Static code analysis report

Francesco Paolo Di Lorenzo
student ID: 1712990
e-mail: dilorenzo.1712990@studenti.uniroma1.it

Contents

1 Static analysis tools

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

1.1 Flawfinder

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

1.2 Splint

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

2 Output description

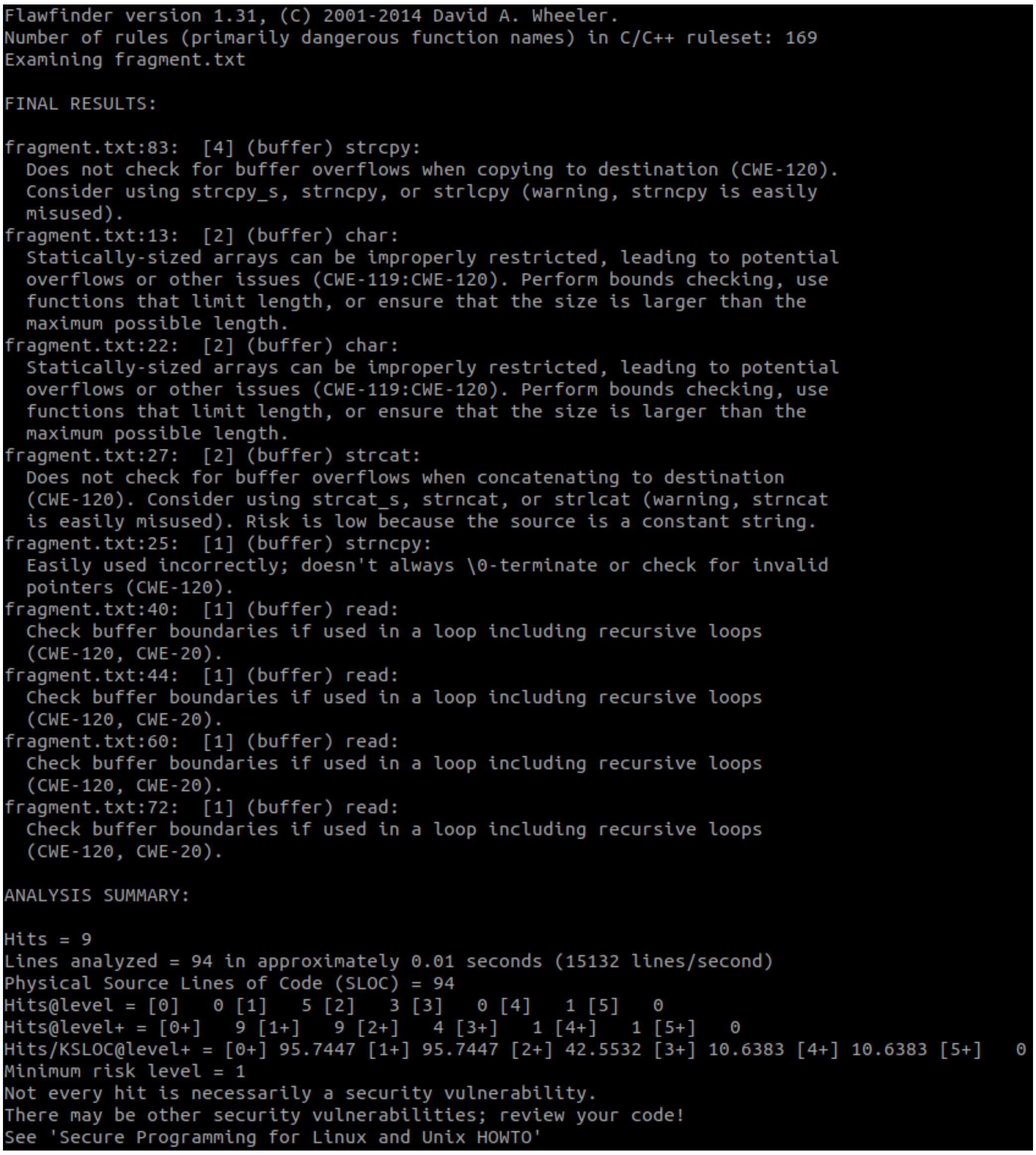
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

