PV005: Služby počítačových sítí (PS 2007)

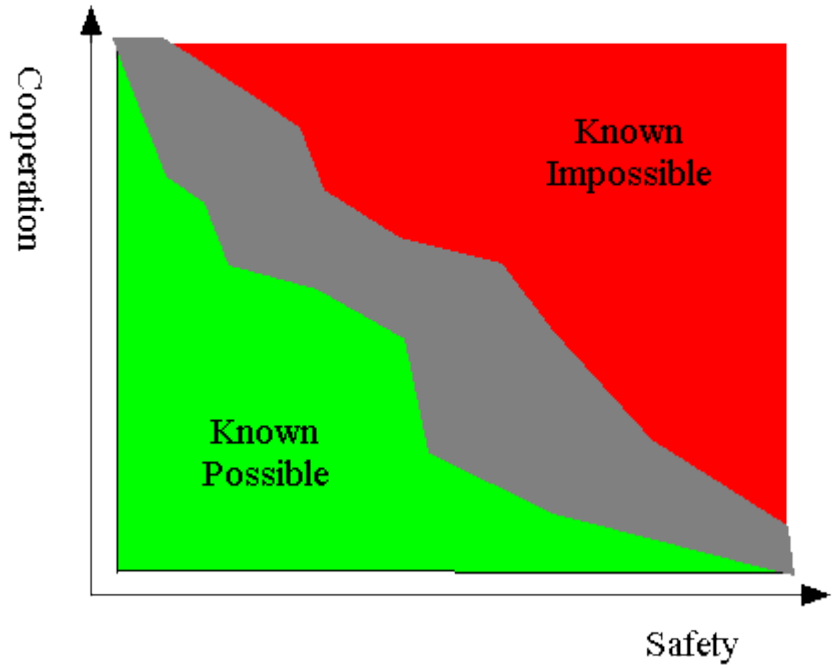
Bezpečnost na síti (vymezení problematiky)

Roman Žilka, unix@fi

ÚVOD

Úplná bezpečnost neexistuje

[http://www.caplet.com/security/taxonomy/boundary.gif]



~ hledání kompromisu mezi vypnutým počítačem a nechráněnou sítí

O čem bude tato přednáška?

- co by se vám mělo vybavit, když se řekne „síťová bezpečnost“ (přátelské vyprávění o hledání nekonečna); nic jiného
- strašákovi jménem „bezpečnost“ se nelze vyhnout skoro na žádné IT pozici (přesněji: snažte se mu nevyhýbat)

Computer security is a bug

- původně nikdo nebral otázky bezpečnosti v úvahu
- přístup opakovaný při návrhu operačních systémů (UNIX – multiuživatelský OS už od počátku!), síťových protokolů (IP, RIP, OSPF, FTP) atd. („hlavně ať to funguje“)

- kdo by taky v Arpanetu potřeboval bezpečnost?
- důsledek: spousta děr v SW, řešených dodatečně; také spousta nadšených *hackerů*

Hackerů? (kulturní vsuvka)

- domníváte se, že víte? (Steven Levy – Hackers: Heroes of the Computer Revolution)
- MIT, AI Lab.
- skutečný hacker: bažící po znalostech, neúnavný, vděčný, nápomocný, apriori dává veškeré znalosti a objevy k dispozici komunitě („human knowledge belongs to the world, like Shakespeare or Aspirin“)
- volné proudění znalostí v komunitě a jeho pozitivní důsledky

- rebelie proti represivní byrokracii, konvencím, překážkám v přístupu k informacím, předsudkům vůči počítačům
- hackerská etika: tak tedy dobrý nebo zlý člověk? (sabotáž chráněných účtů a bezpečnostních mechanismů v UNIXu)
- časem: úpadek etiky, pokroucení a zneužití ideálů v duchu egoistických záměrů (zviditelnění se, vybití adrenalinu, obecná kriminalita – krádeže, vydírání atd.) => posun významu slova „hacker“ do moderní, populární podoby

- úzké hackerské komunity (obzvl. okolo free softwaru) nadále ctí etiku
- dnes: odborníci v oboru vs. *script-kiddies* a jejich hrozba; členění útočníků podle zázemí a kvalifikace

Ochrana informací, specifika

- bezpečnost není stav, ale proces!
- vést útok po síti je levné, pohodlné, anonymní, masově přístupné
- replikovatelnost digitálních dat: neexistuje pojem „originálu“ (digitální data jako důkaz u soudu?)

