
black hat
MOBILE SECURITY SUMMIT
LONDON 2015
JUNE 16 - 18, 2015
EXCEL LONDON | LONDON, UK
WWW.BLACKHAT.COM
Inspecting data from the safety of your trusted execution environment
Explorations in the development of advanced security functions
black hat
MOBILE SECURITY SUMMIT
LONDON 2015
UBM

About me

- John Williams, johnwwil [at] u.washington.edu
- Security consultant by day (Ernst & Young)
- Embedded systems security researcher

Talk roadmap

- Introspection at secure/non-secure boundary
- TrustZone/TEE background
- Non-standard environments
- Developing introspection application
- Demo of system call table hook detection PoC

What is introspection?

- Accessing resources of live host (e.g. memory)
- Analysis of memory without using APIs
- Forensics analysis tools have provided a starting point

Why is introspection relevant?

- Complexity/assurance boundaries are prevalent
- Segmentation is backed by hardware
- Hardware capabilities are prevalent
- Users still largely interact with non-assured code

What does introspection provide for mobile?

- Enables range of possibilities
  - System integrity verification
  - Indicator collection/analysis
  - Trusted memory acquisition, etc.
- Do these things in a generic way
  - Preference towards an open solution

TrustZone trusted execution environment (TEE)

- Resource segmentation guaranteed by hardware
- Enables completely parallel execution environment
  - Implementation complexity varies
    - Android leverages it for key storage
    - More complex manufacturer specific proprietary usage

Experimenting with TrustZone

- Hardware availability
  - Assisted by Freescale/USBArmory
- TEE software flexibility
  - Paradigm lockdown
  - We want to do something different

‘Alternative’ secure-world systems

- Still need minimum complexity
  - And a securable architecture
- Not necessarily GP TEE standards compliant
- Ideally will have POSIX compliance

Genode/Noux as base solution

- Capability-based microkernel
- Configures self in secure world, Linux in normal world
  - Requires a few changes to Linux to run simultaneously
- Enforces user/privileged mode split
- Noux is slim API minimally supporting POSIX
  - Minimal attack surface with decent portability

Extending Genode for complex security applications

- Genode running in secure world
- Configure to run Noux for complex applications
  - Asynchronous execution paradigm
  - Noux requires hardware timer for scheduling
  - POSIX support

Support for normal-world introspection

- Create a block driver within Noux
  - Runs as Genode process
  - Wraps existing tz_vmm demo and allows communication
  - Provides way of controlling state and accesses
  - Provides access to normal world physical memory

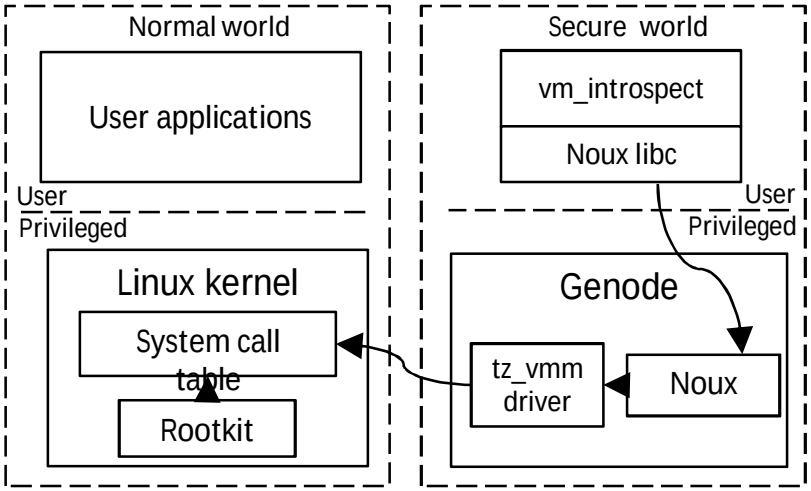
```
<start name="ram_fs">
        <resource name="RAM" quantum="10M"/>
        <provides><service name="File_system"/></provides>
        <config>
                <policy label="noux -> root" root="/" />
        </config>
</start>
<start name="noux">
        <resource name="RAM" quantum="100M"/>
        <provides>
                <service name="Noux"/>
        </provides>
        <config verbose="yes">
                <fstab>
                        <tar name="pylibs.tar" />
                        <tar name="vm_introspect_server.tar" />
                        <dir name="ram"> <fs label="root" /> </dir>
                        <dir name="dev">
                                <terminal name="terminal" label="terminal_fs" />
                                <block name="blkdev0" label="block_session_0" />
                        </dir>
                </fstab>
                <start name="/bin/vm_introspect_server"> </start>
        </config>
</start>
<start name="tz_vmm">
        <resource name="RAM" quantum="14M"/>
        <provides><service name="Block" /></provides>
</start>
```

Initial attempt: Python execution

- Allows for running existing applications
- Compiling libs statically, no dynamic loading
- Successfully ran volatility
- Result: Complex script currently too slow

Developing security application

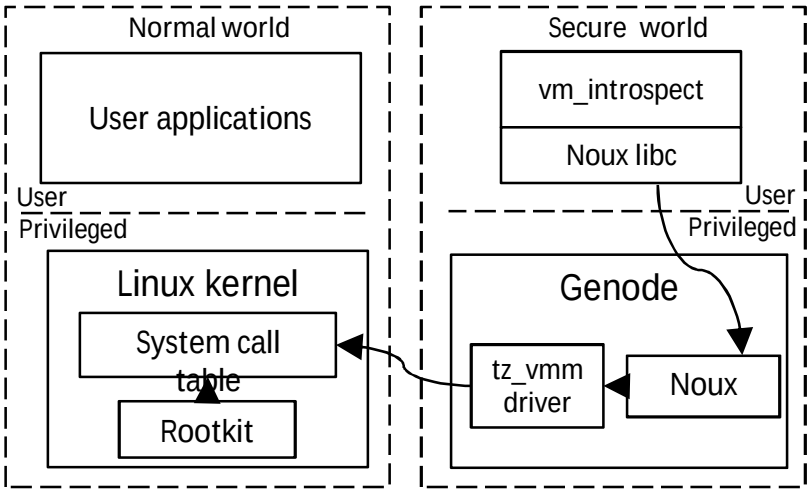
- Based on existing Volatility plugin
  - 'check_syscall_arm'
- Wrote Noux application in C++
  - Executes periodically as scheduled by Noux
  - Validates system call table in normal world
- Tested using MindTrick rootkit


Normal world
User applications
User
Privileged
Linux kernel
System call table
Rootkit
Secure world
vm_introspect
Noux libc
User
Privileged
Genode
tz_vmm driver
Noux

Key takeaways

- Introspection lessons from forensics analysis
- Asynchronous execution provides new use cases
- Hardware extensions are powerful
  - Providing hard segmentation in this case
  - Capable of much more than implementing an API

Demo


Normal world
User applications
User
Privileged
Linux kernel
System call table
Rootkit
Secure world
vm_introspect
Noux libc
User
Privileged
Genode
tz_vmm driver
Noux


black hat
MOBILE SECURITY SUMMIT
LONDON 2015
Thank you!
John Williams
Mobile Introspection using TrustZone
johnwwil [at] u.washington.edu
JUNE 16 – 18, 2015
EXCEL LONDON | LONDON, UK
WWW.BLACKHAT.COM