Vulnerability Prioritization

Hhmmm...

1. Analyze vulnerability
   - Assess injection, impact, existence of exploits, ...
   - Assess **contextual risk** for each instance
2. **Choose** action on each instance
3. **Take** action on each instance

- **Hundreds** of **new** vulnerabilities **each week**
- How to cope with **all** of them?

Vulnerability Management in Organizations (REMIND)

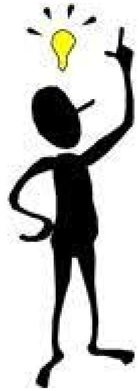
- Vulnerability **Management:**
  - …

- Fundamental components
  - **Asset Management**
    - Which systems
    - Who is in charge
  - **Vulnerability Prioritization**
    - How to allocate defensive efforts to CVEs

Basic Fact #1 (REMIND)

- **Very few CVEs are actually exploited**
- Just to have an idea: **≈5%** of all CVEs **(!)**

Focus on those!



Basic Fact #2 (REMIND)

- CVSS is **not** a good predictor of which vulnerabilities will be **actually exploited**
- Predicting which vulnerabilities will be exploited is a **huge open problem**

- Every predictor you can think of turns out to be
  - "Low" precision:
    You worry about many vulns **unnecessarily**
  - "Low" recall:
    You wrongly **neglect** many vulns

Our path

- ❑ Exploit Prediction Scoring System (**EPSS**)

- ❑ Considered the "state of the art"
- ❑ …but it does have many limitations

- ❑ We will see:
  - ❑ How it works
  - ❑ Public data on its assessment

  - ❑ Some of its limitations

Exploit Prediction: Problem Definition

Exploit Probability: Definition (I)

- Vulnerability `CVE-i`
- `P(CVE-i,` **`d`**`):`

$$\frac{\text{\# Exploitation }\textbf{attempts}\text{ of CVE-i in [}\mathbf{d}\text{,}\mathbf{d}\text{+30]}}{\text{\# Expolitation }\textbf{attempts}\text{ of }\textbf{all}\text{ CVEs in [}\mathbf{d}\text{,}\mathbf{d}\text{+30]}}$$

- Probability that `CVE-i` will be exploited in the next 30 days

- **Against whom?**
- **How computed?**

Exploit Probability: Definition (II)

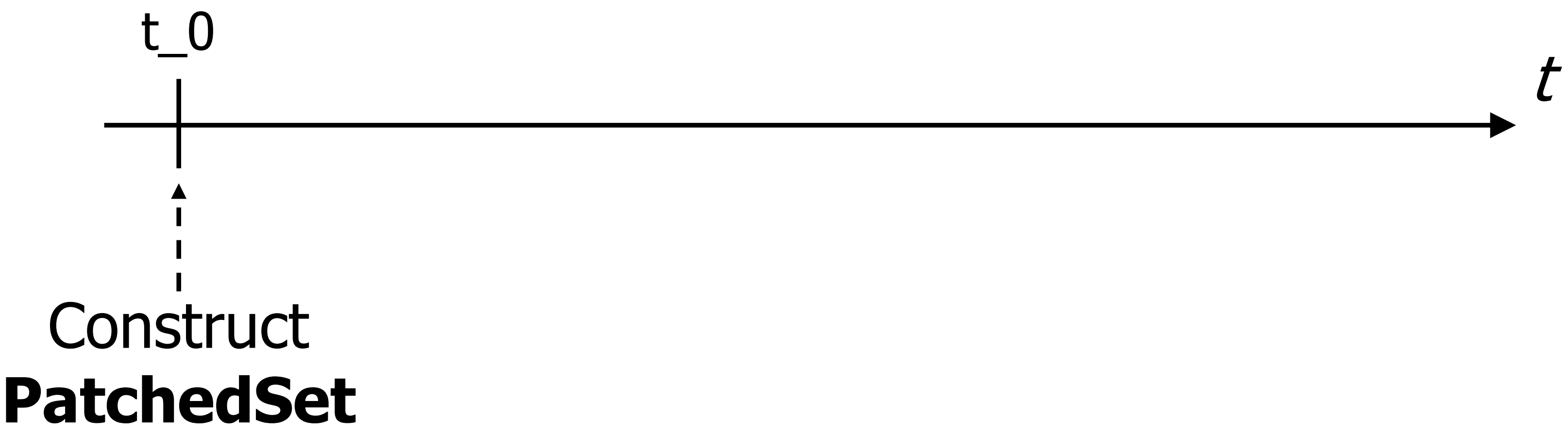
- Vulnerability `CVE-i`
- `P(CVE-i, d):`

$$\frac{\#\ \text{Exploitation } \textbf{attempts} \text{ of } \texttt{CVE-i} \quad \text{in } [\mathbf{d},\mathbf{d}+30]}{\#\ \text{Expolitation } \textbf{attempts} \text{ of } \textbf{all } \texttt{CVE}\text{s in } [\mathbf{d},\mathbf{d}+30]}$$

- Probability that `CVE-i` will be exploited in the next 30 days

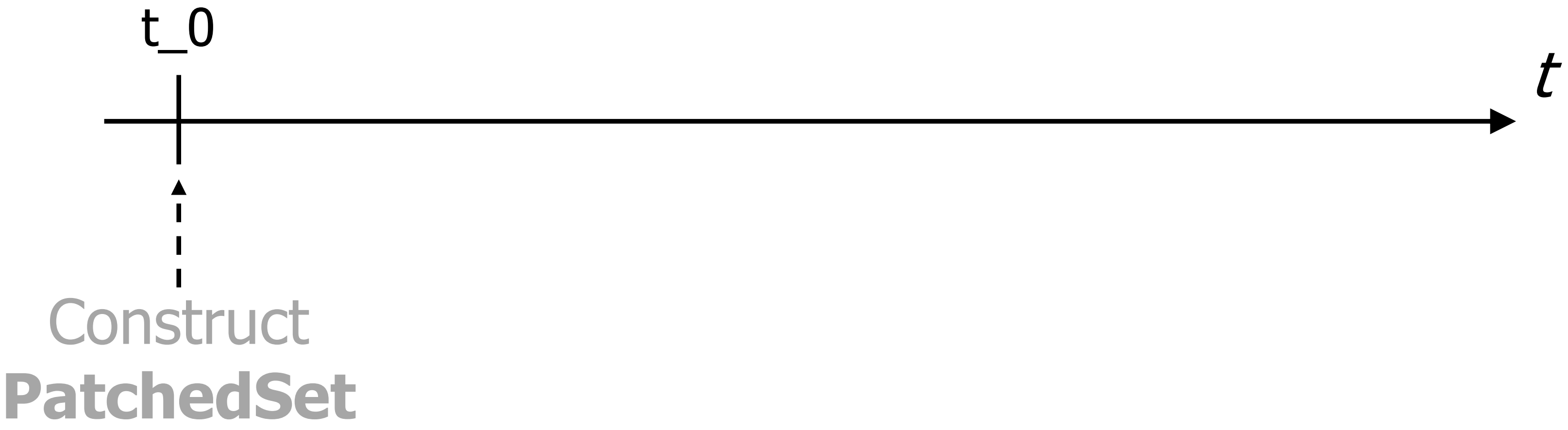
- **Worldwide** (everywhere)
- **Approximated** by collecting many TI feeds
- Computed **a posteriori**

Exploit Prediction: Problem Definition (I)



- We define a criterion for choosing **which vulnerabilities to patch**
- Subset of **all known vulns** at t_0

Many possible criteria

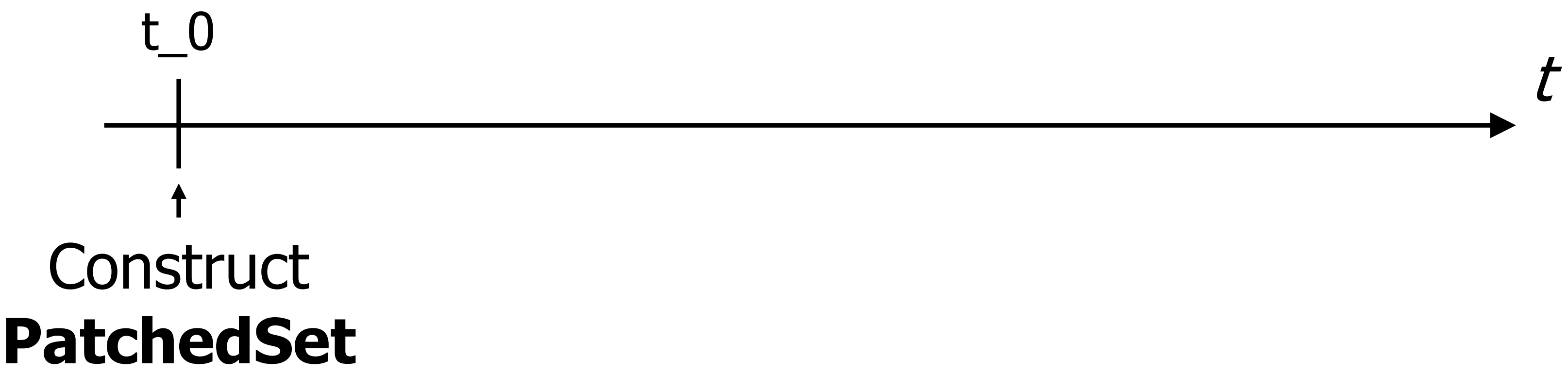

t_0

t

Construct
PatchedSet

- □ All vulns with CVSS >= 9 (Critical)
- □ All vulns with remote injection
- □ All vulns of Windows software
- □ ...

Remark



t_0

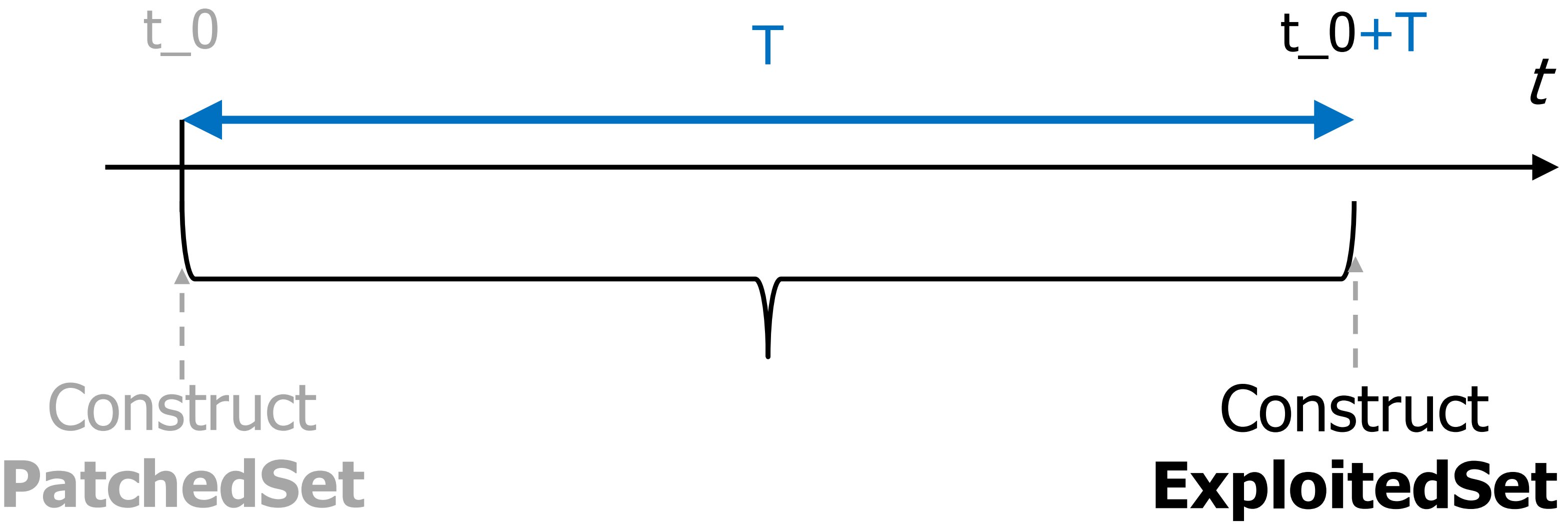
t

Construct
PatchedSet

- Subset of **all known vulns** at t_0

- An organization should focus **only** on vulns on **its** systems (and their **risk**)
- We are **pretending** all the known vulns are **equally relevant**

Exploit Prediction: Problem Definition (II)



t_0

T

t_0+T

t

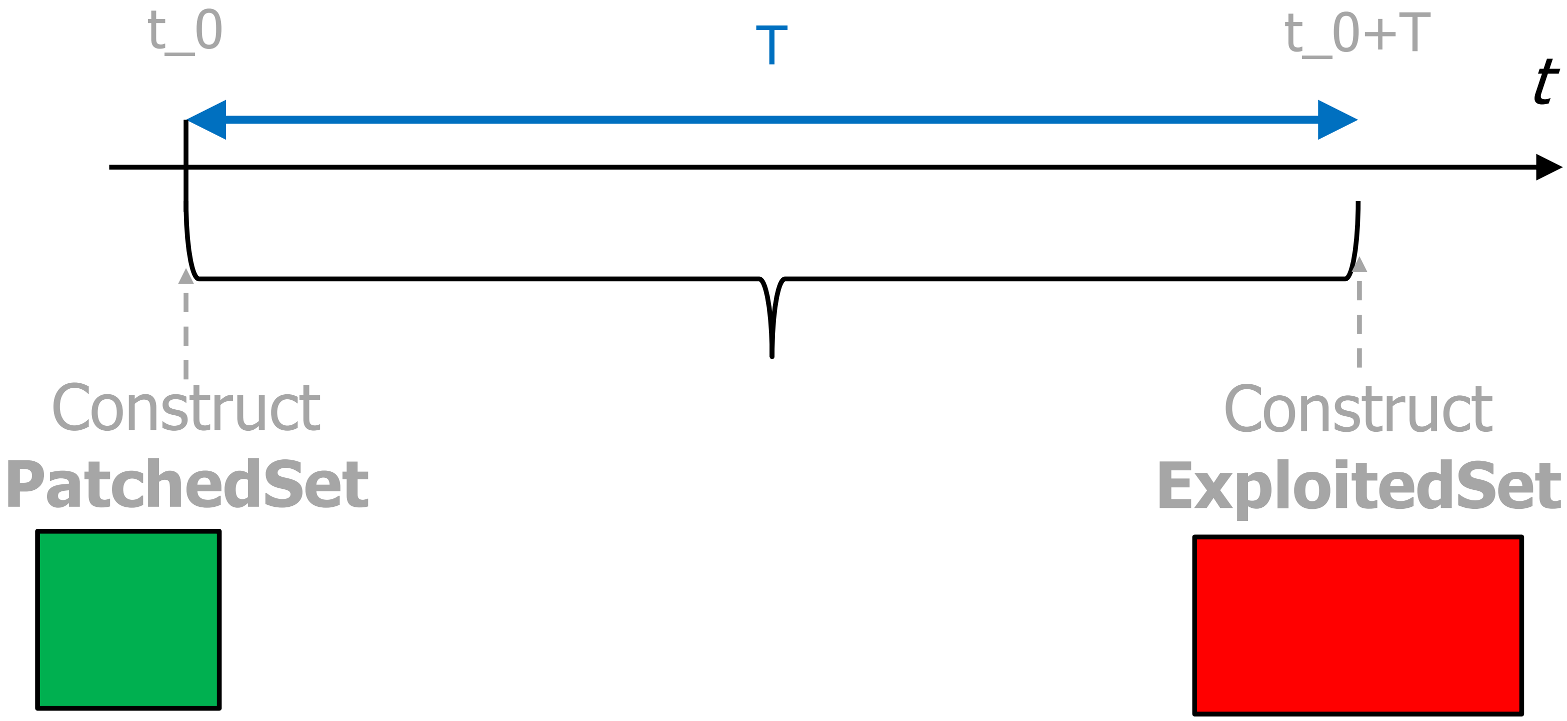
Construct **PatchedSet**

Construct **ExploitedSet**

- Vulnerabilities that have been **actually exploited worldwide** in T

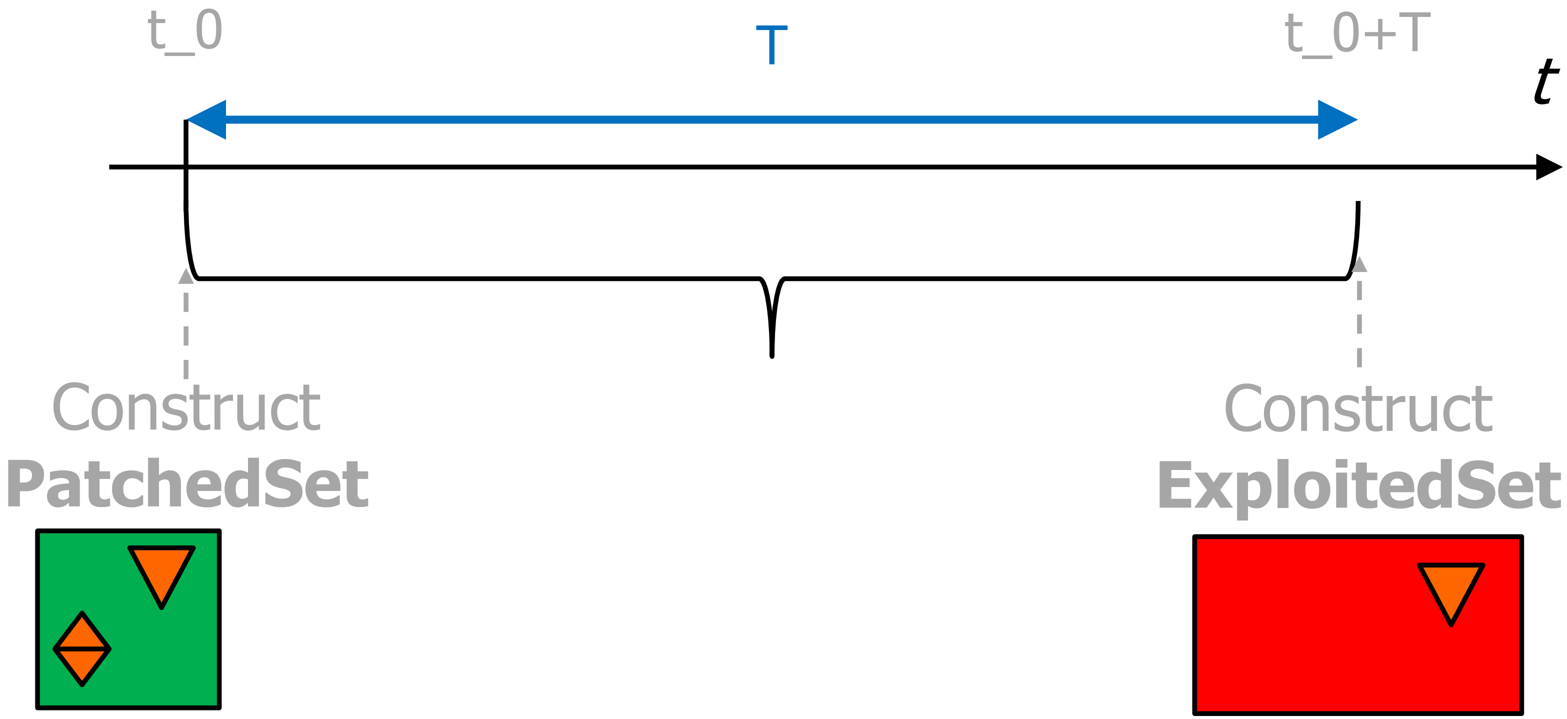
- Those with `P(CVE-i)≠0` (on some day in `[t_0,t_0+T]`)

