The Seventh International Conference on Cyber-Technologies and Cyber-Systems
CYBER 2022
November 13, 2022 to November 17, 2022 - Valencia, Spain

SIEMENS
Ingenuity for life

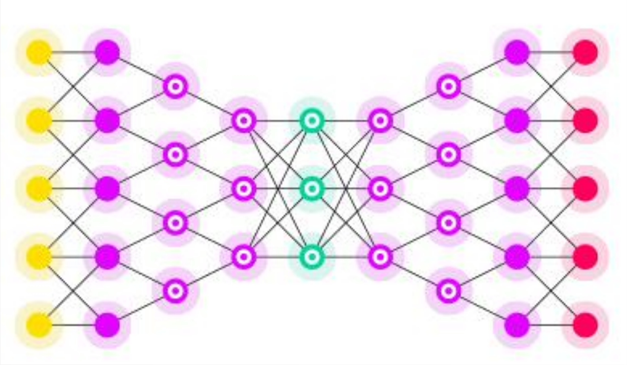
How Good is Openly Available Code Snippets Containing Software Vulnerabilities to Train Machine Learning Algorithms?

Kaan Oguzhan
Dr. rer. nat. Tiago Espinha Gasiba
Akram Louati

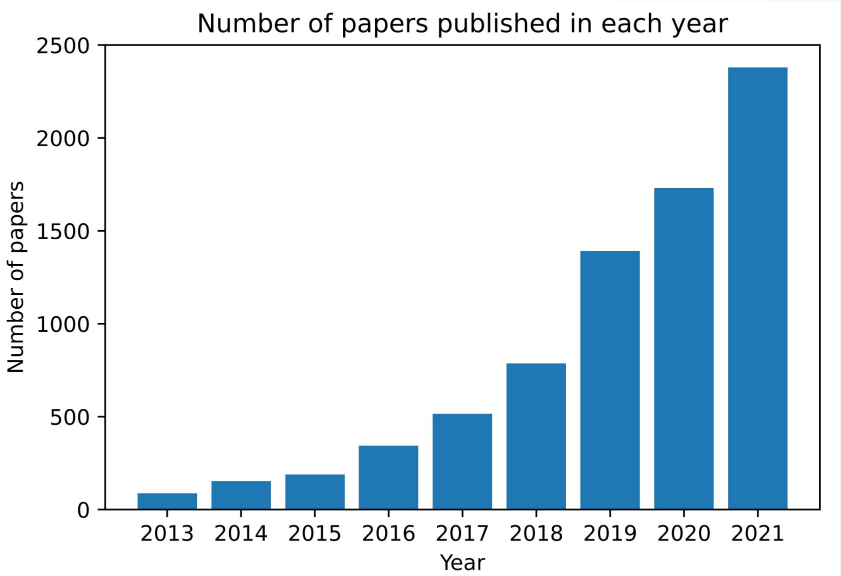
Siemens AG
Munich

Background

Machine Learning



[1]

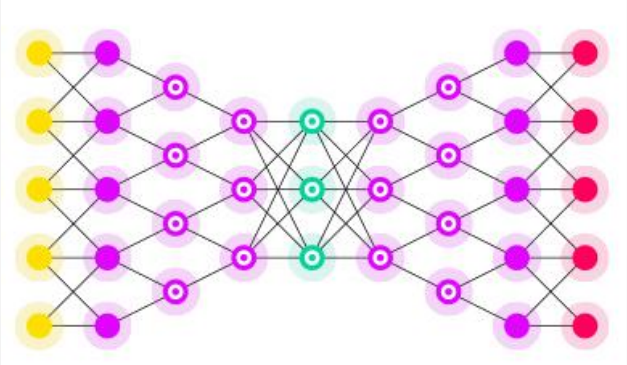


Appearance of both keywords ”Cybersecurity” and ”Machine Learning” in Academic Papers according to Scopus

1) https://becominghuman.ai/cheat-sheets-for-ai-neural-networks-machine-learning-deep-learning-big-data-678c51b4b463

Background

Machine Learning



[1]

Model is as good as your data

Data

Quality & Quantity

1) https://becominghuman.ai/cheat-sheets-for-ai-neural-networks-machine-learning-deep-learning-big-data-678c51b4b463

Motivation

➔ Goal: Developing software vulnerability detection in source code by means of ML Algorithms

➔ Training sounds straightforward, but

- ◆ *“Model is as good as your data”*

Motivation

➔ Research questions:

- Where can we find code snippets to train ML models to detect software vulnerabilities?
- What is the quality of the code snippets which are openly available on the internet for training ML Models?
  - Can they be used to train ML models?

