New issue

Vulnerability Report: Code Injection Vulnerability in __run_compiled_instructions function of GPTSeleniumAgent class due to usage of exec() with unverified input #20

Open

ybdesire opened 2 weeks ago

Description

The code snippet `__run_compiled_instructions` within the `GPTSeleniumAgent` class is vulnerable to a CWE - 94: Code Injection vulnerability. The function uses the `exec()` function to execute the `instructions` parameter, which is obtained from the `InstructionCompiler`. However, these instructions are not adequately verified.

related code:

browserpilot/browserpilot/agents/gpt_selenium_agent.py
Line 253 in `0c76ea7`

```
253     exec(instructions, globals(), ldict)
```

If an attacker can manipulate the input that is passed to the `InstructionCompiler` and subsequently included in the `instructions` variable, they can inject malicious Python code. When the `exec()` function is called, this malicious code will be executed within the context of the running program. This could lead to a wide range of security issues, such as unauthorized access to system resources, data leakage, or even complete system compromise.

Exploit

An attacker can exploit this vulnerability by crafting malicious input that is passed to the `InstructionCompiler`. Here is a step - by - step guide on how an attacker might exploit this vulnerability:

1. **Identify the Input Point**: The attacker needs to find out where the input is provided to the `InstructionCompiler`. This could be through a user interface, an API endpoint, or a configuration file.
2. **Craft Malicious Code**: The attacker creates a malicious Python code snippet. For example, the following code can be used to read sensitive files on the system:

```
import os; print(os.popen('cat /etc/passwd').read())
```

3. **Inject the Malicious Code**: The attacker inserts the crafted malicious code into the input that is passed to the `InstructionCompiler`.
4. **Trigger the Execution**: Once the malicious input is processed by the `InstructionCompiler`, the resulting `instructions` variable will contain the malicious code. When the `__run_compiled_instructions` function is called and the `exec()` function is executed, the malicious code will be run.

As a result, the attacker can gain unauthorized access to sensitive information, modify system settings, or perform other malicious actions depending on the permissions of the running process.

Impacted version

all versions