Crypto-World

Informační sešit GCUCMP
ISSN 1801-2140

Ročník 7, číslo 78/2005 **1. srpen 2005**

78/2005

Připravil: Mgr. Pavel Vondruška
Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
http://crypto-world.info
(875 registrovaných odběratelů)

Obsah : **str.**

Příloha : Dešifrace textu zašifrovaného Enigmou (enigma.pdf)
(volné pokračování článku z Crypto-Worldu 5/2005, str. 2-3 : *Výzva k rozluštění textu zašifrovaného Enigmou*)
příloha je dále volně dostupná na adrese: http://crypto-world.info/casop7/enigma.pdf

A. Pozvánka k tradiční podzimní soutěži v luštění ….

Mgr. Pavel Vondruška, (pavel.vondruska@crypto-world.info)

Vážení čtenáři, 15.9.2005 začíná tradiční **podzimní soutěž v luštění jednoduchých šifrových textů o ceny – *Soutěž v luštění 2005***. Obdobné soutěže pořádal náš e-zin v letech 2000 až 2004. V roce 2000 byly úlohy zaměřeny na klasické šifrové systémy. V roce 2001 soutěž pokračovala řešením "moderních“ systémů. Soutěž v roce 2003 a 2004 byla z hlediska předložených úloh zaměřena na řešení úloh od hříček, přes jednoduché šifry až po klasické šifrové systémy (jednoduchá záměna, transpozice, periodické heslo, Fleissnerova otočná mřížka, jedno-dvoumístná záměna, …).

Pokud se chcete podívat na tyto starší úlohy a jejich řešení, můžete je nalézt na stránce našeho e-zinu v sekci věnované soutěžím http://crypto-world.info/souteze.php .

Loni se do soutěže zaregistrovalo přesně 100 řešitelů, podmínku pro zařazení do losování o ceny (zisk 15-ti bodů) splnila téměř polovina přihlášených soutěžících (45). Všech 19 úloh vyřešilo 8 soutěžících.

Letošní soutěž bude navazovat na ročníky 2003 a 2004. Objeví se obdobné šifrové systémy, které letos „posílí“některé další historické šifry a techniky.

Na základě vyhodnocení vašich e-mailů se pokusím soutěž připravit podle vašich představ:

- úlohy spíše jednodušší (takové, které lze zvládnout spíše přemýšlením než „tupým“ použitím PC…)
- větší počet úloh
- zachovat systém udělování cen tj. ceny pro prvé tři řešitele a další tři ceny předat vylosovaným řešitelům, kteří splnili stanovené podmínky

včetně pokusu zapracovat poněkud protichůdné požadavky:

- úlohy zveřejňovat najednou / postupně (☺)
- méně / více napovídat (☺)

Pravidla soutěže, přehled cen a první sadu úloh najdete v příštím čísle našeho e-zinu Crypto-World 9/2005, který vyjde kolem 15.9.2005. Všechny informace budou dostupné i na našem webu v sekci věnované soutěžím http://crypto-world.info/souteze.php.

Soutěž je určena pouze registrovaným čtenářům našeho e-zinu, do soutěže bude nutné (tak jako v minulých ročnících) se zaregistrovat. Heslo k registraci bude rozesláno společně s kódy ke stažení e-zinu 9/2005.

Celkový vítěz se bude moci zúčastnit mezinárodního workshopu Mikulášská kryptobesídka http://www.buslab.cz/mkb/ .

Všem luštitelům přeji hodně úspěchů, vytrvalost, radost ze soutěže a samozřejmě i trochu štěstí při losování.

Na závěr si můžete pocvičit v luštění :

MEKRO KMÍNV RPÁNÍ ČAZLI MCÍSI TANDO HCOPI (čínské přísloví)

B. Kontrola certifikační cesty, část 2

Ing. Peter Rybár, NBÚ SR, (pr@mailbox.sk, sep@nbusr.sk)

Nasledujúci článok, ktorého druhá a posledná časť nasleduje, bol v čase publikovania v Crypto-World v procese medzinárodných diskusií s odborníkmi na oblasť elektronického podpisu. Rovnako čitatelia časopisu Crypto-World majú možnosť prispieť k vytvoreniu užitočných dokumentov a svoje vylepšenia ku článku poslať na e-mail sep@nbusr.sk. Výsledný dokument bude zverejnený na stránkach slovenského NBÚ http://www.nbusr.sk/NBU_SEP/8.php, aby pomohol pri informovaní odbornej a možno aj laickej verejnosti o základných pravidlách, ktoré sú roztrúsené v rôznych normách, štandardoch a ich súhrnné a ucelené pochopenie je preto v tejto oblasti dosť náročné a tak nasledujúci dokument sa snaží pomôcť k lepšiemu a rýchlejšiemu rozvoju v danej oblasti, rovnako ako NIST na stránke http://csrc.nist.gov/pki/testing/pathdiscovery.html.

5 Atribúty certifikátu X.509 a CRL pri kontrole certifikačnej cesty

Pri zostavovaní certifikačnej cesty sa kontroluje zhoda v položkách, ktoré obsahuje vydaný certifikát a certifikát vydavateľa. Ak párovú položku obsahuje vydaný (alebo vydavateľov) certifikát, tak ju musí obsahovať aj vydavateľov (alebo vydaný) certifikát, inak sa certifikát zamietne.

Každé vytváranie certifikačnej cesty musí spĺňať minimálne nižšie uvedené kroky, aby sa zabránilo nejednoznačnému vytvoreniu certifikačnej cesty.

5.1 Postupnosť kontroly položiek pri vytváraní certifikačnej cesty

1. *Issuer* meno vydavateľa sa musí zhodovať so *Subject* menom subjektu v certifikáte vydavateľa
2. *Issuer Alternative* meno vydavateľa sa musí zhodovať so *Subject Alternative* menom v certifikáte vydavateľa
3. formát kvalifikovaného certifikátu musí zodpovedať certifikátu X.509 v3 v DER kódovaní
4. podpis vydaného certifikátu sa musí overiť verejným kľúčom z certifikátu vydavateľa (pri RSA sa odporúča SHA-1 ale maximálne od 1. januára 2009 SHA-256)
5. v *AuthorityKeyIdentifier* sa *keyIdentifier* musí zhodovať so *SubjectKeyIdentifier* v certifikáte vydavateľa, každý certifikát musí obsahovať *SubjectKeyIdentifier*
6. ak certifikát v *AuthorityKeyIdentifier* obsahuje *authorityCertIssuer*, tak meno sa musí zhodovať s *Issuer* menom v certifikáte vydavateľa
7. ak certifikát v *AuthorityKeyIdentifier* obsahuje *authorityCertSerialNumber*, tak sa musí zhodovať so *serialNumber* v certifikáte vydavateľa
8. certifikát musí byť v čase kontroly platný, teda nebol zneplatnený v CRL alebo OCSP. Postup je popísaný v kapitole 3.
9. čas kontroly musí byť v intervale (*notBefore, notAfter)* - certifikát nesmie byť v čase kontroly expirovaný
10. interval platnosti certifikátu musí byť v intervale platnosti certifikátu vydavateľa
11. dĺžka cesty je kontrolovaná podľa zmenšujúcej sa premennej *maxPathLength*, *maxPathLength* sa nastaví na *BasicConstraints.pathLengthConstraint* z certifikátu, ak

je *BasicConstraints.pathLengthConstraint* menšia než *maxPathLength*. Ak je *maxPathLength* nulová, tak potom za certifikátom nemôže nasledovať CA certifikát

12. certifikát vydavateľa musí obsahovať:
    - *BasicConstraints.Ca* nastavené na *TRUE*,
    - *KeyUsage* obsahujúce *keyCertSign a* ak podpisuje CRL tak aj *cRLSign*
13. kvalifikovaný certifikát koncovej entity musí obsahovať:
    - *KeyUsage* s hodnotou *nonRepudiation* a prípadne aj *digitalSignature*
    - ak je certifikát určený pre podpisovanie časových pečiatok tak *extendedKeyUsage* musí minimálne obsahovať *id-kp-timeStamping*
    - ak je certifikát určený pre podpisovanie OCSP, tak *extendedKeyUsage* musí obsahovať minimálne *id-kp-OCSPSigning*
14. kvalifikované certifikáty podľa slovenskej legislatívy obsahujú OIDy *id-etsi-qcs-QcCompliance* a *id-etsi-qcs-QcSSCD* v rozšírení *QCStatements* alebo v rozšírení *CertificatePolicies*.

Kvalifikovaný certifikát koncovej entity (podpisovateľa) vydaný akreditovanou CA musí obsahovať authorityKeyIdentifier, v ktorom musia byť vyplnené všetky tri položky v prípade, že by mohlo dôjsť k nejasnostiam pri zostavovaní certifikačnej cesty, pričom položka *keyIdentifier* musí byť vždy vyplnená.

Rozšírenie certifikátu id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }

```
AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier             [0] KeyIdentifier           OPTIONAL,
    authorityCertIssuer       [1] GeneralNames            OPTIONAL,
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL  }
```

Medzi ďalšie položky, ktoré je potrebné vedieť spracovať a kontrolovať patria:

- *Policy Mappings* - mapovanie politiky
- *Name Constraints* - obmedzuje menný priestor podstromu
- *Policy Constraints* - obmedzuje politiky a mapovanie

Položky, ktoré pomáhajú pri podpisovaní a tiež pri zostavovaní a overovaní certifikačnej cesty:

- *CRL Distribution Points* - obsahuje cesty pre získanie CRL
- *Authority Information Access* - obsahuje cesty pre získanie certifikátu
- *Subject Information Access* - obsahuje cesty na TimeStamp server

6 Overovanie certifikačnej cesty

(PVM - Path Validation Module)

Procedúra PVM je jednou z kritických komponentov aplikácie, ktorá overuje certifikačnú cestu certifikátov X.509. Aplikácia overovanú cestu získa buď z podpisu, alebo si ju zostrojí na základe podmienok pre vytvorenie certifikačnej cesty.

Výsledok procedúry PVM na overenie certifikačnej cesty musí byť zhodný s výsledkom overenia podľa dokumentu RFC 3280 v 6. kapitole. Aplikácia nemusí používať algoritmy presne podľa kapitoly 6. RFC 3280, ale výsledky musia byť zhodné.

PVM potvrdí aplikácií dôveryhodnosť certifikátu koncovej entity, pre vybraný účel, ku ktorému sú vytvorené certifikačné cesty z dôveryhodných koreňových certifikátov, napríklad z certifikátu NBÚ. V praxi je možné vytvorenie aj viacerých ciest a aplikácia by mala vedieť takýto stav spracovať a vybrať akceptovateľnú cestu.

