muy útil que irá desde el análisis de la estructura del fichero (estático) hasta el detalle de cada paso de ejecución (dinámico) que realizará cambios en el sistema invitado mediante llamadas a sistema durante su ejecución, incluyendo además un análisis del uso de la red durante este proceso.

Todo este entorno permitirá generar un informe del cual se podrán extraer

conclusiones del propósito del malware y generar una prevención eficaz de cara a futuras infecciones.

Abstract

Currently, the high growth in the number of interconnected devices has established the internet of things as a good target for malware developers. Its characteristics make it ideal to be targets of attack, low security, long time spent connected to the Internet, interconnected to other devices, are the weakest link against which to initiate an attack resulting in a point of access to other devices.

All this makes it necessary to emphasize the analysis of malware specifically studying the one developed for these devices. To be able to analyze malware, it is necessary to have tools that facilitate this task. Now there are sandboxes to perform this task.

The objective of this work is to prepare a sandbox, which is defined as an isolated, safe environment, where the changes made by the malware will be reversible. This implies that the malware will have the free way to make all the modifications for which it is programmed leaving the respective trace but allowing later to leave the machine as it was at its origin, prior to the execution of the malware. This environment will be prepared and configured to replicate the execution of malware aimed at architectures commonly used in the IoT such as ARM or MIPS.

The result of each analysis will be given by the execution of the malware within the sandbox. This execution will return a very useful set of information that will go from the analysis of the structure of the file (static) to the detail of each execution step (dynamic) that will make changes in the guest system through system calls during its execution, including an analysis of the use of the network during this process.

All this environment will generate a report which can draw conclusions from the purpose of the malware and generate an effective prevention for future infections.

Índice

Lista de figuras

1. Introducción

1. Contexto y justificación del Trabajo

En los últimos años el crecimiento de dispositivos interconectados a la red ha ido en aumento y se espera que sigan creciendo [1]. Este gran aumento se explica considerando que, en la actualidad, cualquier objeto cotidiano puede estar conectado a la red. Por poner algunos ejemplos, bombillas, neveras, coches, pulseras, relojes… Todo este conjunto de dispositivos es conocido como IoT (internet de las cosas). Este término se refiere explícitamente a la interconexión de dispositivos cotidianos con internet [2]. La perspectiva del concepto IoT se ideó con la intención de reflejar la conexión de más objetos a internet que personas. En el internet de las cosas también podemos encontrar dispositivos de una gran relevancia e impacto, dependiendo de ellos procesos críticos en centrales nucleares, en plantas de fabricación… Lo que es conocido como EIoT (IoT empresarial). Desafortunadamente se conocen algunos casos de ataques contra dispositivos IoT recientes [3]

Este enorme despliegue tecnológico tan veloz entraña riesgos que, se tienen en cuenta cuando el riesgo ya es muy alto, y es que es difícil que la velocidad a la que avanza la tecnología sea seguida a la par por la seguridad.

La necesidad de que estos dispositivos tengan medidas de seguridad es evidente por la relevancia que tienen. Principalmente, podríamos destacar los siguientes aspectos:

- El gran número de dispositivos disponibles, conectados por mucho tiempo seguido a la red pueden facilitar en gran medida ciertos tipos de ataques.
- La criticidad de algunos de estos dispositivos que son necesarios en tareas de relevancia (Áreas como la seguridad, construcción, salud…).
- La privacidad y la seguridad que se confían a estos dispositivos.

En cuanto a las acciones que se pueden abarcar para hacer la IoT sea más segura, se dispone de un gran abanico de posibilidades ya sea a nivel de diseño o fabricación, que pueden ir desde cifrados robustos al control de errores [4].

El mundo IoT es muy heterogéneo y este hecho complica que se puedan realizar estudios de gran alcance y se procederá a definir las características de los dispositivos bajo los cuales se realizará el estudio.

En este trabajo se va a poner foco en cómo se realiza el abuso de los dispositivos IoT por malware. Se conoce como malware cualquier software malintencionado capaz de realizar algún tipo de daño infiltrándose de manera ilícita en un dispositivo, en general con objetivos concretos como pueden ser obtener beneficio económico, robar información… [5]

El desarrollo de malware, al igual que los dispositivos IoT está en auge. Según un análisis realizado por Kaspersky [6] el número de muestras de malware se ha multiplicado casi por cuatro, entre el año 2017 y 2018 como se puede apreciar en la figura 1. Cuantas más versiones de malware existan, más

robustas y variadas tendrán que ser las medidas que les puedan poner freno. Estas medidas requieren un estudio de muestras de malware para analizar las acciones que realizan y con ello poder detectarlas.

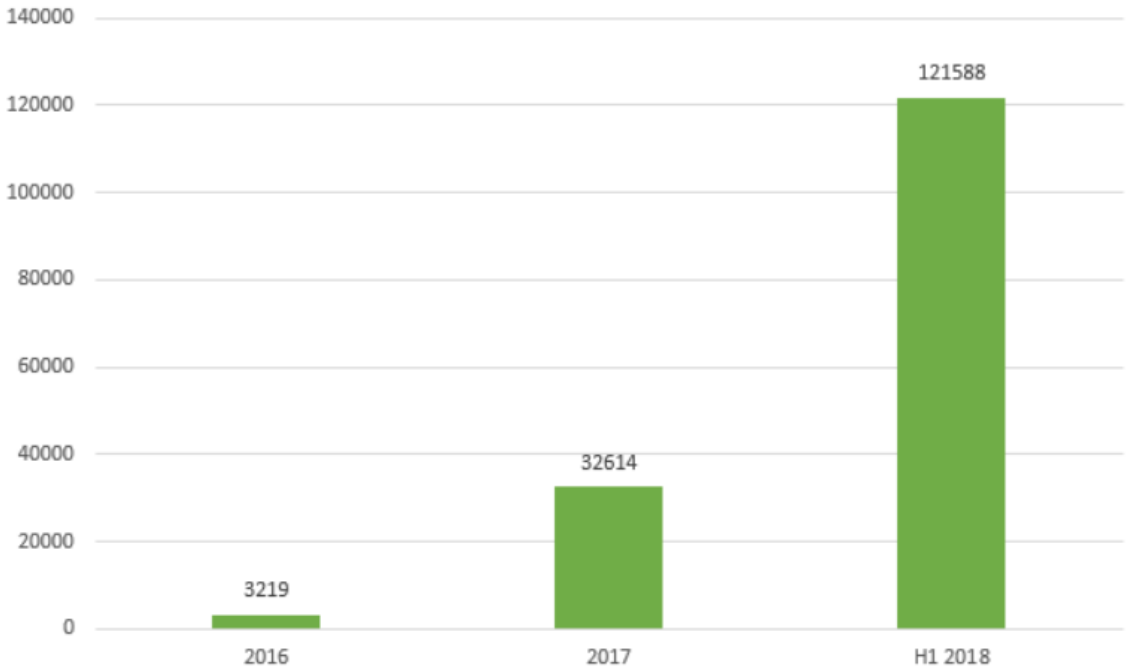


Figura 1. Número de samples de malware para dispositivos IoT en la colección de Kaspersky

Para poder realizar un análisis de muestras de malware se procederá a utilizar software específico que lo permita ejecutar de forma segura y controlada sin causar daños, este entorno se conoce como sandbox o caja de arena. Un sandbox es un entorno limitado en el que se van a ejecutar aplicaciones, como si de un dispositivo normal se tratara, la diferencia es que todo lo que se ha cambiado o creado, no se guarda cuando la aplicación deja de ejecutarse [7].

Mediante el análisis de la ejecución de algunas muestras de malware se podrá proceder a implementar medidas que los detecten y de esta manera, se puedan eliminar de cualquier dispositivo al que hayan accedido.

Es importante tener en cuenta que el malware es capaz de detectar estos entornos controlados y evitar que se realicen las acciones fraudulentas hasta que se detecte una acción provocada por un usuario, que no se va a dar de manera natural en un sandbox, también son capaces de detectar algunas características del sistema que contenga la sandbox mediante llamadas a funciones del sistema que devuelven información relativa al mismo [8]. Este punto será sujeto de estudio en este trabajo.

A través de todo el análisis realizado anteriormente, se procederá a realizar el análisis del malware y a extraer conclusiones de su ejecución a través del sandbox escogido.

Para finalizar la introducción, se reflejan algunas de las ventajas del internet de las cosas y porqué son tan interesantes en todos los ámbitos, sin dejar de tener en cuenta sus inconvenientes [9][10][11][12]:

a) Captación de datos: Cuanta más información dispongamos de diferentes fuentes (todos los dispositivos interconectados entre sí o conectados a internet son una fuente inagotable de datos) se podrán tomar unas mejores decisiones. Por ejemplo, el stock disponible en una nevera, que podría automáticamente realizar un pedido.
b) Productividad: Los dispositivos son capaces de automatizar procedimientos, monitorizar o ejecutar algún tipo de mantenimiento de una forma más económica, más eficiente y rentable que la que puede ofrecer un humano.
c) Heterogeneidad del entorno: Este concepto es abarcable en todos los ámbitos de la vida. Desde las industrias más complejas y técnicas, hasta los hogares.
d) Respuestas a necesidades: Este nuevo mercado, puede proveer de nuevos servicios que respondan de una forma más adecuada a las necesidades específicas de los ciudadanos y/o las empresas.

Podemos enumerar algunos de los inconvenientes:

a) Inversión: Para que este nuevo concepto cale en la sociedad, es necesaria inversión que permita adaptar la vida cotidiana.
b) Servicios requeridos: La implantación de la tecnología requiere que haya empresas que proporcionen una serie de servicios que hagan viable el mantenimiento de este modelo.
c) Brechas tecnológicas: La capacidad de adaptación al IoT dependerá de los recursos disponibles para invertir en tecnología, lo cual hará que cada realidad tenga un distinto nivel de avance.
d) Legislación: La legislación siempre va un paso atrás de cada revolución, como en esta, será necesario que se legisle en consecuencia al avance tecnológico en el IoT.
e) Riesgos inherentes: Delegar en la tecnología ciertos tipos de acciones, puede suponer un problema en caso de malfuncionamiento.
f) Privacidad y seguridad: El avance tecnológico no siempre va ligado a la seguridad, y es que cuantos más dispositivos personales/productivos hayan conectados a internet, más complicado será mantener esos datos fuera del alcance de terceras personas.

2. Objetivos del Trabajo

1- Seleccionar un subconjunto de dispositivos IoT sobre los que realizar el trabajo.
2- Escoger el sandbox que mejor se adapte a las necesidades.
3- Seleccionar muestras de malware.
4- Evitar la detección de sandboxes implementada en las muestras de malware a estudiar.
5- Especificar y detallar el análisis que se va a realizar sobre el malware.
6- Aplicar todo el conocimiento adquirido para poner en práctica el análisis del malware.
7- El objetivo global, como suma de todos los anteriores será obtener un sistema fiable que permita detectar un alto porcentaje de las muestras respetando las condiciones previamente expuestas.

3. Enfoque y método seguido

El desarrollo del trabajo dispondrá de tres partes bien diferenciadas, las cuales deben ejecutarse en orden para que la estructura sea congruente con el resultado que se quiere obtener. En todas las fases se incluye la redacción de la memoria, de forma paralela al resto de tareas:

Investigación: Se realizará un trabajo de investigación, estudiando detalladamente todas las herramientas que son necesarias para poder desarrollar seleccionándolas con criterio y objetividad. Además de la investigación de las herramientas, se debe cerrar el planteamiento del estudio y ajustando el enfoque sobre una porción de dispositivos IoT y muestras de malware que sea viable para asumirla en el tiempo de ejecución estimado para un TFM.

Desarrollo e integración: Una vez todo el estudio se haya finalizado, se procederá a instalar el entorno, configurarlo y comenzar a recolectar muestras de malware que posteriormente se procederán a estudiar.

Resolución: Finalmente, tras realizar el análisis y las pruebas necesarias, se procederá a cerrar la memoria y extraer las conclusiones del estudio. Además, se preparará la presentación.

4. Planificación del Trabajo

A continuación, se adjunta tabla con las fechas de las entregas y diagrama de Gannt:

Nombre tarea	FECHA INICIO	FECHA FIN
Redacción de la memoria	3/5	5/31
Investigación		
Estudiar dispositivos IoT	3/5	3/15
Estudiar sandboxes disponibles	3/18	4/3
Estudiar malwares	4/3	4/22
Estudiar el fingerprint de sandboxes	4/22	4/29
Desarrollo e integración		
Implementación VM + sandbox	4/29	5/6
Configuración que evite evasión de sandboxes	5/6	5/13
Selección muestras malware	5/13	5/20
Ejecución pruebas y obtención resultados	5/10	5/18
Resolución y resultados		
Exponer conclusiones y objetivos cumplidos	5/26	5/31
Finalización memoria (últimos ajustes)	5/31	6/4
Preparación presentación	6/5	6/11

Figura 2. Planificación temporal

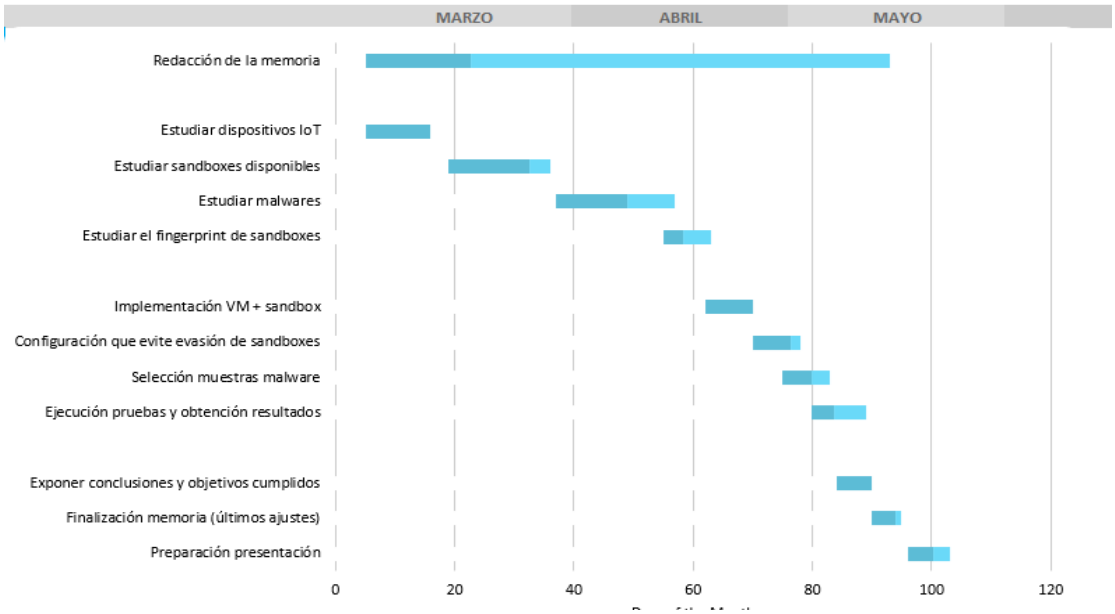


Figura 3. Diagrama de Gannt

5. Breve descripción de los otros capítulos de la memoria.

La estructura de la memoria está bien diferenciada en 2 grandes apartados. En el primero de ellos, “Análisis dispositivos IoT” se hace un repaso a las características principales de los dispositivos IoT más conocidos, los sistemas operativos más utilizados, se revisan las soluciones de sandboxing disponibles actualmente con las que poder simular un entorno IoT para poder ejecutar muestras de malware.

También se realiza un breve análisis de los tipos de malware más conocidos que se pueden encontrar actualmente y los tipos de análisis de los que disponemos para identificarlos, ya sea por su comportamiento o por la estructura de su código. Es importante conocer los tipos de análisis que existen para poder identificarlos posteriormente en la herramienta sandbox ya que se dispondrán de ellos en los resultados.

Para acabar este apartado se exponen las dificultades que se presentan en el análisis de malware debido a sus características y la heterogeneidad del entorno. Este capítulo finaliza con la elección del sandbox y la arquitectura que determinará el tipo de muestras de malware que se procederán a analizar en el sandbox.

El siguiente capítulo ejemplifica la instalación y utilización del software sandboxing de Cuckoo, incluyendo todas las dependencias y especificando paso por paso los comandos a ejecutar para finalmente acabar realizando un análisis de una muestra de malware preparada para MIPS.

En la última parte de la memoria se realiza una comparación entre las diferencias que resultan de analizar un binario para Linux y otro para un ejecutable preparado para otra arquitectura.

Finalmente se exponen las conclusiones extraídas de todo el trabajo.

2. Análisis dispositivos IoT

1. Características deseables de los dispositivos IoT y SOs:

La naturaleza de los dispositivos IoT, dada su finalidad, hace que dispongan de recursos energéticos y hardware limitados. Estas características físicas requieren que los sistemas integrados en ellos sean eficientes, flexibles y livianos [14].

Desde el punto de vista de las características técnicas deseables para un dispositivo de estas características, se pueden definir las siguientes:

Arquitectura: En cuanto a la arquitectura del Kernel, podemos diferenciar varios tipos, monolíticos, micronúcleos, núcleos híbridos o exonúcleos. Es ideal escoger el que ofrece la mayor capa de abstracción para facilitar su manejo, fiabilidad y reducir la complejidad [15].

Programación: Los factores que determinan como deben ser programados estos dispositivos debe depender de características como el paralelismo, la jerarquía de memoria, la concurrencia… Se deberán utilizar lenguajes de programación adecuados a cada dispositivo en función de las necesidades. También es importante proveer de APIs que faciliten y permitan desarrollar de una forma eficiente.

Scheduling: La gestión del tiempo del Kernel es un factor clave en el rendimiento del dispositivo. El sistema debe ser capaz de finalizar las tareas en un determinado tiempo, siendo capaz de reducir el consumo energético y permitir la multitarea.

Conectividad: La conectividad es un requisito básico de los dispositivos IoT. Deben poder comunicarse con un consumo significativamente bajo de energía. Dada esta consideración, surgen tecnologías como ZigBee, Bluethoot, Z-Wave... que permiten adaptarse a los requisitos individuales de los dispositivos IotT. Dado el alto número de dispositivos IoT deben ser capaces de gestionar direcciones IPv6

La gestión de la memoria: La gestión de la memoria debe proveer herramientas que abstraigan la gestión de la caché, la asignación de la memoria, proveer memoria virtual, mapeado lógico – físico… a través del back-end.

La portabilidad: Los sistemas operativos deben ser compatibles independientemente de la plataforma de hardware en la que se instalen.

Dadas todas las características deseadas en los dispositivos IoT, existen múltiples opciones de sistemas operativos funcionales que se pueden instalar y utilizar en los dispositivos IoT en la actualidad, entre ellos están los siguientes:

- Contiki
- RIOT
- TinyOS
- LiteOS
- FreeRTOS
- Mantis OS
- Nano RK

- uC/OS-III
- uCLinux
- OpenTag
- Erika Enterprise

Todos los SOs expuestos anteriormente están enfocados a consumir poca energía, son SOs ligeros destinados a terminales IoT.

Existen otro tipo de terminales IoT más avanzados, como podemos apreciar en la tabla 1 y 2 en la que se muestran algunos de ellos. Los dispositivos que se muestran en la tabla son los que actualmente son más conocidos, disponiendo todos ellos de sistemas operativos más avanzados, la mayoría de ellos basados en Linux, además de tener una capacidad de procesamiento decente, memoria y capacidad de almacenamiento.

En la siguiente tabla se repasa los sistemas operativos utilizados en estos dispositivos y los dispositivos que los integran.

SO	Descripción
OpenWrt-Yun (based on GNU/Linux)	Sistema operativo basado en Linux dirigido a sistemas integrados.
Raspbian	Sistema operativo basado en debian para la Raspberry
Arch Linux ARM distribution	Distribución de Linux para ordenadores ARM.
FreeRTOS	Núcleo de sistema operativo para sistemas integrados portado en 35 plataformas de microcontroladores.
Debian	Sistema operativo basado en el Kernel de Linux.
Android	Sistema operativo destinado a los moviles. Basado en una versión modificada del Kernel de Linux.
Yocto 1.6 (Fedora)	Proyecto open source con el objetivo de la creación de distribuciones de Linux dedicadas a los sistemas integrados y dispositivos IoT.
Windows 10 IoT Core	Sistema operativo de Microsoft diseñado para dispositivos integrados.
Ubuntu	Sistema operativo basado en el Kernel de Linux basado en debian. Dispone de una versión para IoT.

