최신 웹 공격 관련 연구 결과 소개

> [19반] 정민석_7000

CVE-2024-46981

내용 요약

해당 취약점은 2025-01-06에 발표된 Redis 관련 취약점입니다. Redis는 온메모리 DB 중 하나로 많은 어플리케이션에서 캐싱 등을 목적으로 활용하고 있습니다. 해당 취약점은 Lua 스크립트를 활용하여 garbage collector를 조작하고, RCE로 이어질 가능성을 내포합니다.

취약점의 근본 원인 및 공격방법

PoC 링크[1]에는 lua 스크립트와, 이를 Redis에서 실행하게 만드는 exploit.py 코드가 있습니다. 해당 코드는 Lua 스크립트를 통해서 garbage collector의 메모리를 조작할 수 있습니다. 즉, Redis에서 Lua 스크립트를 실행할때, garbage collector를 조작할 수 있습니다. 이를 통해 최종적으로 RCE로 이어질 수 있습니다.

공격의 파급도

Redis는 전세계적으로 많이 이용하는 온메모리 DB입니다. 메모리 조작으로 인한 RCE가 일어날 수 있는 만큼 주의를 기울어야할 것입니다. CVSS는 7.0 HIGH를 기록하였습니다.

방어 방법

Redis는 342ee42 커밋[2]에서 src/eval.c와 src/function_lua.c에 다음의 코드를 추가하는 것으로 취약점을 패치하였습니다.

- src/eval.c

```
        unsigned int lua_tcache = (unsigned int)(uintptr_t)ud;
#endif

+   lua_gc(lua, LUA_GCCOLLECT, 0);
    lua_close(lua);
```

```
#if defined(USE_JEMALLOC)
```

- src/function_lua.c

```
   unsigned int lua_tcache = (unsigned int)(uintptr_t)ud;
#endif

+   lua_gc(lua_engine_ctx->lua, LUA_GCCOLLECT, 0);
    lua_close(lua_engine_ctx->lua);
    zfree(lua_engine_ctx);
```

[3]의 Lua 공식문서를 참고한 결과 `lua_gc` 는 garbage collector를 제어하는 함수이며, `LUA_GCCOLLECT` 인자는 전체 가비지 수집 주기를 수행하도록 하는 파라미터입니다. 즉, 기존에 위험하게 lua를 실행하던 코드를 `lua_gc` 로 안전하게 실행하도록 패치한 것으로 추측됩니다.

해당 취약점은 7.4.2, 7.2.7, 혹은 6.2.17 이상에서 패치되었으며, 사용자는 해당 버전 이상으로 업그레이드 하거나, Lua 스크립트 실행을 막아야할 것입니다.

취약점 관련 자료를 수집한 방법

- [1]의 경우 Github 에 CVE-2024-46981을 검색하였습니다.
- [2]의 경우 Github의 Redis 레포지터리에서, 발표일(2025/1/6) 인근의 Commit 로그 중, lua의 키워드로 검색하였습니다.
- [3]의 경우 구글 검색에 lua_gc를 검색하였습니다.
- [4]의 경우 구글에 CVE-2024-46981을 검색하였습니다.

[참고 문헌]

- [1] https://github.com/p333zy/poc-redis/blob/master/exploit.py
- [2] https://github.com/redis/redis/commit/342ee426ad0d0731b2272553bd4db2cd78e24772
- [3] https://www.lua.org/manual/5.3/manual.html
- [4] https://nvd.nist.gov/vuln/detail/CVE-2024-46981