Украшаем молоток:

как автоматизировать разбор проблем в дебаггере

Сергей
Козлов

Лаборатория Касперского

C++ RUSSIA

OpenSource-библиотека скриптов для отладчиков

Sol lucet omnibus

https://github.com/KasperskyLab/WinDbg-JS-Scripts

- JS-скрипты для WinDBG
- Python-скрипты для GDB

Библиотека скриптов

- [Windows] Список x32-стеков потоков в kernel дампе
- [Windows] Поиск исключений
- [Windows] Memory-eaters with AppVerifier
- [Linux] Memory-eaters without AppVerifier
- [Linux] Coroutines stackless
- [Linux] Coroutines stackful

x32-стеки

Graviora manet

x32-стеки

x64 kernel dump

```
...
    Child-SP          RetAddr           Call Site
    ffffdb83`1d7af7b0 fffff806`20e79850 nt!KiSwapContext+0x76
    ffffdb83`1d7af8f0 fffff806`20e78d7f nt!KiSwapThread+0x500
    ffffdb83`1d7af9a0 fffff806`20e3d6d2 nt!KiCommitThreadWait+0x14f
    ffffdb83`1d7afa40 fffff806`21201eef nt!KeDelayExecutionThread+0x122
    ffffdb83`1d7afad0 fffff806`2101b7b8 nt!NtDelayExecution+0x5f
    ffffdb83`1d7afb00 00000000`77d21cfc nt!KiSystemServiceCopyEnd+0x28
(TrapFrame @ ffffdb83`1d7afb00)
    00000000`00d3ec88 00000000`00000000 0x77d21cfc
...
```

