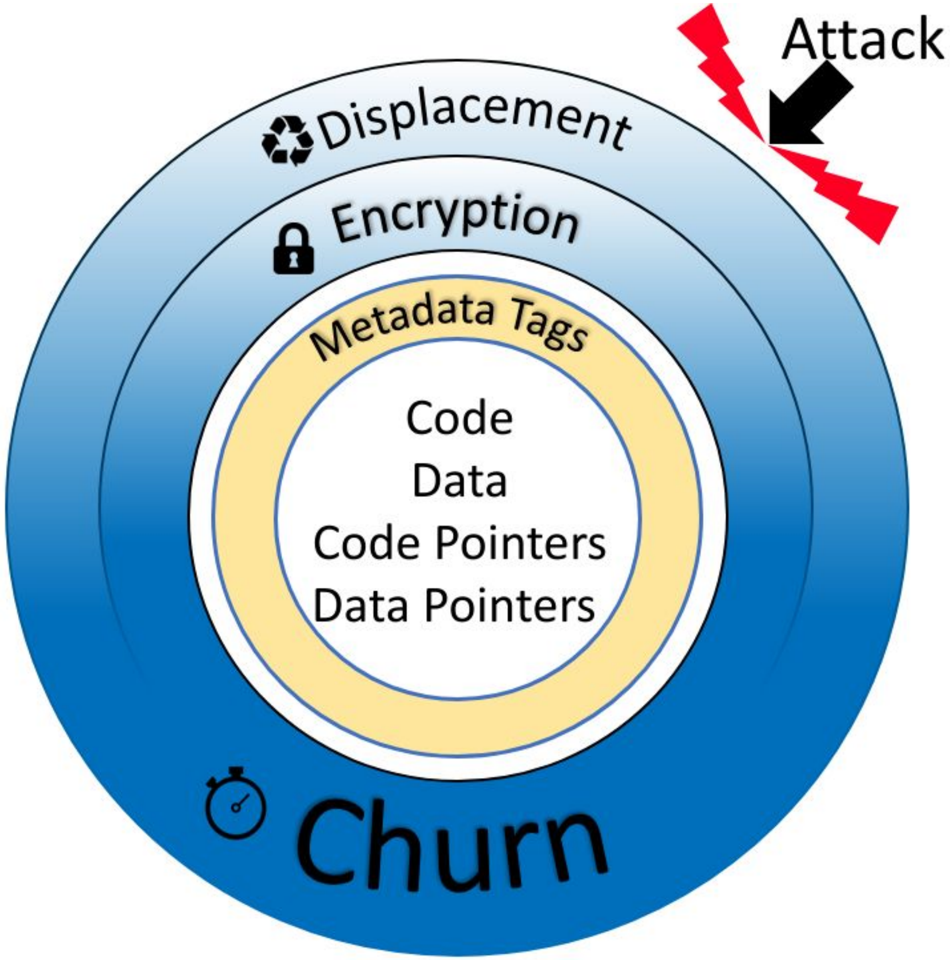
Understanding the Value of Ensemble of Moving Target Defenses in Morpheus

