Custom GitHub Copilot Agent Approval Request

Date: December 10, 2025
Requestor: [Your Name]
Department: [Your Department]
Purpose: Enable creation of security-focused custom agents for vulnerability remediation

---

Executive Summary

We request approval to create and deploy a custom GitHub Copilot agent in our development environment. This agent will automate detection and remediation of Snyk security vulnerabilities across our repositories, improving our security posture and reducing remediation time.

What is a Custom Copilot Agent?

A custom agent is a specialized automation tool that extends GitHub Copilot's built-in coding agent with specific instructions, tools, and workflows tailored to our organization's needs. It is defined in code (`.github/agents/`) and version-controlled like any other repository artifact[1].

Key characteristics:

- Operates only within authorized repositories with explicit user action
- Runs under the same permissions as the triggering user
- Cannot access CI secrets or files outside the repository scope
- All actions are auditable and visible in GitHub's activity logs

---

Proposed Agent: Snyk Vulnerability Fixer

Objective: Automate the process of identifying, analyzing, and proposing fixes for Snyk-reported security vulnerabilities.

Scope:

- Read repository code and Snyk reports
- Analyze vulnerability details (type, severity, remediation steps)
- Edit source files to apply fixes (e.g., dependency updates, code patches)
- Create pull requests for review before any changes are merged

Limitations:

- The agent cannot push directly to `main` or `master` branches; all changes go to `copilot/*` branches
- The agent cannot access GitHub tokens with elevated permissions

- The agent can only operate in repositories where it is configured
- Pull requests require human review and approval before merging

---

Security Controls & Mitigations

Data Protection

- **No training data leakage:** Code in custom agents is not used to train Copilot models[1]
- **No CI secrets access:** Agent cannot read or exfiltrate CI/CD secrets or environment variables
- **Scoped context:** Agent only sees files in the current repository
- **Session-based tokens:** Agent tokens are revoked after each session[1]

Access Control

- **Write access required:** Only team members with write access to a repository can trigger the agent[1]
- **PR-based workflow:** All edits are staged as pull requests requiring human review
- **Branch restrictions:** Agent can only commit to `copilot/` prefixed branches[1]
- **Audit logs:** All agent actions are logged and visible in GitHub's activity feed

Agent Governance

- **Version control:** Agent configuration files (`.github/agents/*.agent.md`) are tracked in Git
- **Code review:** Changes to agent behavior go through standard pull request review
- **Organization-wide policies:** Enterprise can enforce agent policies via ruleset configuration[1]
- **User authorization:** Agent only acts on issues/PRs assigned or triggered by authorized users

---

Transparency & Accountability

- **No invisible directives:** All agent instructions are stored in public or controlled repository files
- **Prompt visibility:** Security team can review the exact prompt/instructions driving the agent
- **Audit trail:** GitHub audit logs track which user triggered the agent and what changes it made
- **No external calls:** Agent does not have internet access to exfiltrate data to external systems

---

Implementation Plan

Phase 1: Repository Setup (Week 1)

- Create `.github/agents/snyk-fixer.agent.md` in target repositories
- Define agent configuration (name, description, allowed tools)
- Embed Snyk remediation workflow as instructions

Phase 2: Testing (Week 2)

- Security team reviews agent definition
- Pilot on non-critical repository with controlled issues
- Validate that edits match expectations

Phase 3: Deployment (Week 3)

- Expand to additional repositories as approved
- Monitor agent activity and logs
- Gather feedback from development teams

Phase 4: Governance (Ongoing)

- Monthly review of agent-generated PRs
- Quarterly audit of agent configuration
- Adjust permissions and scope as needed

Risk Assessment

Risk	Likelihood	Severity	Mitigation
Unintended code changes	Low	Medium	PR review gate + testing in non-critical repos first
Prompt injection attacks	Low	Medium	All agent instructions in version-controlled files (no hidden inputs)
Over-privileged edits	Low	High	Branch restrictions + branch protection rules for `main`/`master`
Agent misuse	Very Low	Medium	Write access check + audit logging

---

Benefits

- **Faster vulnerability resolution:** Automate routine remediation steps
- **Consistency:** Apply standardized fix patterns across all repositories
- **Developer focus:** Free engineers to focus on complex vulnerabilities
- **Compliance:** Demonstrate proactive security posture and rapid remediation
- **Auditability:** All agent actions logged and traceable

Security Team Review Checklist

- [ ] Agent instructions reviewed and approved
- [ ] No sensitive data (API keys, tokens) embedded in agent config
- [ ] Branch protection rules configured for `main`/`master`
- [ ] Audit logging enabled
- [ ] Initial pilot repository identified and approved
- [ ] Escalation process defined (who to contact if agent behaves unexpectedly)

Questions & Contact

Security Review Contact: [Security Team Lead Name] – [email]
Agent Configuration Owner: [Your Name] – [Your Email]

For technical questions about custom agents, see the official GitHub documentation on preparing custom agents in an enterprise[1].

References

[1] GitHub. (2024). "Preparing to use custom agents in your enterprise." GitHub Docs. https://docs.github.com/en/copilot/how-tos/administer-copilot/manage-for-enterprise/manage-agents/prepare-for-custom-agents

[2] GitHub. (2025). "How GitHub's agentic security principles make our AI agents as secure as possible." GitHub Blog. https://github.blog/ai-and-ml/github-copilot/how-githubs-agentic-security-principles-make-our-ai-agents-as-secure-as-possible/

[3] GitHub. (2024). "Creating custom agents." GitHub Docs. https://docs.github.com/en/copilot/how-tos/use-copilot-agents/coding-agent/create-custom-agents