Exploit Prediction: Problem Definition (III)



- We "compare" PatchedSet vs ExploitedSet

Efficiency (Precision) (I)

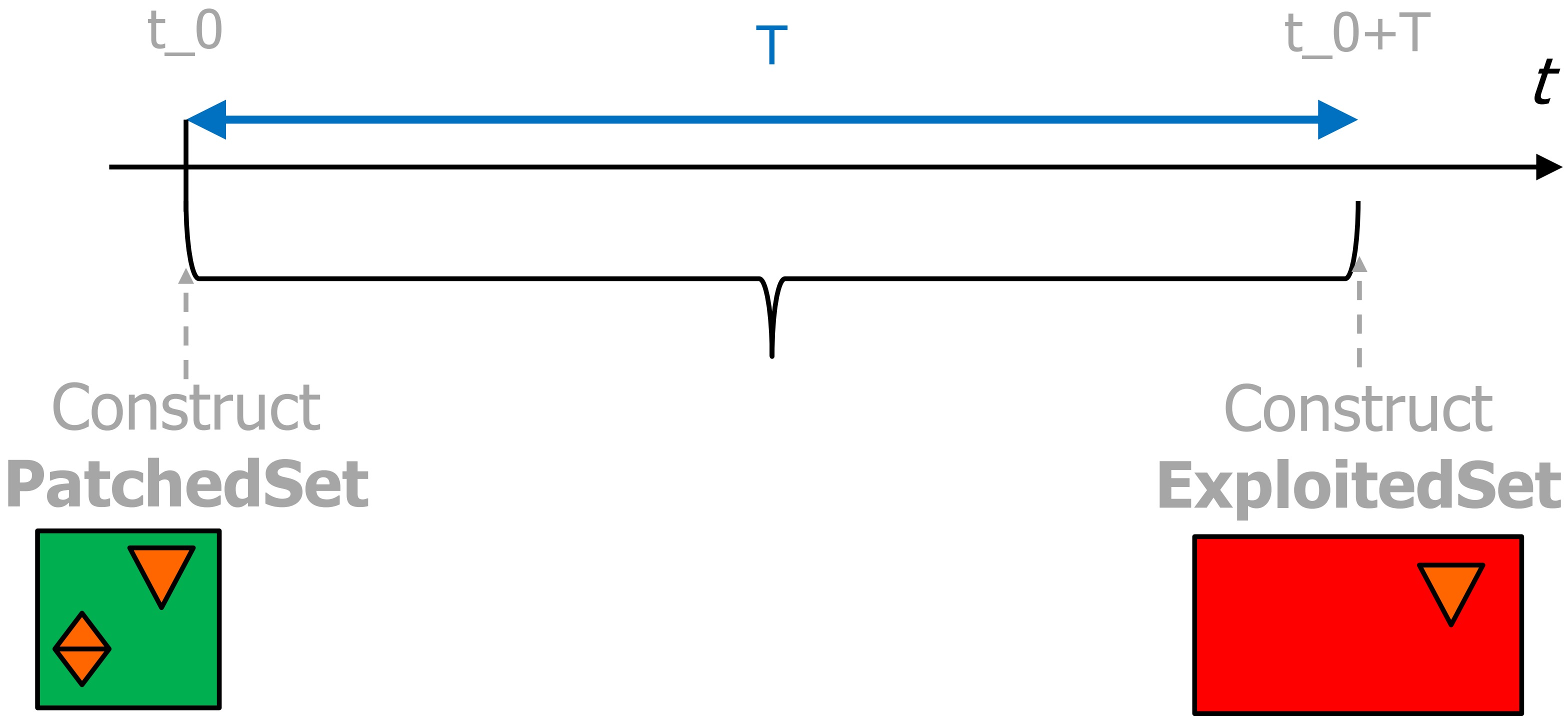


Construct **PatchedSet**

Construct **ExploitedSet**

How many Patched vulns will be Exploited?

Efficiency (Precision) (II)



t_0

T

t_0+T

t

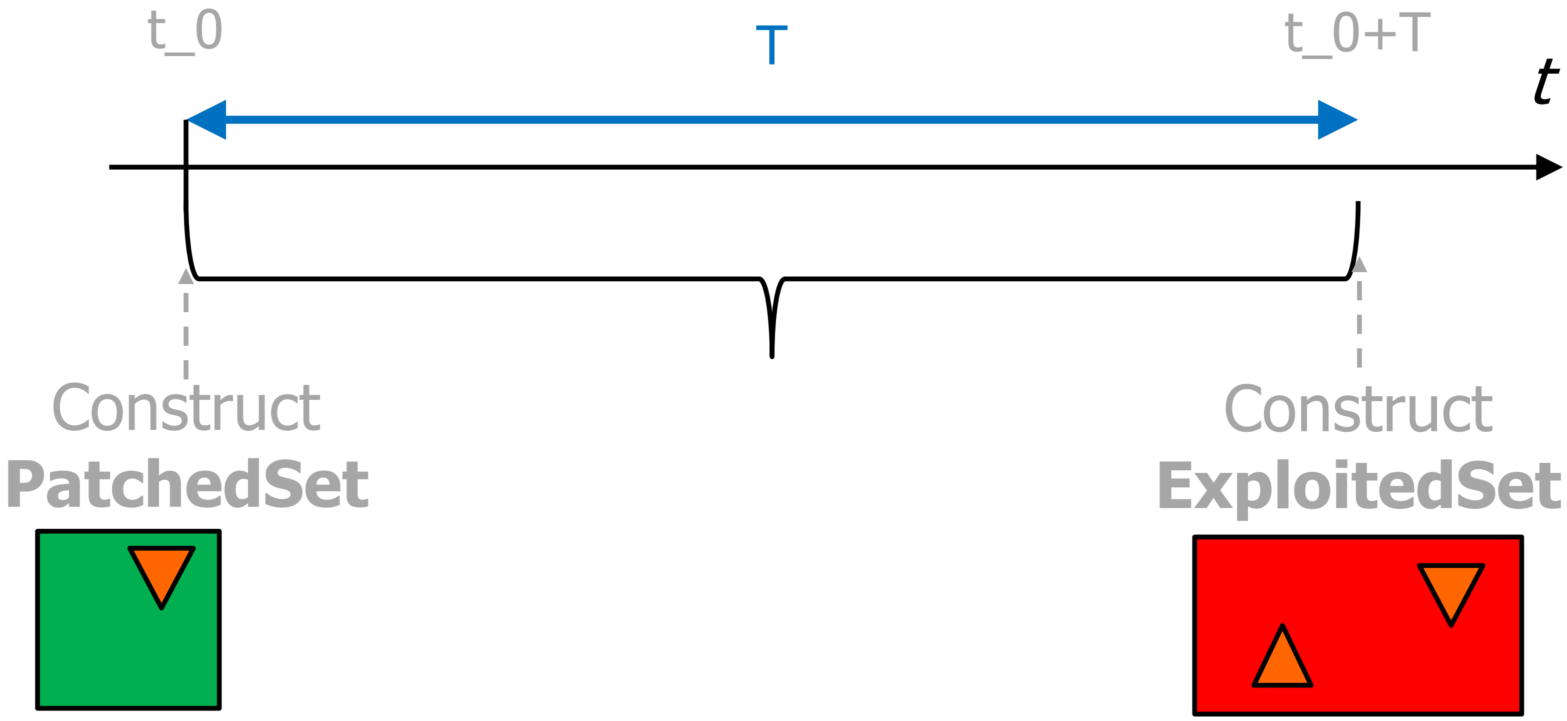
Construct **PatchedSet**

Construct **ExploitedSet**

Efficiency

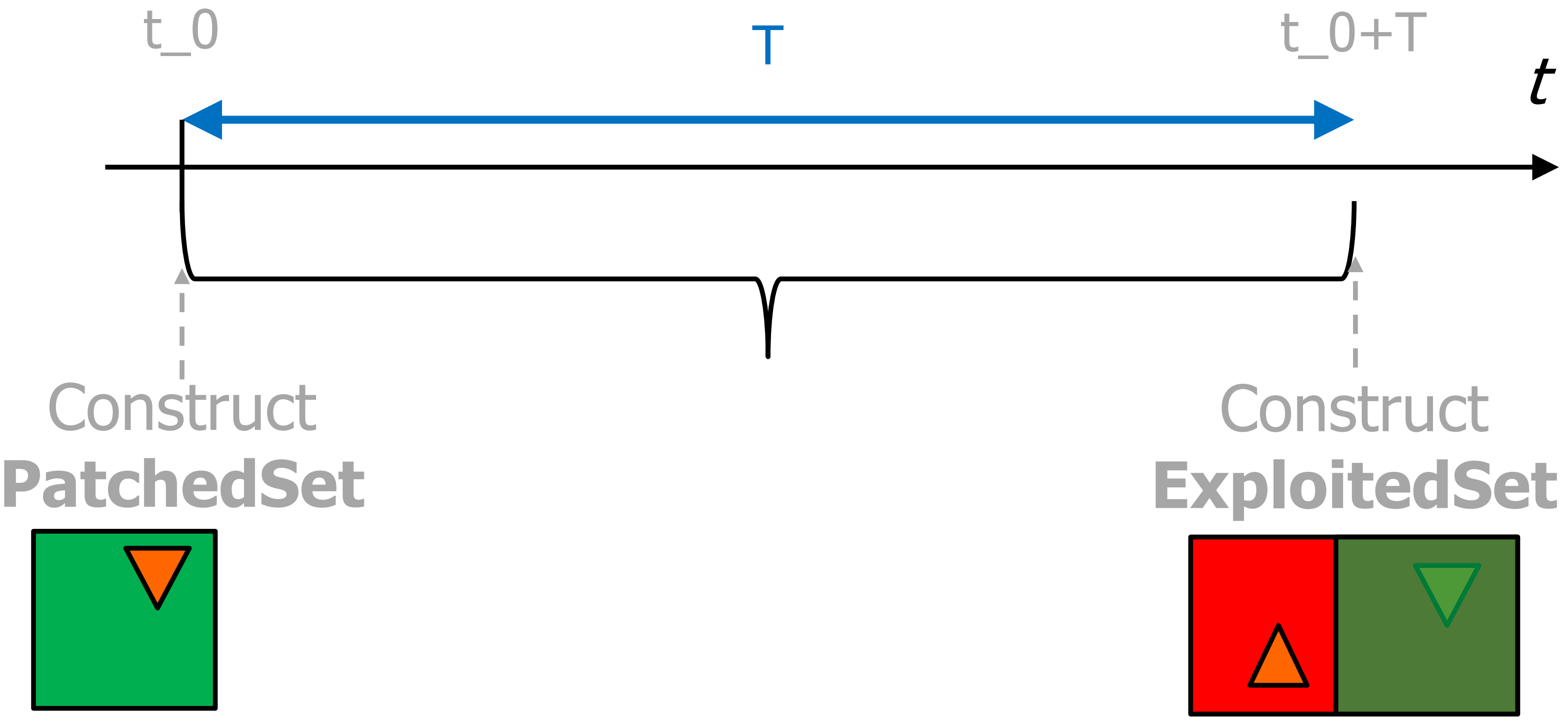
$$\frac{\#(\text{Patched }\textbf{and Exploited})}{\#(\text{Patched})}$$

Coverage (Recall) (I)



How many Exploited vulns we Patched?

Coverage (Recall) (II)



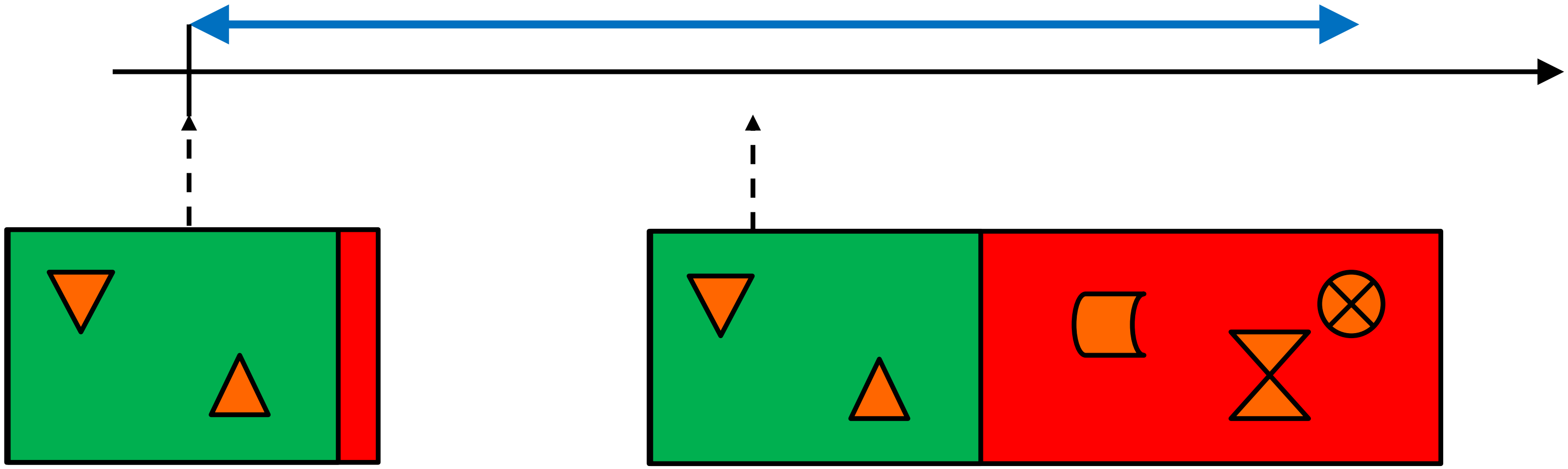
Construct **PatchedSet**

Construct **ExploitedSet**

Coverage:

$$\frac{\#(\text{Exploited } \textbf{and Patched})}{\#\ (\text{Exploited})}$$

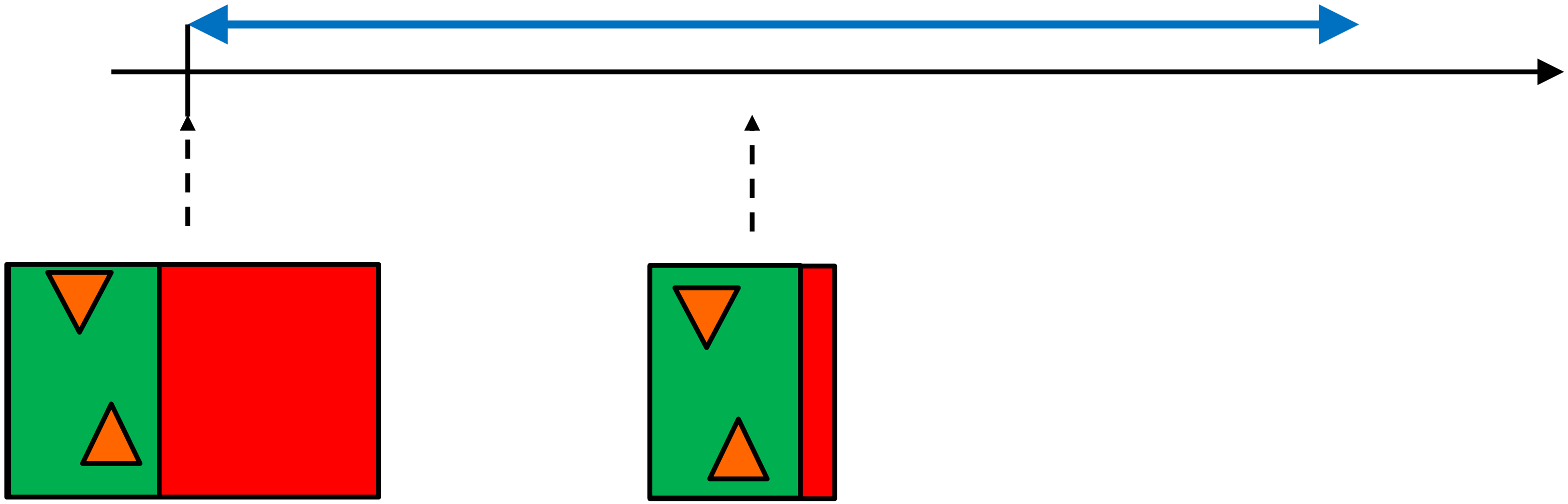
High Efficiency / Low Coverage



I have patched mostly vulns that matter

...but I have missed many vulns

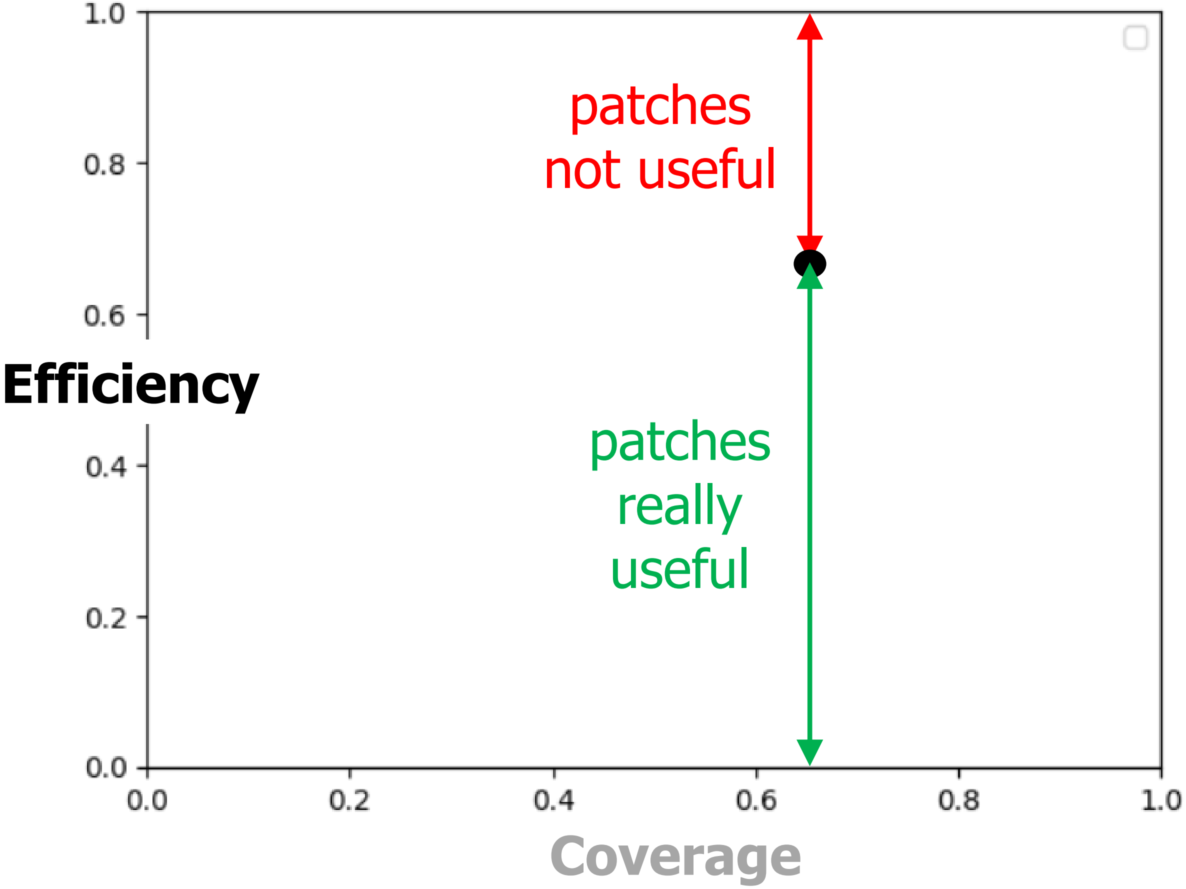
Low Efficiency / High Coverage



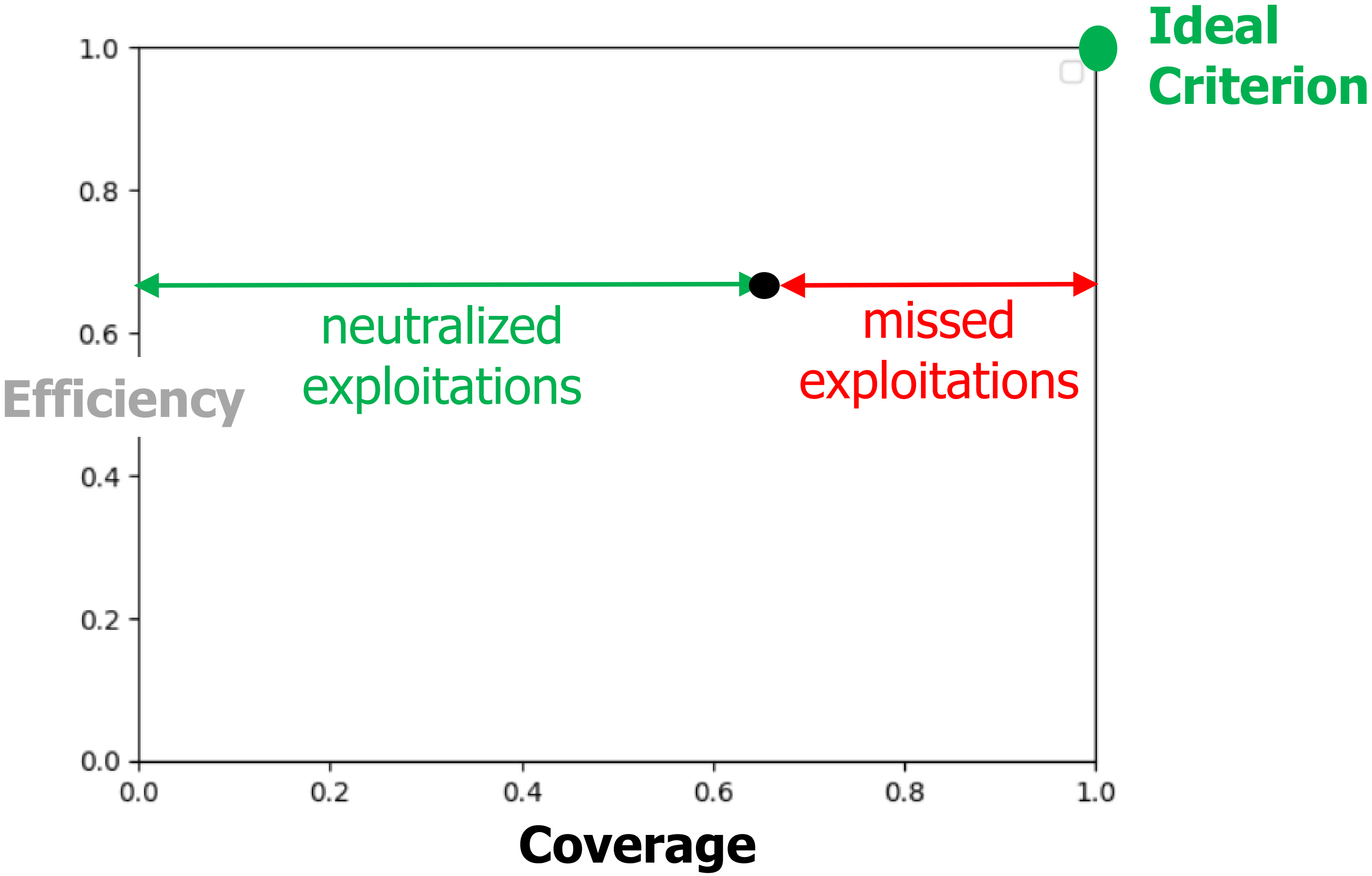
I have wasted lot of patching effort

...but I have covered nearly all vulns that matter

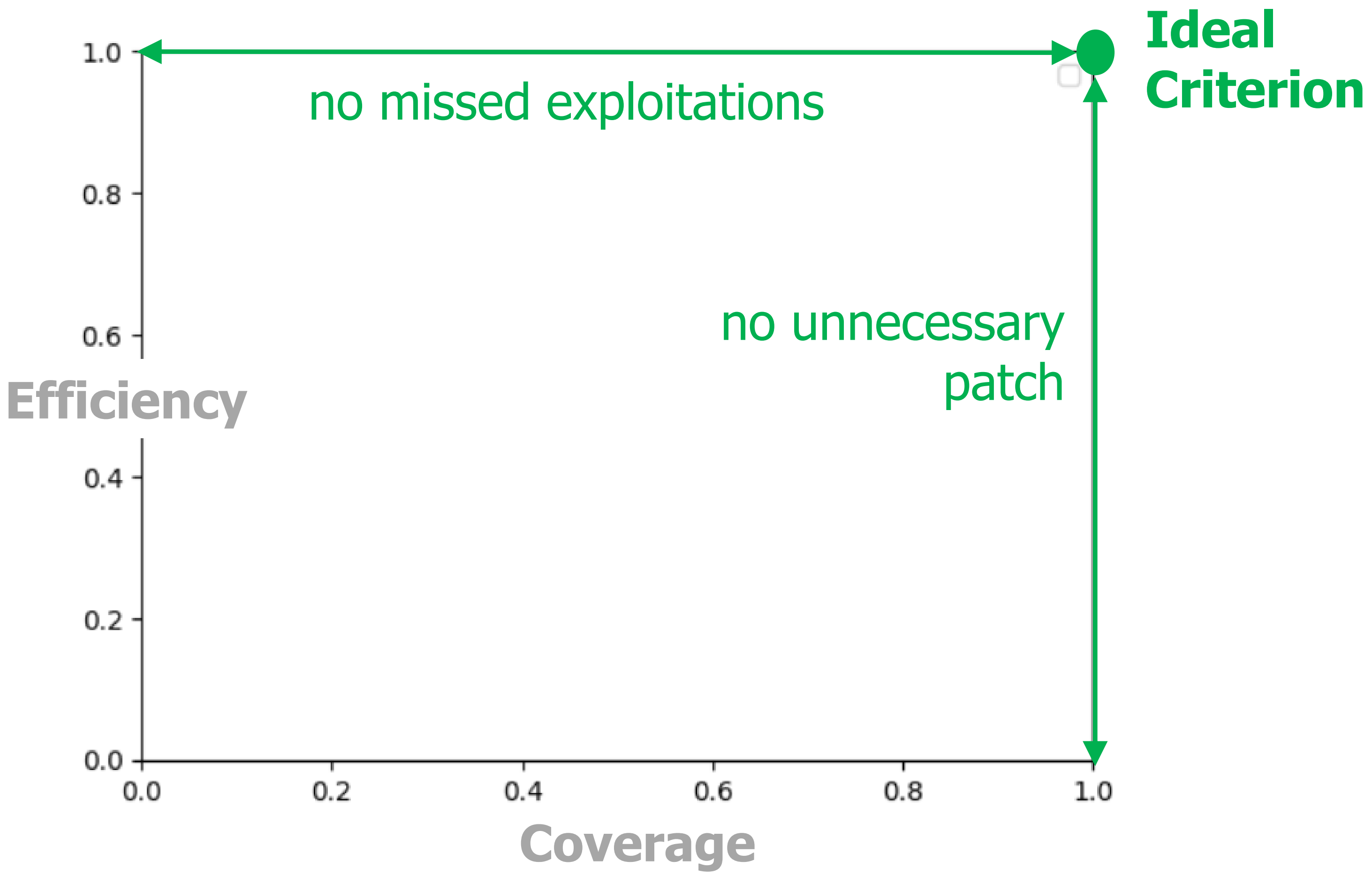
Efficiency


1.0
0.8
0.6
Efficiency
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Coverage
patches not useful
patches really useful

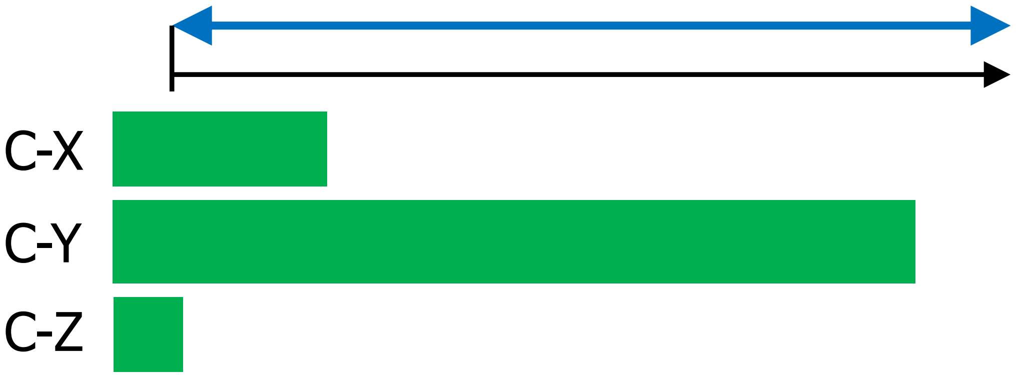
Coverage


Ideal
Criterion
1.0
0.8
0.6
Efficiency
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
neutralized
exploitations
missed
exploitations
Coverage

Ideal Criterion



Patching Effort (I)

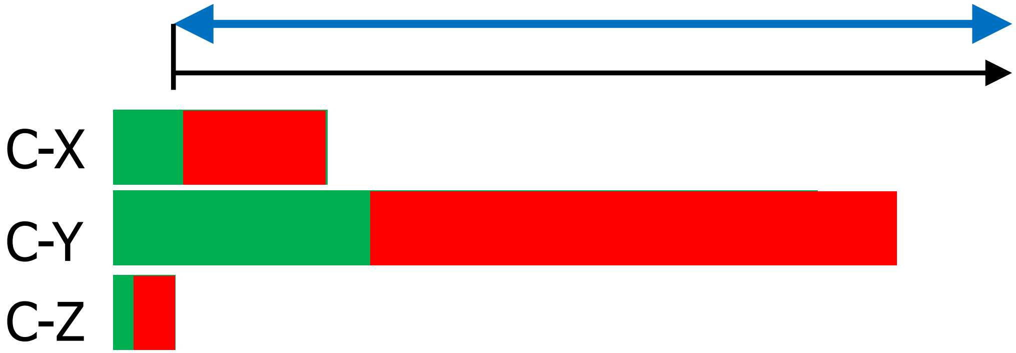


- #(PatchedSet) ≡ **Patching Effort**
- It depends on the chosen criterion:
  - All vulns with CVSS Critical
  - All vulns with remote injection
  - All vulns of Windows software
  - ...

Patching Effort (II)

- Coverage and Efficiency are **relative** indexes
- **Independent** of Patching Effort

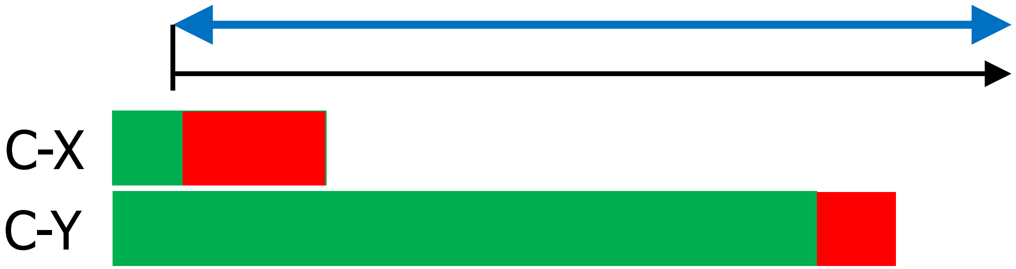
- You could have many **different** criteria with:
  - **Identical** Coverage / Efficiency
  - Widely different PatchingEffort



Patching Effort (II)

- Coverage and Efficiency do **not** tell the whole story
- Assessment of any given criterion requires analyzing all **the 3 indexes**

- Example:
  - Criterion C-Y has excellent Coverage / Efficiency
  - …but our org **cannot afford** its PatchingEffort



Exploit Prediction: Problem Definition Summary

- ❑ Criterion for **choosing which vulnerabilities to patch**

- ❑ Assessment indexes:
  - ❑ How good in defense (Coverage)
  - ❑ How efficient (Efficiency)
  - ❑ How costly (Patching effort)

Exploit Prediction: Problem Definition **Remarks (I)**

- **MANY** factors **NOT** assessed

- Given a certain `vuln`:
  - How **many systems** do I have with `vuln`?
  - How **costly** is an **incident** based on `vuln`?
  - ...

Exploit Prediction: Problem Definition **Remarks (II)**

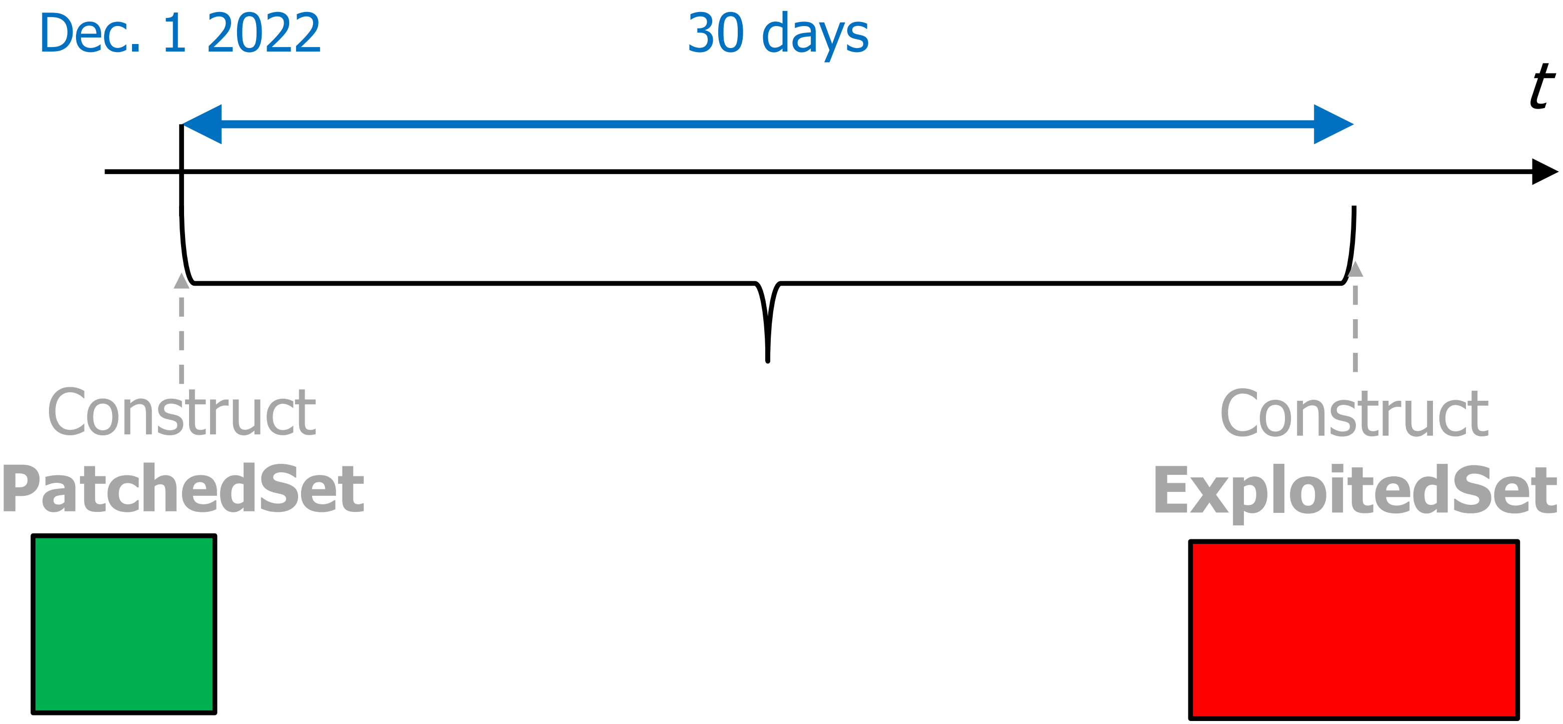
- **MANY** factors **NOT** assessed

- We observe which vulnerabilities have been **actually exploited worldwide** in T
- **Approximation** by collecting many intelligence feeds

- Given a certain `vuln`:
  - …
  - How **likely** is that **I** will be attacked with `vuln`?
  - …

Exploit Prediction: Example Criteria

Experiment Scenario



Various Criteria

≈8000 vulns observed

PatchedSet: CVSS (I)