Shibo Chen
Advisor: Todd Austin
Supervisors: Lauren Biernacki, Mark Gallagher

9/6/2018

Morpheus Defenses



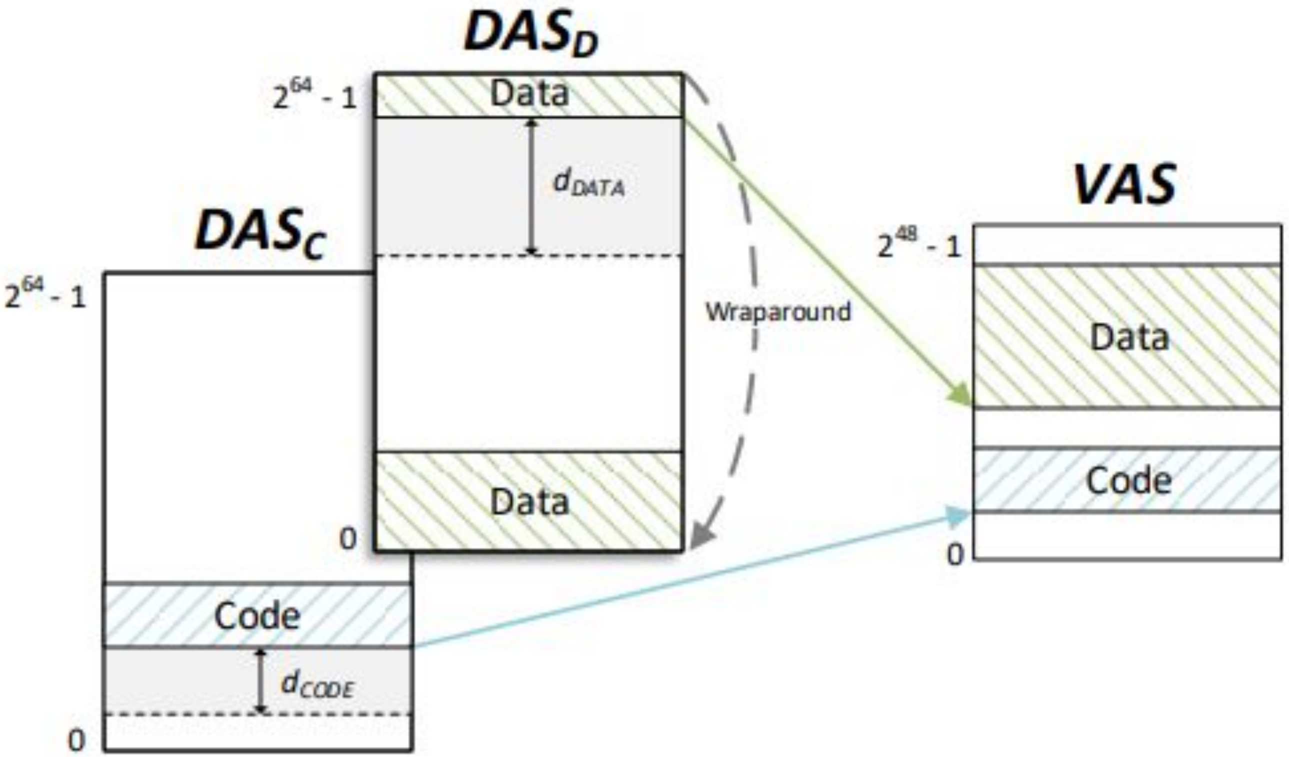
Two Churnable Moving Target Defenses:

- Address Space Disposition
- Domain Encryption

One Attack Detection Unit:

Ramps up churn rate under attacks

Pointer Displacement

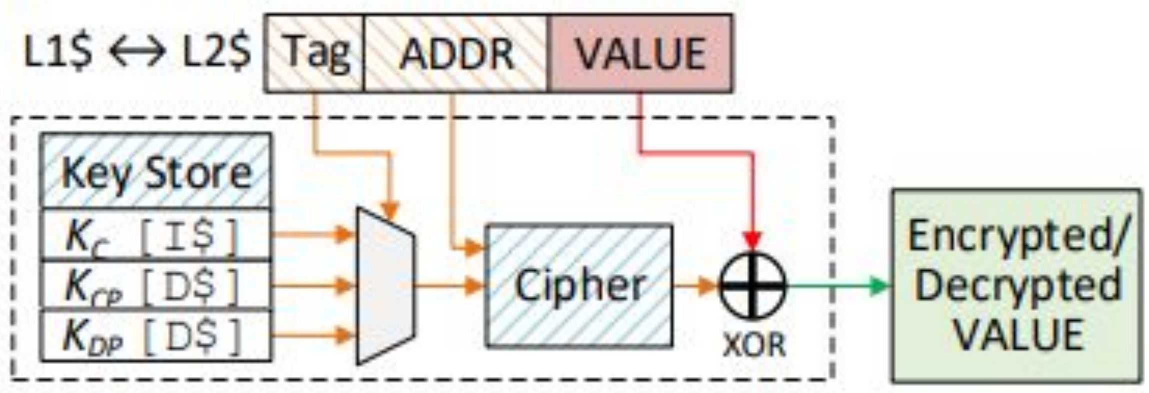


A 60-bit entropy displacement

Data and Code are displaced under different keys

A translation from DAS to VAS is performed at fetches, loads and stores

Domain Encryption



Code, Code Pointer and Data Pointer are encrypted under different keys

Data are getting decrypted when fetched into L1 cache and are encrypted when stored to memory.

How effective is each layer? Is the ensemble necessary? How do we evaluate them?

Idea: Evaluating by Attacking it

How effective is each layer?

We can show the effectiveness of each defense layer by showing how much it deters attacks.