- skutečná bezpečnost systému vs. „pocit bezpečí“; co to je spolehlivost – jak byste ji definovali?
- neexistuje zcela zabezpečený systém (do jaké míry má smysl něco jistit?)

Privátní vs. veřejné sítě

- *privátní*: proprietární, plně ve vlastní režii (dráty -> protokoly -> data), nehrozí nabourání zvenčí
- důraz na jednoduchost, robustnost, propustnost, odezvu
- bezpečnost se zaměřuje na udržení konzistence dat
- pro seriózní nasazení nákladné – kritická data (banky – zálohy dat, ATM)
- *veřejné*: Internet – často je zkrátka potřeba

- k Internetu je připojena spousta potenciálních zákazníků/spotřebitelů způsobem, o který se nestaráme
- protokoly a linky jsou definované, resp. provozované cizími subjekty => menší možnost ovlivnit parametry přenosu
- data může leckdo v síti vidět (omezuje pouze routing)
- Internet není implicitně bezpečný – viz nejpoužívanější protokoly OSI vrstev: HTTP, TCP/IP(v4), Ethernet, …
- navíc třeba řešit autenticitu a důvěrnost dat

Požadavky na bezpečnostní systémy

- v závislosti na dané aplikaci
- *autenticita*: data skutečně pochází z uváděného zdroje
- *integrita*: data nebyla po odeslání autorem změněna
- *důvěrnost*: data nemohou být přečtena neautorizovaným subjektem
- *dostupnost*: autorizovaný subjekt má k datům přístup (… za blíže určených podmínek …)
- *účtovatelnost*: systém umí evidovat informace o přenášených datech; data svou strukturou evidenci podporují

KRYPTOGRAFIE

1. pád: kdo? co?

- *kryptografie*: utajování informací, šifrování
- *kryptoanalýza*: odtajňování informací, dešifrování
- *kryptologie*: souhrnné označení
- *steganografie*: taktéž utajování informací (takže jak?)
- transformace textu A -> B
- proč šifrujeme na síti? – veřejné sítě
- požadované vlastnosti kryptosystému: ze znalosti textu B neodvodíme A (důvěrnost), malá změna v A způsobí fundamentální změnu B (proč je tato vlastnost potřeba?)
- „dětské šifry“ – dohodnutým (tajným) algoritmem získáme B (např. ROT13)

- ~ nevhodné pro masové užívání (vždy vymýšlet nový algoritmus?…)
- ~ pevný (všeobecně známý) algoritmus, ovšem závislý na tajné hodnotě (*klíči*) – *Kerckhoffův princip*
- ~ *symetrický kryptosystém*: $B = f_1(A,k)$; $A = f_2(B,k)$
- ~ *asymetrický kryptosystém*: $B = f_1(A,k_1)$; $A = f_2(B,k_2)$
- ~ *hašování*: $B = f(A)$; $A = ?(B)$

Symetrická kryptografie

- ~ komunikanti sdílejí tajemství – klíč (předaný apriori bezpečnou cestou)
- ~ zajistí důvěrnost (kdo nezná klíč, nepočte si)

- ~ zajistí autenticitu (spolu s otevřenou zprávou putuje i zašifrovaná zpráva; příjemce porovná přijatou zprávu otevřenou a rozšifrovanou)
- ~ [23]DES, RC[245], IDEA, Skipjack, AES (Rijndael), Blowfish
- ~ základní kryptografický princip, velmi rychlé algoritmy
- ~ ale problém: jak si tajně (a dostatečně pohodlně a rychle) předat klíč?