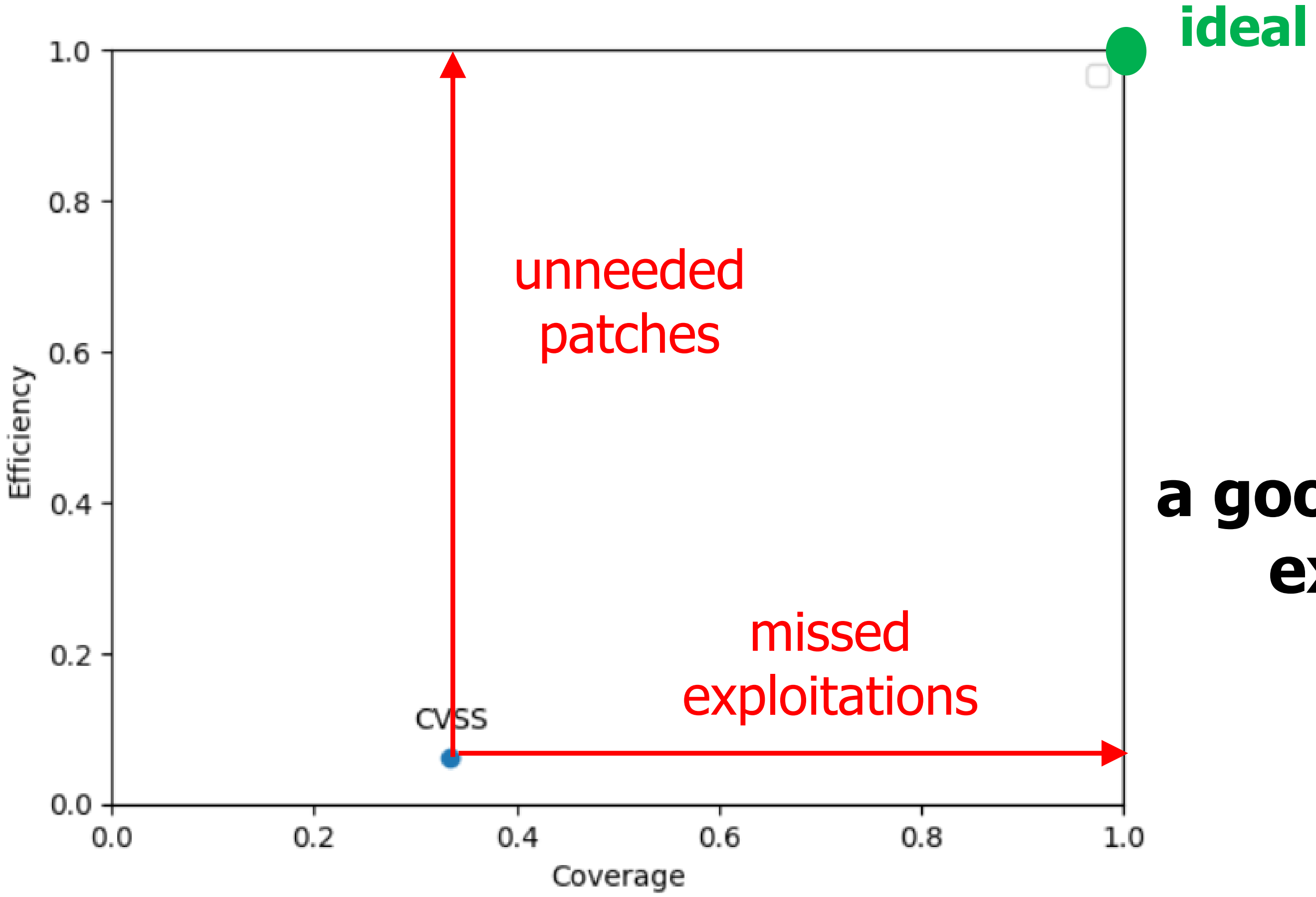
Dec. 1 2022

30 days

PatchedSet

- All the CVEs with **CVSS >= 9.1** (≈15% of all vulns)
- Patching Effort: ≈28000 vulns

PatchedSet CVSS (II)



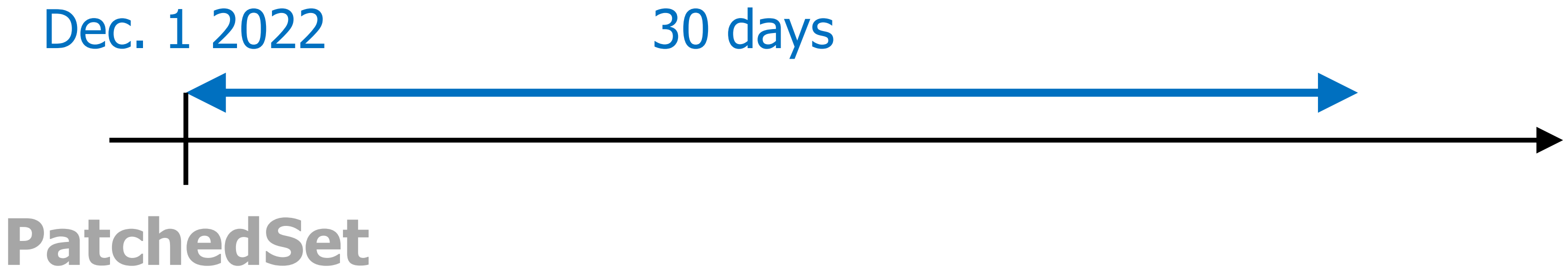
CVSS is not a good predictor of exploitation

CISA - KEV

CYBERSECURITY
& INFRASTRUCTURE
SECURITY AGENCY

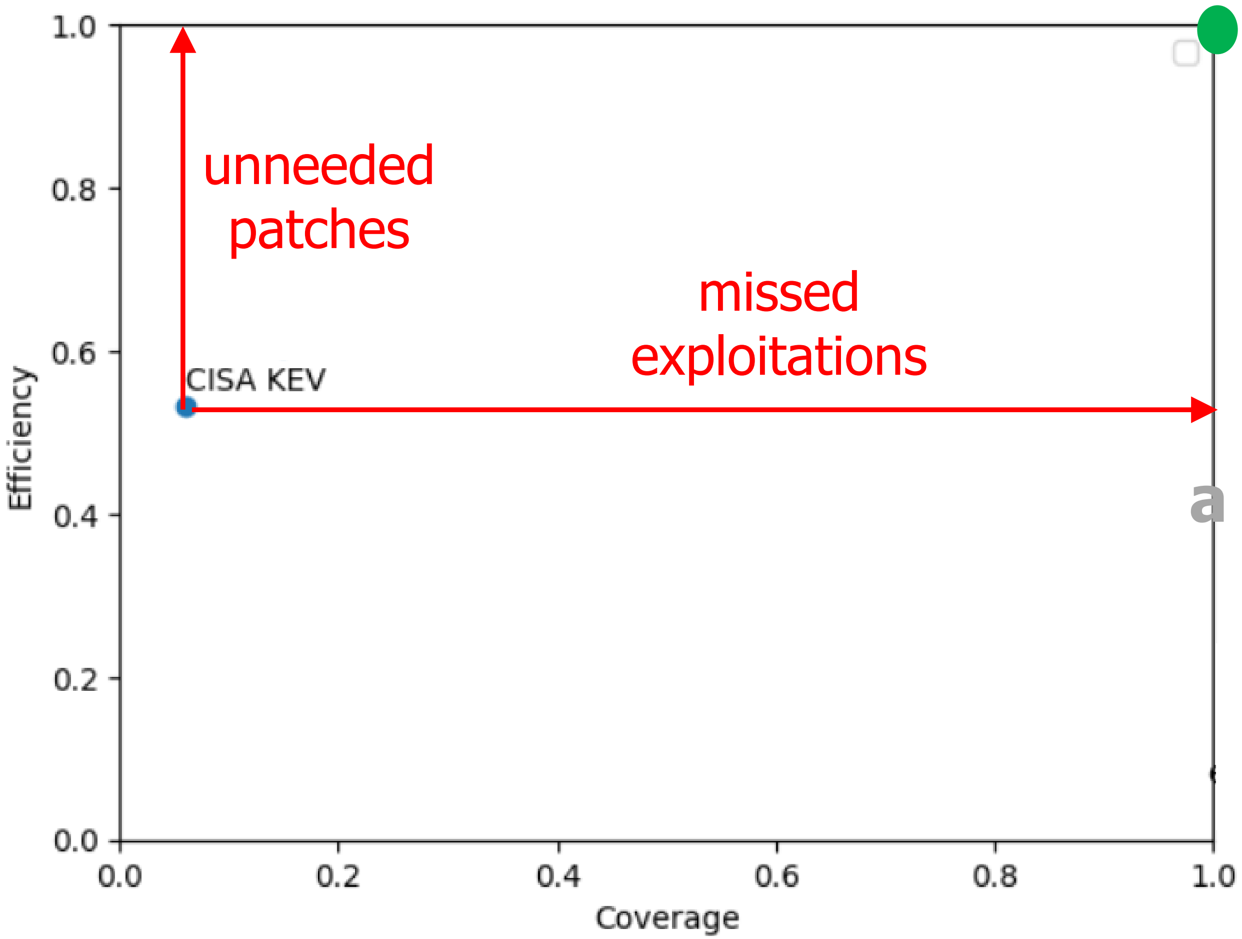
KNOWN EXPLOITED VULNERABILITIES CATALOG

PatchedSet: CISA-KEV (I)



- CVEs **in CISA-KEV**
  (≈0.5% of all vulns)
- Patching Effort: ≈900 vulns

PatchedSet: CISA-KEV (II)



ideal

unneeded patches

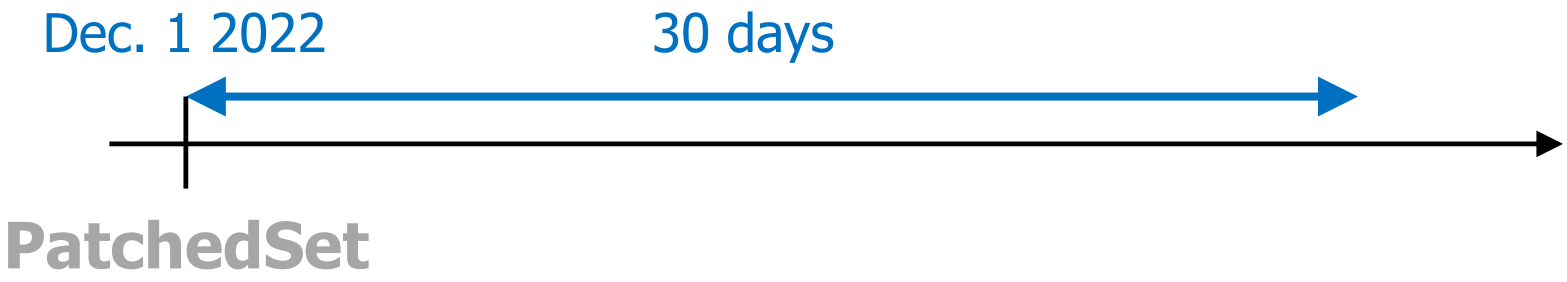
missed exploitations

CISA-KEV

is not

a good predictor of exploitation

Selection based on Other Heuristics (I-a)



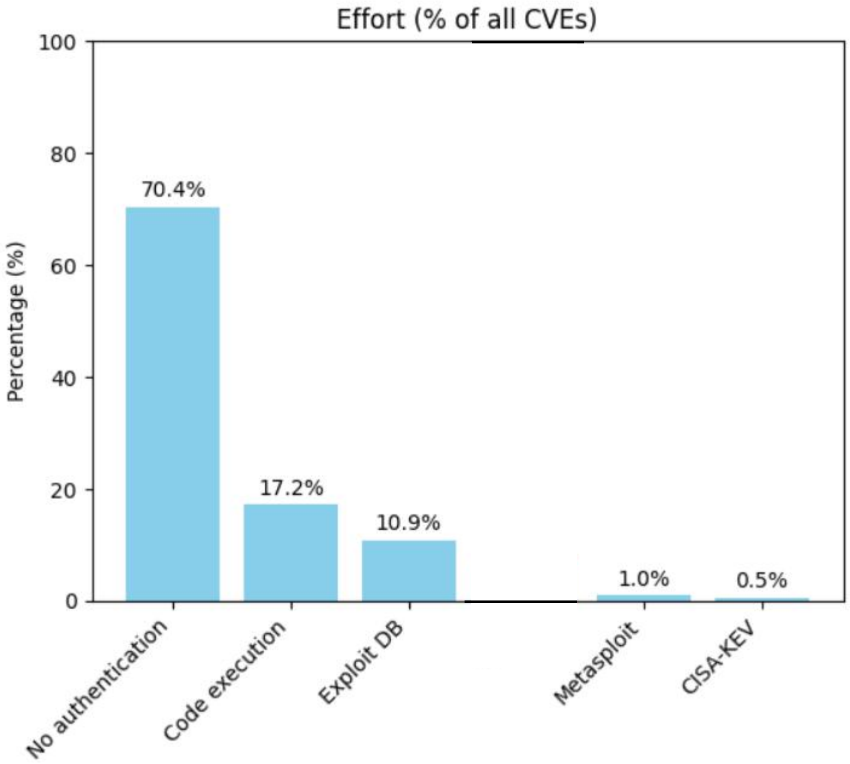
Dec. 1 2022

30 days

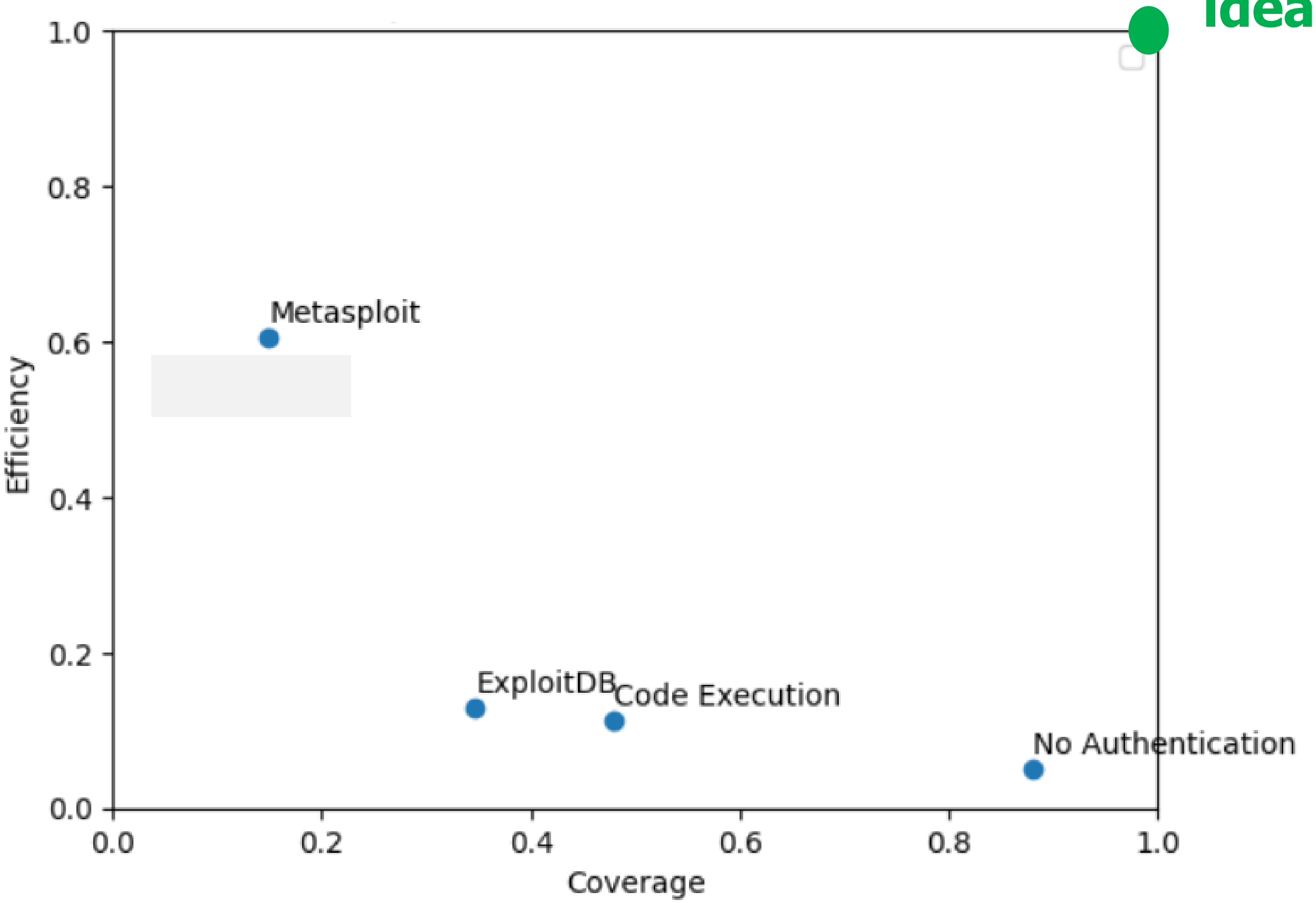
PatchedSet

- Exploit in Metasploit
- Exploit in ExploitDB
- Impact is Code Execution
- Injection does not require Authentication

Selection based on Other Heuristics (I-b)



Selection based on Other Heuristics (II)



ideal

Exploit Prediction Scoring System (EPSS)

Exploit Probability: Definition (REMIND)

- □ Vulnerability `CVE-i`
- □ `P(CVE-i, d):`

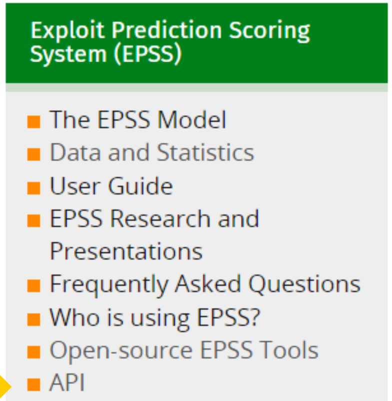
$$\frac{\#\text{ Exploitation }\textbf{attempts}\text{ of CVE-i in [d,d+30]}}{\#\text{ Expolitation }\textbf{attempts}\text{ of }\textbf{all}\text{ CVEs in [d,d+30]}}$$

- □ Probability that `CVE-i` will be exploited in the next 30 days

- □ **Worldwide** (everywhere)
- □ **Approximated** by collecting many TI feeds
- □ Computed **a posteriori**

EPSS Definition

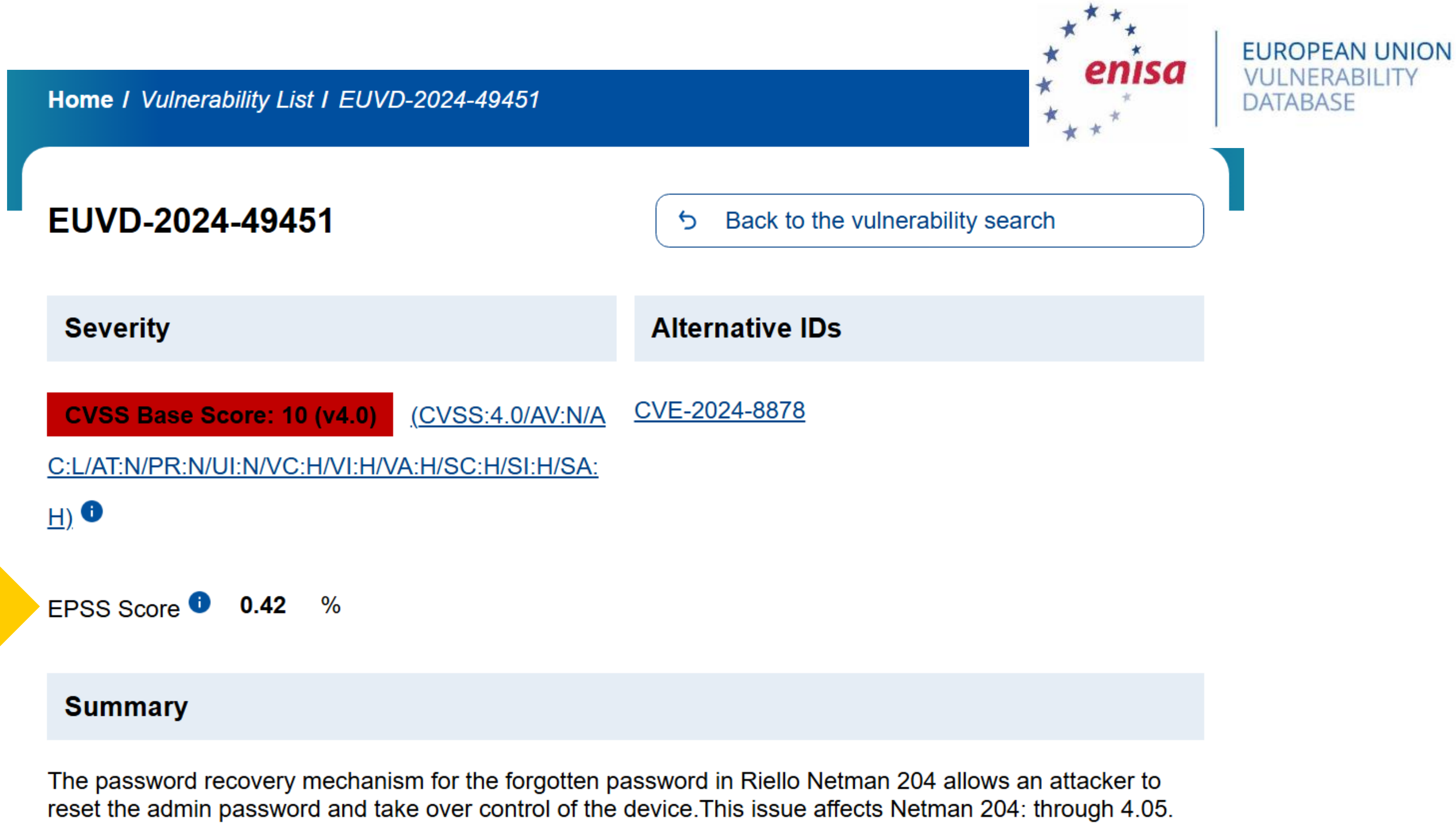
- `EPSS(CVE-i, d)`: **Estimate** of `P(CVE-i, d)`
- Computed and published **daily** by the FIRST Consortium