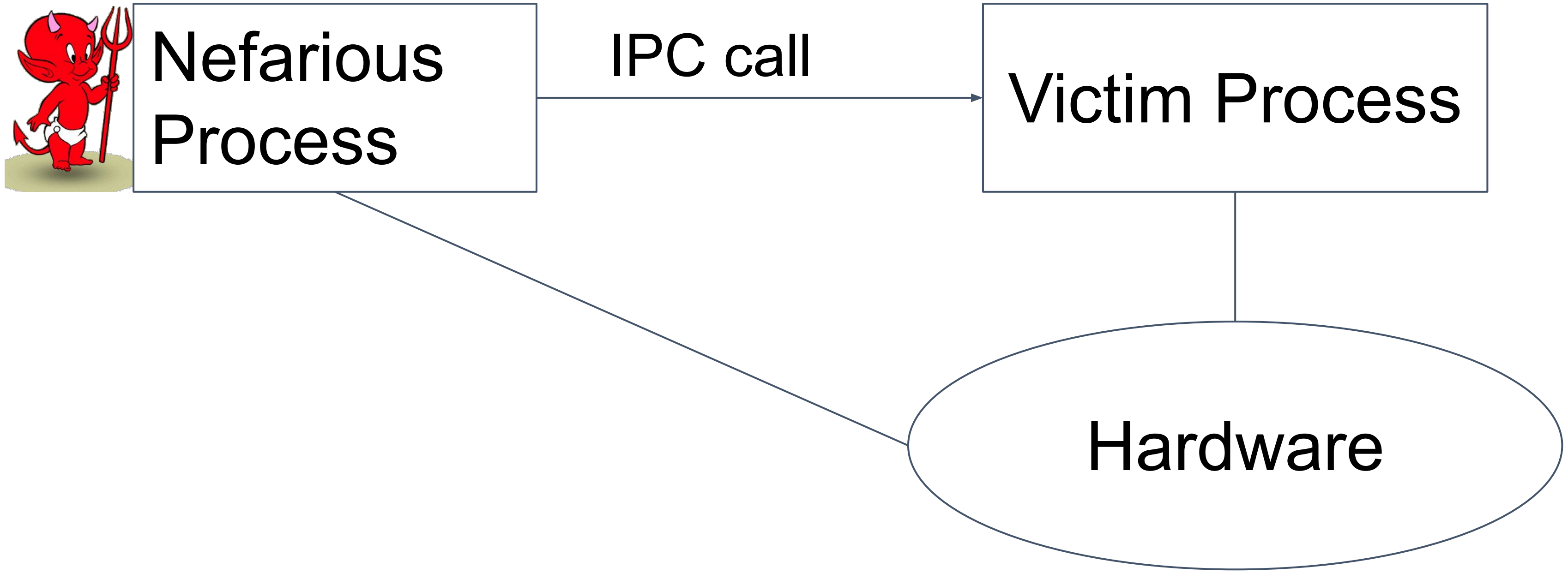
Is the ensemble necessary?

We can show the value of ensemble by showing a combination of two defenses is significantly stronger than any single defense layer.

Threat Model

Fellow Attacker

“I would like to exploit a stack buffer overflow vulnerability in a victim program and open up a shell. (By calling system())”



