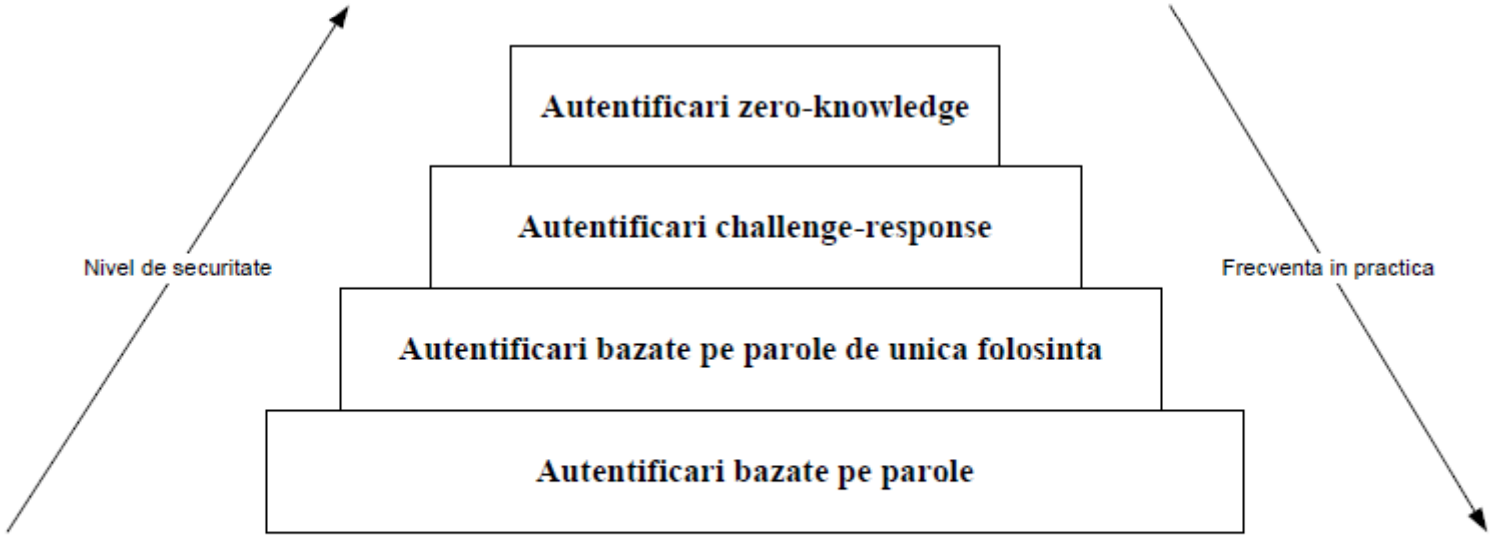
Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013. "Investește în oameni!"

Cap. 5.2. Protocoale de autentificare in sisteme de operare si sisteme bancare (ATM, NTLM, MS-CHAP, ISO TPMA).

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.
"Investește în oameni!"

Protocoale de autentificare



Standarde in criptografie

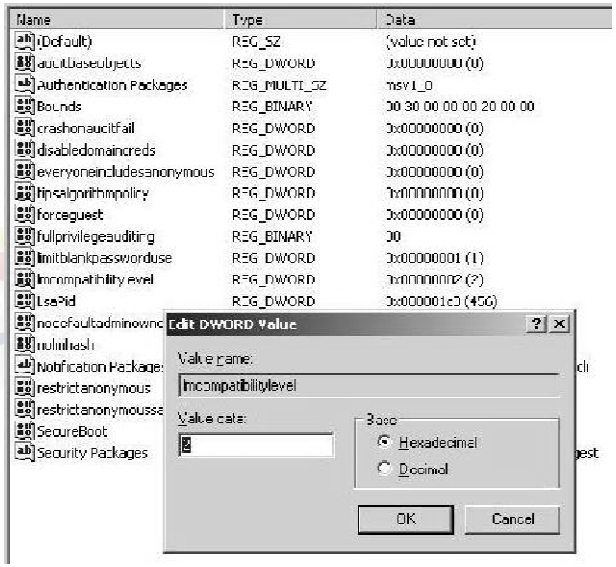
- Discutate pe parcursul cursurilor anterioare: SHA2 (256, 384, 512), AES (128, 194, 256), RSA/OAEP

