

Hardwarové bezpečnostní moduly – API a útoky

Masarykova univerzita v Brně
Fakulta informatiky

Jan Krhovják
Daniel Cvrček
Vašek Matyáš

Shrnutí

- Úvod
 - Základní terminologie
 - Architektura HSM (Hardware Security Module)
- Bezpečnostní požadavky na kryptografické moduly
 - FIPS 140-1 a FIPS 140-2
- Útoky na a přes API
 - Úvod do problematiky API pro HSM
 - Klíče a jejich integrita
 - Nedostatečná kontrola parametrů funkcí
 - Nevynucení politiky – PKCS #11
- Závěr

Úvod

- Základní terminologie
 - Hardwarová bezpečnostní zařízení
 - Hostitelská zařízení
 - Útok
- Architektura HSM
 - Vychází z klasické von Neumannovy architektury
 - + Mechanizmy fyzické ochrany
 - + Generátory náhodných čísel
 - + Speciální koprocесory
 - + NVRAM
 - V/V obvody
 - IBM 4758 (viz obrázek)



Bezpečnostní požadavky na HSM: FIPS 140-1 a FIPS 140-2

- Norma definuje 4 úrovně zabezpečení
 - Úroveň 1 – není vyžadována fyzická ochrana
 - Příkladem jsou osobní počítače
 - Úroveň 2 – vyžaduje zajištění evidence průniků
 - Autentizace založená na rolích
 - OS uvnitř HSM musí splňovat alespoň EAL2 podle CC
 - Příkladem jsou čipové karty
 - Úroveň 3 – již ne pouze pasivní fyzická ochrana
 - V případě detekce průniku musí být citlivá data vymazána
 - Autentizace založená na identitách
 - OS uvnitř HSM musí splňovat alespoň EAL3 podle CC
 - Příkladem zařízení je Luna CA
 - Úroveň 4 – dodržování vnějších provozních podmínek modulu
 - OS uvnitř HSM musí splňovat alespoň EAL4 podle CC
 - Příkladem zařízení je IBM 4758

Oblasti bezpečnostních požadavků

- Celkem 11 oblastí – například:
 - Dokumentace kryptografického modulu
 - Porty a rozhraní
 - Role, služby a autentizace
 - Fyzická bezpečnost
 - Operační prostředí
 - Správa klíčů
 - Zmírnění jiných útoků
- V čem se liší FIPS 140-2 od FIPS 140-1
 - Jinak strukturována + upřesněna a sjednocena terminologie
 - Přidána nová část týkající se zmírnění jiných útoků
 - Zesílení požadavků na autentizační mechanismy a na testování modulu atd.

Útoky na a přes API

- API (Application Programming Interface)
 - Jediné komunikační rozhraní mezi HSM a vnější aplikací
 - Na základě funkcí API jsou budovány protokoly
 - API HSM obsahuje stovky funkcí s mnoha parametry, což poskytuje velmi velký prostor k chybám a vzniku útoků
- Příklady běžně používaných API
 - Common Cryptographic Architecture (CCA)
 - Public Key Cryptographic Standard (PKCS) #11
- Útoky rozdělené podle typu chyb, kterých využívají
 - Klíče a jejich integrita
 - Nedostatečná kontrola parametrů funkcí
 - Nevynucení politiky – PKCS #11

Klíče a jejich integrita – Meet in the Middle Attack I

- Zneužívá
 - Malá velikost šifrovacích klíčů algoritmu DES
 - Špatný návrh rozlišování typů klíčů
 - Neexistenci omezení na generování klíčů
- Myšlenka útoku
 - Mnoho HSM dokáže generovat desetitisíce klíčů během minut
 - Útočník pak vygeneruje např. 2^{16} klíčů a jimi zašifruje stejný testovací vzorek, který si uloží
 - Poté systematicky prohledává klíčový prostor a stejný testovací vzorek každým klíčem zašifruje a porovná se všemi uloženými vzorky
 - Dojde-li ke shodě, tak nalezl hodnotu jednoho tajného klíče
- Výpočetní složitost nalezení jednoho klíče tak klesne z 2^{55} na 2^{39}

Klíče a jejich integrita – Meet in the Middle Attack II

- Je-li nalezený klíč určen k šifrování dalších klíčů – tzv. terminální klíč, lze jeho pomocí přešifrovat veškerá jinými terminálními klíči chráněná data a klíče
 - Tímto způsobem lze kompromitovat osm z devíti typů klíčů, které používá Visa Security Module
- Variantu útoku lze aplikovat i na kryptografický modul Prism TSM 200
 - Při vynaložení stejného úsilí jako v předchozím případě získat dokonce hlavní klíč celého zařízení!

