6.033 Spring 2019
Lecture #26

• Low-level exploits
can we prevent buffer overflows?
why not do bounds checking?
compilers: can we trust them?

```c
int main() {
    x++;
    ...
}
```

C Compiler

program machine code

011001101001
compilers: can we trust them?

int main() {
    x++;
    ...

C Compiler
source code

program
source code

C Compiler
machine code

program
machine code
011001101001

6.033 | spring 2019 | lacurts@mit.edu
compilers: can we trust them?

C Compiler source code → C Compiler machine code → C Compiler machine code
compilers: can we trust them?

UNIX source code → C Compiler machine code → UNIX machine code
compilers: can we trust them?

Hacked UNIX source code → C Compiler machine code → Hacked UNIX machine code (has backdoor)

this backdoor is easily discovered in the hacked UNIX source
compilers: can we trust them?

The hacked C compiler has code that *inserts* a backdoor into UNIX

this backdoor *does not* exist in the UNIX source…
but it does exist in the hacked C Compiler source
compilers: can we trust them?

The hacked C compiler has code that *inserts* a backdoor into UNIX.

what if i just lie, and tell you that the hacked C compiler was generated from the clean C compiler source? can you check?
The hacked C compiler has code that *inserts* a backdoor into UNIX.
compilers: can we trust them?

The hacked C compiler has code that *inserts* a backdoor into UNIX.

what if i just lie, and tell you that the hacked C compiler was generated from the clean C compiler source? can you check?

**Yes**: by recompiling the compiler, and then compiling the UNIX source
compilers: can we trust them?

The hacked C compiler has code that inserts a backdoor into UNIX and code to insert backdoor-inserting code into C compilers.
REFERENCES


where to go from here

6.828 - Operating Systems
6.829 - Computer Networks
6.830/6.814 - Database Systems

6.858 - Computer Systems Security
6.857 - Network and Computer Security
6.875 - Cryptography and Cryptanalysis

6.824 - Distributed Systems
6.826 - Principles of Computer Systems
6.852 - Distributed Algorithms

6.903 - Intellectual Property
6.904 - Ethics for Engineers

more systems

more math

a natural follow-up to 6.033