

# Automated Evaluation

Romain Gaucher  
SATE 2008 - Experience

[romain.gaucher@nist.gov](mailto:romain.gaucher@nist.gov)

# Well, first of all....

Certain trade names or company products may be mentioned in the text or identified.

In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology (NIST), nor does it imply that the products are necessarily the best available for the purpose.

# Agenda

- Flawfinder as a participant
- Automated Analysis of tool report
- Results

# Flawfinder

- Written by David A. Wheeler
- Source code scanner ("grep-like" class of tool)
- Very fast
- Looking for problem based on a DB of function names
- Interesting to compare with other tools...

<http://www.dwheeler.com/flawfinder>

Running Flawfinder...

EASY

# Why automated evaluation?

- Flawfinder reports lots of weaknesses
- Since enhanced grep, many are false-positive
- Need to find automated way to reduce the number

# Automated Evaluation

tools do:

- syntactic/control-flow/data-flow/control-graph analysis
- plus some other stuff...

people do:

- read and understand the actions in the code

# Idea of a secure piece of code

Thinking of a secure piece of code like a puzzle... ?



# Automated Evaluation

- **Probe false-positive ONLY!**
- Focus on the 'buffer' problem
- Look at actions or steps:
  - Reallocation
  - Computing buffer size
  - Comparisons
  - Test for NULL after allocation
  - ...

# Automated Evaluation

Example from Naim

```
001795 | str = malloc(strlen(command)+1+strlen(args)+1);  
001796 | if (str == NULL)  
001797 |     abort();  
001798 | sprintf(str, "%s %s", command, args);
```

Flawfinder reports a buffer problem on line 1798:  
"sprintf: Does not check for buffer overflows. Use  
snprintf or vsnprintf. "

# Automated Evaluation

Example from Naim... cont'd

```
001795 | str = malloc(strlen(command)+1+strlen(args)+1);  
001796 | if (str == NULL)  
001797 |     abort();  
001798 | sprintf(str, "%s %s", command, args);
```

Important points in the code:

- Test for NULL dst pointer
- The destination buffer is allocated with the good size

**the code seems fine to me...**

# Automated Evaluation

Example from Naim... cont'd - our tool report

Source buffer(s) found: command, args,

Destination buffer(s) found: str

## # variable assignment, size computing phase

computed size of 'command, args, ' in [ str = malloc(strlen  
(command) + 1 + strlen(args) + 1); ]

size of 'command, args, ' assigned to 'str ' **noise**

the dst buffer 'str' has been reallocated in [ str = malloc(strlen  
(command)+1+strlen(args)+1); ]

## # structure analysis phase

there is a test here [ if(str==NULL) ]

## # REASON

the allocation is correctly tested

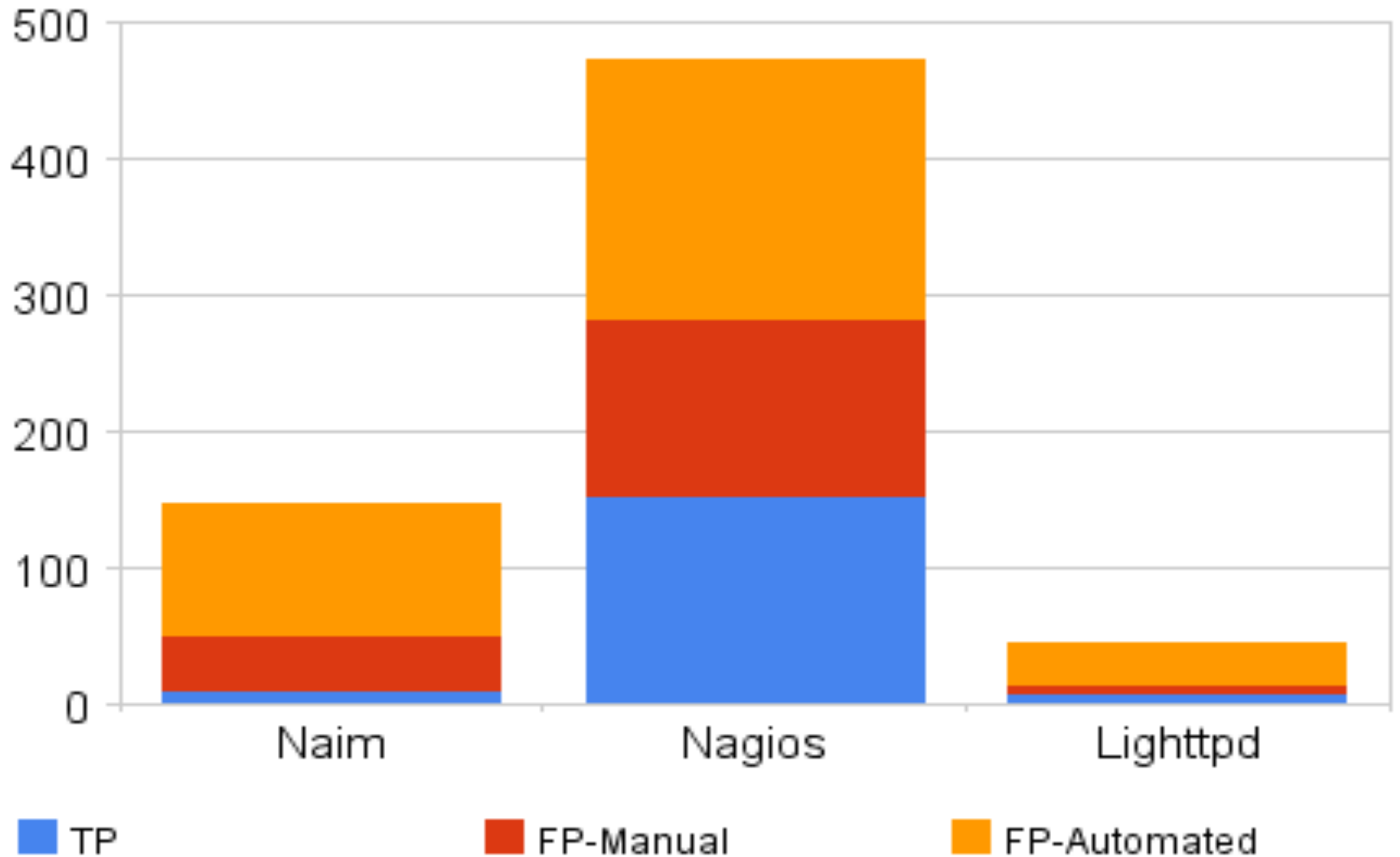
the allocation of [strlen(command)+1+strlen(args)+1] involves the  
size of the source buffer 'command, args, '

# Automated Evaluation

What our tool does...

- Determine source buffers, destination buffer
- Grab lines with these buffers such as computing size, allocation, etc.
- Match some patterns (regex) to get the actions
- Make a conclusion (scoring) about the false-positiveness of the warning

# Results on SATE 2008 test cases



# Conclusion

- Our automated evaluation tool is not perfect
  - noise in that small example
  - regexp based
  - requires many assumptions
  - **makes incorrect judgment sometimes**
- But the scoring helps us find FP
- We were able to **reduce our work** on Flawfinder's report (**~65% of our eval.**)

# Possible improvements

- Do not limit only to buffer overflow
  - XSS, SQLi, RFI, etc. could be analyze the same way
- Extend to other tools
  - Based on the path they report
  - Apply the same analysis
- Improve the parser
  - Limitation: it's getting as complicated as a SCA when you have to deal with AST

Questions?