HUNTING FOR METAMORPHIC ENGINES

Mark Stamp
&
Wing Wong

August 5, 2006

Outline

I. Metamorphic software
- Both good and evil uses

II. Metamorphic virus construction kits

III. How effective are metamorphic engines?
- How to compare two pieces of code?
- Similarity within and between virus families
- Similarity to non-viral code

IV. Can we detect metamorphic viruses?
- Commercial virus scanners
- Hidden Markov models (HMMs)
- Similarity index

V. Conclusion

PART I

Metamorphic Software

What is Metamorphic Software?

- Software is *metamorphic* provided
  - All copies do the same thing
  - Internal structure of copies differs
- Today almost all software is *cloned*
- "Good" metamorphic software…
  - Mitigate buffer overflow attacks
- "Bad" metamorphic software…
  - Avoid virus/worm signature detection

Metamorphic Software for "Good"?

- Suppose program has a buffer overflow
- If we clone the program
  - One attack breaks *every* copy
  - Break once, break everywhere (BOBE)
- If instead, we have metamorphic copies
  - Each copy still has a buffer overflow
  - One attack does not work against every copy
  - BOBE-resistant
  - Analogous to genetic diversity in biology
- A little metamorphism does a lot of good!

Metamorphic Software for Evil?

- Cloned virus/worm can be detected
  - Common signature on *every* copy
  - Detect once, detect everywhere (DODE)
- If instead virus/worm is metamorphic
  - Each copy has different signature
  - Same detection does not work against every copy
  - Provides DODE-resistance
  - Analogous to genetic diversity in biology
- But, effective metamorphism here is tricky!

Virus Evolution

- Viruses first appeared in the 1980s
  - Fred Cohen
- Viruses must avoid signature detection
  - Virus can alter its “appearance”
- Techniques employed
  - encryption
  - polymorphic
  - metamorphic

Virus Evolution - *Encryption*

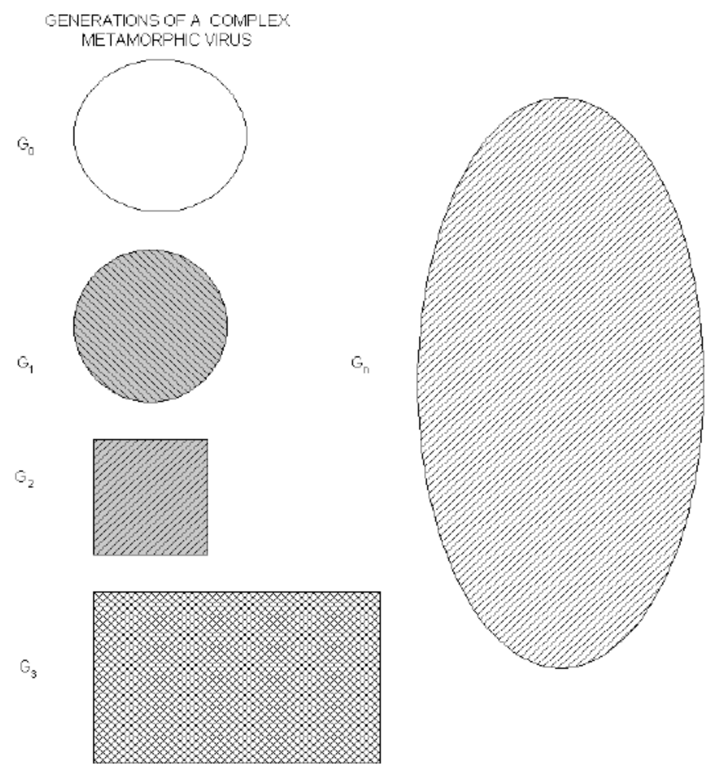
- Virus consists of
  - decrypting module (decryptor)
  - encrypted virus body
- Different encryption key
  - different virus body signature
- Weakness
  - decryptor can be detected

Virus Evolution – *Polymorphism*

- Try to hide signature of decryptor
- Can use *code emulator* to decrypt putative virus dynamically
- Decrypted virus body is constant
  - Signature detection is possible

Virus Evolution – *Metamorphism*

- Change virus body
- Mutation techniques:
  - permutation of subroutines
  - insertion of garbage/jump instructions
  - substitution of instructions



PART II

Virus Construction Kits

Virus Construction Kits – PS-MPC

- According to Peter Szor:
  - "… PS-MPC [*Phalcon/Skism Mass-Produced Code generator*] uses a generator that effectively works as a code-morphing engine…… the viruses that PS-MPC generates are not [only] polymorphic, but their decryption routines and structures change in variants…"

