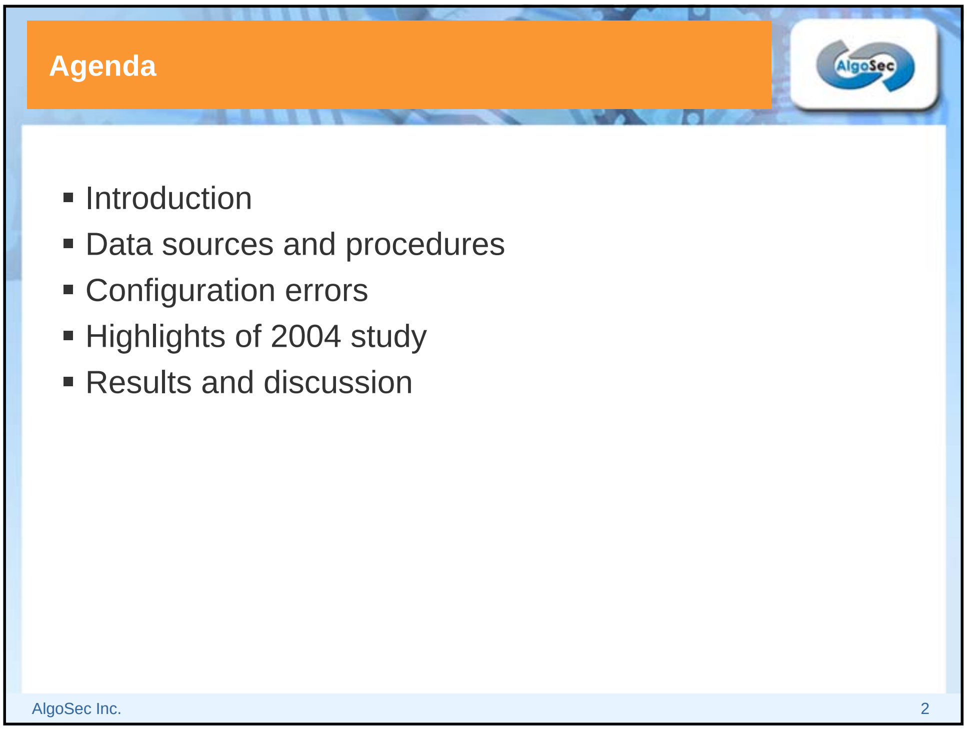
Firewall Configuration Errors Revisited

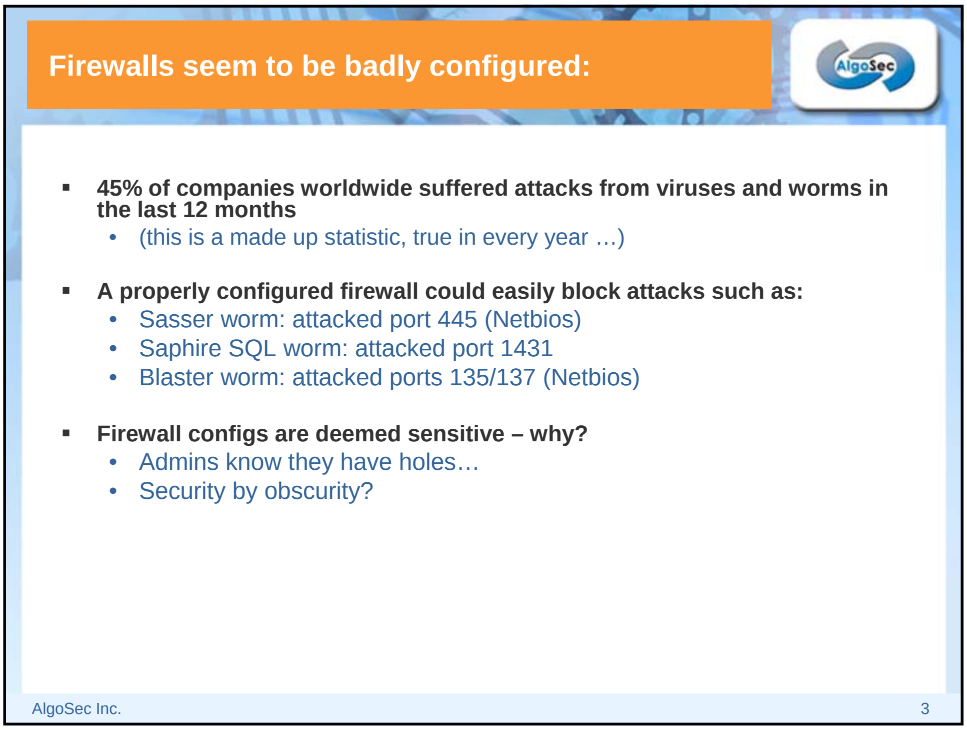
Avishai Wool

CTO & Co-Founder, AlgoSec
and
Prof., Tel Aviv University



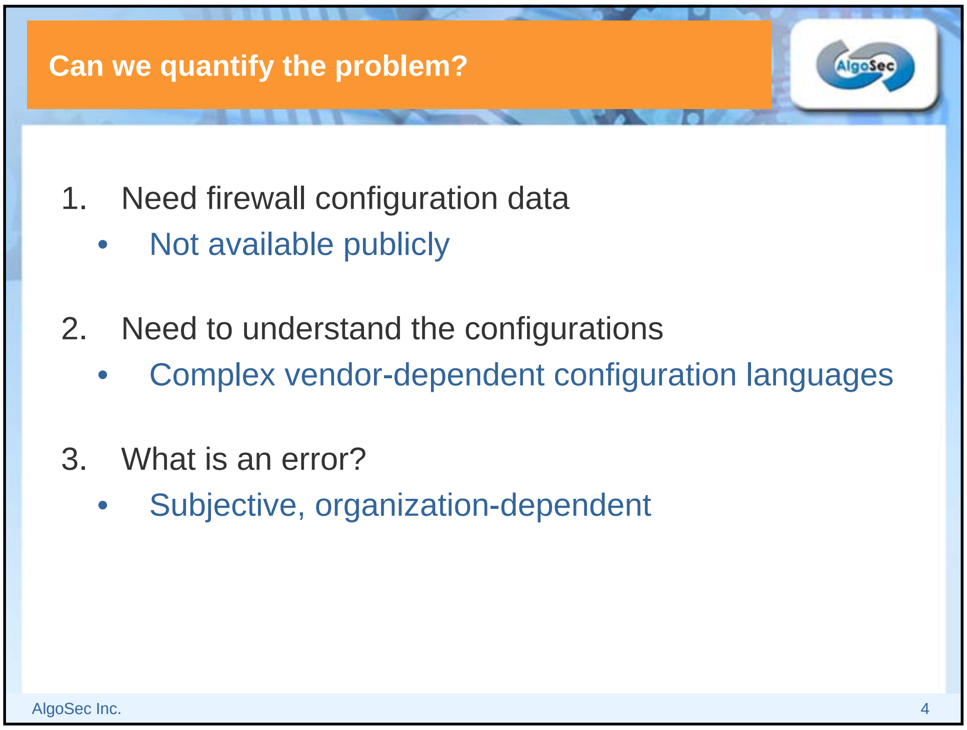
Agenda

- Introduction
- Data sources and procedures
- Configuration errors
- Highlights of 2004 study
- Results and discussion