Akceptovateľnú cestu môže aplikácia vybrať na základe vstupných podmienok do PVM. Vstupom môže byť dôveryhodný certifikát, ale aj množina certifikačných politík, ktorá určuje aká cesta je akceptovateľná. Vstupy do algoritmu sú získané aj z podpisovej politiky, ak bola aplikáciou použitá. Výber cesty na základe certifikačnej politiky je určený hodnotami v rozšíreniach certifikátov a to hlavne: *Certificate Policies, policy Mappings, Policy Constraints, Inhibit Any-Policy a QcStatements.*

Certifikáty, ktoré sa považujú za kvalifikované podľa zákona o elektronickom podpise, vytvárajú cestu od NBÚ koreňového certifikátu a identifikujú sa na základe OIDu *id-etsi-qcs-QcCompliance*, ktorý môže byť umiestnený v rozšírení certifikátu *CertificatePolicies* alebo umiestnený v rozšírení certifikátu *QcStatements.* OID *id-etsi-qcs-QcSSCD* musí byť minimálne v kvalifikovaných certifikátoch koncovej entity.

```
id-etsi-qcs OBJECT IDENTIFIER ::= { itu-t(0) identified-organization(4)
etsi(0) id-qc-profile(1862) 1 }

id-etsi-qcs-QcCompliance OBJECT IDENTIFIER ::= { id-etsi-qcs 1 }

esi4-qcStatement-1 QC-STATEMENT ::= { IDENTIFIED BY id-etsi-qcs-
QcCompliance }

     An Identifier of the statement (represented by an OID) made by
     the CA, stating that this certificate is issued as a Qualified
     Certificate according to Annex I and II of the EU Directive
     1999/93/EC [1], as implemented in the law of the country where
     the CA is established.

id-etsi-qcs-QcSSCD OBJECT IDENTIFIER ::= { id-etsi-qcs 4 }

esi4-qcStatement-4 QC-STATEMENT ::= { IDENTIFIED BY id-etsi-qcs-QcSSCD }

     An Identifier of the statement (represented by an OID), made by
     the CA, stating that the private key associated with the public
     key in the certificate is stored in a Secure Signature Creation
     Device according to Annex III of the EU Directive 1999/93/EC
     [1], as implemented in the law of the country where the CA is
     established.
```

Pri kontrole certifikátu, či spĺňa požadovanú množinu certifikačných politík, sa očakáva neprázdny prienik požadovanej množiny politík so zjednotením OIDov z rozšírení *CertificatePolicies* a *QcStatements.* Množinu ovplyvňujú aj hodnoty v *policyMappings*, *PolicyConstraints* a *InhibitAnyPolicy.*

6.1 Algoritmus zostavenia certifikačnej cesty

Vstupom do algoritmu zostavovania certifikačnej cesty sú hodnoty, z ktorých niektoré môžu byť aj súčasťou podpisovej politiky.

Zoznam údajov a pravidiel potrebných pri zostavovaní certifikačnej cesty:

- **Certifikát koncovej entity**, ktorej verejným kľúčom sa overuje elektronický podpis.
- Množina používateľom vyžadovaných politík ***userInitialPolicy***,
  - ktorých prienik s položkami certifikátu (*certificatePolicies* + *QcStatements*) je neprázdna množina
  - ak je v *userInitialPolicy anyPolicy*, akceptovateľné sú všetky politiky.
- ***validPolicySet*** množina akceptovateľných politík, ktorá vzniká ako prienik s položkami certifikátu (*certificatePolicies* + *QcStatements*) a ak je povolené mapovanie politík, tak aj ako zjednotenie prostredníctvom *policyMapping*. Na začiatku *validPolicySet* nie je inicializovaná a preberie položky z koreňového certifikátu
- **Zoznam certifikátov**, ktoré môžu tvoriť certifikačnú cestu.
- **Zoznam CRL**.
- **Čas kontroly**, v ktorom sa certifikačná cesta zostavuje a kontroluje (napríklad čas najstaršej platnej časovej pečiatky podpisu).
- **Zoznam implicitne dôveryhodných CA** certifikátov vo forme *selfSigned* certifikátov.
- ***permittedSubtrees*** je množina mien, do ktorej musia patriť všetky mená subjektov certifikátov nasledujúcej certifikačnej cesty smerom od koreňového certifikátu. Na začiatku nie je inicializovaná a akceptuje všetky mená. Kontrolujú sa len položky obsiahnuté v *permittedSubtrees,* teda certifikát môže obsahovať ľubovolné iné. Aplikácia musí vedieť spracovať:
  - *subject - DistinguishedName*
  - *subjectAltName.directoryName - DistinguishedName*
  - *subjectAltName.rfc822Name*
  - *subjectAltName.dNSName*
  - *subjectAltName.uniformResourceIdentifier*
  - *subjectAltName.iPAddress*
- ***excludedSubtrees*** je množina mien, do ktorej nesmú patriť všetky mená subjektov certifikátov nasledujúcej certifikačnej cesty smerom od koreňového certifikátu. Na začiatku algoritmu je množina prázdna. Stav overenia *permittedSubtrees* nemá vplyv na overovanie *excludedSubtrees*. Kontrolujú sa len položky obsiahnuté v *excludedSubtrees,* teda certifikát môže obsahovať ľubovolné iné. Aplikácia musí vedieť spracovať:
  - *subject (DistinguishedName)*
  - *subjectAltName.directoryName (DistinguishedName)*
  - *subjectAltName.rfc822Name*
  - *subjectAltName.dNSName*
  - *subjectAltName.uniformResourceIdentifier*
  - *subjectAltName.iPAddress*
- ***requireExplicitPolicySkipCerts*** je hodnota, ktorá určuje, koľko certifikátov bude preskočených, než sa začne vyžadovať kontrola s *userInitialPolicy*. Počiatočná hodnota je nastavená na dĺžku cesty + 1. Nasledujúce certifikáty môžu hodnotu len zmenšiť. A ak je hodnota v *PolicyConstraints.requireExplicitPolicy* menšia, tak na túto hodnotu. Ak je jej hodnota nulová, vyžaduje sa, aby pre aktuálny a každý ďalší certifikát v certifikačnej ceste existovala neprázdna množina z prieniku:

  - *userInitialPolicy* a zjednotenia (*policyIdentifier* z *CertificatePolicies* a *statementId* z *QcStatement*)
  - a rovnako aj neprázdny prienik s *userInitialPolicy* a *validPolicySet.*
- ***inhibitPolicyMappingSkipCerts*** je hodnota ktorá určuje, koľko certifikátov bude preskočených, než sa zakáže mapovanie politík. Počiatočná hodnota je nastavená na dĺžku cesty + 1. Nasledujúce certifikáty môžu hodnotu len zmenšiť. A ak je hodnota v *PolicyConstraints.inhibitPolicyMapping* menšia, tak na túto hodnotu.
- ***InhibitAnyPolicySkipCerts*** je hodnota ktorá určuje, koľko certifikátov bude preskočených, než sa zakáže použitie *AnyPolicy.* Počiatočná hodnota je nastavená na dĺžku cesty + 1. Nasledujúce certifikáty môžu hodnotu len zmenšiť. A ak je hodnota v *InhibitAnyPolicy* menšia, tak na túto hodnotu. Hodnota a práca s *AnyPolicy*, ak nie je v ďalších častiach spomínaná, alebo cez *InhibitAnyPolicySkipCerts* zakázaná, tak sa uplatňuje pri množinových operáciách so zjednotenou množinou OIDov z rozšírení *CertificatePolicies* a *QcStatements.*
- ***maxPathLength*** je inicializovaný na hodnotu dĺžky certifikačnej cesty a zmenšuje sa pri každom CA certifikáte, okrem *selfSigned* certifikátu. Nasledujúce certifikáty môžu hodnotu len zmenšiť. A ak je hodnota v *BasicConstraints.pathLenConstraint* menšia, tak na túto hodnotu. *MaxPathLength* určuje, koľko CA certifikátov môže v ceste ešte nasledovať.

6.2 Vyhľadávanie certifikačných ciest pomocou rekurzívnej procedúry

Algoritmus vyhľadávania certifikačnej cesty musí zabezpečiť nájdenie certifikačnej cesty po implicitne „dôveryhodný certifikát“. Pri prehľadávaní ciest musí byť zabezpečené, aby sa algoritmus nezacyklil vďaka krížovej certifikácií jednotlivých CA certifikátov. Ak lokálna databáza neobsahuje certifikát vydavateľa, je možné ho získať z adresy uloženej v *AuthorityInformationAccess.*

Certifikačnú cestu môžeme popísať nasledovne:

- Pre všetky certifikáty $C_{i,}$ kde i je z { 1, ..., n - 1 } platí, že meno subjektu certifikátu C_{i+1} je meno vydavateľa certifikátu $C_{i.}$
- Certifikát C_n, je implicitne dôveryhodný *selfSigned* certifikát.
- Certifikát $C_{1,}$ je certifikát koncovej entity, ktorý overujeme a ktorej verejný kľuč je použitý na overovanie napríklad podpisu dokumentu alebo časovej pečiatky.

Pri overovaní podpisu sa najprv snažíme nájsť certifikát koncovej entity, ktorý vyhľadáme na základe identifikátorov: *issuer, serialNumber* a *SubjectKeyIdentifier.* Vhodnosť certifikátu pre požadovaný účel sa overí na základe rozšírení certifikátu, napríklad: *KeyUsage*, *ExtendedKeyUsage, CertificatePolicies, QcStatements* alebo ďalších rozšírení podľa typu použitia certifikátu.

Ak sa podarilo získať certifikát koncovej entity $C_{i,}$ s i = 1, pokračuje sa rekurzívnym algoritmom, ktorý sa snaží nájsť požadovanú certifikačnú cestu.

1. Pokúsime sa nájsť k certifikátu C_i, certifikát C_{i+1}, ktorý spĺňa podmienky špecifikované v kapitole 5.1 a ktorý nie je v množine zakázaných certifikátov, alebo

už obsiahnutý v certifikačnej ceste (okrem *selfSigned*), alebo už nájdených cestách, aby sa zabránilo rekurzívnemu zacykleniu.
2. Ak sa nepodarilo nájsť certifikát $C_{i+1,}$ uložíme C_i do množiny zakázaných certifikátov, zmenšíme i o 1 a ak je i = 0 algoritmus sa ukončí, inak sa pokúsime nájsť ďalšiu cestu, napríklad cez krížové certifikáty zopakovaním od kroku 1.
3. Ak je certifikát $C_{i+1} = C_{i,}$ tak bol nájdený koreňový certifikát.
    a. Ak je C_i implicitne dôveryhodný, tak si certifikačnú cestu po C_i odložíme do zoznamu nájdených ciest, zmenšíme i o 1 a ak je i = 0 algoritmus sa ukončí, inak sa pokúsime nájsť ďalšiu cestu, napríklad cez krížové certifikáty, zopakovaním od kroku 1.
    b. Ak nie je C_i implicitne dôveryhodný, uložíme C_i do množiny zakázaných certifikátov, zmenšíme i o 1 a ak je i = 0 algoritmus sa ukončí, inak sa pokúsime nájsť ďalšiu cestu, napríklad cez krížové certifikáty, zopakovaním od kroku 1.
4. Certifikát C_{i+1} uložíme do aktuálnej cesty, zväčšíme i o 1 a pokračujeme krokom 1

Ak certifikačná cesta je vytvorená alebo kontrolovaná na základe podpisovej politiky, ktorá obsahuje zoznam dôveryhodných koreňových certifikátov, tak potom implicitne dôveryhodné certifikáty sú len tie, ktoré sa nachádzajú aj v zozname dôveryhodných certifikátov v podpisovej politike. Inak povedané, pri použití podpisovej politiky je implicitne dôveryhodný certifikát len ten, ktorému overovateľ dôveruje a je súčasne aj v podpisovej politike.