Virus Construction Kits – G2

- From the documentation of G2 (*Second Generation virus generator*):
  - "… different viruses may be generated from identical configuration files…"

Virus Construction Kits - NGVCK

- From the documentation for NGVCK (*Next Generation Virus Creation Kit*):

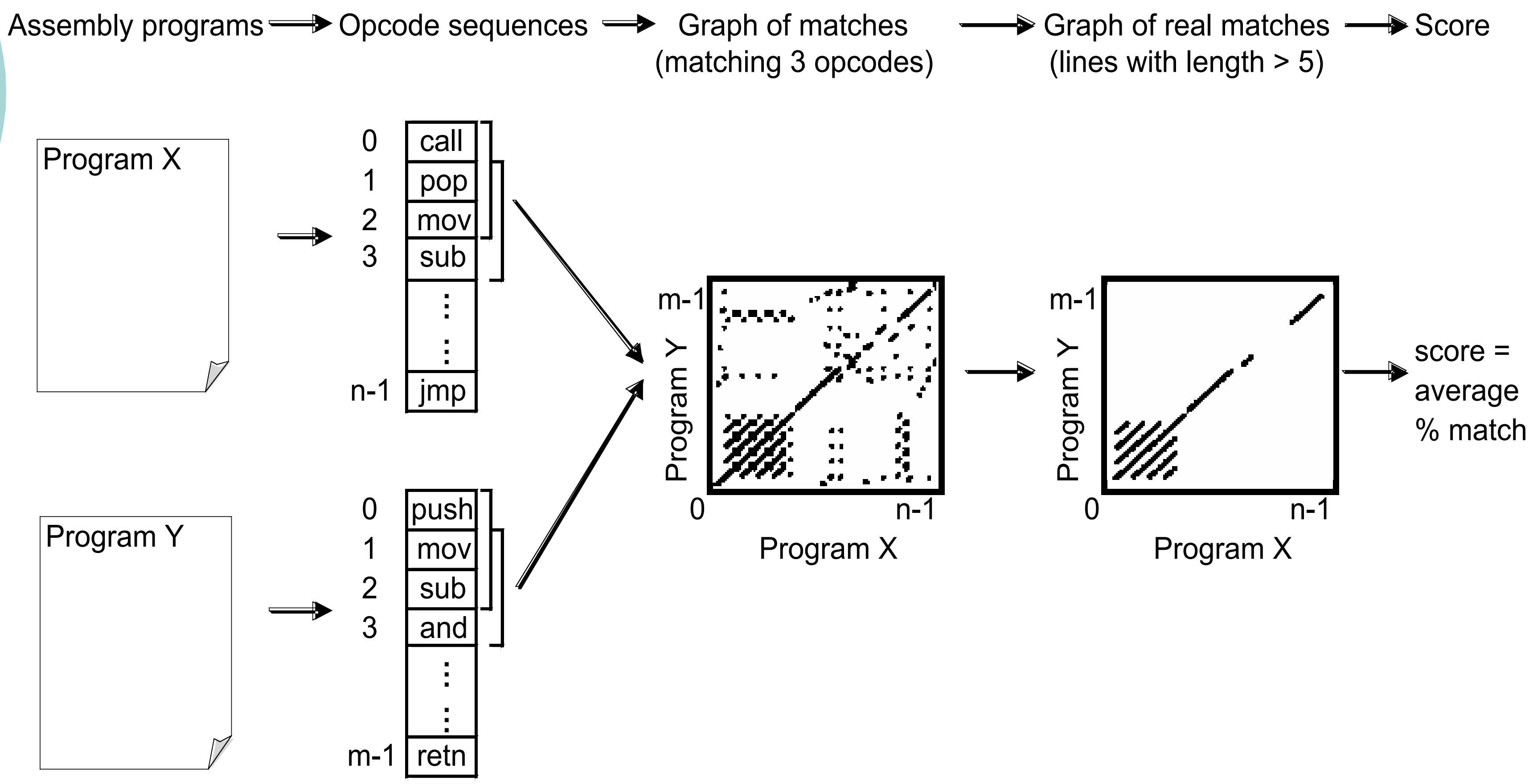
  "… all created viruses are completely different in structure and opcode…… impossible to catch all variants with one or more scanstrings…… nearly 100% variability of the entire code"

- Oh, really?

PART III

How Effective Are Metamorphic Engines?

