Kan vi egentligen göra våra ledningssystem säkra?

Stefan Axelsson
Blekinge Tekniska Högskola

2014-11-18

Vad är problemet?

- ”Allt” spec. ledningssystem är datorbaserat
- Datorer “är” mjukvara (och lite hårdvara)
- Denna vara köps idag på den öppna marknaden
  - COTS – Commercial Off The Shelf
    - Om inte systemet, så iaf *kompilatorn*, *operativsystemet*, *mjukvarubiblioteken*, etc.

Är COTS säkert?

- Nej – Ekonomiska faktorer dominerar
  - Har man monopol så kan man ta betalt efter kundens värde
  - Men vid perfekt konkurrens så är värdet marginalkostnaden för produktion
    - Marginalkostnaden för information är NOLL
- Kan man få monopol då?
  - Nätverksekonomier
    - Om många använder så blir “ekosystemet” värt kvadraten på antalet användare
  - Lock in
    - Vissa hävdar att värdet av ett inbördes interoperabelt system är kostnaden för användarna att byta
      - Nytt system, konvertering av data, utbildning osv.

COTS

- Så den som får in foten först vinner *allt*
  - Jmf Microsoft: “Ship it Tuesday – get it right by version 3.0”
- Säkerhet kommer inte med i första vändan
  - Första versionen kommer vara osäker eftersom ingen kommer att fråga efter det...

Akerlof

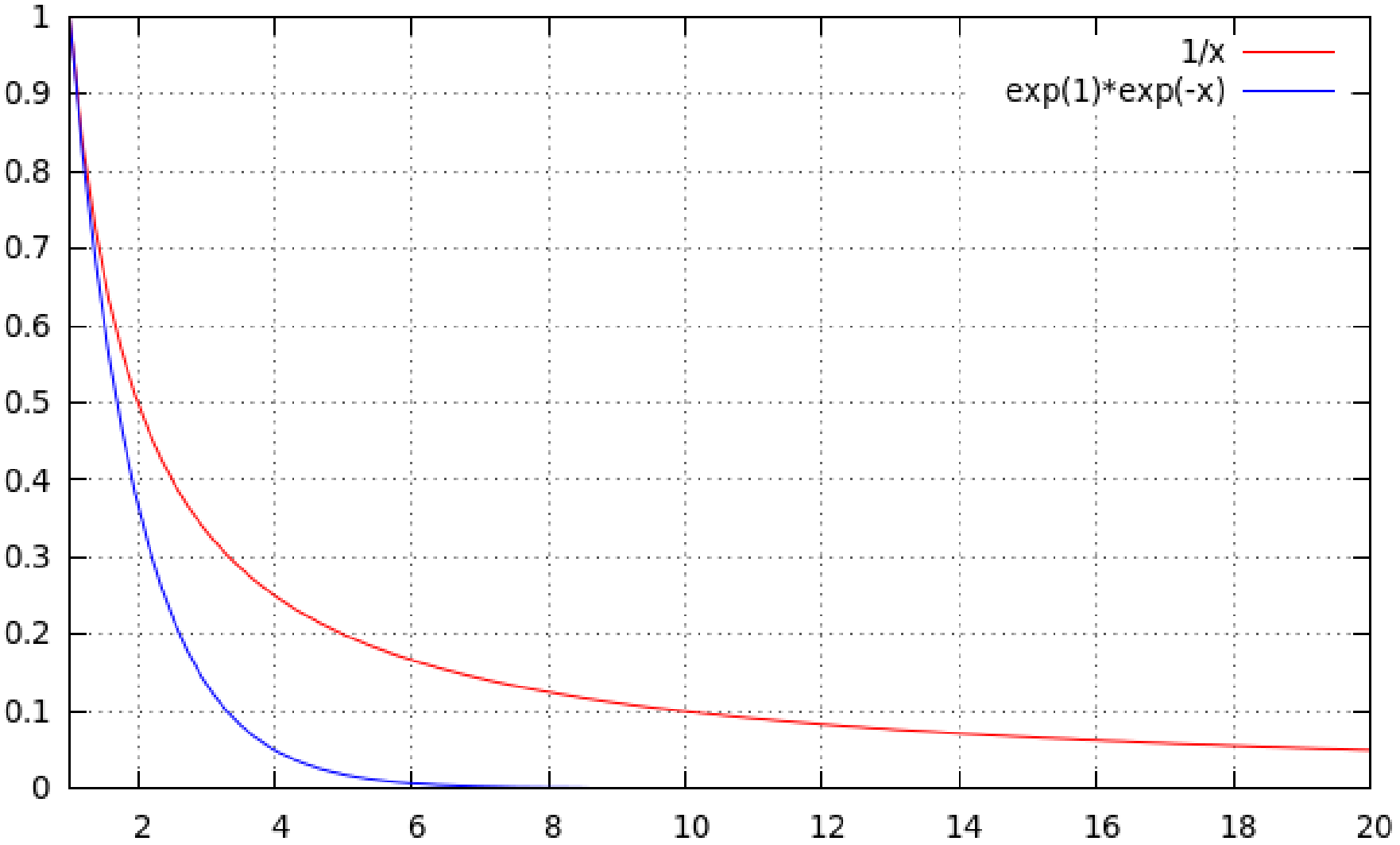
- Varför är produkter osäkra även om kunderna vill ha säkerhet?
  - Akerlof 1970 – A market for lemons

