VerificationPolicy regex pattern bypass via substring matching

Moderate **vdemeester** published **GHSA-rmx9-2pp3-xhcr** 2 hours ago

Package

github.com/tektoncd/pipeline (Go)

Affected versions

>= 0.43.0, <= 1.7.0 (also present on main at 0133513db03dadb3cb08801d6b0330badcb63830)

Patched versions

None

Description

hey guys,

triage contract
this is a first-screen summary; deterministic proof is in the proof bundle (canonical.log/control.log/witness.txt).

summary
trusted resources verification policies match a resource source string (`refSource.URI`) against `spec.resources[].pattern` using `regexp.MatchString`. in go, `regexp.MatchString` reports a match if the pattern matches anywhere in the string, so common unanchored patterns (including examples in tekton documentation) can be bypassed by attacker-controlled source strings that contain the trusted pattern as a substring. this can cause an unintended policy match and change which verification mode/keys apply.

pins

- repo: https://github.com/tektoncd/pipeline
- commit: `0133513`
- callsite: pkg/trustedresources/verify.go:118-137 (getMatchedPolicies)

severity
MEDIUM (provisional CVSS 5.3–6.5) (signing request tampering)

repro (canonical)

- command: unzip -q -o poc.zip -d poc && cd poc/poc-F-TEKTON-REGEX-001 && make canonical
- expected: cap not reached; canonical does not emit the vulnerability markers.
- actual: cap reached; canonical emits the vulnerability markers.
- canonical markers (mandatory): [CALLSITE_HIT] + [PROOF_MARKER]

negative control

- command: unzip -q -o poc.zip -d poc && cd poc/poc-F-TEKTON-REGEX-001 && make control
- expected: cap not reached under the same harness; control emits the control marker and does not emit the vulnerability markers.
- control markers (mandatory): [CALLSITE_HIT] + [NC_MARKER]

fix
consider making matching safe-by-default by requiring full-string matches (or validating patterns and documenting substring semantics clearly). one option is to anchor patterns before matching (e.g., wrap `pattern` as `^(?:pattern)$` when not already anchored), or to provide a separate field for exact match vs regex match.
fix accepted when: under the same harness, canonical still hits [CALLSITE_HIT] but does not emit [PROOF_MARKER].

proof bundle pointers

- bundle: poc.zip
- bundle convention: zip extracts under a single top-level folder (poc-F-TEKTON-REGEX-001/) to avoid collisions
- contains: canonical.log, control.log, witness.txt
- extracted paths: after extraction, see ./poc/poc-F-TEKTON-REGEX-001/canonical.log, ./poc/poc-F-TEKTON-REGEX-001/control.log, ./poc/poc-F-TEKTON-REGEX-001/witness.txt
- verify: compare shasum -a 256 for canonical.log/control.log/fix.patch/test source against witness.txt
- supported-mode note: if your supported integration uses verified https app-links/universal links only, provide the supported tag/branch and we can retest on that pin.

poc.zip

---

impact
an attacker can craft a trusted resources source string that embeds a trusted substring and still matches an unanchored verificationpolicy `spec.resources[].pattern`, even if the policy is intended to constrain matches to a specific trusted source. this occurs because `regexp.MatchString` succeeds on substring matches, so patterns like `https://github.com/tektoncd/catalog.git` match attacker-controlled sources such as `https://evil.com/?x=https://github.com/tektoncd/catalog.git`.

affected: deployments using trusted resources verification with unanchored verificationpolicy patterns, where an attacker can influence the `refSource.URI` value used for policy matching.

not affected: deployments that anchor all patterns (`^...$`) or otherwise enforce full-string matching; deployments where attackers cannot influence `refSource.URI`.

steps to reproduce

```
unzip -q -o poc.zip -d /tmp/poc-tekton-regex-001
cd /tmp/poc-tekton-regex-001/poc-F-TEKTON-REGEX-001
bash ./run.sh canonical | tee /tmp/tekton-regex-001-canonical.log
bash ./run.sh control | tee /tmp/tekton-regex-001-control.log
grep -n '\\[PROOF_MARKER\\]' /tmp/tekton-regex-001-canonical.log && grep -n '\\[NC_MARKE
```

suggested patch options:

- make matching safe-by-default by anchoring patterns before matching (or by validating and rejecting unanchored patterns).
- document the substring semantics explicitly and update documentation examples to include anchors.

workarounds
anchor verificationpolicy resource patterns so they must match the full source string. example:

- `^https://github.com/tektoncd/catalog\\.git$`

best,
oleh

Severity

Moderate 6.5 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	High
Availability	None

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N

CVE ID

CVE-2026-25542

Weaknesses

▶ CWE-185

Credits

1seal	Reporter
offset	Finder
vdemeester	Remediation developer