Exploit Prediction Scoring System (EPSS)

- The EPSS Model
- Data and Statistics
- User Guide
- EPSS Research and Presentations
- Frequently Asked Questions
- Who is using EPSS?
- Open-source EPSS Tools
- API

Enisa VD (I)



Home / *Vulnerability List* / *EUVD-2024-49451*

EUVD-2024-49451

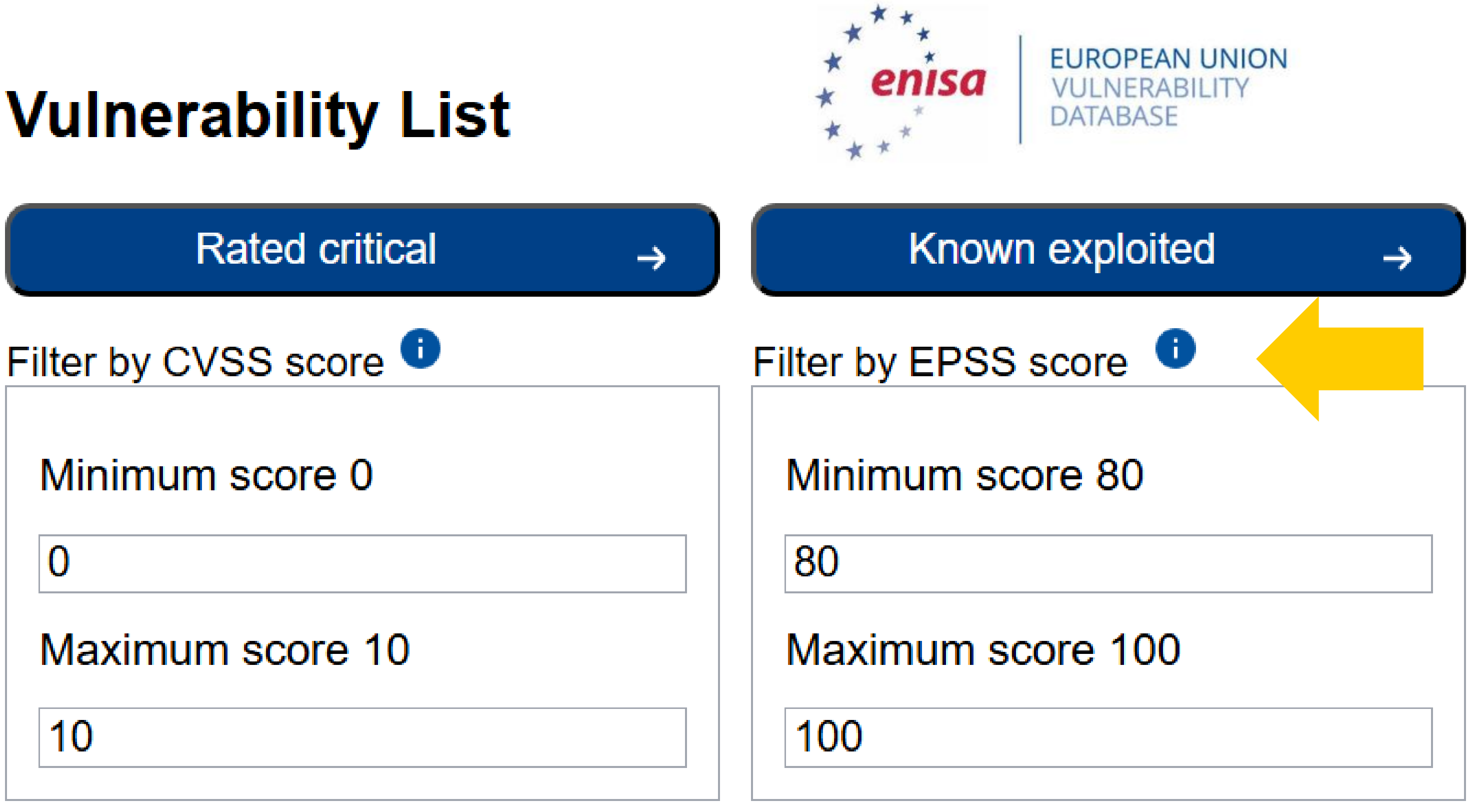
Severity	**Alternative IDs**
CVSS Base Score: 10 (v4.0) (CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H)	CVE-2024-8878

EPSS Score **0.42** %

Summary

The password recovery mechanism for the forgotten password in Riello Netman 204 allows an attacker to reset the admin password and take over control of the device.This issue affects Netman 204: through 4.05.

Enisa VD (II)



Vulnerability List

Filter by CVSS score

Minimum score 0

0

Maximum score 10

10

Filter by EPSS score

Minimum score 80

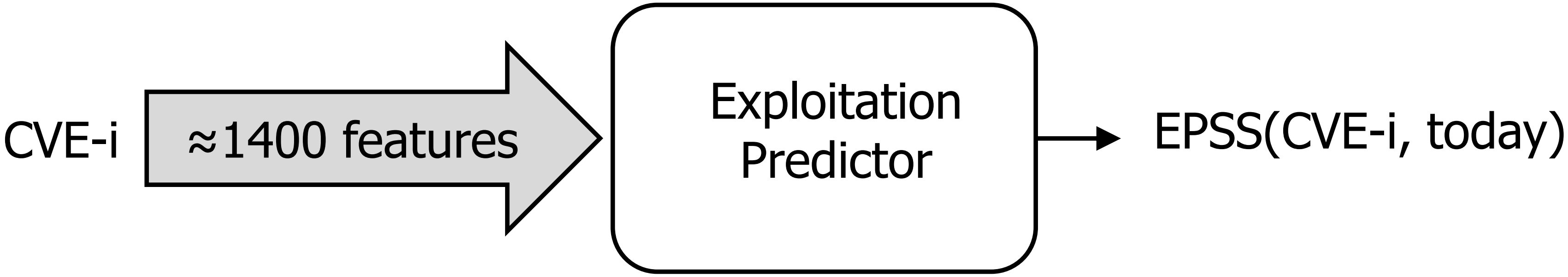
80

Maximum score 100

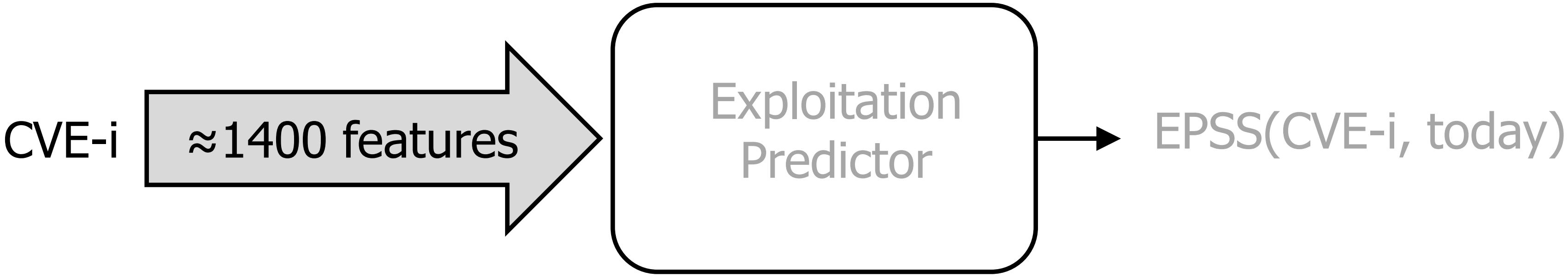
100

EPSS Computation

- Repeat **every day**:
  - For each CVE-i:
    - **Compute features** of CVE-i
    - **Estimate** its probability of exploitation in the next 30 days



How is each CVE represented? (I)



- Array with 1400 elements
  - Numerical features
  - Categorical features (one-hot representation)

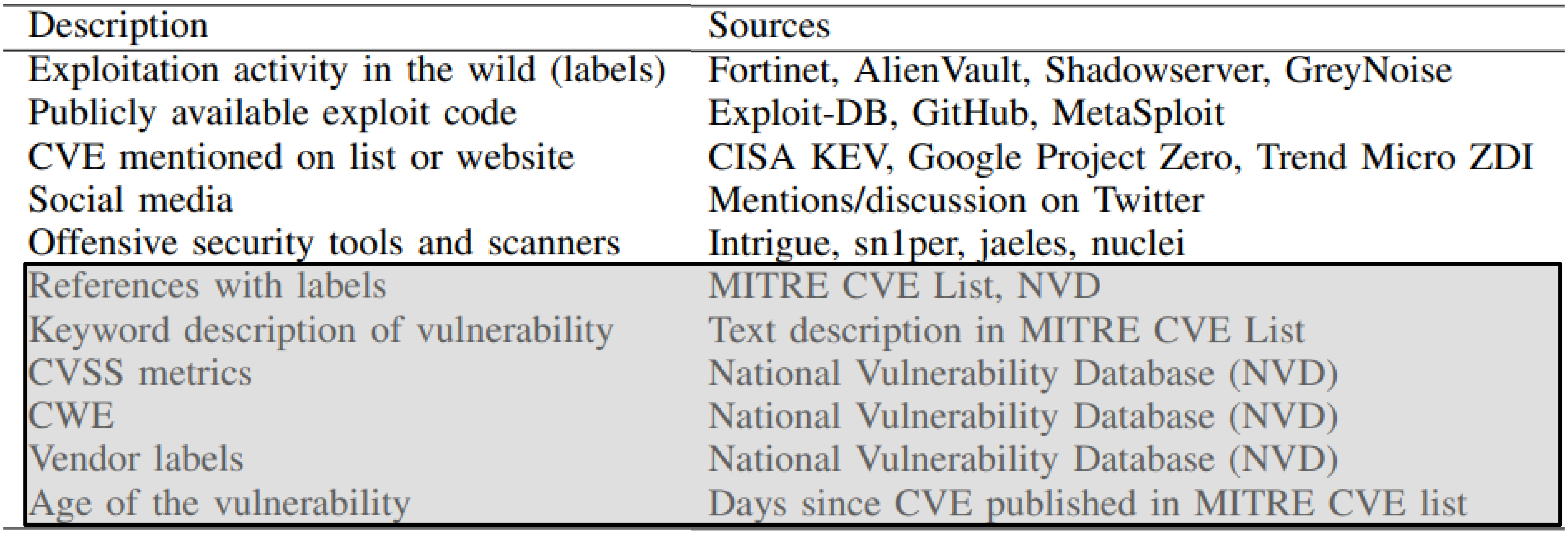
- Details out of scope
- **Information sources** in scope

How is each CVE represented? (II)

Description	Sources
[redacted]	[redacted]
Keyword description of vulnerability	Text description in MITRE CVE List
CVSS metrics	National Vulnerability Database (NVD)
CWE	National Vulnerability Database (NVD)
Vendor labels	National Vulnerability Database (NVD)
Age of the vulnerability	Days since CVE published in MITRE CVE list

Intrinsic properties of CVE-i
(more or less)

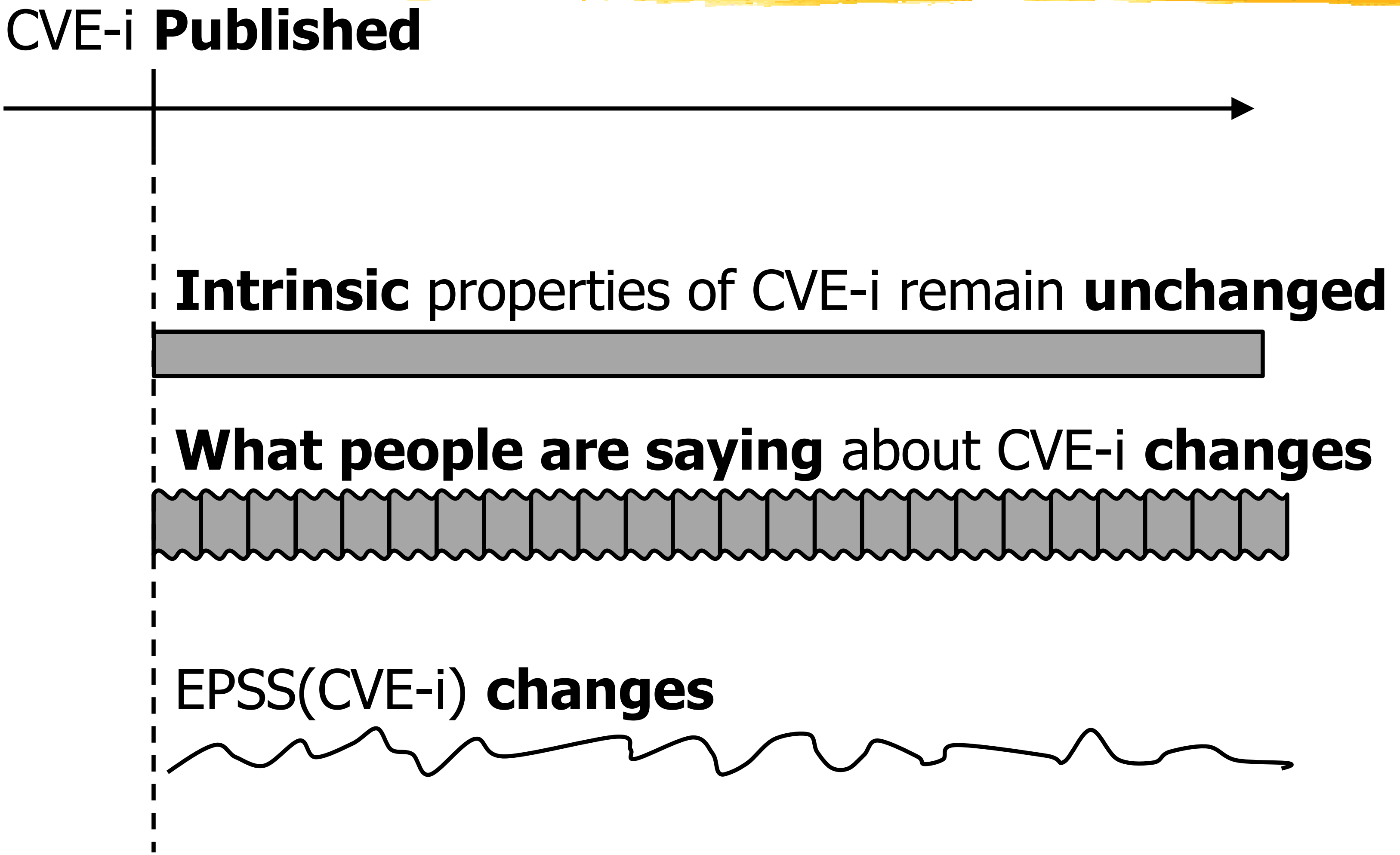
How is each CVE represented? (III)



Description	Sources
Exploitation activity in the wild (labels)	Fortinet, AlienVault, Shadowserver, GreyNoise
Publicly available exploit code	Exploit-DB, GitHub, MetaSploit
CVE mentioned on list or website	CISA KEV, Google Project Zero, Trend Micro ZDI
Social media	Mentions/discussion on Twitter
Offensive security tools and scanners	Intrigue, sn1per, jaeles, nuclei
References with labels	MITRE CVE List, NVD
Keyword description of vulnerability	Text description in MITRE CVE List
CVSS metrics	National Vulnerability Database (NVD)
CWE	National Vulnerability Database (NVD)
Vendor labels	National Vulnerability Database (NVD)
Age of the vulnerability	Days since CVE published in MITRE CVE list

- Summary of "**what people are saying** of CVE-i"
- Updated **daily**

CVE-I features → EPSS



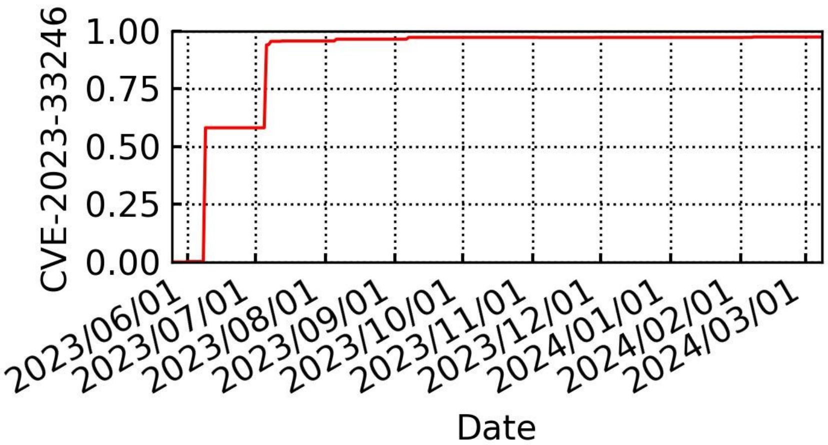
CVE-i **Published**

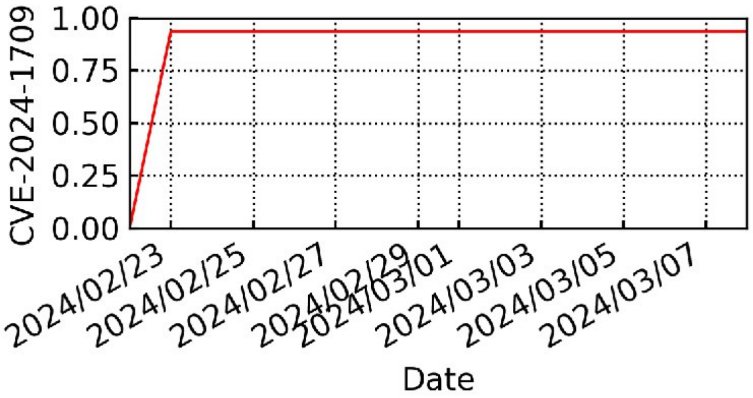
Intrinsic properties of CVE-i remain **unchanged**

What people are saying about CVE-i **changes**

EPSS(CVE-i) **changes**

EPSS evolution: Examples (I)

