Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Aplikačná bezpečnosť

Erik Saller, Ivan Oravec

KPMG
cutting through complexity™

Osnova

- Úvod
- Základné bezpečnostné hrozby pre aplikácie
- Aplikačné bezpečnostné funkcie
- Typické zraniteľnosti aplikácií a opatrenia proti nim
- Používanie otvorených štandardov
- Záver

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

ÚVOD

KPMG
cutting through complexity™

Úvod

- Aplikačná bezpečnosť
  - zahŕňa opatrenia prijaté počas celého životného cyklu aplikácie, aby sa zabránilo výnimkám v bezpečnostnej politike aplikácie alebo systému (zraniteľnosti) na ktorom aplikácia beží, spôsobeným **chybami v návrhu, vývoji, nasadení, aktualizácii, alebo údržbe aplikácie**.
- Primárnym cieľom aplikačnej bezpečnosti:
  - je tvorba aplikácií, ktoré **neobsahujú** rôzne bezpečnostné nedostatky, ktoré môžu byť zneužité útočníkom na kompromitáciu údajov spracovávaných aplikáciou, kompromitáciu samotnej aplikácie alebo infraštruktúry na ktorej je aplikácia prevádzkovaná.

Úvod (pokr.)

- Kompromitácia informačných aktív je porušenie integrity, dôvernosti alebo dostupnosti informačných aktív.
- Aplikačná bezpečnosť má rôzne aspekty, ktorými je potrebné sa zaoberať ak chceme pochopiť:
  - aké riziká hrozia pri používaní zraniteľných aplikácií
  - ktoré sú typické zraniteľnosti aplikácií
  - ako vznikajú a ako sa im úspešne vyhýbať.

Úvod (pokr.)

- Aplikácia má vykonávať to, na čo bola navrhnutá ....

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Úvod (pokr.)

.... ale iba to ☺

KPMG
cutting through complexity™

Úvod (pokr.)

- Dôležité aspekty aplikačnej bezpečnosti zahŕňajú:
  - Riziká predstavované prevádzkovaním zraniteľných aplikácií
  - Typické hrozby pre bezpečnosť aplikácií
  - Typické zraniteľnosti aplikácií
  - Návrh a vývoj bezpečného softvéru

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Riziká prevádzkovania zraniteľných aplikácií

- Únik údajov
- Manipulácia údajov
- Znefunkčnenie aplikácie (DoS)
- Manipulácia logiky aplikácie a s ňou spojených procesov (business logic)
- Rozšírenie útoku na ďalšie IKT
- Prechod útočníka cez sieťový perimeter

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

ZÁKLADNÉ BEZPEČNOSTNÉ HROZBY PRE APLIKÁCIE

KPMG
cutting through complexity™

Základné bezpečnostné hrozby pre aplikácie

- Aplikácia a údaje, ktoré spracúva, môžu byť kompromitované množstvom rôznych metód a zneužitím rozmanitých bezpečnostných slabín.
- Ak dôjde k úspešnému útoku na aplikáciu, väčšinou je to spôsobené bezpečnostnou zraniteľnosťou v samotnom softvéri aplikácie. Nie je to však vždy tak.

Základné bezpečnostné hrozby pre aplikácie (pokr.)

- Zraniteľnosť ohrozujúca aplikáciu sa môže vyskytnúť na rôznych úrovniach, keďže chyby sa môžu dopustiť nielen vývojári ale aj administrátori, operátori alebo iní zamestnanci prevádzky.

Základné bezpečnostné hrozby pre aplikácie (pokr.)

- Ide o nasledovné úrovne:
  - Bezpečnostné chyby v softvéri
  - Konfiguračné chyby
  - Nedostatky v bezpečnosti prevádzky

Bezpečnostné chyby v softvéri

- Najčastejšie ohrozujú aplikácie bezpečnostné zraniteľnosti v kóde aplikácie.
- Výskyt bezpečnostných chýb v kóde môže byť spôsobený rôznymi faktormi ako:
  - nedostatočné vzdelávanie programátorov zo zásad bezpečného programovania,
  - chýbajúce štandardy pre bezpečný vývoj v konkrétnych technológiách,
  - nedostatok času,
  - zlý návrh,
  - chýbajúca analýza rizík a súvisiacich hrozieb
  - nedostatočná testovacie metodológia.