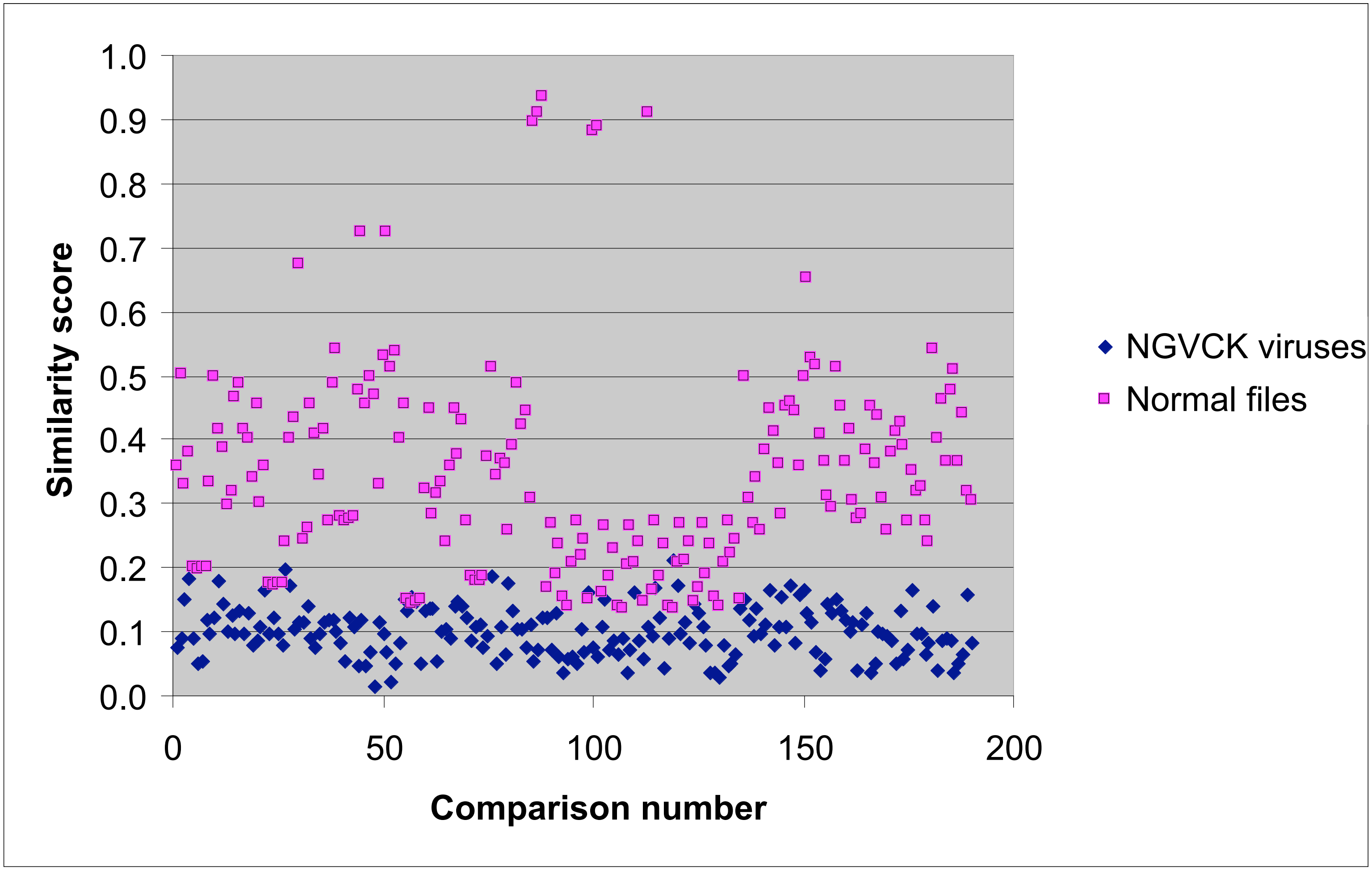
How We Compare Two Pieces of Code



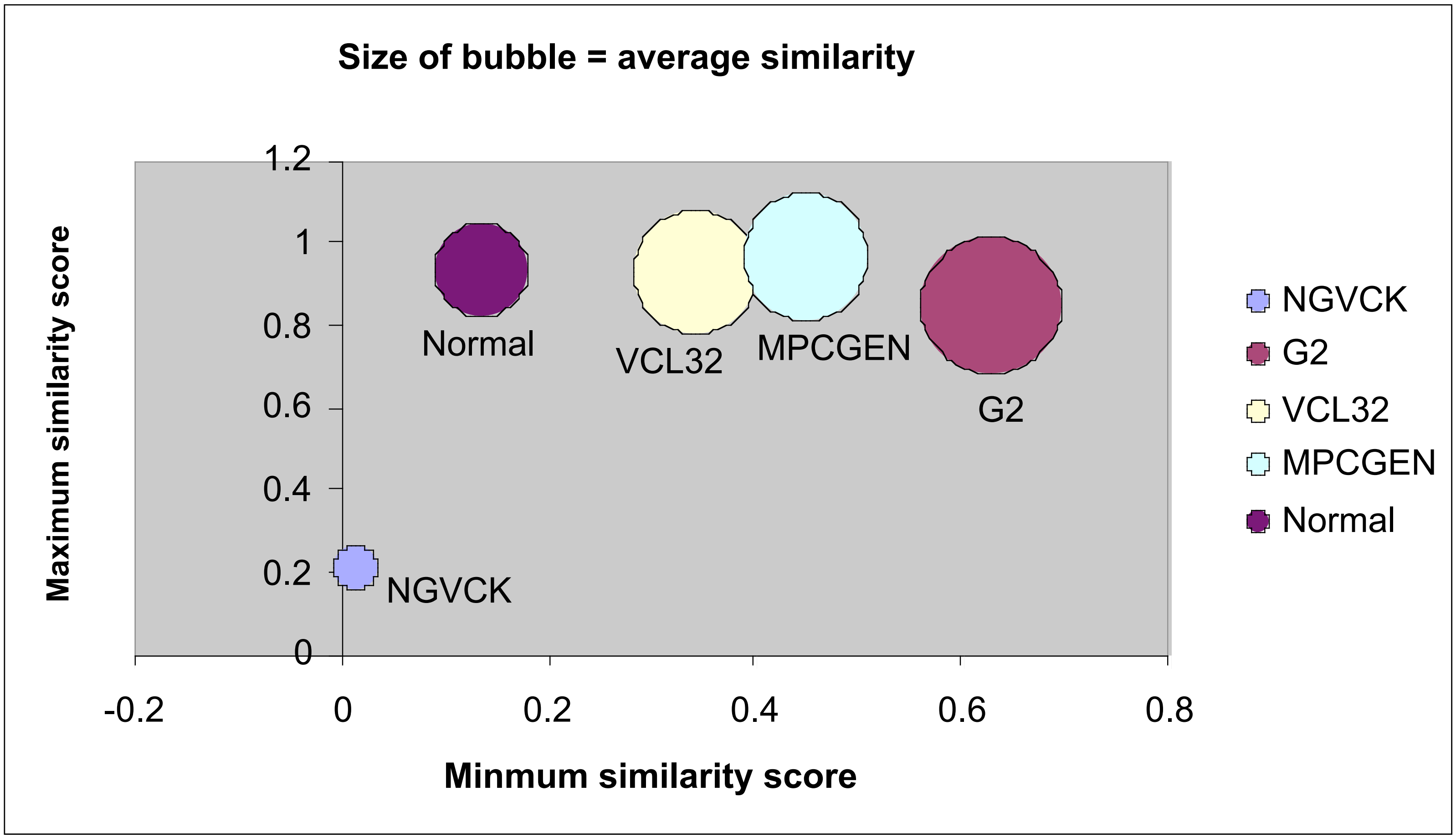
Virus Families – Test Data

- Four generators, 45 viruses
  - 20 viruses by NGVCK
  - 10 viruses by G2
  - 10 viruses by VCL32
  - 5 viruses by MPCGEN
- 20 normal utility programs from the Cygwin DLL

