Predict security vulnerabilities in FOSS

Why you want it and how to do it

Name: Carlos E. Budde
Inst: Università di Trento, Italy
Date: 3rd November 2023
Software's vulnerable lifecycle

d
Jul'22
Aug'22
Sep'22
Oct'22
Nov'22
Dec'22

time

org.redisson:redisson

3.17.5

io.netty:netty-codec

4.1.79
Software’s vulnerable lifecycle

- org.redisson:redisson
  - 3.17.5
- io.netty:netty-codec
  - 4.1.79
  - 4.1.80

时间：
- Jul'22
- Aug'22
- Sep'22
- Oct'22
- Nov'22
- Dec'22

问题1：
- [1. Da problem]$_ _$
>>> Software’s vulnerable lifecycle

org.redisson:redisson
3.17.5
ℓ
α
ℓ
α
3.17.6

io.netty:netty-codec
4.1.79
ℓ
d
4.1.80
ℓ
d

Sep'22 Aug'22 Jul'22 Dec'22 Oct'22 Nov'22

time

Jul'22 Aug'22 Sep'22 Oct'22 Nov'22 Dec'22

time

[1. Da problem]$ _

[2/6]
Software’s vulnerable lifecycle

org.redisson:redisson
3.17.5
ℓ
a
ℓ
3.17.6

io.netty:netty-codec
4.1.79
dl
4.1.80
dl
4.1.81
dl
4.1.82
dl

Jul’22 Aug’22 Sep’22 Oct’22 Nov’22 Dec’22

time

[1. Da problem]$ _

[2/6]
Software’s vulnerable lifecycle

![Diagram showing the vulnerable lifecycle of software with specific versions and dates]

- **org.redisson:redisson**
  - Version: 3.17.5
  - Date: Sep'22
  - Version: 3.17.6
  - Date: Aug'22
  - Version: 3.17.7
  - Date: Jul'22

- **io.netty:netty-codec**
  - Version: 4.1.79
  - Date: Dec'22
  - Version: 4.1.80
  - Date: Nov'22
  - Version: 4.1.81
  - Date: Oct'22
  - Version: 4.1.82
  - Date: Sep'22
Software’s vulnerable lifecycle
Software’s vulnerable lifecycle

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86)
Software’s vulnerable lifecycle

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86)
Software’s vulnerable lifecycle

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86)
Forced
MOVE
Wrong
MOVE
Correct
STAY
Correct
MOVE
Correct
STAY

org.redisson:redisson
3.17.5
ℓ
a
ℓ
3.17.6
3.17.5
ℓ
a
ℓ
3.17.7
3.18.0
ℓ
a
3.18.1
3.17.7
ℓ
a
3.19.0
io.netty:netty-codec
4.1.79
ℓ
4.1.80
4.1.81
4.1.82
4.1.83
4.1.84
4.1.85
4.1.86

Jul'22 Aug'22 Sep'22 Oct'22 Nov'22 Dec'22

CVE-2022-41915 disclosed!
⇒ affects netty [4.1.83, 4.1.86)
Is there a **best time** to update?

[1. Da problem]$_{2/6}$
Security vs. safety
Security vs. safety

* Events are catastrophic
  - Vulnerability exploitation == death
Security vs. safety

* Events are catastrophic
  - Vulnerability exploitation == death

* Events are rare
  - Very few files in very few libraries are vulnerable
Security vs. safety

* Events are catastrophic
  - Vulnerability exploitation == death

* Events are rare
  - Very few files in very few libraries are vulnerable

* No good health indicators
  - Vulnerability exposed ⇒ insta kill
Security vs. safety

* Events are catastrophic
  - Vulnerability exploitation == death

* Events are rare
  - Very few files in very few libraries are vulnerable

* No good health indicators
  - Vulnerability exposed ⇒ insta kill

Can we bridge the gap to use PdM for security?
Idea: fit CVE disclosure time
Idea: fit CVE disclosure time
Idea: fit CVE disclosure time

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86)
Idea: fit CVE disclosure time

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86]
Idea: fit CVE disclosure time

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86)
Idea: fit CVE disclosure time

CVE-2022-41915 disclosed!
affects netty [4.1.83, 4.1.86)

Idea: fit CVE disclosure time

```
Idea: fit CVE disclosure time

>>> Idea: fit CVE disclosure time

Time since lib. release
Probability of CVE public.

1M    2M    3M    4M    ...

[2. Proposal]
```
Idea: fit CVE disclosure time
But:

* Vulnerabilities are rare events ⇒ very few to fit from
But:

* Vulnerabilities are rare events $\Rightarrow$ very few to fit from

* Not every vulnerability (or library) is equivalent
But:

* Vulnerabilities are rare events ⇒ very few to fit from
  - Aggregate vulnerabilities from many libraries

* Not every vulnerability (or library) is equivalent
>>> But:

* Vulnerabilities are rare events $\Rightarrow$ very few to fit from
  - Aggregate vulnerabilities from many libraries

* Not every vulnerability (or library) is equivalent
  - Partition software libraries per type
But:

* Vulnerabilities are rare events $\Rightarrow$ very few to fit from
  - Aggregate vulnerabilities from many libraries

* Not every vulnerability (or library) is equivalent
  - Partition software libraries per type

* **Software features** for classification
  that are **relevant for security**
But:

* Vulnerabilities are rare events $\Rightarrow$ very few to fit from
  - Aggregate vulnerabilities from many libraries

* Not every vulnerability (or library) is equivalent
  - Partition software libraries per type

* **Software features** for classification
  that are **relevant for security**
But:

* Vulnerabilities are rare events $\Rightarrow$ very few to fit from
  - Aggregate vulnerabilities from many libraries

* Not every vulnerability (or library) is equivalent
  - Partition software libraries per type

* Software features for classification that are relevant for security
But:

- Vulnerabilities are rare events ⇒ very few to fit from
  - Aggregate vulnerabilities from many libraries

- Not every vulnerability (or library) is equivalent
  - Partition software libraries per type

- Software features for classification that are relevant for security
* Vulnerabilities are rare events ⇒ very few to fit from
  - Aggregate vulnerabilities from many libraries

* Not every vulnerability (or library) is equivalent
  - Partition software libraries per type

* **Software features** for classification that are **relevant for security**
Predict security vulnerabilities in FOSS

Why you want it and how to do it

Name: Carlos E. Budde†
Inst: Università di Trento, Italy

tcarlosesteban.budde@unitn.it