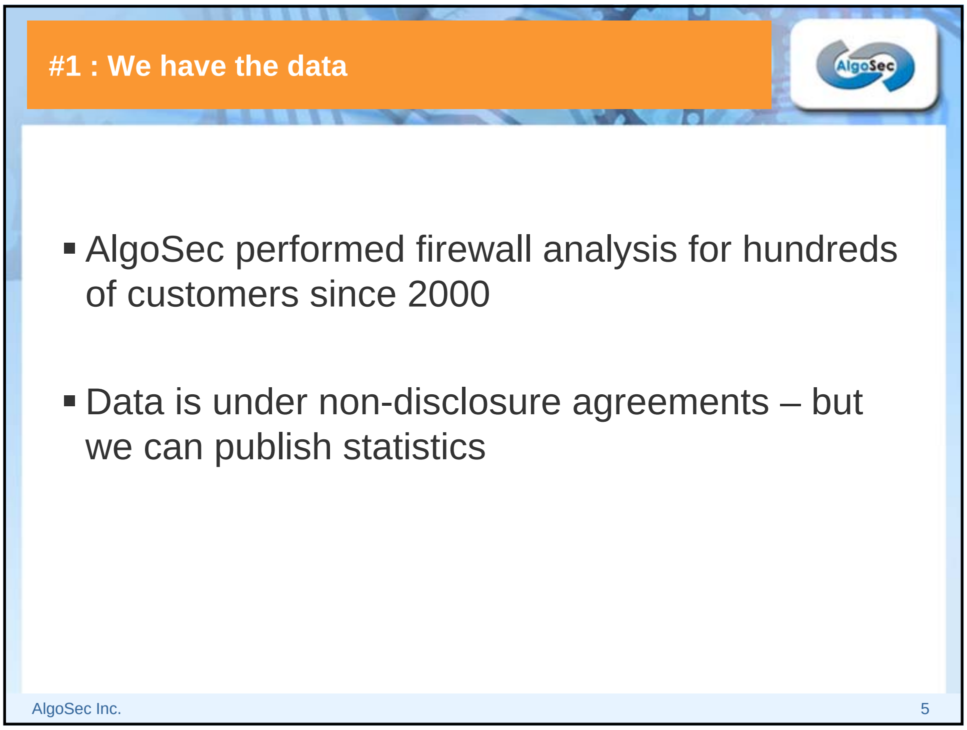
Firewalls seem to be badly configured:

- 45% of companies worldwide suffered attacks from viruses and worms in the last 12 months
  - (this is a made up statistic, true in every year …)
- A properly configured firewall could easily block attacks such as:
  - Sasser worm: attacked port 445 (Netbios)
  - Saphire SQL worm: attacked port 1431
  - Blaster worm: attacked ports 135/137 (Netbios)
- Firewall configs are deemed sensitive – why?
  - Admins know they have holes…
  - Security by obscurity?



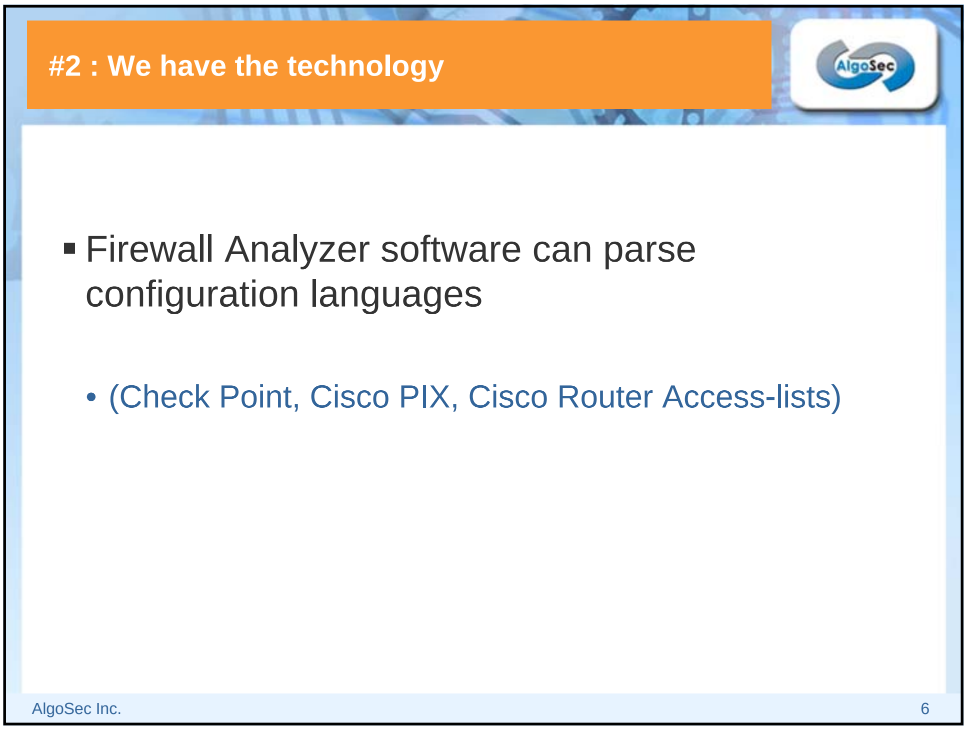
Can we quantify the problem?

