1Panel-dev / **MaxKB** Public

Code | Issues 69 | Pull requests 6 | Actions | Security and quality 19

# Commit `38c4cfe`

**liqiang-fit2cloud** committed 3 weeks ago · ✓ 1 / 1

```
security: fix spoofing bypass in tool execution.
```

v2 · v2.8.0

1 parent 158d417 commit 38c4cfe

## 1 file changed +19 -14 lines changed

Top

Filter files…

apps/common/utils

tool_code.py

## 1 file changed +19 -14 lines changed

Search within code

apps/common/utils/tool_code.py

```
@@ -13,6 +13,7 @@
13 13  import sys
14 14  import tempfile
15 15  import time
   16 +import secrets
16 17  from contextlib import contextmanager
17 18  from contextlib import suppress
18 19  from textwrap import dedent
@@ -90,7 +91,7 @@ def init_sandbox_dir():
90 91              maxkb_logger.error(f'Exception: {e}', exc_info=True)
91 92
92 93          def exec_code(self, code_str, keywords, function_name=None):
93    -             _id = str(uuid.uuid7())
```

```
     94 +          _id = secrets.token_hex(32)
 94  95            action_function = f'({function_name !a}, locals_v.get({function_name !a}))' if function_name else 'locals_v.popitem()'
 95  96            set_run_user = f'os.setgid({pwd.getpwnam(_run_user).pw_gid});os.setuid({pwd.getpwnam(_run_user).pw_uid});' if _enable_sandbox else ''
 96  97            _exec_code = f"""
         @@ -100,6 +101,7 @@ def exec_code(self, code_str, keywords, function_name=None):
100 101        path_to_exclude = ['/opt/py3/lib/python3.11/site-packages', '/opt/maxkb-app/apps']
101 102        sys.path = [p for p in sys.path if p not in path_to_exclude]
102 103        sys.path += {_sandbox_python_sys_path}
     104 +      _id = os.environ.get("_ID")
103 105        locals_v={{}}
104 106        keywords={keywords}
105 107        globals_v={{}}
         @@ -110,29 +112,31 @@ def exec_code(self, code_str, keywords, function_name=None):
110 112            f_name, f = {action_function}
111 113            globals_v.update(locals_v)
112 114            exec_result=f(**keywords)
113     -      sys.stdout.write("\\n{_id}:")
114     -      json.dump({{'code':200,'msg':'success','data':exec_result}}, sys.stdout, default=str)
     115 +      result = {{'code':200,'msg':'success','data':exec_result}}
115 116    except Exception as e:
116 117        if isinstance(e, MemoryError): e = Exception("Cannot allocate more memory: exceeded the limit of {_process_limit_mem_mb} MB.")
117     -      sys.stdout.write("\\n{_id}:")
118     -      json.dump({{'code':500,'msg':str(e),'data':None}}, sys.stdout, default=str)
119     - sys.stdout.write("\\n")
120     - sys.stdout.flush()
     118 +      result = {{'code':500,'msg':str(e),'data':None}}
     119 + finally:
     120 +      sys.stdout.write("\\n" + _id)
     121 +      json.dump(result, sys.stdout, default=str)
     122 +      sys.stdout.write("\\n__END__\\n")
     123 +      sys.stdout.flush()
121 124    """
```

```
122     -           maxkb_logger.debug(f"Sandbox execute code: {_exec_code}")
        125 +       maxkb_logger.debug(f"Tool execution({_id}) execute code: {_exec_code}")
123     126         with tempfile.NamedTemporaryFile(mode='w', suffix='.py', delete=True)
                    as f:
124     127             f.write(_exec_code)
125     128             f.flush()
126     129             with execution_timer(_id):
127     -                   subprocess_result = self._exec(f.name)
        130 +               subprocess_result = self._exec(f.name, _id)
128     131         if subprocess_result.returncode != 0:
129     132             raise Exception(subprocess_result.stderr or
                    subprocess_result.stdout or "Unknown exception occurred")
130     133         lines = subprocess_result.stdout.splitlines()
131     -           result_line = [line for line in lines if line.startswith(_id)]
132     -           if not result_line:
133     -               maxkb_logger.error("\n".join(lines))
        134 +       if len(lines) < 2 or lines[-1] != "__END__":
        135 +           raise Exception("Execution interrupted or tampered")
        136 +       last_line = lines[-2]
        137 +       if not last_line.startswith(_id):
134     138             raise Exception("No result found.")
135     -           result = json.loads(result_line[-1].split(":", 1)[1])
        139 +       result = json.loads(last_line[len(_id):])
136     140         if result.get('code') == 200:
137     141             return result.get('data')
138     142         raise Exception(result.get('msg') + (f'\n{subprocess_result.stderr}' if
                    subprocess_result.stderr else ''))
                    @@ -286,9 +290,10 @@ def get_app_mcp_config(self, api_key):
286     290         }
287     291         return app_config
288     292
289     -   def _exec(self, execute_file):
        293 +   def _exec(self, execute_file, _id):
290     294         kwargs = {'cwd': BASE_DIR, 'env': {
291     295             'LD_PRELOAD': f'{_sandbox_path}/lib/sandbox.so',
        296 +           '_ID': _id,
292     297         }}
293     298         def _set_resource_limit():
```

```
294   299                 if not _enable_sandbox or not sys.platform.startswith("linux"):
              return
```

## Comments 0

Please sign in to comment.