- Alte standarde relevante
  - RSA/PKCS#1 (Public-Key Cryptography Standards) versiunea 2.1 din 2002 http://tools.ietf.org/html/rfc3447 (se doreste aceeasi proprietate de la OAEP: rezistenta IND/NM-CCA2)
  - DSA (Digital Signature Algorithm) (aceeasi idee ca la semnatura ElGamal) versiunea 3 din 2009 in FIPS PUB 186-3 http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

MS-CHAP si NTLM

- MS-CHAP (Challenge-handshake authentication protocol) si NTLM (NT LAN Manager) sunt standarde *de facto* in sistemele de operare Microsoft
- Utilizate pentru autentificare utilizatorilor la acces remote catre fisiere, imprimante etc.
- MS-CHAP are 2 variante si NTLM 5 (inclusiv una bazata pe LM-Hash, i.e. DES)
- Toate cele 5 variante de NTLM au la baza un challenge-response in 3 pasi
- Alegerea variantei depinde e valoarea *lmcompatibility* din Registrul Win

1. *Client → Server : Type 1 Message*
2. *Server → Client : Type 2 Message (includes the 64 bit challenge from the server)*
3. *Client → Server : Type 3 Message (includes the response from the client)*



Variante NTLM

- Trei variante bazate pe DES
- Doua variante bazate pe MD5

$$Client \rightarrow Server:$$
$$DES_{K1}(challenge) \,||\, DES_{K2}(challenge) \,||\, DES_{K3}(challenge)$$

$$Client \rightarrow Server:$$
$$HMAC-MD5_{HMAC-MD5_{MD4(password)}(user||target)}(challenge \,||\, clientNonce)$$

MS-CHAP v2 si NTLM v2 Session

- Ambele (ilustrate in figura dar si toate celelalte variante) sunt vulnerabile la cautari off-line exhaustive ale parolelor
- Pentru detalii vezi: B. Groza, A. Alexandroni, I. Silea , V. Patriciu, On the security of some authentication mechanisms from Windows, Buletinul Stiintific al Universitatii Politehnica din Timisoara, Seria Automatica si Calculatoare, ISSN 1224-600X, 2008.

MS-CHAP v2

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_B$
3. $A \rightarrow B : N_A, H(k_{AB}, N_A, N_B, A)$
4. $B \rightarrow A : H(k_{AB}, N_A)$

NTLMv2-Session

1. $B \rightarrow A : N_B$
2. $A \rightarrow B : N_A, H(k_{AB}), H'(N_A, N_B))$
3. $B \rightarrow A : H(k_{AB}, H'(N_A, N_B)), H'(N_A, N_B)$

Encrypted Key Exchange (EKE)

- Propus de Bellovin si Merritt in 1992, exista in diverse variante (folosind Diffie-Hellman, RSA, 3 participanti etc.)
- Primul protocol de schimb de cheie autentificat bazat pe parole **rezistent la cautari exhaustive** (ale parolei)

$$1.A \rightarrow B : A, E_{pwAB}(pk)$$
$$2.B \rightarrow A : E_{pk}(k)$$
$$3.A \rightarrow B : E_k(N_A)$$
$$4.B \rightarrow A : E_k(N_A, N_B)$$
$$5.A \rightarrow B : E_k(N_B)$$

- Protocolul se incheie cu succes daca nonce-urile trimise de A si B sunt verificate in raspunsurile primite
- Atacuri de cautare exhaustiva asupra parolei nu poti fi facute (doar daca cheia publica pk are o structura speciala)