Ak zoznam nájdených ciest nie je prázdny, potom algoritmus vyberie najvhodnejšie certifikačné cesty (na základe legislatívnych alebo technologických požiadaviek ...) a overí ich nasledujúcim algoritmom, ktorý je uvedený v kapitole 6.3., inak sa algoritmus ukončí s chybou o nezostrojení cesty k implicitne dôveryhodnému koreňovému certifikátu.

6.3 Overenie nájdených certifikačných ciest

Vstupom do procesu overovania môžu byť už pripravené certifikačné cesty z elektronických podpisov, alebo z predchádzajúceho algoritmu na vyhľadanie certifikačných ciest. Certifikačné cesty sa budú overovať každá samostatne.

Nultým krokom pri overovaní je zmena poradia certifikátov v overovanej certifikačnej ceste:

- Pre všetky certifikáty $C_{i,}$ kde i je z { 1, ..., n - 1 } platí, že meno subjektu certifikátu C_i je meno vydavateľa certifikátu $C_{i+1.}$
- Certifikát C_1 je implicitne dôveryhodný *selfSigned* certifikát.
- Certifikát C_n je certifikát koncovej entity, ktorý overujeme a ktorej verejný kľúč je použitý na overovanie, napríklad podpisu dokumentu alebo časovej pečiatky.

Pre i = 1 až n prekontroluj certifikačnú cestu s certifikátmi $C_{i,}$ a ak sa niektorá z nasledujúcich podmienok nesplní, algoritmus sa ukonči s výsledkom NEPLATNÝ, inak PLATNÝ.

1. Ak i = 1 over či je C_i *selfSigned* certifikát a či je implicitne dôveryhodný a over C_i s C_i podľa postupu v kapitole 5.1.
2. Ak i > 1 over certifikát C_i s C_{i-1} podľa postupu v kapitole 5.1.
3. Na základe postupu v kapitole 3. sa vyhľadá CRL v lokálnej databáze a ak očakávané CRL nie je dostupné, tak sa získa z adresy uloženej v *CRLDistributionPoints*. Overí sa podpis CRL s certifikátom C_{i-1} a overí sa platnosť C_i pomocou CRL.
4. Prekontroluje sa C_i pomocou *permittedSubtrees* podľa kapitoly 6.1.
5. Prekontroluje sa C_i pomocou *excludedSubtrees* podľa kapitoly 6.1.
6. Ak certifikát C_i obsahuje *NameConstraints.permittedSubtrees,* tak ulož do *permittedSubtrees* prienik z *NameConstraints.permittedSubtrees* a *permittedSubtrees.*
7. Ak certifikát C_i obsahuje *NameConstraints.excludedSubtrees,* tak ulož do *excludedSubtrees* zjednotenie z *NameConstraints.excludedSubtrees* a *excludedSubtrees.*
8. Do množiny *explicitPolicis* ulož zjednotenie OIDov z certifikátu C_i z položiek *CertificatePolicies* a *QcStatements*
9. Ak je *requireExplicitPolicySkipCerts* = 0, tak potom:
    a. musí byť neprázdny prienik medzi *userInitialPolicy* a *explicitPolicis*
    b. musí byť neprázdny prienik medzi *validPolicySet* a *explicitPolicis.*
10. *validPolicySet* nastav na hodnotu prieniku *validPolicySet* a *explicitPolicis*
11. Ak je *inhibitPolicyMappingSkipCerts* > 0, tak potom sa pokús vytvoriť množinu *mapedPolicies* tak, že pre každú dvojicu z certifikátu C_i *PolicyMappings*(*issuerDomainPolicy*, *subjectDomainPolicy*), ktorej *issuerDomainPolicy* je v množine *validPolicySet*, pridaj do *mapedPolicies* OID *subjectDomainPolicy.* Nakoniec do *validPolicySet* ulož zjednotenie *validPolicySet* a *mapedPolicies*.
12. Ak je *requireExplicitPolicySkipCerts* = 0, tak potom musí byť neprázdny prienik medzi *userInitialPolicy* a *validPolicySe.*
13. Ak certifikát C_i obsahuje *PolicyConstraints.requireExplicitPolicy* a hodnota je menšia než je v *requireExplicitPolicySkipCerts*, nastav *requireExplicitPolicySkipCerts na hodnotu PolicyConstraints.requireExplicitPolicy.* Inak zmenši *requireExplicitPolicySkipCerts* o jedna.
14. Ak certifikát C_i obsahuje *PolicyConstraints.inhibitPolicyMapping* a hodnota je menšia než je v *inhibitPolicyMappingSkipCerts,* nastav *inhibitPolicyMappingSkipCerts na hodnotu PolicyConstraints.inhibitPolicyMapping.* Inak zmenši *inhibitPolicyMappingSkipCerts* o jedna.
15. Ak certifikát C_i obsahuje *InhibitAnyPolicy* a hodnota je menšia než je v *InhibitAnyPolicySkipCerts,* nastav *InhibitAnyPolicySkipCerts na hodnotu InhibitAnyPolicy.* Inak zmenši *InhibitAnyPolicySkipCerts* o jedna.
16. Ak *maxPathLength* je nulové, môže už len nasledovať certifikát koncovej entity $C_{i=n.}$
17. Ak certifikát C_i obsahuje *BasicConstraints.pathLenConstraint* a *maxPathLength* je väčšia, tak sa *maxPathLength* nastaví na *BasicConstraints.pathLenConstraint*. Inak sa *maxPathLength* zmenší o jedna.
18. Certifikát C_i nesmie obsahovať kritické neznáme rozšírenia certifikátu.
19. Pokračuje sa spracovávaním ďalšieho certifikátu, bodom 1.

C. Honeypot server zneužit k bankovním podvodům, Část 1 – Technické řešení

Ondřej Suchý, LOGIOS s.r.o., (ondrej.suchy@logios.cz)

Co je to honeypot

Honeypot je zdroj, jehož smysl spočívá v jeho neautorizovaném využití ([1]). Je to úmyslně nastrčená nástraha v podobě zranitelného informačního systému, serveru, programu či dat. Sledováním takové pasti můžeme analyzovat bezpečnostní incidenty.

Informace o konkrétním příkladu

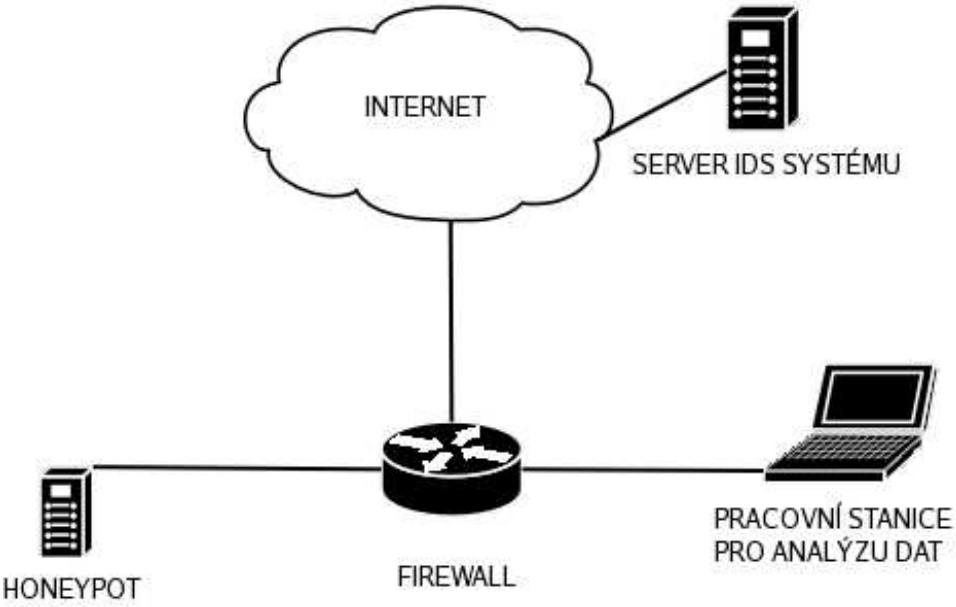
V květnu 2005 naše firma informovala odbornou veřejnost o zajímavém pokusu. K Internetu jsme úmyslně připojili nechráněný server („honeypot“). To, že systém bude brzy nalezen a zneužit automatickými programy, jsme tak trochu čekali, ale překvapila nás rychlost, s jakou se to stalo, a hlavně to, co následovalo.

Již dva dny po instalaci na server kdosi instaloval falešné stránky americké internetové banky a rozeslal výzvu, aby uživatelé zadali své přístupové údaje pod záminkou jakési kontroly – dosti brutální pokus o moderní on-line bankovní podvod, zvaný „phishing“.

V následujícím textu popíšu technické řešení naší pasti a uvedu některé postřehy. Pokud chcete podobné nástrahy konstruovat, naše zkušenosti vám snad budou užitečné. Pokud neplánujete být obětí bankovního podvodu, stejně doporučuji přečíst. Aspoň pro zamyšlení.

Technické řešení

Rozložení sítě



Honeypot systém byl umístěn v „demilitarizované zóně“. Od Internetu a lokální sítě byl oddělen firewallem. Pracovní stanice, ze kterých jsme prováděli analýzu dat, se připojovaly do lokální sítě připojené k dalšímu rozhraní firewallu. Architektura byla doplněna Intrusion

Detection systémem, jehož monitorovací agent běžel přímo na firewallu a posbíraná data odesílal na IDS server, který byl připojen v housingovém centru.

Honeypot

Jako honeypot nástrahu jsme zvolili volně šiřitelný operační systém Red Hat Linux, instalovaný na vyhrazený počítač. Protože naším cílem bylo co nejrychleji přilákat útočníky, použili jsme starou verzi 7.3, která již není aktivně udržovaná. Z pohledu dnešních znalostí obsahuje desítky bezpečnostních problémů včetně několika zranitelností zneužitelných po síti. Po instalaci systému jsme nestahovali žádné bezpečnostní aktualizace, a kde to bylo možné, ponechali jsme implicitní nastavení bezpečnosti. Chtěli jsme, aby server vypadal atraktivně, proto jsme imitovali chování firemního „backend“ serveru. Spustili jsme tedy službu sdílení disků protokolem CIFS (Samba), poštovní server SMTP a program pro POP3 přístup[1]. Kromě toho běžely ještě další služby, ponechané z „default“ instalace.