Asymetrická kryptografie

- ~ každý z komunikantů vlastní unikátní pár klíčů: *veřejný* a *privátní*
- ~ matematické kouzlo: zprávy zašifrované veřejným klíčem lze rozšifrovat pouze příslušným párovým privátním klíčem a naopak
- ~ ze znalosti jednoho z klíčů (veřejný/privátní) nelze odvodit odpovídající párový klíč
- ~ klíče bývají dost dlouhé, aby se statisticky „vyloučilo“, že dva lidé náhodně nezávisle vygenerují stejný pár

- typický scénář: veřejné klíče Alice i Boba jsou všeobecně známé, Alice zašifruje zprávu pro Boba jeho veřejným klíčem, Bob zprávu rozšifruje svým privátním klíčem (zajištěna důvěrnost)
- výhoda proti symetrické kryptografii: není třeba složitě dodávat druhé straně tajný klíč
- pomalé algoritmy; klíčová, ale jen pomocná role – např. *hybridní systémy*
- velké množství sofistikovaných algoritmů, které slouží pro řadu účelů
- např. zajištění autenticity: odesilatel zašifruje zprávu svým privátním klíčem, pošle šifru a otevřenou zprávu příjemci, ten rozšifruje odesilatelovým veřejným klíčem a opět porovná

- RSA, DSA, ElGamal
- zajištění autenticity a integrity veřejných klíčů? (viz dále)

Hybridní kryptosystémy

- široce používaný princip při každodenní zabezpečené komunikaci na Internetu
- řeší problém pomalosti asymetrických algoritmů
- asymetrickým (důvěrnost+autenticita) způsobem dojde k dohodnutí jednorázového, unikátního *klíče sezení*, který je pak použit pro šifrování vlastní výměny dat symetrickým způsobem

Hašování

- *hašovací funkce* generuje známým algoritmem pro libovolnou vstupní zprávu výstup (*hash*, *checksum*) fixní délky
- obor hodnot je dramaticky menší (navíc konečnou) množinou než definiční obor
- *bezkoliznost* (pro každý jeden možný vstup generovat unikátní výstup) není možné zajistit
- ideální hašovací funkce má *uniformní rozložení hodnot*: každý možný výstup je generovaný „stejným počtem“ různých vstupů
- hash je tedy jakási pseudo-unikátní charakteristika daného vstupu, který je řádově delší

- CRC, MD[245], SHA, HAVAL, SNEFRU
- použití: kontrola integrity (downloady, digitální podpis, uložení hesel)

Digitální podpis

- *elektronický podpis*: cokoli (jméno na konci e-mailu)
- *digitální podpis*: navíc zajišťuje autenticitu (obvykle doprovázeno důvěrností)
- v digitálním světě veřejných a privátních klíčů všeobecný princip autentizace autora (podepsat můžete cokoli z nul a jedniček)
- podpisem se rozumí zašifrování zprávy vlastním privátním klíčem a přiložení této šifry ke zprávě

- realita: podepsaná zpráva M' = M ++ $Encrypt_{privk}(Hash(M))$
- sám o sobě nezajišťuje důvěrnost; většina systémů digitálního podpisování zašifruje M' veřejným klíčem příjemce, aby ji jenom ten mohl dešifrovat
- příjemce šifrovanou+podepsanou zprávu rozšifruje svým privátním klíčem a veřejným klíčem odesilatele ověří podpis

Bezpečnostní otázky kryptosystémů

- síla šifry závisí na délce klíče
- proč? - *útok hrubou silou* trvá déle (cca nekonečně dlouho)
- problém uchování soukromých klíčů v tajnosti

~ problém ověření integrity veřejných klíčů (*certifikáty*: důvěryhodná třetí strana podpisem stvrzuje vazbu subjekt<->klíč)
~ řada nezabezpečených protokolů je obalována (*encapsulated*) na prezentační vrstvě OSI šifrovacími metadaty: SSL (HTTPS, SMTPS, IMAPS, POP3S, telnet a SSH, …)

BĚŽNÉ ÚTOKY V SÍTI

Firewall a síťová vrstva

- kdo se nedostane dovnitř, nemůže způsobit žádnou škodu? – IP úroveň => před čím chrání a před čím nikoli? (viry, slabá hesla, rootkity, …)
- porty: blokovat, filtrovat, uzavírat co jde
- vnitřní síť | *demilitarizovaná zóna* (DMZ) | svět – různé funkční topologie
- typicky zastává roli brány/routeru
- *paketový filtr*: monitorování a modifikace paketů
- specializovaná HW zařízení vs. univerzální systémy (PC s univerzálním OS)
- modelový příklad – iptables
- *aplikační brána*: firewall na aplikační vrstvě