Bezpečnostné chyby v softvéri (pokr.)

- Toto sú však väčšinou už len konkrétne prejavy nedostatočného dôrazu na bezpečnosť vo vývoji softvérového produktu a s tým súvisiacej absencie iniciatívy aplikačnej bezpečnosti riadenej alebo aspoň spravovanej (Governance) z najvyšších pozícií organizácie.
- Softvérové zraniteľnosti majúce bezpečnostný dopad môžu mať rôzny typ a formu,
  - môžu sa vyskytnúť na rôznych úrovniach technologických celkov.
  - Niektoré typy zraniteľností sú aktuálne pre všetky alebo aspoň pomerne veľa platforiem,
  - niektoré sú veľmi špecifické a platné len pre jednu špecifickú technológiu.

Bezpečnostné chyby v softvéri (pokr.)

- Medzi časté zraniteľnosti aplikácií patria:
  - Nedostatočná validácia vstupov
  - Možnosť vkladania kódu (Code Injection, injekcia kódu)
  - Možnosť vkladania neoprávnených SQL dotazov (SQL Injection)
  - Cross-site Scripting (XSS)
  - Pretečenie zásobníka
  - Race conditions
  - Nedostatky v nastavení prístupových práv

Konfiguračné chyby

- Ďalšou úrovňou kde sa zvyknú vyskytovať zraniteľnosti ohrozujúce aplikácie sú konfiguračné nastavenia.
- Konfiguračné nastavenia, či už v samotnej aplikácii, systéme na ktorom je prevádzkovaná, databáze ktorú aplikácia využíva alebo v inej súvisiacej komponente, môže bezpečnosť aplikácie vážne narušiť aj bez toho aby bol priamo v kóde aplikácie akýkoľvek bezpečnostný nedostatok.

Konfiguračné chyby (pokr.)

- Príkladom konfiguračnej chyby s ktorou sme sa v praxi veľakrát stretli, môže byť zdieľaný adresár obsahujúci stromovú štruktúru súborového systému aplikácie.
- Ak sú zdieľaný adresár, resp. súbory a adresáre v ňom, zapisovateľné útočníkom, bezpečnostné funkcie aplikácie neochránia aplikáciu pred kompromitáciou.

Nedostatky v bezpečnosti prevádzky

- Nedostatočné bezpečnostné praktiky pri prevádzke aplikácií sú ďalšou kategóriou chýb ktoré môžu aplikáciu ohroziť.
- Kód aplikácie, aj jej konfigurácia môže byť sto percentne bezpečná (v praxi nedosiahnuteľná méta), ale ak nie je prevádzkovaná bezpečným spôsobom, potenciálny útočník môže aplikáciu a jej údaje kompromitovať zneužitím bezpečnostných nedostatkov v prevádzke.
- Napríklad uhádnuteľné heslá, administrácia aplikácie z nezabezpečeného systému alebo nedostatočne chránené súbory aplikácie príp. zálohy, dokážu ohroziť aplikáciu často oveľa vážnejšie než zraniteľnosť v softvéri.

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

APLIKAČNÉ BEZPEČNOSTNÉ FUNKCIE

KPMG
cutting through complexity™

Používateľská prístupnosť aplikácie

- V princípe je možné ľubovoľnú aplikáciu (a vo všeobecnosti systém alebo IKT) extrémne zabezpečiť až natoľko, že pravdepodobnosť jej kompromitácie bude blízka nule.
- Následkom by okrem iného bolo:
  - Na zabezpečenie aplikácie by boli vynaložené neadekvátne vysoké prostriedky, ktorých hodnota by nebola v rozumnom pomere s hodnotou chránených aktív.
  - Aplikácia by bola v praxi nepoužiteľná, pretože implementované bezpečnostné opatrenia by enormne sťažili, prípadne úplne znemožnili jej používanie

Používateľská prístupnosť aplikácie (pokr.)