2.1 Flawfinder output

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

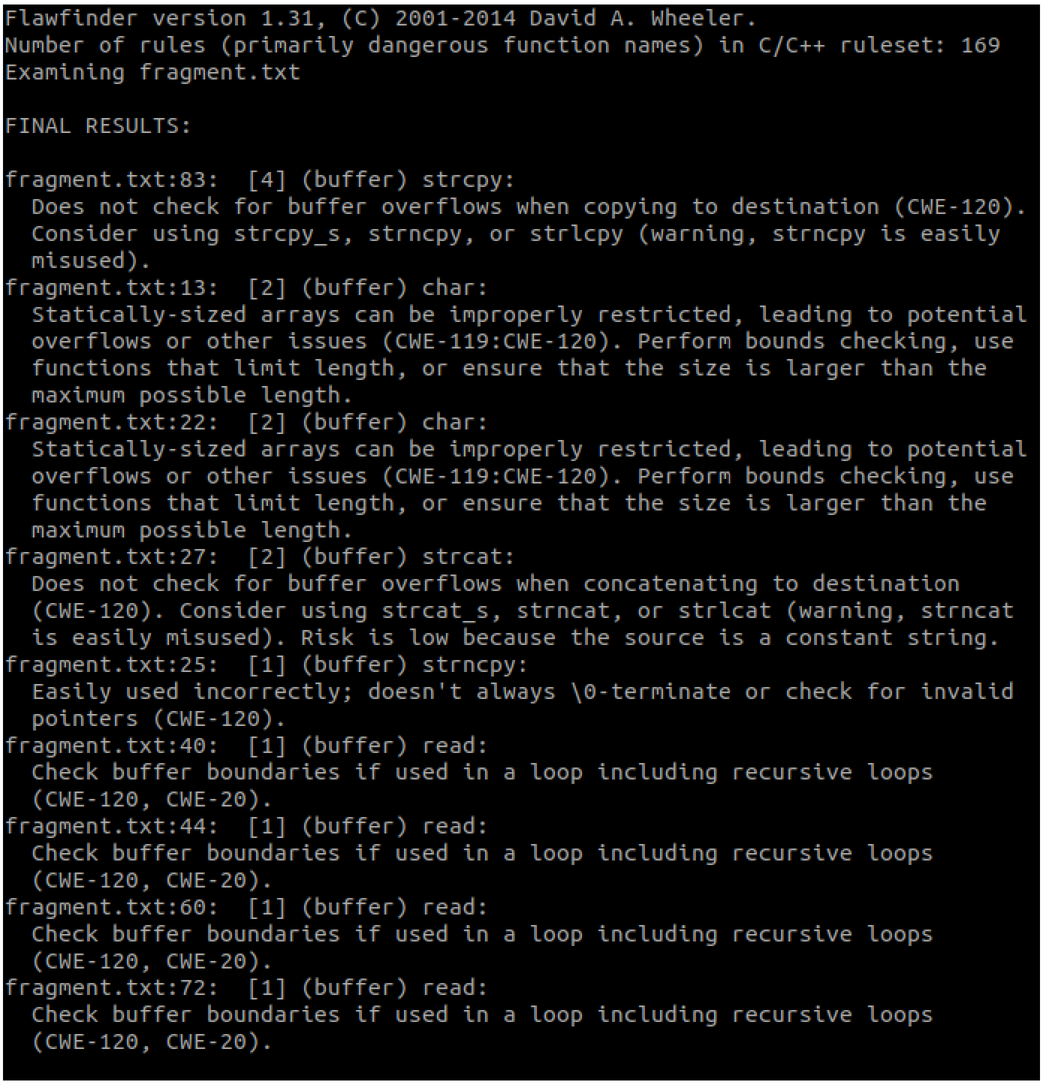
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.