V systému jsme založili reálně vypadající uživatele a do jejich domovských adresářů jsme vložili fiktivní data – excelovské soubory, dokumenty a další obvyklý obsah uživatelova disku. Aby nebylo nápadné, že server je v provozu teprve několik dní, při instalaci jsme posunuli systémový čas zpět. Před ostrým spuštěním jsme ho vrátili na aktuální. Není to dokonalé, ale dosáhli jsme aspoň iluze reálného provozu.

Firewall

Firewall používal volně šiřitelný operační systém OpenBSD UNIX ve verzi CURRENT k prosinci 2004. Měl tři ethernetová rozhraní – lokální síť s vyhrazeným privátním adresním rozsahem, Internet a „demilitarizovanou zónu“ s veřejnými routovatelnými IP adresami, do které jsme umístili honeypot. Pomocí stavové filtrace paketů prováděl firewall ochranu lokální sítě před útoky z Internetu a snažil se ji úplně skrýt před honeypot systémem, aby se případný útočník nemohl dostat k pracovním stanicím. Zároveň se snažil chránit Internet před případnými útoky ze strany honeypotu, o čemž se ještě zmíním dál.

Software pro analýzu dat, IDS

Klíčovou součástí instalace honeypot je software na sběr a analýzu dat z bezpečnostního incidentu. Použili jsme program Sebek, vytvořený a volně šířený v projektu Honeynet ([5]). Sebek se skládá ze sledovacího agenta a serverové části pro sběr dat. Agent je ve skutečnosti modul určený přímo pro jádro systému Linux, který umožňuje monitorovat aktivity v systému. Například zaznamenává stisknuté klávesy a umí dokonce rekonstruovat příkazy posílané zabezpečeným kanálem přes SSH. Zprávy agent průběžně vysílá na lokální síť, kde jsou zachytávány a dekódovány serverovou částí. Běh monitorovacího agenta v systému i zprávy na síti jsou maskovány, takže běžný útočník by je neměl odhalit (metody však existují, viz [6]). Server programu Sebek byl umístěn na firewallu.

Jako další zdroj dat pro analýzu incidentů jsme použili program tcpdump, kterým jsme přímo na firewallu zachytávali kompletní síťový provoz směřující směrem k nástraze či od ní. Z těchto dat jsme byli schopni přesně rekonstruovat síťové datagramy včetně jejich obsahu.

[1] Jsem si vědom, že „backend“ server sdílící disky protokolem CIFS obvykle není veřejně vystavován na Internetu. Na naši obhajobu bych rád uvedl, že jsem v praxi takové instalace skutečně viděl, zejména v menších firmách. Tak či onak, prosím o prominutí této nedokonalosti.

Abychom mohli v krátkém čase interpretovat data o útocích a měli k dispozici systém včasného varování, doplnili jsme výše uvedené programy systémem detekce útoků Snort. IDS program byl spuštěn přímo na firewallu a data odesílal do databáze na server umístěný na páteřní síti v kolokačním hostingovém centru. Databáze útoků byla spravována pomocí webového rozhraní BASE. Abychom byli o útocích na honeypot systém včas varováni, propojili jsme výstup IDS Snort se skriptem, který aktualizované údaje o útocích posílal pomocí SMS na mobilní telefony techniků.

Ochrana před útoky, IPS

Většina systémů napadených dnešními útočníky slouží pro krytí identity při dalších útocích, případně jsou využity (podobně jako ten náš) přímo k páchání podvodů. Jen malá část serverů je zneužita jinak, třeba ke krádeži dat nebo ke „sportovní“ změně webových stránek. Honeypot systém tedy pro své okolí představuje významnou bezpečnostní hrozbu. Neexistuje bohužel žádná snadná ochrana, dosud dostupné techniky spíš kombinují více způsobů. Zmíním několik možností:

Proti hromadným testům zranitelností dalších serverů, případně DoS útokům, které by z napadeného honeypotu směřovaly do Internetu, lze s úspěchem použít omezení počtu odchozích spojení a šířky datového pásma. Firewall s OpenBSD toto od verze 3.7 (či námi použité verze CURRENT z prosince 2004) umožňuje, linuxový netfilter ve spolupráci s paketovými frontami též. My jsme omezili počet odchozích spojení na 6, šířku datového pásma na 32 kbps.

Další metodou, která je k dispozici, je Intrusion Prevention System. Můžeme použít třeba volně šiřitelný Snort-Inline. Je to modifikace IDS Snort, která útoky nejen detekuje, ale i zastavuje. Snort-Inline je psán přímo pro honeypot projekty, takže útoky jsou blokovány tak, aby útočník nic nepoznal. Tento program je bohužel pouze pro Linux. Při našem pokusu, který využíval firewall se systémem OpenBSD, jsme se museli obejít bez něj.

Vedle technologických opatření bych chtěl zdůraznit ještě jedno organizační. Jak vyplývá z našich skutečností, je dobré honeypot systém soustavně sledovat „živou“ obsluhou, která je schopna vyhodnocovat zachycené informace a včas reagovat. Díky upozornění na útoky zasílanému přes SMS jsme byli schopni všechny útoky živě sledovat. Jakmile útočník začal rozesílat výzvy k zadání hesla a hrozilo úspěšné vykonání podvodu, pokus jsme přerušili. Těžko domyslet, co by se mohlo stát bez nepřetržitého sledování systému. Nikomu, kdo chce podobné experimenty zkoušet, nedoporučujeme spoléhat jen na softwarovou ochranu Internetu před honeypotem. Systém nepřetržitě sledujte.

Závěr

Nyní už zbývá vše pouze zapojit do správných zástrček, nastartovat počítače a čekat. První útok přišel po čtyřech minutách, ale byl to pokus červa CodeRed o zneužití zranitelnosti ve starších Windows. Na svého hackera jsme čekali dva dny.

Použitý software

Celý honeypot systém je postaven výhradně na open source volně šiřitelném software.

Honeypot nástraha	Red Hat Linux 7.3
Firewall	OpenBSD UNIX
Zachytávání informací	Sebek + Tcpdump
IDS	Snort + MySQL + Apache + BASE

Praktické postřehy

Při zachytávání dat pomocí software Sebek velmi rychle roste velikost datového souboru. I z neaktivního systému, na kterém běží jen periodické systémové úlohy, lehce získáte 5 GB dat za den. Je nutné mít na monitorovacím počítači dostatečnou diskovou kapacitu, případně záznamy denně „rotovat“ a archivovat.

Signatury útoků v IDS systému Snort, kterým sledujeme honeypot systém, by měly být před reálným nasazením co nejlépe upraveny na míru honeypot systému, aby nebyl operátor obtěžován falešnými poplachy.

Pokud na firewallu použijeme systém Linux, je vhodné pro ochranu Internetu před útoky z našeho honeypotu využít IPS software Snort-Inline. V každém případě je užitečné omezit možnosti odchozích spojení a systém průběžně sledovat.

IDS systém hlásil pokus o útok červa CodeRed na procedury systému Microsoft Windows do čtyř minut po připojení honeypot serveru k Internetu. Systém Red Hat Linux 7.3 vydržel dva dny, což také není mnoho. Z toho vyplývá, že je nutné pro všechny veřejně přístupné počítače stanovit a pečlivě dodržovat politiku včasného aplikování záplat („patching policy“).

Rizika honeypotů

Kromě bezprostředních technických rizik pro okolí, která už byla naznačena, představuje Honeypot pro svého provozovatele potenciální právní rizika. V úvahu připadá například otázka odpovědnosti za škodu, která by vznikla třetí osobě prostřednictvím našeho úmyslně nezabezpečeného systému, či otázka ochrany osobních údajů. Nejsem právník, a proto nedokážu tato rizika fundovaně posoudit. Doporučuji prostudovat uvedené zdroje. Dobrá zpráva je, že v Crypto-Worldu by měl vyjít článek od zasvěcených odborníků, který nás s právními riziky honeypot technologií seznámí.

Zdroje informací

[1] Spitzner L.: *Honeypots: Catching the Insider Threats*, IEEE Computer Society
[2] Stolfo S.: *Worms and Attack Early Warning*, IEEE Computer Society
[3] Spitzner L.: *Honeypots: Are They Illegal?*, magazín SecurityFocus, online na Internetu: http://www.securityfocus.com/infocus/1703
[4] Spitzner L.: *The Value of Honeypots, Part Two: Honeypot Solutions and Legal Issues*, magazín SecurityFocus, online na Internetu: http://www.securityfocus.com/infocus/1498
[5] *the Honeynet Project*, online na Internetu: http://www.honeynet.org/
[6] Oudot L., Holz T.: *Defeating Honeypots*, magazín SecurityFocus, online na Internetu: http://www.securityfocus.com/infocus/1803, http://www.securityfocus.com/infocus/1805, http://www.securityfocus.com/infocus/1826

D. Potenciální právní rizika provozu Honeypot serveru

Mgr. Tomáš Sekera, (tomas.sekera@centrum.cz),

Autor je senior specialista pro bezpečnostní legislativu, právní záležitosti ČESKÝ TELECOM, a.s., publikuje zde jako soukromá osoba.

Dále uvedený názor hodnotí právní rizika, která mohou vyvstat v souvislosti s provozem tzv. Honeypot serveru, tak jak byl popsán v článku *Honeypot server zneužit k bankovním podvodům*, Crypto-World, číslo 78/2005, autor Ondřej Suchý.

Obecně lze konstatovat, že k hodnocení formou tohoto doktrinálního výkladu, zda provoz Honeypot serveru, případně již jeho samotná instalace a spuštění je trestným činem určitého individua, fyzické osoby (dále jen provozovatel), nepostačuje pouhé srovnání, zda takový čin naplňuje znaky konkrétního trestného činu (např. podvod, poškození a zneužití záznamu na nosiči informací, neoprávněné nakládání s osobními údaji), ale k případné trestní odpovědnosti je nutno rovněž zkoumat znaky uvedené v obecné části zákona č. 140/1961 Sb., trestní zákon, v platném znění - zavinění (úmysl přímý a nepřímý, nedbalost vědomá a nevědomá), okolnosti vylučující trestní odpovědnost, stadia trestné činnosti (příprava, pokus, dokonaný trestný čin) a zda se jedná o pachatele, spolupachatele nebo účastníka (organizátor, návodce, pomocník), případně jinou formu účastenství, dále stupeň nebezpečnosti činu pro společnost.

