Image-Based Kernel Fingerprinting

By

Vassil Roussev, Irfan Ahmed and Thomas Sires

Presented At

The Digital Forensic Research Conference

DFRWS 2014 USA Denver, CO (Aug 3rd - 6th) dfrws'14: aug 3-7, 2014

image-based kernel fingerprinting

Vassil Roussev, Irfan Ahmed, Thomas Sires

University of New Orleans

vassil@roussev.net

dfrws'14: aug 3-7, 2014

image-based kernel fingerprinting

a 'Malkovich, Malkovich' story

Vassil Roussev, Irfan Ahmed, Thomas Sires

University of New Orleans

vassil@roussev.net

why are we here?

GIVEN A RAM DUMP, OR VM SNAPSHOT

- find the exact kernel version
  - so that we can do proper memory analysis

WINDOWS → NOT A PROBLEM

LINUX?

- thousands of 'stock' kernels (ubuntu, red hat, ...)
- custom kernels
- different architectures: x86, amd64, arm5/6/...

BOTTOM LINE:

- we need something robust that requires no reversing

first thought:
if you have a hammer → find nails

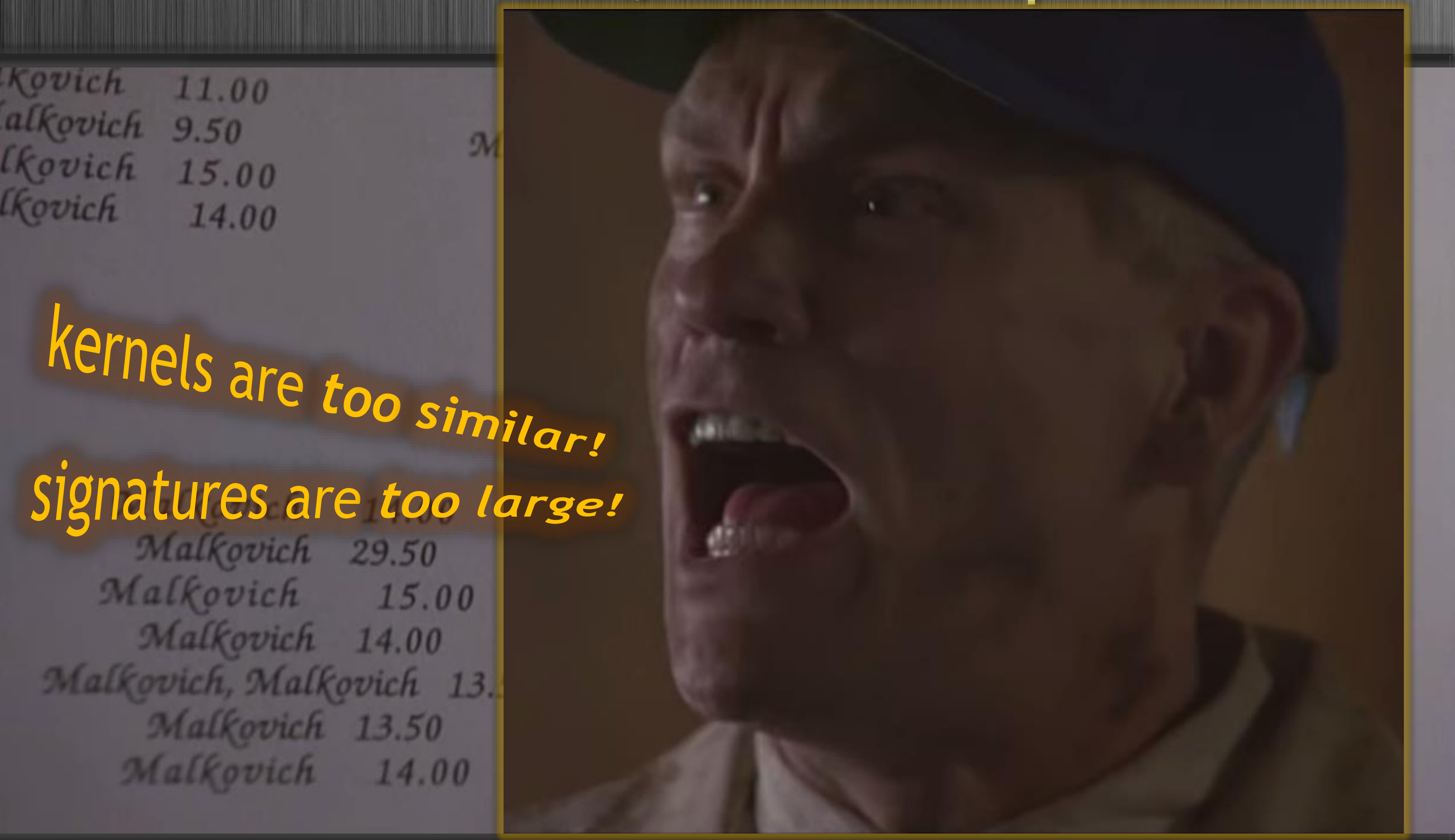
quick solution sketch

SDHASH CAN CORRELATE ANY TWO BLOBS BASED ON COMMONALITY:

```
# sdhash vmlinux-* > kernels.sdbf
# sdhash server.vmem > ram.sdbf
# sdhash –c kernels.sdbf ram.sdbf
```

