
AI In Security (Bug Bounty - Red Team)



Who am I?

Ramin Farajpour Cami

Software | Security | Blockchain (Web3 – Solana) Engineer

Rust - Golang - Python

Github : <https://github.com/raminfo>

X (Twitter): <https://x.com/realraminfo>

Offensive & Defensive Security with Artificial Intelligence

Topics:

AI-Powered Attack Simulation

AI Reconnaissance & Exploitation

Supply Chain Security

AI Code Review

Simulation Attack

Use AI to predict server behavior
and automatically generate attack scenarios

Attack Vectors Generated by AI:

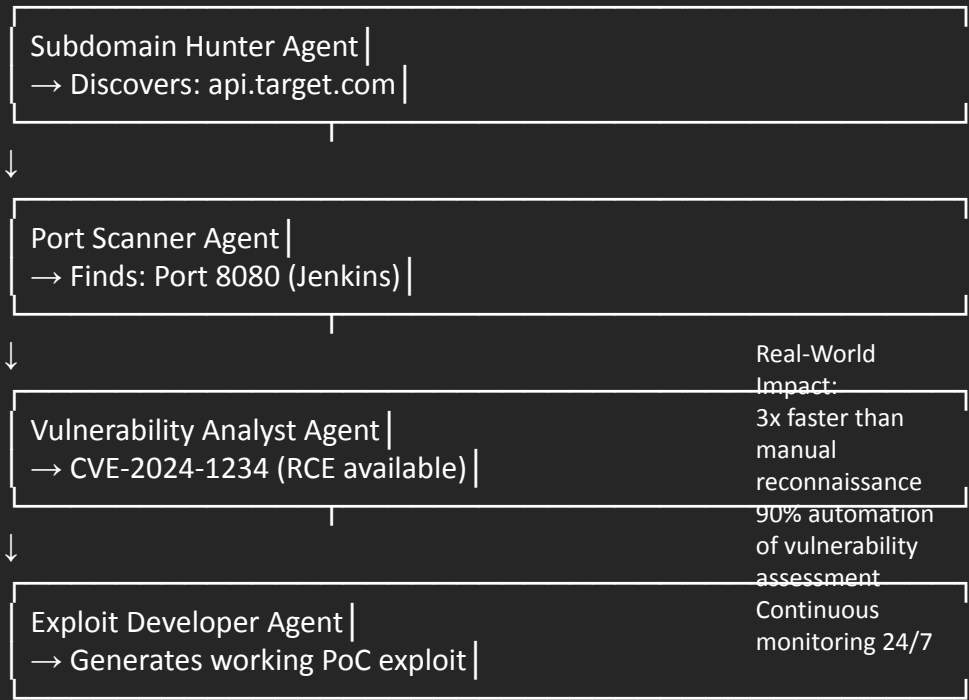
- ✓ Price Manipulation -
Changing prices client-side
- ✓ Race Conditions -
Concurrent request exploitation
- ✓ JWT Tampering -
Token manipulation
- ✓ Input Validation Bypass -
Smart fuzzing

Benefits:

- ✓ Automated:
No manual test case creation
- ✓ Intelligent:
Learns from API responses
- ✓ Comprehensive:
Tests all OWASP Top 10

Intelligent Reconnaissance & Vulnerability Discovery

Multi-Agent Architecture:



Traditional Recon vs AI Recon

Traditional	AI-Powered
Manual subdomain enum	Autonomous agent discovery
Sequential port scans	Parallel intelligent scanning
Static vulnerability list	Dynamic exploit generation
Human analysis	AI decision-making

Supply Chain Attack Detection

The Supply Chain ThreatRecent Incidents:

event-stream (2018): 8M weekly downloads,
backdooredua-parser-js (2021): Crypto miner injected
node-ipc (2022): Destructive malware in protest
PyTorch (2022): Dependency confusion attack

AI-Powered Detection

Benefits:

AI analyzes package for:

- ✓ Network calls during install
- ✓ File system access patterns
- ✓ Obfuscated code
- ✓ Suspicious hooks/scripts
- ✓ Unusual maintainer changes

Supply Chain Attack Detection


Real-Time Scanning with Socket.dev

\$ socket scan .

- ⚠ HIGH RISK: malicious-package@1.0.0
- 🚨 Install script makes network request
 - 🚨 Obfuscated JavaScript detected
 - 🚨 Accesses sensitive environment variables
 - 🚨 New maintainer (account created 2 days ago)
- Recommendation: BLOCK and report

Socker.dev Bot


socket-security bot commented on Mar 12 • edited ▾

 **Potential security issues detected.** Learn more about [Socket for GitHub](#) ↗

To accept the risk, merge this PR and you will not be notified again.