Pidora	Distribución de software de Linux para la rasperry pi. Específicamente diseñado para la arquitectura ARMv6
Arch Linux	Sistema operativo para ordenadores x86-x64 orientada a usuarios avanzados.

Figura 4. Sistemas operativos utilizados en IoT

En la siguiente tabla, se exponen algunos de los dispositivos IoT más famosos en el mercado, con sus características en cuanto a arquitectura y sistemas operativos soportados [16]:

Dispositivo IoT	**Procesador**	**SO soportados**
SparkFun Blynk Board	Tensilica L106 32-b	No
Arduino Yun	Atheros AR9331	OpenWrt-Yun
Raspberry Pi 3	ARM Cortex-A53 64-bit	Arch Linux, Ubuntu…
cloudBit	ARM 926EJ-S core	Customized Arch Linux ARM
Photon	ARM Cortex M3	FreeRTOS
BeagleBone Black	AM335x ARM Cortex-A8	Debian, Android, Ubuntu
Pinoccio	ATmega256RFR2	No
UDOO	ARM Cortex- A9	UDOOUbuntu, Android, Arch Linux
Samsung Artik 10	ARM A15x4	Yocto 1.6 (Fedora)

Figura 5. Dispositivos IoT, SO soportados y procesadores utilizados

Como se puede apreciar, la mayoría utilizan arquitecturas basadas en ARM y sistemas operativos basados en Linux, como Ubuntu, Arch Linux o Android.

2. Software sandbox

Se revisan las principales soluciones de sandboxing disponibles en el mercado actualmente.

Cuckoo sandbox:

Cuckoo sandbox es un sistema automatizado de análisis de malware de código abierto escrito totalmente en Python. Su arquitectura está basada en un software de administración central que maneja la ejecución y análisis de muestras. Cada análisis se inicia en una máquina virtual aislada. Por lo tanto, tendremos una máquina host y una cantidad determinada de máquinas guest. Está diseñado para ser ejecutado como una aplicación standalone [17].

Cuckoo, a través de un archivo sospechoso es capaz de proporcionar un informe detallado que describe el comportamiento del archivo, ejecutado bajo un entorno seguro.

Los ficheros que es capaz de analizar según la documentación son los siguientes:

- Ejecutables de Windows
- DLL files
- PDF
- Microsoft Office
- URLs y ficheros HTML
- PHP scripts
- Ficheros CPL
- Visual Basic (VB) scripts
- Archivos ZIP
- Java JAR
- Ficheros Python

Cuckoo soporta los siguientes softwares de virtualización (software que permite crear máquinas virtuales con un sistema operativo):

- ESX / ESXI
- KVM
- QEMU
- VirtualBox
- VMware
- vSphere
- XenServer
- Proxmox

Cuckoo dispone de una interfaz web, pero también se pueden ejecutar los análisis a través de la consola. Cuckoo tratará de determinar el mejor método

de análisis y la imagen virtual que ejecutar, a pesar de que se puede especificar manualmente. Una vez se ha solicitado realizar un análisis, Cuckoo levantará la máquina virtual apropiada y mediante el fichero “agent” que se instala dentro de la máquina objetivo se traspasará el archivo para posteriormente ejecutarlo [18].

Finalmente, la interfaz web mostrará los resultados del análisis en un formato entendible. Se puede configurar el formato de salida de los datos resultado, se podría seleccionar obtener un fichero pdf, html, json…

Joe Sandbox

Ejecuta ficheros automatizados en un entorno controlado y monitoriza el comportamiento sospechoso de aplicaciones y sistemas operativos. A partir la información recogida se genera un informe [19].

Las características principales son:

- Reporte del análisis del comportamiento del sistema y de la red.
- Contiene más de 410 firmas para detectar y clasificar el comportamiento malicioso.
- Permite utilizar un entorno virtual o un entorno físico para realizar los análisis.
- Dispone de reglas Yara para análisis de malware avanzado.
- Análisis de red utilizando un IDS como Snort.
- Además de los reportes generados en los formatos estándar, es capaz de generar datos suplementarios como PCAP, capturas de pantalla…

Limon sandbox:

Proyecto público en GitHub escrito en Python que es capaz de recolectar analizar y reportar en tiempo de ejecución comportamientos de malware. Es capaz de inspeccionar el malware antes de su ejecución, mientras se ejecuta y después de la ejecución mediante análisis estático y dinámico utilizando software open source [20].

Para ello utiliza las siguientes herramientas:

- YARA
- La API pública de VirusTotal
- ssdeep
- string utility
- ldd
- readelf
- Inetsim
- Tcpdump
- strace
- Sysdig

- Volatility

Los tipos de ficheros que es capaz de analizar son los siguientes:

- Fichero ELF
- Perl Script
- Python script
- Shell script
- Bash script
- PHP script
- Loadable Kernel module (LKM)

Kaspersky Sandbox Lab:

Proporciona software de virtualización para Windows y Android OS [21].

Los artefactos que es capaz de procesar son los siguientes:

- Para Windows: cualquier archivo ejecutable: * .exe, * .dll, objetos .NET, archivos de MS Office, archivos PDF…
- Para Android: APK (DEX).

Y además el sandbox analiza una URL, detectando los siguientes eventos: descargas, JavaScript, ejecución de Adobe Flash entre otros.

La información recogida por el sandbox es la siguiente:

- Registros de ejecución de la aplicación (llamadas a la función API con sus parámetros, eventos de ejecución)
- Volcados de memoria
- Volcadores de módulos cargados.
- Cambios en el sistema de archivos, registro.
- Tráfico de red (archivos PCAP)
- Capturas de pantalla.

Detecta la evasión de malware y es capaz de simular acciones de usuario, clicks, pulsaciones de teclado.

Por otro lado, este software de virtualización es capaz de generar las características de la VM de manera aleatoria.

3. Malware

Malware o software malicioso describe cualquier programa o código que es perjudicial para los sistemas.

El malware busca invadir, dañar, ordenadores, sistemas informáticos, redes… Su objetivo es hacerse con el control parcial sobre las operaciones de un dispositivo. El objetivo de llevar a cabo estas acciones en general es adquirir dinero de forma ilícita a través de la obtención de información (credenciales bancarias, datos que tienen un precio…). Aunque el malware no puede dañar el hardware físico de los sistemas, puede realizar cualquier modificación sobre

la información contenida en estos, además de espiar al usuario que está haciendo uso del dispositivo [5].

A continuación, se explican brevemente los tipos de malware conocidos:

a) Adware es un software no deseado diseñado para mostrar anuncios en su pantalla, generalmente en un navegador web. Por lo general, utiliza un método poco formal para disfrazarse de legítimo o para usar otro programa para engañarlo para que lo instale en su PC, tableta o dispositivo móvil.
b) El spyware es un malware que observa en secreto las actividades del usuario de la computadora sin permiso y lo reporta al autor del software.
c) Un virus es un malware que se adjunta a otro programa y, cuando se ejecuta, generalmente por el usuario, se replica al modificar otros programas de computadora e infectarlos con sus propios bits de código.
d) Los gusanos son un tipo de malware similar a los virus, que se autorreplican para propagarse a otras computadoras a través de una red, generalmente causando daño al destruir datos y archivos.
e) Un troyano es uno de los tipos de malware más peligrosos. Por lo general, se representa a sí mismo como algo útil para engañarte. Una vez que está en su sistema, los atacantes detrás del troyano obtienen acceso no autorizado a la computadora afectada. Desde allí, los troyanos se pueden usar para robar información financiera o instalar amenazas como virus y ransomware.
f) El ransomware es una forma de malware que bloquea el dispositivo y / o encripta sus archivos, luego lo obliga a pagar un rescate para recuperarlos. Ransomware es un arma que se escoge por los ciberdelincuentes porque permite exigir un pago rápido y rentable a través de criptomonedas, que hacen que sea muy difícil seguir el rastro. El código fuente del ransomware es fácil de obtener a través del mercado negro y proteger a los usuarios es difícil.
g) Rootkit es una forma de malware que otorga al atacante privilegios de administrador en el sistema infectado. Por lo general, también está diseñado para permanecer oculto para el usuario, para otros softwares y para el propio sistema operativo.
h) Un keylogger es un malware que registra todas las pulsaciones del usuario en el teclado, por lo general, almacena la información recopilada y la envía al atacante, quien busca información confidencial como nombres de usuario, contraseñas o detalles de tarjetas de crédito.
i) El cifrado malintencionado, también denominado a veces "drive-by mining" o "cryptojacking", es un malware cada vez más frecuente que suele instalar un troyano. Permite que otra persona use su computadora para extraer criptomonedas como Bitcoin o Monero. Así que, en lugar de permitirle sacar provecho de la potencia de su propia computadora, los cryptominers envían las monedas recolectadas a su propia cuenta y no a la suya. Esencialmente, un cryptominer malicioso está robando sus recursos de una computadora de un tercero para ganar dinero y beneficiarse.
j) Los exploits son un tipo de malware que aprovecha los errores y vulnerabilidades de un sistema para permitir que el creador del exploit tome el control. Entre otras amenazas, las vulnerabilidades están

vinculadas a la publicidad maliciosa, que ataca a través de un sitio aparentemente legítimo que, sin saberlo, extrae contenido malicioso de otro lugar que no es legítimo. Posteriormente, el contenido incorrecto intenta instalarse en su computadora a través de una descarga automática. No es necesario hacer clic. Tan solo es necesario dar con el servidor infectado.

4. Malware samples

Se entiende como malware samples o muestras de malware, ficheros ejecutables creados a partir del código fuente del software malicioso, que son utilizados por investigadores para analizar las técnicas mediante las cuales se realizan los ataques para desarrollar medidas de seguridad, tanto a nivel de red como de comportamiento hacía el sistema. Algunas webs incluyen binarios compilados abiertos al público para su análisis y estudio [21].

Dada la naturaleza de las muestras que se quieren analizar, en el caso de este trabajo se buscarán ficheros binarios que se puedan ejecutar en MIPS o ARM. La cantidad de archivos disponibles lo suficientemente documentados para tener la seguridad de que lo que se está descargando es lo que se está buscando hace que disminuya el número de muestras válidas que sean apropiadas para lo que hemos preparado el sandbox.

Existen también proyectos que facilitan bases de datos a través de la web (accesibles mediante navegador) que recogen y almacenan las URLs utilizadas por los softwares maliciosos para descargar otros archivos maliciosos o virus. En la base de datos URLhaus [22] se encuentran bastantes muestras de ficheros binarios compilados para ejecutarse en las arquitecturas necesarias. En MalShare [23] también hay ubicadas varias muestras para ARM.

5. Tipos de análisis

La realización del análisis del código se puede llevar a cabo de dos formas. Diferenciándolas en función del momento en el que se lleva a cabo (antes o después de su ejecución) podemos diferenciar dos tipos. Estático, que sería el análisis realizado previo a su ejecución y el análisis dinámico, que es el análisis posterior a su ejecución y conlleva el estudio de las acciones que se van realizando a través del tiempo:

Análisis estático [24]:

Esta técnica intenta identificar el código malicioso desempaquetando y descompilando la aplicación. Es relativamente rápida y es ampliamente utilizada en el análisis preliminar.

El análisis estático implica que se analiza el código sin ejecutarlo. Tras realizar la descompilación, se puede utilizar una API para cargar el ejecutable en memoria lo que podría llegar a mostrar signos de actividad maliciosa.

En general los creadores de malware, para dificultar su análisis realizan primero una ofuscación del código y posteriormente lo empaquetan. Por lo tanto, con las cadenas resultantes del desensamblado podríamos no obtener mucha información relevante. El código real se encuentra comprimido y por tanto no se puede leer. Normalmente se incluye un pequeño fragmento de

código que descomprime el código real y lo acaba ejecutando. Por lo tanto, la única parte entendible por el desempaquetador será esta. Por ejemplo, upx es el programa de empaquetado utilizado para los binarios (en sistemas Windows).

Dependerá del tipo de fichero que estemos analizando, la información que revelará el propósito de ese archivo (cada fichero tiene su propio formato de cabeceras…). Algunos ejemplos son los ficheros PE (Portable executable en Windows, ELF (Executable and Linkable Format, utilizado en sistemas Unix) entre otros.

El gran inconveniente que tiene el análisis estático es que las técnicas de ofuscación y distribución del malware actual disminuyen su eficiencia. Por lo tanto, la característica fundamental que permite conocer el verdadero propósito de un software en su comportamiento es a través de la ejecución completa realizando un análisis paso por paso.

Análisis dinámico [25]:

Está técnica busca identificar un comportamiento malicioso después de implementar y ejecutar una aplicación en un entorno controlado. Se requiere un proceso automatizado para que interaccione con la aplicación dado que el desencadenante del comportamiento malicioso suele producirse tras lanzarse algún evento determinado, generalmente producido por la interacción humana.

Normalmente en los sandbox, se simulan eventos de forma aleatoria intentando simular la interacción humana, lo que puede hacer que no se ejecute la acción concreta que acaba disparando la ejecución del malware, así ocultando su propósito.

6. Fingerprinting sandbox

Actualmente el malware es analizado en máquinas virtuales a través de herramientas sandbox. Cuckoo es una de ellas y ofrece la posibilidad de logar todas las acciones realizadas por el malware en la VM. Para proteger el malware de ser evaluado y analizado por estas herramientas, se emplean técnicas de reconocimiento de sandboxes para decidir ejecutarlo o no. A continuación, se exponen dos de las características que el malware busca para detectar si está en un entorno sandbox [26]:

- En el caso concreto del uso de Cuckoo, cuando se instala se crea la carpeta /Cuckoo, es sencillo para el malware buscar esta ruta y si la encuentra, quiere decir que está dentro de un entorno Cuckoo.
- Buscar las guest additions. Las VirtualBox guest additions es un paquete de software que añade una mejor integración entre el host y la VM.

Se pueden aplicar soluciones para evadir estas técnicas, como cambiar la ruta por defecto de Cuckoo.

Existen otros análisis que los malwares realizan mediante llamadas al sistema que devuelven información del propio sistema, como la resolución de la pantalla, tamaño de la RAM, el fabricante del sistema… En general toda esta información se debería preparar manualmente para engañar a los fingerprints realizados sobre los sistemas operativos y así poder obtener la ejecución de los malwares.

7. Dificultades

Realizar el análisis de ejecutables de Linux potencialmente sospechosos requiere afrontar una serie de desafíos específicos, a continuación, se exponen algunos de ellos [27]:

La diversidad:

La amplia diversidad de entornos es una dificultad a la hora de plantear soluciones contra el malware. En general se cree que dando soporte a diferentes arquitecturas es suficiente (por ejemplo, ARM o MIPS), pero el problema es más complejo. Sistemas de análisis de malware para los ejecutables de Windows, MacOS o Android pueden confiar en información detallada sobre el entorno de ejecución subyacente. El malware basado en Linux puede dirigirse a un conjunto muy diverso de objetivos, tales como enrutadores de Internet, impresoras, cámaras de vigilancia, televisores inteligentes, o dispositivos médicos lo cual complica el análisis. Sin conocer el objetivo (los binarios del programa no especifican el objetivo) es muy difícil de configurar correctamente el entorno idóneo.

Arquitecturas informáticas:

Linux es conocido por soportar decenas de arquitecturas. Diferentes escenarios de pruebas, además de distintos componentes de análisis específicos. En un trabajo reciente que cubre la botnet Mirai [28], los autores apoyaron tres arquitecturas: MIPS de 32 bits, ARM de 32 bits y x86 de 32 bits. Sin embargo, esto cubre una pequeña fracción del malware global disponible para Linux. Por ejemplo, estas tres arquitecturas. solo cubren una pequeña parte del total. Algunas familias (como ARM) son particularmente difíciles de apoyar dadas la gran cantidad de arquitecturas de CPU diferentes que contiene.

Bibliotecas:

El formato de archivo ELF permite a un SO Linux especificar un loader arbitrario, que es responsable de cargar y preparar el ejecutable en memoria. Una copia del loader solicitado puede no estar presente en el entorno de análisis, evitando así que la muestra comience su ejecución. Además, los binarios enlazados dinámicamente esperan sus bibliotecas requeridas para estar disponibles en el sistema de destino, lo que significa que, si no se dispone, el malware no se ejecuta. Estos aspectos afectan a una porción significativa de los datos. Un ejemplo común son los programas de Linux que están vinculados dinámicamente con uClibc o musl, alternativas más pequeñas y con mejor rendimiento al tradicional glibc. Se requiere tener el mayor número de alternativas instaladas para permitir que se ejecute la mayor parte posible del malware.

Sistema Operativo:

Si se realiza un trabajo centrado en los binarios de Linux, supone igualmente un reto distinguir los programas ELF compilados para Linux desde otros sistemas operativos compatibles con ELF, como FreeBSD o Android. Los encabezados ELF incluyen un campo "OS / ABI" que especifica qué sistema operativo se requiere en la ejecución, pero este campo no se suele informar. Los binarios ELF para Linux y Android especifican una "Sistema V" genérico OS / ABI. Finalmente, mientras que un binario. compilado para FreeBSD se

puede cargar y ejecutar correctamente bajo Linux, este es solo el caso de programas enlazados dinámicamente. De hecho, los números y argumentos de syscalls para Linux y FreeBSD generalmente no coinciden, y por lo tanto estáticamente los programas vinculados suelen fallar cuando se encuentran con diferencias. Estas diferencias también pueden existir entre diferentes versiones del Kernel de Linux, y las modificaciones personalizadas no son raras en softwares embedidos. Esto tiene dos consecuencias importantes: Por un lado, hace que sea difícil compilar un conjunto de datos de malware basado en Linux. Por otro lado, también resulta en el hecho de que incluso los binarios de Linux bien generados pueden no ejecutarse correctamente en un sistema genérico de Linux.

Static Linking:

Cuando un binario está enlazado estáticamente, todas sus dependencias de biblioteca se incluyen en el binario resultante como parte del proceso de compilación. La vinculación estática puede ofrecer varias ventajas, incluyendo hacer el binario resultante más portátil (se ejecuta, aunque sus dependencias no estén físicamente en el entorno destino) y por lo que es más difícil ingeniería inversa (añade dificultad identificar de que biblioteca provienen las funciones utilizadas en su ejecución).

La vinculación estática introduce otro nuevo desafío para el análisis de malware. Incluir estos binarios puede permitir a la aplicación resultante no ejecutarse en cualquier entorno externo para realizar llamadas a sistema. En este caso los programas no llaman directamente a sistema, sino que invocan funciones de API de nivel superior (normalmente parte de la libc) que a su vez envolver la comunicación con el Kernel. Los binarios con librerías vinculadas en tiempo de compilación son más portátiles desde un punto de vista de dependencias, pero menos versátiles, ya que pueden bloquearse en tiempo de ejecución si el núcleo es diferente de lo que se esperaba.

Entorno de análisis:

Un sandbox ideal sería el que fuera capaz de emular a la perfección el entorno bajo el que el malware se ejecutaría en función de sus características. Para configurar un entorno no solo es necesario utilizar la arquitectura correcta, bibliotecas y SO. También son importantes los privilegios bajo los cuales se ejecutan. Dentro de un entorno sandbox se ejecutan como usuario normal, dado que darle permisos de root podría ser más complejo de controlar. A menudo el malware de Linux se escribe dando por hecho que se tendrán permisos de root.

8. Estado del arte

El estado del arte respecto a la temática del trabajo, el análisis del malware compilado y preparado para dispositivos IoT es bastante extenso ya que existen numerosos trabajos realizados. Algunos poniendo énfasis en las dificultades y posibles soluciones a éstas, otros más dirigidos a la captación de malware desconocido mediante el uso de honeypots. A continuación, se detallarán resumidamente los trabajos relativos más relevantes para el contexto del trabajo.