1. Need firewall configuration data
   - Not available publicly
2. Need to understand the configurations
   - Complex vendor-dependent configuration languages
3. What is an error?
   - Subjective, organization-dependent



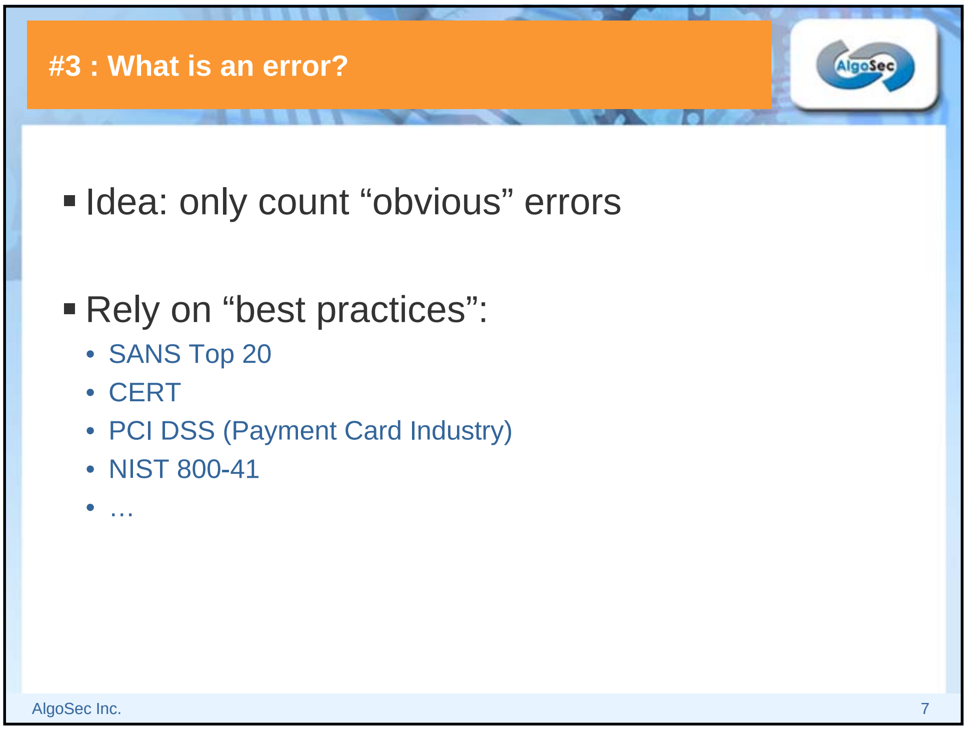
#1 : We have the data

- AlgoSec performed firewall analysis for hundreds of customers since 2000
- Data is under non-disclosure agreements – but we can publish statistics



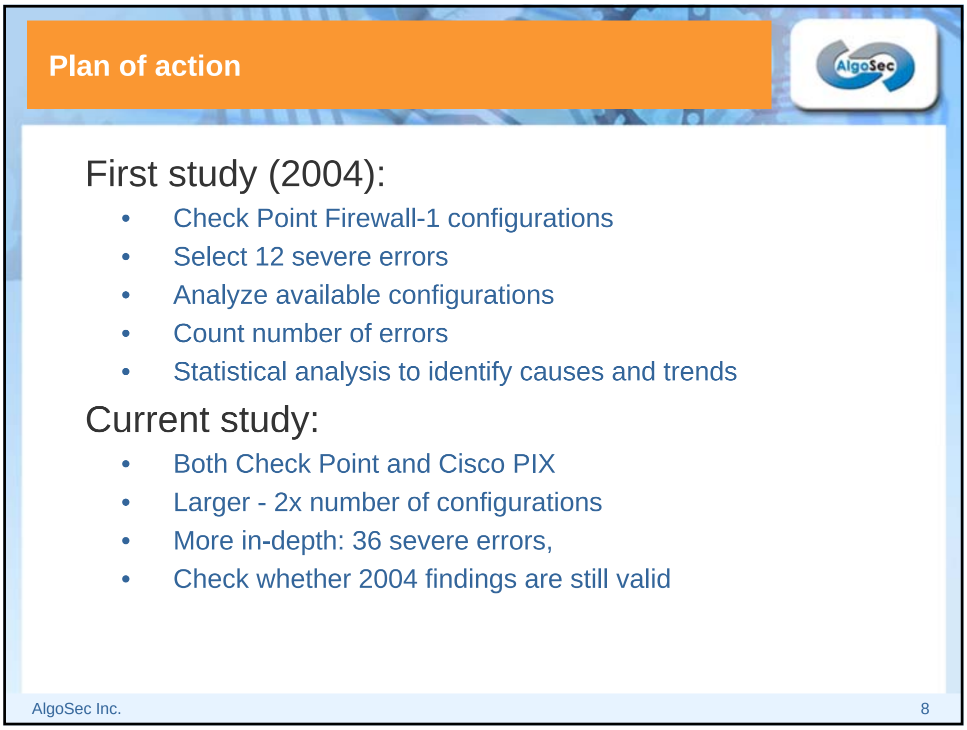
#2 : We have the technology

- Firewall Analyzer software can parse configuration languages
  - (Check Point, Cisco PIX, Cisco Router Access-lists)



#3 : What is an error?

- Idea: only count “obvious” errors
- Rely on “best practices”:
  - SANS Top 20
  - CERT
  - PCI DSS (Payment Card Industry)
  - NIST 800-41
  - …



Plan of action

First study (2004):

- Check Point Firewall-1 configurations
- Select 12 severe errors
- Analyze available configurations
- Count number of errors
- Statistical analysis to identify causes and trends

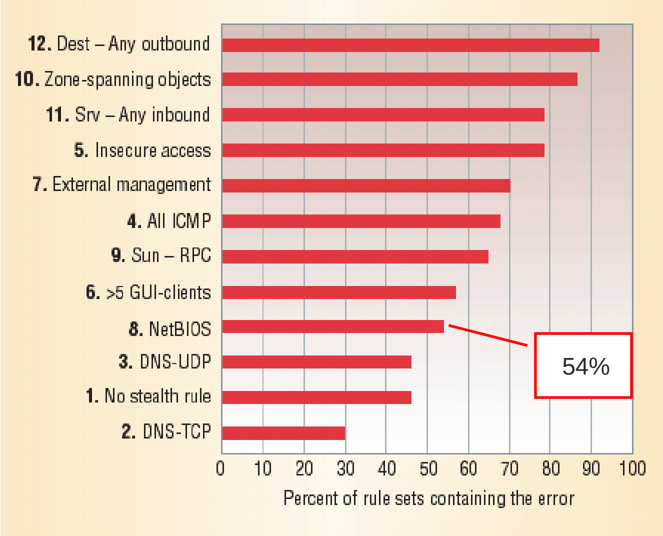
Current study:

- Both Check Point and Cisco PIX
- Larger - 2x number of configurations
- More in-depth: 36 severe errors,
- Check whether 2004 findings are still valid

