Česká společnost uživatelů otevřených systémů EurOpen.CZ
Czech Open System Users' Group
www.europen.cz

XXX. konference – XXXth conference

Sborník příspěvků
Conference proceedings

Hotel MALEVIL, s. r. o., Jablonné v Podještědí
20.–23. května 2007

Sborník příspěvků z XXX. konference EurOpen.CZ, 20.–23. května 2007

Plzeň 2007. První vydání.
Editor: Vladimír Rudolf, Jiří Felbáb
Sazba a grafická úprava: Ing. Miloš Brejcha – Vydavatelský servis, Plzeň
e-mail: servis@vydavatelskyservis.cz

Tisk: TYPOS – Digital printing, spol. s. r. o.
Podnikatelská 1 160/14, Plzeň

Příspěvky neprošly redakční ani jazykovou úpravou.

ISBN 978-80-86583-12-9

Obsah

Jan Okrouhlý
Tutorial – Co obnáší současný standard platebních karet 5

Ladislav Lhotka
History of Unix 23

Peter Sylvester
Identity and Authorization multi-organisation contexts 37

Ralf Knöringer
Identity Management – Compliance and Cost Control 49

Jiří Bořík
Identity Management – ORION implementation 59

Marta Vohnoutová
Identity a Access Manager
řešení jednotné správy uživatelů a uživatelských oprávnění pro heterogenní prostředí 65

Jakub Balada
Implementation of workflow for identity management 75

Libor Dostálek
Reverzní proxy neboli Access manager 83

Martin Čížek
Implementace Access Managementu s open source produkty 95

Aleksej Jerman Blazic
Trusted Archive Authority – Long Term Trusted Archive Services ... 107

Aleksej Jerman Blazic
Long Term Archiving Implementation – Slovenian Experience with Long Term Archiving 121

Peter Sylvester
The French Amdinistration's profile for XAdES 131

Peter Sylvester
Electronic notary services 135

Michal Hojsík
Jak je to se sílou algoritmů pro výpočet hash 145

Petr Břehovský
Jak opravdu anonymně vystupovat na Internetu 155

Radoslav Bodó
Kovářova kobyla... 163

Michal Švamberg
... už nechodí bosa 175

Tutorial – Co obnáší současný standard platebních karet

Jan Okrouhlý

E-mail: jan.okrouhly@hp.com

Abstrakt

Kartu s čipem již má asi téměř každý, co ale její výroba a používání obnáší? Lehký úvod do standardu čipových karet ISO 7816 a do EMV standardu pro platební aplikace bude následovat praktická ukázka komunikace s čipovou kartou, včetně předvedení platební transakce.

Hlavní důraz bude kladen na přiblížení reálného životního cyklu aplikace od přípravy prostředí, výběru a testování čipu, přes výrobu a certifikace až po reálné použití. Jaké jsou další aplikační možnosti využití čipu, bezpečnostní aspekty platebních karet v praxi a kam dále hodlá vývoj standardů kráčet.

Proč čipové karty

Magnetický pásek má nízkou kapacitu a neposkytuje dostatek flexibility ani kapacitu pro nové aplikace. Smart karty[1] přinášejí nové možnosti využití jak v podobě sloučení platební aplikace se zákaznickým bonus programem, tak při použití pro jednotliví nezávislé aplikace a nebo naopak jako kombinaci více různých aplikací na jedné čipové kartě.

Historicky sahají první bezpečnostní požadavky na karty vybavené mikroprocesorem až do konce 70. let minulého století. Smart karty byly původně vyvinuty za účelem snížení bezpečnostních problémů a dosud poskytují dostatečně bezpečné řešení pro mnoho aplikací jakými jsou typicky platební, identifikační či autorizační aplikace, omezující fyzický přístup, či logický přístup k systémům nebo datům. Platby[2] smart kartami jsou pro redukci podvodů založeny na principech a funkcích specifikovaných EMV (Europay MasterCard Visa).