Similarity within Virus Families – Results

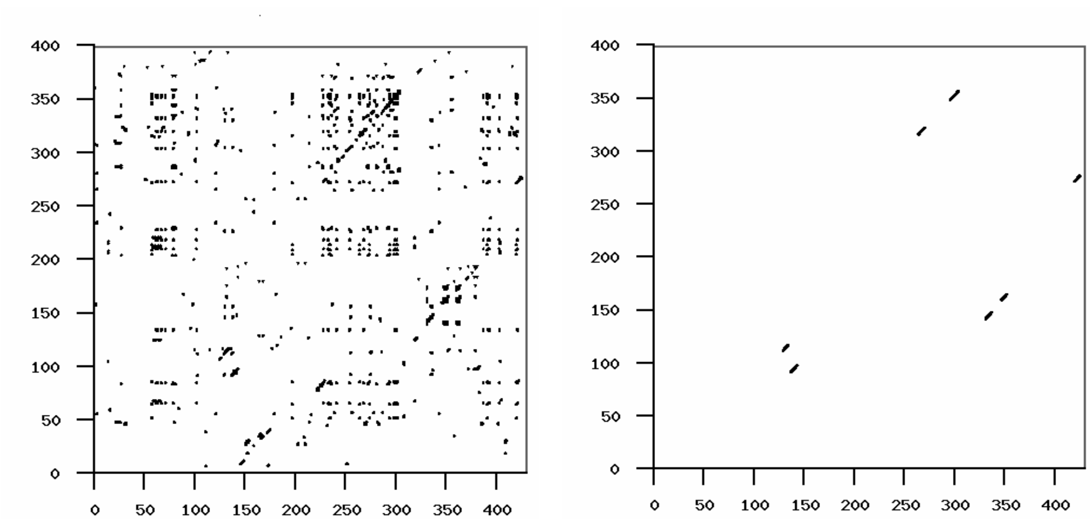


Similarity within Virus Families – Results

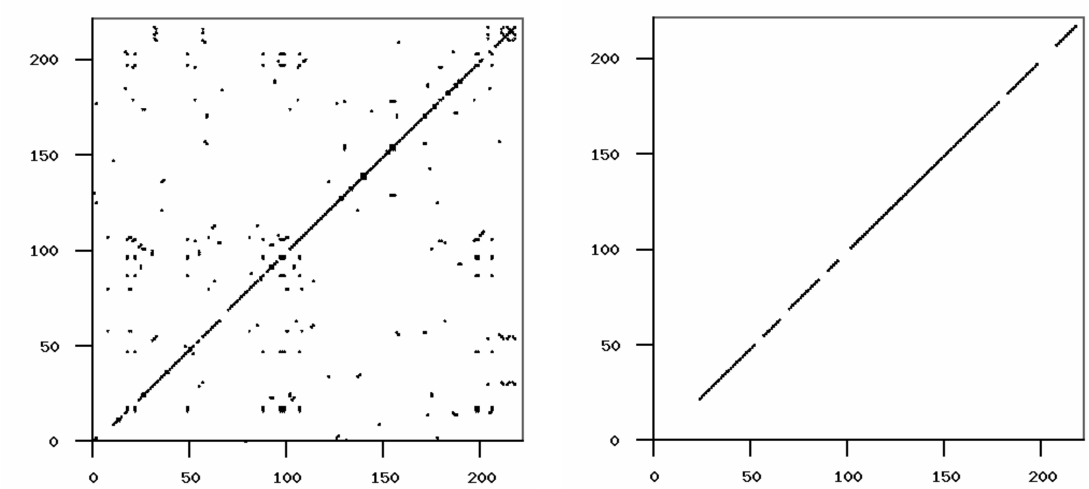


Similarity within Virus Families – Results

IDA_
NGVCK0-
IDA_
NGVCK8
(11.9%)

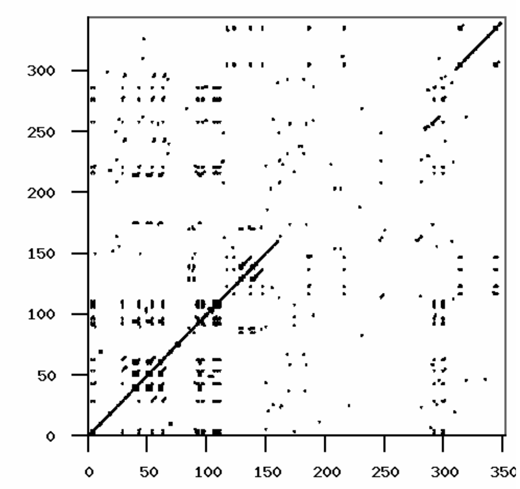


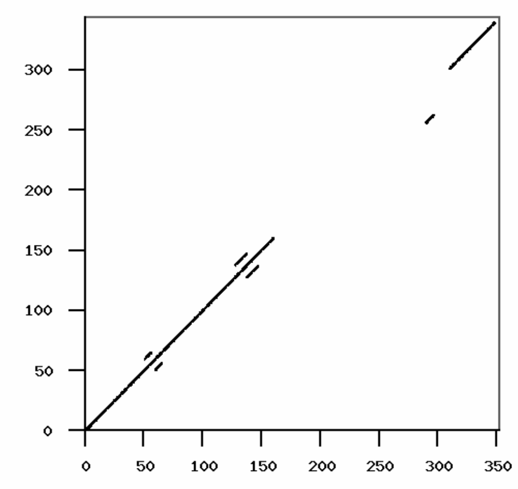
IDA_G4-
IDA_G7
(75.2%)



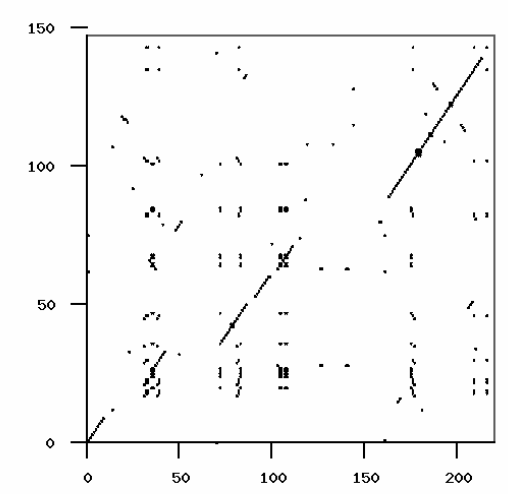
Similarity within Virus Families – Results

