Dynamic k-gram based software birthmark

Y. Bai, X. Sun, G. Sun, X. Deng, X. Zhou

19th Australian Conference on Software Engineering (ASWEC 2008), pp. 644-649, 2008.

Sonny Datt

Summary

When used to create dynamic k-grams, dependence graphs could help software companies reclaim their k-pounds of flesh.

Critical Comment

The paper states:

"(Obfuscating a program is the standard way to attack a birthmark)."

[point 2 in the 'contribution of our paper' section]

- They make this statement without reference.
- The term standard could have multiple meanings.
- Obfuscation is the only threat to birthmarking that the paper uses to compare dynamic k-grams against static k-grams.
- This sentence just means that the paper has found a new k-gram approach that holds up better against obfuscation. Not necessarily a better approach overall.

Appreciative Comment

- The paper has an ‘open style’:
  - The paper clearly defines (mathematically) all of the procedures used to make their claims.
  - Their development process is transparent and written in a way that requires relatively little technical knowledge to follow.
  - The paper presents an algorithm that is easy to understand yet appears difficult to ‘trick.’

The paper has an ‘open style’

- The paper clearly defines (mathematically) all of the procedures used to make their claims.
  - The k-grams paper uses clear procedure descriptions. Something that in general produces a stronger result.
  - Some other security based papers have ambiguous procedures, or they fail to account for borderline cases.
  - Example: In the paper “Accountable Privacy,” the first paper to be presented for this course. The authors define privacy loosely: *“Loosely speaking, privacy is the ability to control private information,...”* [page 1]

The paper has an ‘open style’

- Their development process is transparent and written in a way that requires relatively little technical knowledge to follow.
  - Having a clear development process means that we the reader can test the theory that is being presented easier.
  - The clear development process also means that we the reader aren’t left wondering “what happens if x.”
  - Example: You may recall that for the Lampson article, we as a class questioned what would happen if the guard in Lampson's model was spammed by a single user.

The paper has an ‘open style’

- Easy to understand yet difficult to ‘trick.’
  - In security, this sort of approach could be seen as a deterrent.
  - Consider a lock that appears unbreakable, if you see one you are less likely to attack it.
  - Example: For RSA public key encryption, obtaining the prime factors involved is known to be difficult.

Question

Do you think that the k-gram papers' open style is beneficial to software/systems security?