[1] Zkráceně též označovány PC/SC, což je též název příslušné pracovní skupiny (hlavními členy jsou Gemalto, Infineon, Microsoft a Toshiba).

[2] Pro snížení telekomunikačních výdajů jsou tyto platby zpracovávány v centrálách, které jsou rozmístěny po Evropě, Latinské Americe a Asii.

Hlavní přínosy čipových karet:

- Kapacita paměti čipu poskytuje prostor pro další software, s možností doplnit další softwarové bezpečnostní prvky, a také prostor pro nové aplikace
- Obtížná výroba kopie. Získání stejného čipu a operačního systému včetně získání dat, která jsou šifrována a dostupná jen čipu.
- Vylepšení procesu rozhodování čtečky karet. Podpora online i offline ověření pravosti karty a možnost použití náročnějších šifer, což přináší lepší ochranu na bezobslužných terminálech.
- Ověření identity majitele raději znalostí PIN než podpisem, které může být doplněno i o biometrické informace (zatím se nepoužívá při platbách, ale běžně např. pro přístup k datům).
- Možnost reakce na nově vznikající požadavky – doplnění risk managementu, či aktualizace konfigurace aplikací.

Standard ISO 7816

Mezinárodní standard pro elektronické identifikační karty, zvláště pak smart karty, spravovaný společně ISO (International Organization for Standardization) a IEC (International Electrotechnical Commission).

Skládá se z mnoha rozdílných částí, z nichž pro potřeby současných platebních karet vystačíme s těmito čtyřmi základními:

- 7816-1: Fyzické charakteristiky;
- 7816-2: Rozměry a umístění kontaktů;
- 7816-3: Elektrické charakteristiky a třídy pro čipové karty pracující s 5 V, 3 V a 1,8 V;
- 7816-4: Organizace, bezpečnost a příkazy pro datovou komunikaci.

První tři části jsou podstatné spíše pro výrobu a testování karet a čipů. Při orientaci na platební aplikace bude pro nás z hlediska obsahu nejpodstatnější část 7816-4, která specifikuje:

- Obsah párů příkaz–odpověď probíhajících skrz interface;
- Prostředky pro získávání datových elementů a datových objektů z karty;
- Struktury a obsah bytů historie popisující operační charakteristiku karty;

- Struktury pro aplikace a data na kartě z pohledu rozhraní při zpracování příkazů;
- Přístupové metody k souborům a datům na kartě;
- Bezpečnostní architektura definující přístupová práva k souborům a datům na kartě;
- Prostředky a mechanismy pro identifikaci a adresaci aplikací na kartě;
- Metody pro secure messaging;
- Metody přístupu k algoritmům zpracovávaným kartou (nepopisuje tyto algoritmy).

Tato specifikace nepokrývá interní implementaci, ani její okolí, a je nezávislá na technologii fyzického rozhraní. Používá se pro karty s přístupem jednou či více metodami, tj. kontakty, těsnou indukční vazbou či vysokofrekvenčním přenosem.

Standard EMV 2000

Jedná se o specifikaci čipových karet pro platební systémy aktuálně spravovaný společností EMVCo, jejímiž členy jsou platební asociace JCB, MasterCard a Visa (dále jen asociace).

EMVCo byla zformována v únoru 1999 z Europay International, MasterCard International a Visa International za účelem správy, udržování a vylepšování specifikací EMV. Europay akvizicí pohltila v roce 2002 asociace MasterCard a v roce 2005 se k organizaci připojila JCB International. Společnost EMVCo je rovněž zodpovědná za proces typových zkoušek platebních terminálů. Testování zajišťuje použitelnost platebních karet v různých platebních systémech za dodržení specifikací EMV.

