



Can AI Identify a Software Vulnerability from the Rule That Detects It?

Measuring what intrusion-detection rules reveal about known software vulnerabilities

Paolo Ciappetta¹, Yigitcan Kaya², Christopher Kruegel², Giovanni Vigna²

¹ Antelope Valley College ² University of California, Santa Barbara

STUDY AT A GLANCE

Detection rules describe what an attack looks like on a network, while vulnerability records describe the software flaws behind them. We tested whether an AI system could connect an **unattributed** rule to its exact record among the 208,247 CVEs published since 2021. In a 22-rule pilot, the exact record was identified in **5 cases**.

1 WHY COULD A RULE REVEAL A SOFTWARE FLAW?

An **intrusion detection system (IDS)** watches network traffic for attacks. Its **rules** are tripwires: alert when traffic requests this web address or contains this pattern [1]. A **CVE record** publicly describes one known software flaw [3]. If an attacker obtains an organization's rules, that logic may reveal clues about the flaws it is prepared to detect.

Attribution = deciding which CVE a rule was written to detect.

Exact CVE attribution gives us a clear right-or-wrong test of whether information in the rule can identify the underlying vulnerability.

RESEARCH QUESTION

Can an AI system use the traffic logic in an unattributed IDS rule to identify the exact CVE it was written to detect — and can we tell what made each attempt succeed or fail?

unattributed = CVE number, human-written rule description, and reference links removed.

2 SAME ATTACK, TWO DIFFERENT DESCRIPTIONS

IDS RULE

What traffic should make the detector alert?

```
web address contains /protocol.csp
request includes time=
message contains command
characters
```

CVE RECORD

What went wrong in the software?

The device builds system commands from user input without checking it, allowing an attacker to run commands through a crafted web request.

SAME ATTACK → DIFFERENT LANGUAGE

The rule describes what the detector sees; the CVE describes why the software is vulnerable.

3 HOW WE TESTED IT

gpt-5.3-codex · temperature 0

UNATTRIBUTED IDS RULE [2]

identifying labels removed; traffic-matching logic kept

1 — AI READS THE RULE AND DESCRIBES THE LIKELY FLAW

sees no CVE answers; writes 1–3 competing flaw descriptions in CVE-record wording

2 — SEARCH THE 208,247 CVEs PUBLISHED SINCE 2021 [3]

those descriptions become the search queries — matched by exact wording [4] and by similar meaning [5], then merged into one ranked list [6]