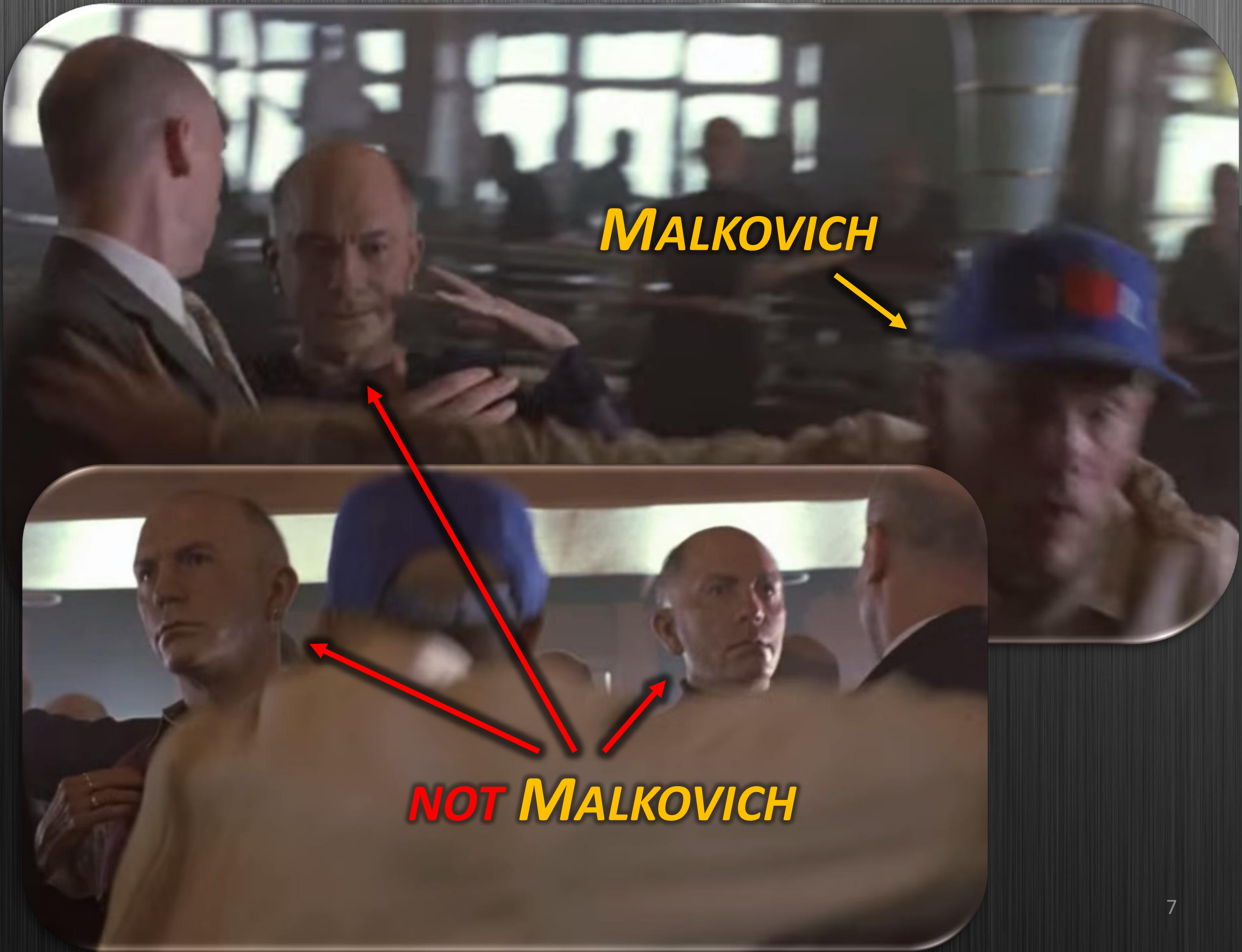
PROBLEM SOLVED?

how is identifying a ***linux*** kernel a *MALKOVICH, MALKOVICH* problem?



kernels are **too similar!**

signatures are **too large!**



Malkovich

not Malkovich

second thought: remove rough commonality, ***then*** apply hammer

sdkernel

1. Obtain a set of kernel images that are of interest
2. Split the images into 4K/16-blocks &
   # eliminate all repetitive blocks
3. Create a sample of the blocks of the desired size from each image
   # 100x4K, or 25x16K
4. Create a block-based sdhash similarity digest

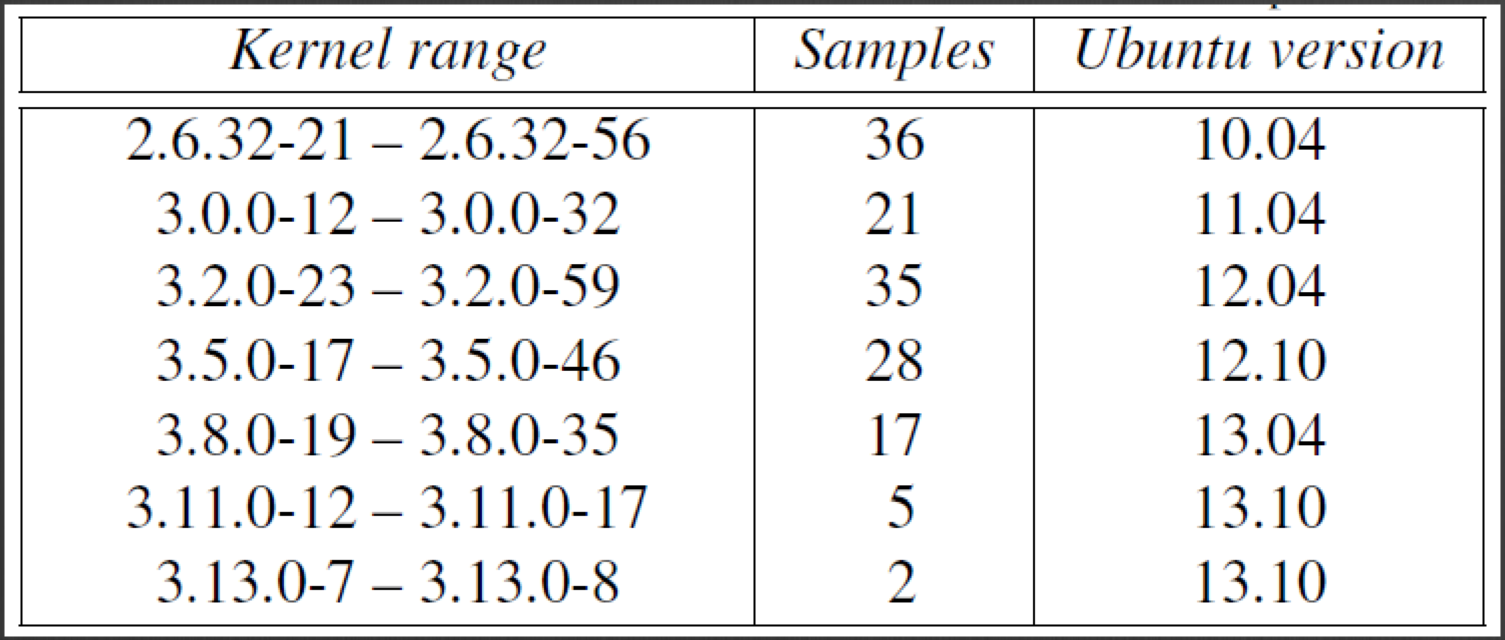
how unique are *linux* kernels?

