A few days ago in a cip not far away...

TRACE WARS

i4 PASST - WS 11/12

Attack Surface Reduction

Lorem ipsum dolor sit amet, consectetur *adipisici elit, sed* eiusmod tempor incidunt ut *labore et dolore* magna aliqua. Ut enim ad *minim veniam,* quis nostrud exercitation *ullamco laboris nisi* ut aliquid ex ea commodi *consequat. Quis* aute iure reprehenderit in *voluptate velit* esse cillum dolore eu fugiat *nulla pariatur.* Excepteur sint obcaecat *cupiditat non proident,* sunt in culpa qui *officia deserunt mollit* anim id est laborum.

Inhaltsverzeichnis

Aufgabenbeschreibung

- Verwendete Funktionalität im Linux-Kern für bestimmte Anwendungsszenarien bestimmen
- Analyse und Auswertungen unter Verwendung des Undertaker Tools aus dem VAMOS-Projekt

Schritt 0: System vorbereiten

In der Kernelconfig

- `ftrace` aktivieren
  `CONFIG_FTRACE=y`
- Module deaktivieren
  `CONFIG_MODULES=n`
- Debugsymbole aktivieren
  `CONFIG_DEBUG_INFO=y`

Schritt 1: Verwendungsszenarien simulieren

1. `ftrace`-Logging starten
2. System (eine Zeit lang) gemäss Einsatz verwenden

```
[...]
     <idle>-0     [000] .N..   156.091000: _raw_spin_lock_irq+0x4/0x11 <ffffffff81506f49> \
                                           <-__schedule+0xc2/0x51b <ffffffff81506028>
     <idle>-0     [000] dN..   156.091000: put_prev_task_idle+0x4/0xb <ffffffff81051766> \
                                           <-__schedule+0x26f/0x51b <ffffffff815061d5>
     <idle>-0     [000] dN..   156.091000: pick_next_task_fair+0x11/0x12d <ffffffff81053961> \
                                           <-pick_next_task+0x26/0x4a <ffffffff8104c64a>
     <idle>-0     [000] dN..   156.091000: clear_buddies+0x4/0x8c <ffffffff810529b8> \
                                           <-pick_next_task_fair+0xc1/0x12d <ffffffff81053a11>
     <idle>-0     [000] dN..   156.091000: set_next_entity+0xb/0x99 <ffffffff81052083> \
                                           <-pick_next_task_fair+0xcc/0x12d <ffffffff81053a1c>
     <idle>-0     [000] dN..   156.091000: update_stats_wait_end+0xb/0xaa <ffffffff81051c48> \
                                           <-set_next_entity+0x21/0x99 <ffffffff81052099>
kworker/0:0-4     [000] d...   156.091000: finish_task_switch+0x11/0xdf <ffffffff8104c2a2> \
                                           <-__schedule+0x4d6/0x51b <ffffffff8150643c>
kworker/0:0-4     [000] ....   156.091000: _raw_spin_lock_irq+0x4/0x11 <ffffffff81506f49> \
                                           <-worker_thread+0x38/0x157 <ffffffff8103fb2c>
kworker/0:0-4     [000] d...   156.091000: need_more_worker+0x5/0x3b <ffffffff8103d19f> \
                                           <-worker_thread+0x88/0x157 <ffffffff8103fb7c>
kworker/0:0-4     [000] d...   156.091000: get_gcwq_nr_running+0x4/0x28 <ffffffff8103d15a> \
                                           <-need_more_worker+0x21/0x3b <ffffffff8103d1bb>
[...]
```

Schritt 1.5: Log-Datenmengen reduzieren

Vorgehensweise

- Adressen (Offset und Base) zusammenführen
- Adressen nicht mehrfach ausgeben
- *Aber:* Fehlererkennung durch Redundanz - Funktionsnamen trotzdem mit speichern

```
[...]
18446744071579168231 account_system_time
18446744071579400989 acct_update_integrals
18446744071579032938 jiffies_to_timeval
18446744071579636760 get_mm_counter
18446744071579064851 run_local_timers
18446744071579136085 hrtimer_run_queues
18446744071579038856 raise_softirq
18446744071579390100 rcu_check_callbacks
18446744071579390008 rcu_is_cpu_rrupt_from_idle
18446744071579388305 __rcu_pending
18446744071579016622 printk_tick
18446744071579170638 scheduler_tick
18446744071579062274 cascade
[...]
```