NOTE: in our work we use industry standard categories of software vulnerabilities

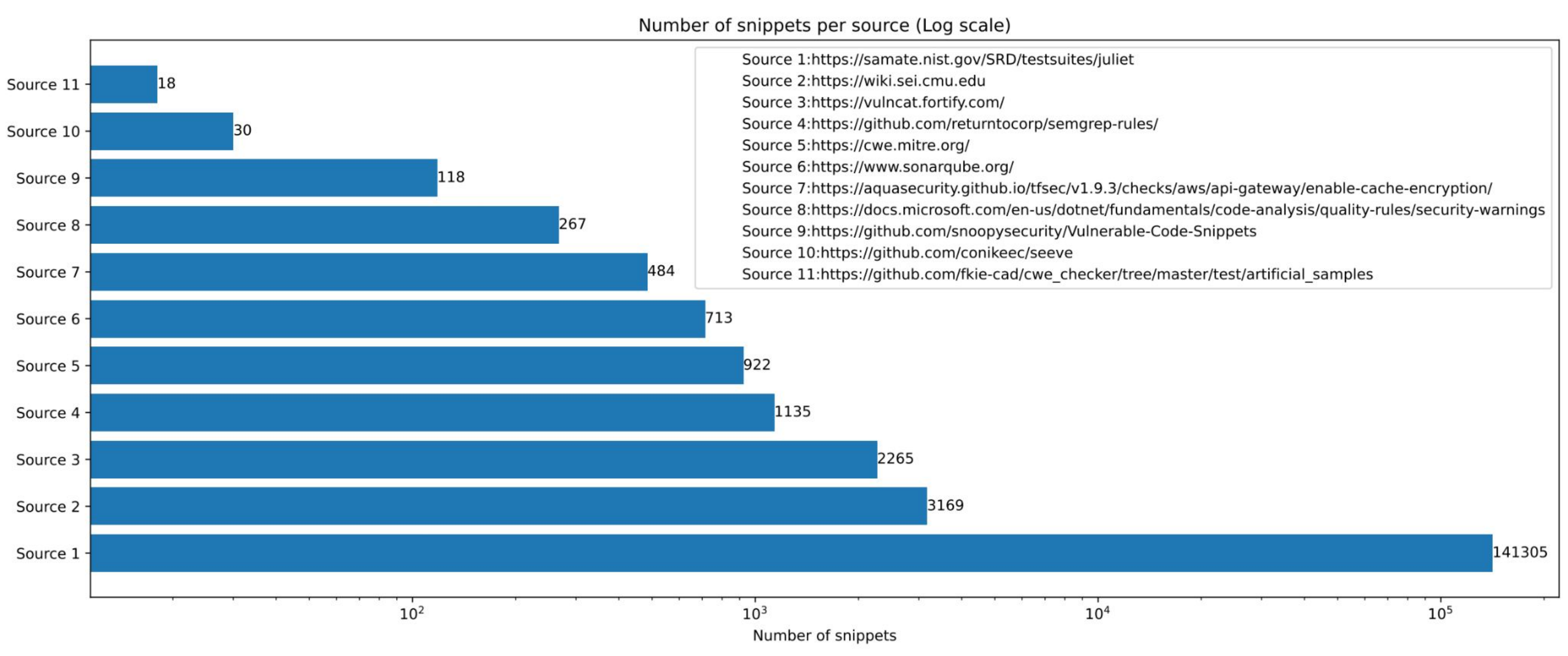
Overview

- ➔ Publicly available code snippets

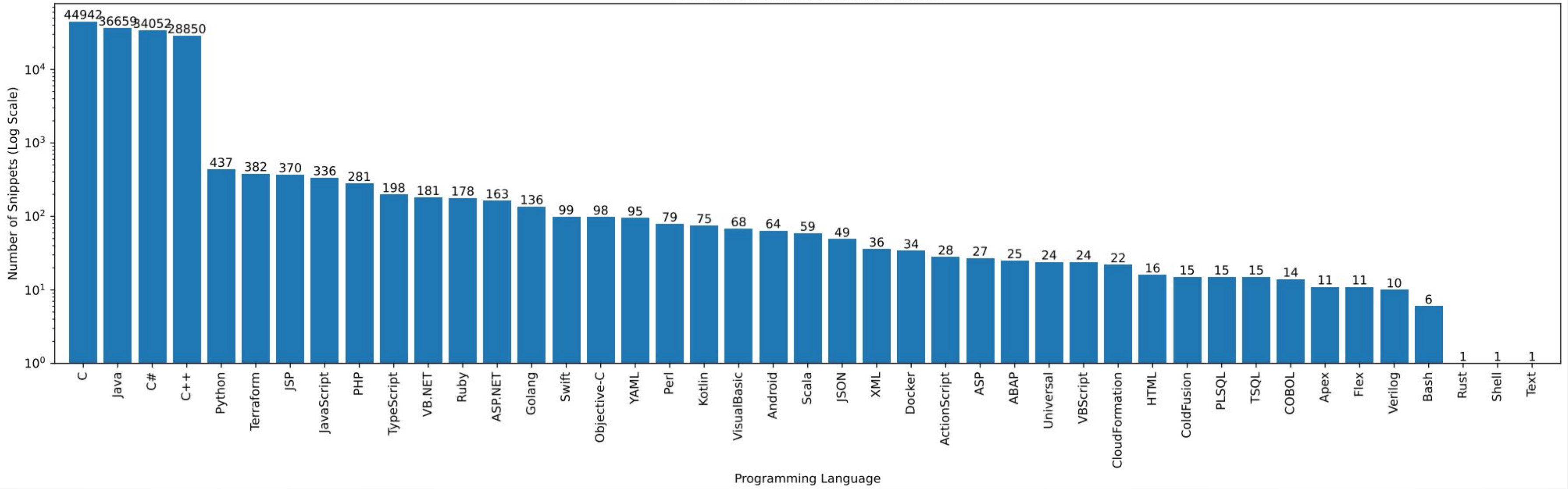
  Quality & Quantity

- ➔ Analysis measures
  1. Categories
     - ○ Programming Language
     - ○ OWASP TOP 10
     - ○ PCI-DSS
     - ○ CWE (Common Weakness Enumeration)
  2. Fitness for ML
- ➔ Conclusion