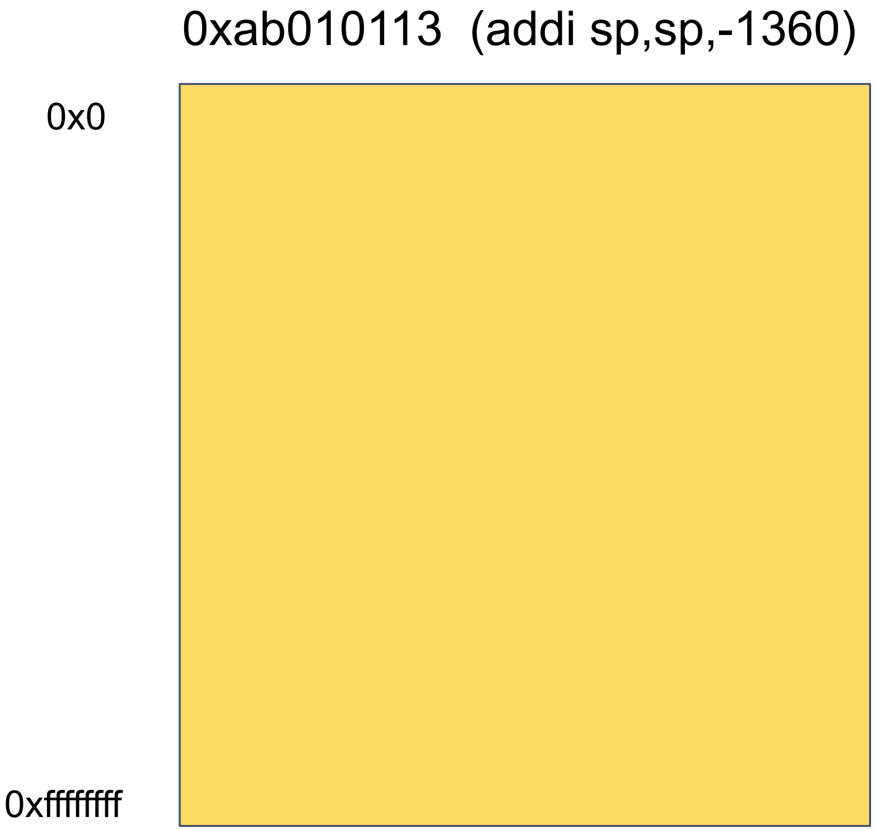
Threat Model and Assumptions

- Function layout get randomized during compilation
- No Churn
- Single encryption key in a process (Forge a code pointer by crafting Data pointer)
- Same displacement in all processes (Nefarious program can attack itself to derandomize the address space)
- Crash Resistant

Attack Ideas

Displacement Off, Encryption Off ($\overline{EP}$):

Victim Address Space



Overall Time Required : 1 * IPC call + Search for *system()*

Attack Details

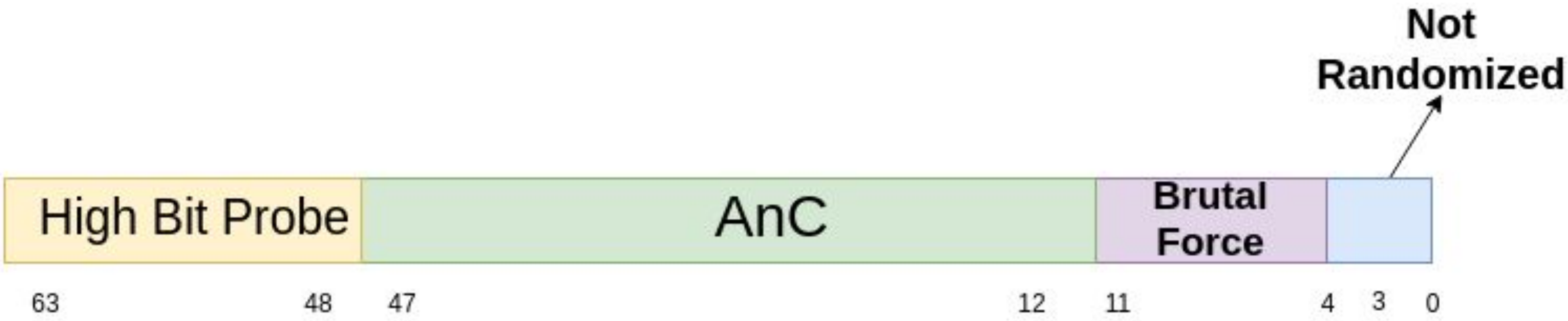
Encryption On, Displacement Off: blind-call into 4-byte aligned positions until a shell pop up.

Overall Time Required:

(IPC call + segfault handling + Mimicry) * (Code Size/4)

Attack Details

Encryption Off, Displacement On:



Overall Time Required: AnC + (IPC call + Segfault) * 2^16 + Search in a program of GO size

Attack Details

Encryption On, Displacement On:

A combination of previous two attacks

Overall Time Required: AnC + (IPC call + Segfault) * 2^16 + (IPC call + segfault handling + Mimicry) * (Code Size/4)

	Cycles	Time (ms)
IPC Call	?	?
Segfault Handling	?	?
AnC	?	?
memstr() the code	?	?
mimicry	?	?

	Cycles	Time (ms)
IPC Call	10000	
Segfault Handling		0.07
AnC		150000
memstr() the code	?	?
mimicry	?	?

Experimental Setups

We use gem5 to simulate the attacks and estimate the time needed for the attacks.