Timeline of data collection

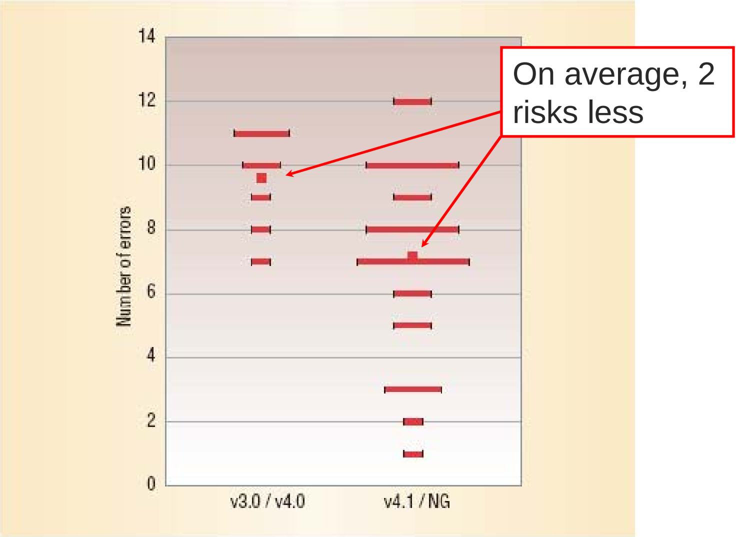
- Configuration files were collected between 2000-2005
- Check Point Firewall-1 versions:
  - 3.0, 4.0 – “end-of-life”
  - 4.1 – was still supported
  - NG – released in 2001, minor versions FP3, R54, R55
- Cisco PIX
  - PIX versions 4.x, 5.x, 6.x, 7.0

Highlights of the 2004 study

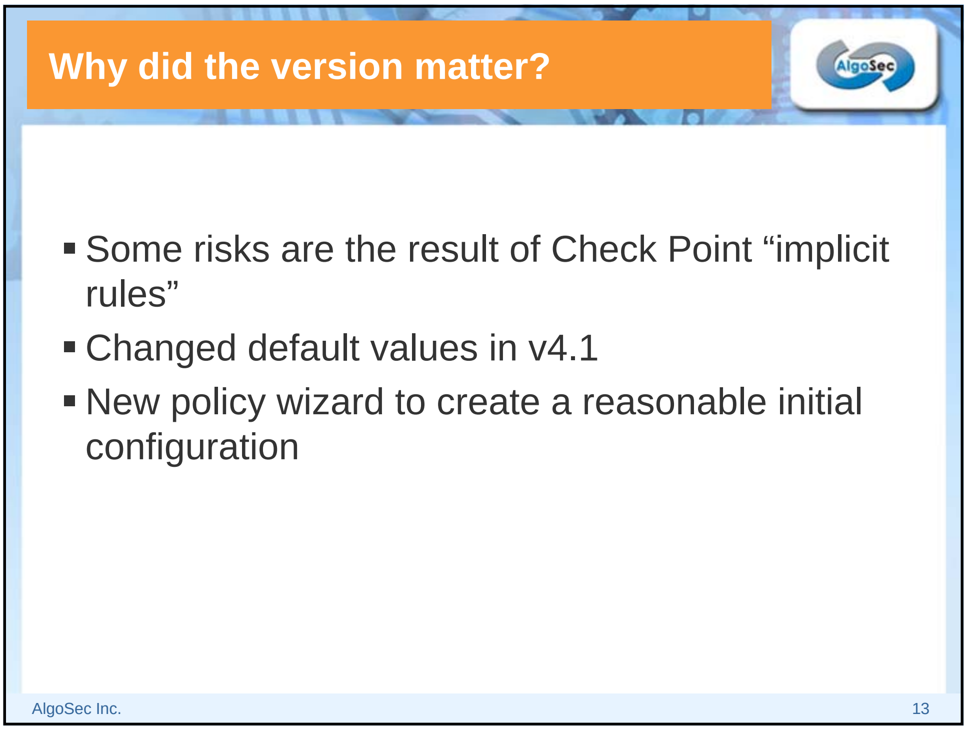


54%

Firewall-1 version helps

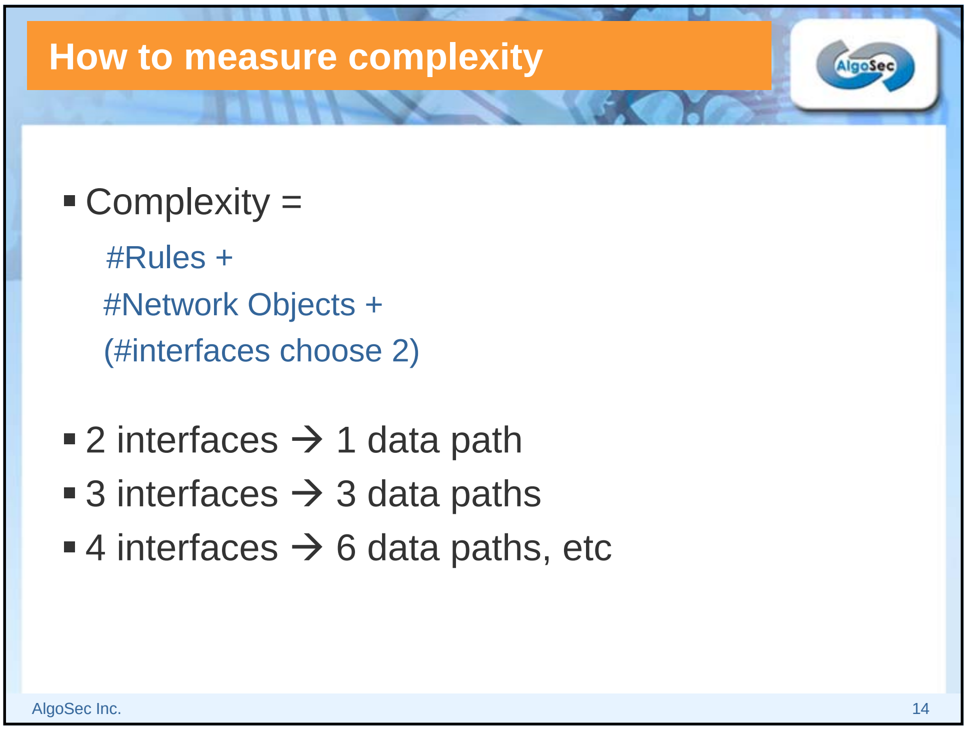


On average, 2 risks less



Why did the version matter?