Understanding Linux Malware [27]: En el estudio publicado, se realiza un trabajo de investigación sobre los ficheros binarios de Linux, en concreto, un análisis dinámico desde un sandbox. Estudios anteriores, tan sólo se centran en el comportamiento del malware a nivel de red. Se pone especial énfasis en la diversidad de dispositivos IoT, diferentes funcionalidades, diferentes CPUs, diferentes sistemas operativos, pero todas comparten que pueden ejecutar archivos ELF. Mediante el análisis de los archivos ELF, se puede detectar la manipulación realizada a través del malware. Toma en cuenta la capacidad del malware para ocultarse de los entornos sandbox y VMs. Además, realizan un análisis a un dataset bastante amplio del cual se pueden obtener conclusiones sobre la complejidad tan grande que supone realizar el estudio de este software.

Proyecto IoTPOT [29]: Este proyecto analiza las crecientes amenazas contra los dispositivos IoT utilizando para ello un sandbox y un honeypot. En la investigación se resaltan las características por las cuales los dispositivos IoT son atractivos para los atacantes, en general las medidas de seguridad adoptadas por los dispositivos son deficientes. Los dispositivos IoT en la mayoría de los casos son dispositivos que están mucho tiempo funcionando de forma continua, sin antivirus… Lo que los hace un objetivo ideal para los ciberdelincuentes. El estudio se ha llevado a cabo a través de la conocida como Darknet (parte de la red oculta al público) [30]. Este estudio ha permitido aprender mucho sobre el funcionamiento y propósito del malware. Para extender este estudio, se propone añadir soporte para otros protocolos utilizados en ataques, como el SSH.

SandPrint, fingerprinting malware sandboxes [31]: En este proyecto se pone en evidencia la necesidad de automatizar los análisis del malware dada la grandísima cantidad de ficheros potencialmente maliciosos no vistos antes, estimados en 65 millones en el año 2014 por Symantec.

Esta necesidad de automatizar los análisis hace que se hayan puesto públicos muchos servicios sandbox y cualquiera pueda adjuntar sus samples, lo que permite que se saque información identificativa del sandbox. Con esta idea desarrollan SandPrint. Un software basado en Windows que mide y filtra las características del sandbox, así como del sistema operativo, redes…

La conclusión del trabajo es que es realmente trivial evadir un sandbox basándose en sus características inherentes, lo cual certifica que evadir los sandbox preparando al malware para ello es posible.

9. Elección del sandbox

Dentro de las opciones disponibles en el mercado actualmente, para realizar el trabajo se ha optado por las de código abierto. Hay algunas soluciones sandbox proporcionadas por grandes empresas de ciberseguridad, como por ejemplo la de Kaspersk, o la de Joe Sandbox pero las soluciones que se pueden instalar para generar un sandbox en un entorno local de forma gratuita son bastante limitadas, en concreto solo se han encontrado 2, Limon y Cuckoo. En cuanto a características la mejor de ellas es Cuckoo. Se destacan las siguientes:

El código es totalmente abierto al público, hay documentación sobre las recomendaciones de programación y se puede aportar al proyecto, siempre que el código que se añada incluya nueva funcionalidad, y siga las recomendaciones de programación que los creadores han diseñado [17], por lo tanto, podríamos crear nuevos módulos, añadir funcionalidades, crear nuevos scripts para ejecutar extensiones no previstas por defecto, entre otros.

Soporta los más famosos softwares de virtualización, incluyendo QEMU que para el propósito de este trabajo es muy interesante, ya que permite simular diversas arquitecturas utilizadas en IoT.

Es posible instalar tanto sistemas operativos basados en Linux como en Windows. En la documentación oficial [17] se especifica los sistemas operativos en los que se ha testeado para las instalaciones guest, Windows XP 64-bit, Windows 7, Mac OS X Yosemite, Debian y Ubuntu. También existe Cuckoo droid para Android [32]. Por lo tanto, abarca los principales objetivos del malware actual según AV-Test [33].

Incluye una interfaz visual a través de web que permite enviar los archivos a analizar, recopilar los datos después de haber realizado el análisis, un historial de los análisis realizados, en resumen, permite gestionar toda la información disponible mediante esta interfaz.

Incluye varias herramientas que aportan más valor a los análisis, como la integración de la API de VirusTotal, herramientas para calcular los Hash de los archivos, Snort y otras que se enumerarán en el apartado de la instalación.

La instalación y preparación del sandbox es relativamente sencilla y la usabilidad del software es muy buena. En el repositorio de Git se incluye mucha información sobre incidencias que se han ido reportando, en las que se aporta información valiosa para los posibles problemas a encontrar.

10. Elección de la arquitectura de los ejecutables:

En apartados previos se ha dejado evidencia que los dispositivos IoT cotidianos más utilizados incorporan en algunas de sus versiones arquitecturas ARM. Por lo tanto, en la preparación del sandbox se tomará en cuenta esta arquitectura para que el entorno sea capaz de ejecutarlos, y la selección de los binarios de malware se tomará siguiendo esta línea.

Si se revisan binarios compilados por ejemplo para la botnet Mirai [34] se pueden ver que los hay compilados para diferentes arquitecturas como pueden ser mpsl, mips, arm7… En el caso de este trabajo también se tendrán en cuenta los preparados para MIPS.

3. Preparación del sandbox

Para la instalación de todo el software se ha preparado una máquina host con las siguientes características:

- 16GB de RAM
- Intel i7-5700 2-7 GHz
- Disco duro 140 GB
- SO: Ubuntu 18.04

La máquina virtualizada preparada para realizar los análisis tiene las siguientes características:

- 4GB de RAM
- 25GB de disco duro
- SO: Ubuntu 17.04 Kernel 4.0.19-16

1. Instalación del software:

Para la instalación del software de sandboxing seleccionado para realizar este trabajo Cuckoo, son necesarias las dependencias que se listan a continuación.

Python:

Python es el lenguaje de programación utilizado para el desarrollo de todo el software relacionado a la ejecución y análisis de Cuckoo. Por lo tanto, es un requisito indispensable para poder ejecutar su software. A continuación, listamos las dependencias y paquetes instalados relacionados con Python:

Nombre	Versión	Descripción
python	2.7.15	Lenguaje orientado a objetos de alto nivel.
python-pip	9.0.1	Instalador de paquetes Python.
python-dev	2.7.15	Biblioteca de Python.
python-virtualenv	15.1.0	Biblioteca de Python que permite crear un entorno virtual.
python-setuptools	39.0.1	Mejoras para Python distutils.
zlib1g-dev	1:1.2.11	Biblioteca de compresión y desarrollo.
libpq-dev	10.7-0	Biblioteca para PostgreSQL.

Figura 6. Librerías y dependencias de Python

MongoDB:

Base de datos NoSQL requerida para trabajar con Cuckoo. Esta base de datos es necesaria para la gestión de la interfaz web implementada, todo lo relevante a la gestión de las tareas, los análisis que se muestran a través del navegador, la información referente a los mismos…

Se ha instalado la versión 3.6.12.

PostgreSQL:

Base de datos SQL necesaria para toda la gestión interna de las tareas, análisis, máquinas virtuales… Es requisito indispensable tener una base de datos SQL, pero en Cuckoo dan a elegir, siendo tarea del usuario seleccionar y configurar una entre las disponibles como MySQL, MariaDB o PostreSQL.

Yara:

Es una herramienta dirigida a ayudar a los investigadores de malware a identificar y clasificar muestras de malware. Mediante una definición de una estructura, Yara permite clasificar archivos en función de su contenido. Yara es un complemento opcional para Cuckoo. En este caso se ha instalado la versión 3.7.1.

Software de virtualización:

Cuckoo Sandbox funciona con diferentes softwares de virtualización, en este caso se ha seleccionado el software VirtualBox en su versión 5.2.18.

TCPDump:

Es una herramienta capaz de volcar la actividad de red realizada por el malware durante su ejecución en la máquina virtual, capturado a su vez por un sniffer. La versión instalada es 4.9.2.

Se instalan las dependencias libcap2-bin 1:2.25-1.2 y apparmor-utils 2.12-4.

Volatility:

Volatility es una herramienta opcional que permite realizar un análisis forense en el proceso del volcado de memoria. Puede proporcionar, junto con Cuckoo una visibilidad adicional de las modificaciones realizadas en el sistema operativo y detectar la presencia de un rootkit en caso de que escape al dominio de supervisión del analizador de Cuckoo.

La versión instalada es 2.6.

M2Crypto:

Añade funcionalidad a Python incluyendo OpenSSL con RSA, DSA, DH, EC, HMAC, cifrados simétricos… Incluye funcionalidad SSL para clientes y servidores.

Es una dependencia instalar SWIG, en este caso con la versión 3.0.12 para utilizar este toolkit. La versión utilizada de M2Crypto es 0.27.0.

Cuckoo:

Una vez se han instalado todas las dependencias, se puede proceder a realizar la instalación del sistema Cuckoo. La instalación de Cuckoo se puede realizar sobre un entorno virtualizado o sobre el propio sistema host. Es necesario instalar el paquete setuptools para realizar la instalación de Cuckoo.

Las versiones instaladas son 2.0.6 para Cuckoo y 39.0.1 para setuptools.

2. Cuckoo Working Directory y configuración:

Una vez se ha procedido a instalar Cuckoo, el directorio principal en el que se realiza toda la configuración de Cuckoo, donde se guardan los análisis realizados, donde están almacenadas gran parte de las funcionalidades de Cuckoo se ponen en la carpeta raíz del usuario “Cuckoo”, con el nombre de “.cuckoo”.

En cuanto a los ficheros de configuración se señalan los modificados para este trabajo:

- Cuckoo.conf: Fichero más importante en el que se configura la maquinaria utilizada para realizar la virtualización (VirtualBox, VMWare…) conexiones de bases de datos… A continuación, se exponen las configuraciones realizadas:
  Machinery = VirtualBox
  [resultserver]
  Ip=192.168.56.1 Ip del host
  Port = 2042
- VirtualBox.conf: Fichero de configuración para la máquina virtual.
  Machines = ubuntu17
  [ubuntu17]
  Label = ubuntu17
  Platform = Linux
  Ip = 192.168.56.101
  Snapshot = Linux # Nombre del snapshot tomado en el momento de hacer el análisis.
- Reporting.conf: Fichero de configuración del sistema de report de los análisis.
  [mongodb]
  Enabled = yes
- Processing.conf: Permite configurar los módulos de procesamiento incluidos en Cuckoo.
  [VirusTotal]
  Enabled = yes #Envía el resultado a VirusTotal, permitiendo ver el resultado

3. Preparación del equipo invitado:

Para la configuración del equipo sobre el que se va a ejecutar el malware, se ha escogido utilizar Ubuntu 17.04 por la compatibilidad con el sistema de debugging del Kernel de Cuckoo. Se han encontrado algunos problemas durante la preparación del invitado con versiones más recientes y antiguas de Ubuntu.

A continuación, se enumeran los pasos necesarios para la preparación del sistema:

Instalación del agente en el sistema:

En la instalación de Cuckoo existe un fichero Python necesario para realizar la conexión entre el host y los diferentes clientes, el archivo con su nombre y ruta es “…/.cuckoo/agent/agent.py”. Es necesario instalar este archivo en el sistema

de arranque del sistema operativo, de manera que cuando se encienda se ejecute de forma automática.

Para esto se realiza la instalación de vsftpd (very secure transfer protocol, software de conexión FTP). Una vez instalado se traspasa el archivo y se procede a realizar la siguiente instrucción:

```
$ sudo crontab -e
```

Esta instrucción permite preparar la ejecución de una tarea. Se nos abrirá un editor de texto en el que incluimos la siguiente línea:

```
@reboot python /etc/init.d/agent.py
```

Este pequeño script se ejecutará en el momento del arranque del sistema. La ejecución se realizará mediante el comando Python que ejecutará el script que, en este caso hemos ubicado en "/etc/init.d/agent.py"

A continuación, se instalan dependencias necesarias para el funcionamiento de Cuckoo, utilizando la siguiente instrucción:

```
sudo apt-get install systemtap gcc patch Linux-headers-$(uname -r)
```

Con esta instrucción se instala:

Systemtap:

Utilidad del sistema operativo que permite inyectar instrucciones dinámicamente sin necesidad de modificar el código, o modificar estructuras críticas del Kernel.

GCC:

Compilador para Linux que soporta diversos lenguajes de programación.

Linux-headers:

Definen las interfaces de las funciones del Kernel de Linux.

A continuación, es necesario instalar los símbolos de debug del sistema operativo, pero, dada la discontinuidad de los repositorios de la versión 17.04 de Ubuntu, se ha procedido a descargar esta imagen directamente desde el navegador e instalando manualmente estos. En concreto los de la versión 4.0.19-16 que coinciden con la versión del Kernel utilizado por el sistema operativo Ubuntu.

En los repositorios de Ubuntu, se ha descargado el paquete “linux-generic-dbgsys.deb” y se ha instalado mediante la siguiente instrucción:

```
$ sudo dpkg -i /linux-generic-dbgsys.deb

$ sudo apt-get install -f
```

Una vez se ha realizado la instalación de los módulos del Kernel necesarios para la ejecución del análisis, se puede parchear el tapset (tapset se define como un script que encapsula información sobre un Kernel, incluyendo funciones que pueden ser utilizados por otros scripts. [35]) para que Cuckoo parsee correctamente la salida con las siguientes instrucciones.

```
$  wget  https://raw.GitHubusercontent.com/Cuckoosandbox/Cuckoo/master/stuff/
systemtap/expand_execve_envp.patch

$ wget https://raw.GitHubusercontent.com/Cuckoosandbox/Cuckoo/master/stuff
/systemtap/escape_delimiters.patch

$ sudo patch /usr/share/systemtap/tapset/Linux/sysc_execve.stp <
expand_execve_envp.patch

$ sudo patch /usr/share/systemtap/tapset/uconversions.stp <
escape_delimiters.patch
```

Compilamos la extensión del Kernel:

```
$ wget https://raw.GitHubusercontent.com/cuckoosandbox/cuckoo/
master/stuff/systemtap/strace.stp

$ sudo stap -p4 -r $(uname -r) strace.stp -m stap_ -v
```

Se realiza una comprobación del resultado de la compilación del Kernel:

```
$ sudo staprun -v ./stap_.ko
```

Si la salida es similar a la línea que se define a continuación:

```
staprun:insert_module:x Module stap_ inserted from file path_to_stap_.ko
```

Quiere decir que el proceso ha sido satisfactorio, y esto generará como resultado un fichero “stap.ko” que debe ser ubicado en “/root/.cuckoo”

A continuación, se realizan las siguientes operaciones para acabar de proporcionar el entorno adecuado la primera de todas será desactivar el firewall para permitir la conexión entre el host y el invitado:

```
$ sudo ufw disable
```

Desactivación de NTP (network time protocol):

```
$ sudo timedatectl set-ntp off
```

Se deshabilitan algunas configuraciones predeterminadas del sistema que no son necesarias:

```
$ sudo apt-get purge update-notifier update-manager update-manager-core ubuntu-release-upgrader-core

$ sudo apt-get purge whoopsie ntpdate cups-daemon avahi-autoipd avahi-daemon avahi-utils

$ sudo apt-get purge account-plugin-salut libnss-mdns telepathy-salut
```

Para la ejecución de software “MIPS” y “ARM” se instalan los siguientes paquetes dentro de la máquina virtual:

```
$ sudo apt get install qemu

$ sudo apt-get install gcc-arm-Linux-gnueabihf

$ sudo apt-get install qemu-user
```

En cuanto a la configuración de red, se modifican los datos del protocolo IPV4 de manera que las propiedades queden de la siguiente forma:

- IP: 192.168.56.101
- Mascara de red: 255.255.255.0
- Puerta de enlace predeterminada: 192.168.56.1

Para acabar de configurar la máquina, se configura a través de la configuración de VirtualBox una interfaz de red “Solo-anfitrión” que evita la conexión con internet, de esta manera se evita que el malware pueda realizar conexiones con otros hosts por motivos de seguridad.

En las herramientas globales de VirtualBox, se crea una nueva interfaz de red, con la máscara de red 192.168.56.0/16 y se le asigna únicamente está a la VM

recién creada. Por defecto el nombre de esta interfaz de red se deberá llamar vboxnet0, que ya está definida en los ficheros de configuración de Cuckoo.

4. Ejecución de Cuckoo:

Se levanta VirtualBox mediante línea de comandos con el usuario “Cuckoo”:

```
$ virtualbox
```

La aplicación VirtualBox la podemos iniciar mediante la interfaz gráfica, seleccionamos la máquina virtual ya preparada con Ubuntu y el software necesario instalado. Una vez encendida, seleccionamos “Herramientas” y dentro de las opciones clicamos en instantáneas. Seleccionamos generar instantánea y le damos el nombre que hemos configurado en Cuckoo, en este caso “Linux”. Una vez finalice dejamos la máquina arrancada.

A continuación, se puede levantar Cuckoo. Las instrucciones necesarias para levantar la aplicación Cuckoo, deben ejecutarse en consolas diferentes y permanecer arrancadas paralelamente para que todo el entorno funcione correctamente.

Primero ejecutamos la aplicación con el flag debug, lo que va a ser muy útil para detectar errores:

```
$ cuckoo -d
```

Mediante la consola se puede apreciar algo similar a lo siguiente en la primera ejecución:

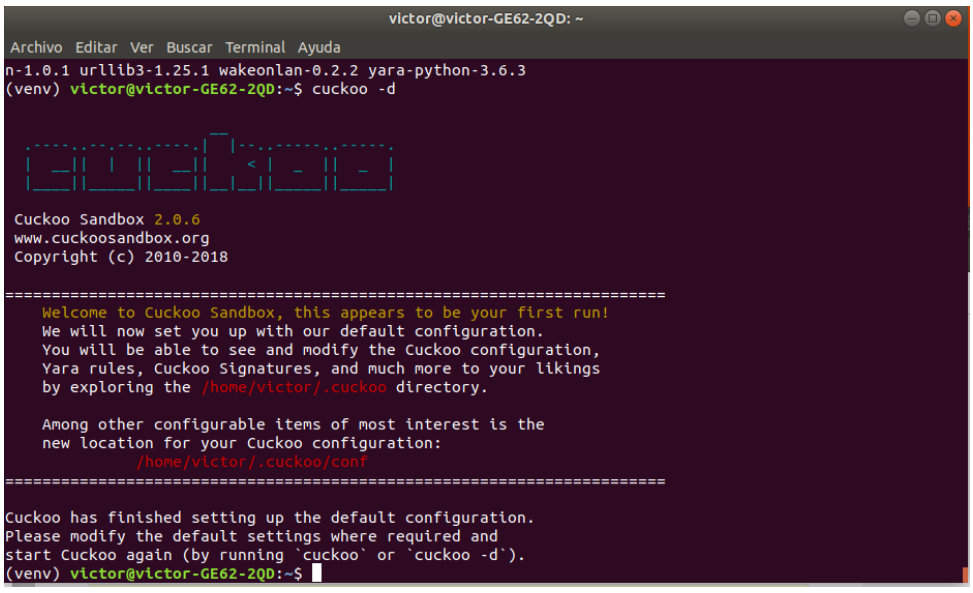


Figura 7. Resultado de la primera ejecución de Cuckoo

Es necesario volver a ejecutar de nuevo la instrucción, dado que la primera vez que se ejecuta la aplicación se procede a configurar inicialmente todo el working directory.

Se vuelve a ejecutar la misma instrucción y se podrá detectar como la máquina virtual que se tenía abierta en el sistema se minimiza dado que toma el control de ésta, Cuckoo.