HTTP://SECURITY.UBUNTU.COM/UBUNTU/POOL/MAIN/L/LINUX/:

# 943 packages

# 300 *generic* (150 32-/64-bit each)

# 288 *unique* kernels (144 each)



Kernel range	*Samples*	*Ubuntu version*
2.6.32-21 – 2.6.32-56	36	10.04
3.0.0-12 – 3.0.0-32	21	11.04
3.2.0-23 – 3.2.0-59	35	12.04
3.5.0-17 – 3.5.0-46	28	12.10
3.8.0-19 – 3.8.0-35	17	13.04
3.11.0-12 – 3.11.0-17	5	13.10
3.13.0-7 – 3.13.0-8	2	13.10

measuring uniqueness

SLICE KERNEL IMAGE INTO 4K BLOCKS

# (rationale: it's a *ramfs* image)

(CRYPTO-) HASH

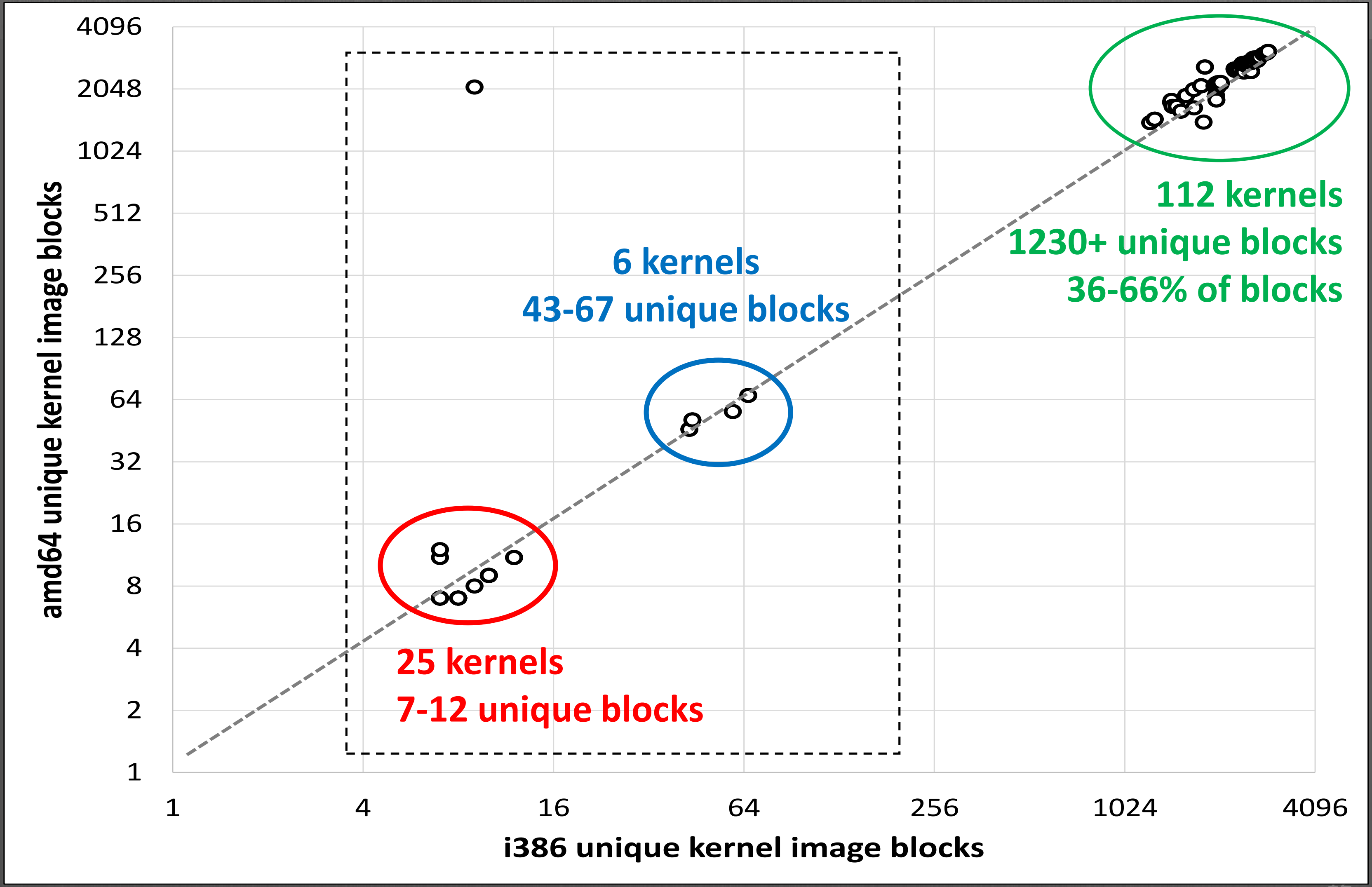
REMOVE REPETITIVE BLOCKS

# results in *signature base*

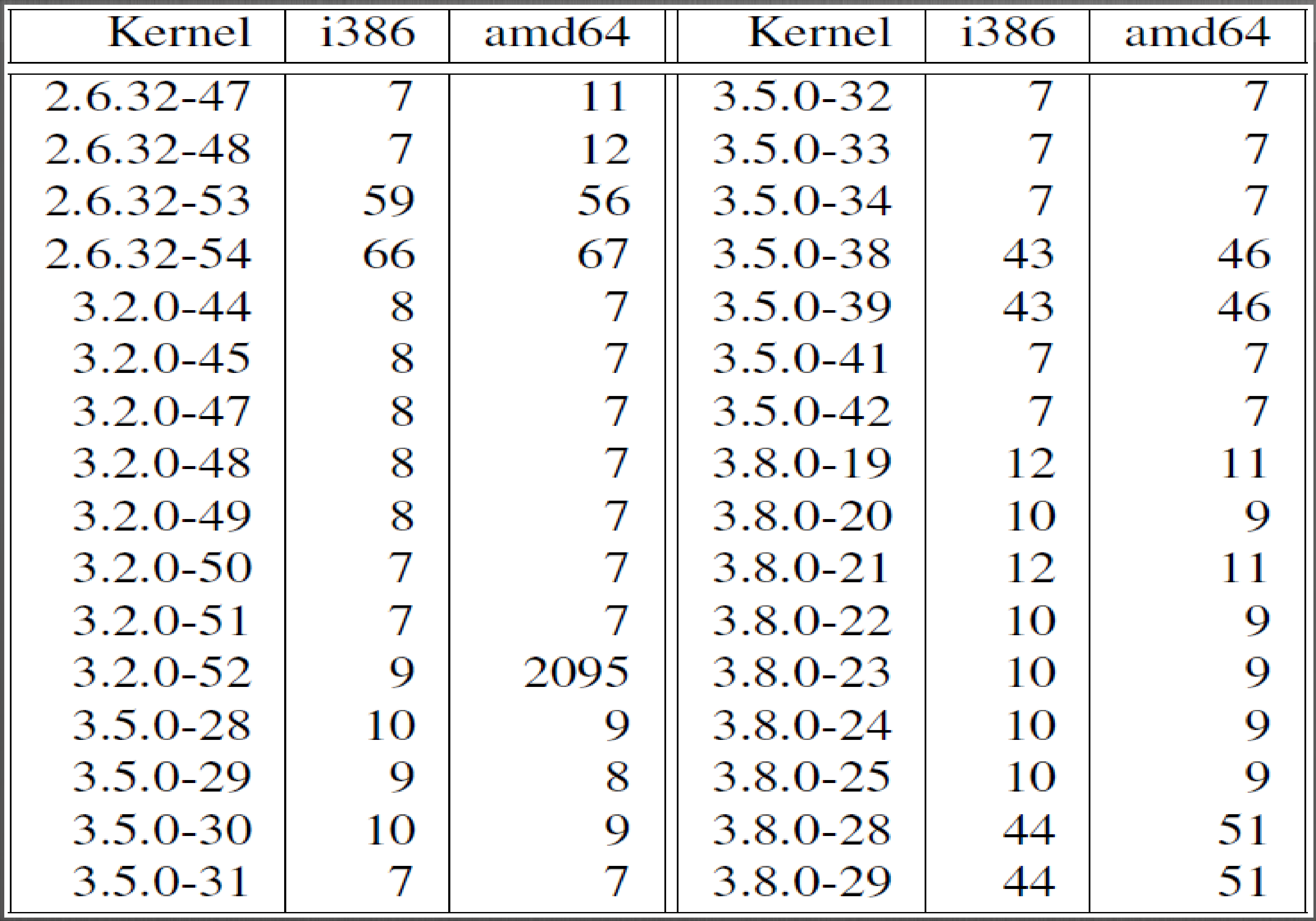
COUNT UNIQUE BLOCKS AS A ROUGH MEASURE OF UNIQUENESS

# any shared block is useless as part of a signature

uniqueness: i386 vs amd64 kernels