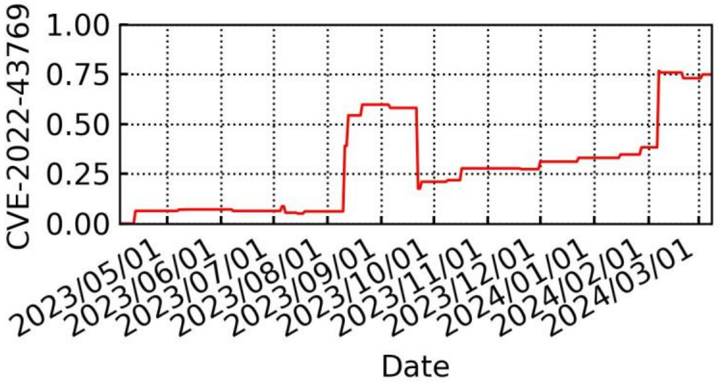


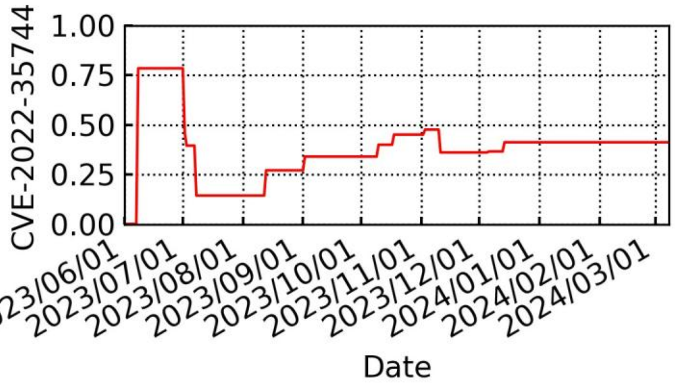


- Significant growth **after 1 day**
- ...and then again on the next day
- Heavily exploited for more than 6 months

- **Immediately** exploited heavily, for several weeks

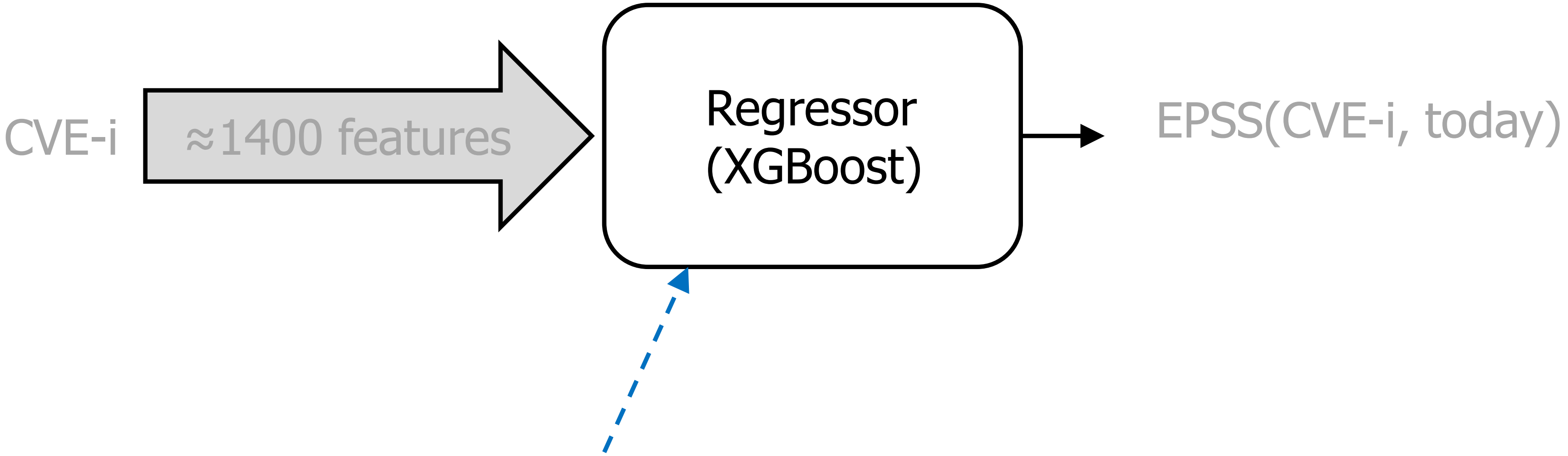
EPSS evolution: Examples (II)





- Temporal evolution may often be:
  - Very "irregular"
  - Very hard to predict (even in the short term)

How is EPSS computed?



CVE-i

≈1400 features

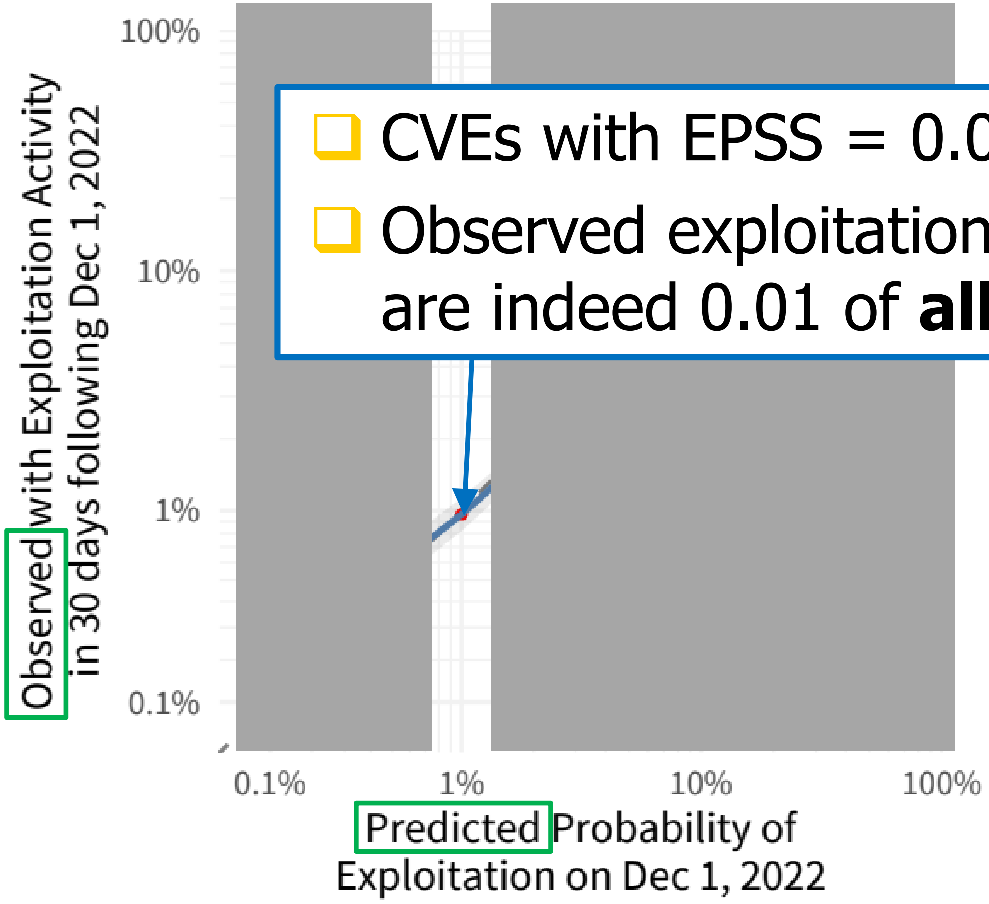
Regressor (XGBoost)

EPSS(CVE-i, today)

- Data driven model
- Trained on 1 year of data

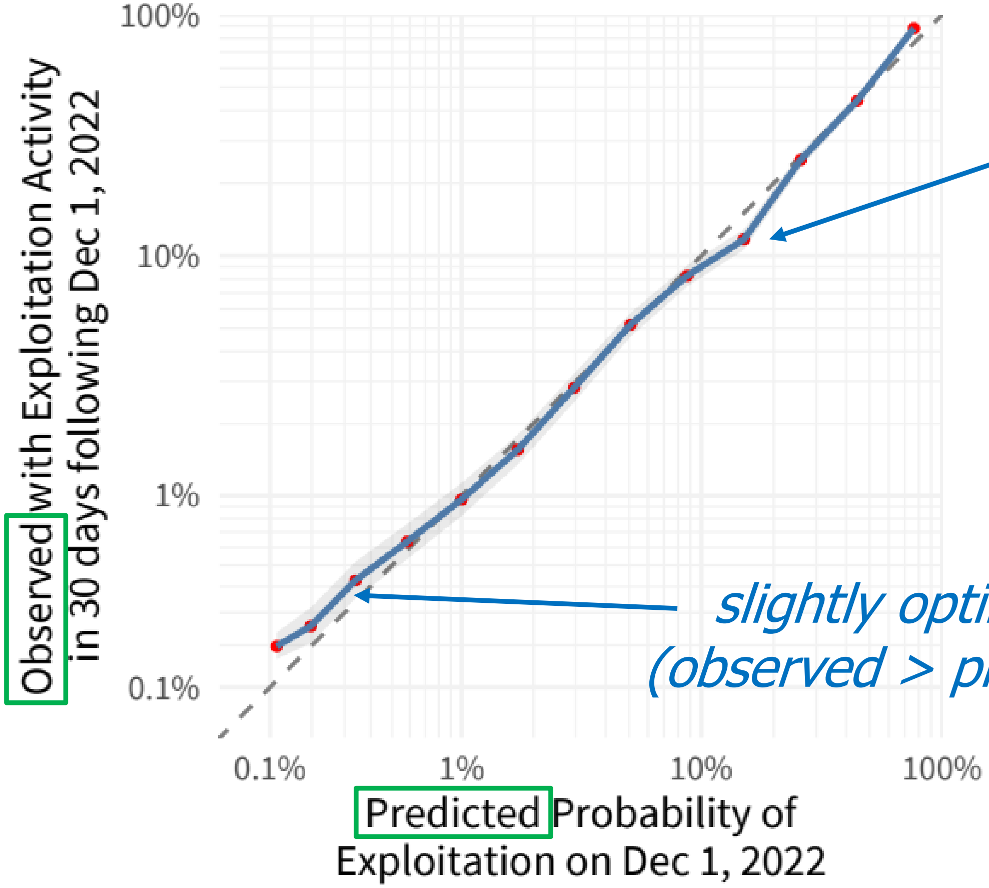
- March 2023: 3rd model refinement
- March 2025: 4th release

P_observed(CVE-i) vs P_predicted(CVE-i)



- CVEs with EPSS = 0.01
- Observed exploitation attempts of these CVEs are indeed 0.01 of **all** the observed attempts

P_observed(CVE-i) ≈ P_predicted(CVE-i)

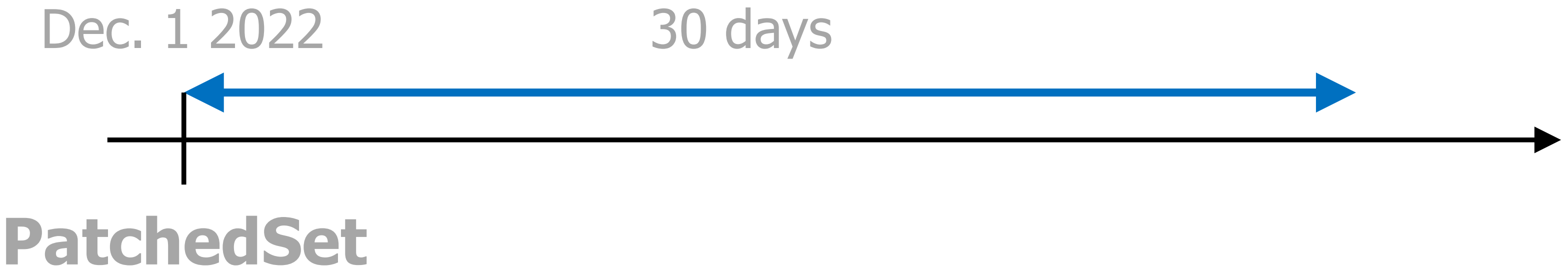


slightly pessimistic

slightly optimistic (observed > predicted)

EPSS for Exploit Prediction: Coverage and Efficiency?

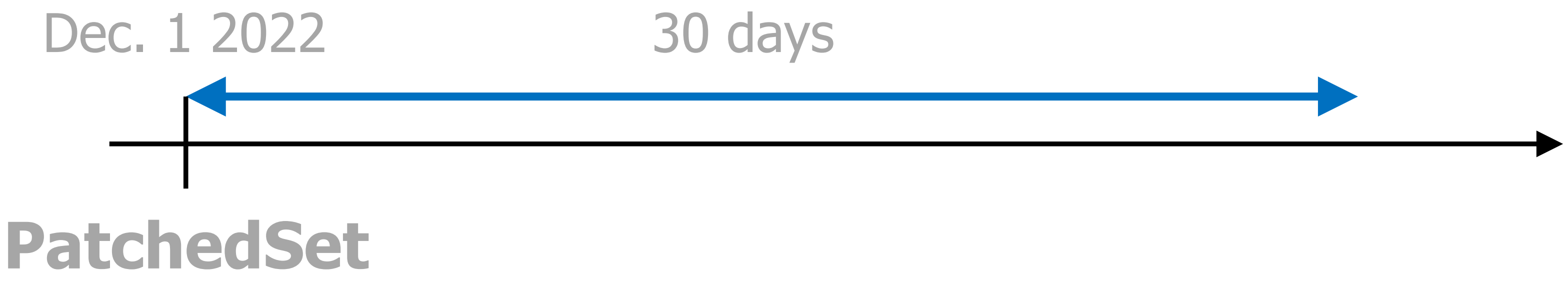
EPSS vs CVSS



- ❑ CVEs with "large" **CVSS**
- ❑ CVEs with "large" **EPSS**

- ❑ Which criterion is better?
- ❑ Is it better to prioritize patches based on CVSS…or on EPSS?

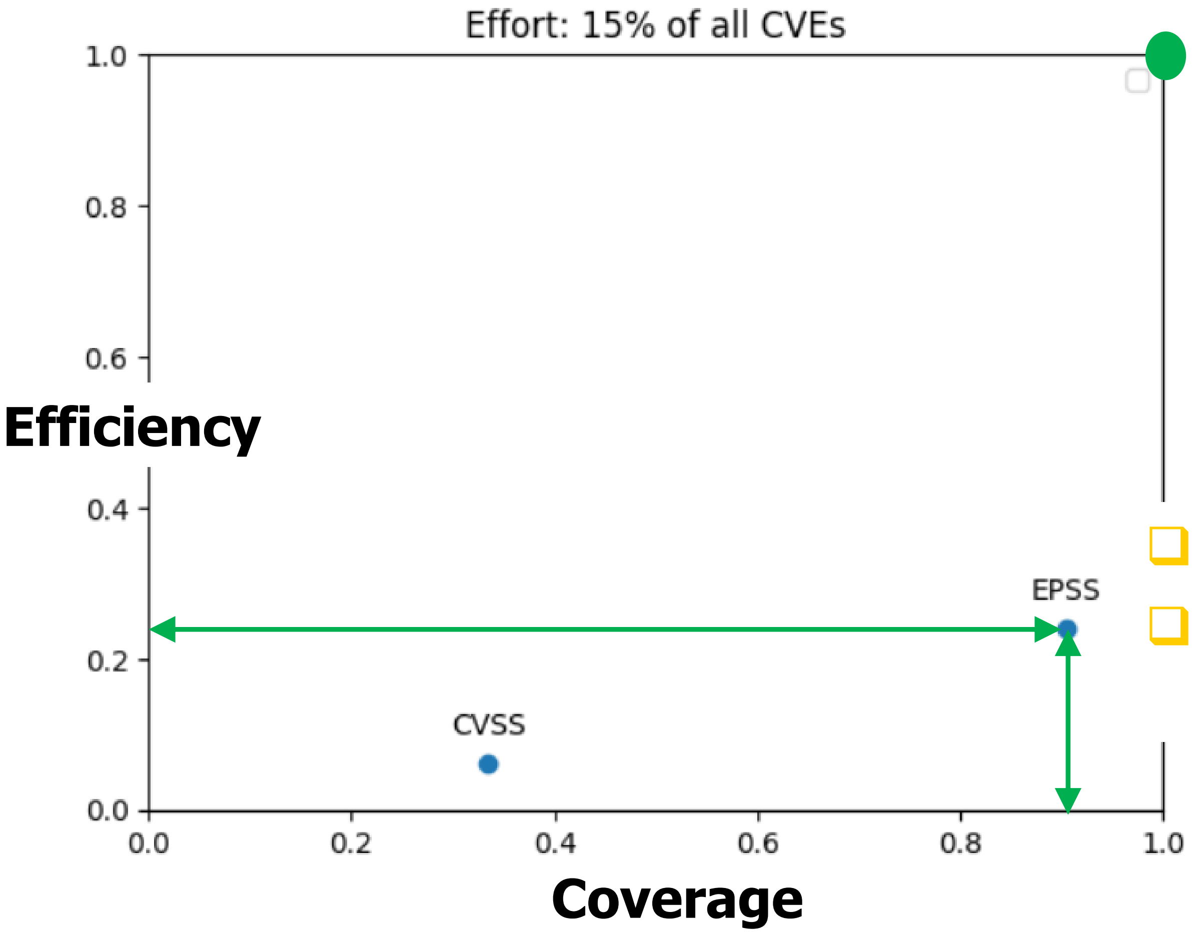
EPSS vs CVSS: Identical Patching Effort (I)



- Set1: CVSS(CVE-i) >= 9.1 **(15% of all CVEs)**
- Set2: EPSS(CVE-i) >= **0.022** **(15% of all CVEs)**

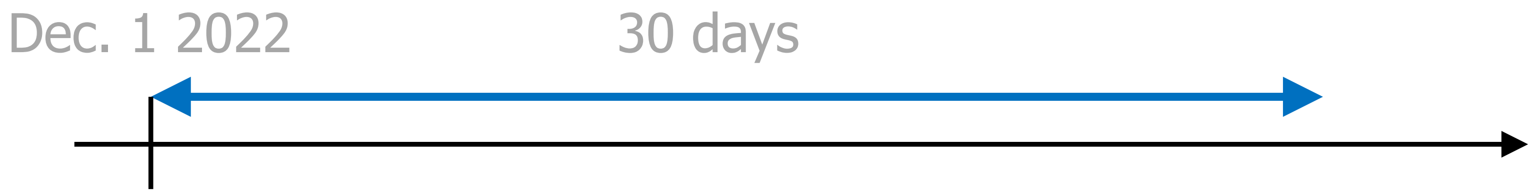
- **Identical Patching Effort**
- Efficiency?
- Coverage?