- Niektoré bezpečnostné funkcie (vrátane ich parametrov) sú pre používateľov transparentné, takže si ich prítomnosť nemusia ani uvedomovať. Iné bezpečnostné funkcie sú zase pre používateľov veľmi citeľné už zo svojej podstaty ako napríklad zopakovanie autentifikácie pri vypršaní relácie, kratšia životnosť relácie alebo re-autentifikácia pri volaní citlivej operácie.
- Používateľská prístupnosť aplikácie je dôležitým aspektom ktorý treba brať do úvahy už pri návrhu aplikácie, ale taktiež neskôr pri riešení implementačných detailov.

Vhodnosť bezpečnostnej funkcie

- Jednu a tú istú bezpečnostnú požiadavku vieme väčšinou na technickej úrovni implementovať rôznymi bezpečnostnými funkciami.
- Treba zvážiť plánovanú architektúru, platformy, prostredie, ...
- Jedným z dôležitých kritérií pri výbere bezpečnostných opatrení musí byť ich dopad na používateľskú prístupnosť aby bezpečnostné opatrenia neinterferovali s vykonávanými biznis aktivitami do tej miery, že sa stráca ich zmysel.

Náročnosť správy aplikácie

- Podobné úvahy o bezpečnostných opatreniach ako v prípade používateľskej prístupnosti, platia aj pre náročnosť správy aplikácie.
- Ak budú napr. bezpečnostná architektúra, implementované opatrenia, modularizácia, prístupový a dátový model aplikácie natoľko komplikované alebo neprehľadné, že celková administrácia a bezpečné prevádzkovanie aplikácie bude komplikované a časovo náročné, bude mať veľký počet a rozmanitosť opatrení na bezpečnosť aplikácie skôr opačný účinok než bolo pôvodne zamýšľané.

Prostredie aplikácie

- Rôzne bezpečnostné funkcie aplikácie a aj jej celková architektúra, sa už vo fáze návrhu opiera o množstvo predpokladov o prostredí v ktorom bude aplikácia nasadená.
- Neskoršou postupnou zmenou prostredia, v ktorom aplikácia beží alebo jej migráciou do nového prostredia môžu prestať platiť pôvodné bezpečnostné predpoklady o prostredí čo môže ohroziť efektívnosť existujúcich opatrení.

Aplikačné bezpečnostné funkcie

- Aplikácie pre svoje zabezpečenie používajú celý rad bezpečnostných funkcií, ktoré majú na starosti rôzne aspekty jej bezpečnosti.
- Existujú hotové riešenia, ktoré treba uprednostniť pre vyvíjaním vlastného kódu pre tento účel.
- Niektoré bezpečnostné funkcie je veľmi ťažké správne (bezpečne) navrhnúť a následne implementovať

Aplikačné bezpečnostné funkcie (pokr.)

- Preto ak skutočne nejde o funkčnosť, ktorá musí byť z nejakého dôvodu šitá na mieru vyvíjanej aplikácii, treba uprednostniť využívanie bezpečnostných funkcií a mechanizmov poskytovaných platformou, operačným systémom prípadne databázou aplikácie.

Aplikačné bezpečnostné funkcie

- V nasledujúcich podkapitolách sa pozrieme na nasledovné najčastejšie typy bezpečnostných funkcí:
  - Autentifikácia
  - Autorizácia
  - Správa relácie
  - Validácia vstupov
  - Spracovanie chýb
  - Vytváranie auditných záznamov
  - Kryptografia

Autentifikácia

- Autentifikačné bezpečnostné funkcie spájajú identitu reálneho používateľa s jeho systémovou identitou (účtom) pomocou overenia prihlasovacích údajov. Autentifikačné mechanizmy musia byť adekvátne rizikovosti aplikácie a aby dokázala odolávať útočníkom využívajúcim pri svojich útokoch rôzne metódy.
- Podrobné informácie o autentifikácii a spôsobom, ako používať autentifikačné mechanizmy v bežných aplikáciách sú poskytnuté vo štvrtej kapitole.

Autorizácia

- Autorizačné funkcie musia zaistiť, že legitímni používatelia systému smú vykonať len tie operácie a pristupovať k tým údajom pre ktoré sú autorizovaní, t.j. ktoré zodpovedajú ich oprávneniam. Riadia prístup ku chráneným zdrojom povolením alebo zamietaním prístupu na základe rolí alebo úrovne oprávnení.
- Podrobné informácie o autorizácii v bežných aplikáciách sú poskytnuté vo štvrtej kapitole.