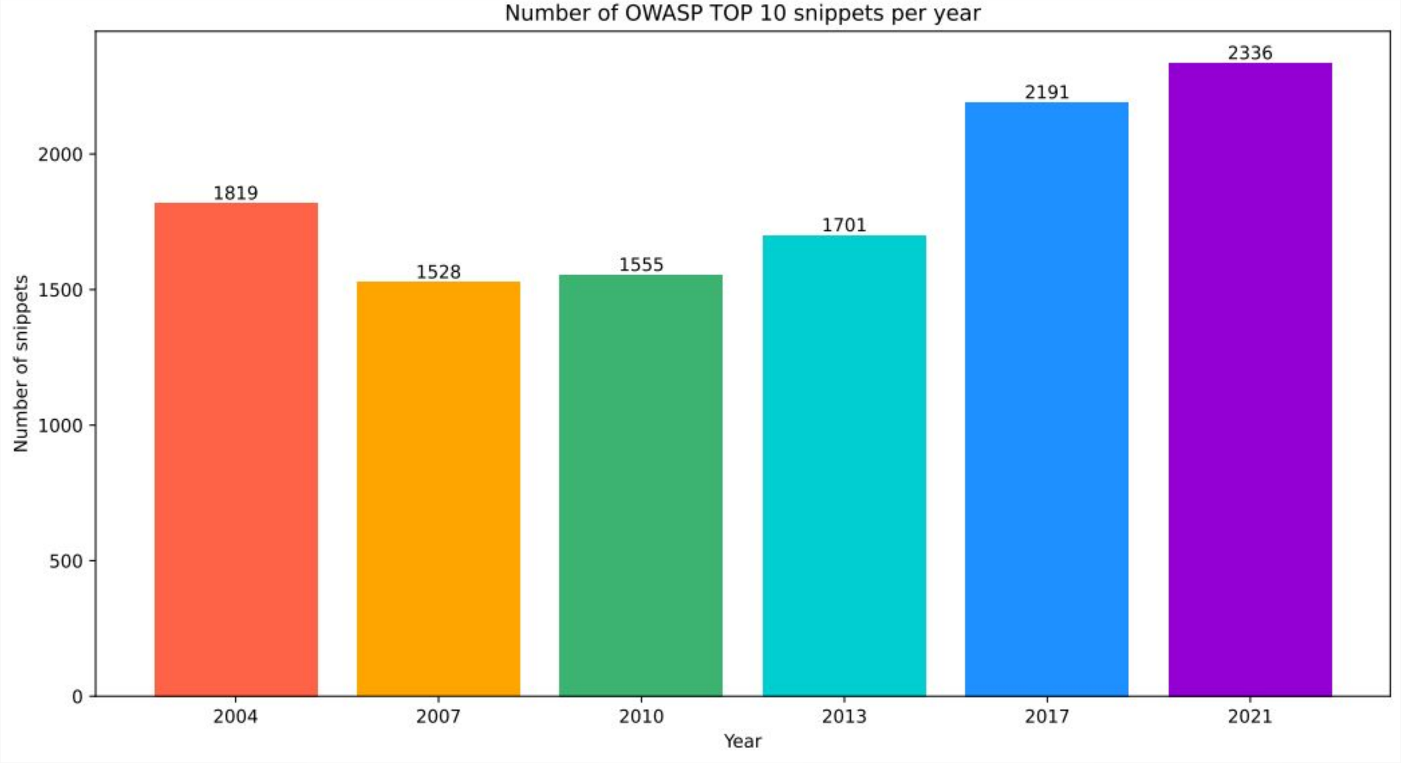
Publicly Available Snippets per Source



Non-Compliant snippets per language



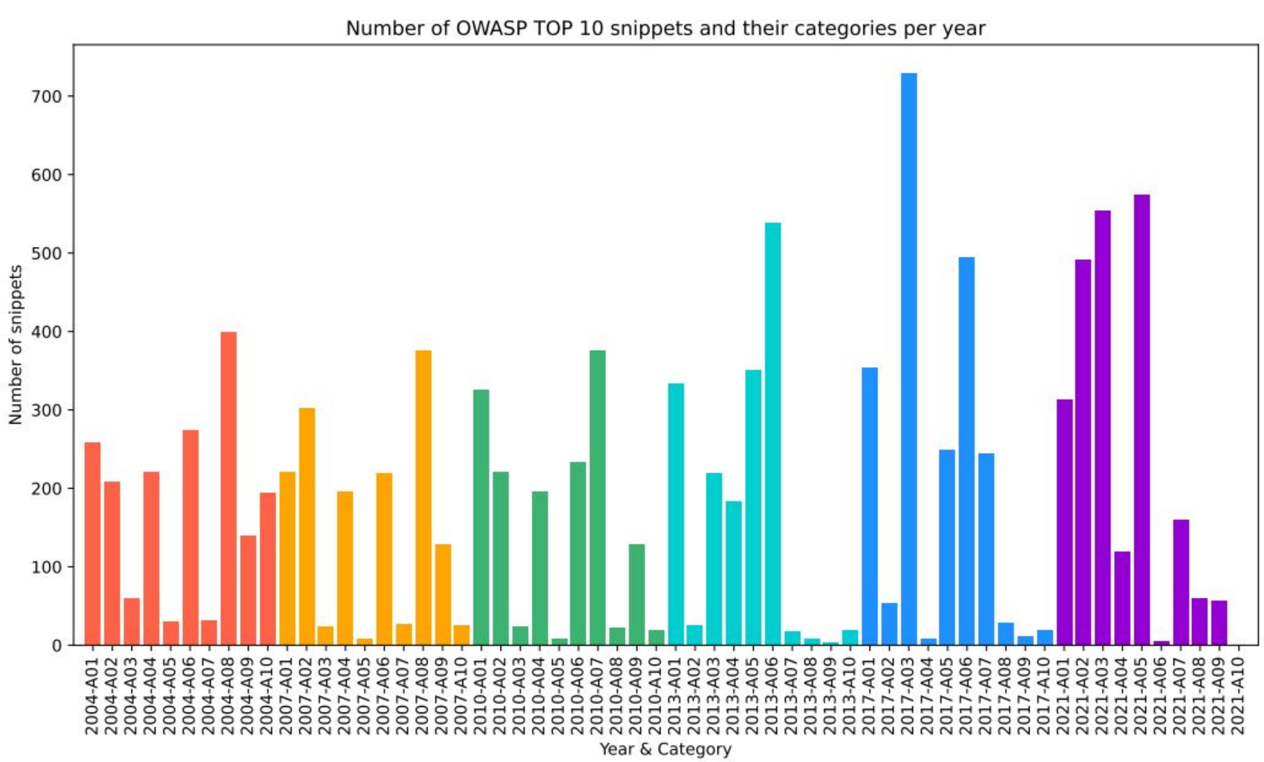
OWASP TOP 10 - Years