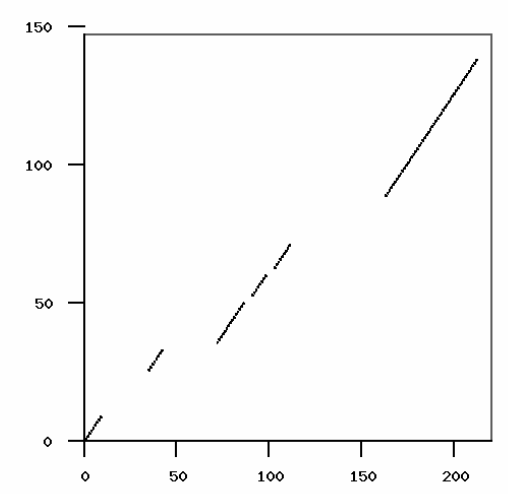
IDA_VCL0-IDA_VCL9 (60.2%)





IDA_MPC1-IDA_MPC3 (58.0%)





NGVCK Similarity to Virus Families

- NGVCK versus other viruses
  - **0%** similar to G2 and MPCGEN viruses
  - **0 – 5.5%** similar to VCL32 viruses (43 out of 100 comparisons have score > 0)
  - **0 – 1.2%** similar to normal files (only 8 out of 400 comparisons have score > 0)

NGVCK Metamorphism/Similarity

- NGVCK
  - By far the highest degree of metamorphism of any kit tested
  - Virtually no similarity to other viruses or normal programs
  - Undetectable???

PART IV

Can Metamorphic Viruses Be Detected?

Commercial Virus Scanners

- Tested three virus scanners
  - eTrust version 7.0.405
  - avast! antivirus version 4.7
  - AVG Anti-Virus version 7.1
- Each scanned 37 files
  - 10 NGVCK viruses
  - 10 G2 viruses
  - 10 VCL32 viruses
  - 7 MPCGEN viruses

Commercial Virus Scanners

- Results
  - eTrust and avast! detected **17** (G2 and MPCGEN)
  - AVG detected **27** viruses (G2, MPCGEN and VCL32)
  - **none** of NGVCK viruses detected

Detection with Hidden Markov Models

- Use *hidden Markov models* (HMMs) to represent *statistical properties* of a set of metamorphic virus variants
  - Train the model on family of metamorphic viruses
  - Use trained model to determine whether a given program is *similar* to the viruses the HMM represents