unique blocks of select kernels



Kernel	i386	amd64	Kernel	i386	amd64
2.6.32-47	7	11	3.5.0-32	7	7
2.6.32-48	7	12	3.5.0-33	7	7
2.6.32-53	59	56	3.5.0-34	7	7
2.6.32-54	66	67	3.5.0-38	43	46
3.2.0-44	8	7	3.5.0-39	43	46
3.2.0-45	8	7	3.5.0-41	7	7
3.2.0-47	8	7	3.5.0-42	7	7
3.2.0-48	8	7	3.8.0-19	12	11
3.2.0-49	8	7	3.8.0-20	10	9
3.2.0-50	7	7	3.8.0-21	12	11
3.2.0-51	7	7	3.8.0-22	10	9
3.2.0-52	9	2095	3.8.0-23	10	9
3.5.0-28	10	9	3.8.0-24	10	9
3.5.0-29	9	8	3.8.0-25	10	9
3.5.0-30	10	9	3.8.0-28	44	51
3.5.0-31	7	7	3.8.0-29	44	51

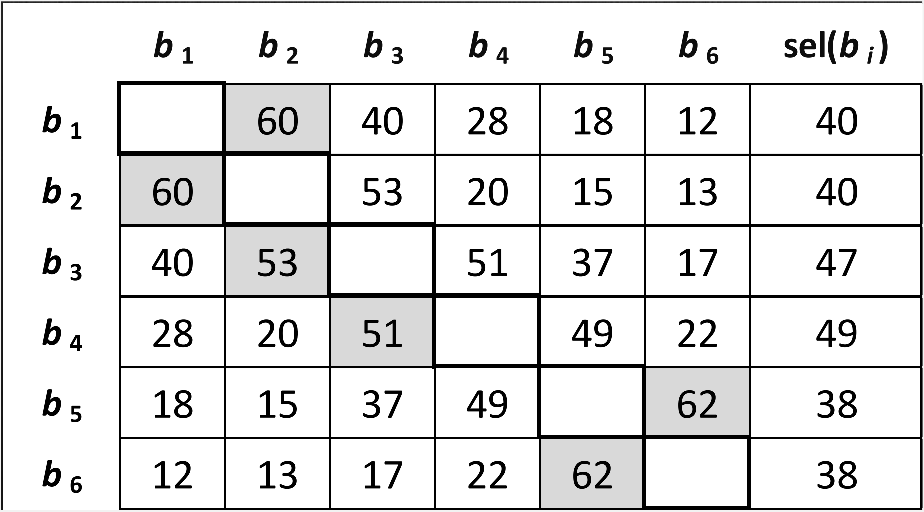
measuring selectivity

GIVEN SIGNATURE BASES $b_1, \ldots, b_n$ ***SELECTIVITY*** *IS:*

$$sel(b_i) = 100 - \max\left(sdhash_{4k}(b_i, b_j) : 1 \leq j \leq n, j \neq i\right)$$

