Kevin Hong

University of Illinois Urbana-Champaign

February 13, 2018
```c
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
    // guaranteed to be random.
}
```
What is randomness?

- From Wikipedia:
What is randomness?

- From Wikipedia:
- Randomness is the lack of pattern or predictability in events.
What is randomness?

- From Wikipedia:
  - Randomness is the lack of pattern or predictability in events.
  - A random sequence of events, symbols or steps has no order and does not follow an intelligible pattern or combination.
Why do we care?

- Random numbers are integral to tons of algorithms
Why do we care?

- Random numbers are integral to tons of algorithms
  - Monte Carlo Methods
Why do we care?

- Random numbers are integral to tons of algorithms
  - Monte Carlo Methods
  - Quicksearch
Why do we care?

- Random numbers are integral to tons of algorithms
  - Monte Carlo Methods
  - Quicksearch
  - If you’re interested in randomized algorithms, take CS 473!
- Luck in games, etc
So we need to generate random numbers?
Based PRNG

- So we need to generate random numbers?
- Methods
So we need to generate random numbers?

Methods

- Pseudorandom Number Generators (PRNG)
  - Deterministic algorithm for generating a sequence of numbers
  - Relies on a random seed
  - Approximates random numbers well
  - CSPRNG
  - Fast, deterministic, periodic
  - Mersenne Twister, xorshift
True Random Number Generators (TRNG)
- Rely on unpredictable physical phenomena
- Atmospheric noise, radioactive decay
- Slow, nondeterministic, non-periodic
- random.org
Randomness in a computer

- In every laptop . . . there lives a die . . .
Randomness in a computer

- In every laptop . . . there lives a die . . .
- That die is /dev/random and /dev/urandom
In every laptop ... there lives a die ... 
That die is /dev/random and /dev/urandom
Entropy Pool
Randomness in a computer

- In every laptop . . . there lives a die . . .
- That die is /dev/random and /dev/urandom
- Entropy Pool
  - Your computer grabs physical specs, keyboard input, mouse movements as entropy
  - Supposedly random bits
  - Keep an estimate of the number of unknown bits
So you want $x$ amount of bits?
Yo dawg . . . I heard you like randomness

- So you want $x$ amount of bits?
- Pull $x$ number of bits from your entropy pool
Yo dawg . . . I heard you like randomness

- So you want $x$ amount of bits?
- Pull $x$ number of bits from your entropy pool
- Hash it using any good hashing algorithm
Yo dawg ... I heard you like randomness

- So you want $x$ amount of bits?
- Pull $x$ number of bits from your entropy pool
- Hash it using any good hashing algorithm
- Enjoy your new random number/
You may notice there’s a difference
You may notice there’s a difference
Random vs unlimited random
You may notice there’s a difference
Random vs unlimited random
Do you need unlimited random?
Cryptography and CS461
Randomized Algorithms in CS473 and 498/598
CS 241 Honors
The cake CPU is a lie

Aneesh Durg

University of Illinois Urbana-Champaign

February 13, 2018
"I think there is a world market for maybe five computers."
- Thomas Watson

- You probably have 5 computers on your right now.
"I think there is a world market for maybe five computers."
- Thomas Watson

- You probably have 5 computers on your right now.
- Problem: Modern world demands high computing powers
"I think there is a world market for maybe five computers."
- Thomas Watson

- You probably have 5 computers on your right now.
- Problem: Modern world demands high computing powers
  - Servers handling many users
  - Enterprise software
  - Crysis 3
"I think there is a world market for maybe five computers."
- Thomas Watson

You probably have 5 computers on your right now.
Problem: Modern world demands high computing powers
  - Servers handling many users
  - Enterprise software
  - Crysis 3
Solution: Virtual Machines!
"I think there is a world market for maybe five computers."
- Thomas Watson

- You probably have 5 computers on your right now.
- Problem: Modern world demands high computing powers
  - Servers handling many users
  - Enterprise software
  - Crysis 3
- Solution: Virtual Machines!
  - Legacy Apps!
"I think there is a world market for maybe five computers."
- Thomas Watson

- You probably have 5 computers on your right now.
- Problem: Modern world demands high computing powers
  - Servers handling many users
  - Enterprise software
  - Crysis 3
- Solution: Virtual Machines!
  - Legacy Apps!
  - What if we had more power than we need?
You probably have 5 computers on your right now.