EPSS vs CVSS: Identical Patching Effort (II)



- ❑ Excellent coverage
- ❑ More efficient than CVSS (But not very efficient)

EPSS vs CVSS: Identical Coverage (I)



Dec. 1 2022

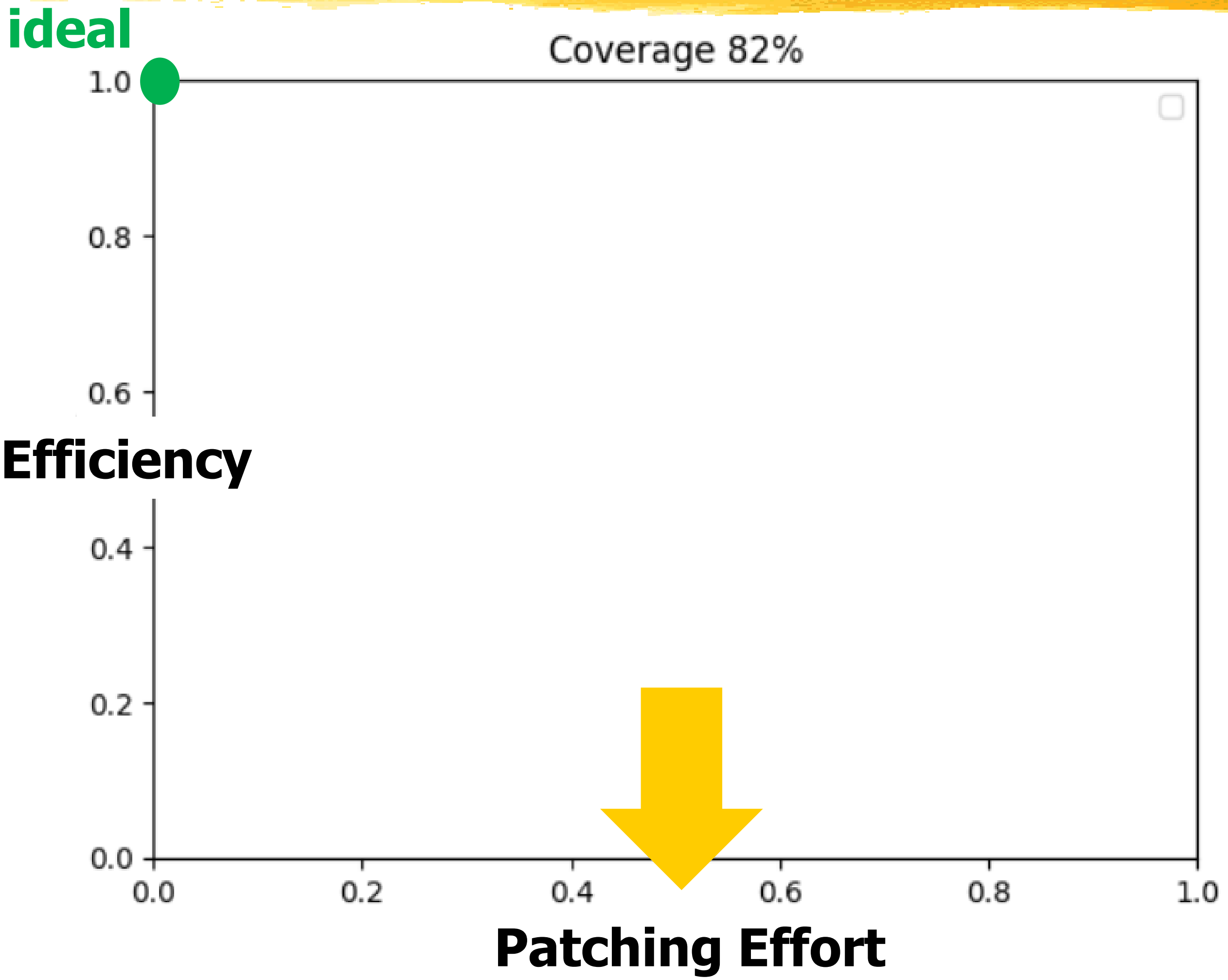
30 days

PatchedSet

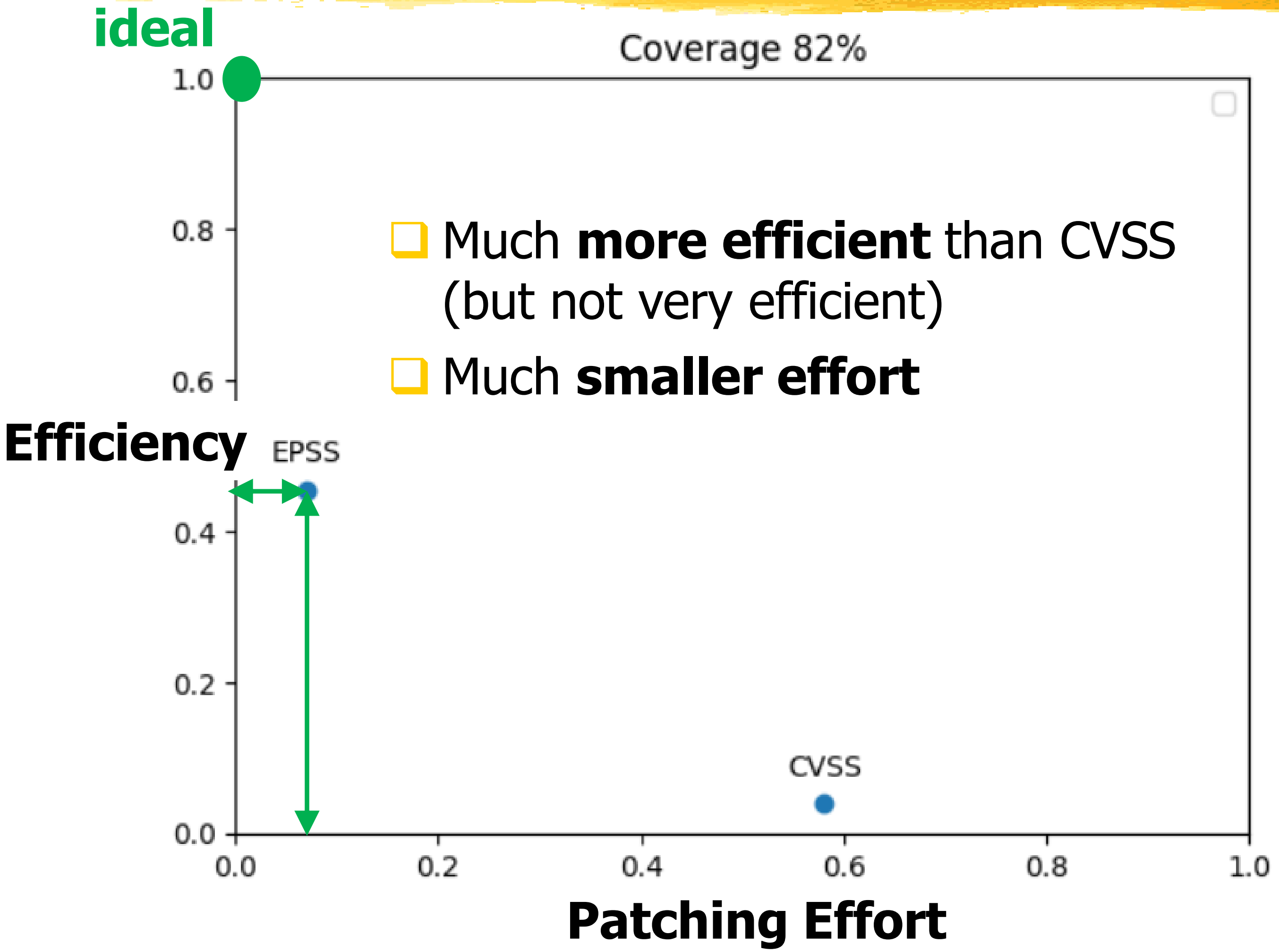
- Set1: CVSS(CVE-i) >= 7 (**Coverage 82%**)
- Set2: EPSS(CVE-i) >= **0.088** (**Coverage 82%**)

- **Identical Coverage**
- Efficiency?
- Patching Effort?

EPSS vs CVSS: Identical Coverage (II-a)

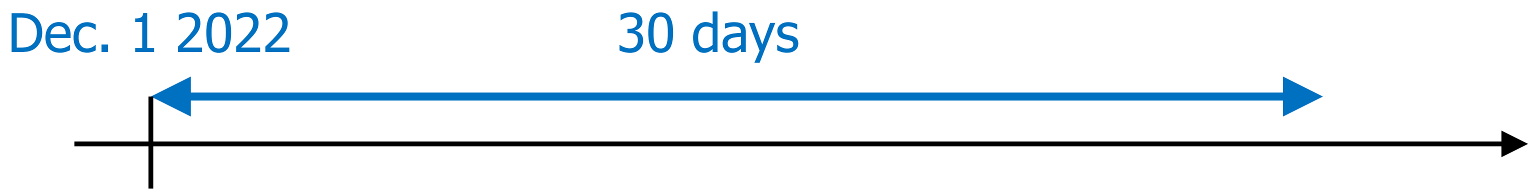


EPSS vs CVSS: Identical Coverage (II-b)



- Much **more efficient** than CVSS (but not very efficient)
- Much **smaller effort**

EPSS vs Other Heuristics (I)



Dec. 1 2022

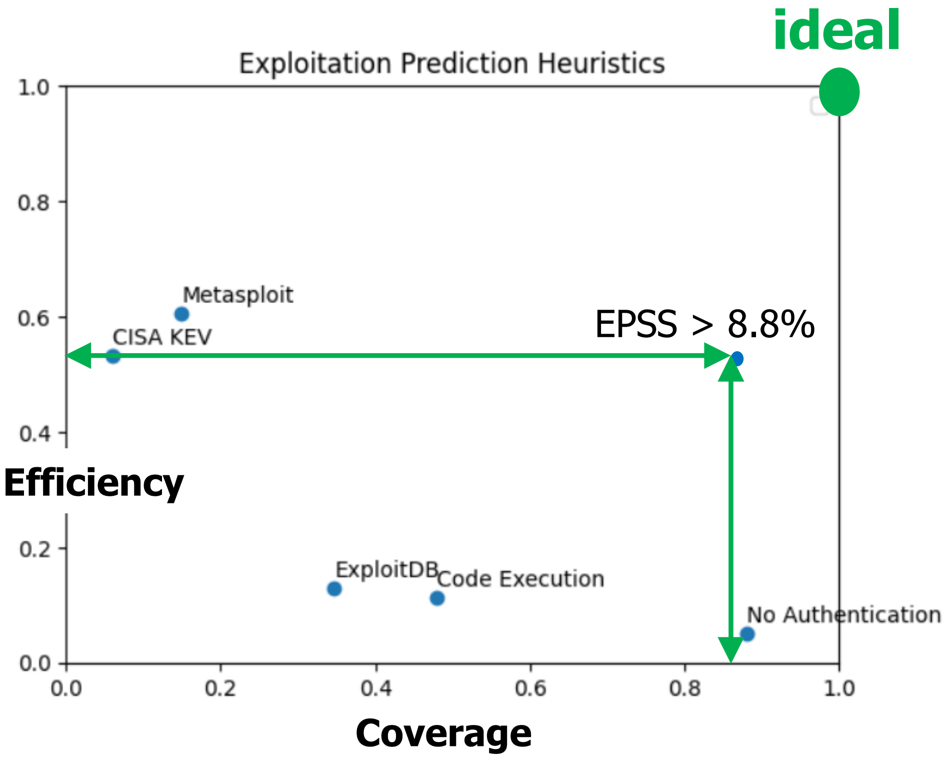
30 days

PatchedSet

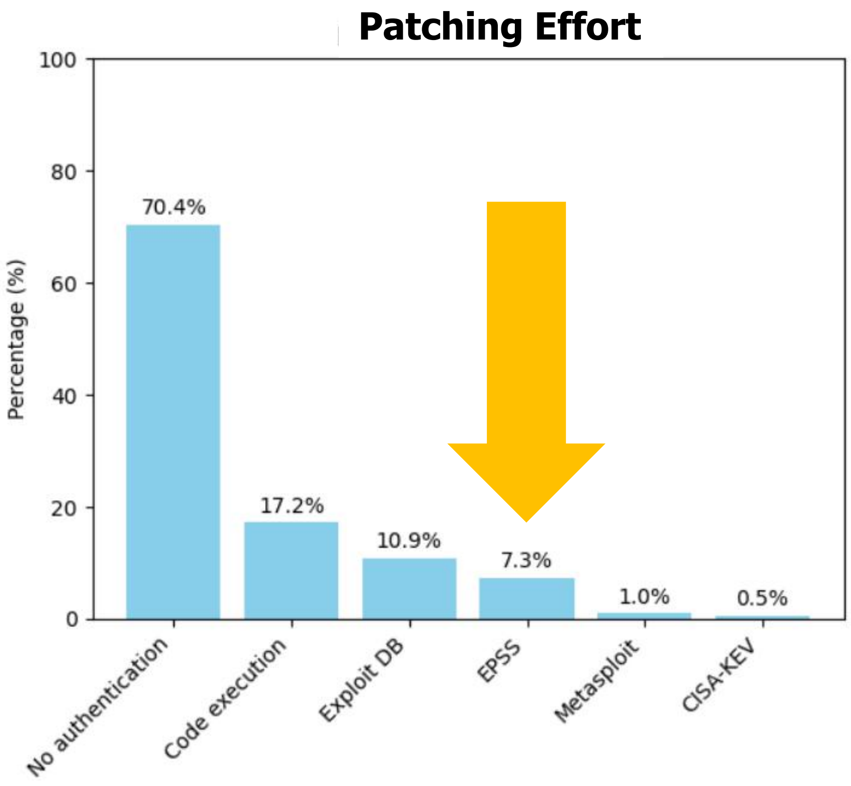
- □ Exploit in Metasploit
- □ Exploit in ExploitDB
- □ Impact is Code Execution
- □ Injection does not require Authentication

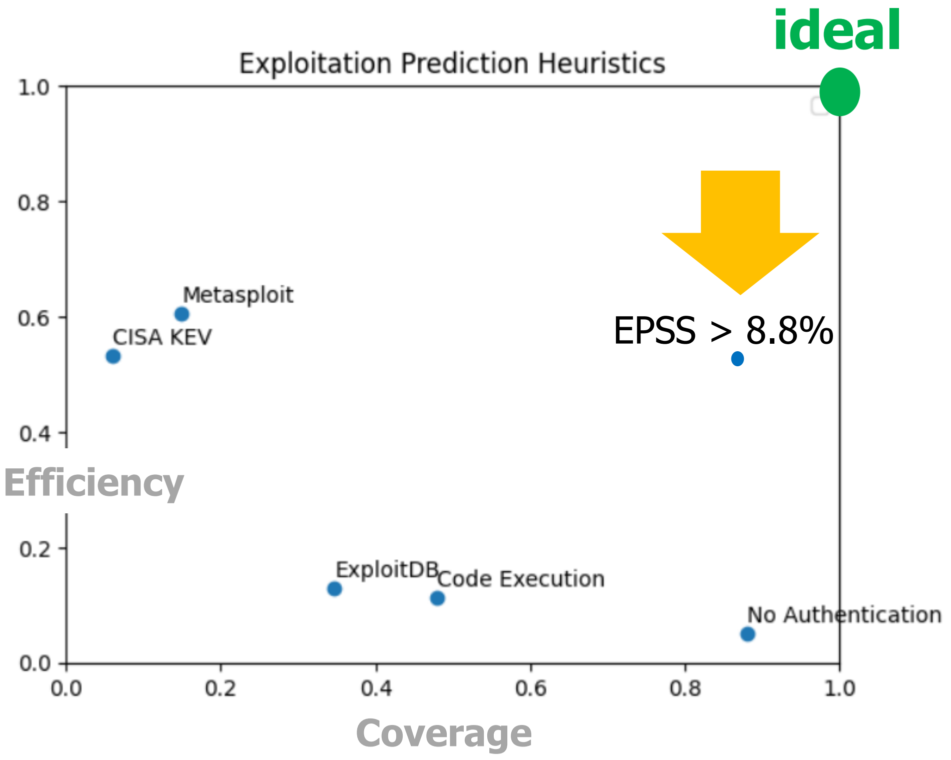
- □ EPSS > 8.8%

EPSS vs Other Heuristics (II-a)



EPSS vs Other Heuristics (II-a)

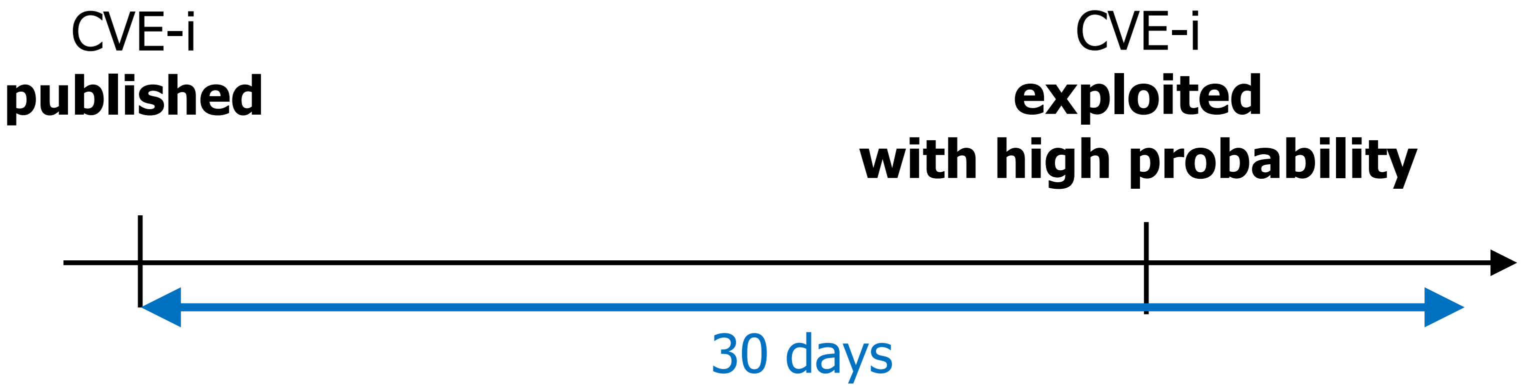




Understanding EPSS for Exploit Prediction

- Based on some of our (yet unpublished) analyses

Using EPSS in practice



CVE-i **published**

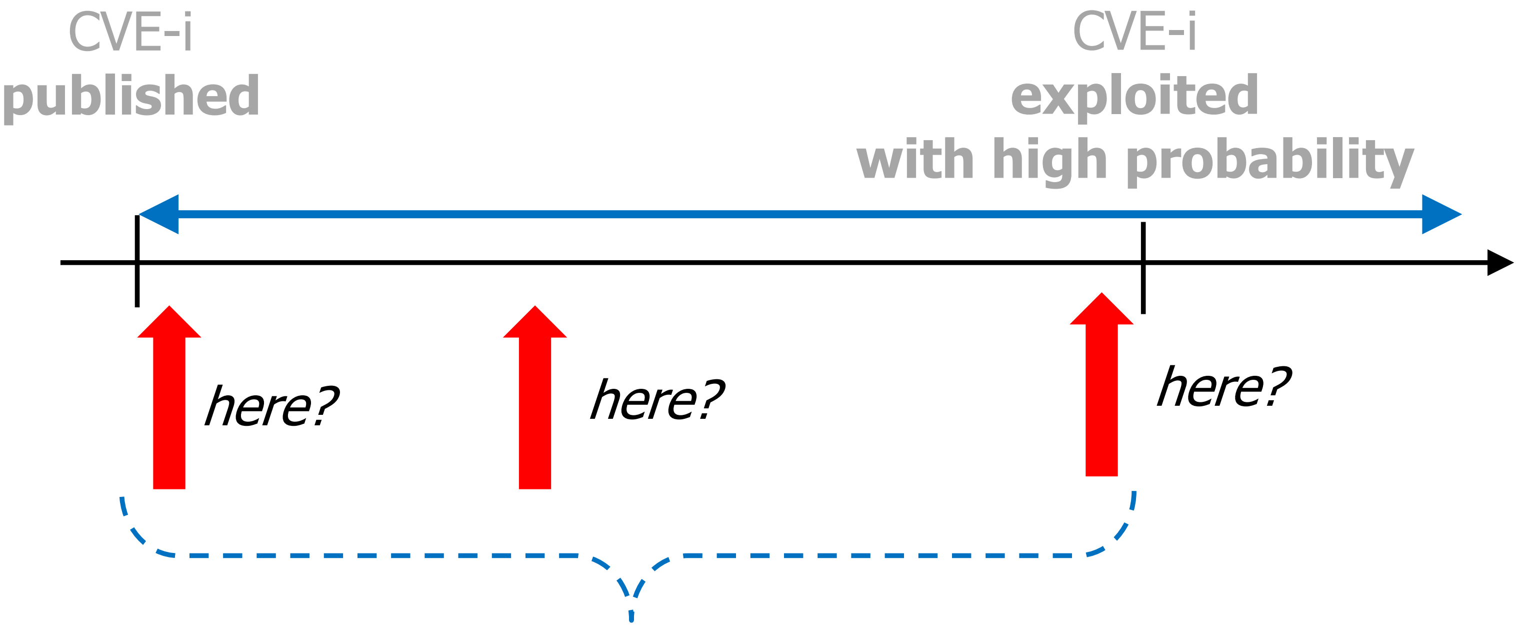
CVE-i **exploited with high probability**

30 days

- Prioritizing CVE-i with "large" EPSS is a good criterion

- But what about **newly published** CVE-i?
- **When** we will prioritize them?