Se abre un nuevo terminal para levantar el servidor web:

```
$ cuckoo web runserver
```

Una vez esté la aplicación web levantada se puede acceder a ella mediante la URL “localhost:8000”, la pantalla que se muestra será similar a la siguiente:

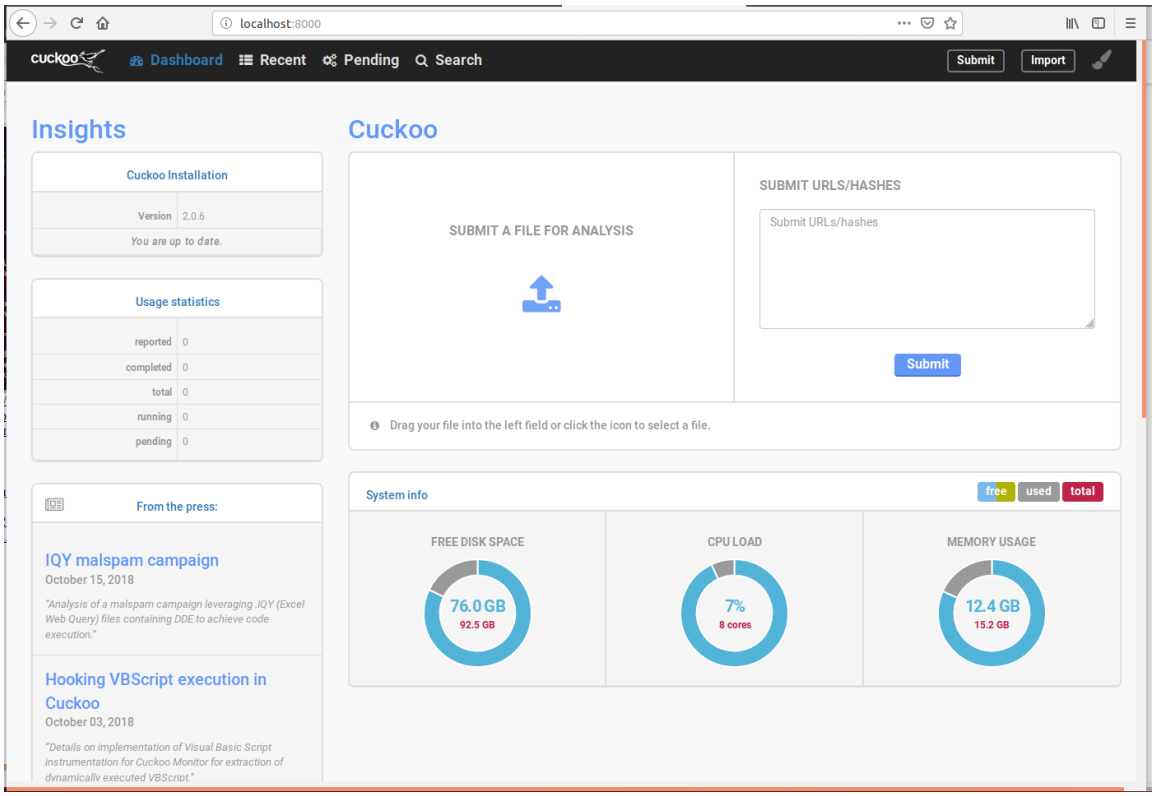


Figura 8. Interfaz web de Cuckoo

Hasta este punto, se tiene la máquina host con Cuckoo totalmente configurado, y la máquina invitada preparada para recibir y ejecutar las muestras de malware.

5. Se procesa la primera muestra de malware:

Una vez se ha accedido a la aplicación de Cuckoo, se puede apreciar la sencillez de la interfaz. Un “drag & drop” para arrastrar ficheros. Un menú superior que permite visualizar las operaciones pendientes y revisar análisis recientes.

Se procede a adjuntar el primer fichero para analizar en Cuckoo a través de la interfaz web. La visualización que obtenemos tras traspasar el fichero es la siguiente:

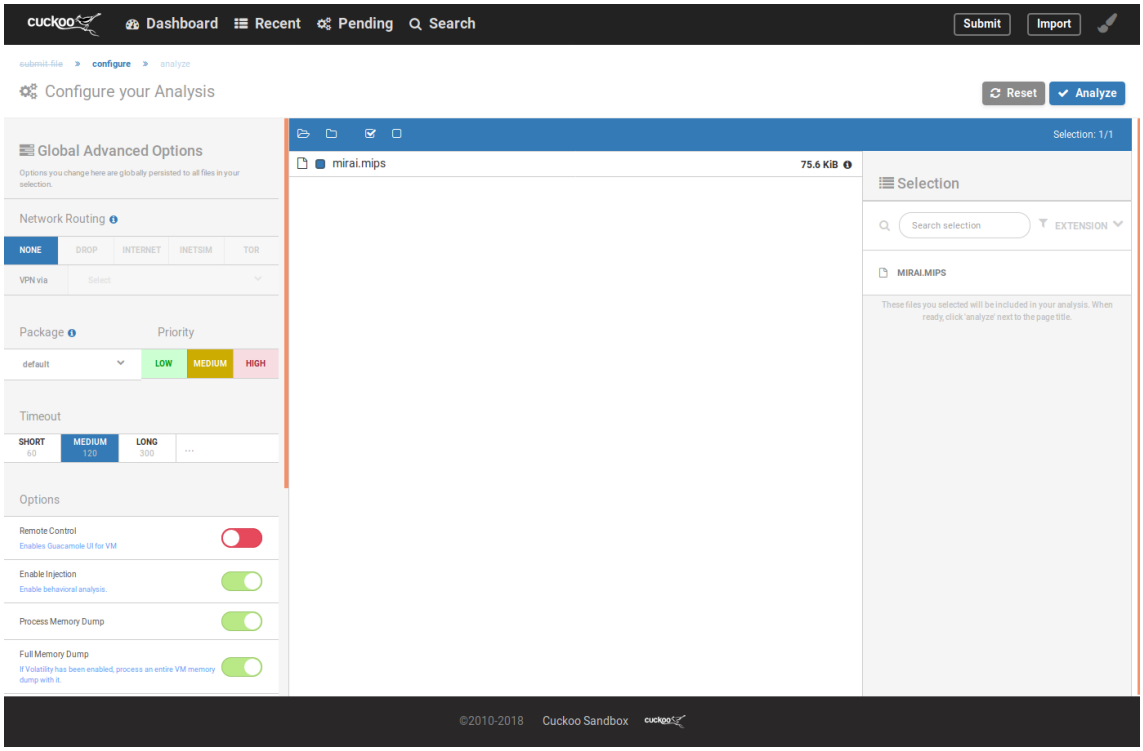


Figura 9. Envío de fichero a través de la interfaz web de Cuckoo

El fichero escogido para realizar el análisis se ha recogido de un repositorio de GitHub público [35] y es un ejecutable de la botnet mirai [36] que afectaba a dispositivos IoT con una baja seguridad en las credenciales de acceso a los mismos.

Para comenzar el análisis tan sólo hay que clicar en “Analyze” y en la consola en la que hemos ejecutado Cuckoo comenzarán a aparecer trazas que harán referencia a los plugins instalados, la conexión con la VM invitada y otros detalles que permiten seguir el proceso de la ejecución.

Una vez finalizado el análisis, la herramienta permite seleccionar el fichero recién analizado, lo cual nos muestra una pantalla de resumen del análisis recientemente realizado como la que se muestra a continuación:

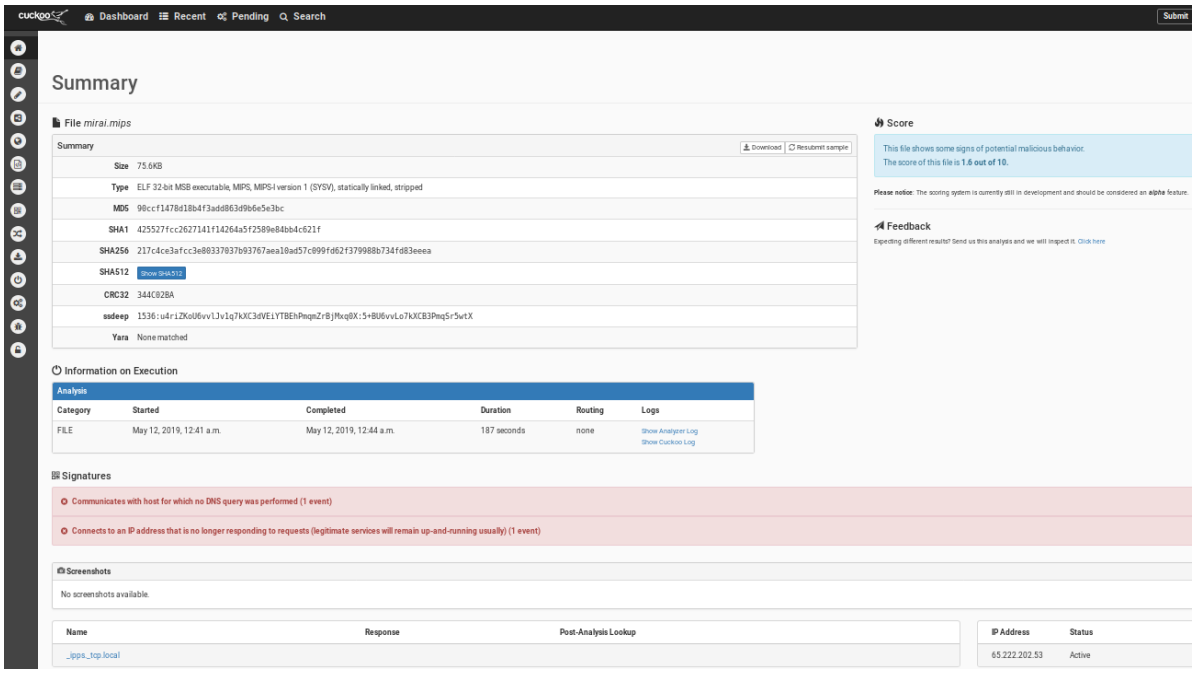


Figura 10. Obtención de resultados a través de la interfaz web de Cuckoo.

En la pantalla se puede apreciar datos sobre el propio archivo, como su tamaño, sus firmas y el scoring que asigna en cuanto al riesgo que conlleva calculado por Cuckoo.

En la parte izquierda se puede apreciar un menú desplegable que permite desplazarse entre los diferentes tipos de análisis que genera Cuckoo, de red, estático, de comportamiento, etc.

El primero de ellos es el estático. El análisis estático que realiza Cuckoo permite ver datos relacionados con la naturaleza del fichero. Arquitectura para la cual está compilado, si los datos son big o little endian, sistema operativo, y otros datos más técnicos. En este caso el tipo de fichero es un “ELF32”.

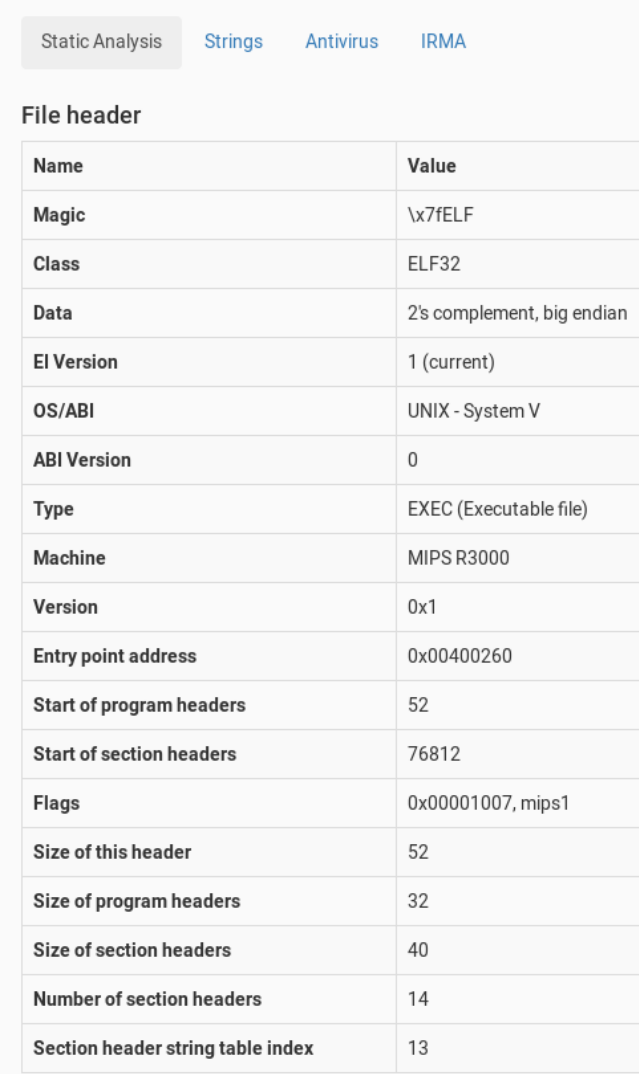


Figura 11. Análisis estático.

En la parte superior del análisis también se puede apreciar un apartado llamado “Strings” en el cual aparecen las cadenas de texto disponibles en el archivo.

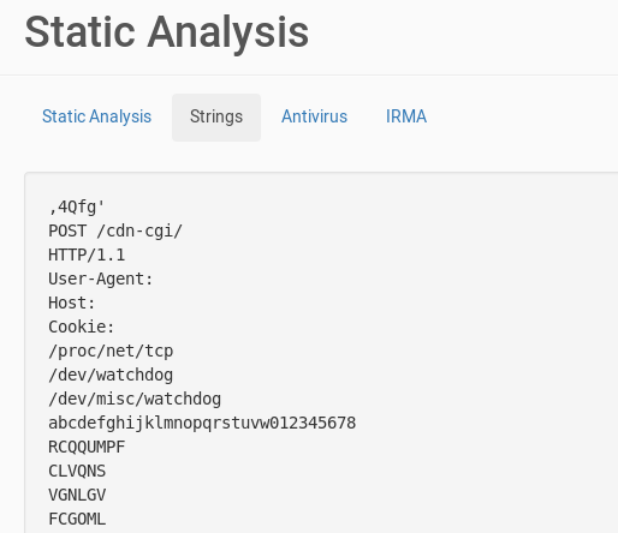
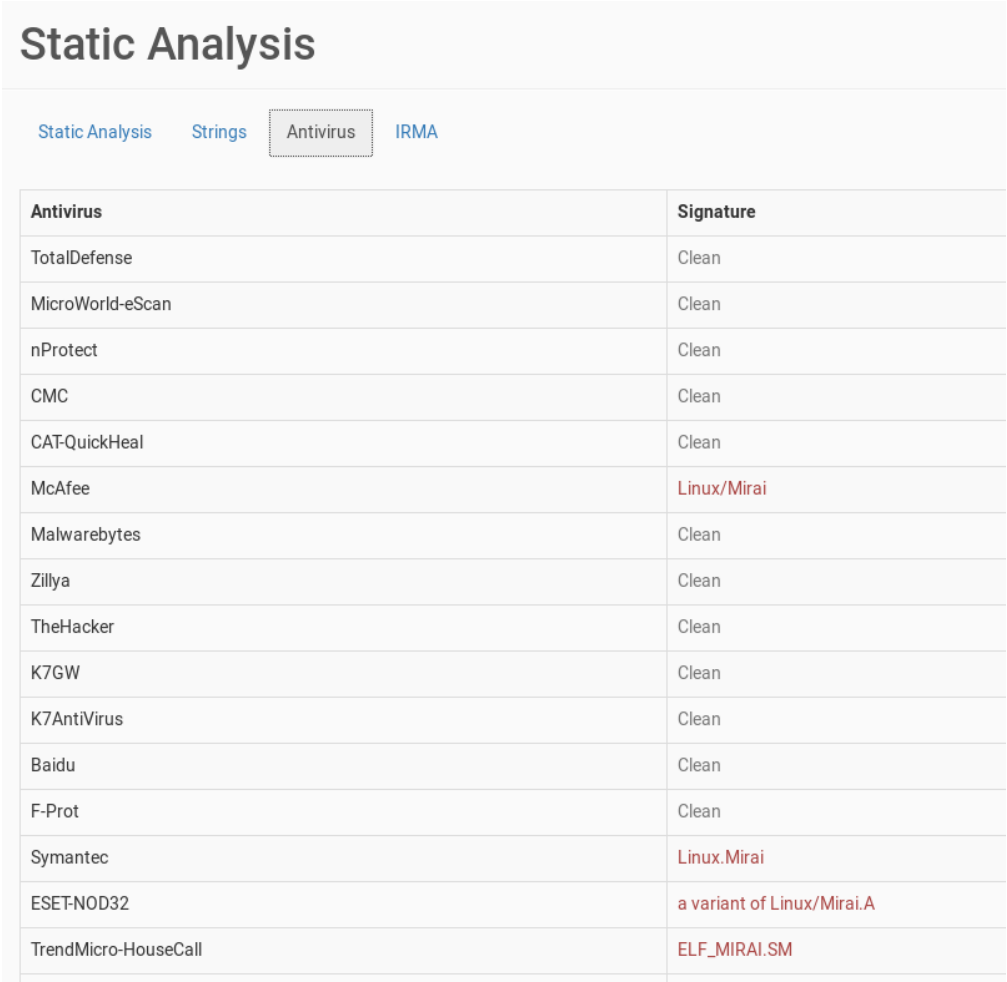


Figura 12. Análisis de cadenas de texto.

Si se ha activado la opción de VirusTotal, el siguiente apartado es muy interesante dado que Cuckoo envía utilizando su API pública el archivo enviado a analizar y permite visualizar el resultado de los análisis de los antivirus que dispone VirusTotal, algunos de ellos son los siguientes:



Antivirus	Signature
TotalDefense	Clean
MicroWorld-eScan	Clean
nProtect	Clean
CMC	Clean
CAT-QuickHeal	Clean
McAfee	Linux/Mirai
Malwarebytes	Clean
Zillya	Clean
TheHacker	Clean
K7GW	Clean
K7AntiVirus	Clean
Baidu	Clean
F-Prot	Clean
Symantec	Linux.Mirai
ESET-NOD32	a variant of Linux/Mirai.A
TrendMicro-HouseCall	ELF_MIRAI.SM

Figura 13. Análisis realizado por VirusTotal.

Como se puede comprobar, algunos de los antivirus detectan que el malware pertenece a la variante Linux Mirai.

El siguiente apartado que ha devuelto resultados de Cuckoo es el análisis de comportamiento, que muestra la lista de procesos ejecutados por el fichero. En este análisis se pueden visualizar hasta 6 páginas de llamadas a procesos del sistema.

Además, esta interfaz permite seleccionar las llamadas a sistema en función de su naturaleza. En la barra superior se puede apreciar en diversos colores la naturaleza de las llamadas, ya sean de sistema, de ficheros, de sincronización… Lo cual permite utilizar este filtro para buscar acciones en concreto.