OWASP TOP 10 - Years & Categories

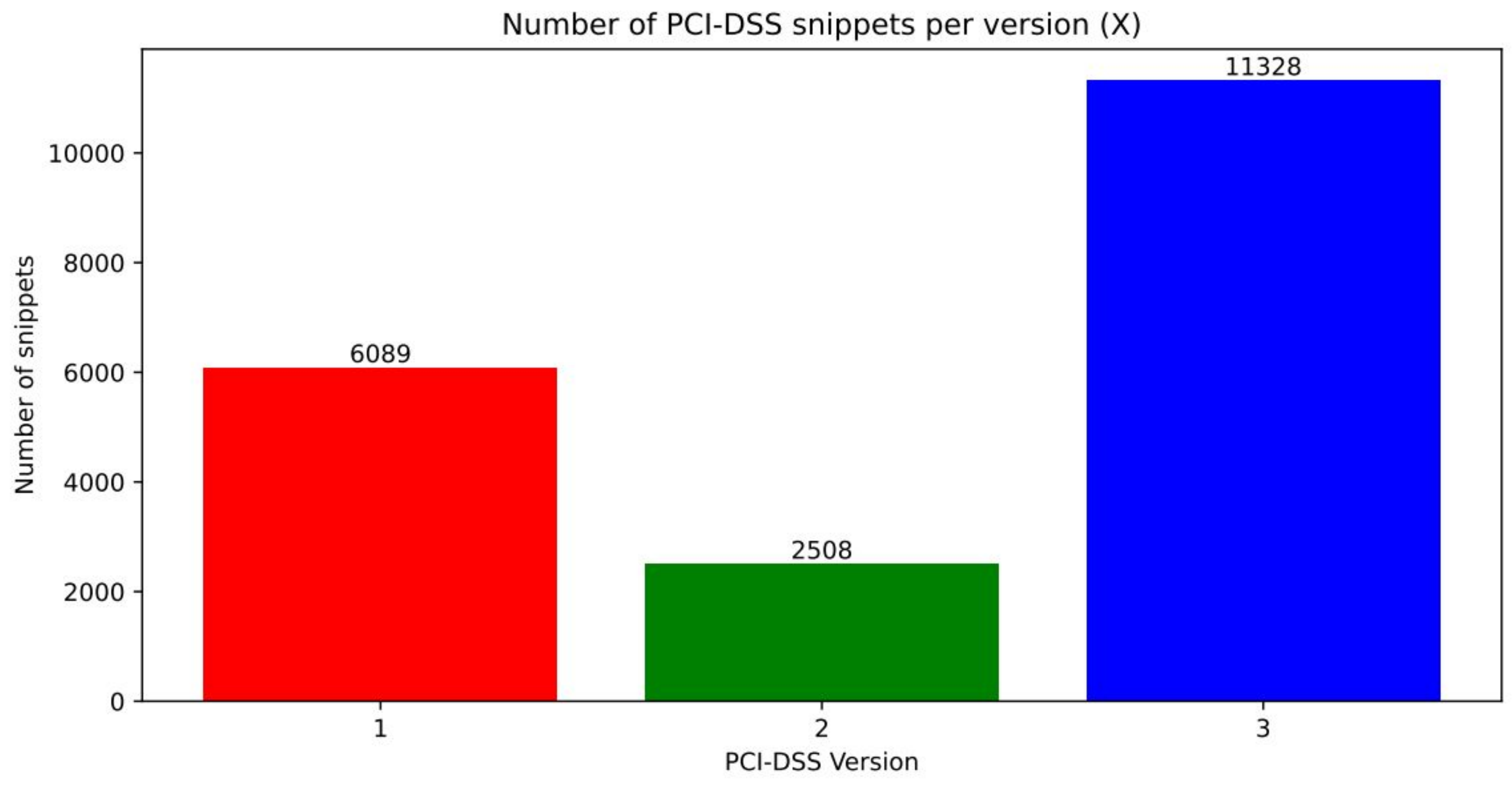
SIEMENS

Ingenuity for life

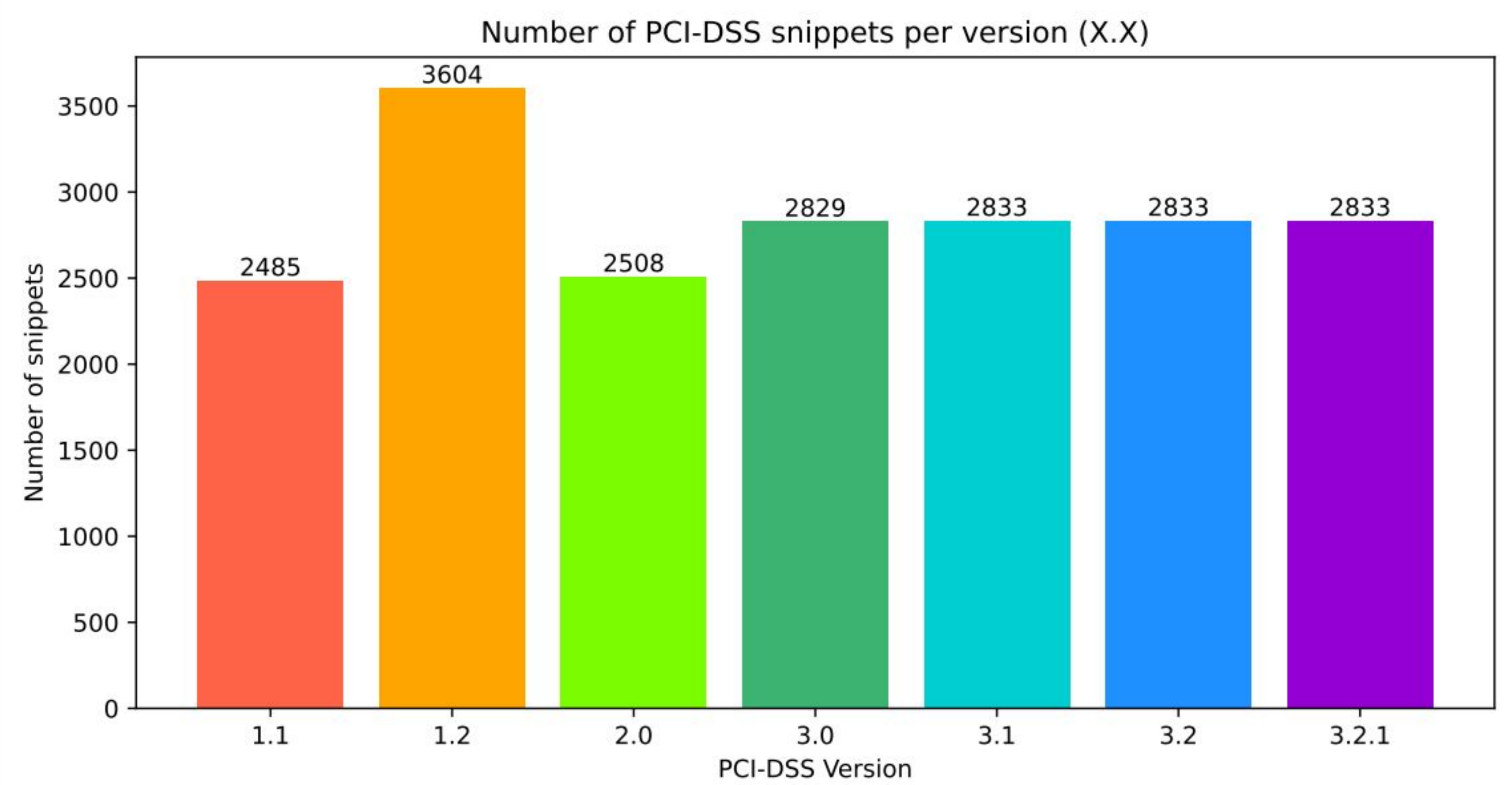


Unrestricted

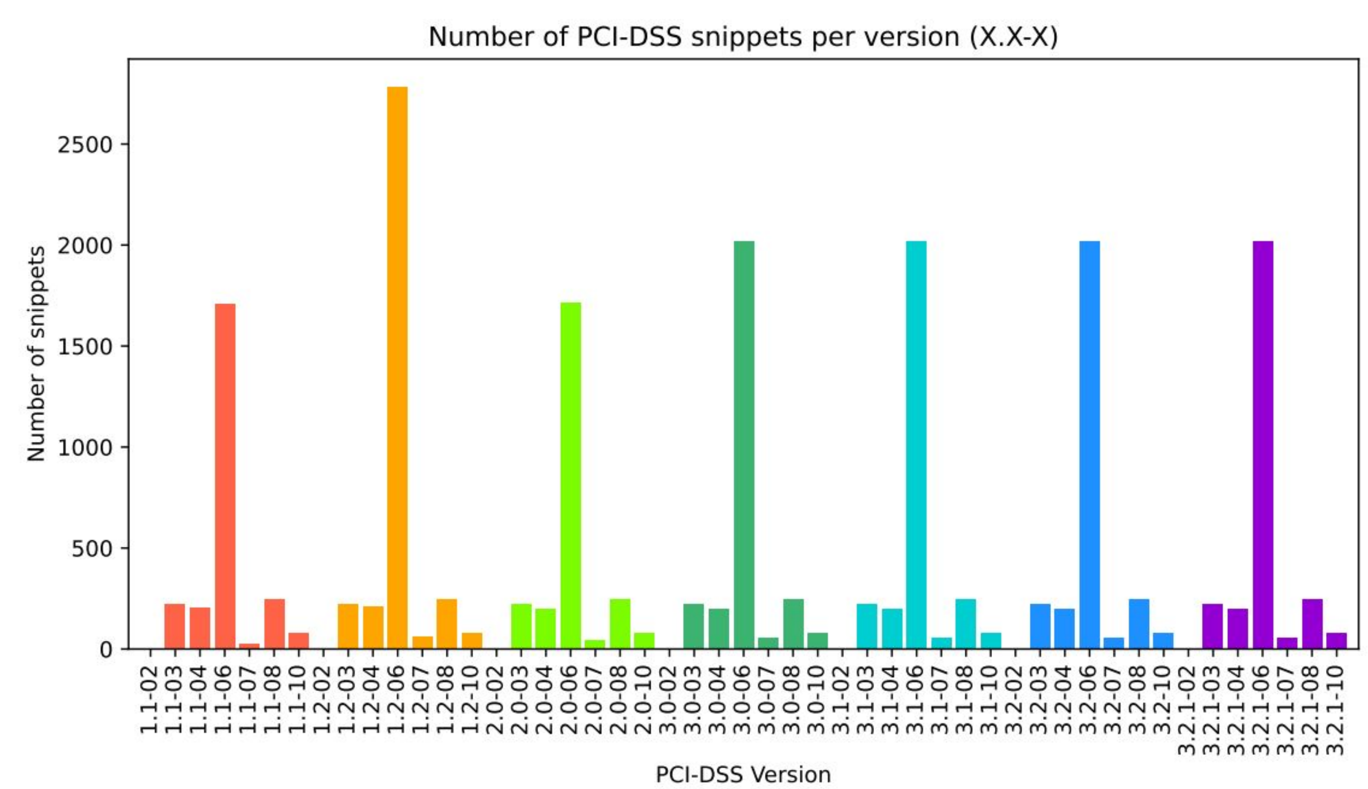
PCI-DSS Version



PCI-DSS Categories



PCI-DSS Sub-Categories



Unrestricted

Category Analysis Conclusion

- ➔ Uniformity of data on high level is not enough

- ➔ Neither snippet count for OWASP TOP 10 nor PCI-DSS is uniform on a sub category analysis

- ➔ Training on snippets for OWASP TOP 10 or PCI-DSS
  - ➔ Results in heavily biased models towards some (sub)categories