ISO/IEC 9798-3 Three-Pass Mutual Authentication

- Relizeaza autentificare mutuala folosind primitive asimetrice

$$1.B \rightarrow A : N_B$$

$$2.A \rightarrow B : Cert_A, TokenAB = N_A \,||\, N_B \,||\, B \,||\, Sig_A\left(N_A \,||\, N_B \,||\, B\right)$$

$$2.B \rightarrow A : Cert_B, TokenBA = N_B \,||\, N_A \,||\, A \,||\, Sig_B\left(N_B \,||\, N_A \,||\, A\right)$$

- Varianta initiala (se observa semnarea de catre B a unui nonce nou, masura care se credea ca imbunatateste securitatea deoarece A nu stie apriori ce mesaj va semna B)

$$1.B \rightarrow A : N_B$$

$$2.A \rightarrow B : Cert_A, TokenAB = N_A \,||\, N_B \,||\, B \,||\, Sig_A\left(N_A \,||\, N_B \,||\, B\right)$$

$$2.B \rightarrow A : Cert_B, TokenBA = N_B' \,||\, N_A \,||\, A \,||\, Sig_B\left(N_B' \,||\, N_A \,||\, A\right)$$

Atacul lui Wiener (Atacul Canadian)

- Descoperit de Wiener, membru al partii canadiene din comitetul ISO
- Pasul 1: Adv pretinde ca este B si incepe o conversatie cu A

$$1.Adv(B) \rightarrow A : N_{adv}$$

$$2.A \rightarrow Adv(B) : Cert_A, TokenAB = N_A \,||\, N_{adv} \,||\, B \,||\, Sig_A(N_A \,||\, N_{adv} \,||\, B)$$

- Pasul 2: Adv pretinde ca este A si incepe o conversatie cu B

$$1'.Adv(A) \rightarrow B : N_A$$

$$2'.B \rightarrow Adv(A) : Cert_B, TokenBA = N_B \,||\, N_A \,||\, A \,||\, Sig_B(N_B \,||\, N_A \,||\, A)$$

- Pasul 3: Adv incheie cu succes conversatia inceputa cu A in pasul 1 folosind mesajul primit de la B in pasul 2

$$3.Adv(B) \rightarrow A : Cert_B, TokenBA = N_B \,||\, N_A \,||\, A \,||\, Sig_B(N_B \,||\, N_A \,||\, A)$$

- Final: A crede ca B a inceput cu el o conversatie si a incheiat-o cu succes, in timp ce B crede ca A a inceput cu el conversatia si asteapta in continuare mesajul 3

Autentificarea in Windows

- Bazata pe parole, stocate criptat in SAM (Security Accounts Manager)

$$c:\windows\system32\config\SAM$$

- Diverse softuri intitulate pwdump pentru a extrage parola criptata (vezi http://en.wikipedia.org/wiki/Pwdump)

- Parolele sunt criptate sub forma

$$DES_{KD1(password)}(\text{"KGS!@\#\$\%"}) \,||\, DES_{KD2(password)}(\text{"KGS!@\#\$\%"})$$

Problema 1: Utilizarea DES

$$DES_{KD1(password)}(KGS!@\#\$\%)||DES_{KD2(password)}(KGS!@\#\$\%)$$

- DES este inlocuit ca standard inca din 2001
- Hardware criptografic capabil sa sparga DES in cateva zile (Copacobana in medie 3.5 zile)
- DES are o cheie de 56 biti, in timp ce recomandarile curente NIST sunt

Bits	AES	RSA	ECC	LifeTime
80	x	1024	160-223	Until 2010
112	x	2048	224-255	Until 2030
128	128	3072	256-383	After 2030
192	192	7680	384-511	x
256	256	15360	512+	x