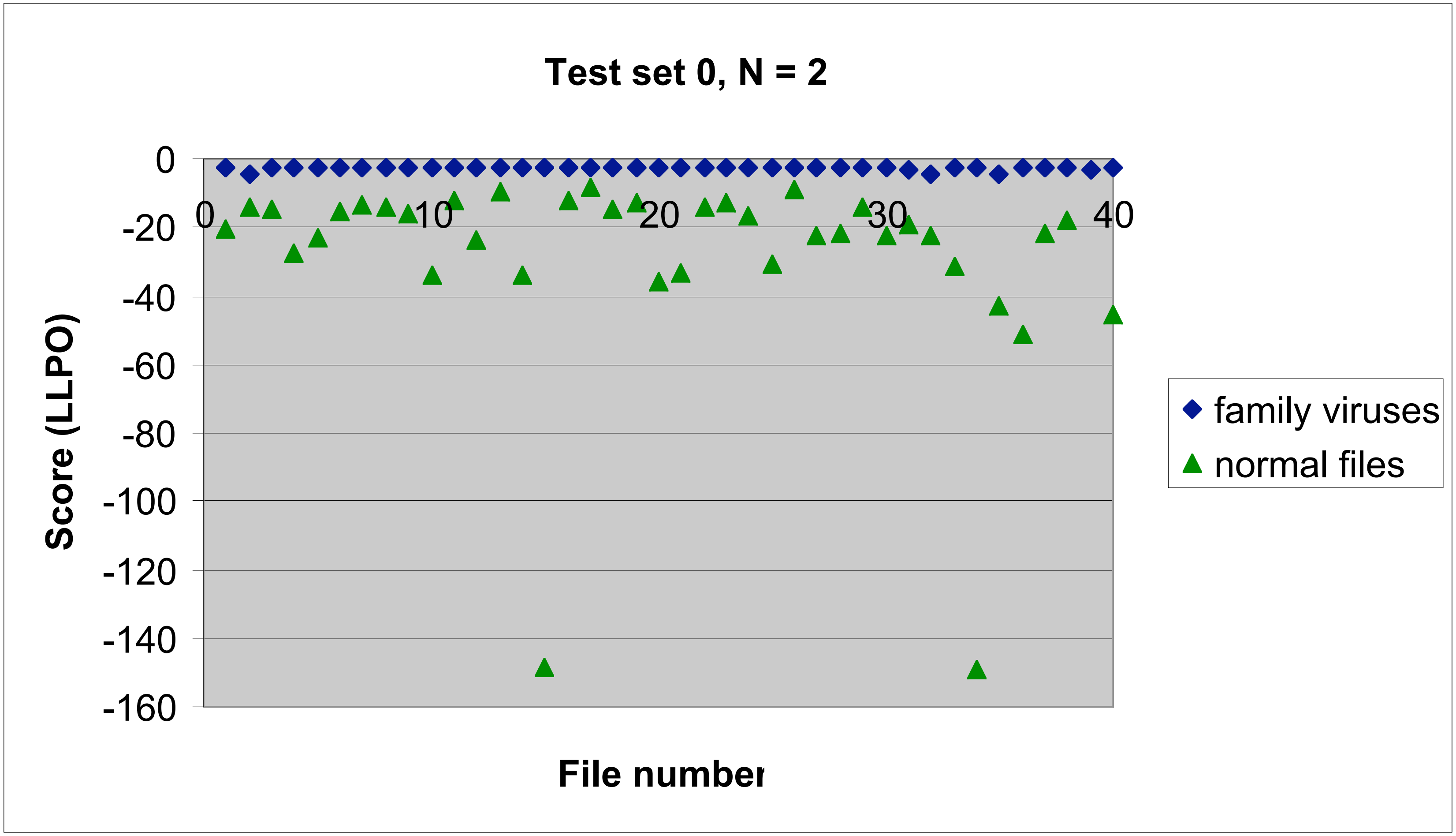
Detection with HMMs – Theory

- A trained HMM
  - maximizes the probabilities of observing the training sequence
  - assigns high probabilities to sequences similar to the training sequence
  - represents the “average” behavior if trained on multiple sequences