Juliet Dataset

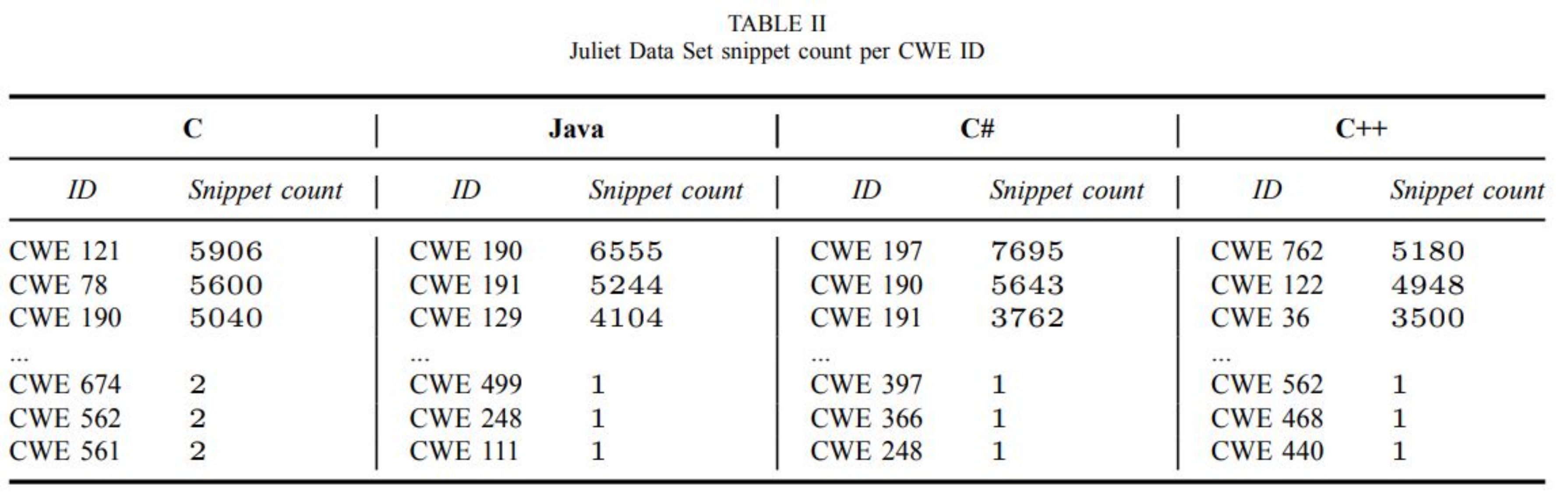


TABLE II
Juliet Data Set snippet count per CWE ID

C		Java		C#		C++	
ID	*Snippet count*	*ID*	*Snippet count*	*ID*	*Snippet count*	*ID*	*Snippet count*
CWE 121	5906	CWE 190	6555	CWE 197	7695	CWE 762	5180
CWE 78	5600	CWE 191	5244	CWE 190	5643	CWE 122	4948
CWE 190	5040	CWE 129	4104	CWE 191	3762	CWE 36	3500
...		...		...		...	
CWE 674	2	CWE 499	1	CWE 397	1	CWE 562	1
CWE 562	2	CWE 248	1	CWE 366	1	CWE 468	1
CWE 561	2	CWE 111	1	CWE 248	1	CWE 440	1

Juliet Dataset Analysis Conclusion

- ➔ Has huge number of snippet examples
- ➔ Very valuable resource for testing tools
- ➔ Not good for training Machine learning models
    - ◆ Underlying snippet bias

Conclusion

➔ Where can code snippets be found?

- 11 possible sources of information
- Not all represented the same (most prominent: C, Java, C#, C++)

➔ What is the quality of the code snippets?

- Varies with the programming language
- Within a programming language → imbalance between vulnerability categories

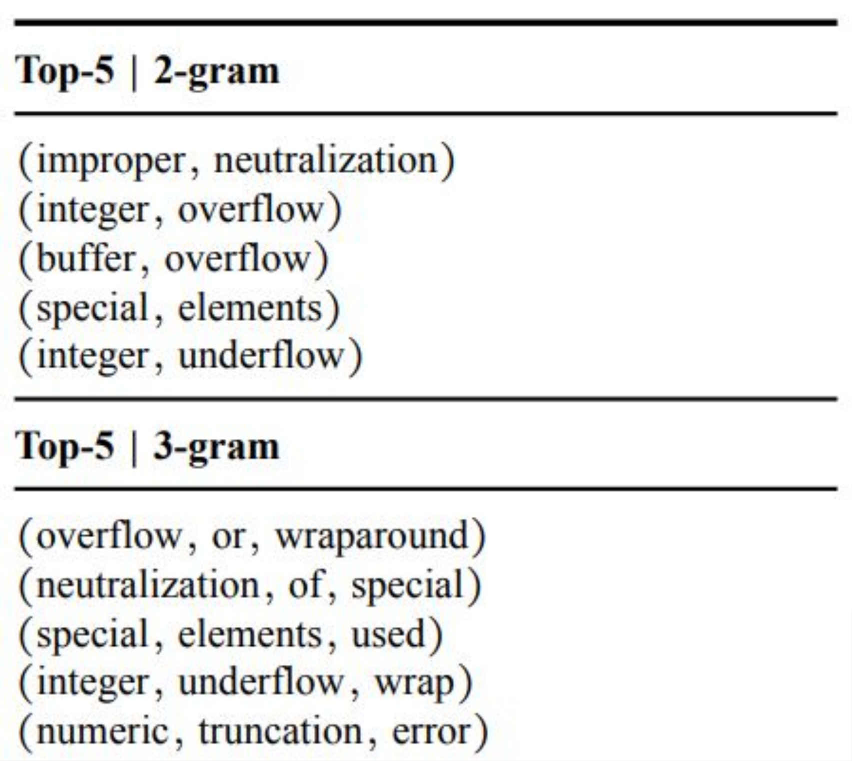
Main conclusion:

- Some programming languages hugely underrepresented
- Juliet set - mostly synthetic data
- Not clear how good the snippets are to train ML models

Further work:

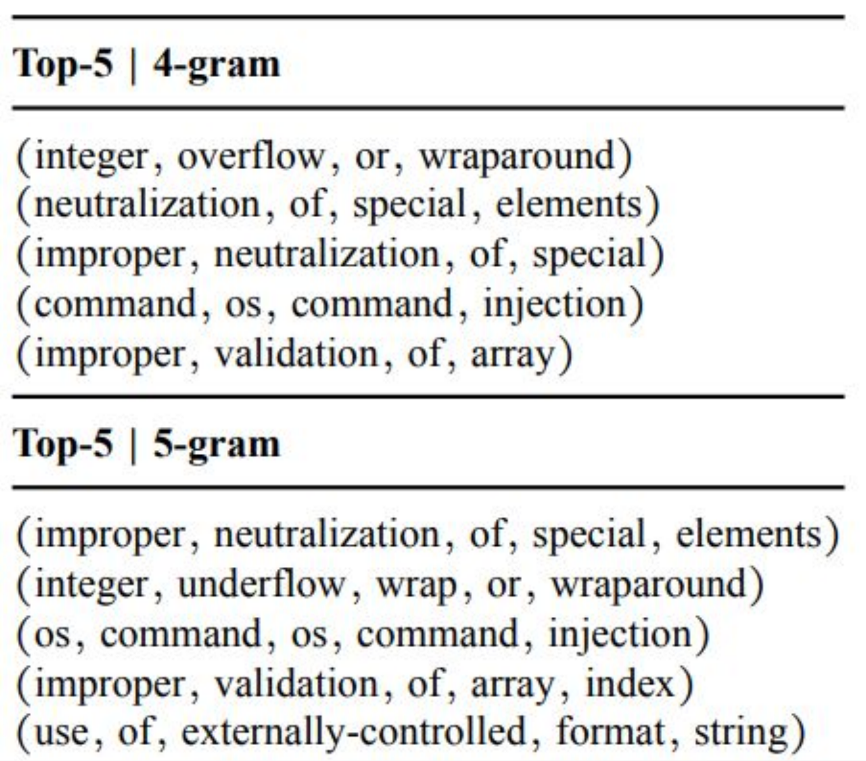
- Investigate “real-world” code snippets based on check-in comment

Keyword Occurrences



Top-5 | 2-gram
(improper, neutralization)
(integer, overflow)
(buffer, overflow)
(special, elements)
(integer, underflow)

Top-5 | 3-gram
(overflow, or, wraparound)
(neutralization, of, special)
(special, elements, used)
(integer, underflow, wrap)
(numeric, truncation, error)



Top-5 | 4-gram
(integer, overflow, or, wraparound)
(neutralization, of, special, elements)
(improper, neutralization, of, special)
(command, os, command, injection)
(improper, validation, of, array)

Top-5 | 5-gram
(improper, neutralization, of, special, elements)
(integer, underflow, wrap, or, wraparound)
(os, command, os, command, injection)
(improper, validation, of, array, index)
(use, of, externally-controlled, format, string)

information leak
query ldap
elements used
information cleartext
management routines
control process
path element
scripting poor
splitting improper
weak cryptographic
incorrect check
path manipulation
integer underflow
index improper
null pointer
truncation error
command sql
access control
system information
scripting reflected
management android
return value
read buffer
loop condition
value memory
pointer dereference
resource consumption
cryptographic hash
injection sql
external control
java confluence
consumption uncontrolled
cookie security
special element
integer overflow
expressions xpath
free double
bad practice
web page
ldap query
error numeric
key key
uncontrolled resource
encryption key
password password
reflected cross
privacy violation
configuration setting
related html
heap free
format string
key management
uninitialized variable
without use
validation cross
ldap injection
improper validation
based buffer
management empty
basic xss
management hardcoded
value incorrect
insecure storage
dereference unchecked
code unsafe
single metavariable
command injection
weak encryption
sensitive information
underwrite buffer
crlf sequences
encryption insecure
buffer free
controlled input
yaml semgrep_slow
use assignment
buffer underflow
traversal relative
injection improper
underflow wrap
violation privacy
function return
element uncontrolled
privilege management
size value
excessive size
sql injection
untrusted site
controlled format
component communication
condition unchecked
sign extension
double free
sql command
page improper
page basic
buffer overflow
key size
absolute path
error message
semgrep_slow pattern
wraparound integer
cert oracle
search path
verification disabled
externally controlled
heap based
unchecked input
site open
xss improper
sei cert
process control
empty password
missing release
effective lifetime
start
user controlled
dereference null
array index
unchecked return
cross sitestring use
cert c
routines mismatched
memory allocation
redirect url
error signed
buffer underwrite
scripting inter
reflection use
test yaml
size weak
log forging
open redirect
standard confluence
stack based
improper neutralization
coding standard
attribute
c coding
overflow heap
overflow integer
salt weak
site scripting
variable use
insufficient key
oracle coding
script related
password management
metavariable test
extension unexpected
response splitting
command os
hard coded
unsigned conversion
insecure ssl
cleartext storage
select classes
pattern single
path traversal
header
mismatched memory
numeric truncation
persistent cross
relative path
within xpath
underflow buffer
html tags
uncontrolled search
scripting persistent
insecure transport
injection command
traversal absolute
security sensitive
inter component
conversion error
management null
value unchecked
data within
os command
overflow stack
memory management
zero divide
lifetime missing
poor validation
xpath expressions
setting external
script injection
xpath injection
pbe password
overly broad
url redirection
unsafe reflection
unexpected sign
manipulation path
message web

Thank you for Listening 😊

Kaan Oguzhan

kaan-oguzhan@siemens.com

Dr. rer. nat. Tiago Espinha Gasiba

tiago.gasiba@siemens.com

Akram Louati

akram.louati@siemens.com