- Some risks are the result of Check Point “implicit rules”
- Changed default values in v4.1
- New policy wizard to create a reasonable initial configuration



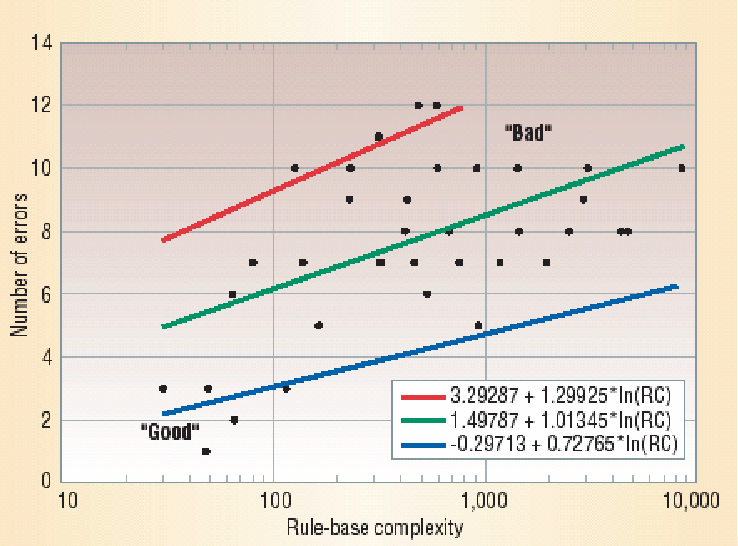
How to measure complexity

- Complexity =

  #Rules +
  #Network Objects +
  (#interfaces choose 2)

- 2 interfaces → 1 data path
- 3 interfaces → 3 data paths
- 4 interfaces → 6 data paths, etc

Small is Beautiful



Current Results

Why should anything change?

- Regulation and Compliance:
  - Sarbanes-Oxley
  - Payment Card Industry (PCI DSS)
  - NIST 800-41
  - …
- Different vendors – different issues?
- New software versions – continue the trend?

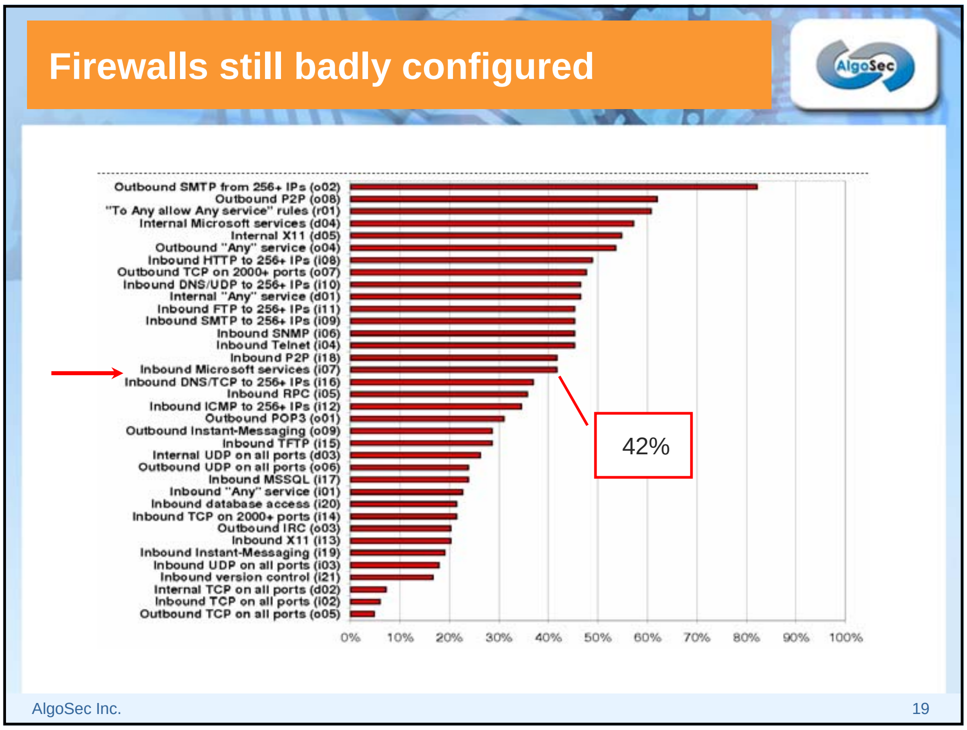
Differences from 2004 report

- Both Check Point and PIX
- 2x configurations tested
- Newer software versions

- Vendor-neutral risk items
  - 8 of 12 properties in 2004 study were specific to Check Point