  - represents an entire virus family, as opposed to individual viruses

Detection with HMMs – Data

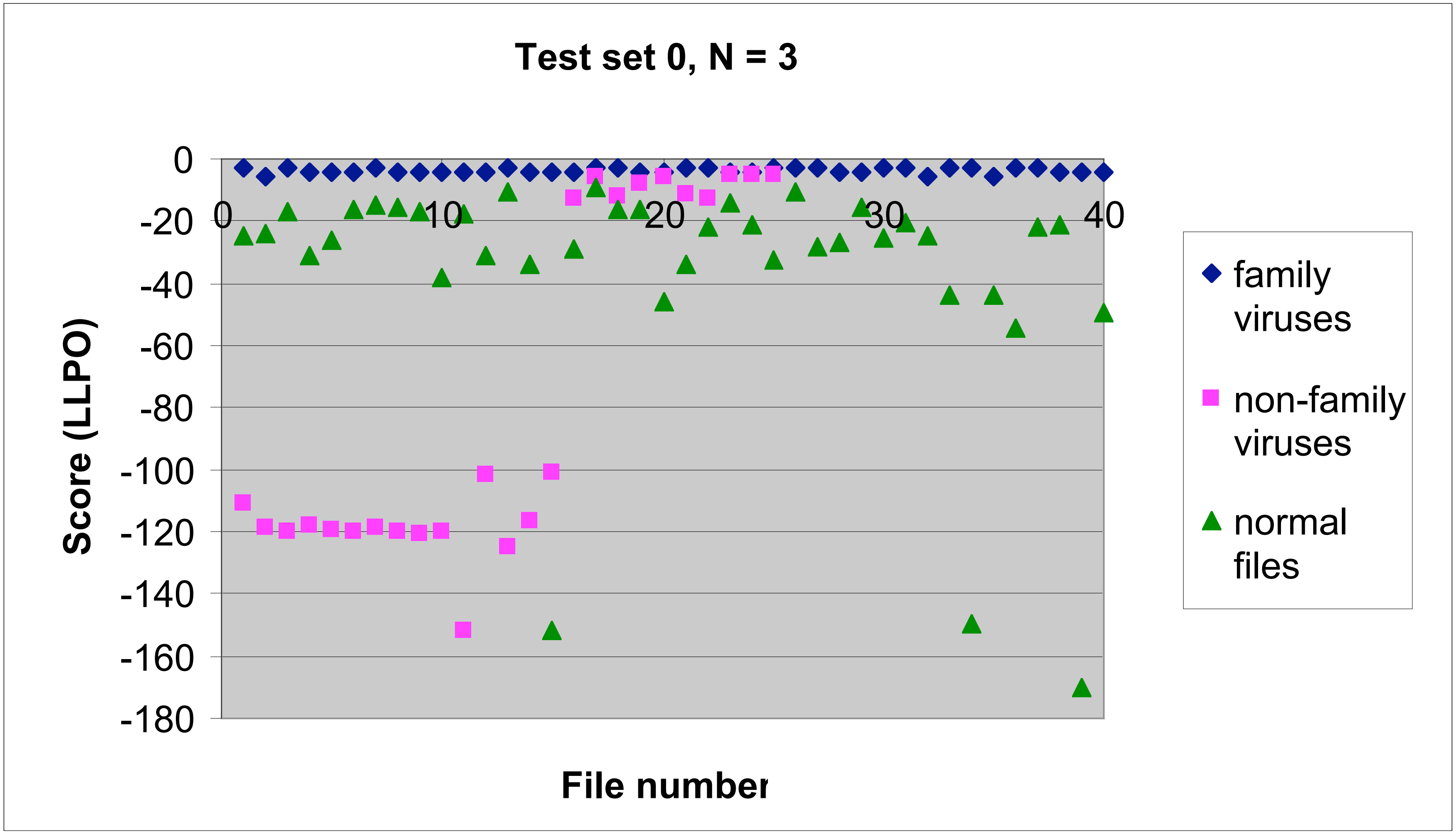
- *Data set*
  - 200 NGVCK viruses
- *Comparison set*
  - 40 normal exes from the Cygwin DLL
  - 25 other "non-family" viruses (G2, MPCGEN and VCL32)
- Many HMM models generated and tested

