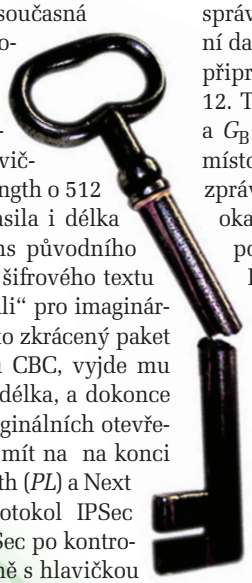
Průlom do VPS a protokolu IPSec (2)

V návaznosti na první díl článku v minulém čísle ST pokračujeme ukázkou konkrétního útoku na protokol IPSec. Využíváme přitom odkazy na obrázky uvedené v minulém článku.

Uvědomme si, že protokol IP kontroluje mnohem více dat v IP-hlavičce, než bylo uvedeno v příkladu v minulém čísle ST. Vytváří se tak další postranní kanály odhalující zašifrovaná data. Některé kanály lze použít proti 64bitové blokové šifře (TripleDES), jiné proti 128bitové blokové šifře (AES). Ukážeme si nyní využití chybové zprávy ICMP typu 12 „parameter problem“. Příjemce ji vydává, pokud zjistí, že pole Options v IP-hlavičce (*obr. 2*) je špatně formátované, a my ji na šifrovaném kanálu poznáme opět podle typické délky.

Zvolme libovolný zašifrovaný blok C, který prošel komunikačním kanálem (viz *obr. 1*). Ukážeme jak jej dešifrovat, tj. jak zjistit hodnotu $D(C)$. Takto pak odšifrujeme všechny bloky, co nás zajímají. Pro tento případ uvažujme 64bitovou blokovou šifru (pro 128 bitů platí totéž), tj. $D(C)$ má 8 bajtů. Luštění probíhá ve dvou krocích:

V ***prvním kroku*** si z dat přenášených z G_A do G_B vezmeme libovolný dostatečně dlouhý paket (asi 600–1023 bajtů). Chceme, aby jeho datová část obsahovala jeden celý blok 512 bajtů (512–1023 bajtů). Označme jeho inicializační vektor a šifrované bloky jako IV, C_0, C_1, ..., C_r. Nejprve z něho vyrobíme umělý zašifrovaný paket X, který po odšifrování v G_B způsobí, že IP-protokol ohlásí chybovou zprávu (č. 12). (Pokud nám věříte, můžete přejít rovnou na druhý krok.) Zdánlivě nesmyslný paket X pak využijeme v druhém kroku. Postup vytvoření paketu X je tento: V inicializačním vektoru IV změníme 6. bit. To znamená, že při dešifrování v modu CBC podle vzorce $P=IV$ xor $D(C_0)$ docílíme v prvním bloku otevřeného textu (P) rovněž změnu v bitu č. 6. Protože v tomto bitu bloku P se nachází hodnota IHL (a původní hodnota byla téměř jistě IHL=5, což je normální délka hlavičky IP v 32bitových slovech), způsobíme tím, že P bude mít hodnotu IHL=7. Teď si ale protokol IP bude myslet, že hlavička je o dvě 32bitová slova, čili o osm bajtů, delší. Do položky Options pak IP-parser zahrne dalších osm bajtů z originální datové části paketu. Hlavička paketu je však chráněna aritmetickým kontrolním součtem „Header Checksum“, který by byl narušen. Nicméně pokud změníme také bit č. 22 ve IV, změníme podobným způsobem i odpovídající bit v položce „Total Length (TL)“, která znamená délku datové části IP-paketu. Tento bit (je to devátý bit v bitovém vyjádření délky dat) měl původně hodnotu 1, protože délka dat byla zvolena mezi 512 a 1023, a tak devátý bit je vždy 1. Ten se teď změní na 0. Protože oba bity, číslo 6 a 22, mají v aritmetickém součtu (Header Checksum) stejnou váhu, jejich současná změna z 0 na 1 a z 1 na 0 ponechá součet korektní. Docílili jsme tedy změny v IHL a TL, aniž bychom narušili kontrolní součet hlavičky. Nyní ale máme Total Length o 512 menší, a proto, aby souhlasila i délka dat, stačí za polem Options původního paketu vynechat 504 bajtů šifrového textu (osm bajtů jsme už „odebrali“ pro imaginární pole Options). Až se tento zkrácený paket X bude dešifrovat v modu CBC, vyjde mu správná hlavička, správná délka, a dokonce i na konci bude několik originálních otevřených bloků, tj. paket bude mít na na konci správný Padding, Pad Length (PL) a Next Header (NH), čili pro protokol IPSec vznikne korektní paket. IPSec po kontrole těchto položek je společně s hlavičkou ESP odstraní a předá odšifrovaný vnitřní paket protokolu IP. Tento paket však už nebude korektní pro IP-protokol. Bude mít posunutou hranici pole Options a za ním zkrácenou datovou část. Protokol IP nyní zjistí, že (uměle natažené) pole Options nemá požadovaný formát (pro formát platí přísná pravidla, která budou narušena), a proto vydá chybovou zprávu číslo 12. Tím jsme jednoduchou modifikací šifrového textu (změna dvou bitů IV a vyškrtnutí 504 bajtů) obdrželi zašifrovaný paket X, který projde bez kolize kontrolou IPSec a až protokol IP vydá chybovou zprávu 12.