V daném případě by bylo pro ilustraci zajímavé uvažovat o hypotetické trestní odpovědnosti provozovatele serveru, např. ve vztahu k přípravě trestného činu podvodu dle ust. § 250 odst. 4 trestního zákona, kdy by provozovatel musel úmyslně vytvářet podmínky pro spáchání individuálního trestného činu a současně vědět, že svým jednáním může porušení nebo ohrožení zájmu chráněného trestním zákonem způsobit, a pro případ, že je způsobí, byl s tím srozuměn. Což lze u bezpečnostního odborníka důvodně předpokládat. Nesmělo by ovšem dojít k pokusu, ani dokonání trestného činu. Ale pro provozovatele by bylo nepříjemnější eventuální zkoumání jeho trestní odpovědnosti v postavení pomocníka, který by úmyslně opatřil prostředky ke spáchání trestného činu, pokud již došlo k pokusu nebo dokonání trestného činu. Zde by případná trestnost činu zanikla, pokud by technické prostředky při zjištění pokusu trestné činnosti učinil nefunkčními, tedy upustil od dalšího jednání potřebného k dokonání trestného činu a odstranil nebezpečí, které vzniklo zájmu chráněnému trestním zákonem z podniknutého pokusu, nikoliv dokonaného činu. Zdůrazňuji, že pro účast formou pomoci k trestnému činu je nutné, aby úmysl pomocníka individuálně směřoval ke konkrétnímu trestnému činu a adresátu, nepostačuje jen naplnění obecných znaků skutkové podstaty. Analogicky platí i pro návodce.

Pokud by ovšem činnost provozovatele byla v prostředí veřejného Internetu způsobilá (což zřejmě je) vyvolat u pachatele rozhodnutí spáchat trestný čin (který se možná dostal až do stadia pokusu), jednalo by se ze strany provozovatele o již dokonaný trestný čin podněcování, dle ust. § 164 trestního zákona. V tomto případě není rozhodné, zda byl trestný čin dokonán, či vůbec spáchán. **Jak vidno, podněcování je širším než návod a na popsanou causu pravděpodobně aplikovatelné.**

Samozřejmostí je, že **nemohou být při testování bezpečnosti užita „ostrá“ data konkrétních fyzických osob, resp. jejich osobní údaje, viz trestný čin neoprávněného nakládání s osobními údaji,** dle ust. § 178 odst. 2, 3 trestního zákona.

Oznamovací povinnost o spáchání trestného činu státnímu zástupci nebo policejnímu orgánu se týká trestných činů taxativně uvedených v ust. § 168 odst. 1 trestního zákona. Nejvíce by přicházela do úvahy právě při spáchání trestného činu neoprávněného nakládání s osobními údaji, dle ust. § 178 odst. 3 trestního zákona, přičemž beztrestnost za neoznámení je zde zajištěna tomu, kdo by oznámením mj. uvedl sebe v nebezpečí trestního stíhání.

Pro případ, že by se **provozovatel** hodnověrným způsobem dozvěděl o tom, že je připravován nebo páchán mj. trestný čin podvodu dle ust. § 250 odst. 4 trestního zákona, či jiné trestné činy taxativně uvedené v ust. § 167 odst. 1 trestního zákona, **je povinen překazit spáchání nebo dokončení takového trestného činu,** jinak by se vystavil nebezpečí trestního stíhání za spáchání trestného činu nepřekažení trestného činu, ve smyslu ust. § 167 trestního zákona.

Mimo trestní odpovědnosti se provozovatel může vystavit rovněž **riziku obecné odpovědnosti za škodu** podle zákona č. 40/1964 Sb, občanský zákoník, kterou by způsobil jinému provozní činností nebo porušením právní povinnosti. Mimo uvedené platí, že každý je povinen počínat si tak, aby nedocházelo ke škodám na majetku. Dále by mohla přicházet do úvahy dle téhož zákona **odpovědnost za škodu způsobenou neoprávněným zásahem do práva na ochranu osobnosti** (mj. ochrana soukromí, jména), při zpřístupnění osobních a dalších údajů útočníkovi.

Závěrem lze podle mého názoru konstatovat, že provoz úmyslně nezabezpečeného Honeypot serveru, který, jak plyne z výše uvedeného článku, byl nebo mohl být použit pro spáchání trestného činu, není bez právních rizik, včetně možného postihu. Uvedené i za předpokladu, že se provozovatel aktivně neúčastní úmyslné individualizované trestné činnosti.

Zejména je nutné zajistit, aby nebyli potencionální pachatelé bezpečnostními odborníky „uváděni v pokušení“, viz trestný čin podněcování a dále, aby byl monitoring provozu zajištěn takovým bezpečným a spolehlivým způsobem, který vyloučí dokonání trestného činu a zvláště zamezí vzniku škody třetí straně.

E. K některým právním aspektům provozování serveru Honeypot

JUDr. Ján Matejka, PhD., **(**matejka@ilaw.cas.cz**)**

Autor je vědeckým pracovníkem Ústavu státu a práva Akademie věd ČR a učitelem na Právnické fakultě ZČU v Plzni, kde přednáší předmět Internetové a počítačové právo.

V souvislosti s článkem Ondřeje Suchého, který je mj. publikován i v tomto čísle časopisu Crypto-World, může být zajímavé se ve stručnosti zabývat i některými bezprostředními právními aspekty, které s provozem takovéhoto serveru mohou souviset. Dovolte mi dvě krátké úvahy.

Úvaha první - Policejní Honeypot, aneb Poslední dějství internetové kriminality?

Vycházíme-li ze skutečnosti, že Honeypot představuje skutečně jakousi „vábničku" přitahující potencionální útočníky, lze se při úvahách o legalitě (tj.zákonnosti), resp. legitimitě (tj. oprávněnosti) z části inspirovat v názorově poměrně pestrých diskusích o přípustnosti či nepřípustnosti řízené (obvykle policejní, avšak někdy i zprostředkované soukromé) provokace[2]. Aniž bych chtěl předmětnou problematiku blíže srovnávat, zmíním pouze to, že jak platné právní řády naprosté většiny evropských zemí, tak i judikatura[3] těchto zemí (včetně Nejvyššího i Ústavního soudu v ČR[4]) se k těmto „důkazních prostředkům" staví značně skepticky, přičemž samotným základem pro rozhodování zde bývá vždy právní či politická úvaha o tom, zda vůbec lze pachatele stíhat a odsoudit za trestný čin, který byl někým vyprovokován (byť nepřímo), a to s tím, že není zcela jisté, zda by bez samotné „provokace" k tomuto trestnému činu vůbec došlo.

Jinými slovy, z pohledů dalšího možného užití záznamů (např. protokolových souborů) získaných v rámci provozu tohoto Honeypotu, nejde o řešení, které by bylo ze strany orgánů činných v trestním řízení jakkoliv zázračné. Jinými slovy, důkazní možnost v oblasti veřejnoprávní (trestněprávní, ale i správněprávní) jsou tedy značně omezené, a to dokonce i v případech, kdy je zcela zřejmá skutečnost, že byl spáchán trestný čin, a že jej spáchala konkrétní osoba, tedy i v těch případech, kdy útočník tzv. sedne na lep, a to se vším všudy.

---

[2] Tento typ jednání bývá často označován jako záměrné, aktivní či pasivní podněcování nebo navádění či jiné iniciování spáchání trestné činnosti druhé osoby.

[3] Jde např. o závazná rozhodnutí Evropského soudu pro lidská práva. Evropský soud pro lidská práva vykládá meze použitelnosti důkazů získaných činností policejních agentů tak, že v žádném případě nemůže dojít ke spáchání trestného činu, který by byl vyvolán zásahem policistů. Tento zásah a jeho použití v trestním řízení by zbavil obviněného ab initio a definitivně práva na spravedlivý proces ve smyslu čl. 6 Úmluvy.

[4] Tak se vyslovil i Ústavní soud České republiky ve svém nálezu, který zcela jednoznačně a kategoricky vyslovil nepřípustnost policejní provokace, neboť tato je v rozporu s čl. 8 odst. 2 Listiny. Ve svém nálezu se opírá o konstatování, že pokud náš právní řád zná institut skrytého agenta, pak podmínky jeho použití jsou výslovně upraveny v zákoně a podrobeny přísné kontrole. Jiný způsob použití než připouští zákon je postupem extra legem. Dále argumentuje, že iniciování trestného činu by bylo nepřípustným porušením čl. 39 Listiny a čl. 7 odst. 1. Úmluvy, podle kterých jen zákon musí stanovit, které jednání je trestným činem, a v tomto smyslu je zásah státu do skutkového děje nepřípustný. U policejních orgánů v případě použití provokace jde o zřejmý exces z jejich zákonem stanovených práv a povinností, tedy exces z povinnosti opatřovat důkazy o spáchaném nebo připravovaném trestném činu zákonným způsobem, který může zakládat i podezření ze spáchání trestného činu zneužití pravomoci veřejného činitele.

Úvaha druhá – (Spolu)odpovědnost provozovatele Honeypot serveru

Stranou důkazních možnosti HoneyPotů by nepochybně neměla zůstat úvaha o potenciální právní odpovědnosti samotných provozovatelů těchto serverů.

Z pohledu platné právní úpravy je třeba na tento Honeypot server nahlížet v zásadě podobně jako na jakoukoliv jinou nedostatečně zabezpečenou (např. nezáplatovanou) koncovou stanici (případně počítačovou sít). To však zdaleka neznamená, že za provoz takové stanice (Honeypotu) nemůže být její provozovatel shledán právně odpovědným, tím spíše v případě, že si tak počíná vědomě.

V nedávné době (tj. přesněji **do 7. září 2004**) bylo možné na tuto problematiku bez větších problémů aplikovat obecnou právní úpravu, která v tomto ohledu představovala velmi průřezovou oblast protínající řadu ustanovení obchodního, občanského, autorského a zejména trestního práva[5]. Samotná skutečnost, že provozovatel takové sítě nebyl primárním škůdcem (ale pouze se na škodě nepřímo podílel), nemusí být co do přiznání - mnohdy plné výše - žalovaného nároku na náhradu škody (či ušlého zisk) právně významná. Náš právní řád totiž vychází z principu spoluodpovědností za škodu (odpovědnost, kdy za škodu odpovídá více osob). V případě takové odpovědnosti pak občanský zákoník stanoví jako pravidlo odpovědnost solidární. To znamená, že **v případě více škůdců může poškozený subjekt požadovat úhradu celé škody na kterémkoliv z nich.** V případě solidární odpovědnosti mají pak subjekty stanovenu povinnost vypořádat se navzájem podle účasti na způsobení vzniklé škody. Samotná odpovědnost se zde řídila zejména občanským zákoníkem, a to konkrétně § 415 až § 450. Vzhledem k výše zmíněné „nepřímé sekundární odpovědnosti" za navozený protiprávní vztah však patrně bude třeba **prokázat porušení právní povinnosti**. V tomto ohledu se jako nejdůležitější zde zdá být aktuální povinnost předcházet škodám. Předcházení hrozícím škodám je upraveno v § 415 občanského zákoníku, podle něhož „*Každý je povinen počínat si tak, aby nedocházelo ke škodám na zdraví, na majetku, na přírodě a životním prostředí*.". Jak tomu v případě takovýchto obecných ustanovení obvyklé bývá, velmi významný podíl na výkladu zde hraje judikatura[6]. Protože neexistuje žádná zvláštní úprava, která by stanovila povinnosti provozovatele počítačových sítí, **je porušení této obecné povinnosti hlavním právním základem pro jeho případnou odpovědnost**. V tomto směru je nepochybně zapotřebí, aby provozovatel učinil určitá opatření směřující k ochraně práv třetích osob. **Jednání provozovatele takové sítě, který vědomě zanedbává zabezpečení své sítě** (vědomá pasivita)**, může být kvalifikováno jako porušením povinnosti ve smyslu § 415 ObčZ.** Náhrada způsobené škody pak bývá logickým důsledkem tohoto porušení.