Klíče a jejich integrita – Conjuring Keys From Nowhere

- Jedná se o neautorizované generování klíčů ukládaných mimo HSM
 - Útočník nejprve náhodně vytvoří hodnotu zašifrovaného klíče a podstrčí jej HSM
 - Při dešifrování je hodnota klíče také náhodná a v případě DES má s pravděpodobností $1/2^8$ správnou paritu
 - V případě dvou-klíčového 3DES-2 má správnou paritu s pravděpodobností $1/2^{16}$, což je stále dosažitelné
- Takto vložené klíče mohou posloužit k vytvoření dalších a složitějších útoků
- Obrana spočívá v pečlivějším návrhu formátu klíčů, který bude obsahovat větší množství entropie

Klíče a jejich integrita – 3DES Key Binding Attack I

- Zneužívá
 - Nedostatečnou vazbu jednotlivých částí 3DES-2 klíčů
- Myšlenka útoku
 - Útočník nejprve vygeneruje velké množství klíčů se stejnými polovinami a stejného typu jako požadovaný klíč
 - Pomocí *Meet in the Middle* útoku nalezne hodnoty dvou z těchto klíčů (prohledávání 2^{41} možností)
 - Výměnou jejich polovin vytvoří dva 3DES-2 klíče s odlišnými polovinami
 - Je-li požadovaným (tj. i nalezeným) klíčem exportní klíč, může nyní s jeho pomocí exportovat a poté i dešifrovat všechny klíče v HSM určené k exportu

Klíče a jejich integrita – 3DES Key Binding Attack II

- Získat však lze i klíč, který nemá povolen export
 - Útočník nejprve zamění jednu jeho polovinu s polovinou známého klíče
 - Tím vzniknou dva klíče, jejichž jedna polovina je známá a druhou polovinu získáme prohledáváním prostoru 2^{56} (hledáme oba klíče současně)
- To je již ale práce pro speciální hardware či distribuované systémy

Nedostatečná kontrola parametrů funkcí – Decimalisation Table Attacks I

- Techniky generování a verifikace PINů
 - Metody IBM 3624 a IBM 3624 Offset
 - Generování je založeno na validačních datech (např. PAN)
 - Ta jsou dále zašifrována PIN generujícím klíčem a poté je požadovaný počet číslic převeden do desítkové soustavy (decimalizován) a zvolen jako PIN
 - Generování u IBM 3624 Offset probíhá stejným způsobem, ale výsledek se nazývá IPIN (Intermediate PIN) a offset je získán odečtením IPINu od zvoleného PINu (po cifrách modulo 10)
 - Verifikace probíhá podobně a vypočítaný PIN se nakonec porovná s PINem získaným z příslušného zašifrovaného PIN-bloku (EPB), což je 8bajtová struktura, do níž je PIN před zašifrováním formátován

Nedostatečná kontrola parametrů funkcí – Decimalisation Table Attacks II

- Příklad verifikace PINu metodou IBM 3624 Offset
 - PIN odeslaný bankomatem v EPB je 7816 (zná jen vlastník)
 - Veřejně přístupný offset (typicky uložen na kartě) je 4344
 - Číslo účtu zákazníka (PAN) je 4556 2385 7753 2239
 - Co se děje uvnitř verifikační funkce?
 - PAN je zašifrován na např. 3F7C 2201 00CA 8AB3
 - Zkrácení na požadovaný počet číslic (typicky 4) tj. na 3F7C
 - Převedení do desítkové soustavy použitím decimalizační tabulky 0123 4567 8901 2345 tj. z 3F7C na 3572
 - Přičtení offsetu 4344 tj. vygenerovaný PIN je nyní 7816
 - Porovnání tohoto PINu s PINem extrahovaným z EPB
 - OK 😊

Nedostatečná kontrola parametrů funkcí – Decimalisation Table Attacks III

- Útoky využívající známých zašifrovaných PINů
 - Předpokládejme použití čtyřmístných PINů a offsetu 0000
 - Nastavíme-li decimalizační tabulku (DT) na samé nuly, vždy se vygeneruje PIN roven čtyřem nulám
 - Tímto trikem lze získat EPB obsahující PIN 0000
 - Necht' $D_{orig} = 0123\ 4567\ 8901\ 2345$ je korektní DT
 - D_i jsou nové binární DT, které mají jedničku na těch pozicích, kde D_{orig} měla i tj. např. $D_5 = 0000\ 0100\ 0000\ 0001$
 - Nyní již útočník pouze pro $i = 0$ až 9 volá verifikační funkci s EPB nulového PINu a s DT D_i
 - Není-li v hledaném PINu číslice i obsažena, změna v DT se neprojeví a verifikace proběhne úspěšně

