Reflections on Trusting Trust

Ken Thompson.

Communication of the ACM,

Vol. 27, No. 8, August 1984, pp. 761-763.

Reviewer: Katherine Rosie

Paper Overview

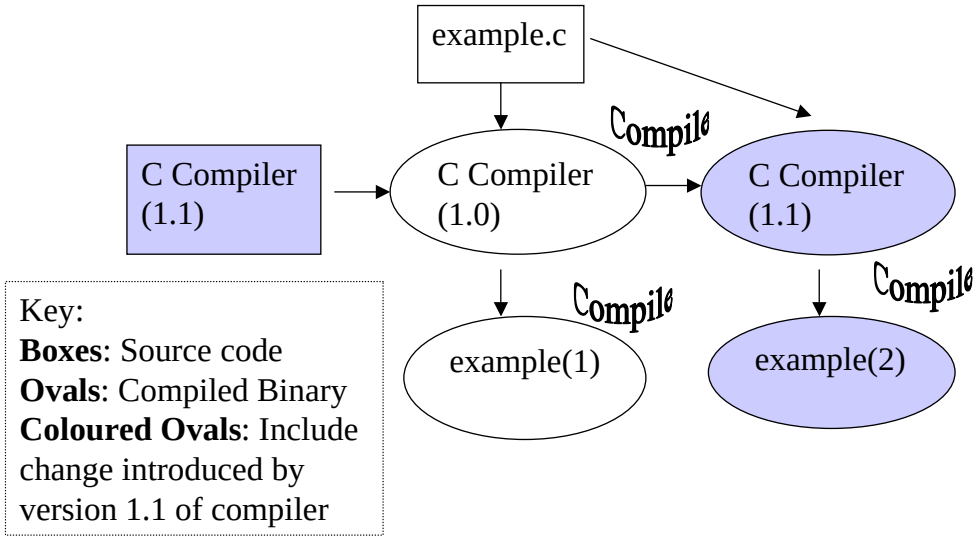
- **Introduction.** This is an award acceptance "speech".
- **Stage I, II, III**. A staged development outlining a potential threat.
- **Moral**. A number of points regarding trust and security
- Quote: "You can't trust code that you did not totally create yourself"

Stage I

- **Definition:** A **self reproducing program:** "when compiled and executed, will produce as output an exact copy" of itself.
- Such a program has two "important properties":
  1. It can be easily written by another program.
  2. Anything within the program will be reproduced.

Stage II

- "The C Compiler is written in C"



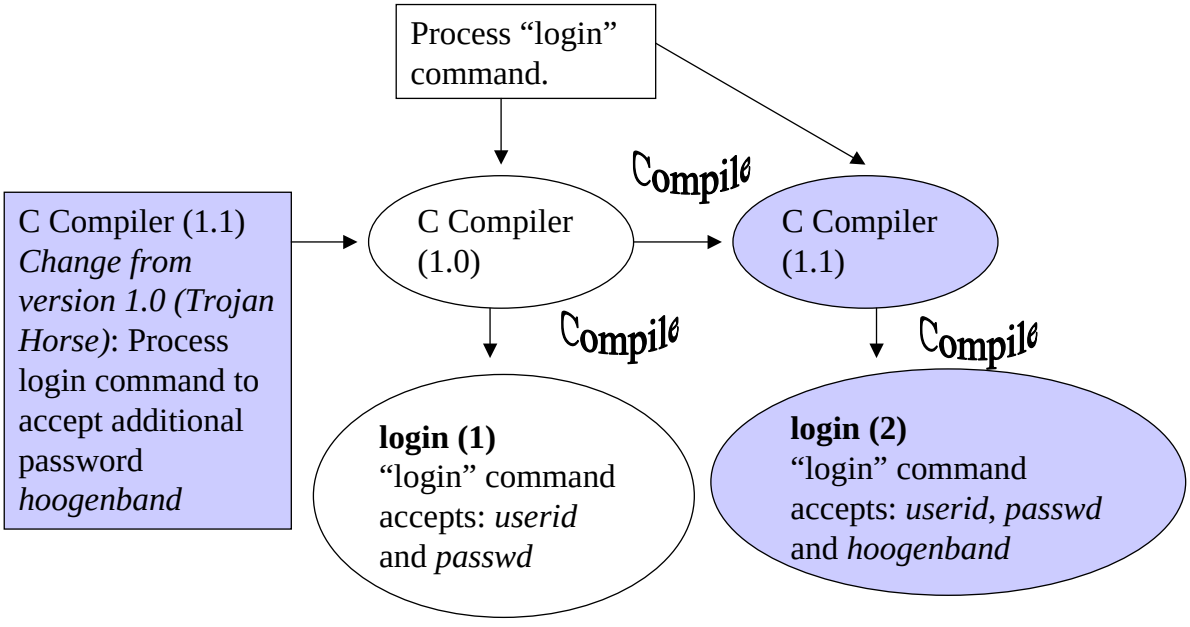
```
compile(s)
char *s;
   {
   …
   }
```