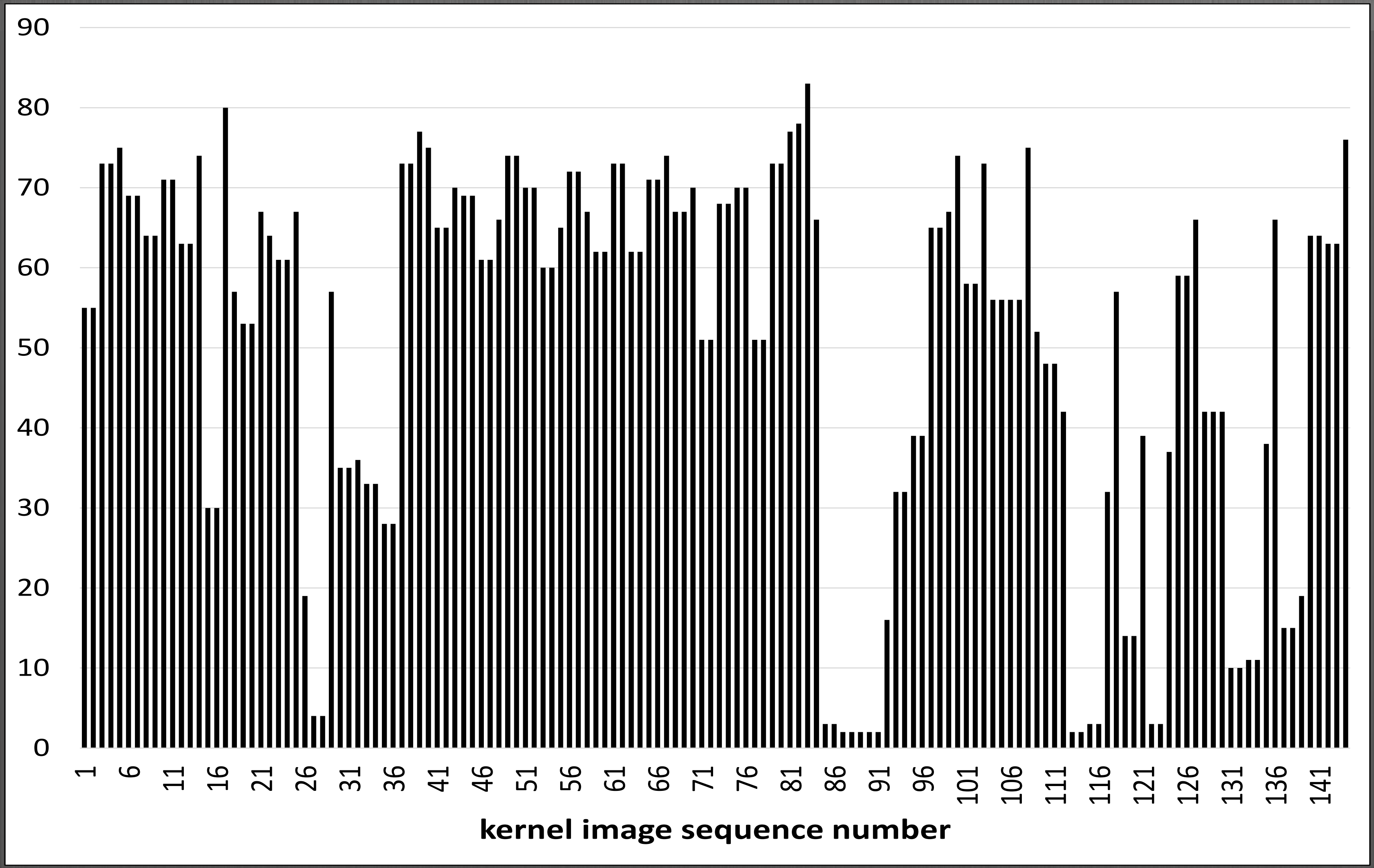
RATIONALE

# shrink a confusion matrix row to a single number:

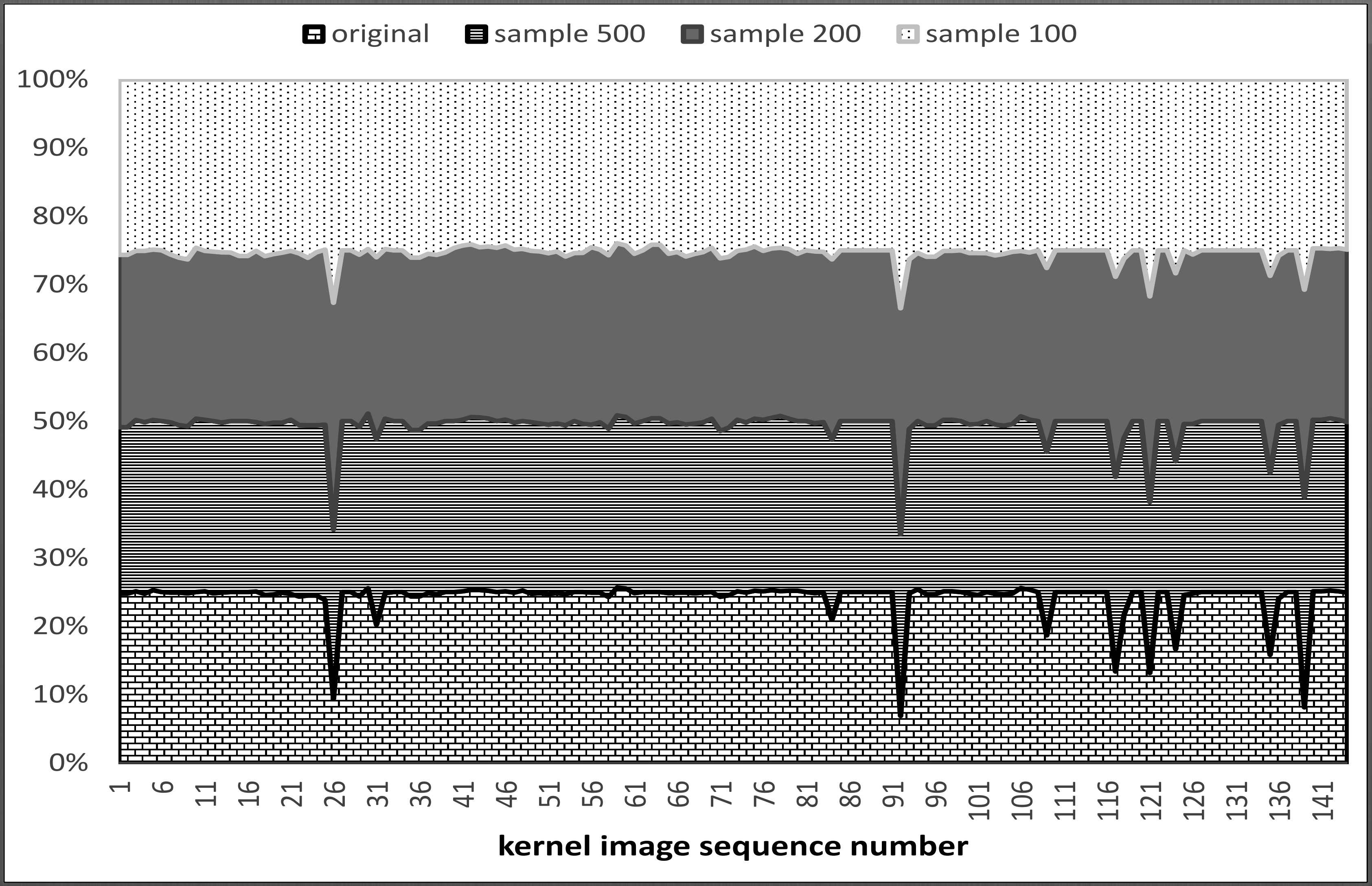


	b_1	b_2	b_3	b_4	b_5	b_6	sel(b_i)
b_1		60	40	28	18	12	40
b_2	60		53	20	15	13	40
b_3	40	53		51	37	17	47
b_4	28	20	51		49	22	49
b_5	18	15	37	49		62	38
b_6	12	13	17	22	62		38