Schritt 2: Quellen herausfinden

- Verwenden der `DWARF` Debugsymbole (via `objdump -l -d -j .text` oder `nm -l` auf `vmlinux`)
- Finden der entsprechenden source file locations

```
[...]
./kernel/sched/fair.c:2714
./fs/read_write.c:479
./kernel/extable.c:94
./arch/x86/kernel/dumpstack.c:163
./kernel/rcutree.c:163
./fs/stat.c:245
./drivers/usb/core/hcd.c:737
./security/capability.c:299
./drivers/tty/serial/serial_core.c:551
./fs/select.c:234
./fs/super.c:216
./kernel/workqueue.c:466
./arch/x86/kernel/apic/apic.c:445
[...]
```

Schritt 3: Reduzierte Konfiguration generieren

- Durch gezielte Blockanalysen wird mit Hilfe der VAMOS-Tools eine Konfiguration mit relevanten Optionen erstellt

```
[...]
CONFIG_SERIAL_VT8500=n
CONFIG_SERIAL_VT8500_CONSOLE=n
CONFIG_SERIAL_XILINX_PS_UART=y
CONFIG_SERIAL_XILINX_PS_UART_CONSOLE=y
CONFIG_SERIAL_XILINX_PS_UART_MODULE=n
CONFIG_SERIAL_ZS=n
CONFIG_SERIAL_ZS_CONSOLE=n
CONFIG_SERIAL_ZS_MODULE=n
CONFIG_SGI_GRU=y
CONFIG_SGI_GRU_MODULE=n
CONFIG_SGI_XP=y
CONFIG_SGI_XP_MODULE=n
CONFIG_SIGNALFD=y
CONFIG_SLAB=y
CONFIG_SLOB=n
CONFIG_SLUB=n
[...]
```

Schritt 4: System verwenden

- Ein im Idealfall schlankeres System ohne nichtbenötigte Funktionalität verwenden
- und Feierabend machen

Jabba The Hutt

- C-Tool zum Echtzeit-Minimieren der Ftrace-Logdaten
- **Eingabe:** ftrace pipe via `stdin`
- **Ausgabe:** einmalig Addressen und Funktionsnamen (sofern vorhanden) via `stdin`
- Verwendet `read` und `write` Systemaufruf, eigene minimale Hashset-Implementierung, sonst nichts
- Analyse der `trace_pipe` mit sehr geringem Aufwand (in $\mathcal{O}(1 \cdot n)$)
- Vereinfachte Kontrolle über `bloatedone.sh`

Obi Wan

- Pythonskript zum Matchen von Addressen auf {Sourcefile:Zeilennummer}
- **Eingabe:** Trace-Logdaten, vmlinux-File mit Debugsymbolen
- **Ausgabe:** relativer Pfad zur Datei und Zeilennummer im Linux-Tree
- Verwendet `nm`, `objdump` und das eigene `jabba`-Log-Format

Undertaker (Vader Release)

Erweiterung des Undertakers um Funktionen der Konfigurationserstellung

- `-j blockconf`: Find configuration enabling specified block
  - **Eingabe:** Quelltextangabe (Format: `<file>:<line>`)
  - **Ausgabe:** Analysedaten, weiterhin wird eine Konfigurationsdatei zu dieser Quelltextangabe erstellt
- `-j mergeblockconf`: Find configuration enabling specified blocks in the given file
  - **Eingabe:** Datei mit Quelltextangaben `(<file>:<line>\n)+`
  - **Ausgabe:** Analysedaten, weiterhin wird eine Konfigurationsdatei erstellt, welche alle Quelltextangaben beinhaltet
- `-B`: Blacklist (analog zur modifizierten `-W`: Whitelist)
  - **Angabe:** Datei mit Blöcken (bzw. Konfigurationsangaben = Features), welche nicht aktiviert werden sollen

Ergebnis

- durch klar definierte Meilensteine war eine gezielte und effiziente Herangehensweise möglich und zugleich wurde der Überblick gewahrt
- Analyse von Modulen ist zu aufwendig (im Rahmen des Praktikums)
- Manlobby dient gut als **Wohn**zimmer
- Black-/Whitelisting für funktionierendes System notwendig

Testlauf

	allyesconfig	defconfig	traced	with lines
features	5.261	941	406	411
golem -l	9.088	1.855	1.216	1.040
vmlinux size	904,1 MB	150,7 MB	92,3 MB	89,1 MB
objdump files	10.211	2.294	1.559	1.390
lines	2.657.436	407.758	243.788	225.235
instructions	13.227.293	1.935.934	1.017.193	1.141.978

Ablauf:

Ca. 5 Minuten Trace in qemu auf standardkonfigurierten Kernel (defconfig), jeweils selbe Funktionalität ausgeführt

Danke

- an `siretart`, `stettberger` und `siccegge` für die geduldige Unterstützung bei der Einführung in VAMOS, Umsetzung und Implementierung
- und sowieso an das gesammte PASST-Team für die erstklassige Betreuung in diesem Semester
- natürlich an alle Zuhörer, die bis jetzt ohne einzuschlafen durchgehalten haben!