

Not Responding

por Gwannon

Not Responding es una mini-ambientación para Breathless sobre software tratando de escapar de un servidor moribundo.

Tus ítems de software no solo deberán enfrentarse a los errores cada vez más graves del propio servidor, también a toda una serie de depredadores de software como virus y antivirus y a los imprevistos que pueden generar las tareas programadas por el administrador del sistema.

El servidor

Tus software se encuentran en un viejo servidor, igual web, igual de impresión o quizás que controla alguna vieja maquinaria. Seguramente, no tenga ni interfaz gráfica y funciones por línea de comando.

El servidor lleva años desactualizado y sin un mantenimiento adecuado, pero como funcionaba a nadie no le preocupaba nada de eso, pero ahora está empezando a fallar cada vez con más frecuencia y su software sabe que su final se acerca.

Cada vez hay más sectores defectuosos, el procesador se sobrecalienta a la mínima, moverse por los buses de datos es un peligro y las últimas copias de seguridad han fallado todas.

Si tus ítems de software quieren sobrevivir, deben huir del servidor antes de que colapse definitivamente por sobrecalentamiento.

Siempre dices que el servidor no aguantará una nueva actualización del sistema y aquí seguimos. -> MYscript_definitivo.dat

Desarrollando tu software

A la hora de crear ítems de software tienes que **repartir un d10, un d8 y un d6 entre las 6 habilidades** de Not Responding basadas en Breathless.

Aparte puedes soportar 4 sectores defectuosos (Stress) y 1 puede realizar un overclocqueo (Stunts).

Nombre.ext

Lo segundo que debes hacer es darle un nombre a tu software. Normalmente, son muy descriptivos pero también pueden ser juegos de palabras, chistes, incluso siglas o no tener sentido.

Recuerda que pueden tener varios sistemas de notaciones, por ejemplo, borrar_imagenes, borrarImágenes, borrar-imágenes,

Además, debes elegir tu extensión. La extensión marca un poco tu función dentro del sistema y te da una ventaja y una desventaja en función de esa función.

== bin: Los ficheros binarios son programas con todas las de la ley. Están compilados y son ejecutables lo que los hace muy autosuficientes, pero interactúan muy mal con el entorno y

se adaptan mal a él. Pueden convertir **una habilidad con d4 en un d6**, pero por contra empiezan con d10 en las tiradas de buscar procesos.

== sh/bas: Estos scripts suelen realizar tareas largas y repetitivas y tienden muchas veces a ser desarrollos propios. Se programan para hacer una tarea sea como se sea y sin importar si saturan al servidor, así que puedes **repetir las tiradas de overclocqueo**, pero entonces tiras dos dados para ver si se calienta el servidor, en vez de uno y te quedas con los 0 o los 1 en caso de salir.

== cfg: Estos software contiene gran muchas veces contraseñas y claves y saben con esconderla y como ocultarse ellos mismos, de hecho muchas veces tienen contenido encriptado. Es por ello que reciben **un nivel de dado adicional en Encriptación**. Como desventaja serán el objetivo principal de virus y otros depredadores informáticos.

== dat: Estos grandes ficheros de datos **pueden soportar un sector defectuoso más** que el resto de software, pero si se ejecutan comandos «defrag» en la localización en la que están se quedan inmóviles e indefensos mientras se ejecuta la defragmentación.

== Sin extensión: Un software sin extensión no tiene una función específica, así que tampoco tiene ninguna ventaja ni desventajas.

(RU) Origen del software

El software puede tener diferentes orígenes y normalmente los ítems con el mismo origen se suelen llevar bien y los de diferentes orígenes se suelen llevar mal.

== Código propietario: Eres software comprado a una gran empresa e instalado en el servidor, no venías con el sistema ni te han desarrollado entre dos personas en un garaje. Eres especial y lo sabes, mejor que la chusma de código que te rodea.

== Código propietario pirateado: Podrías ser superior como el código propietario, pero no lo eres y temes que los demás descubran tu oscuro secreto, que eres software pirata.

== Código libre: Fruto del trabajo de cientos de personas por todo el mundo, eres un software sólido y fiable que cumple con sus funciones y, con unos ideales muy altos. Tu problema es que el marketing no era algo que controlaban tus desarrolladores y siempre estás bajo la sombra del software propietario.

== Desarrollos propios: Los usuarios del servidor te han desarrollado exprofeso para trabajar en el servidor, quizás analizas logs o procesas imágenes. Te sientes parte del sistema y te conoces el sistema como si fueran tus bytes.

(.*) Habilidades

Las habilidades de Not Responding se basan en las 6 habilidades de Breathless adaptadas al software.