Problema 2: Fara salt

$$DES_{KD1(password)}(KGS!@\#\$\%)||DES_{KD2(password)}(KGS!@\#\$\%)$$

- Salt este o procedura elementara de intarire a parolelor
- Utilizata in toate distributiile UNIX
- Absenta saltului face posibila atacuri de tip dictionar precalculat
- Exemplu: tabele rainbow pentru a sparge toate parolele de 1 - 14 charactere lower/upper/numbere si caractere speciale se pot cumpara la http://www.rainbowtables.net/products.php#LM

Problema 3: Divizarea parolei !?

$$DES_{KD1(password)}(\text{"KGS!@\#\$\%"}) \,||\, DES_{KD2(password)}(\text{"KGS!@\#\$\%"})$$

- Daca parola este de 14 charactere sau mai scurta este divizata in 2 parole care sunt folosite ca si chei pentru cele doua criptari DES
- Ambele parti ale parolei pot fi atacate independent
- Efecte:
  - a sparge 14 caractere e aproape la fel de usor cu a sparge 7
  - o parola de 8, 9, 10, 11 (discutabil 12 si 13) e mai usor de spart decat una de 7

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.
"Investește în oameni!"

Solutii

- Nota Microsoft pentru dezactivarea LM Hash
  http://support.microsoft.com/kb/299656

- Parole de peste 14 caractere sunt ascunse cu MD5 in loc de DES (ceva mai sigur)

Bibliografie suplimentara

- Discutie despre autentificare in Windows

B. Groza, A. Alexandroni, I. Silea , V. Patriciu, On the security of some authentication mechanisms from Windows, Buletinul Stiintific al Universitatii Politehnica din Timisoara, Seria Automatica si Calculatoare, ISSN 1224-600X, 2008.

- si cod sursa detaliat pentru toate protocoalele de autentificare din Windows:

E. Glass, "The NTLM Authentication Protocol and Security Support Provider", http://davenport.sourceforge.net/ntlm.html

Parole in UNIX

- Din nou fisiere criptate, dar de data asta cu salt si o functie mai intensa
- Stocate traditional in */etc/passwd*
- Mutate mai nou in */etc/shadow* (accesibil doar de root)
- Criptate folosind o modificare a MD5 (anterior de folosea DES) aplicata pe parola si salt (comanda *crypt*, vezi *man crypt*)
- Exemplu, fisier *passwd*

  x:x:501:501:x:/home/x:/bin/bash
  Alice:x:502:502:Alice:/home/Alice:/bin/bash
  Bob:x:503:503:Bob:/home/Bob:/bin/bash
- Exemplu, fisier *shadow*

  x:$1$7lSmIrJ4$nFh23Pb8xK8xW7VnHjOGm1:13338:0:99999:7:::
  Alice:1VSxlZUhg$bGH7FRq.5jhcCsDPS.Zdl1:13338:0:99999:7:::
  Bob:1ZRw9H/6L$0SJjGvTxJ2UrBrR2bQ7gA/:13338:0:99999:7:::

Securitate in sisteme bancare (bancomate)

- ATM (Automatic Teller Machines) sunt cel mai larg raspandit si solicitat dispozitiv al comertului
- Aceeasi tehnologie se foloseste si in POS (EFTPOS – Electronic Funds Transfer at the Point of Sales)
- Modul de functionare uzual este: un cod PIN este folosit pentru a cripta contul, valoare este transformata in zecimal si trunchiata rezultat denumit PIN natural. Un offset se poate adauga si se obtine PIN-ul utilizatorului

Account number N (on the mag stripe):	8807012345691715
PIN key KP:	FEFEFEFEFEFEFEFE
Result of DES $\{N\}_{KP}$:	A2CE126C69AEC82D
$\{N\}_{KP}$ decimalized:	0224126269042823
Natural PIN:	0224
Offset:	6565
Customer PIN:	6789

poza din: Ross Anderson, Security Engineering: A guide to Building Dependable Distributed Systems, John Wiley & Sons, 2001