Úlohy na precvičenie – OPS 2019 – séria F

Úlohy riešte samostatne a podrobne. Celý postup zaznamenajte a komentujte. Odpovedajte podľa možnosti na všetky položené otázky v úlohe. V záhlaví uveďte svoje meno, priezvisko a zdroje, ktoré ste pri riešení použili (citácie, URL adresy internetových zdrojov a mená osôb, s ktorými ste riešenie prípadne konzultovali). Za každé správne a vyčerpávajúce riešenie (s komentovaným postupom) možno získať bod (ak nie je uvedené inak). Zlomky bodov možno získať aj za čiastočné riešenia. Riešenia odovzdajte do 16. 4. 2019, 14:25 (do začiatku prednášky). Riešenia pred termínom môžete odovzdať na sekretariáte Ústavu informatiky do môjho priečinka. Neskôr dodané a opisované riešenia nebudú opravované ani hodnotené.

1. Predpokladajme, že pri nadväzovaní TCP spojenia použijeme len dvojfázové potriasanie rúk (two-way handshake) - teda po prvých dvoch SYN segmentoch považujeme spojenie za nadviazané. Uvážte možnosti uviaznutia (deadlocku) v tomto prípade. Uveďte príklad na uviaznutie alebo dokážte, že nenastane. Čo sa stane, ak spojenie začnú nadväzovať súčasne obidve strany ?

2. Uvedenie príznaku RST v záhlaví TCP segmentu spôsobí okamžité ukončenie relácie. Predpokladajme, že útočník pozná adresy účastníkov relácie a pozná aj čísla portov, na ktorých relácia prebieha. S akou pravdepodobnosťou sa mu podarí prerušiť spojenie, keď pošle RST segment niektorému účastníkovi? (uvážte, že to závisí hlavne od voľby potvrdzovacieho čísla – kvôli týmto útokom musí byť stále ACK príznak zapnutý a ak potvrdzovacie číslo nie je v rozsahu aktuálneho okna, segment sa považuje za chybný). Uvažujte situáciu s maximálne otvorenými posuvnými oknami (65535 oktetov) a situáciu s oknami otvorenými na 1500 oktetov.

3. TCP SYN flood attack – záplava SYN segmentami TCP spojenia je spôsob útoku, v ktorom útočník začína naraz veľa spojení (s príznakom SYN), pričom na odpovede servera (s príznakmi SYN a ACK) už neodpovedá. Štandardné riešenie predpokladá, že server pre každú výzvu SYN vytvorí samostatný port pre spojenie a alokuje pamäť pre posuvné okno podľa požiadavky klienta. Potom vyšle odpoveď (SYN/ACK) a čaká na potvrdenie (správu od klienta s príznakom ACK). Alokácia portu, pamäte a následné čakanie až do time-outu spôsobí na strane servera pri veľkej záťaži nedostatok zdrojov (pamäť, voľné porty). Ako by sa dal riešiť tento problém na strane servera bez alokácie zdrojov (pamäť a port by sa alokovali až po potvrdení ACK od klienta) ? Navrhnite konkrétne úpravy algoritmu, prípadne protokolu.

4. (2 body) Odsledujte Wiresharkom priebeh SSH spojenia (napr. PuTTY k GPU serveru 158.197.31.56). Odfiltrujte príslušnú TCP reláciu a dekódujte ako SSH. Ktoré algoritmy podporuje klient a ktoré server ? Nájdite miesto, kde sa generuje hlavný kľúč – vypíšte parametre vybratého algoritmu a spôsob autentifikácie servera. Nájdite miesto, od ktorého je komunikácia šifrovaná.