6 Assignment 6 (11 Points)

Hinweis:	Abgabe in $\{2, 3, 4\}$-er Gruppen.
Abgabe:	**03.06.2017, 23.59**
Email:	Betreff "[Compsec] Ex6'
	(bitte nur .pdf oder .txt, kein .doc, .jpeg, etc)
	Source code: bitte inkl. `signify` Signatur

Exercise 17 (Honeywords **(3+2 Points)**).
Consider two users Alice and Bob, both having accounts at three web sites `1.example.org`, `2.example.org` and `3.example.org`. Alice chooses to use three separate passwords for 1, 2 and 3 while Bob wants to use the same.

1. Illustrate how websites 1 and 2 could implement *Honeywords* [1] two protect Alice and Bob from password cracking. In particular,
   - How should decoy passwords be constructed?
   - How many decoy passwords should be generated?

   Explain your decisions. (If you need to, you may make assumptions on how Alice's and Bob's passwords look like on 1 and 2 - for example enforced by the password registration process.)

2. Using your construction, analyze the success probability of an adversary impersonating Alice and Bob to 3, who successfully breaches (excluding the Honeychecker)
   - only `1.example.org` or `2.example.org`
   - both.

   Note: you must assume that the process *how* decoys are generated is publicly known (no security through obscurity).

Exercise 18 (x86 assembly recap **(2 Points)**).
Write a simple, minimal hello-world program in x86 assembly. Write it yourself and do not only disassemble a C-program. Your program should compile with `gcc` on your OpenBSD virtual machine (please provide a Makefile) and should run without any errors.

Exercise 19 (Case study μ`-shout` (II): Secure C Coding (**2 Points + 1 Bonus**)).
We want to improve our μ`-shout` prototype. Fix all of your previous programming mistakes (if any). Additionally, have a look at the *SEI CERT C Coding Standard* at

https://www.securecoding.cert.org/

and adjust your implementation by following at least one rule or recommendation. For your submission: document wich programming mistakes you fixed and all the changes you made according to the secure C coding rules (recommendations), as well as the relevant rules (recommendations).

In case you followed any tutorials in order to program your previous exercises please have a look at [2] as well.

Exercise 20 (Keeping your systems secure (**Bonus: 1 Points**))**.**
Are there any new vulnerabilities for your Debian or OpenBSD system since last week (27.05.2016 at 23.59)? If so: state one, **name the programming mistake**, decide if you are affected or not, and report if there are any known work-arounds or patches.

References

[1] Ari Juels and Ronald L. Rivest. Honeywords: making password-cracking detectable. In Proceedings of CCS '13.

[2] Tommi Unruh, Bhargava Shastry, Malte Skoruppa, Federico Maggi, Konrad Rieck, Jean-Pierre Seifert, Fabian Yamaguchi. *Leveraging Flawed Tutorials for Seeding Large-Scale Web Vulnerability Discovery*. arXiv:1704.02786 [cs.CR].