Správa relácie

- V počítačových vedách a špeciálne v počítačových sieťach sa pod pojmom relácia (session) rozumie semi-permanentná interaktívna výmena informácií, tiež známa ako dialóg, konverzácia alebo stretnutie, medzi dvoma alebo viacerými komunikujúcimi zariadeniami, alebo medzi systémom a používateľom.
- Po prihlásení používateľa do aplikácie je používateľovi pridelený tzv. identifikátor relácie (session ID), ktorý slúži na identifikáciu používateľa pri jeho ďalšej interakcii s aplikáciou, až do jeho odhlásenia alebo vypršania životnosti relácie. Na strane aplikácie je s identifikátorom relácie spojená identita používateľa a stav jeho relácie.

Správa relácie (pokr.)

- Pre úspešný prienik na účet útočníkovi postačuje získanie, alebo uhádnutie identifikátora jeho relácie.
- Aby sa hodnoty platných identifikátorov relácie nedali uhádnuť, musia byť generované kryptograficky bezpečnými algoritmami.

Validácia vstupov

- Bezpečnostné funkcie pre validáciu vstupov majú zabezpečiť, že aplikácia je dostatočne odolná voči všetkým hodnotám vstupných údajov, či už tieto údaje pochádzajú od používateľa, infraštruktúry, externých entít alebo databázových systémov.
- Na nedostatkoch vo validácii vstupov je založené množstvo iných zraniteľností ako napríklad možnosť vkladania kódu, možnosť vkladania SQL príkazov, Cross-site scripting zraniteľnosti, pretečenie zásobníka, atď.

Spracovanie chýb

- Každá aplikácia by mala mať ošetrené všetky známe možné chybové stavy
- Identifikácia miest v kóde, kde môže nastať chybový stav
- Implementácia relevantných bezpečnostných funkcií, ktoré daný chybový stav vhodným a bezpečným spôsobom ošetria

Vytváranie auditných záznamov

- Ide o bezpečnostné funkcie, ktoré majú za úlohu vytváranie auditných záznamov, ktoré zachytávajú rôzne dôležité udalosti v aplikácii ako napr. prihlásenie a odhlásenie používateľa, zamietnutie prístupu k určitému zdroju a chybové stavy v aplikácii. Aplikácia by však mala poskytovať možnosť konfigurácie ako typov udalostí, ktoré sa budú zaznamenávať tak aj úrovne detailnosti informácie, ktorá sa bude ku každej udalosti zaznamenávať.
- Pokiaľ auditné záznamy nemajú formu jednoduchých textových súborov, ale majú napr. binárny formát - aplikácia by mala tiež poskytovať nástroj na prezeranie a prácu s jej auditnými súbormi.

Kryptografia

- Kryptografické bezpečnostné funkcie slúžia primárne na ochranu integrity a dôvernosti citlivých údajov (či už počas ich prenosu alebo pri uskladnení) sú však aj súčasťou rôznych bezpečnostných protokolov (napr. pri autentifikácii).

Kryptografia (pokr.)

- Medzi typické kryptografické nástroje a ich použitie v aplikáciách patria:
  - Generátory pseudo-náhodných čísel
    - *Generovanie dostatočne náhodných číselných hodnôt pre rôzne účely, kedy je vyžadované aby tieto čísla neboli útočníkom uhádnuteľné*
  - Hašovacie funkcie
    - *Kontrola integrity údajov*
    - *Utajenie prístupových hesiel pri zachovaní možnosti ich neskoršieho overenia oproti zadanému heslu*
  - Autentifikačné protokoly
    - *Overenie identity používateľa*
  - Šifrovacie algoritmy
    - *Ochrana dôvernosti a integrity uskladnených údajov*
    - *Ochrana dôvernosti a integrity prenášaných údajov*
  - Algoritmy pre elektronický podpis
    - *Overenie identity odosielateľa údajov*

Kryptografia (pokr.)