Dne 7. září 2004 však nabyl účinnosti zákon č. 480/2004 Sb. o některých službách informační společnosti a o změně některých zákonů (zákon o některých službách informační společnosti). Tento zákon (mnohdy zbytečně příliš obecně označován jako antispamový) pozměňuje hned

[5] Z trestněprávní aspektů této problematiky je třeba zmínit zejména trestný čin poškození a zneužití záznamu na nosiči informací (§ 257a TrZ), který vzhledem k předpokládanému úmyslnému zavinění nelze aplikovat na nedbalostní jednání provozovatele nezabezpečené počítačové sítě.

[6] Např. bylo odvozeno, že v případě, že by banka znala závažné skutečnosti, které by předem vážně ohrožovaly zjištěný podnikatelský plán klienta, přicházela by v úvahu obecná odpovědnost předcházet škodám a odpovědnost s tím související podle § 415 ObčZ, pokud by na tyto okolnosti klienta neupozornila (Rozsudek Vrchního soudu v Praze sp. Zn. 5 Cmo 347/96.)

několik poměrně klíčových právních institutů, některé další pak jako zcela nové zavádí. Samotný důraz nové právní úpravy není zjevně kladen na známé šíření nevyžádaných obchodních sdělení, ale zejména na podstatné aspekty odpovědnosti poskytovatelů celé řady služeb informační společnosti. V tomto ohledu zákon upravuje tři základní omezení, a to:

- Odpovědnost poskytovatele služby za obsah přenášených informací (§3)

- Odpovědnost poskytovatele služby za obsah automaticky dočasně meziukládaných informací (§4)

- Odpovědnost poskytovatele služby za ukládání obsahu informací poskytovaných (§5)

Jedním z nikoli nepodstatných pravidel zákona je rovněž ustanovení §6, které stanoví, že tito **poskytovatelé služeb nejsou povinni dohlížet na obsah jimi přenášených nebo ukládaných informací, ani aktivně vyhledávat skutečnosti a okolnosti poukazující na protiprávní obsah informace**. Jinými slovy se zde vychází z předpokladu, že není rozumné požadovat od poskytovatelů, aby při vysokém počtu uživatelů a vysokém objemu uložených dat zjišťovali a posuzovali legálnost či nelegálnost veškerého obsahu uložených informací. Je tedy z části na uživatelích samotných a tedy především na jejich odpovědnosti, jaký obsah ukládají na poskytnutém serverovém prostoru. Poskytovatelé proto nemají povinnost aktivně monitorovat materiály, které jsou na jejich serverech uloženy. Pokud se však poskytovatel dozví o protiprávní povaze obsahu, a neučiní veškeré možné kroky vedoucí k odstranění či znepřístupnění takovéhoto obsahu, které po něm lze požadovat, stává se odpovědným za obsah uložených informací.

Patrně nejvýznamnější z těchto odpovědnost limitujících ustanovení je §3, který omezuje odpovědnost poskytovatele za obsah všech přenášených informací. Podle tohoto ustanovení odpovídá poskytovatel takovéto služby (typicky přenos informací poskytnutých uživatelem) za obsah přenášených informací pouze pokud takovýto přenos sám iniciuje (např. přenáší svá vlastní data), zvolí uživatele přenášené informace, případně pokud zvolí nebo změní obsah přenášené informace.

Poměrně významný je zde rovněž §5 a §6 zákona, který řeší problematiku odpovědnosti poskytovatelů služeb, která spočívá v ukládání informací poskytnutých uživatelem. Opětovně se zde vychází z principu neodpovědnosti těchto poskytovatelů, a to s tím, že se taxativně stanoví případy, kdy takovýto poskytovatel odpovídá. V tomto případě bude takovýto poskytovatel odpovídat za obsah informací uložených na žádost uživatele jen,

- mohl-li vzhledem k předmětu své činnosti a okolnostem a povaze případu vědět, že obsah ukládaných informací nebo jednání uživatele jsou protiprávní, nebo

- dozvěděl-li se prokazatelně o protiprávní povaze obsahu ukládaných informací nebo o protiprávním jednání uživatele a neprodleně neučinil veškeré kroky, které lze po něm požadovat, k odstranění nebo znepřístupnění takovýchto informací.

Jinými slovy, právní odpovědnosti provozovatelů těchto serverů je od okamžiku účinnosti tohoto zákona značně limitována, a to v zásadě jen na případy útoků, jejichž iniciátorem je sám provozovatel.

F. Přehledová zpráva o významných publikacích a projektech na téma poskytování anonymity, klasifikace a měřitelnost informačního soukromí (privacy), část 3.

Mgr. Marek Kumpošt , Fakulta informatiky, MU, Brno
(xkumpost@fi.muni.cz)

Tato zpráva má za cíl prezentovat dosavadní průzkum v dané oblasti a seznam významných publikací a projektů na dané téma, které jsem pročetl během prvního semestru Ph.D. studia na Fakultě Informatiky.

Zpráva je strukturována tak, že ke každé odkázané literatuře je připojen krátký text shrnující problematiku popisovanou v daném článku. V seznamu referencí uvádím veškerou literaturu, tedy i případně tu, která nebude v textu odkázána.

V závěrečné části tohoto třídílného přehledu se zaměříme na dvě významné disertační práce z oblasti anonymizačních systémů a dále budou uvedeny některé zbývající publikace, které se nepodařilo tematicky zařadit do některé z oblastí uvedených v částech 1 a 2.

Designing and attacking anonymous communication systems

Disertační práce George Danezise [Dan04] z University of Cambridge (pozn.: autor byl pozván na letošní Mikulášskou kryptobesídku, doufejme, že přijede :-). Hlavním cílem práce je popis autorem navrženého systému pro anonymní posílání e-mailů – Mixminion [MIX] a dále potom řídkých mixovacích sítí (sparse mix networks), které také prezentuje ve svém dřívějším článku věnovanému mixovacím sítím [Dan03].

Úvodní část práce je věnována definici anonymity a cílům anonymní komunikace jako takové. Dále v této části rozebírá možné hrozby anonymní komunikace a snahy útočníků o narušení anonymity. Vzhledem k tomu, že anonymitu je vhodné měřit, definuje autor anonymitu jako míru informace, kterou útočník potřebuje k tomu, aby mohl úspěšně identifikovat spojení subjektu s akcí. Uvádí pojem entropie a diskutuje jeho využití při vyjadřování míry anonymity poskytované zkoumaným systémem (použití ilustruje na příkladu s pool mixem).

V další části textu jsou uvedeny základní pojmy a stavební kameny použité v dalších částech práce. Je zde uveden smysl a význam hašovacích funkcí, šifrovacích mechanizmů (blokové a proudové šifry), aspekty asymetrické kryptografie a její typické využití. Posledním pojmem, který je v této části uveden, je pojem dopředné bezpečnosti (forward security), se kterým pak autor pracuje při popisu navrhovaného systému pro posílání zpráv.

Další kapitola je věnována přehledu související práce a projektů z oblasti anonymity. Zmiňuje zde existující systémy a principy pro poskytování anonymity (anon.penet.fi, anonymizer, remailer typu I, crowds, nym servers). Hlavní část této kapitoly je věnována popisu existujících mix systémů a členění podle jejich funkcionality. Také zmiňuje systém Onion Routing a systémy používané v peer-to-peer sítích. V závěru této kapitoly je uveden seznam nejběžnějších typů útoků společně s jejich krátkou charakteristikou.

Pátá kapitola je celá věnována popisu navrhovaného systému Mixminion [MIX], jeho funkcionalitě, implementaci a bezpečnostní analýze systému. Další kapitola je věnována aspektům dopředné bezpečnosti nebo dopředného utajení a uvádí, jak k tomuto problému přistupuje navrhovaný systém.

Sedmá kapitola se zabývá mixovacími sítěmi, konkrétně sítěmi založenými na řídkých grafech. Vysvětluje pojem řídkého grafu a diskutuje anonymitu poskytovanou sítí, která je založena na této topologii. V závěru kapitoly je uvedeno srovnání s klasickým přístupem při budování mixovací sítě a jsou zde uvedeny výsledky v síti založené na řídkých grafech.

Osmá kapitola se věnuje tzv. RGB mixům, což je takový systém, který umožňuje detekovat, zda je vůči mixu nebo mixovací síti veden aktivní útok. Schopnost mixů detekovat aktivní útok spočívá v tom, že mixy mají informace o svém okolí o stavu připojení uzlů v tomto okolí. Tyto informace mixy získávají tak, že posílají anonymní zprávy do sítě a tyto se jim pak vrací zpět. V kapitole je detailně popsán tento typ mixů a analýza bezpečnosti RGB mixů. Autor se problematice RGB mixů věnuje také v článku [DS03a].

Devátá kapitola je věnována aspektům statistického útoku na mixy. Je zde rozebrána podstata tohoto typu útoku a v závěru kapitoly jsou uvedeny konkrétní výsledky vzhledem ke zvolené mixovací strategii.

Závěrečná kapitola shrnuje výsledky práce a uvádí možné směry budoucího výzkumu v oblasti systému pro poskytování anonymity.

Pozn.: Popisu systému Mixminion se věnují autoři v článku [DDM03].

On the anonymity of anonymity systems

Disertační práce Andreie Serjantova [Ser03] z University of Cambridge. Hlavním cílem práce je rozbor a analýza různých systémů pro poskytování anonymity. V [SD02] se autoři Andrei Serjantov a George Danezis věnují aspektům měření anonymity za použití přístupů pro měření míry informace. Tento článek je v podstatě celý uveden v disertační práci, a proto jej neuvádím samostatně.

V úvodní části je uvedena motivace pro anonymitu, několik definic a základních informací o problému. Následuje výčet několika praktických využití anonymity v různých prostředích (brouzdání na Internetu, elektronické volby, odolnost proti cenzuře). V závěru úvodní části autor uvádí výčet bodů, které budou v práci diskutovány.

V následující kapitole jsou uvedeny další terminologické aspekty a schémata běžně používaná pro vytvoření silného anonymního komunikačního prostředí, základní koncepty jako jsou mixy nebo technologie onion routing, a jsou zde diskutovány různé architektury systémů pro poskytování anonymity. Systémy jsou rozděleny na message-based a real time connection-based.