V současné době je standard ve verzi EMV 4.1 a skládá se ze čtyř celků:

- **Book 1** specifikuje aplikačně nezávislé požadavky na čipové karty a platební terminály (především elektromechanické charakteristiky jako jsou rozměry, či úrovně napájení), přenosové protokoly (znakový $T = 0$ a blokový $T = 1$), mechanismus volby aplikací a strukturu souborů a příkazů;
- **Book 2** specifikuje bezpečnostní požadavky, především mechanismus offline autentizace, šifrování PINů, generování aplikačních kryptogramů a management kryptografických klíčů;
- **Book 3** specifikuje požadavky na jednotlivé aplikace, např. definice jednotlivých APDU (Application Protocol Data Unit) příkazů;

- **Book 4** specifikuje povinné, doporučené a volitelné požadavky na platební terminály pro zajištění kompatibility s platebními kartami.

V praxi se v současnosti nejčastěji setkáváme s platebními kartami dle standardu EMV 2000, odpovídající EMV 4.0, příp. obsahujícími implementaci aktualizací a dodatků k této verzi a tím se funkčně blížícími k verzi 4.1.

Specifikace EMV je postavena především na ISO/IEC 7816-4, ale z důvodů standardizace čerpá též z dalších specifikací jakými jsou např. tyto:

- **ISO 639** definující reprezentaci jmen a jazyků;
- **ISO/IEC 7811** definujících umístění a provedení embosingu;
- **ISO 8859** pro osmibitové kódování grafických znakových sad.

Komunikace na aplikační úrovni

Krok v aplikačním protokolu se skládá ze zprávy s příkazem, jejího zpracování v příjemci a odeslání příslušné odpovědi.

APDU (Application protocol data unit) obsahují buď příkaz nebo odpověď. Obě součásti páru příkaz–odpověď mohou obsahovat data, což sumárně dává čtyři možné typy zapouzdření.

Číslování základních typů zapouzdření:

Zapouzdření	Data v příkazu	Očekávaná data v odpovědi
1	Žádná	Žádná
2	Žádná	Data
3	Data	Žádná
4	Data	Data

Rámec každého příkazu musí mít definováno, dle kterého zapouzdření má být sestaven[3].

Příkazová APDU

ISO 7816 definuje příkazovou APDU jako povinnou hlavičku a podmíněné tělo proměnné velikosti.

[3]Pokud toto při implementaci příkazů nedůsledně implementováno, těžko se taková chyba odhalí a může se následně v praxi náhodně projevit v podobě různých problémů.

Hlavička se skládá ze 4 bytů:

- CLA – třída (zda je rozsah příkazu a odpovědi dle ISO 7816, zda je použit secure messaging atp.)
- INS – instrukce (vlastní příkaz typu SELECT FILE, READ RECORD, GET DATA, VERIFY atp.)
- P1 – parametr 1
- P2 – parametr 2

Struktura příkazové APDU:

Hlavička	**Tělo**
CLA INS P1 P2	[L_c pole] [Data] [L_e pole]

Počet bytů v datovém poli je dáno L_c. Množství očekávaných bytů v odpovědi udává L_e. Nulové L_e znamená maximální počet požadovaných dat.

L_e i L_c mohou být za jistých okolností i dvou-bytové hodnoty a tak při parsování příkazů může reálně nastat až sedm možných variant předání informací a aplikace musí být schopna náležitě je zpracovat.

APDU odpovědi

Odpověď se dle ISO 7816 skládá z podmíněného těla proměnné délky a povinného dvou-bytového zakončení.

Struktura APDU odpovědi:

Tělo	**Zakončení**
[Data]	SW1 SW2

Zakončení kóduje stav příjemce po zpracování páru příkaz–odpověď. Význam kombinací SW1 a SW2 detailně popisuje ISO 7816. Specifikace příslušné aplikace pak udávají jakých konkrétních hodnot má přesně zakončení nabívat v daném případě, resp. stavu aplikace.