- Návrh dostatočne silných kryptografických algoritmov, protokolov a iných kryptografických nástrojov vyžaduje pomerné široké a hlboké matematické znalosti a tiež dlhšiu prax v danej oblasti.
- Ešte aj v takomto prípade je na potvrdenie ich bezpečnosti potrebná analýza odbornou verejnosťou.

Kryptografia (pokr.)

- Preto sa vo všeobecnosti neodporúča vývoj vlastných elementárnych kryptografických nástrojov ale použitie známych softvérových knižníc, ktoré implementujú bezpečné kryptografické mechanizmy.

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

TYPICKÉ ZRANITEĽNOSTI APLIKÁCIÍ A OPATRENIA PROTI NIM

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Typické zraniteľnosti aplikácií a opatrenia proti nim

Nedostatočná validácia vstupov:

- Najčastejšou bezpečnostnou slabinou aplikácií je neschopnosť správne overiť vstup od klienta alebo od prostredia.
- Špeciálne znaky
- Cieľom validácie vstupov je zabezpečenie toho, aby aplikácia bola schopná prijímať validné vstupné údaje a aby bola zároveň dostatočne odolná voči všetkým hodnotám vstupných dát, či už získaných od používateľa, infraštruktúry, externých subjektov alebo databázových systémov.

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Typické zraniteľnosti aplikácií a opatrenia proti nim

Nedostatočná validácia vstupov (pokr.):

- Pod validáciou vstupov rozumieme takú kontrolu vstupných údajov, ktorá zabezpečí, že prijatím a následným spracovaním týchto údajov nedôjde k narušeniu bezpečnosti aplikácie alebo iných komponentov IKT.
- Validácii musia podliehať všetky vstupy do aplikácie bez ohľadu na zdroj.
- Blacklist vs Whitelist

KPMG
cutting through complexity™

Účelové vkladanie kódu (Code Injection, Injekcia kódu)

- Injekcia kódu je súborným názvom pre mnoho druhov útokov, ktoré sú založené na vkladaní kódu (do aplikácie), ktorý je spracovaný aplikáciou.
- Takýto útok môže byť napr. vykonaný pridaním reťazca znakov do cookie alebo do hodnôt argumentov v linke URL.

Účelové vkladanie kódu (Code Injection, Injekcia kódu) – pokr.

- Tento typ útoku zneužíva nedostatočnú validáciu vstupno/výstupných dát, napríklad:
  - triedu dovolených znakov (štandardné triedy regulárnych výrazov, alebo vlastné),
  - dátový formát,
  - množstvo očakávaných dát,
  - pre číselný vstup, povolené hodnoty vstupu.

Vkladanie neoprávnených SQL dotazov (SQL Injection)

- Útoky „SQL injection“ sa realizujú vkladaním, alebo tiež injekciou SQL kódu do vstupných dát prenášaných medzi klientom a aplikáciou.
- Útoky SQL injekcie sú takými typmi injekčných útokov, v ktorých sú príkazy SQL injektované do existujúcich legitímnych SQL vstupov tak, aby došlo k spusteniu útočníkom preddefinovaných modifikovaných SQL príkazov.

Vkladanie neoprávnených SQL dotazov (SQL Injection) – pokr.

- Využitie SQL injekcie, ktorá úspešne prekoná bezpečnostné funkcie aplikácie je umožňuje útočníkovi:
  - čítať citlivé dáta z databázy,
  - modifikovať dáta v databáze (vkladanie, aktualizácia, mazanie),
  - spúšťať administratívne operácie nad databázou (vypnutie systému riadenia databázy – SRBD, anglicky DBMS),
  - obnovenie obsahu určitého súboru prítomného v databáze a
  - v špeciálnych prípadoch tiež spúšťanie príkazov na operačnom systéme.

Cross-site Scripting (XSS)

- Cross-Site Scripting útoky sú takým typom injekčného útoku, v ktorom sú potenciálne škodlivé skripty injektované do inak neškodných a dôveryhodných webových stránok.
- Cross-Site Scripting (XSS) útoky nastávajú, keď útočník použije webovú aplikáciu na zaslanie škodlivého kódu inému používateľovi.

Cross-site Scripting (XSS) – pokr.