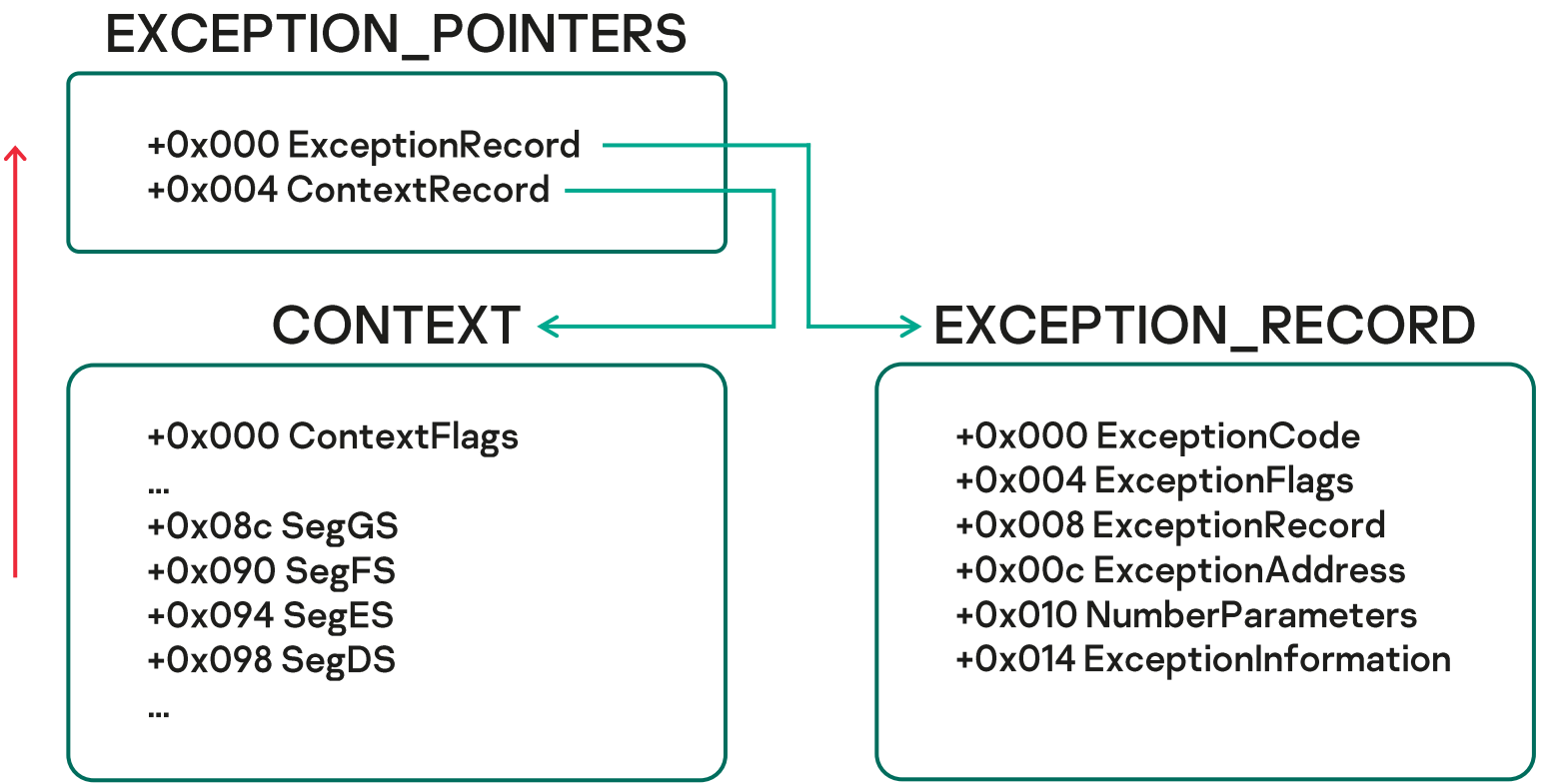
x32-стеки

x64 kernel dump

```
...
 #  ChildEBP  RetAddr
00 00e3fda8 76e43b5b ntdll_77d30000!NtDelayExecution+0xc
01 00e3fe10 76e43aff KERNELBASE!SleepEx+0x4b
02 00e3fe20 74ed5614 KERNELBASE!Sleep+0xf
03 00e3fe44 00922119 MSVCP140!_Thrd_sleep+0x34
04 (Inline) -------- threads_test!std::this_thread::sleep_until+0x109
05 00e3fe90 00921d20
threads_test!std::this_thread::sleep_for<__int64,std::ratio<1,1000> >+0x1d9
06 00e3fea0 0092218a threads_test!DoTask<TaskC>+0x20
07 (Inline) -------- threads_test!std::invoke+0x7
08 00e3febc 77304f9f threads_test!std::thread::_Invoke<std::tuple<void
(__cdecl*)(void)>,0>+0x2a
09 00e3fef4 771efa29 ucrtbase!thread_start<unsigned int (__stdcall*)(void *),1>+0x3f
0a 00e3ff04 77d97a4e KERNEL32!BaseThreadInitThunk+0x19
0b 00e3ff60 77d97a1e ntdll_77d30000!__RlUserThreadStart+0x2f
0c 00e3ff70 00000000 ntdll_77d30000!_RtlUserThreadStart+0x1b
...
```