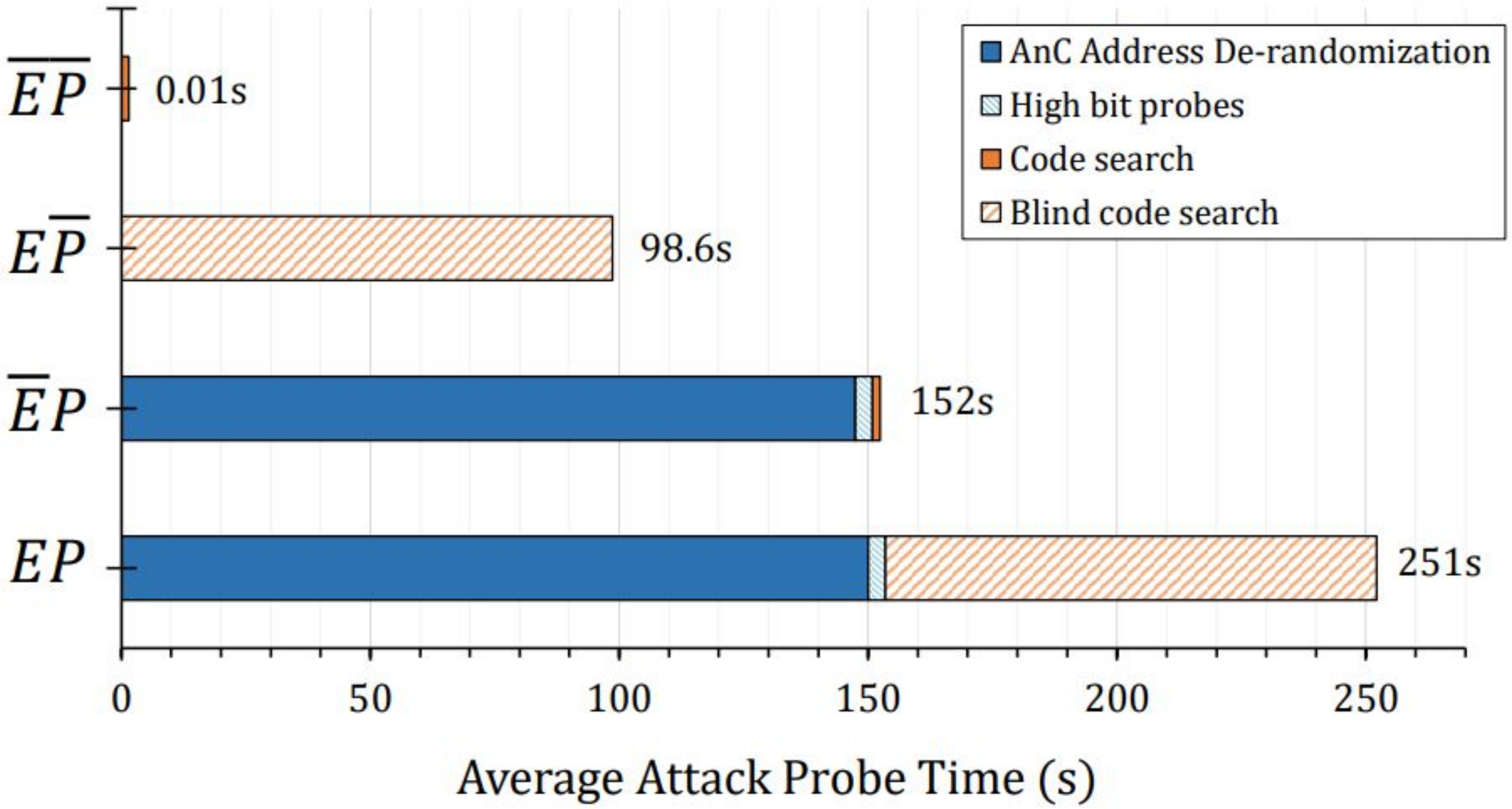
Hardware Configurations:

CPU frequency : 2.5GHz

BTB entries: 4096

	Cycles	Time (ms)
IPC Call	10000	~0
Segfault Handling		0.07
AnC		150000
memstr() the code	28,608,736	11.44
mimicry	79447770	31.8

Results


$\overline{E}\overline{P}$
$E\overline{P}$
$\overline{E}P$
EP
0.01s
98.6s
152s
251s
■ AnC Address De-randomization
▧ High bit probes
■ Code search
▨ Blind code search
0
50
100
150
200
250
Average Attack Probe Time (s)

Discussion and Conclusion

- Each of two defenses layer makes it significantly harder for the attacker to penetrate than no defense.

- The ensemble of two defenses is more concrete than any individual defense and makes the time needed to penetrate the system 5000x greater than the target churn rate - 50ms.

Acknowledgement

- Thank Prof. Austin for giving me the opportunity to work in the lab and his help & patience throughout the program.
- Thank Lauren and Mark for being such responsible supervisors and all the help and suggestions on the project.
- Thank Jacqui, Zelalem and anyone else who has given me help on the project and carried Morpheus forward.
- Thank all the ThundaCats. I really enjoy working with you and I will miss all the happy time we have this summer.
- Thank everyone in ABLab. You make me feel at home!

Thanks for listening!