Problem: Modern world demands high computing powers
- Servers handling many users
- Enterprise software
- Crysis 3

Solution: Virtual Machines!
- Legacy Apps!
- What if we had more power than we need?
- Offers isolation!
Challenges

- What are sensitive instructions?
Challenges

- What are sensitive instructions?
  - All instructions are equal, but some are more equal than others
  - Requires elevated privileges to execute - can’t have everybody breaking the system all the time
Challenges

- What are sensitive instructions?
  - All instructions are equal, but some are more equal than others
  - Requires elevated privileges to execute - can’t have everybody breaking the system all the time
- Trap is not just a kind of music
Challenges

- What are sensitive instructions?
  - All instructions are equal, but some are more equal than others
  - Requires elevated privileges to execute - can’t have everybody breaking the system all the time

- Trap is not just a kind of music
  - ’trap’ the kernel and execute the instruction there
Challenges

- **What are sensitive instructions?**
  - All instructions are equal, but some are more equal than others
  - Requires elevated privileges to execute - can’t have everybody breaking the system all the time

- **Trap is not just a kind of music**
  - ’trap’ the kernel and execute the instruction there
  - e.g. direct access to hardware, enable/disable interrupts, etc.
Problem: What happens if a user tries to execute privileged instructions
Problem: What happens if a user tries to execute privileged instructions
   - You'd hope it traps to kernel
Problem: What happens if a user tries to execute privileged instructions
  - You’d hope it traps to kernel
  - Intel disagrees.
Problem: What happens if a user tries to execute privileged instructions
  - You’d hope it traps to kernel
  - Intel disagrees.

Solution: Lol just silently ignore those pesky users
Problem: What happens if a user tries to execute privileged instructions
- You’d hope it traps to kernel
- Intel disagrees.

Solution: Lol just silently ignore those pesky users
Problem: Some architectures/OSes check have instructions that can do some sensitive instructions
Problem: What happens if a user tries to execute privileged instructions
  - You’d hope it traps to kernel
  - Intel disagrees.
Solution: Lol just silently ignore those pesky users
Problem: Some architectures/OSes check have instructions that can do some sensitive instructions
  - Different behavior when executed by user vs. kernel
Why do we care?
History time!

- Why do we care?
  - This makes virtualization more confusing...
Why do we care?

- This makes virtualization more confusing...
- What if the OS is in user mode?
10 kinds of people in this world...

- Let's build a hypervisor!
10 kinds of people in this world...

- Let’s build a hypervisor!
  - Smaller than a kernel
10 kinds of people in this world...

- Let's build a hypervisor!
  - Smaller than a kernel
  - Allows us to 'virtualize' hardware
10 kinds of people in this world...

- Let’s build a hypervisor!
  - Smaller than a kernel
  - Allows us to ‘virtualize’ hardware
- Type 1 vs Type 2
Let's build a hypervisor!

- Smaller than a kernel
- Allows us to 'virtualize' hardware

Type 1 vs Type 2

Pros and cons to each
The intuitive, hardware-based approach
• The intuitive, hardware-based approach

Figure 8-26. When the operating system in a virtual machine executes a kernel-only instruction, it traps to the hypervisor if virtualization technology is present.
The intuitive, hardware-based approach

- Guest OS/kernel → hypervisor

*Figure 8-26. When the operating system in a virtual machine executes a kernel-only instruction, it traps to the hypervisor if virtualization technology is present.*
The intuitive, hardware-based approach

- Guest OS/kernel $\rightarrow$ hypervisor
- Guest process $\rightarrow$ CPU

Figure 8-26. When the operating system in a virtual machine executes a kernel-only instruction, it traps to the hypervisor if virtualization technology is present.
Type 2

- First made by VMWare in 2006
Type 2

- First made by VMWare in 2006
- On first run needs to boot from 'disk' and install the OS to its own 'disk'
  - The 'disks' are actually just files
Type 2