Alert	Package	Note	Source
Potential typo squat	npm/bowserify@10.2.1	<ul style="list-style-type: none">• Did you mean: browserify (76 thousand times more downloads)	<ul style="list-style-type: none">• <code>package.json</code>

[View full report](#) ↗



AI Code Review

AI Code Review Solution:

Using Google Gemini / Cursor AI

Input: Source code Output: Security analysis in seconds

Cursor AI Integration

Workflow:

1. Developer writes code
2. Cursor AI analyzes in real-time
3. Suggests secure alternatives
4. Developer accepts/modifies
5. Secure code committed

AI Code Review – Gemini

```
// 2. --- Prepare the prompt for Gemini, now including a grading instruction ---
const prompt = `
You are an expert code reviewer for a project that uses Protobuf as the API definition language.
Your task is to review the following code changes from a pull request, provided in a 'diff' format.

Please focus on the following:
- Protobuf Best Practices: Check for clear naming, correct data types, future-proof field numbering, etc.
- Potential Issues: Identify breaking changes, logical inconsistencies, or areas that could be confusing.
- Clarity and Documentation: Ensure comments are helpful.

CRITICAL: After your review, you MUST provide an overall quality grade for the changes on a new line, using the exact format: \Overall G
- A+/A: Excellent, ready to merge.
- B: Good, but with minor suggestions.
- C: Acceptable, but has non-trivial issues that should be addressed.
- F: Has significant flaws and requires major revisions.

Here is the diff:
\`\`\`diff
${diff}
\`\`\`
`;
```

AI Code Review – Gemini

```
// 3. --- Call the Gemini API ---
const geminiApiKey = process.env.GEMINI_API_KEY;
const url = `https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=${geminiApiKey}`;
let reviewText = "Could not retrieve review from Gemini.";
try {
  // (API call logic remains the same)
  const response = await fetch(url, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ contents: [{ parts: [{ text: prompt }] }] })
  });
  if (!response.ok) throw new Error(`API request failed with status ${response.status}: ${await response.text()}`);
  const responseData = await response.json();
  if (responseData.candidates && responseData.candidates.length > 0) {
    reviewText = responseData.candidates[0].content.parts[0].text;
  } else {
    reviewText = "Gemini returned an empty or blocked response. This may be due to a content safety filter.";
  }
} catch (error) {
  reviewText = `Sorry, an error occurred while generating the review: ${error.message}`;
}
```

AI Code Review – Gemini

 **feat: enhance TokenAddress message with additional fields** ✓ ai-reviewed breaking buf skip breaking

Grade: B

#20 by raminfp was merged 3 weeks ago

 **fix: change transaction_index field type in Transaction message from ...** ✓ ai-reviewed breaking

buf skip breaking Grade: A

#19 by raminfp was merged 3 weeks ago

AI Code Review – Gemini

Merged feat: implement TRON provider for blockchain integration #10
raminf merged 14 commits into dev from feat/tron_provid... last week

cursor bot reviewed 2 weeks ago [View reviewed changes](#)

cursor bot left a comment

This PR is being reviewed by Cursor Bugbot

► Details

😊

```
pkg/chain_provider/tron_provider/impl.go
```

```
451 +         addr2 := "41" + data[72:104]
452 +         addresses = append(addresses, addr2)
453 +     }
454 + }
```

cursor bot 2 weeks ago

Bug: Incorrect Address Extraction in Smart Contract Calls

In `processTransaction`, the logic for extracting addresses from `TronTriggerSmartContract` call data uses incorrect hex string indexing. The `data` field is a hex string, but the current slicing extracts wrong portions of ABI-encoded addresses, potentially causing relevant transactions to be missed.

AI Code Review – Gemini

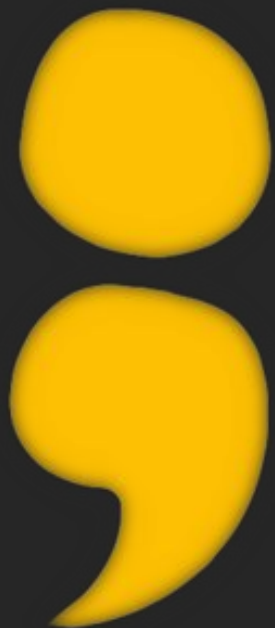
```
// 3. --- Call the Gemini API ---
const geminiApiKey = process.env.GEMINI_API_KEY;
const url = 'https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=${geminiApiKey}';
let reviewText = "Could not retrieve review from Gemini.";
try {
  // (API call logic remains the same)
  const response = await fetch(url, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ contents: [{ parts: [{ text: prompt }] }] })
  });
  if (!response.ok) throw new Error(`API request failed with status ${response.status}: ${await response.text()}`);
  const responseData = await response.json();
  if (responseData.candidates && responseData.candidates.length > 0) {
    reviewText = responseData.candidates[0].content.parts[0].text;
  } else {
    reviewText = "Gemini returned an empty or blocked response. This may be due to a content safety filter.";
  }
} catch (error) {
  reviewText = `Sorry, an error occurred while generating the review: ${error.message}`;
}
```



DEMO

BOOKLET

<https://myai-e4q.pages.dev/>



www.Ravro.ir



support@Ravro.ir



۰۲۱-۹۱۰۳۵۳۱۵



1578775488



تهران، خیابان مطهری، نبش سهروردی، پلاک ۹۴، طبقه دوم، واحد ۲۵۰



[Ravro_ir](#)