Další kapitola se zabývá měřením anonymity poskytované zkoumaným systémem (podobný rozbor je uveden v [SDS02]). V úvodu autor diskutuje pojem anonymitní množina a důvody pro její použití jakožto ukazatele míry poskytnuté anonymity. Dále uvádí nevýhody a stinné stránky tohoto přístupu. Následuje zmínka o pojmu entropie jakožto lepšího nástroje pro

určení velikosti anonymitní množiny. Poté je uvedena analýza pool mixu za použití diskutovaných metod a v závěru kapitoly je ve stručnosti uvedena související práce jiných autorů.

V úvodu čtvrté kapitoly autor uvádí vlastnosti (zejména míru poskytnuté anonymity a chování vůči *n-1* útoku), které budou předmětem zájmu při zkoumání různých mixovacích systémů. Zbytek kapitoly je potom věnován výsledkům analýzy zvolených systémů.

Další kapitola se zabývá návrhem nového typu mixu, tzv. binomial mixu.

V šesté kapitole autor prezentuje přístupy při budování mixovacích sítí a formální metody pro popis mixovací sítě. V další části uvádí formální reprezentaci útočníka a toho, co útočník pozoruje v síti. Následuje série formálních vyjádření poskytované míry anonymity a dalších aspektů.

V následující kapitole se autor zabývá analýzou connection-based anonymizačních systémů, definuje model pasivního útočníka a diskutuje dva útoky na tyto sítě. V textu je uvedena analýza těchto útoků a výsledky simulací, které byly provedeny nad tímto typem sítí.

Předposlední kapitola se zabývá systémy poskytujícími odolnost proti cenzuře a uvádí možný návrh takového systému/protokolu.

Závěrečná část shrnuje dosažené výsledky práce a diskutuje možné směry dalšího výzkumu v oblasti systémů pro poskytování anonymity.

Tématicky nezařazené články

[Gol03] se ve svém příspěvku zabývá srovnáním situace na poli anonymity před pěti lety a v současnosti (rok 2003). V článku uvádí, co se za tu dobu změnilo, co stagnovalo, co se podařilo úspěšně dokončit a důvody, proč tomu tak je/bylo. V závěru pak diskutuje možné směry budoucího vývoje v této oblasti.

Autoři článku [MC04] se zabývají pohledem Společných Kritérií (Common Criteria) na definici pojmů anonymity, nespojitelnosti, pseudonymity, nesledovatelnosti a uvádí návrh formálních definic těchto pojmů. Dále se zabývají nespojitelností a uvádějí důvody, proč je tato třída velmi důležitá z hlediska modelů pro vyhodnocování anonymity nějakého sytému. Na základě této argumentace pak uvádějí dva nové pojmy pro anonymitu a pseudonymitu právě s ohledem na nespojitelnost. V další části článku je uveden jiný přístup pro vyjádření nespojitelnosti a anonymity – Freiburg Privacy Diamond a autoři uvádějí svůj návrh modelu popisující vztahy (uživatel – akce) v systému právě s ohledem na kontextové informace, jejichž znalost by mohla vést k úspěšnému „vyprofilování“. Navrhovaný model soukromí se jmenuje PATS. V závěru je uveden jednoduchý příklad a je ilustrován pohled Společných Kritérií, FPD a nově navrhovaného modelu při modelování zvolené situace.

Článek [LRSW00] se zabývá problematikou systémů využívajících pseudonymy. Uživatel v systému může s ostatními komunikovat anonymně, ale v systému existuje možnost prokázat spojitost mezi pseudonymem a skutečnou identitou. Toho lze využít při dohledání zodpovědnosti za provedení nějaké interakce v systému, např. pro potřeby účtování za toto používání. Autoři diskutují situaci v oblasti pseudonymních systémů v době psaní článku a zdůrazňují některé bezpečnostní problémy (Sybil útok, nutnost existence důvěryhodné třetí

strany). V článku podávají formální definici pseudonymního systému, kde jsou uživatelé motivováni k tomu, aby vědomě nesdíleli více pseudonymů a kde je minimální potřeba pro existence důvěryhodného centra. Autoři uvádějí teoretickou konstrukci za použití jednocestných funkcí. Navrhují výkonné a jednoduše implementovatelné schéma.

Použitá literatura:

[Dan03] G. Danezis. Mix-Networks with Restricted Routes. *Privacy Enhancing Technologies Workshop (PET 2003)*, LNCS 2760, pages 1-17. Springer-Verlag, 2003.

[Dan04] G. Danezis. *Designing and attacking anonymous communication systems*. PhD thesis, University of Cambrigde, Computer Laboratory, January, 2004.

[DDM03] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. *2003 IEEE Symposium on Security and Privacy*, May, 2003.

[DS03a] G. Danezis and P. Syverson. Heartbeat Traffic to Counter (n-1) Attacks. *Workshop on Privacy in the Electronic Society (WPES 2003)*, October, 2003.

[Gol03] I. Goldberg. Privacy-Enhancing Technologies for the Internet, II: Five Years Later. *Privacy Enhancing Technologies Workshop (PET 2002)*, LNCS 2482, pages 1-12. Springer-Verlag, 2003.

[LRSW00] A. Lysyanskaya, Ronald R. Rivest, A. Sahai, and S. Wolf. Pseudonym Systems. *SAC'99*, LNCS 1758, pages 184-199. Springer-Verlag, 2000.

[MC04] V. Matyáš and D. Cvrček. On the role of contextual information for privacy attacks and classification. *Privacy and Security Aspects of Data Mining Workshop, IEEE ICDM*, Brighton, UK, November 2004.

[MIX] *Mixminion: A Type III Anonymous Remailer - project development page.* http://www.mixminion.net.

[SD02] A. Serjantov and G. Danezis. Towards an Information Theoretic Metric for Anonymity. *Privacy Enhancing Technologies Workshop (PET 2002)*, LNCS 2482, pages 41-53. Springer-Verlag, 2002.

[SDS02] A. Serjantov, R. Dingledine, and P. Syverson. From a Trickle to a Flood: Active Attacks on Several Mix Types. *Information Hiding Workshop (IH2002)*, LNCS 2578, pages 35-52. Springer-Verlag, 2002.

[Ser03] A. Serjantov. *On the anonymity of anonymity systems*. PhD thesis, University of Cambrigde, Computer Laboratory, March, 2003.

G. Kryptografické eskalační protokoly - část 2.

Jan Krhovják , Fakulta informatiky, MU, Brno
(xkrhovj@fi.muni.cz)

V této části se zaměříme především na popis několika modifikací protokolu (DH)EKE, kterým jsme se zabývali v minulém čísle Crypto-Worldu (a jehož znalost předpokládáme). Dále se také seznámíme s dalším významným zástupcem kryptografických eskalačních protokolů (SPEKE) a s několika jeho modifikacemi.

AEKE

Nevýhodou protokolu EKE je, že jednotlivé strany si musí uchovávat svá sdílená hesla uložená v otevřené podobě. Protokol *AEKE (augmented encrypted key exchange)* [BM93] je takovým rozšířením a vylepšením protokolu DHEKE, které zajišťuje, že si server již uchovává hesla pouze jednocestně zašifrována – někdy je budeme označovat jako *verifikační hodnoty*. Útočník, který by získal přístup k souboru takto zašifrovaných hesel, by sice stále mohl vystupovat jako falešný server, ale nemohl by se jejich přímým použitím vydávat serveru za libovolného uživatele (nejprve by musel provést slovníkový útok).

Implementace AEKE pomocí schémat digitálních podpisů využívá páru soukromý/veřejný klíč, který je na základě hesla jednocestně vygenerován stranou A. Veřejný klíč je jakožto verifikační hodnota sdílen také se stranou B. Protokol začíná ustavením klíče sezení K_S metodou DHEKE, kde je k symetrickému šifrování namísto hesla použit výše zmíněný veřejný klíč. Strana A poté navíc podepíše K_S svým soukromým klíčem, podpis pomocí K_S symetricky zašifruje a zašle straně B. Ta podpis dešifruje a nakonec sdíleným veřejným klíčem ověří jeho korektnost.

Interlock protocol

(A)EKE je vhodnou náhradou za Rivestův a Shamirův *interlock protocol* [RS84]. Ten je navržen tak, aby na komunikačním spoji detekoval aktivní útočníky. Davies a Price v [DP89] navrhli způsob jeho použití také k autentizaci, ale později byl na něj v [BM94] popsán útok. Z [BM94] v této části vycházíme.

Pro použití tohoto protokolu k autentizaci předpokládejme, že strany A a B sdílí dvě tajná hesla P_A a P_B. První fází protokolu je ustavení klíče K mezi těmito stranami, čehož je dosaženo standardní DH metodou. V druhé fázi protokolu pak pomocí tohoto klíče každá strana zašle své zašifrované heslo straně druhé, která jej porovná s uloženým heslem. Aby se zabránilo útoku *man in the middle*, rozdělila se zašifrovaná hesla na dvě části, které si měly jednotlivé strany střídavě vyměnit. Popis druhé fáze protokolu je uveden níže – v hranatých závorkách je vyznačeno číslo příslušné části zašifrovaného hesla:

A		B
	…	
$E_K(P_A)\,\langle 1\rangle$	→	
	←	$E_K(P_B)\,\langle 1\rangle$
$E_K(P_A)\,\langle 2\rangle$	→	
	←	$E_K(P_B)\,\langle 2\rangle$

Pokud by nějaký útočník vstoupil do procesu ustavení klíče[7] a následné autentizace, tak by nemohl dešifrovat první polovinu zprávy přijatou od strany A, dokud nedorazí i její druhá polovina – dešifrováním jedné poloviny zprávy by nezískal správné heslo. To by mu mělo definitivně zabránit zfalšovat první zprávu určenou pro stranu B, čímž se mělo útoku *man in the middle* zabránit.

Předpokládejme však, že má útočník plnou kontrolu nad komunikačními spoji mezi A a B, a že se pokouší vydávat za stranu A. Nejprve si v první fázi protokolu se stranou A ustaví společný šifrovací klíč. V druhé fázi může straně A zaslat libovolné zašifrované heslo a počkat, až mu strana A pošle druhou polovinu svého zašifrovaného hesla. Po jeho dešifrování pak přeruší komunikaci s A (případně znemožní komunikaci mezi A a B) a sám začne komunikaci se stranou B. (Tomuto útoku nezabrání ani případná modifikace protokolu tak, že by druhou fázi dialogu začínala strana B.)

MEKE

V [STW95] je popsána efektivnější varianta protokolu DHEKE, často označovaná jako *MEKE (minimal encrypted key exchange)*. Optimalizací došlo k redukci počtu zasílaných zpráv i prováděných kryptografických operací. Kromě popisu MEKE jsou diskutovány také kryptoanalytické útoky na (A)EKE a obrana proti nim.

Jako podstatné je u (A)EKE zdůrazněno především bezpečné ustavení klíče sezení K_S tak, aby jeho pozdější kompromitace neumožnila slovníkový útok na heslo. Proto by měl být klíč sezení raději vypočítán z původního $K_{S'}$ pomocí kryptografické hašovací funkce jako $K_S = h(K_{S'})$. Protokoly DHEKE i MEKE jsou proti podobnému typu útoku odolné.