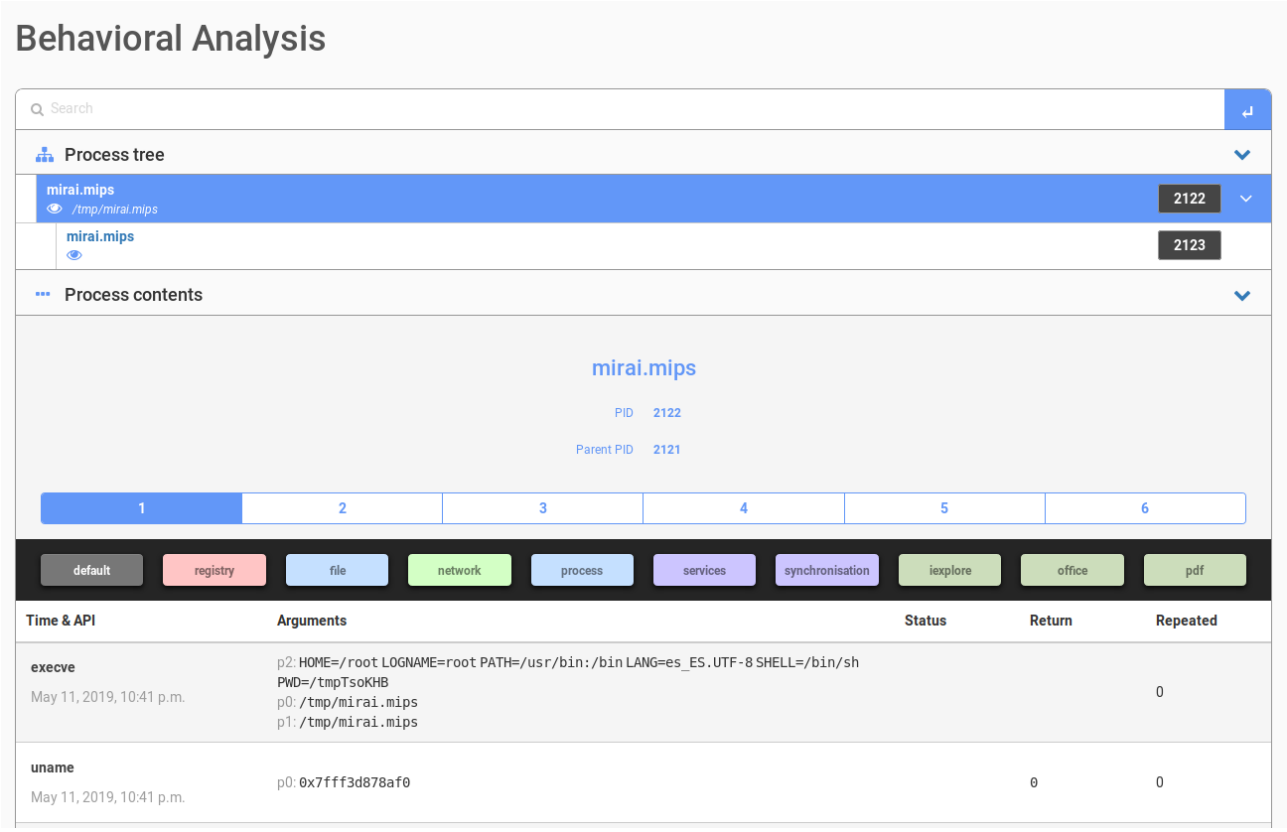


Figura 14. Análisis de comportamiento.

Cuckoo dispone de 6 páginas con todas las llamadas realizadas por el malware, en la siguiente figura se pueden visualizar algunas de ellas:

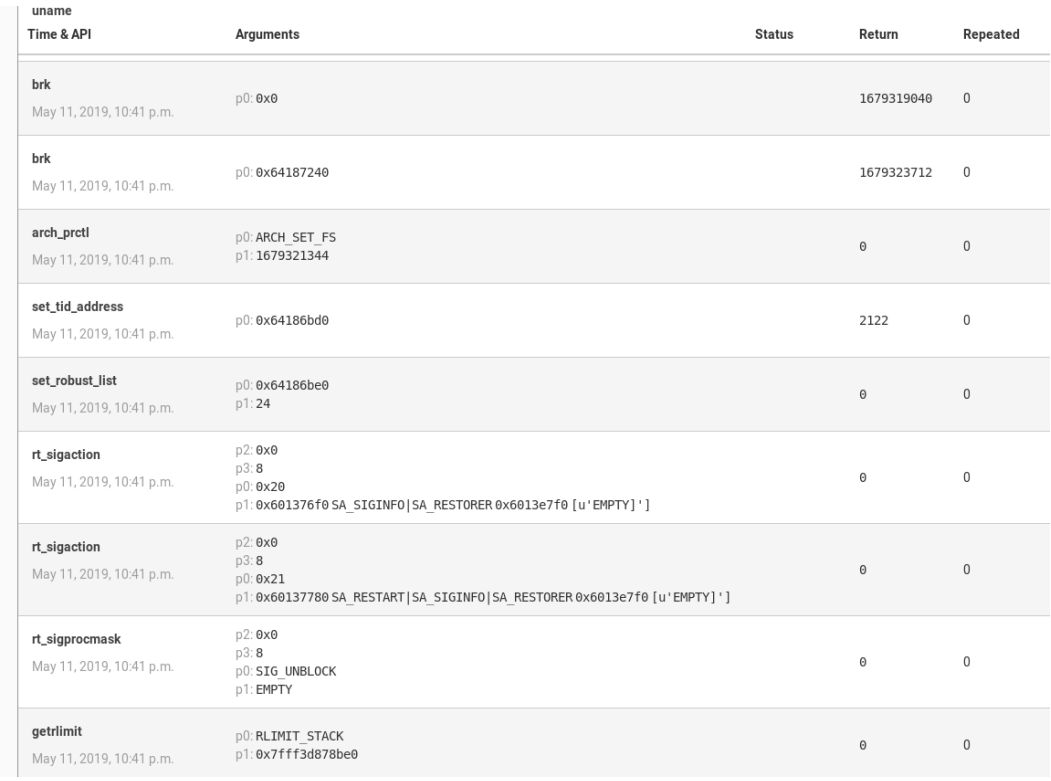
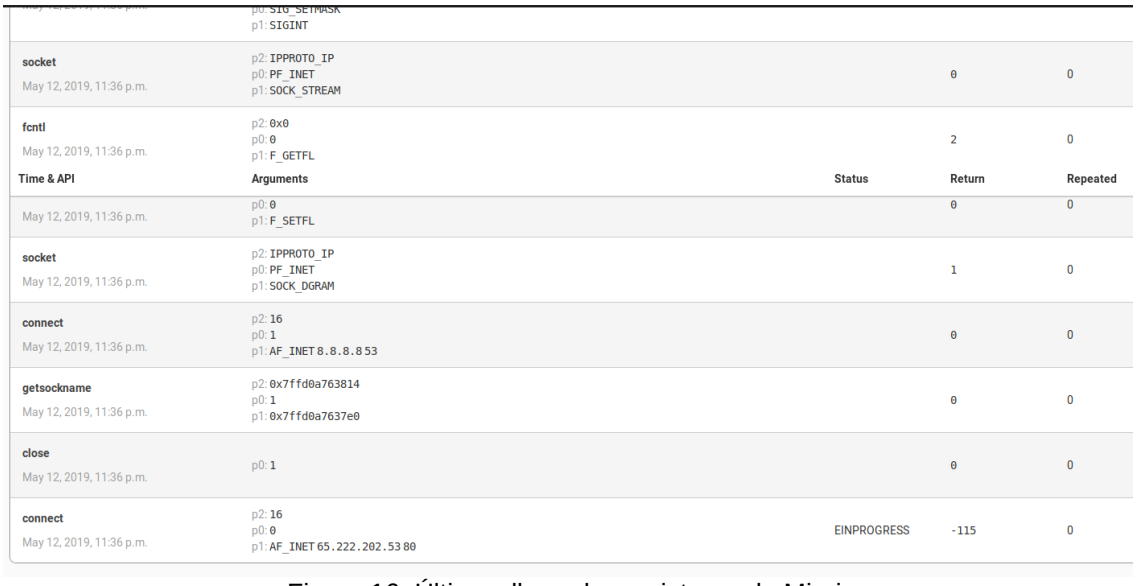


Figura 15. Ejemplo de las llamadas realizadas por el malware Mirai

Finalmente, para comprobar que la ejecución del archivo ha ido correctamente, la evidencia que interesa es la que conocemos del malware que se está analizando. Visualizando la última página del análisis como se muestra en la siguiente figura:



Time & API	Arguments	Status	Return	Repeated
	p1: SIGINT			
socket May 12, 2019, 11:36 p.m.	p2: IPPROTO_IP p0: PF_INET p1: SOCK_STREAM		0	0
fcntl May 12, 2019, 11:36 p.m.	p2: 0x0 p0: 0 p1: F_GETFL		2	0
May 12, 2019, 11:36 p.m.	p0: 0 p1: F_SETFL		0	0
socket May 12, 2019, 11:36 p.m.	p2: IPPROTO_IP p0: PF_INET p1: SOCK_DGRAM		1	0
connect May 12, 2019, 11:36 p.m.	p2: 16 p0: 1 p1: AF_INET 8.8.8.8 53		0	0
getsockname May 12, 2019, 11:36 p.m.	p2: 0x7ffd0a763814 p0: 1 p1: 0x7ffd0a7637e0		0	0
close May 12, 2019, 11:36 p.m.	p0: 1		0	0
connect May 12, 2019, 11:36 p.m.	p2: 16 p0: 0 p1: AF_INET 65.222.202.53 80	EINPROGRESS	-115	0

Figura 16. Últimas llamadas a sistema de Mirai.

El malware intenta realizar una conexión a una IP externa “65.222.202.53”. Debido a la configuración de la máquina virtual, esta conexión no va a ser exitosa. Según algunos blogs [37], esta IP pertenecía a la red Mirai. Con esta información parece que Cuckoo ha ejecutado el archivo de manera correcta. Durante la ejecución del análisis, Cuckoo ejecuta un sniffer que permite capturar el tráfico de red durante el período de tiempo en el que se ejecuta el malware, lo que resulta con la evidencia de las llamadas que se intentan realizar utilizando la red. Como se muestra en la siguiente figura, se ve la llamada a la IP vista en el análisis de comportamiento:

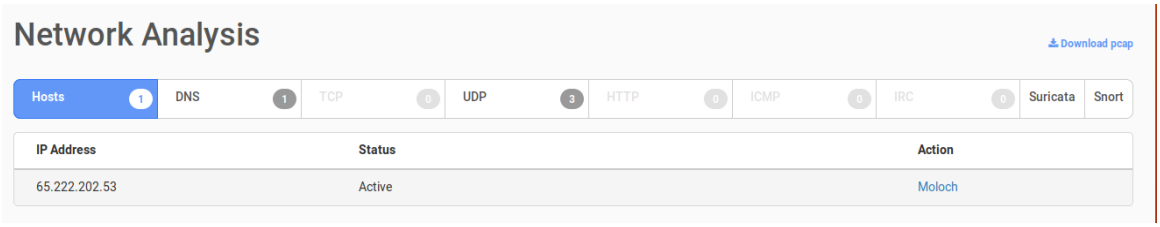


Figura 17. Análisis del tráfico de la red.

Se puede apreciar un intento de conexión al host 65.222.202.53. Cuckoo genera un archivo con extensión ”.pcap” con las capturas del tráfico generado a partir de la ejecución del malware. Es posible descargar este archivo y ver los payloads de las conexiones realizadas.

6. Análisis de la ejecución del binario mirai.mips

La muestra de malware ejecutada ha sido un binario de mirai. Este malware ha sido profundamente estudiado y se conocen con detalle las acciones que realiza. Dado que se dispone de un análisis de la ejecución dentro del Anexo C, procedemos a analizar paso a paso que se espera que haga este malware con los logs de la ejecución recogidos por Cuckoo. Para entender mejor las acciones realizadas por el malware, se resume brevemente su propósito.

Mirai es un malware que convierte los dispositivos conectados a la red con sistemas operativos basados en Linux en bots controlados de forma remota que se pueden utilizar como parte de una botnet con el propósito de realizar ataques a gran escala. La infección de este malware se dirige principalmente contra dispositivos conectados principalmente IoT, como cámaras IP y routers domésticos. El código fuente está publicado y se puede encontrar fácilmente en internet [38].

Según la documentación disponible en internet sobre el funcionamiento del malware Mirai [37], las acciones realizadas son las siguientes:

El malware intentará abrir “dev/whatchdog”, utilizando para ello distintos directorios:

```
open("/dev/watchdog", O_RDWR)
```

Posteriormente cambiará el work directory a la raíz:

```
chdir("/")
```

Utiliza el socket PF_INET para abrir el puerto UDP / 53 y acceder al servidor DNS de Google 8.8.8.8 y estableciendo la conexión.

```
connect(3, {sa_family=AF_INET, sin_port=htons(53),
sin_addr=inet_addr("8.8.8.8")}, 16)
```

El malware detectará la interfaz de salida y reutilizará el socket, abriendo una conexión TCP a través de un puerto aleatorio a la dirección IP local.

```
getsockname(3, {sa_family=AF_INET, sin_port=htons(39221),
sin_addr=inet_addr("YOUR-IP")}, [16])
close(3)
```

A continuación, el malware realiza decodificación de los strings contenidos en el binario. Después de esta decodificación el malware realiza una copia en otra ruta, y prepara el stderr y stdout para realizar una ejecución silenciosa.

El malware se autoelimina pero el proceso sigue corriendo.

Se genera un proceso de red que intenta buscar dispositivos visibles en la misma red para replicarse.

Mientras el proceso de conexión continua, el malware abre el puerto PF_INET para TCP y empieza a escuchar conexiones entrantes para 127.0.0.1:48101

```
socket(PF_INET, SOCK_STREAM, IPPROTO_IP)
bind(3, {sa_family=AF_INET, sin_port='htons(48101'),
sin_addr=inet_addr("127.0.0.1")}, 16)
listen(3, 1)
```

Mientras se intenta realizar conexiones contra otros dispositivos, se realiza la puerta trasera contra el host 65.222.202.53:

```
connect(0, {sa_family=AF_INET, sin_port=htons(80),
sin_addr=inet_addr("65.222.202.53")}, 16)
```

Todas estas acciones realizadas por el malware extraídas de un análisis del malware se pueden encontrar en el Anexo C. Se adjuntan las evidencias de las acciones recogidas del log a continuación:

```
mirai.mips@60050b2e[1739] openat(AT_FDCWD, "/dev/watchdog", O_RDWR) = -2
(ENOENT)

mirai.mips@60050b2e[1739] openat(AT_FDCWD, "/dev/misc/watchdog", O_RDWR) = -
2 (ENOENT)
mirai.mips@601aa527[1739] chdir("/") = 0

mirai.mips@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 6

mirai.mips@60050b2e[1739] connect(6, {AF_INET, 8.8.8.8, 53}, 16) = 0

bind(6, {AF_INET, 127.0.0.1, 48101}, 16) = 0

mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16)
= -115 (EINPROGRESS)
```

Se puede comprobar que la ejecución del malware ha sido exitosa y que las acciones realizadas en la máquina virtual preparada con Cuckoo son específicamente las del malware que se ha ejecutado.

4. Binarios tradicionales respecto a IoT y dificultades

En este proyecto se ha preparado un entorno virtual basado en Ubuntu, con el software necesario para poder ejecutar binarios compilados para arquitecturas como MIPS y ARM.

Para resaltar la diferencia entre la ejecución de un binario preparado para IoT y un binario tradicional, se ha procedido a compilar un programa escrito en C que tan sólo imprime “hello_world” por pantalla, se especifica el código fuente en la figura 18.

```
#include<stdio.h>
main(){
     printf("Hello world");
}
```

Figura 18. Código fuente usado en la generación de los binarios

Este primer fichero compilado estará preparado para arquitectura x86-x64, lo que podríamos determinar como un binario estándar de Linux. Para compilar este fichero se ha utilizado el código expuesto anteriormente, en el que se imprime un “hello_world” utilizando para ello la función “printf” estándar de C. En el Anexo A se incluyen los logs toda la ejecución de este binario.
Si analizamos el fichero con el comando “file” se puede identificar la siguiente información proveniente de los headers:

```
root@pc01:/home/test/ftp/test# file hello_input_here
hello_input_here: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynami
cally linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, Bui
ldID[sha1]=cb5e147a6398334487cb856f7b74a32389618f64, not stripped
```

Figura 19. Propiedades estáticas mediante comando file en binario Linux

Si se revisa toda la ejecución del Anexo A, al final de las evidencias se puede apreciar la siguiente instrucción en el log:

```
hello_input_her@7f336c4c0cf0[1717] write(1, "hello world\n", 12) = 12
```

El sistema requiere de bastantes llamadas a sistema para preparar toda la ejecución del fichero. Se sobreentiende que estas llamadas son necesarias para preparar las funciones, los recursos… para que finalmente se pueda obtener el resultado del programa.

Para que se pueda ejecutar un binario preparado para una arquitectura concreta es necesario que el loader especifico esté presente en el sistema. Lo que determina el loader que se va a asociar para realizar la ejecución viene determinada por los headers del fichero binario. Por lo tanto, si en el sistema operativo que se pretende ejecutar este fichero existe el loader necesario para ejecutar este fichero, va a ser posible la ejecución. Para disponer de los loaders

necesarios para ejecutar un fichero preparado para ARM se necesita tener un emulador, en este caso se ha utilizado Qemu. Qemu dispone de dos modos de funcionamiento [39], el primero de ellos permite ejecutar ficheros binarios preparados para una arquitectura concreta en otra (como en el ejemplo que se ha preparado del binario “hello world”, un binario preparado para ARM en un sistema x64). El segundo modo de funcionamiento permite simular todo un sistema operativo compilado para una arquitectura concreta instalado en otra arquitectura distinta (por ejemplo, un kernel compilado para ARM levantado sobre una máquina x86).

Por lo tanto, no existe una gran diferencia entre ejecutar entre una arquitectura y otra siempre que se tenga el software necesario debidamente instalado. Revisando las diferencias entre la ejecución del Anexo A y el Anexo B se aprecia que hay muchas más llamadas a sistema que parece que no tienen efecto en el resultado del final del programa (se entiende que son llamadas necesarias para la gestión de la memoria, de las instrucciones a ejecutar, que no son propiamente del ejecutable, si no de la preparación del mismo por el sistema operativo).

En la siguiente figura se puede examinar mediante el comando file que la arquitectura necesaria para ejecutar el binario es ARM. Se expone todo el resultado de la ejecución en el Anexo B:

Figura 20. Propiedades estáticas mediante comando file en binario ARM

Si se revisa la salida de los logs para el Anexo B el que ha sido preparado para ARM, casi al final de la ejecución se dispone de la misma salida, aunque con un set más grande de llamadas a sistema:

```
hello_arm_stati@60052d9e[1739] write(1, "hello world\n", 12) = 12
```

El resultado de ambos binarios ha sido el mismo, aunque con ligeras diferencias en la preparación y ejecución.

5. Conclusiones

La información recolectada en los primeros apartados a cerca de los dispositivos IoT revela que en el trabajo no se puede asumir el 100% de los dispositivos IoT existentes actualmente. Esto es debido a que existen numerosas arquitecturas distintas, para las cuales habría que preparar prácticamente un entorno con sus propias características para que cada una de ellas se pueda ejecutar. Esto significa que cada nueva muestra que obtengamos que requiera de otra configuración, se deberá revisar cómo aplicarla y instalar las dependencias, librerías, o software necesario para llevar a cabo la ejecución.

En este trabajo se ha demostrado que los dispositivos IoT son susceptibles a convertirse un objetivo cada vez más popular para recibir ataques por el software malicioso dado su actual crecimiento y la facilidad que se tiene para explotarlos debido a su baja configuración de seguridad. Asumiendo que los ataques más extendidos se realizan explotando vulnerabilidades conocidas o credenciales deficientes.

Los ataques más populares han sido destinados a crear botnets, con el objetivo de realizar ataques DDoS (Denegación de servicio), utilizando para ello la propagación del malware de forma muy similar a la que realiza un gusano.

La muestra de malware que se ha ejecutado durante la elaboración del trabajo evidencia este hecho, una vez infectado el dispositivo, infección realizada por ejecución de código mediante la explotación de una vulnerabilidad o por acceso no autorizado mediante credenciales débiles, éste comienza a realizar acciones para autorreplicarse y abrir una conexión contra un host remoto que tomará el control. Éste además deja un puerto en escucha para recibir peticiones entrantes.

En cuanto a la heterogeneidad de los dispositivos IoT se evidencia que es posible preparar un entorno transversal configurado para soportar diferentes clases de dispositivos y arquitecturas.

Si se quisiera realizar un análisis más realista o cercano al mundo real, se podría llegar a realizar una división de los dispositivos en dos grandes grupos.

Dispositivos que disponen unas limitaciones grandes en cuanto a hardware como para obligar a desarrollar un sandbox transversal que sea capaz de simular las arquitecturas que encontramos en entornos tiny, y a través de este realizar los análisis.

Por otro lado, se podrían contemplar dispositivos más potentes como la Raspberry, basados en Linux que a pesar de no ser igual de potentes que un ordenador personal, podrían llegar a ser capaces de ejecutar y levantar un sistema preparado con un Linux y un sandbox. Esto permitiría realizar pruebas en un entorno más similar al mundo real, por ejemplo, con otros dispositivos sandbox, permitiendo que haya conexión entre ellos y ver la replicación del malware.