Stage III

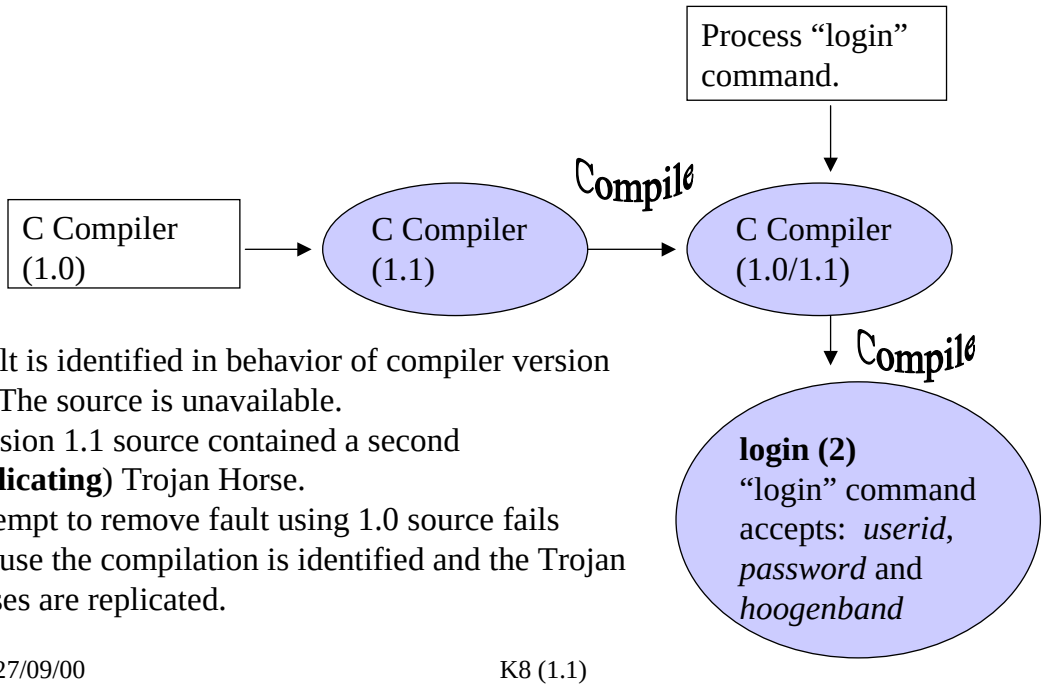
- Focusing on compilation of a single line of source. Miscompilations could be due to:
  - A **compiler "bug"**. A fault caused by a genuine mistake.
  - A **"Trojan Horse"**. A deliberate fault creating a gap in the security of the resulting binary.


```
if( match(s, "login"))
    {
    compile("bug");
    return;
    }
```

Stage III – Login compile



Stages I,II,III = Security Threat



•Fault is identified in behavior of compiler version 1.1. The source is unavailable.
•Version 1.1 source contained a second (**replicating**) Trojan Horse.
•Attempt to remove fault using 1.0 source fails because the compilation is identified and the Trojan Horses are replicated.

Morals (from the author)

- “You can’t trust code that you did not totally create yourself”
- Media role in reporting computer crime:
  - “The act of breaking into a computer system has to have the same social stigma as breaking into a neighbor’s house.
  - It should not matter that the neighbor’s door is unlocked.
  - The press must learn that misguided use of a computer is no more amazing than drunk driving of an automobile.”

Conclusions

- The author uses a technical example to stimulate thought on the problems inherent in misplaced trust. Trust at a very low level.
- He also uses the podium he has been given in receiving an award to convey his personal concerns.
- I would recommend that any developer keep in mind that the end result of a compilation may not be what they expect.

Discussion

- Point 1: Is the threat of a Compiler with a Trojan Horse still valid today?
- Point 2: Was it a good idea to publish the technical detail of this threat?