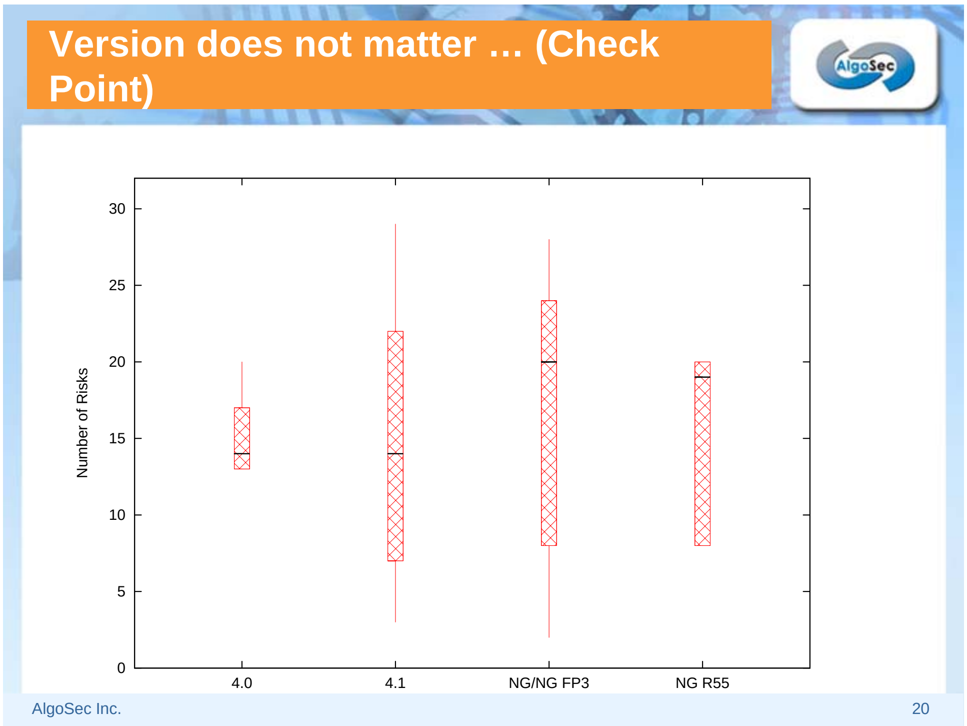
→Pick a new set of 36 risk items

→Inbound / Outbound / Internal traffic



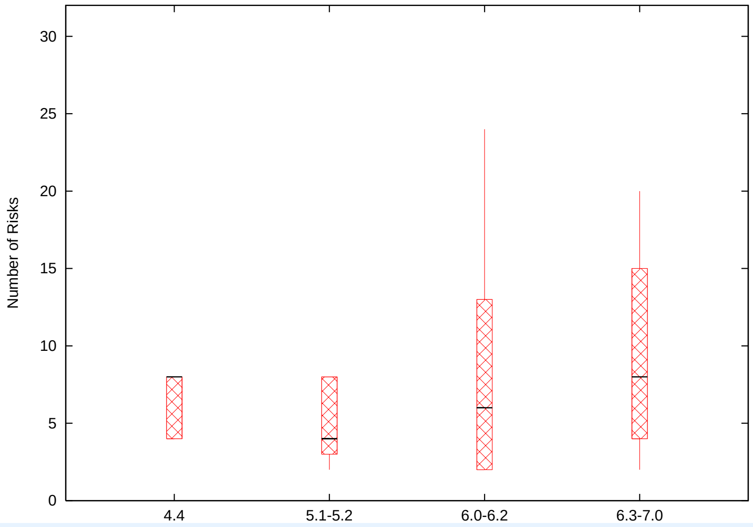
Firewalls still badly configured

42%



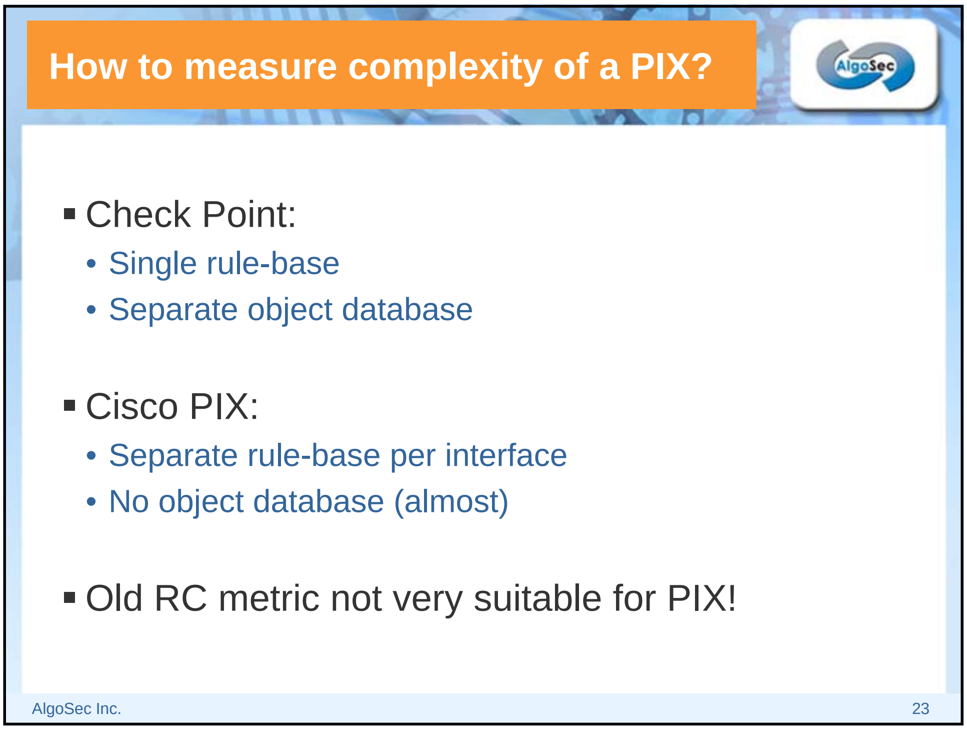
Version does not matter … (Check Point)

Version does not matter … (PIX)



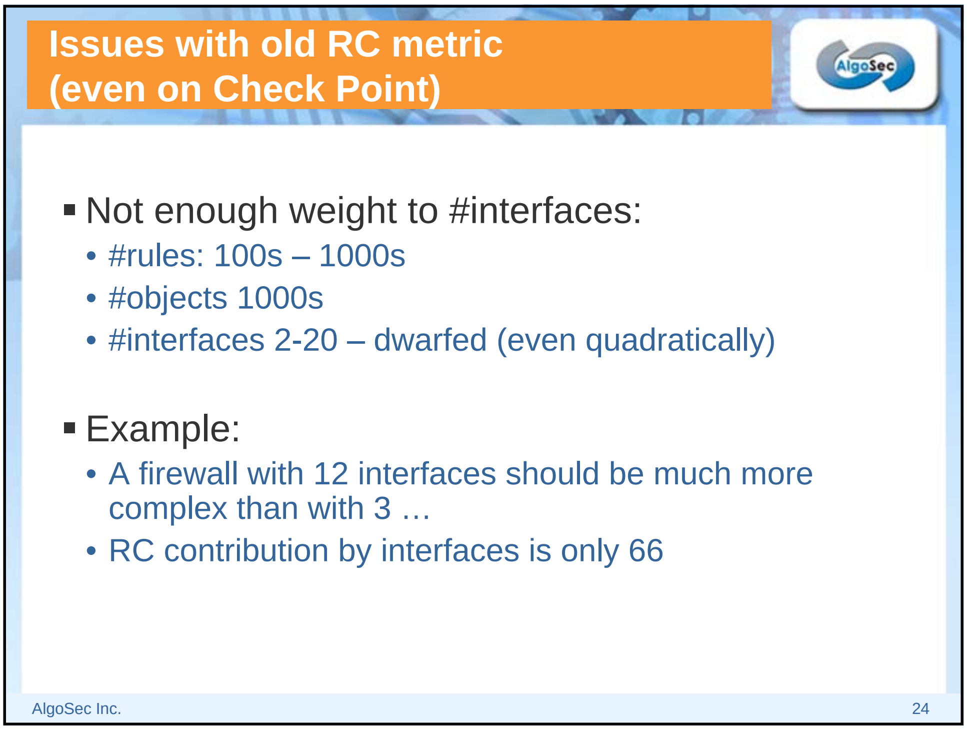
Why?