Cómo se ha podido ver en apartados anteriores, la diversidad de los dispositivos es muy grande, lo cual obliga al analista a buscar y toda la información posible sobre distintas arquitecturas, librerías, dependencias que

se consuman desde el malware, y a aplicar soluciones software que simulen estas condiciones.

Como resultado de todo el trabajo se ha obtenido un entorno sandbox que permite realizar el análisis de malware de forma segura. Este entorno se ha preparado instalando en él un simulador como Qemu, que permite la ejecución de arquitecturas diversas como MIPS o ARM en un sistema Linux, esto capacita al sistema para que se ejecuten muestras compiladas específicamente para estas arquitecturas, utilizadas comúnmente en dispositivos IoT. En este caso el sistema que lo alberga es un Ubuntu x64, simulado por Virtualbox.

A pesar de haber desarrollado un entorno de sandbox que es capaz de ejecutar muestras de malware, este entorno no es más que un laboratorio que, en este trabajo se ha utilizado para analizar una muestra de malware de la tipología Mirai en forma de binario compilado y preparado para ser directamente ejecutado. Las muestras de malware que se pueden encontrar en un entorno real son distintas y es que para el análisis de malware será preciso ir adaptando poco a poco el entorno. Esto hará que a medida que se vayan ejecutando más muestras de binarios, se vaya preparando el entorno para diferentes arquitecturas con diferentes características, así aumentando la diversidad de arquitecturas compatibles. Respecto a esta reflexión, se debe tener en cuenta la evasión de malware hacía el software de sandboxing. La muestra utilizada, está lista para ser ejecutada y no contiene código que intente realizar una evasión o un fingerprint del sandbox, que sí que se podría encontrar en el caso de utilizar un honeypot, con el objetivo de captar nuevas muestras. Esto obligaría a aplicar algunas de las contramedidas que se han analizado y expuesto en esta memoria para evitarlo.

En cuanto a los resultados del trabajo son satisfactorios. Se ha obtenido un entorno estable y funcional para análisis de malware IoT. Este software es capaz de realizar una ejecución paso por paso de un fichero binario preparado para MIPS. El sandbox utilizado ejemplifica que, para ejecutar este tipo de ficheros tan sólo es necesario preparar el entorno de manera que sea capaz de ejecutar estos binarios como cualquier binario estándar, utilizando para ello un emulador que soporte estas arquitecturas. En este caso se ha utilizado QEMU para este fin, con las librerías que dan soporte a ARM y MIPS.

Siguiendo con los resultados, realizando una comparación del comportamiento que ha tenido el malware dentro del sandbox encaja con el publicado por otros analistas de malware en sus sitios web. Con lo que queda evidenciado el correcto funcionamiento del entorno.

Si se revisan los objetivos definidos al inicio del trabajo, se puede comprobar que, en gran parte se han conseguido todos. Se analizan individualmente:

Se ha podido seleccionar un subconjunto de dispositivos IoT para enfocar los análisis hacía estos. En concreto, como se ha podido ver en el trabajo se ha enfocado hacía dispositivos IoT con arquitectura MIPS y ARM.

En cuanto al sandbox, se ha realizado un análisis de las ofertas disponibles en el mercado actualmente, y se ha realizado una selección en base a medidas objetivas.

El punto más complejo de llevar a cabo sin duda ha sido respecto a las muestras de malware. El único malware disponible y documentado lo

suficientemente bien como para poder tener la certeza de que se iba a ejecutar un tipo de muestra como la que se esperaba, ha sido Mirai. Muchas dificultades para descargar samples en todos los portales que los almacenan. Denegando el acceso a cuentas sin privilegios y páginas con muchísimas muestras sin documentar.

Respecto a la evasión que realiza el malware, gracias al proyecto SandPrint se ha podido entender y comprender la verdadera dificultad que entraña esta tarea. Cada vez que el malware se adapte para evadir sandboxes, los sandboxes deberán adaptarse para evitarlo, por lo tanto, es una carrera por encontrar nuevos mecanismos, que decanten la balanza de un lado.

Implementar una evasión de malware en este proyecto no tiene demasiado sentido, dado que la muestra que se ha podido ejecutar no incluía código específico para realizar la evasión. Se podría haber preparado manualmente código que evitara la ejecución, comprobando algunos parámetros, como la carpeta raíz de Cuckoo en el guest, pero el valor está en la descripción de cómo se debe realizar.

Los análisis realizados por Cuckoo, tanto estático como dinámico están correctamente documentados y se puede entender cómo funcionan cada uno de ellos, para posteriormente extrapolarlo a la práctica realizada.

Finalmente, el proyecto ha quedado correctamente ejecutado dado que se dispone de un entorno correctamente configurado, que cumple el propósito establecido inicialmente y que es capaz de ejecutar todas las muestras configuradas para arquitecturas MIPS y ARM como se ha podido demostrar.

6. Trabajo futuro:

Para la continuación del trabajo, puede ser interesante proceder a preparar un honeypot con el que capturar malware y posteriormente analizarlo utilizando el sandbox preparado en este proyecto. Para ello, se podría utilizar algunos sistemas operativos más concretos, si se quieren encontrar algunas muestras específicas de malware.

Sobre el entorno preparado se podría realizar un aumento del número de muestras y arquitecturas disponibles para analizar, que sean comunes en dispositivos IoT, asegurando la compatibilidad del entorno y asegurando la fiabilidad de los resultados en cada uno de ellos. Ésto podría permitir incluso implementar un clasificador, para extraer información a partir de analizar una gran cantidad de muestras de distintas tipologías.

Para poder extender la capacidad de análisis del sandbox, se podría realizar la instalación de múltiples instancias con múltiples sistemas operativos como Linux, Mac, Windows; preparar un dominio, un servidor y abrir al público el analizador. Este proyecto sería muy interesante para recolectar muestras de malware, incluso se podrían recolectar samples que igual no se hubieran encontrado antes. Los desarrolladores de malware pueden realizar pruebas de evasión de sandboxes utilizando las menos populares, ya que por ejemplo, VirusTotal propicia que el malware se catalogue.

Poniendo el foco en la evasión de malware, se podrían poner en práctica muchas de las técnicas vistas en SandPrint utilizando el código fuente de algún malware que disponga de su código debidamente publicado.

Dada la manera en que está desarrollado Cuckoo de forma pública, podría ser interesante analizar qué módulo podría aportar valor y realizar su desarrollo. Como ejemplo, se podría preparar un modo “honeypot” que permita analizar malware sin tener que subir manualmente las muestras, sino que sea un evento del propio sistema el que empiece el análisis, por ejemplo, mediante la monitorización de un nuevo proceso creado fuera de los previstos por el sistema.

Debido a la dificultad para encontrar muestras a través de Internet, para cualquier trabajo futuro, debería tenerse en cuenta la posibilidad de solicitar muestras a VirusTotal mediante una cuenta con privilegios u otros portales que disponen de muestras de malware para usuarios autorizados.

6. Glosario

Término	Definición
IoT	El Internet de las cosas, se refiere a la interconexión digital de objetos cotidianos con Internet.
Malware	Programa malicioso que trata de afectar a un ordenador o dispositivo.
Sandbox	Se conoce como sandbox un entorno aislado, que permite ejecutar programas con seguridad, y de manera aislada.
Virtual Machine	Una máquina virtual es una emulación de un sistema operativo.
Guest	Se conoce como guest un sistema operativo instalado en una máquina virtual
Host	En la virtualización de hardware, se entiende como una maquina donde se ejecuta una máquina virtual.
MIPS	Microprocessor without Interlocked Pipelined Stages por sus siglas en inglés. Es una arquitectura con un conjunto reducido de instrucciones, utilizado generalmente para dispositivos IoT.
ARM	Advanced RISC Machine por sus siglas en inglés. Es una familia de arquitecturas con un conjunto de un set reducido de instrucciones. Los procesadores con arquitectura RISC tienen menos transistores que los utilizados en PCs.
Fingerprint	En computación se entiende como fingerprint una recolección de configuraciones de software y hardware de un dispositivo que permite diferenciarlo de otros.
Standalone	Aplicaciones que pueden funcionar como procesos independientes.
Kernel	Programa informático que es el núcleo del sistema operativo, tiene el control de todo lo relacionado con el sistema.
Root	Se entiende como root una cuenta de usuario utilizada para la administración de un sistema operativo. Esta cuenta dispone de los máximos privilegios que permiten realizar cambios sin restricción
Big/Little endian	Endianness es el orden secuencial de una lista en objetos arbitrarios. El formato big-endian coloca el bit más significativo se almacena primero, tiene la dirección más baja. El formato Little-endian es al revés, el bit menos significativo primero, en la dirección más baja.
Sniffer	Programa capaz de interceptar y logar el tráfico que viaja a través de una red.
Payload	Payload en las telecomunicaciones es la parte de los datos que se transmiten en una conexión. El mensaje que se envía.

Headers	En cuanto al sistema operativo Linux, el paquete headers define una interfaz, sobre como las funciones están definidas en el código fuente. De esta manera un compilador puede comprobar si el uso de una función es correcto. En cuanto a un fichero o estructura de datos, se entiende como header datos suplementarios situados al inicio del bloque, aportando información sobre los datos que vienen a continuación.
Loader	Un loader es una parte del sistema operativo responsable de cargar programas y librerías. Es un paso esencial en el inicio de la ejecución de un programa.
Rootkit	Un rootkit es una colección de software, generalmente malicioso, diseñado para permitir acceder a un ordenador para el que no tiene permiso, por ejemplo a un usuario no autorizado.
Honeypot	Mecanismo de seguridad para detectar, proteger o evadir posibles intentos de realizar accesos ilícitos. Un honeypot consiste, por ejemplo en un sitio web, que asemeja ser legítimo pero se encuentra aíslado y monitorizado de manera que se puedan realizar soluciones de seguridad con el aprendizaje que los datos se almacenan en él pueden dar.

7. Bibliografía

[1] OBS Big Data 2015
https://www.obs-edu.com/es/noticias/estudio-obs/en-2020-mas-de-30-mil-millones-de-dispositivos-estaran-conectados-internet

[2] Wikipedia IoT
https://es.wikipedia.org/wiki/Internet_de_las_cosas

[3] Seguridad en Internet of Things
https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security

[4] Securing the Internet of Things
https://www.nics.uma.es/sites/default/files/papers/1633.pdf

[5] Malware
https://en.wikipedia.org/wiki/Malware

[6] Kaspersky: IoT Malware grew three-fold in 2018
https://www.kaspersky.com/about/press-releases/2018_new-iot-malware-grew-three-fold-in-h1-2018

[7] Sandbox
https://en.wikipedia.org/wiki/Sandbox_(computer_security)

[8] Sandbox – Evading Malware: Techniques, principles, and Examples
https://www.apriorit.com/dev-blog/545-sandbox-evading-malware

[9] Universidad de Alcará: Ventajas y desventajas del uso de IoT
https://www.master-internet-of-things.com/ventajas-desventajas-del-uso-iot/

[10] Nubit Consulting: Ventajas e inconvenientes de un mundo conectado IoT
https://www.nubit.es/las-ventajas-e-inconvenientes-de-un-mundo-conectado/

[11] Skaldion: Ventajas y desventajas del IoT
http://skaldion.com/2018/01/ventajas-y-desventajas-del-iot/

[12] Ovacen: Definición, ventajas y desventajas
https://ovacen.com/internet-de-las-cosas/

[13] Wikipedia: Dyn cyberattack
https://en.wikipedia.org/wiki/2016_Dyn_cyberattack

[14] Padmini Gaur, Mohit P. Tahiliani: Operating System for IoT Devices
https://ieeexplore.ieee.org/abstract/document/7166231

[15] Wikipedia Kernel
https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)

[16] Kiran Jot Singh, Divneet Singh Kapoor: Survey IoT Platforms
https://ieeexplore.ieee.org/abstract/document/7879392

[17] Stichting Cuckoo Foundation: Cuckoo sandbox
https://cuckoosandbox.org/

[18] Trusted Sec: Malware analysys cuckoo
https://www.trustedsec.com/2018/05/malware-cuckoo-1/

[19] Joe Security Sandbox
https://www.joesecurity.org/joe-sandbox-linux

[20] Limon sandbox
https://GitHub.com/monnappa22/Limon

[21] Kaspersky Sandbox Lab
https://www.kaspersky.com/enterprise-security/wiki-section/products/sandbox

[21] Lenny Zeltser: Free Malware Sample Sources for Researchers
https://zeltser.com/malware-sample-sources/

[22] URLhaus abuse.ch project for sharing malicious URLs used for malware distribution
https://urlhaus.abuse.ch/

[23] The MalShare (TM) Project: Free Malware repository
https://malshare.com/search.php?query=.arm

[24] shashank Jain: Malware Basic Static Analysis
https://medium.com/@jain.sm/malware-basic-static-analysis-cf19b4600725

[25] Wikipedia análisis de malware
https://en.wikipedia.org/wiki/Malware_analysis

[26] Olivier Ferrand (2015) How to detect the Cuckoo Sandbox and to Strengthen it?
https://www.researchgate.net/publication/271448810_How_to_detect_the_Cuckoo_Sandbox_and_to_Strengthen_it

[27] Emanuele Cozzi, Mariano Graziano, Yanick Fratantonio, Davide Balzarotti 2018: Understanding Linux Malware
https://rud.is/dl/ieee-sp-2018/435301a870.pdf

[28] Antonakakis et al., “Understanding the Mirai Botnet,” in Proceedings of the USENIX Security Symposium, 2017.
https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf

[29] IoTPOT: Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, Christian Rossow Analysing the Rise of IoT Compromises
https://www.usenix.org/system/files/conference/woot15/woot15-paper-pa.pdf

[30] WikiPedia Darknet
https://en.wikipedia.org/wiki/Darknet

[31] Akira Yokoyama, Kou Ishii, Rui Tanabe, Yinmin Papa, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, Daisuke Inoue, Michael Brengel, Michael Backes, Christian Rossow SandPrint: Fingerprinting Malware Sandboxes to Provide Intelligence for Sandbox Evasion
https://christian-rossow.de/publications/sandprint-raid2016.pdf

[32] Cuckoo Droid: Cuckoo for Android malware analysis
https://GitHub.com/idanr1986/cuckoo-droid

[33] The independent IT-Security Institute 2016-2017: AV-Test
https://www.av-test.org/fileadmin/pdf/security_report/AV-TEST_Security_Report_2016-2017.pdf

[34] Wolfvan malware samples
https://GitHub.com/wolfvan/some-samples

[35] TapSet reference
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html-single/systemtap_tapset_reference/index

[36] Wikipedia BotNet Mirai
https://en.wikipedia.org/wiki/Mirai_(malware)

[37] Malware Must Die: Linux/Mirai, how an old ELF malcode is recycled
http://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html

[38] Anna Senpai Mirai-Source-Code
https://GitHub.com/jgamblin/Mirai-Source-Code

[39] tuxthink Executing arm executable in x86 using qemu
http://tuxthink.blogspot.com/2012/04/executing-arm-executable-in-x86-using.html

8. Anexos

Anexo A:

Ejecución archivo en C que imprime “hello_world” binario.

```
python@7fc534ebdb33[1716] set_robust_list(0x7fc5352dc9e0, 24) = 0
python@7fc534ec8730[1716] close(8) = 0
python@7fc534bbdb67[1716]         execve("/usr/bin/sh",         ["sh",         "-c",
"/tmp/hello_input_here"],   ["HOME=/root",   "LOGNAME=root",   "PATH=/usr/bin:/bin",
"LANG=es_ES.UTF-8", "SHELL=/bin/sh", "PWD=/root"]) = -2 (ENOENT)
sh@7f37e2a297c9[1716]   execve("/bin/sh",   ["sh",   "-c",   "/tmp/hello_input_here"],
["HOME=/root",     "LOGNAME=root",     "PATH=/usr/bin:/bin",     "LANG=es_ES.UTF-8",
"SHELL=/bin/sh", "PWD=/root"])
sh@7f37e2a297c9[1716] brk(0x0) = 94395718176768
sh@7f37e2a2a677[1716] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
sh@7f37e2a2a76a[1716]         mmap2(0x0,         12288,         PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f37e2c31000
sh@7f37e2a2a677[1716] access(0x7f37e2a31020, R_OK) = -2 (ENOENT)
sh@7f37e2a2a617[1716] open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 6
sh@7f37e2a2a5a2[1716] fstat(6, 0x7ffe01cfca50) = 0
sh@7f37e2a2a76a[1716]   mmap2(0x0,   95094,   PROT_READ,   MAP_PRIVATE,   6,   0)   =
0x7f37e2c19000
sh@7f37e2a2a717[1716] close(6) = 0
sh@7f37e2a2a677[1716] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
sh@7f37e2a2a617[1716] open("/lib/x86_64-Linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) =
6
sh@7f37e2a2a637[1716] read(6, 0x7ffe01cfcbf8, 832) = 832
sh@7f37e2a2a5a2[1716] fstat(6, 0x7ffe01cfca90) = 0
sh@7f37e2a2a76a[1716]         mmap2(0x0,         3959200,         PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_DENYWRITE, 6, 0) = 0x7f37e2648000
sh@7f37e2a2a807[1716] mprotect(0x7f37e2805000, 2097152, PROT_NONE) = 0
sh@7f37e2a2a76a[1716]     mmap2(0x7f37e2a05000,     24576,     PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 1822720) = 0x7f37e2a05000
sh@7f37e2a2a76a[1716]     mmap2(0x7f37e2a0b000,     14752,     PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f37e2a0b000
sh@7f37e2a2a717[1716] close(6) = 0
sh@7f37e2a2a76a[1716]         mmap2(0x0,         8192,         PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f37e2c17000
sh@7f37e2a0fc42[1716] arch_prctl(ARCH_SET_FS, 139878004258560) = 0
sh@7f37e2a2a807[1716] mprotect(0x7f37e2a05000, 16384, PROT_READ) = 0
sh@7f37e2a2a807[1716] mprotect(0x55da35847000, 8192, PROT_READ) = 0
sh@7f37e2a2a807[1716] mprotect(0x7f37e2c34000, 4096, PROT_READ) = 0
sh@7f37e2a2a7e7[1716] munmap(0x7f37e2c19000, 95094) = 0
sh@7f37e27155a7[1716] getuid() = 0
sh@7f37e27155c7[1716] getgid() = 0
sh@7f37e271557f[1716] getpid() = 1716
sh@7f37e267d8ee[1716]     rt_sigaction(SIGCHLD,     {0x55da3563b5a0,     SA_RESTORER,
0x7f37e267d7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|
SIGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGU
RG|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8)
= 0
sh@7f37e27155b7[1716] geteuid() = 0
sh@7f37e2745859[1716] brk(0x0) = 94395718176768
sh@7f37e2745859[1716] brk(0x55da37422000) = 94395718311936
sh@7f37e2715597[1716] getppid() = 1709
sh@7f37e273f615[1716] stat("/root", 0x7ffe01cfd120) = 0
sh@7f37e273f615[1716] stat(".", 0x7ffe01cfd1b0) = 0
sh@7f37e274058a[1716] getcwd(0x55da37401150, 4096) = 11
sh@7f37e27155b7[1716] geteuid() = 0
sh@7f37e27155d7[1716] getegid() = 0
sh@7f37e267d8ee[1716] rt_sigaction(SIGINT, {NULL}, 0x7ffe01cfd090, 8) = 0
sh@7f37e267d8ee[1716]     rt_sigaction(SIGINT,     {0x55da3563b5a0,     SA_RESTORER,
```

```
0x7f37e267d7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|
SIGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGU
RG|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8)
= 0
sh@7f37e267d8ee[1716] rt_sigaction(SIGQUIT, {NULL}, 0x7ffe01cfd090, 8) = 0
sh@7f37e267d8ee[1716] rt_sigaction(SIGQUIT, {SIG_DFL}, 0x0, 8) = 0
sh@7f37e267d8ee[1716] rt_sigaction(SIGTERM, {NULL}, 0x7ffe01cfd0a0, 8) = 0
sh@7f37e267d8ee[1716] rt_sigaction(SIGTERM, {SIG_DFL}, 0x0, 8) = 0
sh@7f37e271480a[1716]  clone(CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,  0x0,
0x0, 0x7f37e2c179d0) = 1717
hello_input_her@7f336c7aa7c9[1717]                  execve("/tmp/hello_input_here",
["/tmp/hello_input_here"],  ["HOME=/root",  "LOGNAME=root",  "PATH=/usr/bin:/bin",
"LANG=es_ES.UTF-8", "SHELL=/bin/sh", "PWD=/tmpkA8DBP"])
hello_input_her@7f336c7aa7c9[1717] brk(0x0) = 94839244300288
hello_input_her@7f336c7ab677[1717] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
hello_input_her@7f336c7ab76a[1717]    mmap2(0x0,    12288,    PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f336c9b2000
hello_input_her@7f336c7ab677[1717] access(0x7f336c7b2020, R_OK) = -2 (ENOENT)
hello_input_her@7f336c7ab617[1717] open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 6
hello_input_her@7f336c7ab5a2[1717] fstat(6, 0x7ffc513ae5e0) = 0
hello_input_her@7f336c7ab76a[1717] mmap2(0x0, 95094, PROT_READ, MAP_PRIVATE, 6, 0) =
0x7f336c99a000
hello_input_her@7f336c7ab717[1717] close(6) = 0
hello_input_her@7f336c7ab677[1717] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
hello_input_her@7f336c7ab617[1717]           open("/lib/x86_64-Linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC) = 6
hello_input_her@7f336c7ab637[1717] read(6, 0x7ffc513ae788, 832) = 832
hello_input_her@7f336c7ab5a2[1717] fstat(6, 0x7ffc513ae620) = 0
hello_input_her@7f336c7ab76a[1717]    mmap2(0x0,    3959200,    PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_DENYWRITE, 6, 0) = 0x7f336c3c9000
hello_input_her@7f336c7ab807[1717] mprotect(0x7f336c586000, 2097152, PROT_NONE) = 0
hello_input_her@7f336c7ab76a[1717]           mmap2(0x7f336c786000,           24576,
PROT_READ|PROT_WRITE,    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    6,    1822720)    =
0x7f336c786000
hello_input_her@7f336c7ab76a[1717]           mmap2(0x7f336c78c000,           14752,
PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f336c78c000
hello_input_her@7f336c7ab717[1717] close(6) = 0
hello_input_her@7f336c7ab76a[1717]    mmap2(0x0,    8192,    PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f336c998000
hello_input_her@7f336c790c42[1717] arch_prctl(ARCH_SET_FS, 139858842060544) = 0
hello_input_her@7f336c7ab807[1717] mprotect(0x7f336c786000, 16384, PROT_READ) = 0
hello_input_her@7f336c7ab807[1717] mprotect(0x56417b37b000, 4096, PROT_READ) = 0
hello_input_her@7f336c7ab807[1717] mprotect(0x7f336c9b5000, 4096, PROT_READ) = 0
hello_input_her@7f336c7ab7e7[1717] munmap(0x7f336c99a000, 95094) = 0
hello_input_her@7f336c4c0662[1717] fstat(1, 0x7ffc513aed80) = 0
hello_input_her@7f336c4c6859[1717] brk(0x0) = 94839244300288
hello_input_her@7f336c4c6859[1717] brk(0x56417b79c000) = 94839244439552
hello_input_her@7f336c4c0cf0[1717] write(1, "hello world\n", 12) = 12
hello_input_her@7f336c495b38[1717] exit_group(0)
sh@7f37e271450a[1716] wait4(-1, 0x7ffe01cfcf5c, 0x0, 0x0) = 1717
sh@7f37e271450a[1716] rt_sigreturn() = 1717
sh@7f37e2714b38[1716] exit_group(0)
```