Svenwerk – Lemons, flickr CC BY-NC-ND 2.0

Termodynamik

- Hur växer tillförlitligheten i programvara?
  - Vi kan anta fel är oberoende så beskriver upptäckta fel en Poissonprocess med exponentialfördelade feltider (MTBF)
  - $p=e^{-Et}$ (slh att **ett** fel är kvar efter tiden t)
  - Men summerar man över hela systemet så visar det sig att felen inte försvinner som $p=e^{-Et}$ utan som K/t (Termodynamik)
  - Så MTBF blir t/K, där k bara beror på hur bra programvaran var från början
    - Så börjar man inte bra så kommer man inte ner till “noll”.

Termodynamik



Termodynamik

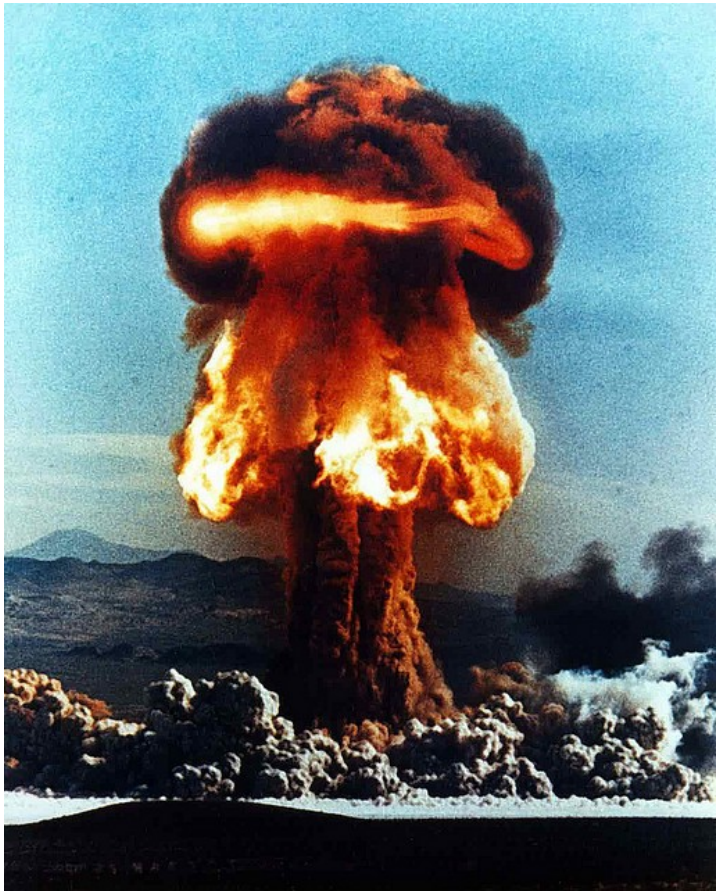
- Så behöver du en MTBF på 100k timmar så måste du test i 100k timmar
- Fel i programvara överlever så länge som möjligt – Murpy's lag
  - Testning hittar så få fel som ”möjligt”
  - Det är bättre att låta flera olika (typer) av människor/tester dela på tillgänglig tid än att låta en typ köra längre

Försvar eller anfall?

- WWI: Kulspruta och taggtråd – försvar
- Kalla kriget: Strategiska kärnvapen – anfall

James Morley, Cpl Thomas Morley, 34th Battalion Machine Gun Corps,

flickr (CC BY-NC 2.0)



No copyright, US Govt work

Försvar eller anfall

- Säg 10 fel per 1000 rader kod
  - Linux 0.17, kommersiell kod 10-20-30, Rymdfärjan 0.11 men $1000 per rad
  - En på tio är ett säkerhetsproblem i Linuxkärnan
- Windows ungefär **50** miljoner rader kod => säg **50000** säkerhetsfel kvar
- Säg att Adam Angripare kan testa **1000** timmar/år
- Säg att Filip Försvarare kan testa **1** miljon timmar/år
- Efter ett år så har (säg) Adam hittat **ett** fel, då har Filip hittat **1000**, men slh att Filip har hittat Adams fel är bara **2%**

Anfall

- Detta Går inte komma ifrån om det bara finns tillräckligt med fel...
- Mot programvara så vill du vara i angreppsbranschen

Taktikanpassning

- Andra metoder än försöka förhindra angrepp
  - Halme; preemption, deterrence, deflection, detection, countermeasures
- Därför forskar jag själv inom just
  - Detection/övervakning
    - Anomalidetektion för havsövervakning
  - Forensics

Slut

- Frågor, kommentarer?

- Mer läsning/källor:
  - *Murphy's law, the fitness of evolving species, and the limits of software Reliability*, Robert M. Brady, et.al.
  - *Why information security is hard-an economic perspective,* Ross Anderson
  - *Security Engineering*, 2nd ed. Ross Anderson (kap 7 & 22)