Detection with HMMs – Results



Detection with HMMs – Results

- Detect some other viruses "for free"



Detection with HMMs

- Summary of experimental results
  - All normal programs distinguished
  - VCL32 viruses had scores close to NGVCK family viruses
  - With proper threshold, 17 HMM models had 100% detection rate and 10 models had 0% false positive rate
  - No significant difference in performance between HMMs with 3 or more hidden states

Detection with HMMs – Trained Models

- Converged probabilities in HMM matrices may give insight into the *features* of the viruses it represents
- We observe
  - opcodes grouped into “hidden” states
  - most opcodes in one state only
- What does this mean?
  - We are not sure…

Detection via Similarity Index

- Straightforward *similarity index* can be used as detector
  - To determine whether a program belongs to the NGVCK virus family, compare it to any randomly chosen NGVCK virus
  - NGVCK similarity to non-NGVCK code is small
  - Can use this fact to detect metamorphic NGVCK variants

Detection with Similarity Index

- Experiment
  - compare 105 programs to one selected NGVCK virus
- Results
  - 100% detection, 0% false positive
- Does not depend on specific NGVCK virus selected

PART V

Conclusion

Conclusion

- Metamorphic generators vary a lot
  - NGVCK has highest metamorphism (**10%** similarity on average)
  - Other generators far less effective (**60%** similarity on average)
  - Normal files **35%** similar, on average
- But, NGVCK viruses can be detected!
  - NGVCK viruses *too different* from other viruses and normal programs

Conclusion

- NGVCK viruses not detected by commercial scanners we tested
- Hidden Markov model (HMM) detects NGVCK (and other) viruses with high accuracy
- NGVCK viruses also detectable by similarity index

Conclusion

- All metamorphic viruses tested were detectable because
  - High similarity within family and/or
  - Too different from normal programs
- Effective use of metamorphism by virus/worm requires
  - A high degree of metamorphism *and* similarity to other programs
  - This is not trivial!

The Bottom Line

- Metamorphism for “good”
  - For example, buffer overflow mitigation
  - A little metamorphism does a lot of good
- Metamorphism for “evil”
  - For example, try to evade virus/worm signature detection
  - Requires high degree of metamorphism and similarity to normal programs
  - Not impossible, but not easy…

References

- X. Gao, "Metamorphic Software for Buffer Overflow Mitigation", masters thesis, Department of Computer Science, San Jose State University, 2005
- P. Szor, *The Art of Computer Virus Research and Defense*, Addison-Wesley, 2005
- M. Stamp, *Information Security: Principles and Practice*, Wiley Interscience, 2005
- W. Wong, "Analysis and Detection of Metamorphic Computer Viruses", masters thesis, Department of Computer Science, San Jose State University, 2006