- dívá se dovnitř probíhající komunikace na vyšších protokolech
- parsování paketů často náročné na výpočetní výkon brány (IRC/ICQ chat, P2P sdílení, ...)
- *proxy*: klient komunikuje s protějškem zprostředkovaně přes proxy (např. HTTP proxy)
- aplikační brána vs. proxy – často překrývající se pojmy

Denial of Service (DoS)

- znepřístupnění služby zahlcením/vyřazením serveru
- zahlcení z jednoho zdroje: silnější linka na slabší (dnes už netypické)
- DDoS (*Distributed DoS*) – zapojení více zdrojů

- speciální pakety: nezvykle velké, nezvykle malé, …
- častěji: zneužití chyby v softwaru, zhroucení (aplikačního) serveru
- ochrana: firewall (počet paketů/čas), bezpečný SW

Spoofing

- podvržení údajů v komunikaci
- fenomén zasahující spektrum protokolů (vrstva datových spojů až aplikační)
- IP: podvržení odesilatele (obejití firewallu); ochrana: IPSec, IPv6
- MAC: (získání zajímavé IP adresy z DHCP)
- DNS: ovlivnění globálních dat v DNS za cílem přesměrování komunikace

- WWW: obvykle *phishing* (viz dále)
- SMTP: podvržené hlavičky (From, Reply-To, …)

Útoky na hesla

- když už je port otevřený a aplikace „nemá bezpečnostní chyby“; jak může jeden ledabylý uživatel ohrozit celou síť
- ukládání hesel: plain, hash („nevadí“ čitelnost hesla)
- dobrá vs. špatná hesla a proč: *útoky hrubou silou*, *slovníkové útoky*
- dostupný hash usnadní lámání hesla
- ochrana: dobrá hesla, skryté hashe hesel

Phishing (rhybaření)

- ~ mystifikace uživatelů Internetu za účelem získání jejich soukromých informací (čísla/hesla k účtům, data v počítači – formou virů a rootkitů z podvržených updatů či mailů)
- ~ masová aplikace sociálního inženýrství (Kevin Mitnick)
- ~ novodobá záležitost; průvodní jev spamu
- ~ pěkný příklad loni u nás: Česká spořitelna (spam odkazující na zinscenovaný web)
- ~ důvěřovat, ale prověřovat!

Viry, rootkity & spol.

- ~ Windows bodují
- ~ specifikum *virů*: replikují a šíří sebe sama
- ~ šíření: chyby v programech (obvykle e-mail a WWW klienti a různé servery) – spuštění vlastního kódu s právy uživatele, aneb firewall nestačí
- ~ specifikum *rootkitů*: instalovaný hackerem po získání administrátorských privilegií na obsazeném stroji
- ~ malware pracuje obvykle nenápadně; někdy sám sebe zlikviduje, až splní svůj účel
- ~ společné rysy: odesílání zajímavých dat do Internetu, distribuce spamu, inventarizace okolí (scanování sítě, atp.), provoz FTP a jiných serverů s „warezem“, otevření zadních dvířek pro útočníky

Odbočka: typické chyby v programech

- dotýká se každého kusu spustitelného kódu v počítači (tar, wmf lib, Thunderbird, Apache, …)
- programátoři je stále opakují
- kontrola návratových kódů funkcí, kontrola vstupních parametrů (délka, nečekané znaky), ošetření výjimek, správné chování mezí struktur (např. *off-by-one*)
- následky: *buffer overflow*, *arbitrary code execution*

ZÁVĚREM

Rady do života

- ~ paranoidnější vyhrává; vždycky myslete na to, že existuje nějaký (byť sebepodivnější) vám momentálně neznámý způsob, jak váš systém nabourat
- ~ celý systém je tak bezpečný, jako je bezpečný jeho nejslabší článek
- ~ šifrujte (SSL kde je možné – hlavně web a pošta oběma směry)
- ~ udržujte veškerý (opravdu!) SW na počítači aktuální
- ~ vypínejte nepotřebné síťové služby (které otevírají porty)
- ~ používejte silná hesla, nikam si je nepište a nechte si je pro sebe

- buďte skeptičtí k tomu, co spouštíte (stažené programy – checksumy, skripty schované v HTML e-mailech, …)
- Windows: antiviry&spol.
- naučte se naslouchat svému počítači: když začne z ničeho nic pracovat harddisk, zjistěte proč; položte si modem na stůl a ujistěte se, že víte, proč se právě rozblikaly kontrolky