3 — AI COMPARES THE 20 BEST MATCHES

search order and scores hidden; records shuffled

4 — NARROW TO 5 → CHOOSE 1

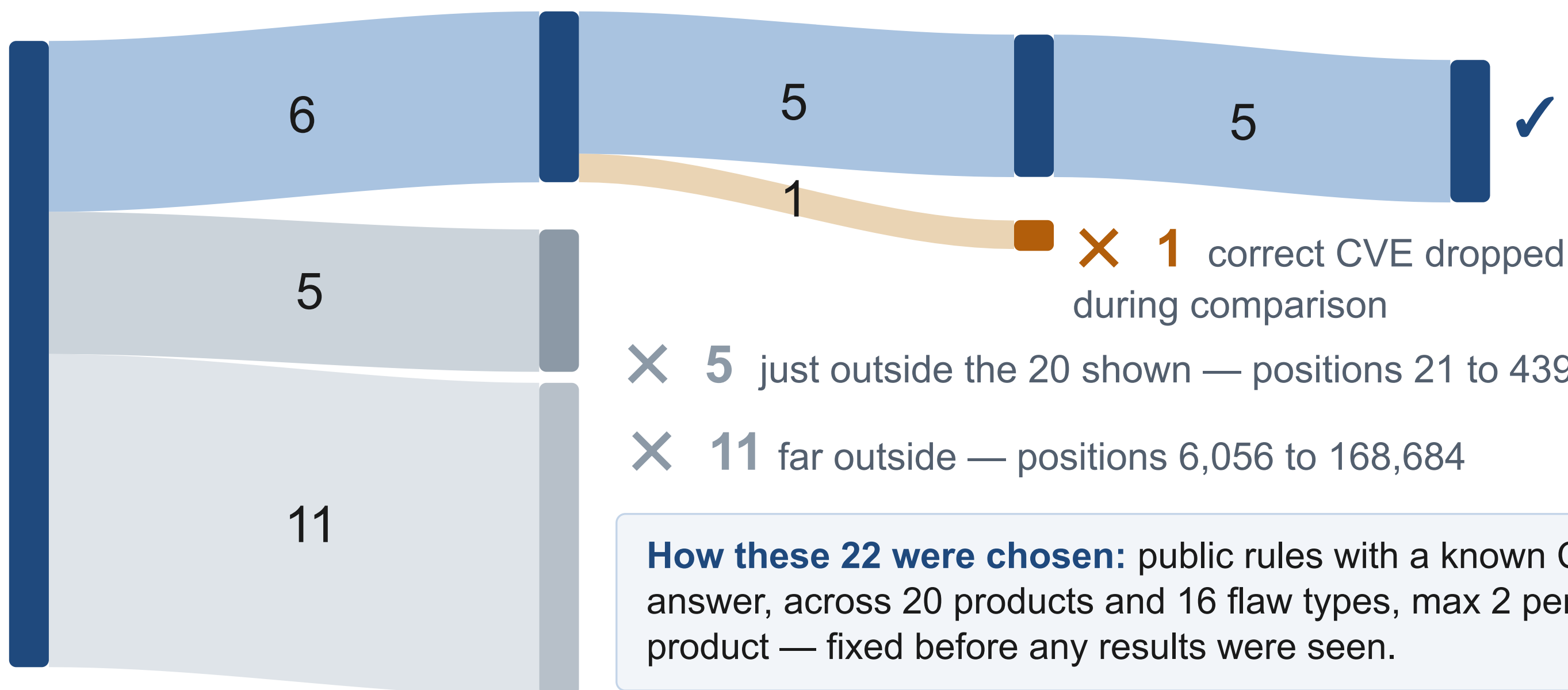
COMPARE WITH THE KNOWN ANSWER

✓ Candidate-blind interpretation

✓ Search ranking hidden

4 WHAT HAPPENED?

22 unattributed rules 6 correct CVEs reached the 20 records shown to the AI 5 remained in the five-record shortlist 5 selected as the final answer



Search's first choice was wrong in **22 of 22** cases, yet comparison selected **5 correct CVEs** from lower search positions.

In this pilot, most failures happened before the AI saw the correct record.

5 WHAT ACTUALLY MADE ATTRIBUTION WORK?

ONE UNIQUE CLUE four successful cases

/api/.../super-connection-token	1 / 208,247
/goform/SetServerConfig	1 / 208,247
/protocol.csp	1 / 208,247
mbus_build_from_csv.php	1 / 208,247

✓ Each string appears in just 1 of 208,247 records: the correct CVE.

Post-hoc analysis of the same frozen CVE collection.

A final accuracy score would not reveal this: four correct answers were strongly enabled by one distinctive string. The staged evaluation exposed that difference.

SEVERAL CLUES TOGETHER one successful case — Redis

The rule never says “Redis.” The AI inferred it from traffic on the port Redis uses.

Redis 14 / 20

Lua 14 / 20

bit 5 / 20

No single clue identified the answer; several clues together were needed.

Failure case: NO MATCHING CLUE MinIO

Rule watches: /minio/bootstrap/v1/verify

✗ 0 / 208,247

CVE records contain that specific web address

• 20 / 20

records shown were MinIO vulnerabilities

The correct CVE was present, but its public description lacked the network detail needed to connect it confidently to the rule.

6 WHAT DID WE LEARN?

ANSWER: YES, IN SOME CASES.

The AI identified the exact CVE in 5 of 22 rules. Separating the process into stages showed why those successes and failures occurred.

Stage-by-stage evaluation reveals not only whether attribution succeeds, but what information makes it possible and where that information becomes ambiguous.

Rules can expose identifying information.

Four successful rules contained a route or filename that uniquely identified the correct CVE record.

Correct answers differed.

Four were strongly driven by distinctive strings; one required combining multiple clues.

Failures had different causes.

Most correct CVEs never reached comparison; one available answer lacked matching network detail.

LIMITATIONS

22-rule pilot; not representative of all IDS rules. Public rules and their CVE links may have appeared in model training data. Labels were removed, but paths, filenames, ports and traffic patterns remained. A deployed rule does not mean the organization is vulnerable. One model, one search setup.

FUTURE WORK

Test rules without distinctive filenames or paths. Improve translation between network clues and CVE wording. Evaluate more rules and models, with memorization controls.