Organizace dat na kartách – souborová struktura

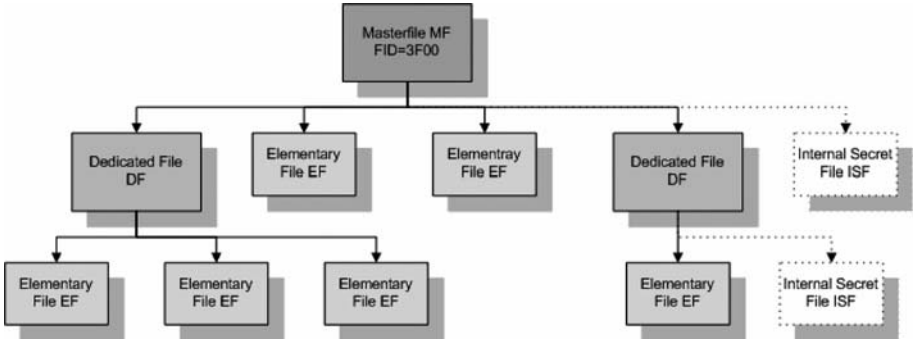
ISO 7816 podporuje dvě kategorie souborů:

- **Dedicated File** (DF)
- **Elementary File** (EF)

Logická organizace dat spočívá v hierarchické struktuře DF.

DF označovaný jako Master File (MF) reprezentuje kořen souborového systému. Po resetu karty automaticky vybrán a jeho přítomnost na čipu je povinná. Ostatní DF jsou nepovinné.

Příklad organizace souborů:



EF jsou principiálně dvou typů:

- **Internal EF** – ISF jsou interpretovány pouze managementem karty[4]
- **Working EF** – pro ukládání dat neinterpretovaných kartou

Základní struktury souborů EF:

- **Transparentní** – nemají vnitřní strukturu a lze je číst po bytech, či blocích
- **Záznamový** – sekvence jednotlivých adresovatelných záznamů

Pro záznamové EF je definován počet záznamů a způsob jejich organizace (lineární, či kruhový).

Karta musí podporovat alespoň jeden ze způsobů strukturování záznamů:

- **Transparent EF** – viz výše
- **Linear Fixed EF** – stejně dlouhé záznamy dat
- **Linear Variable EF** – různě dlouhé záznamy dat
- **Cyclic EF** – obdoba Linear Fixed, záznam číslo 1 vždy ukazuje na poslední zápis atd.

[4] V daném MF, či DF existuje vždy pouze jeden Internal Secret File.

V praxi se lze setkat s dalšími rozšířeními jako je např. *Compute* EF, což je obdoba *Cyclic* EF, ale každý záznam obsahuje několik bytů konkrétního významu, které mohou být pouze inkrementovány či dekrementovány aplikací a tyto záznamy lze jinak jen číst.

Příklad struktury záznamu *Compute* EF:

MSB		LSB	CS2	CS1
Hodnota čítače – 3 byty			Checksum – 2 byty (řízeno interně)	

Metody přístupu k souborům

Každý soubor má tzv. File Identifier (FID) kódovaný do dvou bytů. MF má fixní FID '3F00'. EF a DF pod každým daným DF musí mít jednoznačné FID. Hodnoty FID '3FFF' a 'FFFF' jsou rezervovány.

Soubory lze adresovat prostřednictvím:

- FID identifikátoru – v aktuálním DF (nemusí být vždy jednoznačné);
- Cestou – složením jednotlivých FID za sebe (absolutní cestu lze zadat s pomocí rezervované hodnoty '3FFF', která označuje kořen cesty);
- Krátkým EF identifikátorem (SFI) – adresace 5 bity v daném DF (0 znamená aktuálně vybraný EF);
- Jménem DF – Application ID (AID) kódováno 1 až 16 byty a musí být jednoznačné v rámci celé karty.

Výběr souboru se provádí příkazem SELECT, jednotlivými parametry příkazu lze získat podrobné informace o vybraném souboru z File control parameters (FCP), File management data (FMD), či File control information FCI. Podrobnosti viz ISO 7816-4.

Komunikace s čipovými kartami

Po zasunutí karty do čtečky jí je terminálem zaslán signál Reset. Na ten karta musí odpovědět zasláním sekvence Answer To Reset (ATR).