- Vendor-neutral risks are controlled by **basic filtering** functionality
- Basic filtering controlled by **explicit user-defined rules**, rather than “check boxes” with vendor “know-how” (??)
- Neither vendor has changed the basic filtering capabilities in years (and it’s unlikely that they will)



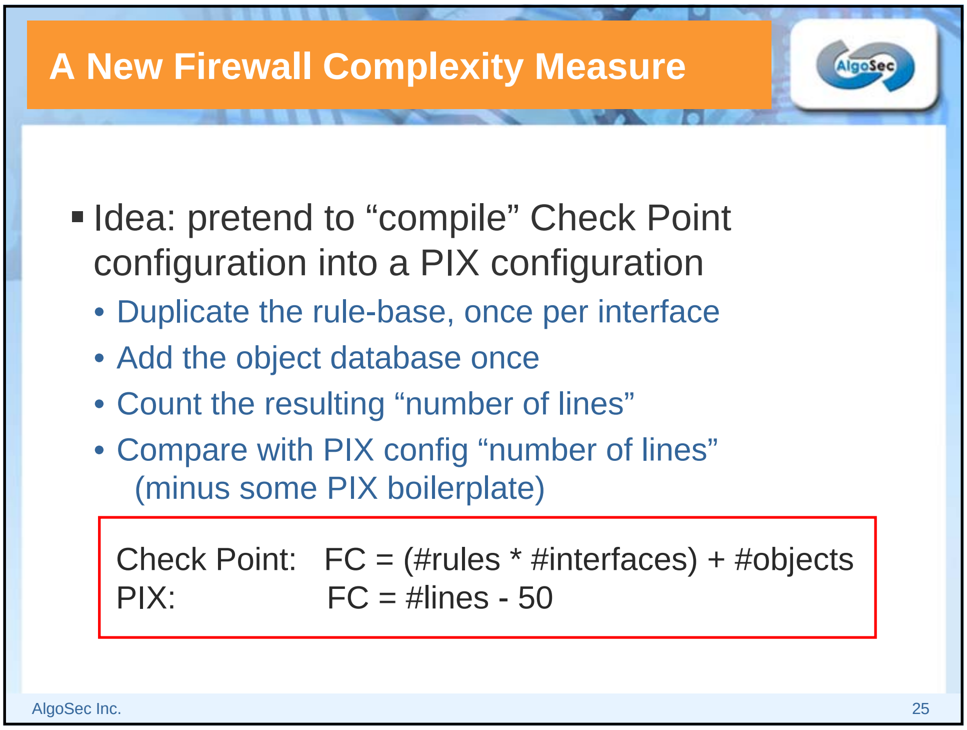
How to measure complexity of a PIX?

- Check Point:
  - Single rule-base
  - Separate object database
- Cisco PIX:
  - Separate rule-base per interface
  - No object database (almost)
- Old RC metric not very suitable for PIX!



Issues with old RC metric (even on Check Point)

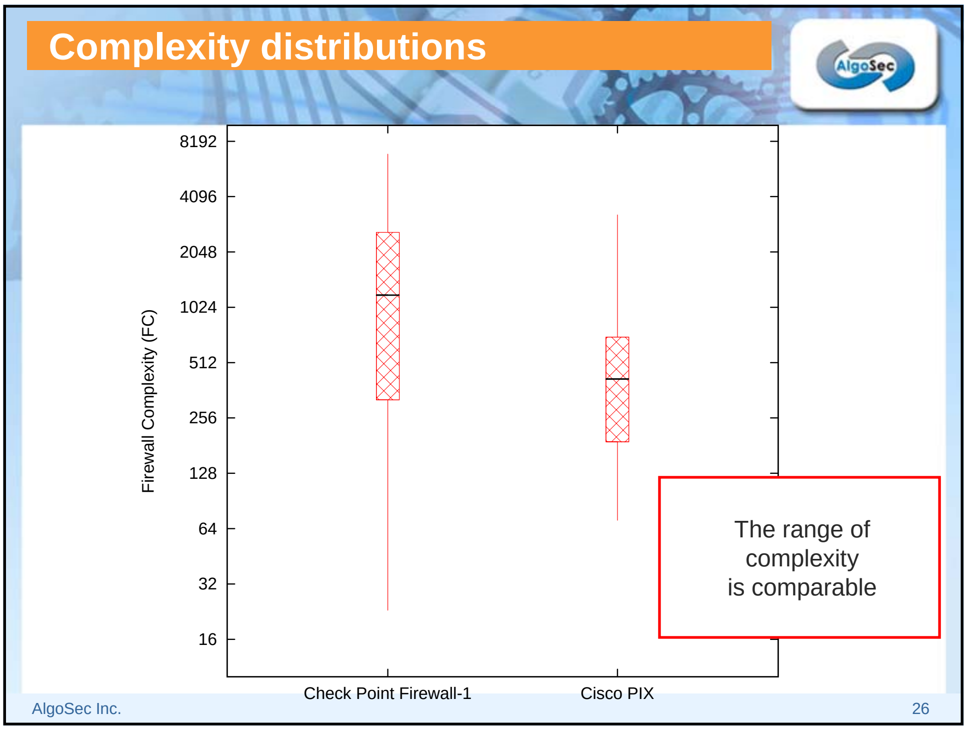
- Not enough weight to #interfaces:
  - #rules: 100s – 1000s
  - #objects 1000s
  - #interfaces 2-20 – dwarfed (even quadratically)
- Example:
  - A firewall with 12 interfaces should be much more complex than with 3 …
  - RC contribution by interfaces is only 66