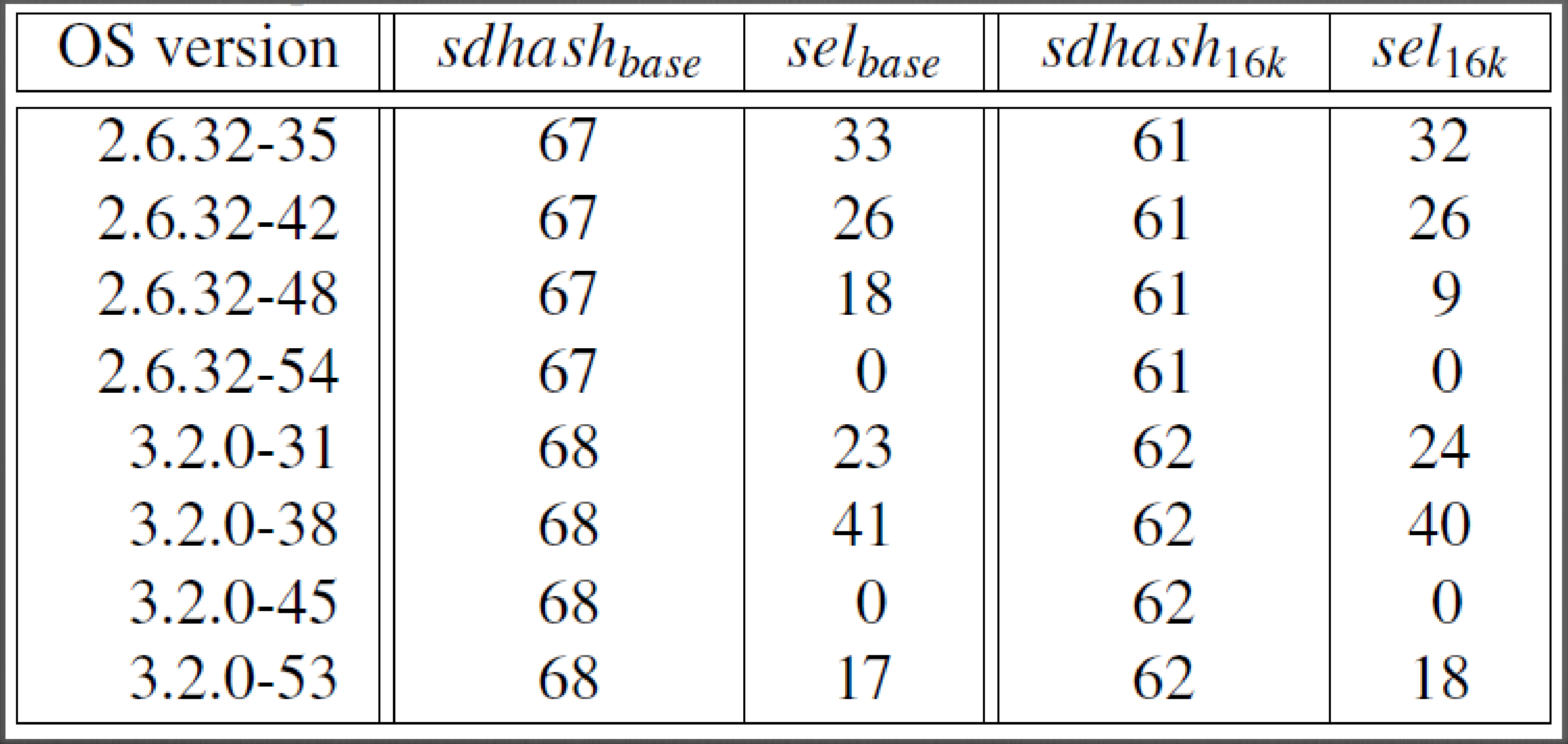
selectivity: amd64 signature bases



relative selectivity: original vs. sampled

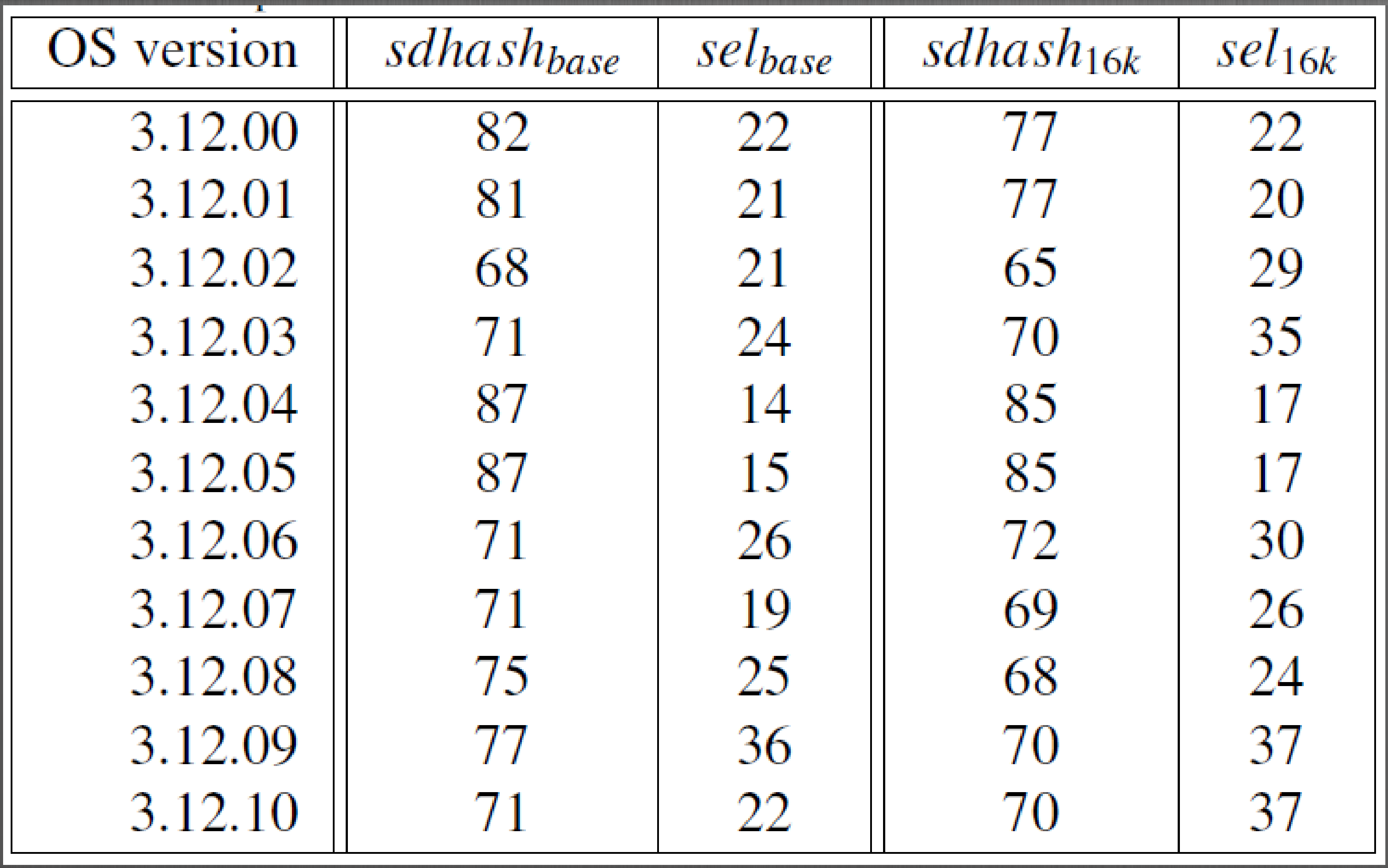


effects of sampling: amd64



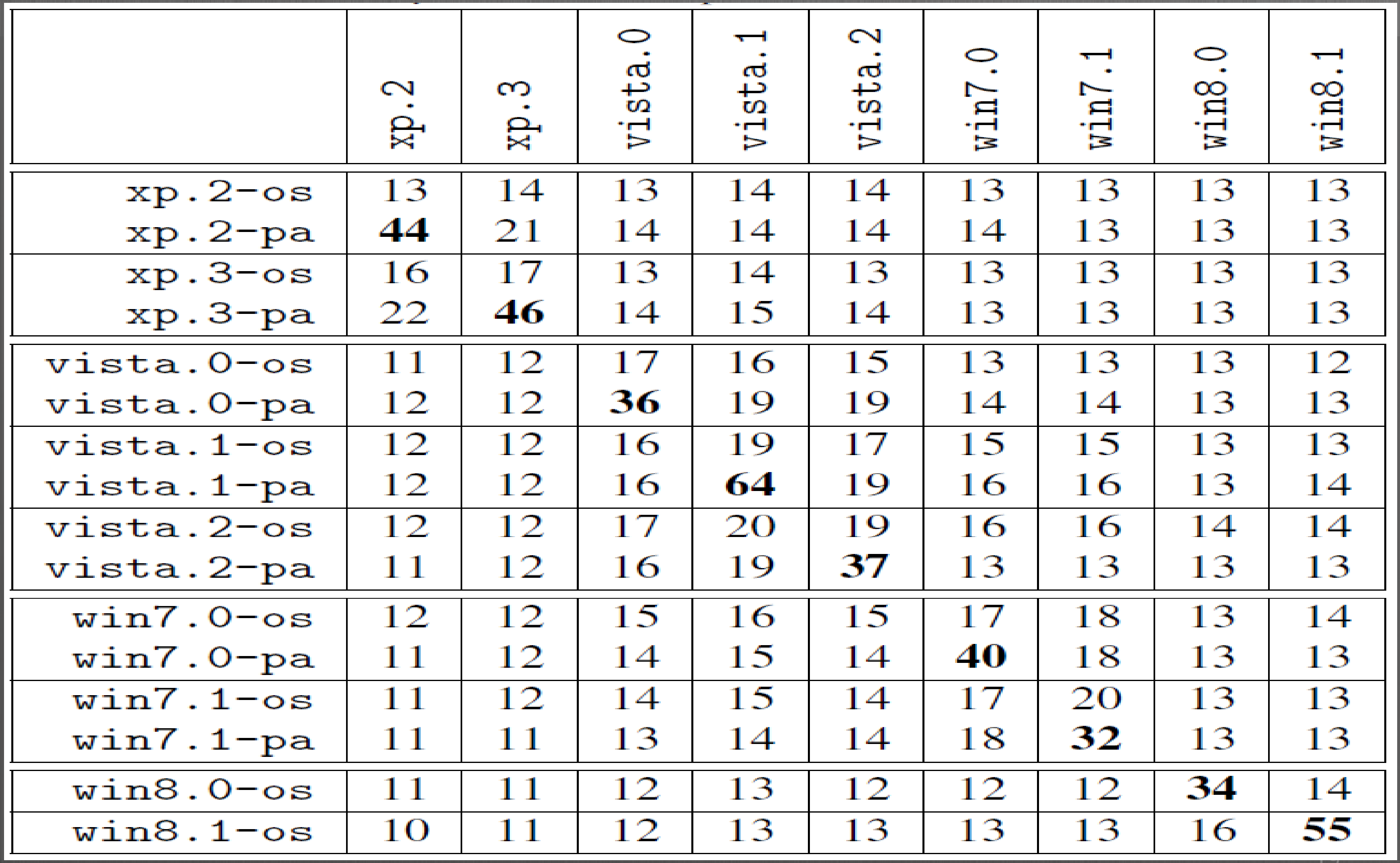
OS version	$sdhash_{base}$	sel_{base}	$sdhash_{16k}$	sel_{16k}
2.6.32-35	67	33	61	32
2.6.32-42	67	26	61	26
2.6.32-48	67	18	61	9
2.6.32-54	67	0	61	0
3.2.0-31	68	23	62	24
3.2.0-38	68	41	62	40
3.2.0-45	68	0	62	0
3.2.0-53	68	17	62	18