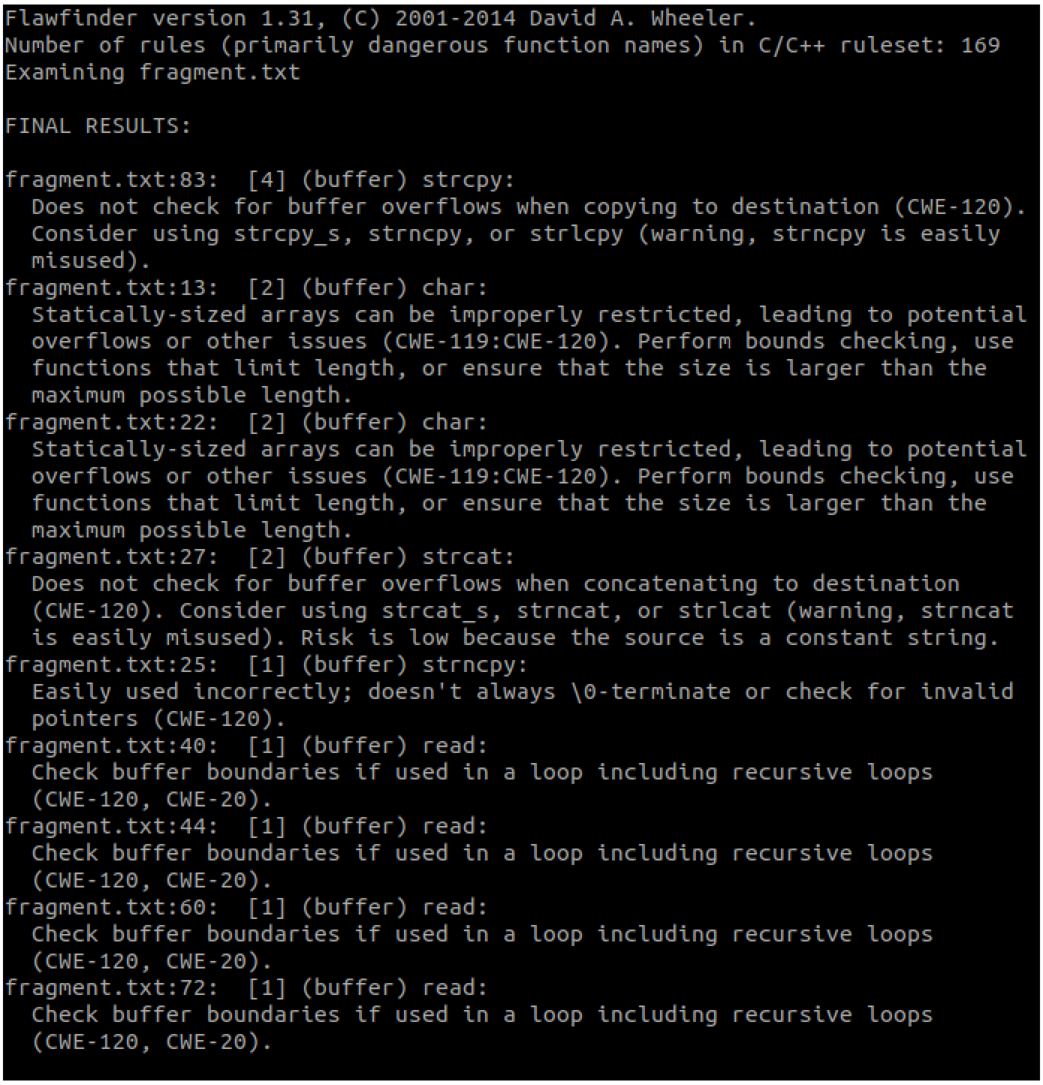
```
Flawfinder version 1.31, (C) 2001-2014 David A. Wheeler.
Number of rules (primarily dangerous function names) in C/C++ ruleset: 169
Examining fragment.txt

FINAL RESULTS:

fragment.txt:83:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Consider using strcpy_s, strncpy, or strlcpy (warning, strncpy is easily
  misused).
fragment.txt:13:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119:CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
fragment.txt:22:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119:CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
fragment.txt:27:  [2] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  (CWE-120). Consider using strcat_s, strncat, or strlcat (warning, strncat
  is easily misused). Risk is low because the source is a constant string.
fragment.txt:25:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers (CWE-120).
fragment.txt:40:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:44:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:60:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:72:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).

ANALYSIS SUMMARY:

Hits = 9
Lines analyzed = 94 in approximately 0.01 seconds (15132 lines/second)
Physical Source Lines of Code (SLOC) = 94
Hits@level = [0]   0 [1]   5 [2]   3 [3]   0 [4]   1 [5]   0
Hits@level+ = [0+]   9 [1+]   9 [2+]   4 [3+]   1 [4+]   1 [5+]   0
Hits/KSLOC@level+ = [0+] 95.7447 [1+] 95.7447 [2+] 42.5532 [3+] 10.6383 [4+] 10.6383 [5+]   0
Minimum risk level = 1
Not every hit is necessarily a security vulnerability.
There may be other security vulnerabilities; review your code!
See 'Secure Programming for Linux and Unix HOWTO'
```