Nedostatečná kontrola parametrů funkcí – Decimalisation Table Attacks IV

- Tím útočník zjistí číslice obsažené v zákaznickově PINu a prohledávaný prostor PINů tak omezí z 10 000 na nejvýše 36
- Útok bez známého zašifrovaného PINu
 - Předpokládejme, že se nám podařilo zachytit zákazníkuv EPB obsahující správný PIN, a že offset je stále 0000
 - D_i jsou nové DT, které mají hodnotu $i-1$ na těch pozicích, kde D_{orig} měla i tj. např. $D_5 = 0123 \ 4\mathbf{4}67 \ 8901 \ 234\mathbf{4}$
 - Nyní stačí, aby útočník pro každou číslici i zavolal verifikační funkci se zachyceným EPB, správným offsetem (tj. 0000) a DT D_i
 - Tím opět zjistí číslice obsažené v zákaznickově PINu
 - Jejich pořadí pak dokáže určit manipulací s offsety

Nedostatečná kontrola parametrů funkcí – Decimalisation Table Attacks V

- Příklad na PINu, který má všechny 4 číslice odlišné
 - Řekněme, že PIN v EPB je 3621
 - Pokusme se určit pozici číslice 1
 - Použitím D_1 lze hodnota generovaného PINu změnit na 3620
 - Verifikace však nyní neproběhne úspěšně
 - Postupným voláním s offsety 1000, 0100, 0010, 0001 lze však naopak jednotlivé číslice generovaného PINu zvyšovat
 - V případě offsetu 0001 se jeho hodnota vrátí zpět na 3621
 - Nyní verifikace proběhne úspěšně a útočník podle použitého offsetu ví, že číslice 1 je v PINu až na čtvrté pozici
 - K určení pozic všech číslic čtyřmístného PINu stačí nejvýše 6 volání verifikační funkce

Nevynucení politiky u PKCS #11

- Doposud jsme se zabývali útoky na API navržené přímo pro konkrétní HSM
- Nyní se zaměříme na PKCS #11 API
 - Navrženo pouze jako standardní rozhraní mezi aplikacemi a jednouuživatelskými bezpečnostními zařízeními
- Hlavní problém tohoto API
 - Jedná se pouze o sadu funkcí bez jakékoliv politiky
 - Ta by například zajistila konzistentnost vlastností klíčů

Nevynucení politiky u PKCS #11 – Symmetric Key Attacks I

- 3DES Key Binding Attack
 - 3DES-2 klíč K a jeho jednotlivé poloviny K_1 a K_2
 - Při exportu $E_{KEK}(K)=(E_{KEK}(K_1), E_{KEK}(K_2))$
- Key Separation Attack
 - Konfliktní nastavení vlastností klíčů
 - Například klíč určený k šifrování klíčů a dešifrování dat
- Weaker Key/Algorithm Attack
 - Šifrování klíčů pomocí slabých algoritmů jako RC2 či DES
- Related Key Attack
 - 3DES-3 klíč $K_1=(K_A, K_B, K_C)$ a klíč $K_2=(K_A \oplus \text{DELTA}, K_B, K_C)$
 - $P' = D_{K_A \oplus \text{DELTA}} (E_{K_B} (D_{K_C} (E_{K_C} (D_{K_B} (E_{K_A} (P)))))) = D_{K_A \oplus \text{DELTA}} (E_{K_A} (P))$

Nevynucení politiky u PKCS #11 – Symmetric Key Attacks II

- **Reduced Key Space Attack**
 - Jedna z funkcí PKCS #11 (`C_DeriveKey`) umožňuje vytvořit klíč z části po sobě jdoucích bitů již existujícího klíče
 - Toho lze využít ke zmenšení prohledávaného prostoru klíčů
 - Útočník například nejprve použitím čtyřiceti po sobě jdoucích bitů z 56bitového DES klíče vytvoří 40bitový RC2 klíč
 - Ten pak hrubou silou dešifruje a s jeho pomocí najde i zbylých 16 bitů původního DES klíče

Nevynucení politiky u PKCS #11 – Public Key API Attacks

- Výčet útoků, které se opírají o podporu API pro kryptografické operace s veřejným klíčem
 - Small Public Exponent with No Padding Attack
 - Trojan Public Key Attack
 - Trojan Wrapped Key Attack
 - Private Key Modification Attack
- Množství relativně snadných útoků na PKCS #11
 - PKCS #11 API není skutečně vhodné pro použití ve vysoce fyzicky zabezpečených HSM

Závěr

- Bezpečnost současné generace API není dostačující
 - Důkazem toho je právě množství nově objevených útoků
 - Tyto útoky samy o sobě neznamenaají, že by HSM byly k ničemu
 - Krátkodobě se však musí posílit administrativní kontrola přístupu k modulům
 - Dlouhodobě se musí odstranit přímo chyby způsobující zranitelnost modulů
- Mnoho dalších útoků je zpracováno v dokumentu, který vychází z diplomové práce prvního z autorů a je dostupný na:

http://www.fi.muni.cz/~xkrhovj/apinf/sdipr/DP_upravena_v1.pdf.