effects of sampling: arm5



OS version	$sdhash_{base}$	sel_{base}	$sdhash_{16k}$	sel_{16k}
3.12.00	82	22	77	22
3.12.01	81	21	77	20
3.12.02	68	21	65	29
3.12.03	71	24	70	35
3.12.04	87	14	85	17
3.12.05	87	15	85	17
3.12.06	71	26	72	30
3.12.07	71	19	69	26
3.12.08	75	25	68	24
3.12.09	77	36	70	37
3.12.10	71	22	70	37

effects of sampling: windows



	xp.2	xp.3	vista.0	vista.1	vista.2	win7.0	win7.1	win8.0	win8.1
xp.2-os	13	14	13	14	14	13	13	13	13
xp.2-pa	**44**	21	14	14	14	14	13	13	13
xp.3-os	16	17	13	14	13	13	13	13	13
xp.3-pa	22	**46**	14	15	14	13	13	13	13
vista.0-os	11	12	17	16	15	13	13	13	12
vista.0-pa	12	12	**36**	19	19	14	14	13	13
vista.1-os	12	12	16	19	17	15	15	13	13
vista.1-pa	12	12	16	**64**	19	16	16	13	14
vista.2-os	12	12	17	20	19	16	16	14	14
vista.2-pa	11	12	16	19	**37**	13	13	13	13
win7.0-os	12	12	15	16	15	17	18	13	14
win7.0-pa	11	12	14	15	14	**40**	18	13	13
win7.1-os	11	12	14	15	14	17	20	13	13
win7.1-pa	11	11	13	14	14	18	**32**	13	13
win8.0-os	11	11	12	13	12	12	12	**34**	14
win8.1-os	10	11	12	13	13	13	13	16	**55**

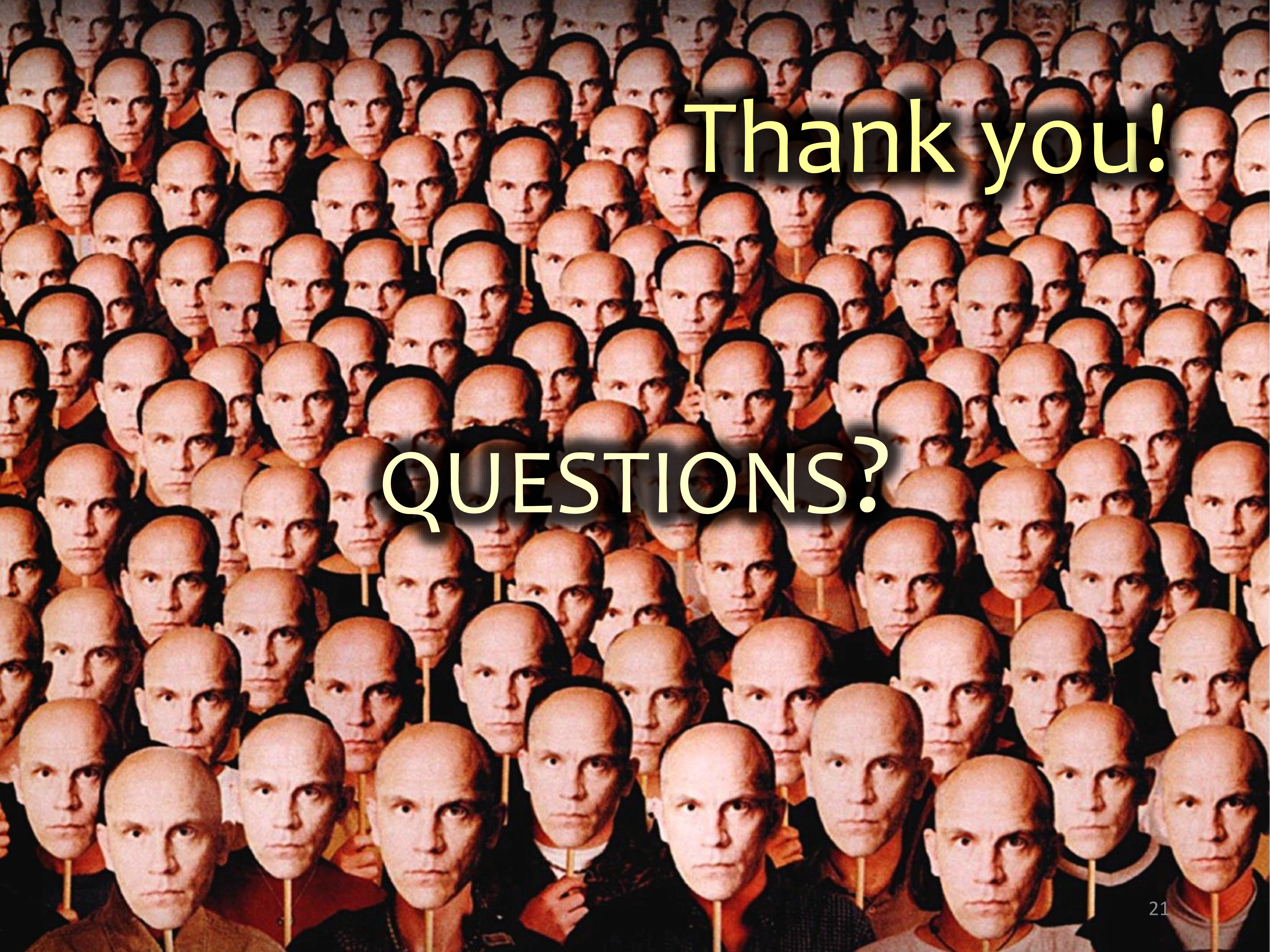
sdkernel summary

NEW APPROACH TO KERNEL ID

- # fully automated, zero reverse engineering
- # uses only on-disk kernel image to build signatures
- # accurate & robust
- # works across different architectures (x86/arm)
- # efficient → signature can produced from 25 blocks
  - » |signature| = 25x256 = 6,400 bytes

SURVEY OF LINUX KERNELS

- # varying degree of similarity
- # build options **substantially** influence outcome
  - » more than (neighboring) code versions
  - » x86: *generic* vs. *lowlatency*; arm5: *default* vs. *qemu*
  - » custom kernels should be quite unique



Thank you!

QUESTIONS?