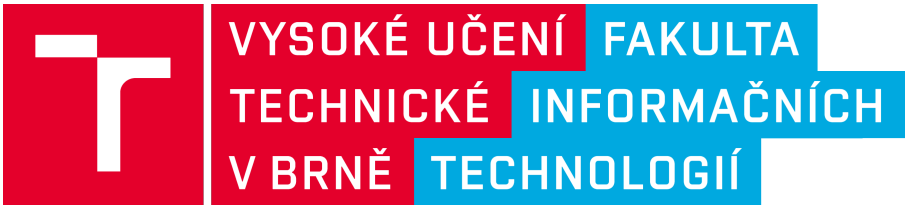

BIS Projekt 2022

Vojtěch Fiala (xfiala61)

1 Úvod

Tato dokumentace popisuje postup vypracování projektu do předmětu BIS[1]. Obsahuje popis sítě a způsob získání jednotlivých tajemství, která se mi povedlo odhalit.

2 Schéma sítě

S využitím příkazu **ifconfig** jsem si po připojení na domovskou stanici zobrazil informace o síťovém rozhraní. Zjistil jsem, že maska sítě je `255.255.255.0` a že broadcast adresa je `192.168.122.255`, z čehož plyne, že síť je na adresách `192.168.122.0-192.168.122.255`. S pomocí nástroje **nmap** jsem tedy síť zmapoval příkazem `nmap -P 192.168.122.0/24`, čímž jsem zjistil přítomnost zařízení a otevřených portů na následujících adresách:

IP adresa	Název	Otevřené porty
192.168.122.1	_gateway	22 (ssh), 53 (dns)
192.168.122.3	/	22 (ssh)
192.168.122.35	fedora	22 (ssh), 9090 (zeus-admin)
192.168.122.90	/	22 (ssh)
192.168.122.149	/	22 (ssh), 5432 (postgreSQL)
192.168.122.235	server2	22 (ssh)

3 192.168.122.35 – Domovská stanice / fedora

Po připojení na domovskou stanici jsem jako první zkontroloval její obsah. Ukázalo se, že obsahuje v domovském adresáři 3 viditelné složky – *jsapp*, *myprog* a *library*.

3.1 jsapp

Jako první jsem si prohlédnul složku *jsapp*, která obsahuje jednoduchou webovou aplikaci. Po průzkumu zdrojových kódu jsem zjistil, že se v kódu nachází

[1] `https://www.fit.vut.cz/study/course/BIS/.cs`

funkce *checkpasswd*. Ta byla ale tzv. *obfuscated*[2], nicméně po analýze tohoto kódu jsem přišel na to, že si stačí proměnnou, vůči které se heslo porovnává, vypsat do konzole, nehledě na kód, který ji produkuje. Tím jsem získal heslo *6f2624ba9*. Následně jsem zjistil, že mám k dispozici SHA1 hash uživatelského jména. Vyzkoušel jsem různé webové stránky obsahující původní řetězce spojené s jejich SHA1 tvary, kde jsem tento hash zadával, a konečně jsem na jedné ze stránek[3] našel odpověď a zjistil, že uživatelské jméno je *user26496*. Po zadání těchto údajů do stránky jsem získal **Tajemství 1**.

3.2 myprog

Ve složce myprog jsem našel program s totožným názvem. Program jsem dekompiloval s využitím nástroje **Ghidra**[4], který z assembly kódu vytvořil relativně čitelný kód v jayzce C. Na základě toho jsem zjistil heslo, které program požaduje (*3a1792027e*), to jsem mu zadal a získal jsem **Tajemství x**.

3.3 library

Ve složce library jsem našel program *secret_application* a k němu přiloženou knihovnu *libfoo.h*, hlavičkový soubor *foo.h* a textový soubor *odposlech*. Textový soubor obsahoval kus funkce main a já tedy opět s využitím nástroje *Ghidra* program secret_application dekompiloval. Kód byl složitější než v předchozím myprog programu, takže jsem jej, namísto snahy ho nějak interpretovat sám, nakopíroval do nového souboru, natypoval proměnné, které Ghidra nedokázala určit, mírně upravil a s překladačem *gcc* přeložil. Výsledkem byla ovšem hláška "Stack-Smashing detected"a jelikož se výstup v podobě tajemství vždy mírně lišil, já musel zvolit jiný přístup. Na základě podrobnější analýzy kódu a nápovědy v podobě odposlechu jsem si uvědomil, že mi stačí vytvořit vlastní knihovnu, která bude obsahovat funkci *secret_function* takovou, co vrací číslo *123* a po této implementaci jsem již úspěšně získal správnou verzi **Tajemství 3**.

