# Computer Security

Prof. Dr.-Ing. Volker Roth
Freie Universität Berlin

May 14, 2012

**Question 1: Transformation between the Chinese wall security policy and the Bell-La Padula model**

The following composition of objects in the BLP model was derived from a set of objects in the Chinese wall model according to *The Chinese wall security policy, Dr. David F.C. Brewer and Dr. Michael J. Nash, IEEE 1989*.
Sketch the original composition of objects in the Chinese wall model.

Table 1: Composition of object in the BLP

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 & 0 | (A,f) | 2 & 0 | (B,z) | 3 & 0 | (C,t) | 4 & 0 | (D,f) |
| 5 & 0 | (A,g) | 6 & 0 | (B,f) | 7 & 0 | (C,e) | 8 & 0 | (D,l) |
| 9 & 0 | (A,h) | 10 & 0 | (B,l) | 11 & 0 | (C,p) | 12 & 0 | (D,k) |
| 13 & 0 | (A,i) | 14 & 0 | (B,m) | 15 & 0 | (C,q) | 16 & 0 | (D,j) |
| $(X_0, Y_0)$ | | | | | | | |

**Question 2: Assembly quine**

Write an assembly program which prints it's own source.
Explain every line of your program and why the line is relevant. Your results will be judged on the explanation and on the output of your program. If piped to file a *diff* between your source and the output must not show any differences.
The chain of commands to verify your submission will be: *unzip -R group_name zipfile*, *cd group_name*, *make*, *./out > out.s*, *diff out.s orig.s*
If you don't obey any of the following rules your submission will not be accepted.

**Rules**

1. The program has to be written in assembly
2. The program should be as short as possible
3. The program must not be empty
4. The program has to run on Debian 6.0.4 i386
5. The program is not allowed to just read a file and print it out
6. The program has to exit cleanly
7. Name the source file *orig.s*
8. Name the resulting binary *out*
9. Hand in a *Makefile* to build your program
10. Pack your source file and *Makefile* into a zip file with no subdirectories
11. Hand in your zip file via email to jan-ole.malchow@fu-berlin.de before May 21st 12:00 a.m.

**Question 3: Covert channels**

Write a program in pseudocode which transfers the message given below via computational delays from one process to another secretly (see *A Note on the Confinement Problem. Lampson, ACM 1973*).
Since a covert channel has to keep a low profile use a Huffman encoding for the message. Don't forget to write down the encoding table.

Message: *The confinement problem, as identified by Lampson, is the problem of assuring that a borrowed program does not steal for its author information that it processes for a borrower.*