Figure 1: Main results



```
Flawfinder version 1.31, (C) 2001-2014 David A. Wheeler.
Number of rules (primarily dangerous function names) in C/C++ ruleset: 169
Examining fragment.txt

FINAL RESULTS:

fragment.txt:83:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Consider using strcpy_s, strncpy, or strlcpy (warning, strncpy is easily
  misused).
fragment.txt:13:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119:CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
fragment.txt:22:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119:CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
fragment.txt:27:  [2] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  (CWE-120). Consider using strcat_s, strncat, or strlcat (warning, strncat
  is easily misused). Risk is low because the source is a constant string.
fragment.txt:25:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers (CWE-120).
fragment.txt:40:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:44:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:60:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:72:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
```

Figure 2: Final results



```
Flawfinder version 1.31, (C) 2001-2014 David A. Wheeler.
Number of rules (primarily dangerous function names) in C/C++ ruleset: 169
Examining fragment.txt

FINAL RESULTS:

fragment.txt:83:  [4] (buffer) strcpy:
  Does not check for buffer overflows when copying to destination (CWE-120).
  Consider using strcpy_s, strncpy, or strlcpy (warning, strncpy is easily
  misused).
fragment.txt:13:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119:CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
fragment.txt:22:  [2] (buffer) char:
  Statically-sized arrays can be improperly restricted, leading to potential
  overflows or other issues (CWE-119:CWE-120). Perform bounds checking, use
  functions that limit length, or ensure that the size is larger than the
  maximum possible length.
fragment.txt:27:  [2] (buffer) strcat:
  Does not check for buffer overflows when concatenating to destination
  (CWE-120). Consider using strcat_s, strncat, or strlcat (warning, strncat
  is easily misused). Risk is low because the source is a constant string.
fragment.txt:25:  [1] (buffer) strncpy:
  Easily used incorrectly; doesn't always \0-terminate or check for invalid
  pointers (CWE-120).
fragment.txt:40:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:44:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:60:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
fragment.txt:72:  [1] (buffer) read:
  Check buffer boundaries if used in a loop including recursive loops
  (CWE-120, CWE-20).
```

Figure 3: Analysis summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

2.2 Splint output

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

3 Corrected version of the fragment

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

4 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent elementum aliquam massa, eget vestibulum arcu porta lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc et turpis dignissim, viverra arcu id, eleifend risus. Nulla facilisi. Sed venenatis placerat nibh, vel vehicula mi molestie a. Donec porttitor pharetra tortor id sodales. In ultricies, ex quis aliquet laoreet, sapien est pulvinar elit, vitae gravida metus lorem et mi. Nulla lacus ante, feugiat a diam at, sodales rutrum sapien. Nam sit amet odio nisl. Sed in auctor metus. Fusce viverra aliquet metus a condimentum.

References

[1] Baker, N. 1966, in Stellar Evolution, ed. R. F. Stein & A. G. W. Cameron (Plenum, New York) 333

[2] Balluch, M. 1988, A&A, 200, 58

[3] Cox, J. P. 1980, Theory of Stellar Pulsation (Princeton University Press, Princeton) 165

[4] Cox, A. N.,& Stewart, J. N. 1969, Academia Nauk, Scientific Information 15, 1

[5] Mizuno H. 1980, Prog. Theor. Phys., 64, 544

[6] Tscharnuter W. M. 1987, A&A, 188, 55

[7] Terlevich, R. 1992, in ASP Conf. Ser. 31, Relationships between Active Galactic Nuclei and Starburst Galaxies, ed. A. V. Filippenko, 13

[8] Yorke, H. W. 1980a, A&A, 86, 286

[9] Zheng, W., Davidsen, A. F., Tytler, D. & Kriss, G. A. 1997, preprint