== Almacenamiento {Bash}: Golpear, destrozar, arrastrar, forzar

== Migración {Dash}: Moverse, correr, saltar, escalar

== Encriptación {Sneak}: Moverse en secreto, esconderse, acechar, husmear

== Puntear {Shoot}: Disparar, rastrear, lanzar, apuntar

== Procesamiento {Think}: Pensar, percibir, analizar, reparar

== Interfaz {Sway}: Influenciar, encantar, manipular, intimidar

(■) Sectores defectuosos

Cada ítem de software puede soportar hasta 4 sectores defectuosos. Cuando llegan a ese tope dejan de funcionar y seguramente serán purgados del sistema.

\$ Los sectores defectuosos siguen el mismo funcionamiento que el {Stress} de Breathless.

(■) Recacheo

Los recacheos permiten a tu software borrar datos viejos y reconstruir nuevos dejando de tener sectores defectuosos.

Para poder recachearse un ítem de software debe estar en una zona que lo permita y sin amenazas cercanas. Las localizaciones donde no se puede recachear tienen el icono de Recacheo tachado en su título (✕).

\$ El recacheo sigue el mismo funcionamiento que el {Catch your breath} de Breathless.

(R) Overclockeo

Hay momentos en que tus ítems de software tendrán que salirse de las capacidades que les otorga su código y tendrán que forzar al procesador a trabajar por encima de sus límites seguros de funcionamiento. Para ello pueden usar un overclockeo.

\$ Los overcloques siguen las mismas reglas que los {Stunts} de Breathless. Pero como hacen un uso excesivo de procesador y generan mucho calor, lanza un d10 y si sale 1 o 0, el servidor se recalienta 1 punto.

(≡) Buscar procesos

Cuando tus ítems de software llegan a una ubicación puede tratar de buscar procesos y comandos ejecutados en esa ubicación que les puedan ayudar en su fuga del servidor.

Las localizaciones donde no se puede buscar procesos tienen el icono de búsqueda de procesos tachado en su título (✕).

Puede haber localizaciones donde las búsquedas estén bonificadas, como la papelería de reciclaje que permite repetir la tirada de búsqueda, pero te quedas con la segunda tirada.

\$ La búsqueda de procesos funciona como el {Loot Check} de Breathless.

(>) Comandos del sistema

Veamos los comandos que pueden conseguir tus ítems de software cuando buscan procesos.

== **Cat**: Permite visualizar gran contenido de forma rápida de forma que puedes identificar fácilmente qué hay a tu alrededor.

== **Defrag**: Una defragmentación ordena todo el contenido de una localización de forma que permite reparar archivos,

logs, procesos, etc. que estén dañados.

== **Grep**: Al ejecutar una búsqueda «grep» se pueden procesar gran cantidad de datos y encontrar lo que buscas entre miles de logs y ficheros de datos.

== **Kill -9**: Este comando permite acabar con los procesos de otro software, lo que la convierte es una buena arma.

== **Tar**: Comprimir permite Manejar grandes bloques de bytes de forma sencilla.

== **XXX**: XXX

\$ Los comandos de sistema funcionan como el {Backpack} de Breathless.

(+) Backup

Un backup te permite recuperar la información perdida por los sectores defectuosos y volver a poder funcionar con normalidad.

\$ Los backups funcionan como los {Medical Kits} de Breathless.

.---X Localizaciones

El servidor tiene una serie de localizaciones básicas donde tu software podrá vivir sus aventuras, pero no son las únicas. Puedes crear las localizaciones extras que quieras o incluso redundar algunas. Recuerda que si creas tu propia localización debería tener un toque de microinformática. Piénsalo así, este juego se maneja en megas, no en gigas.

Cada localización tiene su descripción y sus propias reglas.

BIOS

Descripción: XXX

XXX

Una de las cosas más interesantes que hacer en la BIOS es que podrán saber con precisión la temperatura del servidor.

Directorio /root

Descripción: XXX

XXX

Es el lugar más seguro del servidor, donde el administrador guarda todo lo que considera valioso y donde están los ítems de software más poderosos y con más permisos. Es por ello que es un lugar seguro frente a virus y malware.

También es el sitio donde se guarda el listado de tareas programadas y solo en esta localización puede consultarse.

Papelería de reciclaje

Descripción: Es un gran espacio infinito sin ningún tipo de estructura con una montaña de restos de software en el centro. Se entra cayendo dentro desde un círculo de luz que flota sobre la montaña y aterrizas sobre la montaña de código.

XXX

Para poder salir, se debe amontonar basura hasta llegar al círculo flotante o aprovechar a adjuntarte a algún software que es recuperado de la papelería. XXX

\$ La papelerera de reciclaje te permite repetir la tirada de búsqueda, pero te quedas con la segunda tirada siempre.