Z ATR[5] terminál zjistí způsob kódování logických jedniček a nul, zda karta bude komunikovat protokolem T=0 nebo T=1 a případné další základní parametry komunikace.

[5] Na soft reset opět karta musí odpovědět ATR.

V danou chvíli karta musí poskytovat alespoň tyto základní služby:

- **Card identification service** – Služba umožňující identifikovat kartu rozhraní – viz ATR;
- **Application selection service** – Služba umožňující zjistit, jaká aplikace je na kartě aktivní a implicitní či explicitní výběr (přes jméno DF) a spuštění aplikace na kartě;
- **Data object retrieval service** – Tato služba umožňuje získat standardními přístupovými metodami datové objekty definované ISO/IEC 7816, např. soubor DIR s cestou '3F002F00', či ATR s cestou '3F002F01';
- **File selection service** – Služba pro SELECT nepojmenovaných DF a EF;
- **File I/0 service** – Služba umožňující čtení dat uložených v EF souborech.

Aplikace jsou po resetu ve stavu IDLE[6]. Přechody do dalších stavů jsou přesně definovány a ovlivňují je různé události (zadání PIN, vzájemná autentizace, nepovelená operace atp.). Po výběru aplikace se dostává do stavu SELECTED a kontrolu nad příkazy přebírá aplikace. Aplikace může například interpretovat vlastní APDU příkazy zapadající do definovaných typů rámců, definovat vlastní stavový diagram aplikace a řídit přechody mezi jednotlivými stavy. Po ukončení opět převede kartu do stavu IDLE.

Realizace platební transakce

Platební aplikace je přesně specifikována EMV a ještě více zpřesněna asociacemi, jako např. Visa, MasterCard, či JCB (definují např. minimální požadavky na implementaci bezpečnosti, použité funkce, algoritmy atp.). Specifikace asociací obsahují krom povinných nastavení také volitelné konfigurace, doporučená nastavení a podmínky použití některých nastavení a funkcí. Rovněž definují způsob provedení transakce, tj. chování jak karet, tak i terminálů. V několika následujících bodech bude přiblížen průběh typické EMV transakce.

1. Výběr platební aplikace

Jak bylo uvedeno výše, na kartě se může vyskytovat jedna implicitně vybíraná aplikace a nebo jich tam může být více[7]. V tom případě je nutno zajistit explicitní výběr aplikace.

[6] Při implicitním výběru aplikace bude ve stavu SELECTED.

[7] Záleží též na velikosti paměti karty.

Teoreticky na kartě může být dokonce více různých platebních aplikací, ale v praxi se to zatím příliš nepoužívá, neboť by potisk povrchu takové karty byl zavádějící.

Existují dva způsoby, jak terminál vybere platební aplikaci:

- Načte z karty soubor se seznamem aplikací;
- Periodicky se snaží vybrat všechny jím podporované AID platebních aplikací

Nalezené nezablokované (pozná se dle SW1 a SW2) aplikace terminál setřídí dle jejich priorit a podle konfigurace karty buď implicitně sám vybere tu s nejvyšší prioritou, nebo klientovi umožní explicitně ji vybrat.

2. Iniciace transakce

Po úspěšném výběru aplikace požádá terminál kartu příkazem Get Processing Options (GPO) o zaslání informací o datech aplikace a jí podporovaných funkcích. Zároveň kartě může předat požadované informace, které specifikovala již při odpovědi na SELECT prostřednictvím Processing Options Data Object Listu (PDOL) v FCI.

Jako odpověď na GPO zašle karta Application Interchange Profile (AIP) a Application File Locator (AFL). AIP definuje jaké karta podporuje metody pro ověření pravosti dat (viz dále), způsob autentizace, zda bude použit terminal risk management atp.

AFL udává, ve kterých záznamech jsou uložené informace, které terminál bude potřebovat při zpracování transakce (např. jméno, či číslo účtu klienta) a které z nich jsou zahrnuty do ověření pravosti (např. číslo účtu klienta, či tabulka Application Usage Control udávající, k jakým typům plateb lze aplikaci použít).