- Vo väčšine prípadov sa jedná o škodlivý kód vo forme skriptu na strane prehliadača.
- Chyby, ktoré umožňujú úspešnosť takéhoto útoku sú vo všeobecnosti pomerne rozšírené a nastávajú kdekoľvek, kde webová aplikácia využíva vstup od používateľa na generovanie vlastného výstupu bez toho, aby ho overovala, alebo zakódovala.

Pretečenie zásobníka (pamäte)

- Chyba typu pretečenie zásobníka nastáva, keď aplikácia pri zapisovaní údajov do rôznych dátových štruktúr nekontroluje, prípadne kontroluje nedostatočne, či veľkosť zapisovaných údajov nie je väčšia než veľkosť dátovej štruktúry do ktorej sa zápis vykonáva.
- Ak je veľkosť zapisovaných údajov väčšia, po vyčerpaní priestoru v alokovanej dátovej štruktúre môže dôjsť k prepísaniu rôznych aplikačných alebo systémových riadiacich dátových štruktúr. V takomto prípade najčastejšie dochádza k havárii aplikácie a ukončeniu jej činnosti.

Pretečenie zásobníka (pamäte)

- Ak však útočník dokáže ovplyvňovať (dodať) údaje, ktorými bude tento prepis vykonaný, môže okrem zhodenia aplikácie dosiahnuť aj spustenie vlastného kódu na cieľovom systéme.
- Kód, ktorý chce útočník spustiť obvykle posiela aplikácii v rámci údajov, ktorými bude vykonaný prepis spomínaných dátových štruktúr.

Atomickosť operácií a race conditions

- V databázových systémoch je nedeliteľnosť (alebo aj atomickosť) jedna z vlastností ACID transakcie. ACID transakcia, je transakcia ktorá spĺňa nasledovné požiadavky:
  - Atomickosť
    - *Vyžaduje, aby každá transakcia bola spracovaná spôsobom „všetko alebo nič“. Ak nezbehne časť transakcie, nezbehne ani transakcia ako celok a databáza zostane nezmenená.*
  - Konzistentnosť
    - *Táto vlastnosť zabezpečuje to, že každá transakcia presúva databázu z jedného validného stavu do iného validného stavu.*

Atomickosť operácií a race conditions (pokr.)

- Izolácia
  - *Táto vlastnosť zabezpečuje to, že súčasne prebiehajúce transakcie tranzakcie presunú systém do stavu, ktorý by nastal ak by boli tieto transakcie vykonané sériovo, t.j. jedna po druhej.*
- Trvalosť
  - *Trvalosť znamená, že keď už raz bola transakcia potvrdená a vykonaná (commit), zostane v platnosti aj v prípade výpadku prúdu, havárie alebo prípadných chýb.*

Atomickosť operácií a race conditions (pokr.)

- V atomickej transakcii, je rad operácií nad databázou vykonaný tak, že buď sú vykonané všetky, alebo nedôjde k žiadnej.
- Záruka atomicity bráni tomu aby došlo k aktualizácia databázy len čiastočne, čo môže spôsobiť väčšie problémy, než odmietnuť celý rad operácií úplne.

Race condition

- Race condition je druh chyby v softvéri, ku ktorej môže dôjsť v dôsledku toho, že aplikácia predpokladá, že určité operácie prebehnú v určitom poradí. Ak to však nie je pravda, a zmení sa poradie týchto operácií, aplikácia môže zostať v nekonzistentnom stave, čo môže mať za následok porušenie integrity údajov alebo aj prerušenie behu aplikácie.
- Race condition situácia môže mať aj bezpečnostné implikácie a môže poslúžiť potenciálnemu útočníkovi napríklad na manipulácie aplikácie alebo obídenie rôznych bezpečnostných opatrení.
- Chybou je, že aplikácia niečo predpokladá, ale neoveruje si, či je predpoklad splnený, race condition túto chybu využíva.

Zneužitie prístupových práv

- Často majú aplikácie oveľa viac prístupových práv a privilégií než v skutočnosti potrebujú na svoju činnosť.
- Typickým prípadom je situácia, keď aplikácia beží s administrátorskými oprávneniami.

Zneužitie prístupových práv (pokr.)