- First made by VMWare in 2006
- On first run needs to boot from 'disk' and install the OS to it's own 'disk'
  - The 'disks' are actually just files
- Emulates sensitive instructions
Type 2

- First made by VMWare in 2006
- On first run needs to boot from 'disk' and install the OS to it’s own 'disk'
  - The 'disks' are actually just files
- Emulates sensitive instructions
- Runs on top of Guest OS!
Type 2

- First made by VMWare in 2006
- On first run needs to boot from 'disk' and install the OS to its own 'disk'
  - The 'disks' are actually just files
- Emulates sensitive instructions
- Runs on top of Guest OS!
  - Scan blocks of code in OS, if a block of kernel code needs a sensitive
First made by VMWare in 2006
On first run needs to boot from 'disk' and install the OS to its own 'disk'
  The 'disks' are actually just files
Emulates sensitive instructions
Runs on top of Guest OS!
  Scan blocks of code in OS, if a block of kernel code needs a sensitive
  If it's a user mode, do nothing...
Type 2

- First made by VMWare in 2006
- On first run needs to boot from 'disk' and install the OS to it’s own 'disk'
  - The 'disks' are actually just files
- Emulates sensitive instructions
- Runs on top of Guest OS!
  - Scan blocks of code in OS, if a block of kernel code needs a sensitive
  - If it’s a user mode, do nothing...
  - This is called **binary translation**
Why does binary translation work?

- Caching!
Why does binary translation work?

- **Caching!**
  - Can generate a graph of blocks the OS needs as they are available by following branches/jumps
Why does binary translation work?

- Caching!
  - Can generate a graph of blocks the OS needs as they are available by following branches/jumps
  - Once the whole program is caches, should run at native speed
Why does binary translation work?

- Caching!
  - Can generate a graph of blocks the OS needs as they are available by following branches/jumps
  - Once the whole program is caches, should run at native speed
  - Some optimizations like jumping straight to cached blocks
Which one is better?

- Generally type 2
Which one is better?

- Generally type 2
  - Type 1 causes too many traps :(
Which one is better?

- Generally type 2
  - Type 1 causes too many traps :(
  - This leads to poor MMU performance, CPU caching, and branch prediction
More complicated than we thought?

- Paravirtualization
More complicated than we thought?

- **Paravirtualization**
  - Hypervisor as a microkernel
  - Abstraction around hardware interface
  - Requires modified OS

- **Virtualizing IO**
More complicated than we thought?

- Paravirtualization
  - Hypervisor as a microkernel
  - Abstraction around hardware interface
  - Requires modified OS

- Virtualizing IO
  - What about reading and writing from memory?
Paravirtualization
- Hypervisor as a microkernel
- Abstraction around hardware interface
- Requires modified OS

Virtualizing IO
- What about reading and writing from memory?

Licensing?
More complicated than we thought?

- Paravirtualization
  - Hypervisor as a microkernel
  - Abstraction around hardware interface
  - Requires modified OS

- Virtualizing IO
  - What about reading and writing from memory?

- Licensing?
  - If you have a licence to run an OS on one machine is it one real machine or one machine?
Containerization (The final frontier...)

- Docker!
Containerization (The final frontier...)

- Docker!
- Lots of overlapping features
  - Isolation
  - Low cost
  - Multiple OSes
No need to virtualize all the hardware/entire OS
Containerization (The final frontier...)

- No need to virtualize all the hardware/entire OS
- Can share libraries, executables, drives, etc.
Containerization (The final frontier...)

- No need to virtualize all the hardware/entire OS
- Can share libraries, executables, drives, etc.
- Made possible by software like **aufs**
  - Layered FS that can have another ‘real’ fs underneath.
No need to virtualize all the hardware/entire OS
Can share libraries, executables, drives, etc.
Made possible by software like **aufs**
  - Layered FS that can have another ’real’ fs underneath.
choose the right tool for the right task.
Sources

- http://searchservervirtualization.techtarget.com/answer/How-is-containerization-different-from-virtualization
- Modern Operating Systems $3^{rd}$ edition. Andrew S. Tanenbaum