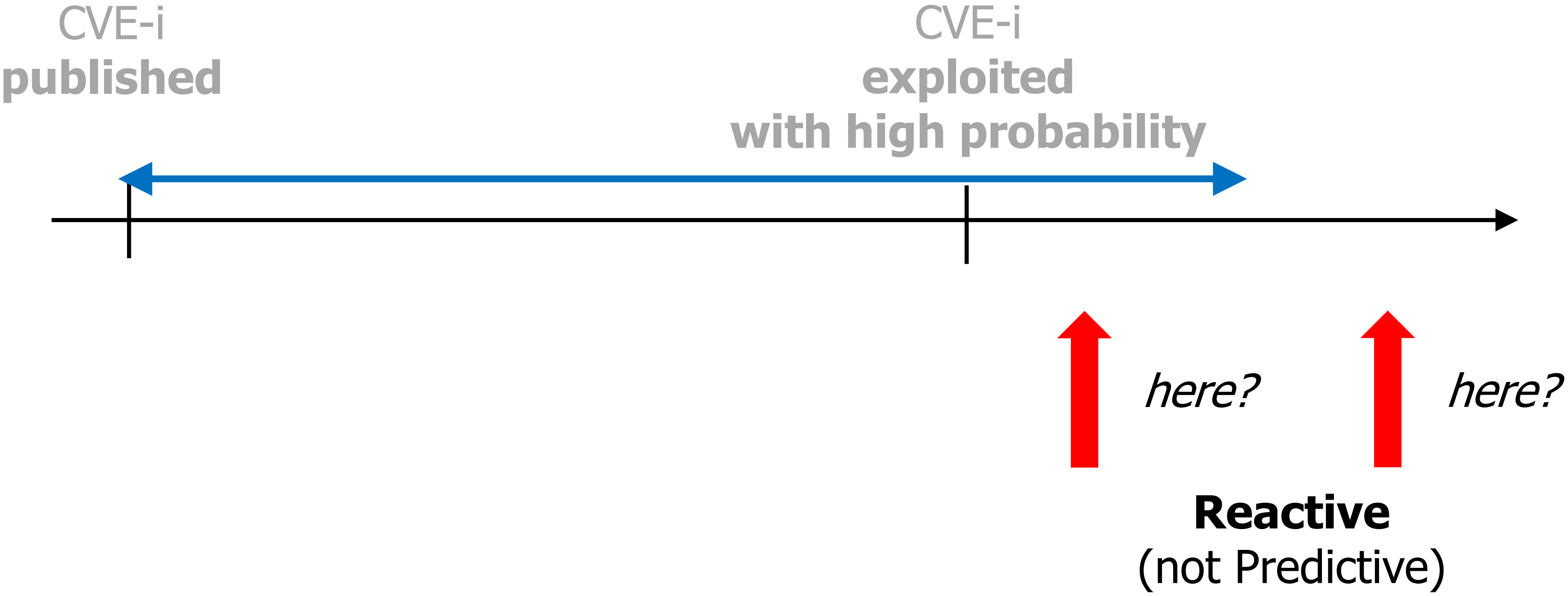
Does EPSS predict FUTURE exploitation?



Day on which EPSS(CVE-i)
becomes "large enough"

...so that CVE-i is prioritized

…or does it summarize what is ALREADY happening?



CVE-i
published

CVE-i
exploited
with high probability

here?

here?

Reactive
(not Predictive)

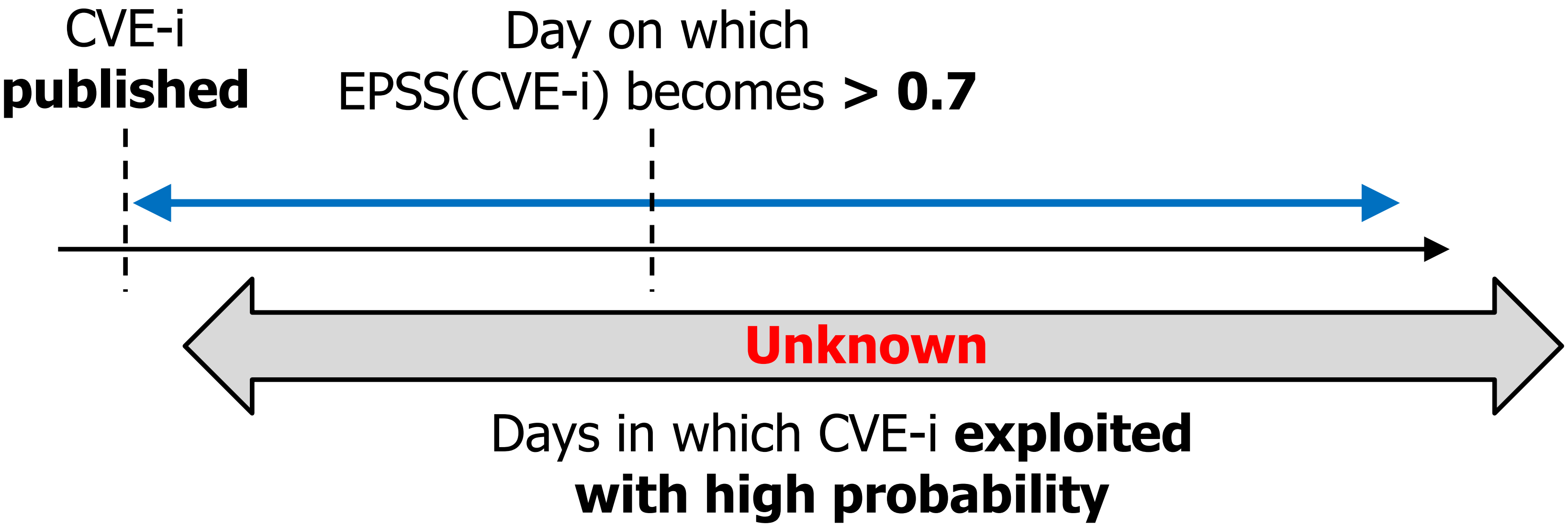
EPSS: Dynamic behavior (I)

- Prioritized-CVE :=
  - All CVE-i **published** in [March 2023, September 2024] → 45080
  - Such that **EPSS(CVE-i) > 0.7** (at some day) → 137

- Does EPSS
  - "**predict** actual exploitation by **many** days"?
  - "**predict** actual exploitation by **a few** days"?
  - "**summarizes after** the fact what is **already** happening"?

EPSS: Dynamic behavior (II)

- Does EPSS
  - "**predict** actual exploitation by **many** days"?
  - "**predict** actual exploitation by **a few** days"?
  - "**summarizes after** the fact what is **already** happening"?
- We **cannot** answer

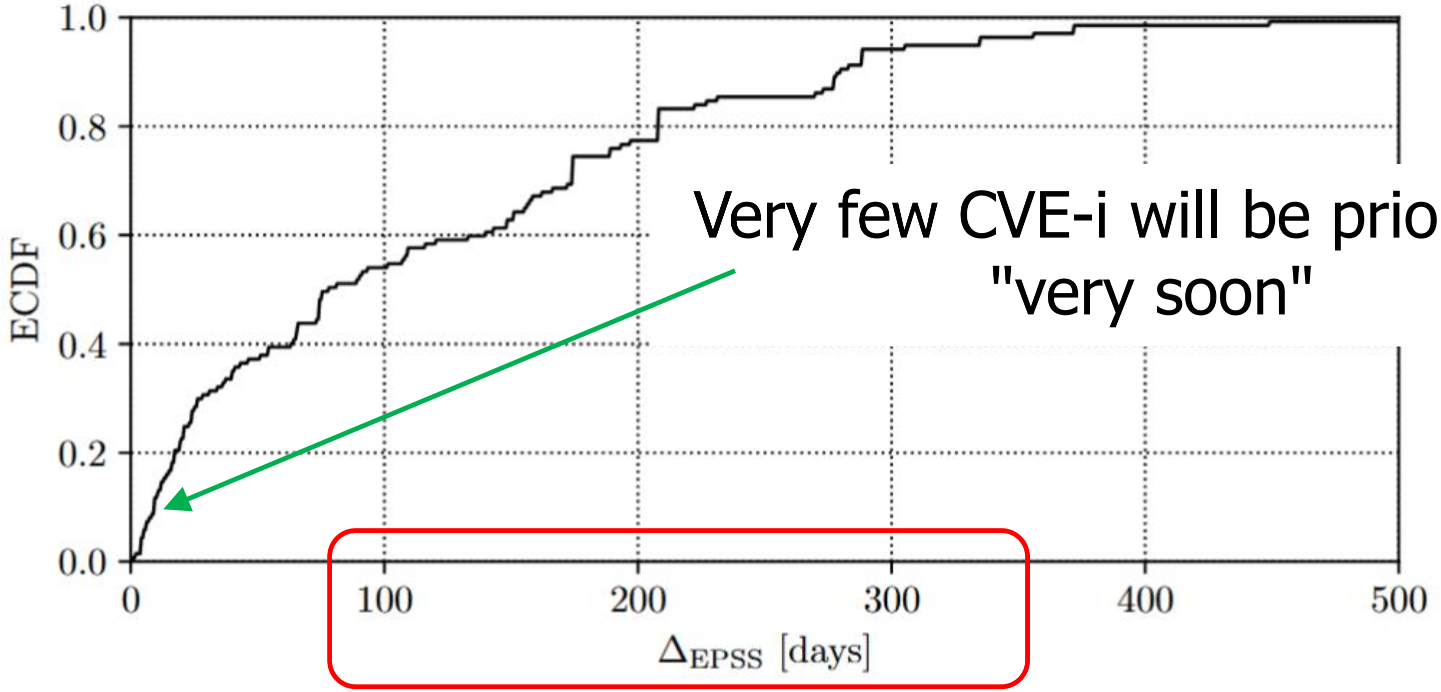

CVE-i
published
Day on which
EPSS(CVE-i) becomes **> 0.7**
Unknown
Days in which CVE-i **exploited**
with high probability

EPSS: Dynamic behavior (III)

- Does EPSS
  - "**predict** actual exploitation by **many** days"?
  - "**predict** actual exploitation by **a few** days"?
  - "**summarizes after** the fact what is **already** happening"?
- We **cannot** answer

- We answer a **different** question:
  How long does it take for the EPSS to reach 0.7?
  - A few **days**?
  - …or a few **weeks**?
  - …or a few **months**?

EPSS: Dynamic behavior (II)

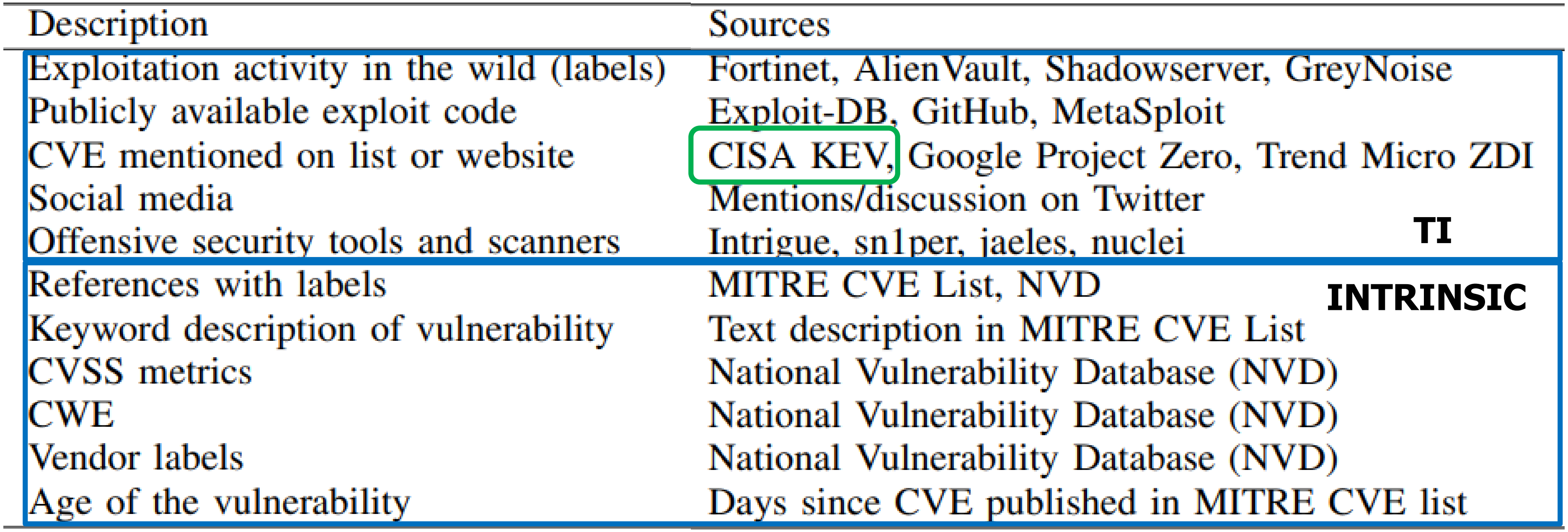


Very few CVE-i will be prioritized "very soon"

Fig. 5. Distribution of the time taken to reach the EPSS threshold for the vulnerabilities in NVD_{EPSS}

CVEs whose EPSS becomes > 0.7

What is the influence of CISA KEV? (I)



Description	Sources
Exploitation activity in the wild (labels)	Fortinet, AlienVault, Shadowserver, GreyNoise
Publicly available exploit code	Exploit-DB, GitHub, MetaSploit
CVE mentioned on list or website	CISA KEV, Google Project Zero, Trend Micro ZDI
Social media	Mentions/discussion on Twitter
Offensive security tools and scanners	Intrigue, sn1per, jaeles, nuclei
References with labels	MITRE CVE List, NVD
Keyword description of vulnerability	Text description in MITRE CVE List
CVSS metrics	National Vulnerability Database (NVD)
CWE	National Vulnerability Database (NVD)
Vendor labels	National Vulnerability Database (NVD)
Age of the vulnerability	Days since CVE published in MITRE CVE list

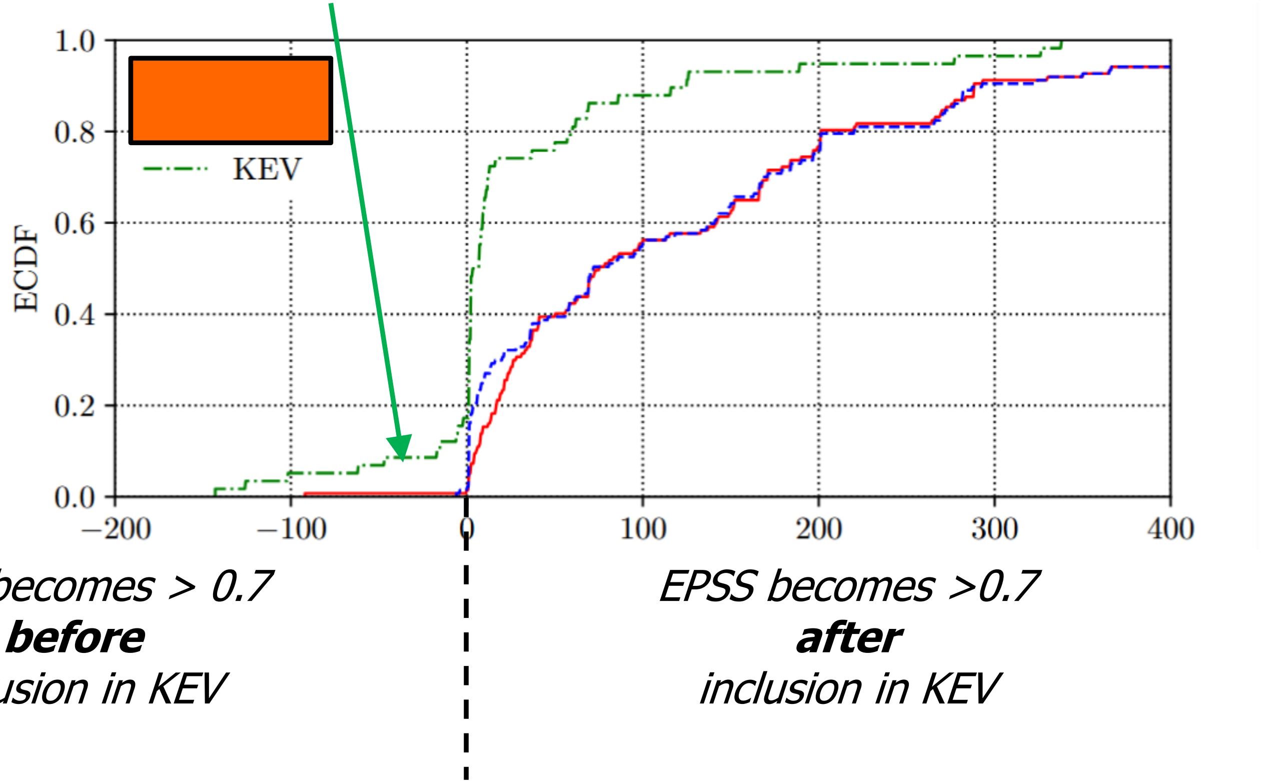
TI

INTRINSIC

- How likely that EPSS(CVE-i) becomes > 0.7 **before inclusion** in CISA KEV ?
- IF not very likely
  THEN probably it is more reactive than predictive

What is the influence of CISA KEV? (II)

Very few CVE-i will be prioritized **before** being inserted in KEV



EPSS becomes > 0.7 ***before*** *inclusion in KEV*

EPSS becomes >0.7 ***after*** *inclusion in KEV*

Remark

- E**P**SS(CVE-i, d):
  - Probability **estimate** that CVE-i **will be** exploited in [d, d+30]
- Updated daily
- Based on:
  - Intrinsic properties of CVE-I
  - What people are saying about CVE-i

- It is called a **predictor**
- …but there are strong indications that it is more a **summary** of what is **already** happening