- Z tohto dôvodu potom dochádza k zneužitiu prístupových práv a to hlavne v dvoch situáciách:
  - Po úspešnom zneužití chyby v aplikácii má útočník širšie možnosti ďalšej eskalácie kompromitácie údajov, samotnej aplikácie alebo systému na ktorom aplikácia beží
  - Samotná aplikácia môže poskytovať možnosti čítania a manipulácie údajov, ku ktorým by používateľ (alebo útočník) za bežných okolností nemal prístup.

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

OPATRENIA NA ZNÍŽENIE DOPADOV APLIKAČNÝCH CHÝB

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Opatrenia na zníženie dopadov aplikačných chýb

- V ideálnom prípade by softvér nemal obsahovať žiadne bezpečnostné nedostatky. Žijeme však v reálnom svete, kde je väčšinou proces návrhu a vývoja rôznym spôsobom limitovaný.
- Preto je rozumné predpokladať, že každá aplikácia v konečnom dôsledku obsahuje zraniteľnosti, ktoré zatiaľ neboli identifikované.

KPMG
cutting through complexity™

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Opatrenia na zníženie dopadov aplikačných chýb

- Existuje riziko, že po nasadení našej aplikácie do prevádzky, v nej môže neskôr niekto (potenciálny útočník) objaviť zraniteľnosť, ktorú aj následne úspešne zneužije.
- Na takýto prípad sa vieme pripraviť využitím rôznych techník a nástrojov, ktoré zmiernia dopad takéhoto útoku.

KPMG
cutting through complexity™

Opatrenia na zníženie dopadov aplikačných chýb

- Medzi uvedené techniky a nástroje patria:
  - Separácia oprávnení (Privilege separation)
  - Modularizácia (Kompartmentalizácia)
  - Obmedzenie prístupu k aplikáciám
  - Obmedzenie prístupových práv aplikácie
  - Uväznenie aplikácie (Jailing)
  - Aplikačné firewally

Separácia oprávnení (Privilege separation)

- Separácia oprávnení je technika používaná pri programovaní aplikácií a v počítačovej bezpečnosti, pri ktorej je program rozdelený do častí (modulov), ktoré majú pridelené len tie oprávnenia, ktoré potrebujú na plnenie určitej úlohy.
- Používa sa pre zmiernenie potenciálnych škôd pri úspešnom útoku na aplikáciu.

Separácia oprávnení (Privilege separation) – pokr.

- Bežnou metódou pre realizáciu separácie oprávnení, je rozdelenie (pomocou systémovej funkcie „fork“) bežiaceho procesu na dve oddelené procesy.
- Hlavný program (väčší) sa vzdá vysokých oprávnení a menší program si tieto vysoké oprávnenia uchová, aby mohol neskôr vykonať určité privilegované operácie.

Separácia oprávnení (Privilege separation) – pokr.

- Menší program (ten s vysokými oprávneniami) je vytvorený tak, aby čistým spôsobom (t.j. jednoducho a prehľadne) vykonával privilegované operácie a má menej rozhraní a jednoduchšie rozhrania, než druhý väčší program vykonávajúci všetky ostatné aktivity.
- Takýto návrh zvyšuje pravdepodobnosť odhalenia prípadných zraniteľností v privilegovanej časti kódu.

Obmedzenie prístupu k aplikáciám

- Každá aplikácia môže obsahovať bezpečnostné nedostatky, ktoré dosiaľ neboli objavené, prípadne nie sú známe prevádzkovateľovi aplikácie.
- Z toho dôvodu je dobrou praxou obmedziť prístup k aplikácii len pre tých legitímnych používateľov a systémy, ktoré danú aplikáciu potrebujú pre svoju prácu.

Obmedzenie prístupu k aplikáciám (pokr.)

