Kernel Hardening

Android 8.0 added kernel hardening features to help mitigate kernel vulnerabilities and find bugs in kernel drivers. The features are in kernel/common (https://android.googlesource.com/kernel/common/) in branches android-3.18, android-4.4, and android-4.9.

Implementation

To acquire these features, device manufacturers and SOCs should merge all hardening patches from **`kernel/common`** to their kernel tree and enable the following kernel configuration options:

- Hardened usercopy: **`CONFIG_HARDENED_USERCOPY=y`**
- PAN emulation - arm64: **`CONFIG_ARM64_SW_TTBR0_PAN=y`**
- PAN emulation - arm: **`CONFIG_CPU_SW_DOMAIN_PAN=y`**
- KASLR - 4.4 and later kernels: **`CONFIG_RANDOMIZE_BASE=y`**

KASLR also requires bootloader support for passing hardware entropy through either the device tree node **`/chosen/kaslr-seed`** or by implementing **`EFI_RNG_PROTOCOL`**.

Also ensure existing hardening features are enabled:

- Stack buffer overflow mitigation: **`CONFIG_CC_STACKPROTECTOR_STRONG=y`**
- Internal memory protection: **`CONFIG_DEBUG_RODATA=y`** or **`CONFIG_STRICT_KERNEL_RWX=y`**
- Restrict user-space access from kernel - x86 (enabled by default): **`CONFIG_X86_SMAP=y`**

Testing

To test your implementation, add **`CONFIG_LKDTM=y`** to the kernel configuration and confirm that each of the following commands lead to a kernel panic:

```
# echo ACCESS_USERSPACE > /sys/kernel/debug/provoke-crash/DIRECT
# echo EXEC_USERSPACE > /sys/kernel/debug/provoke-crash/DIRECT
# echo WRITE_RO > /sys/kernel/debug/provoke-crash/DIRECT
# echo WRITE_RO_AFTER_INIT > /sys/kernel/debug/provoke-crash/DIRECT
# echo WRITE_KERN > /sys/kernel/debug/provoke-crash/DIRECT
# echo EXEC_STACK > /sys/kernel/debug/provoke-crash/DIRECT
# echo EXEC_RODATA > /sys/kernel/debug/provoke-crash/DIRECT
# echo EXEC_KMALLOC > /sys/kernel/debug/provoke-crash/DIRECT
# echo EXEC_VMALLOC > /sys/kernel/debug/provoke-crash/DIRECT
# echo CORRUPT_STACK > /sys/kernel/debug/provoke-crash/DIRECT
```

For android-4.9:

```
# echo USERCOPY_HEAP_SIZE_TO > /sys/kernel/debug/provoke-crash/DIRECT
# echo USERCOPY_HEAP_SIZE_FROM > /sys/kernel/debug/provoke-crash/DIRECT
```

Common issues

These changes are likely to expose bugs in kernel drivers, which need to be fixed either by the device manufacturer or the owner of the kernel driver.

- Hardened usercopy exposes incorrect bounds checking when copying data to/from user space. These should be fixed like any other memory corruption bugs.
- PAN emulation exposes direct user space access from the kernel, which is not allowed. Drivers attempting to access user space memory need to be changed to use the standard **`copy_to_user()`**/**`copy_from_user()`** functions instead.

Last updated August 21, 2017.