Поиск исключений

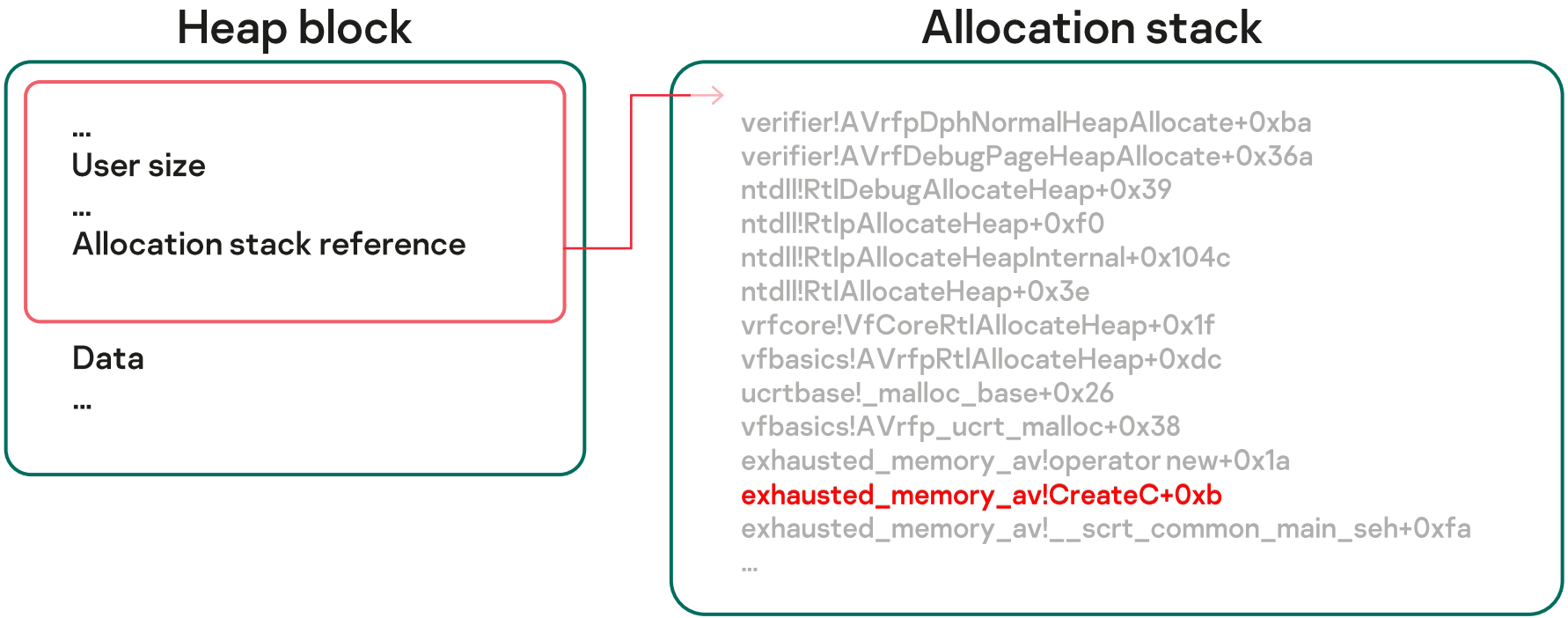
Si quaesiveris, invenies


EXCEPTION_POINTERS
+0x000 ExceptionRecord
+0x004 ContextRecord
CONTEXT
+0x000 ContextFlags
...
+0x08c SegGS
+0x090 SegFS
+0x094 SegES
+0x098 SegDS
...
EXCEPTION_RECORD
+0x000 ExceptionCode
+0x004 ExceptionFlags
+0x008 ExceptionRecord
+0x00c ExceptionAddress
+0x010 NumberParameters
+0x014 ExceptionInformation