3. Načtení a ověření aplikačních dat

Terminál následně načte záznamy definované AFL a ověří jejich pravost. Načtená data náležitě interpretuje ve fázi Processing Restrictions, kde testuje verze aplikace, datum její použitelnosti a zda ji může pro požadovaný druh platby za zboží, za služby či pro hotovostní operace, použít a to včetně kontroly, zda je jedná o transakci domácí nebo zahraniční[8]. Je-li vše v pořádku, následuje ověření držitele karty.

[8]Každý terminál má kromě měny též definovánu příslušnost ke státu jeho poskytovatele, příp. obchodníka

4. Ověření držitele karty

Terminál z obdrženého seznamu aplikací podporovaných metod ověření držitele (Cardholder Verification Method, resp. CVM List) vybere odpovídající metodu a tu realizuje. Dle EMV 2000 tyto metody mohou být:

- **Offline Plaintext PIN**
- **Offline Enciphered PIN**
- **Online PIN**
- **Signature**
- **Offline Plaintext PIN and Signature**
- **Offline Plaintext PIN and Signature**
- **No CVM required**

Výběr metody terminálem je závislý na částce a typu transakce, možnostech terminálu a pravidlech, která jsou nedílnou součástí CMV Listu.

5. Terminal Risk Management

Po načtení dat viz bod 3, terminál může zkontrolovat, zda se karta nenachází na seznamu blokovaných karet. Dále testuje, zda částka transakce nepřevyšuje interní limit daného terminálu, v opačném případě transakce musí být zpracována online. Rovněž obsluha terminálu má možnost vynutit online zpracování transakce, např. pokud se jí zdá chování či vzezření majitel karty podezřelé. Některé transakce však buď sám terminál náhodně zpracovává online a nebo se tak rozhodne na základě překročení daného limitu počtu kartou realizovaných offline transakcí bezprostředně po sobě[9]. V praxi také nové karty mohou vyžadovat zpracovat první transakce online.

Na základě vyhodnocení uvedených kritérií terminál vyšle kartě žádost o vygenerování příslušného typu aplikačního kryptogramu (AC).

Typy aplikačních kryptogramů:

- TC (Transaction Certificate) – provedení transakce offline
- ARQC (Authorization Request Cryptogram) – provedení online transakce
- AAC (Application Authentication Cryptogram) – zamítnutí transakce

[9] Pokud hodotu čítače karta terminálu poskytuje

6. Rozhodování karty

Karta obdrží od terminálu žádosti o generování příslušného typu AC a zároveň kartou požadovaná data poskytující nezbytné informace o prováděné transakci.

Karta provede vlastní risk management (obdobně jako terminál může provádět různé testy na počet offline provedených transakcí, jejich finanční objem atp.) a buď příslušný požadovaný kryptogram vygeneruje nebo transakci zamítne. Bylo-li terminálem požadováno offline schválení (TC) a karta se rozhodne, že schválení nechá na vydavateli (bance), pak místo TC vygeneruje ARQC.

7. Online Processing/Issuer Authentication

Pokud karta vygenerovala ARQC, putuje tento kryptogram společně s dalšími informacemi o transakci k vydavateli. Vydavatel transakci buď přijme nebo zamítne a zašle kartě žádost o generování kryptogramu TC nebo AAC. Zpráva od vydavatele obsahuje zašifrované kontrolní údaje, které dokáže ověřit pouze karta a tím je rozhodnutí vydavatele zabezpečeno – jedná se o tzv. issuer authentication data.

Pokud terminál není schopen online funkce, může požadovat, aby karta transakci offline přijala či zamítla s tím, že jí signalizuje nemožnost online spojení (issuer authentication data kartě v tom případě nezašle).