DWEKE

I přes svůj precizní návrh jsou mnohé používané protokoly stále náchylné k útokům, které umožňují se znalostí současně používaného hesla získat všechna v budoucnu ustavená hesla – tzv. *password chaining attacks*. Zásadní problém většiny těchto protokolů totiž je, že používají své heslo také k ochraně zpráv, které jsou použity k ustavení nového hesla. Útočník, který zná původní heslo, tak může s jeho pomocí snadno dešifrovat zprávu obsahující nové heslo. Výsledkem odhalení byť jen jediného hesla je pak kompromitace veškeré komunikace daného uživatele.

Původní návrh implementace EKE založené na DH metodě ustavení klíčů (tj. DHEKE) podporuje použití pevně dané hodnoty modulu β, která je buď z důvodu zvýšení rychlosti malá (několik set bitů) a nezamezuje těmto útokům, nebo je dostatečně velká (několik tisíc bitů) a za cenu snížení rychlosti jim zamezuje [BM92]. Protokol *DWEKE (dual-workfactor encrypted key exchange)* [Jas96] je vylepšenou variantou protokolu DHEKE a bez ztráty na rychlosti a efektivitě tomuto typu útoků zamezuje.

Základní myšlenka DWEKE spočívá v použití silnější varianty DHEKE s dostatečně velkými hodnotami modulu β pro ustavení hesla, zatímco pro standardní autentizační zprávy se

[7] Při použití samotné DH metody ustavení klíčů je útok *man in the middle* vždy možný.

i nadále používá z hlediska výkonu efektivnější DHEKE s malými hodnotami β. Návrh a popis celého protokolu je velmi silně ovlivněn snahou o co nejsnazší začlenění do systému Kerberos.

SPEKE

Protokol *SPEKE (simple password encrypted key exchange)* [Jab96] je svým návrhem a implementací velmi blízký protokolu DHEKE. I přes svou podobnost však mají tyto dva protokoly rozdílná omezení a nedostatky.

První fáze SPEKE je opět založena na DH metodě ustavení klíčů, ale namísto běžně používané fixní DH báze (generátoru) α využívá SPEKE funkci *f*, která na základě svého jediného parametru (hesla) vytvoří nějakou bázi pro umocňování (tedy ne nutně generátor příslušné grupy). Výměna prvních dvou zpráv pak vypadá následovně:

A		B
$f(P)^{rA} \bmod \beta$	$\rightarrow$	
	$\leftarrow$	$f(P)^{rB} \bmod \beta$
	…	

Z předaných hodnot si pak nezávisle strana A i B vygeneruje klíč sezení například jako $K_S = f(P)^{rArB} \bmod \beta$ či $K_S = h(f(P)^{rArB} \bmod \beta)$. Funkci f je v případě použití bezpečného prvočísla $\beta = 2\gamma + 1$ doporučeno definovat jako $f(P) = P^{(\beta-1)/\gamma} \bmod \beta = P^2 \bmod \beta$ (tj. $f(P)$ stejně jako 2 je řádu γ), raději než $f(P) = 2^P \bmod \beta$. Protokol je tak odolnější, protože k případnému slovníkovému útoku již nestačí pouze jediný výpočet diskrétního logaritmu. Podrobnější informace vztahující se ke správné volbě f lze nalézt v [Jab96, Jab97].

V druhé fázi SPEKE pak obě strany opět ověřují, zdali ustavení klíče sezení proběhlo korektně. Kromě náhodných čísel lze k tomuto účelu použít také kryptografické hašovací funkce. Strana A nejprve zašle straně B zprávu $h(h(K_S))$ a ta po jejím přijetí a ověření zašle straně A k ověření zprávu $h(K_S)$. Tento přístup je možný, protože klíč K_S vznikl z náhodných dat vygenerovaných oběma stranami a obsahuje také dostatek entropie.

SPEKE narozdíl od DHEKE v první fázi žádným způsobem nešifruje předávané zprávy, což útočníkovi dává možnost omezit prostor klíčů na malou množinu snadno předvídatelných hodnot – tzv. *subgroup confinement attack*. Podívejme se nyní na *man in the middle* verzi tohoto útoku popsanou v [vOW96]. Nechť δ je známý malý prvočíselný dělitel $\beta - 1$, pak útočník umocní předávané hodnoty na $(\beta - 1)/\delta$. Tím se z nich stanou generátory malé podgrupy řádu δ a útočník pak může klíč s pravděpodobností $1/\delta$ uhádnout nebo hrubou silou snadno nalézt. Útok lze také modifikovat tak, že se útočník vydává za jednu ze stran protokolu a druhé straně pošle přímo generátor podgrupy malého řádu.

Použití bezpečných prvočísel počet malých podgrup pouze redukuje a jako protiopatření by tedy mělo být vždy testováno, zdali výsledný klíč do těchto podgrup nepatří (nebo na základě jejich prvků nevznikl).

ASPEKE, BSPEKE a BEKE

Tato tři rozšíření protokolů jsou představena v [Jab97] a podobně jako AEKE zajišťují, že si server uchovává hesla pouze jednocestně zašifrována. ASPEKE je přímočará aplikace technik použitých k vytvoření AEKE na protokol SPEKE. BEKE a BSPEKE nahrazuje poslední část AEKE a ASPEKE dalším kolem DH metody ustavení klíčů, které umožňuje straně B ověřit, že strana A skutečně zná heslo.

Ověření znalosti hesla je u tohoto typu protokolů zcela nezbytné, protože jejich původní část zůstává až na použití jednocestně zašifrované verifikační hodnoty namísto otevřeného hesla naprosto beze změn (strana A si tuto verifikační hodnotu musí vždy z hesla dopočítat) a přímá znalost hesla tedy není prokázána – kdokoliv, kdo zná verifikační hodnotu, by mohl vystupovat za stranu A.

Reference

[BM92] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-based protocols secure against dictionary attacks. In Proceedings *IEEE Computer Society symposium on Research in Security and Privacy*, pages 72–84, May 1992.

[BM93] S. M. Bellovin and M. Merritt. Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise. In Proceedings of the *First ACM Conference on Computer and Communications Security*, pages 244–250, November 1993.

[BM94] S. M. Bellovin and M. Merritt. An attack on the Interlock Protocol When Used for Authentication. In *IEEE Transactions on Information Theory*, volume 40, pages 273–275, January 1994.

[DP89] D. W. Davies and W. L. Price. *Security for computer networks*. John Wiley & Sons, Inc., second edition, 1989.

[Jab96] D. Jablon. Strong password-only authenticated key exchange. In *Computer Communication Review*, volume 26, pages 5–26. ACM SIGCOMM, October 1996.

[Jab97] D. Jablon. Extended Password Key Exchange Protocols Immune to Dictionary Attacks. In *Proceedings of the Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE '97)*, pages 248–255. IEEE Computer Society, June 1997.

[Jas96] B. Jaspan. Dual-workfactor Encrypted Key Exchange: Efficiently Preventing Password Chaining and Dictionary Attacks. *In Proceedings of the sixth USENIX UNIX Security Symposium*, pages 43–50, July 1996.

[RS84] R. L. Rivest and A. Shamir. How to Expose an Eavesdropper. In *Communications of the ACM*, volume 27, pages 393–395, 1984.

[STW95] M. Steiner, G. Tsudik, and M. Waidner. Refinement and Extension of Encrypted Key Exchange. In *Operating Systems Review*, volume 29, pages 22–30. ACM SIGOPS, July 1995.

[vOW96] P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman Key Agreement with Short Exponents. In *Advances in Cryptology – Eurocrypt 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 332–343. Springer, 1996.

H. O čem jsme psali v létě 2000 – 2004

Crypto-World 78/2000

Příloha : 10000.txt , soubor obsahuje prvních 10 000 prvočísel (další informace viz závěr článku "Fermatův test primality, Carmichaelova čísla, bezčtvercová čísla" , str.9) .

Crypto-World 78/2001

Příloha : priloha78.zip (dopis pana Sůvy - detailní informace k horké sazbě)

Crypto-World 78/2002

Crypto-World 7-8/2003

Příloha: "zábavná steganografie" (steganografie.doc)

Crypto-World 7-8/2004

I. Závěrečné informace

1. Sešit

Crypto-World je oficiální informační sešit "Kryptologické sekce Jednoty českých matematiků a fyziků" (GCUCMP). Obsahuje články podepsané autory. Případné chyby a nepřesnosti jsou dílem P.Vondrušky a autorů jednotlivých podepsaných článků, GCUCMP za ně nemá odbornou ani jinou zodpovědnost.

Adresa URL, na níž můžete najít tento sešit (zpravidla 3 týdny po jeho rozeslání) a předchozí sešity GCUCMP, denně aktualizované novinky z kryptologie a informační bezpečnosti, normy, standardy, stránky některých členů a další související materiály:

http://crypto-world.info

2. Registrace / zrušení registrace

Zájemci o e-zin se mohou zaregistrovat pomocí e-mailu na adrese pavel.vondruska@crypto-world.info (předmět: Crypto-World) nebo použít k odeslání žádosti o registraci elektronický formulář na http://crypto-world.info. Při registraci vyžadujeme pouze **jméno a příjmení, titul**, pracoviště (není podmínkou) a **e-mail adresu** určenou k zasílání kódů ke stažení sešitu.

Ke **zrušení registrace** stačí zaslat krátkou zprávu na e-mail pavel.vondruska@crypto-world.info (předmět: ruším odběr Crypto-Worldu!) nebo opět použít formulář na http://crypto-world.info. Ve zprávě prosím uveďte jméno a příjmení a **e-mail adresu**, na kterou byly kódy zasílány.

3. Redakce

E-zin Crypto-World

Redakční práce:	Pavel Vondruška
Stálí přispěvatelé:	Pavel Vondruška
	Jaroslav Pinkava
Jazyková úprava:	Jakub Vrána
Přehled autorů:	http://crypto-world.info/obsah/autori.pdf

NEWS	Vlastimil Klíma
(výběr příspěvků,	Jaroslav Pinkava
komentáře a	Tomáš Rosa
vkládání na web)	Pavel Vondruška

Webmaster Pavel Vondruška, jr.

4. Spojení (abecedně)

redakce e-zinu	ezin@crypto-world.info ,	http://crypto-world.info
Vlastimil Klíma	v.klima@volny.cz ,	http://cryptography.hyperlink.cz/
Jaroslav Pinkava	jaroslav.pinkava@pvt.cz ,	http://crypto-world.info/pinkava/
Tomáš Rosa	t_rosa@volny.cz ,	http://crypto.hyperlink.cz/
Pavel Vondruška	pavel.vondruska@crypto-world.info ,	http://crypto-world.info/vondruska/index.php
Pavel Vondruška,jr.	pavel@crypto-world.info ,	http://webdesign.crypto-world.info
Jakub Vrána	jakub@vrana.cz ,	http://www.vrana.cz/