3.4 korespondence

Po nalezení předchozích 3 tajemství jsem pokračoval v průzkumu serveru a narazil jsem v kořenovém adresáři na podezřele vypadající složku *prace*. V té se

[2] `https://en.wikipedia.org/wiki/Obfuscation_(software)`
[3] `https://hashes.com/en/decrypt/hash`
[4] `https://github.com/NationalSecurityAgency/ghidra/`

nacházel privátní klíč a složka mail, která obsahovala textový soubor *korespondence*. Ten obsahoval spousty e-mailů, mezi kterými něco hledat by bylo složité, a já proto využil **grep** a s příkazem *cat korespondence | grep "ajemstv"* jsem získal **Tajemství h**.

4 192.168.122.149 (PostgreSQL)

V rámci předchozího serveru, domovské stanice, byl ve domovském adresáři ve složce *.ssh* uložen soubor *config* a ssh klíč. Config obsahoval adresu a uživatelské jméno (pepa), kterému patří zmíněný ssh klíč. Připojil jsem se na něj tedy a tím jsem se dostal na postgresqlserver. Už z názvu bylo patrné, že nejspíš zde běží PostgreSQL databáze, což potvrdil i nmap.

Ve snaze se do databáze připojit jsem z chybové hlášky zjistil, že je používán *peer authentication*, který, co jsem pochopil, říká, že se musí shodovat uživatelské jméno profilu s uživatelským jménem v databázi, což zjevně účet *pepa* nebyl. Šel jsem o složku výše a zjistil jsem, že v domovských složkách je i složka uživatele *database_user*. Přepnul jsem se na něj tedy příkazem *su database_user* (uživatel neměl heslo) a otevřel jsem si interaktivní PostgreSQL konzoli příkazem **psql**.

Ve výchozí databázi patřící uživateli database_user jsem našel tabulku *secret_advice*, která říkala, ať zkusím superuživatele. V databázi byly uživatelé pouze 2 – *database_user* a *postgres*. Zkusil jsem se tedy přepnout na databázi *postgres*, což fungovalo, neboť uživatel *database_user* má do databáze postgres přístup, přestože by asi mít neměl. To může být způsobeno špatně nastavenými právy a nebo výsledkem práce jiného studenta. V této databázi jsem našel tabulku *secret_table* a jednoduchým SQL příkazem *select * from secret_table;* jsem si zobrazil její obsah a tím získal **Tajemství w**.

5 192.168.122.235 – server2

Při hledání dalších tajemství jsem procházel již objevená ve snaze najít něco nového. Na domovské stanici ve složce *korespondence* jsem našel privátní klíč, jak bylo popsáno již v kapitole 3.4, a v totožné složce jsem objevil s pomocí výpisu příkazem **ls -a** skrytý soubor **.new_message**, který byl podepsán někým jménem *joe*. Privátní klíč byl vytvořen algoritmem RSA a jeho obsah je zakódovaný v base64. Tento klíč jsem tedy dekódoval a zjistil, že na konci klíče byl řetězec *joe@fedora*, což potvrdilo má podezření, že *joe* může být uživatelským jménem.

Nebyl jsem si ovšem jistý, na jakou stanici se s tímto klíčem připojit, neboť *fedora* je název domovské stanice a já tedy vyzkoušel postupně přihlášení na všechny, s uživatelským jménem *joe* a využitím zmíněného privátního klíče. Všechny stanice vyžadovaly heslo, až u stanice **192.168.122.235** namísto žádosti o heslo jsem dostal chybovou hlášku o špatně zabezpečeném klíči. Jeho zabezpečení jsem tedy změnil využitím příkazu **chmod 600 idrsa.key** a na stanici se úspěšně připojil.

Po připojení se v terminálu objevil textový řetězec jako zpráva dne (motd, nachází se v /etc/motd), který byl vzdáleně podobný textovým řetězcům, které se objevily například ve zmíněném programu *myprog* po zadání špatného hesla a já tedy začal přemýšlet, jestli se nemůže taktéž jednat o něco společného s tajemstvím. Po tom, co jsem žádnou indicii vedoucí k odhalení nenašel, jsem vyzkoušel online stránku na dešifrování substitučních šifer[5]. Jako první jsem vyzkoušel klasickou Caesarovu šifru a po spuštění mi nástroj na dané stránce odhalil, že s posunutím o 5 znaků dozadu se začátek řetězce *yfojrxy* změní na *tajemst*, ale zbytek řetězce stále nevypadá správně.

Na základě této znalosti jsem si vytvořil vlastní jednoduchý Python skript, který snížil ASCII hodnotu všech znaků v řetězci o 5 a tím jsem získal **Tajemství w** (přestože už jedno s tímto písmenem mám, toto mělo odlišný zbytek řetězce).

[5] `https://www.dcode.fr/tools-list#substitution_cipher`