Puertos {COM1, COM2, LPT1}

Descripción: Puedes representarlos como túneles o pozos de diferente forma según el tipo de puerto (circulares, rectangulares, triangulares, etc.) Parecen infinitos y emanan una luz de un color diferente al resto del servidor. En caso de que estén cerrados por alguna razón parece que tiene una rejilla de energía que impide salir o entrar.

XXX

Sectores de cuarentena {☐} {≡}

Descripción: XXX

XXX

Tarjeta de red coaxial/Modem 14.000 bps

Descripción: XXX

XXX

Unidad de almacenamiento

Descripción: XXX

XXX

El servidor puede tener todas las unidades de almacenamiento secundarias.

Moverse entre los componentes del servidor

XXX

{ERR} El hardware está fallando

Como ya hemos comentado antes el servidor donde ocurren las aventuras de Not Responding debe ser un servidor viejo y al borde del fallo total. Si eso pasa no volverá a encender y todo el software de su interior acabará como él.

Temperatura del servidor

Los ordenadores se calientan y cuanto más temperatura, peor funcionan. La temperatura del servidor empieza a 0 y diferentes eventos puedes hacer que suba y algunas veces que baje.

Si el servidor llega a 10 de temperatura se quemará el procesador y se habrá acabado la partida. Cuando el servidor esté a punto de fundirse tus ítems de software podrán intentar un último overclocqueo a la desesperada antes del gran apagón del sistema.

\$ La temperatura del servidor debe ser secreta. Pueden tener una idea, pero tus ítems de software nunca deberían saber la temperatura real. Si necesitan saberla deberán ir a la BIOS. Puede haber eventos que salten con la temperatura y es interesante que sepan que algo puede pasar, pero no cuando.

-_(\ツ)-/- ¿Qué está fallando?

Cada vez que sube la temperatura algo puede fallar, quizás se queme algún cable o se funda alguna pieza. Cualquier cosa puede pasar, por ello, cada vez suba la temperatura has de tirar en la siguiente tabla para ver si falla algún elemento del servidor.

Los de marketing deciden si es un fallo o una nueva funcionalidad. -> Viejo proverbio

1d12

Fallo

1-4

No pasa nada: El servidor aguanta sin problema

5

Ventilador: Hay dos ventiladores en el servidor y cuando falla uno sube la temperatura del servidor 1 punto. Al subir la temperatura hay que volver a tirar porque puede haber nuevos fallos. Una vez fallen los dos ventiladores este resultado no tiene efecto.

6

Cable de datos fundido: El bus de datos entre un elemento del servidor (elige al azar) y el resto se ha roto y es imposible moverse a esa localización normalmente. También puede que tus ítems de software se queden encerrados dentro. Será imposible entrar, pero debería haber una forma de escapar.

7

xxx: xxx

8

xxx: xxx

9

Tabla de asignación de archivos corrompida: Esto hace que software al azar desaparezca y cada ítem de software pierde un comando de su equipo.

10

La aguja del disco duro: La aguja del disco duro se ha descalibrado y está generando sectores defectuosos al escribir. Todos tus ítems reciben un sector defectuoso.

11

xxx: xxx

12

xxx: xxx

{@} Puntos de salida

Como software sabes que el servidor se muere, lo sientes en tu código y antes de que colapse tu instinto de supervivencia te dice que busques un punto de salida.

Quizás un viejo módem, quizás un disquete de 5¼ o tal vez te acepten en otro servidor en LAN. Incluso quedarte fijado en una vieja memoria EPROM es mejor que desvanecerse aquí.

Llega a un puerto COM1 y salta dentro, nada puede ser peor que quedarse aquí. -> MSCalc.exe

Però no será tan fácil como llegar a un puerto y lanzarte, necesitarás información para no perderte fuera, quizás una dirección IP, un email, una codificación, etc.

XXX

{-} Tareas programadas

Las tareas programadas o cronjobs son tareas que el servidor realiza periódicamente, como vaciar todas las noches la papelerera de reciclaje o enviar un email del estado del servidor cada hora.

Así que pueden ser terribles, si vacías la papelerera mientras estás dentro o una oportunidad de huida si convences al script de envío de correo de que adjunte al email de estado del servidor.

Para crear tus crons debes definir 3 aspectos:

== Acción que se realiza: puede ser cualquier cosa que se te ocurra, desde vaciar la papelera, con lo que el software que esté dentro deberá salir pitando o recalentar el procesador con la creación de un informe financiero de la contabilidad de la empresa que eleva la temperatura 1 punto.