V následujícím ***druhém kroku*** vezmeme náš zájmový blok C a vyrobíme šifrovaný paket X', který vznikne z X, když poslední jeho dva bloky (..., C_{r-1}, C_r) nahradíme bloky (..., R, C). R je náhodný blok, u něhož systematicky měníme jeho poslední dva bajty. Je to 65 536 pokusů, v průměru postačí polovina. Pakety X' odesíláme bráně G_B (to jsou ony chytré dotazy útočníka protokolu IP přijímající strany). Co se nyní stane? IPSec odšifruje X'. Z modu CBC vyplývá, že poslední odšifrovaný blok bude roven hodnotě R xor $D(C)$. IPSec očekává, že v posledních jeho dvou bajtech (PL, NH) budou korektní hodnoty délky doplňku (PL) a čísla protokolu (NH). NH musí být rovno 4, neboť odšifrovaný paket se předává protokolu IP, čemuž odpovídá právě NH=4. Pokud je R náhodný blok, u kterého měníme systematicky poslední dva bajty, hodnota dešifrovaného posledního bloku (rovná se R xor $D(C)$) nutně vystřídá na posledních dvou bajtech všechny možné hodnoty, tedy určitě i tu správnou (PL, NH). Běžná reakce bude, že hodnoty PL a NH nebudou správné. V tomhle případě IPSec prostě paket zahodí a nikomu nic neoznamuje, čili na kanálu bude klid. Jednou však nastane situace, že PL a NH budou správné, a proto tentokrát G_B poskytne vnitřní data protokolu IP. Avšak my jsme paket X' připravili tak, že protokol IP ohlásí chybu 12. Tuto situaci útočník na kanálu mezi G_A a G_B jednoduše zaregistruje podle toho, že místo obvyklého klidu G_B odesílá ICMP zprávu 12 (má příslušnou délku). V tom okamžiku (díky modu CBC) ví, že hodnota posledního bajtu výrazu R xor $D(C)$ je 4.

Protože hodnotu posledního bajtu R sám volí, snadno dopočte hodnotu posledního bajtu $D(C)$. Suma sumárum jsme slavnou VPN přelstili, neboť jsme IP-protokol donutili, aby nám sdělil otevřenou hodnotu posledního bajtu z jakéhokoliv zašifrovaného bloku C.

Podobným způsobem lze s využitím kontrol Paddingu a Pad Length s lineárně narůstající složitostí zjistit i ostatní bajty zašifrovaného bloku C. Celý blok C lze takto odšifrovat v průměru během 33 144 zkoušek (experimentálně ověřeno), což trvá desítky sekund. Analogicky lze využít další chybové zprávy a další kontroly položek hlavičky IP-protokolu. Poznamenejme, že náš (zde uvedený útok) pomocí IHL a Total Length je 32 768krát efektivnější než útok popsaný na *http://eprint.iacr.org/2007/125.pdf*. Podobně mohou být zlepšovány i útoky s využitím jiných zpráv ICMP. Jednou z možných obran je filtrovat zpětnou komunikaci z G_B a využívané pakety ICMP zakázat. To je dnes nezávisle na popsaném útoku běžná praxe v situacích, kdy zařízení provádějící tunelování je zároveň také firewall. Z hlediska kryptologie však takové opatření rozhodně nelze považovat za kvalitní. Velmi rázně je potřeba zabránit jakékoliv manipulaci se šifrovým textem, kterou jsme právě viděli, a to nakonfigurováním příslušných zařízení tak, aby prováděla šifrování současně s autentizací dat. Popsané útoky postranními kanály na protokol IPSec, používaný v mnoha zařízeních a virtuálních privátních sítích, jsou velmi nebezpečné, protože útočí na šifrátory (konfigurované bez autentizace), které jsou implementovány přesně podle platných standardů IPSec (a to i minulých i současných). Je pochopitelné, že přední světoví výrobci se těchto norem drželi a implementovali je. Tyto standardy je však třeba opravit a šifrátory překonfigurovat. Účinnou obranou je překonfigurovat tato zařízení a protokoly IPSec tak, aby se šifrováním byla vždy současně použita autentizace. **Šifrování bez autentizace budiž navěky zapovězeno.**

Vlastimil Klíma, Tomáš Rosa,
v.klima@volny.cz, trosa@ebanka.cz