A New Firewall Complexity Measure

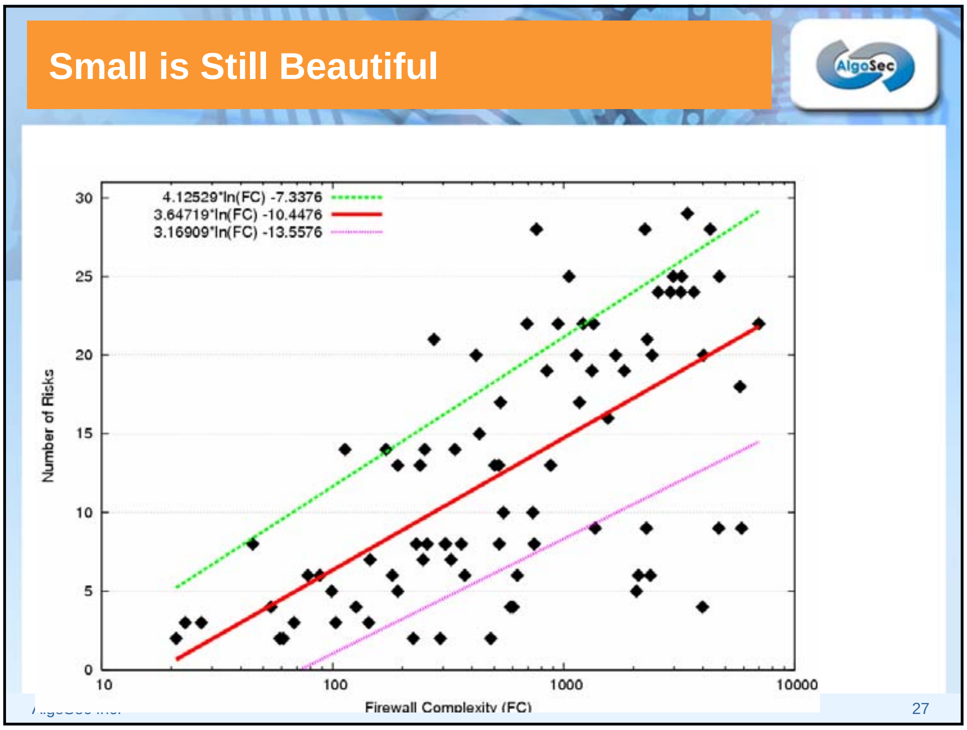
- Idea: pretend to “compile” Check Point configuration into a PIX configuration
  - Duplicate the rule-base, once per interface
  - Add the object database once
  - Count the resulting “number of lines”
  - Compare with PIX config “number of lines” (minus some PIX boilerplate)

Check Point: FC = (#rules * #interfaces) + #objects
PIX: FC = #lines - 50

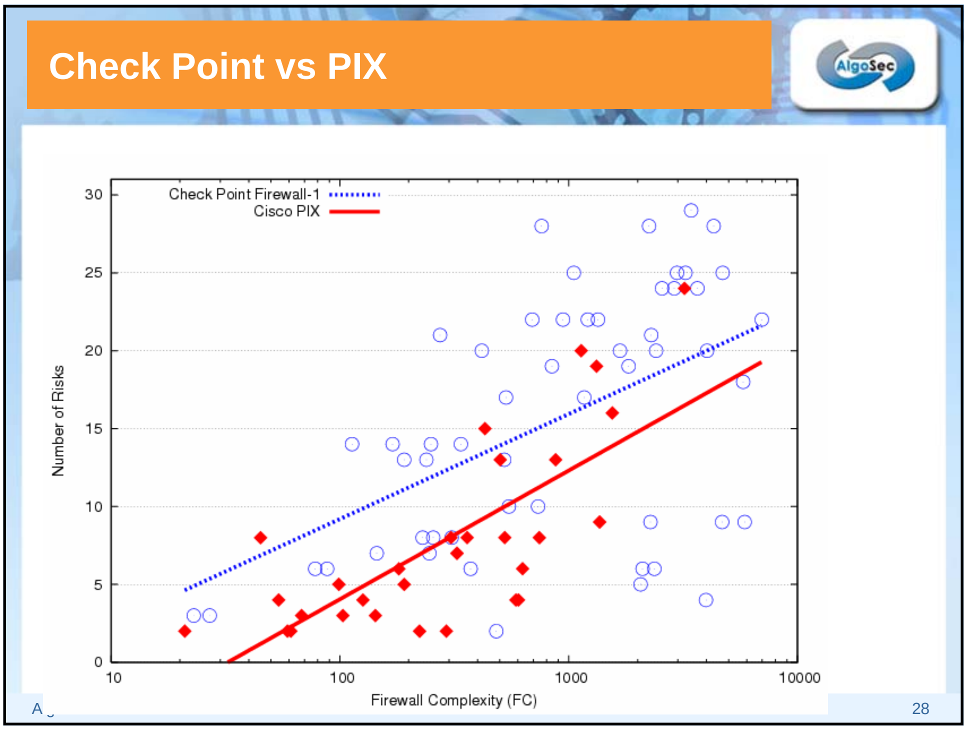


Complexity distributions

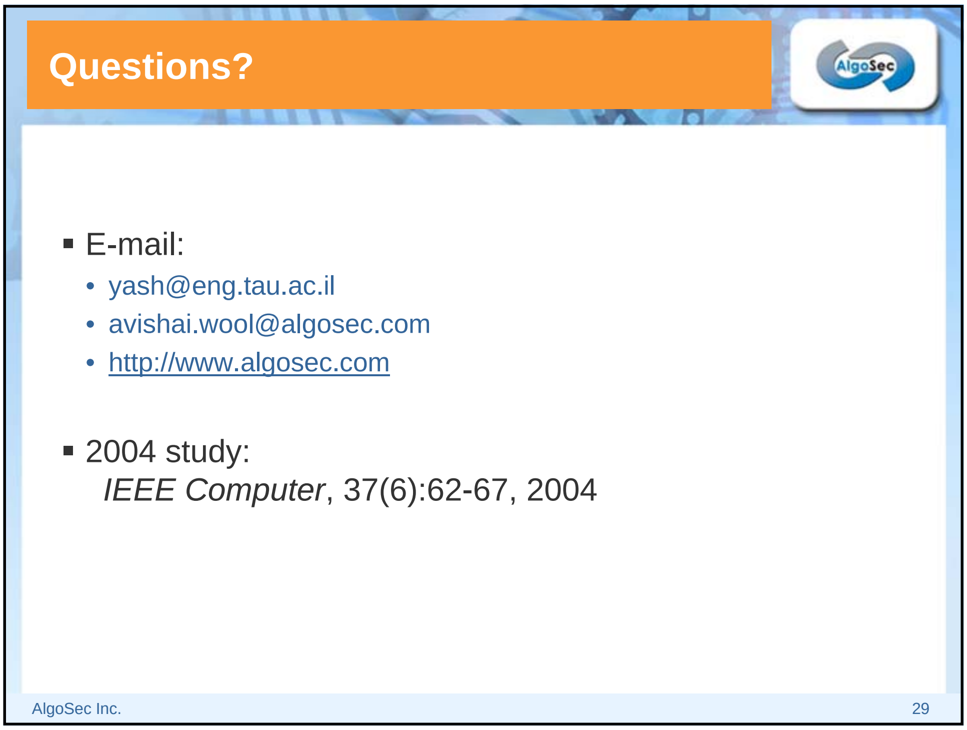
The range of complexity is comparable



Small is Still Beautiful



Check Point vs PIX



Questions?

- E-mail:
  - yash@eng.tau.ac.il
  - avishai.wool@algosec.com
  - http://www.algosec.com
- 2004 study:
  IEEE Computer, 37(6):62-67, 2004