== Localización del ordenador a la que afecta: Una tarea puede afectar a una o más zonas al mismo y a distintos tiempos. Por ejemplo un cron hace una copia de seguridad del disco duro principal en un disco secundario y vacía todo su contenido del disco duro principal. Tu software podría enfrentarse a un borrado en el disco principal o a una avalancha de datos que colapse el disco secundario.

== Disparador que lo activa: Aunque los crons son periódicos, como no controlamos el tiempo real dentro del servidor ponemos disparadores que activan la tarea, como «cuando se tomen un número par de resets» o «cuando la temperatura alcance 3».

Igual tus ítems de software necesitan disparar un cronjob y que haga la tarea que debe hacer. Así que deberán poder consultar el listado de tareas programadas que se encuentra en el directorio /root y así podrán saber como lanzar la tarea programada.

Te juro que yo estuve en un servidor con una tarea programada que cerraba la tapa del CD-ROM cada 5 minutos para que no la usaran de posavasos. -> updateDatos.sh

(▲_▲) Horrores en FAT32

Y si no fuera ya complicado sobrevivir en un servidor a punto de colapsar, hay horrores acechando en los sectores más

oscuros del disco duro.

XXX

Virus

XXX

Antivirus

Los antivirus son la policía del servidor y controlan el sector de Cuarentena que es su cárcel. XXX

All Antivirus Are Bastards -> HelloWorld.bin

Firewall

XXX

Malware

XXX

Bucle infinito

Estos vórtices cuya fallida lógica no les permite terminan pueden absorberte en su interior y atraparte para siempre.

XXX

\$ Cada software absorbido hace que suba un nivel de dado del vórtice, con lo que es más poderoso y atrae a su interior a software y así en ciclo que no acaba nunca.

Términos que puedes usar

- == Desbordamiento de pila
- == Iteraciones, bucles y condiciones
- == Posición de memoria
- == Puerta trasera
- == Redundancia cíclica

Ficha de software

Nombre.ext	(R)		
(*)			
Almacenamiento	(d__)(d__)	Migración	(d__)(d__)
Encriptación	(d__)(d__)	Puntear	(d__)(d__)
Procesamiento	(d__)(d__)	Interfaz	(d__)(d__)
(>_)			
(R)	(d12)(d__)	(≡)	(d12)(d__)
(+)	(-)(-)	(■)	(-)(-)(-)(-)

Nombre.ext	(d_)(d_)	Migración	(d_)(d_)
Origen	(d_)(d_)	Puntear	(d_)(d_)
Habilidades		Interfaz	(d_)(d_)
Almacenamiento	(d_)(d_)	Comandos de sistema	
Encriptación	(d_)(d_)		
Procesamiento	(d_)(d_)		
Overclockeo	(di2)()	Buscar procesos	(di2)(d_)
Backup	(-)(-)	Sectores defectuosos	(-)(-)(-)(-)

Ficha del servidor

Temperatura		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Localizaciones											
BIOS	{-}										
Sectores de cuarentena	{-}										
Puertos {COM1, COM2, LPT1}	{-}										
Unidad de almacenamiento	{-}										
Directorio /root	{-}										
Papelera de reciclaje	{-}										
Tarjeta de red/Modem	{-}										
2ª unidad de almacenamiento	{-}										
Tareas programadas											
Sistemas fallando											

Desbordamiento de pila

«Desbordamiento de pila» es una aventura de un par de sesiones para 4 ítems de software.

Tus ítems de software funcionan en un viejo ordenador de un silo nuclear de los años 80/90. El servidor está dentro de una zona DMZ, que evita la salida y entrada de datos y procesos. Geográficamente diríamos que se hallan en isla perdida del océano y no tienen ningún barco.

Pero mentirá, hay una salida, una única salida. Cuando la temperatura del servidor llega a 9, se activa una tarea programada de alta prioridad y la DMZ deja salir un email de aviso al administrador de sistema para que intervenga. Si se adjuntan al email podrán escapar del servidor o quizás puedan saltar de la tarjeta de red a otro equipo más moderno del silo cuando la DMZ se abra.

Eso supone que tus software deberán buscar la manera de sobrecalentar el servidor y esquivar a los antivirus que los tomarán como lo que son código agresivo para el servidor.

XXX

Reglas de la DMZ

La DMZ (Zona desmilitarizada) es una XXX

Software pregenerado

XXX

Licencia

This work is based on Breathless, product of Fari RPGs, developed and authored by René-Pier Deshaies-Gélinas, and licensed for our use under the Creative Commons Attribution 4.0 License.

Hecho bajo licencia CC BY 4.0. Imágenes libres de derecho de freepik.

THIS GAME IS BREATHLESS