- Principiálne sa obmedzenie prístupu zabezpečuje hlavne:
  - Lokálne v rámci systému
  - Na sieťovej úrovni

Obmedzenie prístupových práv aplikácie

- V informačnej bezpečnosti existuje tzv. princíp najmenších oprávnení (tiež známy ako zásada minimálnych oprávnení alebo zásada najnižšej autority), ktorý požaduje, aby na určitej vrstve abstrakcie výpočtového prostredia, mal každý modul (napr. proces, používateľ, program alebo jeho časť v závislosti na kontexte), prístup len k informáciám a zdrojom, ktoré sú nevyhnutné pre jeho legitímny účel.
- Znamená to, že napr. používateľský účet má mať pridelené iba tie oprávnenia, ktoré sú nevyhnutné pre prácu konkrétneho používateľa.

Obmedzenie prístupových práv aplikácie (pokr.)

- Napríklad používateľ pre zálohovanie nemusí inštalovať softvér, takže má mať práva iba na spustenie zálohovania a spustenie aplikácií súvisiacich so zálohovaním.
- Akékoľvek ďalšie privilégiá, napríklad pre inštaláciu nového softvéru, sú blokované.
- Táto zásada by mala platiť aj pre osobný počítač používateľa, ktorý zvyčajne pracuje na svojom normálnom používateľskom účte a na administrátorský účet chránený heslom sa prihlasuje (a následne pod ním pracuje) iba vtedy, keď si to situácia nevyhnutne vyžaduje.

Uväznenie aplikácie (Jailing)

- Uväznenie aplikácie (Jailing) môžeme zjednodušene charakterizovať ako obmedzenie aplikačných účtov na konkrétny adresárový strom a vybrané príkazy.
- Na vytvorenie Jail prostredia v prostredí operačných systémov UNIX slúži systémové volanie chroot.
- Funkcia chroot v operačných systémoch UNIX vykoná operáciu, ktorá zmení zdanlivý koreňový adresár pre aktuálne bežiaci proces a jeho pod-procesy.
- Program, ktorý je spustený v takto upravenom prostredí sa nedokáže odkazovať (a preto zvyčajne nemá prístup) na súbory mimo určenej adresárovej štruktúry. Upravené prostredie sa nazýva "chroot jail" alebo jednoducho Jail.

Uväznenie aplikácie (Jailing) – pokr.

- Chroot prostredie možno použiť na vytvorenie a prevádzkovanie samostatnej virtualizovanej kópie softvérového systému. To môže byť užitočné okrem iného pre:
  - Testovanie a vývoj
  - Riadenie softvérových závislostí
  - Kompatibilita
  - Zotavenie systému
  - Oddelenie privilégií

Aplikačné firewally

- Aplikačný firewall je typ firewall-u, ktorý kontroluje vstup, výstup a/alebo prístup z, alebo na aplikácie alebo služby.
- Pracuje tak, že monitoruje a prípadne blokuje vstup, výstup alebo volanie systémovj služby, ak nespĺňa politiku nastavenú na firewall-e.
- Aplikačný firewall je zvyčajne schopný riadiť sieťovú prevádzku na ľubovoľnej vrstvy OSI až hore k aplikačnej vrstve.

Aplikačné firewally (pokr.)

- Je schopný kontrolovať aplikácie alebo konkrétne služby, na rozdiel od stavového sieťového firewall-u, ktorý bez dodatočného softvéru nie je schopný kontrolovať prevádzku v sieti s ohľadom na konkrétnu aplikáciu.
- Existujú dve základné kategórie aplikačných firewallov:
  - Sieťové aplikačné firewall-y,
  - Systémové (host-based) aplikačné firewall-y;

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

POUŽÍVANIE OTVORENÝCH ŠTANDARDOV

KPMG
cutting through complexity™

Používanie otvorených štandardov

- Vytvárané alebo nakupované aplikácie, či vo všeobecnosti softvér ako taký, by mali byť postavené na otvorených štandardoch a nie na uzavretých proprietárnych, ktorých detaily nie sú verejne dostupné.
- Hlavné nevýhody aplikácií postavených na proprietárnych štandardoch:
  - Nemusia byť interoperabilné s existujúcimi alebo budúcimi produktmi iných výrobcov
  - Väčšinou nie je známa úroveň bezpečnosti týchto štandardov
  - Zákazník môže zostať v závislosti na službách a produktoch konkrétneho dodávateľa

Ministerstvo financií
Slovenskej republiky

EFQM
Recognised for excellence
5 star - 2012

Otázky?

KPMG
cutting through complexity™