Memory-eaters with AppVerifier

Auri sacra fames

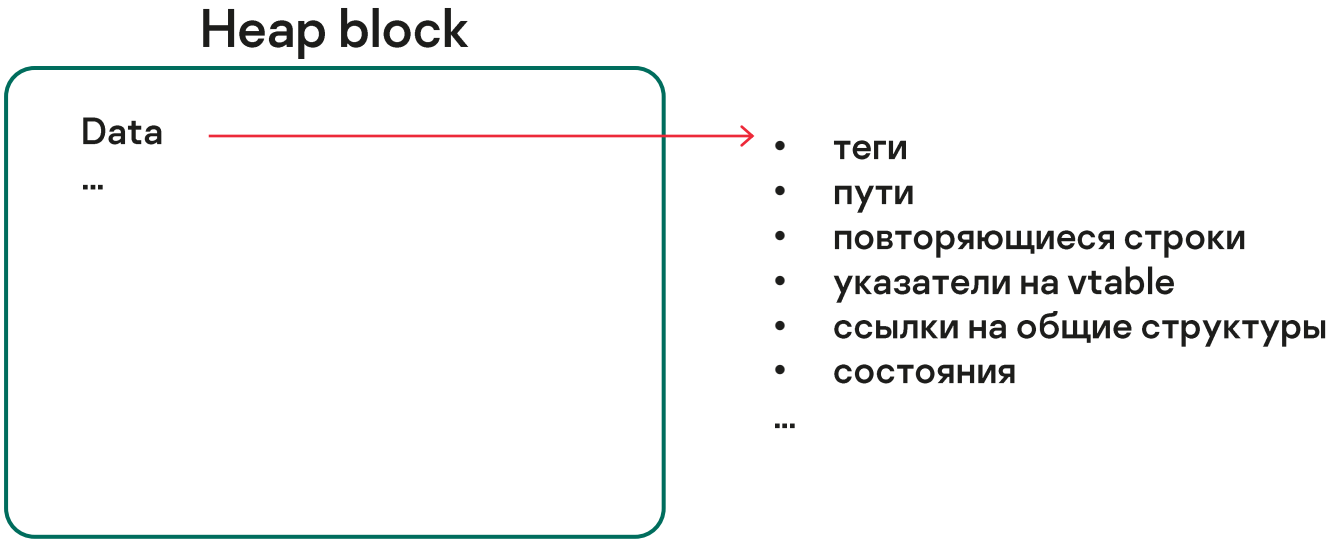
Memory-eaters with AppVerifier



Memory-eaters without AppVerifier

Auri sacra fames

Memory-eaters without AppVerifier



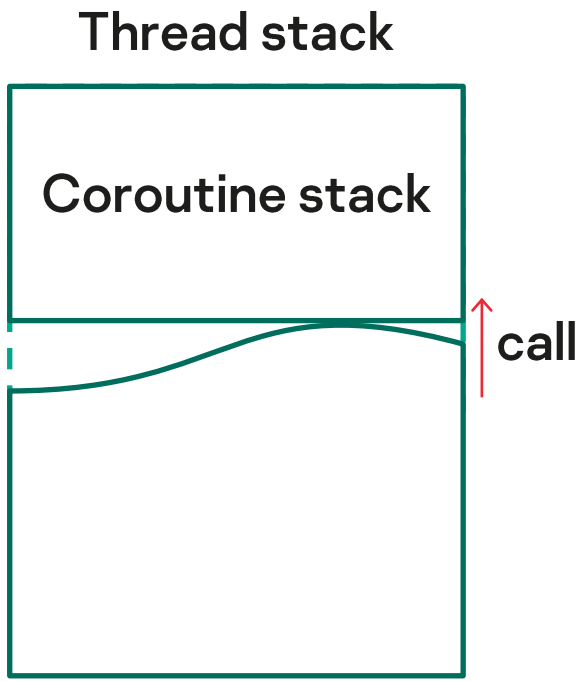
- теги
- пути
- повторяющиеся строки
- указатели на vtable
- ссылки на общие структуры
- состояния

...

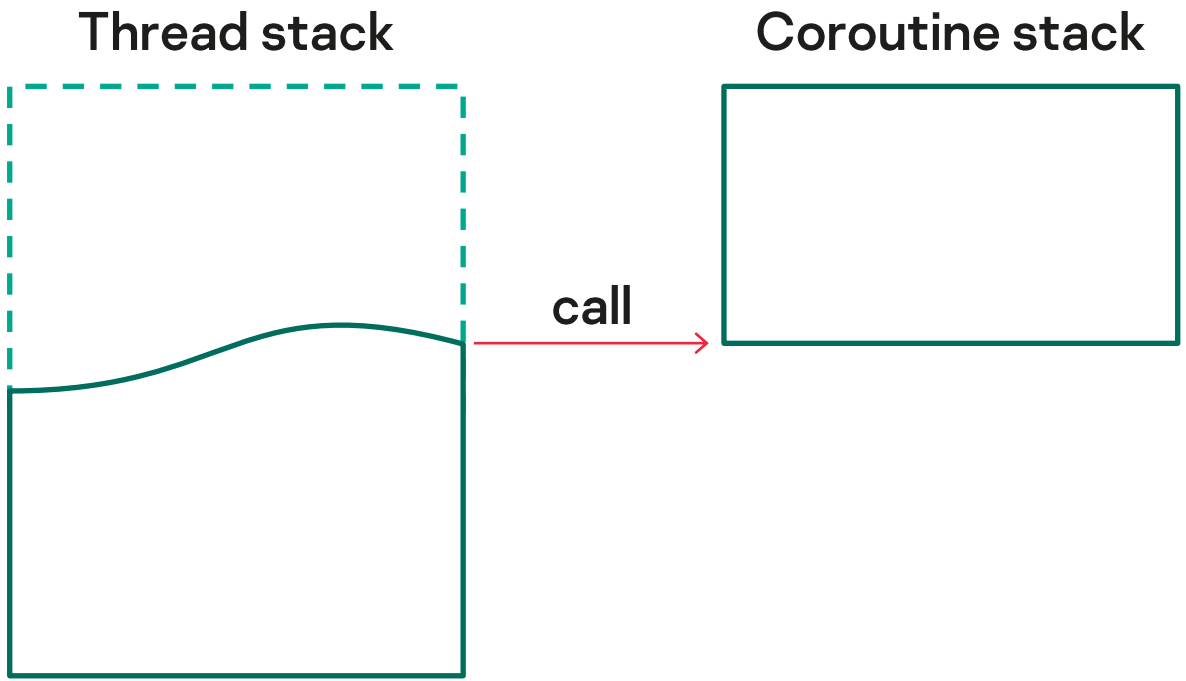
Coroutine

Crescite, nos qui vivimus, multiplicamini

C++20 coroutine – stackless



Boost coroutine – stackful



Thank you!

Сергей Козлов Software Expert sergey.kozlov@kaspersky.com

kaspersky