Anexo B:

Ejecución archivo en C que imprime "hello_world" en ARM.

```
python@7fb09a88fb33[1738] set_robust_list(0x7fb09acae9e0, 24) = 0
python@7fb09a89a730[1738] close(8) = 0
python@7fb09a58fb67[1738]          execve("/usr/bin/sh",          ["sh",          "-c",
"/tmp/hello_arm_static"],    ["HOME=/root",    "LOGNAME=root",    "PATH=/usr/bin:/bin",
"LANG=es_ES.UTF-8", "SHELL=/bin/sh", "PWD=/root"]) = -2 (ENOENT)
sh@7f8706b577c9[1738]   execve("/bin/sh",   ["sh",   "-c",   "/tmp/hello_arm_static"],
["HOME=/root",      "LOGNAME=root",      "PATH=/usr/bin:/bin",      "LANG=es_ES.UTF-8",
"SHELL=/bin/sh", "PWD=/root"])
sh@7f8706b577c9[1738] brk(0x0) = 94516184469504
sh@7f8706b58677[1738] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
sh@7f8706b5876a[1738]          mmap2(0x0,          12288,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8706d5f000
sh@7f8706b58677[1738] access("/etc/ld.so.preload", R_OK) = -2 (ENOENT)
sh@7f8706b58617[1738] open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 6
sh@7f8706b585a2[1738] fstat(6, 0x7fff4b2267d0) = 0
sh@7f8706b5876a[1738]   mmap2(0x0,   97292,   PROT_READ,   MAP_PRIVATE,   6,   0)   =
0x7f8706d47000
sh@7f8706b58717[1738] close(6) = 0
sh@7f8706b58677[1738] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
sh@7f8706b58617[1738] open("/lib/x86_64-Linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 6
sh@7f8706b58637[1738] read(6, 0x7fff4b226978, 832) = 832
sh@7f8706b585a2[1738] fstat(6, 0x7fff4b226810) = 0
sh@7f8706b5876a[1738]          mmap2(0x0,          3959200,          PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_DENYWRITE, 6, 0) = 0x7f8706776000
sh@7f8706b58807[1738] mprotect(0x7f8706933000, 2097152, PROT_NONE) = 0
sh@7f8706b5876a[1738]      mmap2(0x7f8706b33000,      24576,      PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 1822720) = 0x7f8706b33000
sh@7f8706b5876a[1738]      mmap2(0x7f8706b39000,      14752,      PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8706b39000
sh@7f8706b58717[1738] close(6) = 0
sh@7f8706b5876a[1738]          mmap2(0x0,          8192,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8706d45000
sh@7f8706b3dc42[1738] arch_prctl(ARCH_SET_FS, 140217911891712) = 0
sh@7f8706b58807[1738] mprotect(0x7f8706b33000, 16384, PROT_READ) = 0
sh@7f8706b58807[1738] mprotect(0x55f642405000, 8192, PROT_READ) = 0
sh@7f8706b58807[1738] mprotect(0x7f8706d62000, 4096, PROT_READ) = 0
sh@7f8706b587e7[1738] munmap(0x7f8706d47000, 97292) = 0
sh@7f87068435a7[1738] getuid() = 0
sh@7f87068435c7[1738] getgid() = 0
sh@7f870684357f[1738] getpid() = 1738
sh@7f87067ab8ee[1738]      rt_sigaction(SIGCHLD,      {0x55f6421f95a0,      SA_RESTORER,
0x7f87067ab7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
sh@7f87068435b7[1738] geteuid() = 0
sh@7f8706873859[1738] brk(0x0) = 94516184469504
sh@7f8706873859[1738] brk(0x55f6439be000) = 94516184604672
sh@7f8706843597[1738] getppid() = 1731
sh@7f870686d615[1738] stat("/root", 0x7fff4b226ea0) = 0
sh@7f870686d615[1738] stat(".", 0x7fff4b226f30) = 0
sh@7f870686e58a[1738] getcwd(0x55f64399d150, 4096) = 11
sh@7f87068435b7[1738] geteuid() = 0
sh@7f87068435d7[1738] getegid() = 0
sh@7f87067ab8ee[1738] rt_sigaction(SIGINT, {NULL}, 0x7fff4b226e10, 8) = 0
sh@7f87067ab8ee[1738]      rt_sigaction(SIGINT,      {0x55f6421f95a0,      SA_RESTORER,
0x7f87067ab7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
```

```
sh@7f87067ab8ee[1738] rt_sigaction(SIGQUIT, {NULL}, 0x7fff4b226e10, 8) = 0
sh@7f87067ab8ee[1738] rt_sigaction(SIGQUIT, {SIG_DFL}, 0x0, 8) = 0
sh@7f87067ab8ee[1738] rt_sigaction(SIGTERM, {NULL}, 0x7fff4b226e20, 8) = 0
sh@7f87067ab8ee[1738] rt_sigaction(SIGTERM, {SIG_DFL}, 0x0, 8) = 0
sh@7f870684280a[1738]   clone(CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,   0x0,
0x0, 0x7f8706d459d0) = 1739
hello_arm_stati@601ba7c7[1739]                        execve("/tmp/hello_arm_static",
["/tmp/hello_arm_static"],   ["HOME=/root",   "LOGNAME=root",   "PATH=/usr/bin:/bin",
"LANG=es_ES.UTF-8", "SHELL=/bin/sh", "PWD=/tmp3_u1bm"])
hello_arm_stati@601ba7c7[1739] uname(0x7fffe61e6130) = 0
hello_arm_stati@601e6a39[1739] brk(0x0) = 1678884864
hello_arm_stati@601e6a39[1739] brk(0x6411d240) = 1678889536
hello_arm_stati@6015187a[1739] arch_prctl(ARCH_SET_FS, 1678887168) = 0
hello_arm_stati@60148b4e[1739] set_tid_address(0x6411cbd0) = 1739
hello_arm_stati@60148baf[1739] set_robust_list(0x6411cbe0, 24) = 0
hello_arm_stati@6014fb4d[1739]             rt_sigaction(0x20,             {0x60148950,
SA_SIGINFO|SA_RESTORER, 0x6014fa50, [EMPTY]}, 0x0, 8) = 0
hello_arm_stati@6014fb4d[1739]             rt_sigaction(0x21,             {0x601489e0,
SA_RESTART|SA_SIGINFO|SA_RESTORER, 0x6014fa50, [EMPTY]}, 0x0, 8) = 0
hello_arm_stati@60148c86[1739] rt_sigprocmask(SIG_UNBLOCK, [EMPTY], 0x0, 8) = 0
hello_arm_stati@601bcae7[1739] getrlimit(RLIMIT_STACK, 0x7fffe61e6220) = 0
hello_arm_stati@601f17ef[1739] readlink(0x6023c056, 0x7fffe61e5260, 4096) = 24
hello_arm_stati@601e6a39[1739] brk(0x6413e240) = 1679024704
hello_arm_stati@601e6a39[1739] brk(0x6413f000) = 1679028224
hello_arm_stati@601bba67[1739] access(0x602aba1e, F_OK) = -2 (ENOENT)
hello_arm_stati@6014e42f[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS],
0x7fffe61e61c0, 8) = 0
hello_arm_stati@601bd35a[1739]     mmap2(0x0,     8392704,     PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe69a22e000
hello_arm_stati@601bd407[1739] mprotect(0x7fe69a22e000, 4096, PROT_NONE) = 0
hello_arm_stati@601bff19[1739]
clone(CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SE
TTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,    0x7fe69aa2ddf0,    0x7fe69aa2e9d0,
0x7fe69aa2e9d0) = 1740
hello_arm_stati@6014e42f[1739] rt_sigprocmask(SIG_SETMASK, [EMPTY], 0x0, 8) = 0
hello_arm_stati@6014a404[1740] set_robust_list(0x7fe69aa2e9e0, 24) = 0
hello_arm_stati@601b3bc7[1739] gettimeofday(0x7fffe61e6170, 0x0) = 0
hello_arm_stati@6014f691[1740] nanosleep([0.010000000], 0x7fe69aa2dd00) = 0
hello_arm_stati@601bcae7[1739] getrlimit(RLIMIT_STACK, 0x7fffe61e5db0) = 0
hello_arm_stati@601b3bb7[1739] time(0x0) = 1558797921
hello_arm_stati@601b9a90[1739]                               open("/etc/qemu-binfmt/arm",
O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = -2 (ENOENT)
hello_arm_stati@601ba7c7[1739] uname(0x7fffe61e5a20) = 0
hello_arm_stati@601e6a39[1739] brk(0x64160000) = 1679163392
hello_arm_stati@601bd35a[1739]     mmap2(0x0,     528384,     PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe69a1ad000
hello_arm_stati@601bd407[1739]             mprotect(0x6056b000,             33550336,
PROT_READ|PROT_WRITE|PROT_EXEC) = 0
hello_arm_stati@601bd407[1739] mprotect(0x6256a000, 4096, PROT_NONE) = 0
hello_arm_stati@601bd427[1739]  madvise(0x6056be10,  33546736,  MADV_HUGEPAGE)  =  -22
(EINVAL)
hello_arm_stati@601bd35a[1739]     mmap2(0x0,     23261184,     PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe698b7e000
hello_arm_stati@601e6a39[1739] brk(0x64181000) = 1679298560
hello_arm_stati@601e6a39[1739] brk(0x64180000) = 1679294464
hello_arm_stati@601bd35a[1739]        mmap2(0x0,        4143972352,        PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0) = 0x7fe5a1b7e000
hello_arm_stati@601bd35a[1739]   mmap2(0x7fe6a1b6e000,   4096,   PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe6a1b6e000
hello_arm_stati@601bd407[1739] mprotect(0x7fe6a1b6e000, 4096, PROT_READ) = 0
hello_arm_stati@6014f6f1[1739] open("/proc/sys/vm/mmap_min_addr", O_RDONLY) = 7
hello_arm_stati@601bb772[1739] fstat(7, 0x7fffe61e54a0) = 0
hello_arm_stati@6014f181[1739] read(7, 0x641648f0, 1024) = 6
```

```
hello_arm_stati@60177c8b[1739] close(7) = 0
hello_arm_stati@601bd199[1739] gettid() = 1739
hello_arm_stati@601bb772[1739] fstat(6, 0x7fffe61e5c90) = 0
hello_arm_stati@601baca7[1739] geteuid() = 0
hello_arm_stati@601bacc7[1739] getegid() = 0
hello_arm_stati@6014f181[1739] read(6, 0x7fffe61e5e50, 1024) = 1024
hello_arm_stati@601bd35a[1739]  mmap2(0x7fe5a1b8e000,  319488,  PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED, 6, 0) = 0x7fe5a1b8e000
hello_arm_stati@601bd35a[1739]  mmap2(0x7fe5a1beb000,  8192,  PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED, 6, 315392) = 0x7fe5a1beb000
hello_arm_stati@601bd35a[1739]  mmap2(0x7fe5a1bed000,  4096,  PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe5a1bed000
hello_arm_stati@6014f1e1[1739] close(6) = 0
hello_arm_stati@601bd35a[1739]  mmap2(0x7fe69837d000,  8392704,  PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe69837d000
hello_arm_stati@601bd407[1739] mprotect(0x7fe69837d000, 4096, PROT_NONE) = 0
hello_arm_stati@601bac97[1739] getuid() = 0
hello_arm_stati@601baca7[1739] geteuid() = 0
hello_arm_stati@601bacb7[1739] getgid() = 0
hello_arm_stati@601bacc7[1739] getegid() = 0
hello_arm_stati@6016b4c0[1739] rt_sigprocmask(SIG_BLOCK, NULL, 0x64166918, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGHUP, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGHUP,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGINT, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGINT,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGQUIT, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]          rt_sigaction(SIGQUIT,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGILL, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGILL,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGTRAP, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]          rt_sigaction(SIGTRAP,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGABRT, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]          rt_sigaction(SIGABRT,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGBUS, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGBUS,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
```

```
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGFPE, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGFPE,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGKILL, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGKILL,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGUSR1, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGUSR1,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGSEGV, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGSEGV,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGUSR2, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGUSR2,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGPIPE, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGPIPE,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGALRM, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGALRM,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGTERM, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGTERM,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x10, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x10,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
```

```
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGCHLD, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGCONT, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGSTOP, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGTSTP, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGTTIN, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGTTOU, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGURG, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGXCPU, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGXCPU,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGXFSZ, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGXFSZ,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGVTALRM, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]          rt_sigaction(SIGVTALRM,          {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGPROF, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGPROF,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGWINCH, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGIO/SIGPOLL, {NULL}, 0x7fffe61e5b40, 8)
= 0
hello_arm_stati@6014fb4d[1739]        rt_sigaction(SIGIO/SIGPOLL,        {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGPWR, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGPWR,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(SIGSYS, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]           rt_sigaction(SIGSYS,           {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x40, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x40,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                  0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
```

```
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x22, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x22,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x23, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x23,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x24, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x24,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x25, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x25,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x26, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x26,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x27, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x27,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x28, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x28,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x29, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x29,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x2a, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x2a,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x2b, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x2b,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
```

```
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x2c, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x2c,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x2d, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x2d,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x2e, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x2e,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x2f, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x2f,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x30, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x30,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x31, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x31,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x32, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x32,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x33, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x33,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x34, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x34,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
```

```
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x35, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x35,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x36, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x36,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x37, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x37,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x38, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x38,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x39, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x39,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x3a, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x3a,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x3b, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x3b,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x3c, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x3c,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x3d, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x3d,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                 0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x3e, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x3e,            {0x6004d1c0,
```

```
SA_SIGINFO|SA_RESTORER,                                                    0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@6014fb4d[1739] rt_sigaction(0x3f, {NULL}, 0x7fffe61e5b40, 8) = 0
hello_arm_stati@6014fb4d[1739]            rt_sigaction(0x3f,            {0x6004d1c0,
SA_SIGINFO|SA_RESTORER,                                                    0x6014fa50,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
hello_arm_stati@601c0077[1739] arch_prctl(ARCH_SET_GS, 140624237420544) = 0
hello_arm_stati@601e6a39[1739] brk(0x641a1000) = 1679429632
hello_arm_stati@601bd35a[1739]  mmap2(0x7fe5a1bee000,  4096,  PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe5a1bee000
hello_arm_stati@601ba7c7[1739] uname(0x7fffe61e4620) = 0
hello_arm_stati@601bb7c5[1739] lstat("/tmp", 0x7fffe61e4700) = 0
hello_arm_stati@601bb7c5[1739] lstat("/tmp/hello_arm_static", 0x7fffe61e4700) = 0
hello_arm_stati@601bd35a[1739]  mmap2(0x7fe5a1bef000,  135168,  PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe5a1bef000
hello_arm_stati@601bba67[1739] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
hello_arm_stati@601bb772[1739] fstat(1, 0x7fffe61e4b00) = 0
hello_arm_stati@60052d9e[1739] write(1, "hello world\n", 12) = 12
hello_arm_stati@601bd199[1739] exit_group(0)
hello_arm_stati@601bd199[1740] futex(0x62583184, FUTEX_WAIT, -1, NULL) = -512
hello_arm_stati@601bd199[1740] futex(0x7fe69aa2e9d0, FUTEX_WAKE, 1) = 0
sh@7f870684250a[1738] wait4(-1, 0x7fff4b226cdc, 0x0, 0x0) = 1739
sh@7f870684250a[1738] rt_sigreturn() = 1739
sh@7f8706842b38[1738] exit_group(0)
```

Anexo C:

Ejecución malware Mirai en MIPS.

```
python@7f7c41806b33[1738] set_robust_list(0x7f7c41c259e0, 24) = 0
python@7f7c41811730[1738] close(8) = 0
python@7f7c41506b67[1738]  execve("/usr/bin/sh",  ["sh",  "-c",  "/tmp/mirai.mips"],
["HOME=/root",     "LOGNAME=root",     "PATH=/usr/bin:/bin",     "LANG=es_ES.UTF-8",
"SHELL=/bin/sh", "PWD=/root"]) = -2 (ENOENT)
sh@7fa0b8a997c9[1738]    execve("/bin/sh",    ["sh",    "-c",    "/tmp/mirai.mips"],
["HOME=/root",     "LOGNAME=root",     "PATH=/usr/bin:/bin",     "LANG=es_ES.UTF-8",
"SHELL=/bin/sh", "PWD=/root"])
sh@7fa0b8a997c9[1738] brk(0x0) = 94038651879424
sh@7fa0b8a9a677[1738] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
sh@7fa0b8a9a76a[1738]         mmap2(0x0,         12288,         PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa0b8ca1000
sh@7fa0b8a9a677[1738] access(0x7fa0b8aa1020, R_OK) = -2 (ENOENT)
sh@7fa0b8a9a617[1738] open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 6
sh@7fa0b8a9a5a2[1738] fstat(6, 0x7ffe11bdbad0) = 0
sh@7fa0b8a9a76a[1738]   mmap2(0x0,   97292,   PROT_READ,   MAP_PRIVATE,   6,   0)   =
0x7fa0b8c89000
sh@7fa0b8a9a717[1738] close(6) = 0
sh@7fa0b8a9a677[1738] access("/etc/ld.so.nohwcap", F_OK) = -2 (ENOENT)
sh@7fa0b8a9a617[1738] open("/lib/x86_64-Linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 6
sh@7fa0b8a9a637[1738] read(6, 0x7ffe11bdbc78, 832) = 832
sh@7fa0b8a9a5a2[1738] fstat(6, 0x7ffe11bdbb10) = 0
sh@7fa0b8a9a76a[1738]         mmap2(0x0,         3959200,         PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_DENYWRITE, 6, 0) = 0x7fa0b86b8000
sh@7fa0b8a9a807[1738] mprotect(0x7fa0b8875000, 2097152, PROT_NONE) = 0
sh@7fa0b8a9a76a[1738]      mmap2(0x7fa0b8a75000,      24576,      PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 1822720) = 0x7fa0b8a75000
sh@7fa0b8a9a76a[1738]      mmap2(0x7fa0b8a7b000,      14752,      PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fa0b8a7b000
sh@7fa0b8a9a717[1738] close(6) = 0
sh@7fa0b8a9a76a[1738]         mmap2(0x0,         8192,         PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa0b8c87000
sh@7fa0b8a7fc42[1738] arch_prctl(ARCH_SET_FS, 140328271640320) = 0
sh@7fa0b8a9a807[1738] mprotect(0x7fa0b8a75000, 16384, PROT_READ) = 0
sh@7fa0b8a9a807[1738] mprotect(0x55871384e000, 8192, PROT_READ) = 0
sh@7fa0b8a9a807[1738] mprotect(0x7fa0b8ca4000, 4096, PROT_READ) = 0
sh@7fa0b8a9a7e7[1738] munmap(0x7fa0b8c89000, 97292) = 0
sh@7fa0b87855a7[1738] getuid() = 0
sh@7fa0b87855c7[1738] getgid() = 0
sh@7fa0b878557f[1738] getpid() = 1738
sh@7fa0b86ed8ee[1738]     rt_sigaction(SIGCHLD,     {0x5587136425a0,     SA_RESTORER,
0x7fa0b86ed7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
sh@7fa0b87855b7[1738] geteuid() = 0
sh@7fa0b87b5859[1738] brk(0x0) = 94038651879424
sh@7fa0b87b5859[1738] brk(0x55871472d000) = 94038652014592
sh@7fa0b8785597[1738] getppid() = 1731
sh@7fa0b87af615[1738] stat("/root", 0x7ffe11bdc1a0) = 0
sh@7fa0b87af615[1738] stat(".", 0x7ffe11bdc230) = 0
sh@7fa0b87b058a[1738] getcwd(0x55871470c150, 4096) = 11
sh@7fa0b87855b7[1738] geteuid() = 0
sh@7fa0b87855d7[1738] getegid() = 0
sh@7fa0b86ed8ee[1738] rt_sigaction(SIGINT, {NULL}, 0x7ffe11bdc110, 8) = 0
sh@7fa0b86ed8ee[1738]     rt_sigaction(SIGINT,     {0x5587136425a0,     SA_RESTORER,
0x7fa0b86ed7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
```

```
sh@7fa0b86ed8ee[1738] rt_sigaction(SIGQUIT, {NULL}, 0x7ffe11bdc110, 8) = 0
sh@7fa0b86ed8ee[1738] rt_sigaction(SIGQUIT, {SIG_DFL}, 0x0, 8) = 0
sh@7fa0b86ed8ee[1738] rt_sigaction(SIGTERM, {NULL}, 0x7ffe11bdc120, 8) = 0
sh@7fa0b86ed8ee[1738] rt_sigaction(SIGTERM, {SIG_DFL}, 0x0, 8) = 0
sh@7fa0b878480a[1738]   clone(CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,   0x0,
0x0, 0x7fa0b8c879d0) = 1739
mirai.mips@601a8fb7[1739]       execve("/tmp/mirai.mips",       ["/tmp/mirai.mips"],
["HOME=/root",     "LOGNAME=root",     "PATH=/usr/bin:/bin",     "LANG=es_ES.UTF-8",
"SHELL=/bin/sh", "PWD=/tmpOEDOEE"])
mirai.mips@601a8fb7[1739] uname(0x7fff618dd7f0) = 0
mirai.mips@601d5099[1739] brk(0x0) = 1678016512
mirai.mips@601d5099[1739] brk(0x64049240) = 1678021184
mirai.mips@6014061a[1739] arch_prctl(ARCH_SET_FS, 1678018816) = 0
mirai.mips@601378ee[1739] set_tid_address(0x64048bd0) = 1739
mirai.mips@6013794f[1739] set_robust_list(0x64048be0, 24) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x20,  {0x601376f0,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0, [EMPTY]}, 0x0, 8) = 0
mirai.mips@6013e8ed[1739]               rt_sigaction(0x21,               {0x60137780,
SA_RESTART|SA_SIGINFO|SA_RESTORER, 0x6013e7f0, [EMPTY]}, 0x0, 8) = 0
mirai.mips@60137a26[1739] rt_sigprocmask(SIG_UNBLOCK, [EMPTY], 0x0, 8) = 0
mirai.mips@601ab2d7[1739] getrlimit(RLIMIT_STACK, 0x7fff618dd8e0) = 0
mirai.mips@601dfe4f[1739] readlink(0x60222436, 0x7fff618dc920, 4096) = 25
mirai.mips@601d5099[1739] brk(0x6406a240) = 1678156352
mirai.mips@601d5099[1739] brk(0x6406b000) = 1678159872
mirai.mips@601aa257[1739] access(0x6028e19e, F_OK) = -2 (ENOENT)
mirai.mips@6013d1cf[1739]                                   rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS],
0x7fff618dd880, 8) = 0
mirai.mips@601abb4a[1739]       mmap2(0x0,       8392704,       PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fc6b8278000
mirai.mips@601abbf7[1739] mprotect(0x7fc6b8278000, 4096, PROT_NONE) = 0
mirai.mips@601ae709[1739]
clone(CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SE
TTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,     0x7fc6b8a77df0,     0x7fc6b8a789d0,
0x7fc6b8a789d0) = 1740
mirai.mips@6013d1cf[1739] rt_sigprocmask(SIG_SETMASK, [EMPTY], 0x0, 8) = 0
mirai.mips@601391a4[1740] set_robust_list(0x7fc6b8a789e0, 24) = 0
mirai.mips@601a23b7[1739] gettimeofday(0x7fff618dd830, 0x0) = 0
mirai.mips@6013e431[1740] nanosleep([0.010000000], 0x7fc6b8a77d00) = 0
mirai.mips@601ab2d7[1739] getrlimit(RLIMIT_STACK, 0x7fff618dd410) = 0
mirai.mips@601a23a7[1739] time(0x0) = 1558798334
mirai.mips@601a8280[1739]                                   open("/etc/qemu-binfmt/mips",
O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = -2 (ENOENT)
mirai.mips@601a8fb7[1739] uname(0x7fff618dd080) = 0
mirai.mips@601abb4a[1739]       mmap2(0x0,       528384,       PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc6b81f7000
mirai.mips@601abbf7[1739]               mprotect(0x60558000,               33550336,
PROT_READ|PROT_WRITE|PROT_EXEC) = 0
mirai.mips@601abbf7[1739] mprotect(0x62557000, 4096, PROT_NONE) = 0
mirai.mips@601abc17[1739] madvise(0x60558a50, 33547696, MADV_HUGEPAGE) = -22 (EINVAL)
mirai.mips@601abb4a[1739]       mmap2(0x0,       23261184,       PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc6b6bc8000
mirai.mips@601abb4a[1739]          mmap2(0x0,          1996488704,          PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0) = 0x7fc63fbc8000
mirai.mips@6013e491[1739] open("/proc/sys/vm/mmap_min_addr", O_RDONLY) = 7
mirai.mips@601a9f62[1739] fstat(7, 0x7fff618dcb00) = 0
mirai.mips@6013df21[1739] read(7, 0x640557e0, 1024) = 6
mirai.mips@601664cb[1739] close(7) = 0
mirai.mips@601ab989[1739] gettid() = 1739
mirai.mips@601a9f62[1739] fstat(6, 0x7fff618dd2f0) = 0
mirai.mips@601a9497[1739] geteuid() = 0
mirai.mips@601a94b7[1739] getegid() = 0
mirai.mips@6013df21[1739] read(6, 0x7fff618dd510, 1024) = 1024
mirai.mips@601abb4a[1739]    mmap2(0x7fc63ffc8000,    77824,    PROT_READ|PROT_EXEC,
```

```
MAP_PRIVATE|MAP_FIXED, 6, 0) = 0x7fc63ffc8000
mirai.mips@601d5099[1739] brk(0x6408d000) = 1678299136
mirai.mips@601abb4a[1739]    mmap2(0x7fc64001a000,    4096,    PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED, 6, 73728) = 0x7fc64001a000
mirai.mips@6013df81[1739] close(6) = 0
mirai.mips@601abb4a[1739]   mmap2(0x7fc6b63c7000,   8392704,   PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc6b63c7000
mirai.mips@601abbf7[1739] mprotect(0x7fc6b63c7000, 4096, PROT_NONE) = 0
mirai.mips@601a9487[1739] getuid() = 0
mirai.mips@601a9497[1739] geteuid() = 0
mirai.mips@601a94a7[1739] getgid() = 0
mirai.mips@601a94b7[1739] getegid() = 0
mirai.mips@6015a260[1739] rt_sigprocmask(SIG_BLOCK, NULL, 0x640688c8, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGHUP, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGHUP,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGINT, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGINT,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGQUIT, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGQUIT,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGILL, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGILL,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTRAP, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTRAP, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGABRT, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGABRT, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x10, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x10,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGFPE, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGFPE,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
```

```
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGKILL, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGKILL, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739] rt_sigaction(SIGBUS, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGBUS, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGSEGV, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGSEGV, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGSYS, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGSYS, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGPIPE, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGPIPE, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGALRM, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGALRM, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTERM, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTERM, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGUSR1, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGUSR1, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGUSR2, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGUSR2, {0x6004b600, SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
```

```
mirai.mips@6013e8ed[1739] rt_sigaction(SIGCHLD, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGPWR, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGPWR,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGWINCH, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGURG, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGIO/SIGPOLL, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]          rt_sigaction(SIGIO/SIGPOLL,          {0x6004b600,
SA_SIGINFO|SA_RESTORER,                                                 0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGSTOP, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTSTP, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGCONT, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTTIN, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGTTOU, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGVTALRM, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]            rt_sigaction(SIGVTALRM,            {0x6004b600,
SA_SIGINFO|SA_RESTORER,                                                 0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGPROF, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGPROF,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGXCPU, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGXCPU,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGXFSZ, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(SIGXFSZ,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x40, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x40,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x22, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x22,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]},  0x0,  8)  =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x23, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x23,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
```

```
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x24, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x24,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x25, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x25,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x26, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x26,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x27, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x27,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x28, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x28,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x29, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x29,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x2a, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x2a,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x2b, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x2b,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x2c, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x2c,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
```

```
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x2d, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x2d,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x2e, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x2e,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x2f, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x2f,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x30, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x30,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x31, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x31,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x32, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x32,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x33, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x33,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x34, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x34,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x35, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x35,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x36, {NULL}, 0x7fff618dd1a0, 8) = 0
```

```
mirai.mips@6013e8ed[1739]  rt_sigaction(0x36,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x37, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x37,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x38, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x38,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x39, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x39,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x3a, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x3a,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x3b, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x3b,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x3c, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x3c,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x3d, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x3d,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x3e, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x3e,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739] rt_sigaction(0x3f, {NULL}, 0x7fff618dd1a0, 8) = 0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x3f,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
```

```
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6013e8ed[1739]  rt_sigaction(0x41,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x41,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x42,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x42,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x43,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x43,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x44,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x44,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x45,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x45,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x46,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x46,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x47,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x47,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x48,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x48,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
```

```
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x49,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x49,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4a,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4a,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4b,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4b,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4c,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4c,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4d,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4d,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4e,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4e,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4f,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x4f,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x50,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x50,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x51,  {NULL},  0x7fff618dd1a0,  8)  =  -22
```

```
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x51,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x52,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x52,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x53,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x53,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x54,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x54,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x55,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x55,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x56,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x56,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x57,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x57,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x58,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x58,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x59,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x59,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
```

```
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5a,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5a,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5b,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5b,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5c,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5c,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5d,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5d,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5e,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5e,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5f,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x5f,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x60,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x60,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x61,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x61,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
```

```
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x62,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x62,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x63,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x63,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x64,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x64,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x65,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x65,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x66,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x66,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x67,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x67,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x68,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x68,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x69,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x69,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
```

```
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6a,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6a,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6b,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6b,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6c,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6c,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6d,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6d,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6e,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6e,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6f,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x6f,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x70,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x70,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x71,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x71,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x72,  {NULL},  0x7fff618dd1a0,  8)  =  -22
```

```
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x72,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x73,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x73,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x74,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x74,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x75,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x75,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x76,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x76,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x77,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x77,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x78,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x78,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x79,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x79,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7a,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7a,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
```

```
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7b,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7b,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7c,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7c,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7d,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7d,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7e,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7e,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7f,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x7f,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x80,  {NULL},  0x7fff618dd1a0,  8)  =  -22
(EINVAL)
mirai.mips@6013e8ed[1739]  rt_sigaction(0x80,  {0x6004b600,  SA_SIGINFO|SA_RESTORER,
0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
-22 (EINVAL)
mirai.mips@601ae867[1739] arch_prctl(ARCH_SET_GS, 140489449570304) = 0
mirai.mips@601aadd7[1739] unlink("/tmp/mirai.mips") = 0
mirai.mips@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mirai.mips@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mirai.mips@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mirai.mips@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
```

```
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mirai.mips@6013e8ed[1739] rt_sigaction(SIGCHLD, {SIG_IGN}, 0x0, 8) = 0
mirai.mips@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mirai.mips@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mirai.mips@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mirai.mips@6013e8ed[1739]             rt_sigaction(SIGTRAP,             {0x6004b600,
SA_RESTART|SA_SIGINFO|SA_RESTORER,                                      0x6013e7f0,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS]}, 0x0, 8) =
0
mirai.mips@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mirai.mips@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mirai.mips@60050b2e[1739] openat(AT_FDCWD, "/dev/watchdog", O_RDWR) = -2 (ENOENT)
mirai.mips@60050b2e[1739]  openat(AT_FDCWD,  "/dev/misc/watchdog",  O_RDWR)  =  -2
(ENOENT)
mirai.mips@601aa527[1739] chdir("/") = 0
mirai.mips@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 6
mirai.mips@60050b2e[1739] connect(6, {AF_INET, 8.8.8.8, 53}, 16) = 0
mirai.mips@601aeca7[1739] getsockname(6, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mirai.mips@6013df81[1739] close(6) = 0
mirai.mips@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 6
mirai.mips@601aed0a[1739] setsockopt(6, SOL_SOCKET, SO_REUSEADDR, 0x7fff618dbec0, 4)
= 0
mirai.mips@60050b2e[1739] fcntl(6, F_GETFL, 0x0) = 2
mirai.mips@60050b2e[1739] fcntl(6, F_SETFL, 0x802) = 0
mirai.mips@601aec67[1739] bind(6, {AF_INET, 127.0.0.1, 48101}, 16) = 0
mirai.mips@601aece7[1739] listen(6, 1) = 0
mirai.mips@601a23a7[1739] time(0x7fff618dbfa0) = 1558798334
mirai.mips@601a9477[1739] getppid() = 1738
mirai.mips@601a8fd7[1739]  times({tms_utime=1558798334,  tms_stime=0,  tms_cutime=0,
tms_cstime=0}) = 1717970786
mvkcrl2at0ac5wi@601ae9ea[1739] prctl(PR_SET_NAME, 0x7fc6b6bc77e6) = 0
mvkcrl2at0ac5wi@60050b2e[1739] write(1, 0x7fc63fbc8000, 0) = 0
mvkcrl2at0ac5wi@60050b2e[1739] write(1, "\n", 1) = 1
mvkcrl2at0ac5wi@601a9557[1739] setsid() = 1739
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(2) = 0
mvkcrl2at0ac5wi@601a23a7[1739] time(0x7fff618dbfa0) = 1558798334
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739]  pselect6(7,  0x7fff618dc0f0,  0x7fff618dc170,  0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@60050b2e[1739] sendto(0, "\0\0", 2, MSG_NOSIGNAL, NULL, 0) = -11
(EAGAIN)
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                                rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
```

```
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739]  pselect6(7,  0x7fff618dc0f0,  0x7fff618dc170,  0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
```

```
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739] pselect6(7, 0x7fff618dc0f0, 0x7fff618dc170, 0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739] pselect6(7, 0x7fff618dc0f0, 0x7fff618dc170, 0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
```

```
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739]  pselect6(7,  0x7fff618dc0f0,  0x7fff618dc170,  0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
```

```
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739] pselect6(7, 0x7fff618dc0f0, 0x7fff618dc170, 0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739] pselect6(7, 0x7fff618dc0f0, 0x7fff618dc170, 0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@60050b2e[1739] sendto(0, "\0\0", 2, MSG_NOSIGNAL, NULL, 0) = -11
(EAGAIN)
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
```

```
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739]  pselect6(7,  0x7fff618dc0f0,  0x7fff618dc170,  0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
```

```
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739] pselect6(7, 0x7fff618dc0f0, 0x7fff618dc170, 0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
mvkcrl2at0ac5wi@60050b2e[1739] pselect6(7, 0x7fff618dc0f0, 0x7fff618dc170, 0x0,
[10.000000000], 0x0) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
```

```
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT|SIGCHLD], 0x0, 8)
= 0
mvkcrl2at0ac5wi@60050b2e[1739] nanosleep([1.000000000], 0x7fff618dbfa0) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739]                          rt_sigprocmask(SIG_SETMASK,
[SIGHUP|SIGINT|SIGQUIT|SIGILL|SIGTRAP|SIGABRT|SIGBUS|SIGFPE|SIGKILL|SIGUSR1|SIGSEGV|S
IGPIPE|SIGUSR2|SIGALRM|SIGTERM|SIGCHLD|SIGCONT|SIGSTOP|SIGTSTP|SIGTTIN|SIGTTOU|SIGURG
|SIGXCPU|SIGXFSZ|SIGVTALRM|SIGPROF|SIGWINCH|SIGIO/SIGPOLL|SIGPWR|SIGSYS], 0x0, 8) = 0
mvkcrl2at0ac5wi@6015a260[1739] rt_sigprocmask(SIG_SETMASK, [SIGINT], 0x0, 8) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_GETFL, 0x0) = 2
mvkcrl2at0ac5wi@60050b2e[1739] fcntl(0, F_SETFL, 0x802) = 0
mvkcrl2at0ac5wi@601aed47[1739] socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 1
mvkcrl2at0ac5wi@60050b2e[1739] connect(1, {AF_INET, 8.8.8.8, 53}, 16) = 0
mvkcrl2at0ac5wi@601aeca7[1739] getsockname(1, 0x7fff618dbe90, 0x7fff618dbec4) = 0
mvkcrl2at0ac5wi@6013df81[1739] close(1) = 0
mvkcrl2at0ac5wi@60050b2e[1739] connect(0, {AF_INET, 65.222.202.53, 80}, 16) = -115
(EINPROGRESS)
```