Disassembler

Disassembler funguje jako „překladač pro detektivy“. Zatímco běžný překladač (compiler) mění kód napsaný člověkem (např. v C++) na kód pro stroj, disassembler se pokouší tento proces obrátit. Výsledkem však není původní zdrojový kód, ale nízkoúrovňové instrukce procesoru.

1. Jak disassembler pracuje?

Program čte binární soubor (např. .exe, .dll, .bin) a podle tabulek instrukční sady daného procesoru (např. x86, ARM) přiřazuje číselným hodnotám (operačním kódům - opcodům) jejich textové zkratky (mnemoniky).

Příklad transformace:

- **Binární data:** `B8 05 00 00 00`
- **Assembler:** `MOV EAX, 5` (Instrukce říká: „Vlož číslo 5 do registru EAX“)

2. Klíčové funkce moderních disassemblerů

- **Analýza toku řízení (Control Flow Graph):** Program vykreslí diagram, který ukazuje, jak se program větví (podmínky, cykly, skoky).
- **Identifikace funkcí:** Pokročilé nástroje dokáží rozpoznat standardní knihovní funkce (např. z Visual C++) a automaticky je pojmenovat.
- **Křížové odkazy (Cross-references):** Ukáže vám všechna místa v programu, která volají konkrétní funkci nebo přistupují k určitému textovému řetězci.

3. Nejpoužívanější nástroje

Název	Charakteristika
IDA Pro	Průmyslový standard. Extrémně výkonný, ale velmi drahý. Obsahuje interaktivní grafy.
Ghidra	Open-source nástroj vyvinutý americkou NSA. Nabízí funkce srovnatelné s IDA zdarma.
Hiew	Legendární textový disassembler a hex editor, oblíbený pro rychlé modifikace (patchování).
Capstone	Disassemblery engine (knihovna), na kterém staví mnoho dalších bezpečnostních nástrojů.

4. Disassembler vs. Debugger vs. Dekompilátor

Je důležité tyto nástroje nezaměňovat:

- **Disassembler:** Statický nástroj. Kód nezkoumá za běhu, jen ho „čte“.
- **Debugger:** Dynamický nástroj. Umožňuje program spustit, zastavit na určitém místě a sledovat měnící se hodnoty v paměti.
- **Dekompilátor:** Pokročilejší verze disassembleru, která se snaží převést assembler do jazyka C

nebo jiného vyššího jazyka pro snazší pochopení logiky.

5. Praktické využití

- **Analýza malwaru:** Zjišťování, co virus dělá, aniž by musel být spuštěn.
- **Hledání zranitelností:** Hledání chyb v kódu, které by mohl zneužít hacker (např. přetečení bufferu).
- **Crackování softwaru:** Odstraňování ochran (DRM) změnou instrukcí typu „pokud není licence, skonči“ na „pokud není licence, pokračuj“.
- **Vývoj ovladačů:** Pochopení toho, jak komunikuje hardware, ke kterému výrobce nevydal dokumentaci.

> **Zajímavost:** Práce s disassemblerem je jako čtení knihy v cizím jazyce, kde chybí interpunkce, názvy kapitol a jména postav. Analytik musí být trpělivý a postupně si kód „otagovat“ vlastními poznámkami, aby pochopil jeho příběh.

Zpět na Vývoj a Bezpečnost