8. Completion

Karta po obdržení druhé žádosti o generování kryptogramu TC opět provede risk management (při požadavku o AAC, není co řešit). Na základě výsledku buď transakci přijme (vygeneruje TC) a nebo zamítne vygenerováním AAC. Pokud proběhla úspěšně autentizace vydavatele, tj. transakce proběhla online, dochází v této fázi většinou rovněž k vynulování čítačů počtu a hodnoty transakcí offline offline a několika dalších stavových bitů, které se úspěšně přenesly do banky při kroku 7.

9. Issuer Script Processing

Během kroku 7 může karta obdržet od terminálu tzv. issuer skript. Ten může obsahovat speciální příkazy určené například pro změnu PIN, odblokování PIN, blokaci aplikace, atp. Skripty zasílá kartě vydavatelská banka a jednotlivé příkazy jsou symetrickou šifrou zajištěny proti modifikaci – provádí se tzv. Message Authentication Code (MAC). Použité klíče jsou unikátní pro každou kartu a nikdy neopouští kartu ani banku. Citlivé informace, jako např. měněný PIN jsou navíc symetricky šifrovány prostřednictvím dalšího klíče unikátního pro každou kartu.

Skripty mohou být dvou druhů – critical a non-critical. První typ je vykonáván ještě před generováním druhého kryptogramu a výsledek zpracování se tak bezprostředně dostane do systému vydavatele společně s vygenerovaným AC. Výsledky vykonání druhého typu skriptu zůstávají uloženy v příznakových bitech na kartě až do příští online transakci, kdy jsou společně s prvním generovaným AC přeneseny do systému vydavatele.

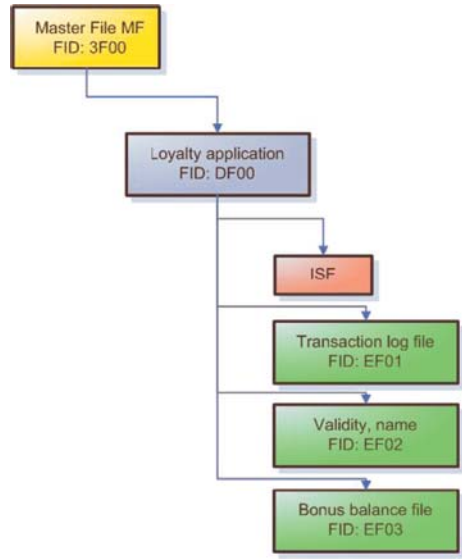
Při kroku 9 (příp. 8 či 6) je transakce ukončena a běh aplikace končí, tj. funkce platební aplikace přestanou být k dispozici.

Další aplikační možnosti využití čipu

Vedle platební aplikace může být na čipu karty nahráno (dle kapacity její paměti) několik dalších nezávislých aplikací. Karty s podporou asymetrického šifrování mohou být například použity pro autentizaci uživatele pro přístup do různých systémů. Nejčastěji je v praxi snaha o implementaci tzv. věrnostní aplikace. Ta může být buď pouze pro jednoho obchodníka, resp. řetězec a nebo pro celou řadu odlišných bonus programů.

Obrázek níže zobrazuje příklad takové aplikace, která obsahuje identifikaci držitele v podobě jména a adresy uložené v souboru s FID ‘EF02’, bodovou hodnotu nákupů v souboru ‘EF03’ a informace o poslední transakci v souboru ‘EF01’. Soubor ISF obsahuje klíče pro vzájemnou autentizaci aplikace na kartě a bonus aplikace na terminálu obchodníka.

Příklad možné datové struktury věrnostní aplikace na čipu:



Bezpečnostní aspekty platební aplikace

Pro offline ověření pravosti dat prezentovaných kartou terminálu jsou tři různé metody:

- Static Data Authentication (SDA) – ověření statického (je stejný pro všechny transakce) podpisu dat na kartě, který ujišťuje o platnosti a neměnnosti dat karty;
- Dynamic Data Authentication (DDA) – ověření dynamického podpisu (je různý pro každou transakci) generovaného kartou, který ujišťuje o platnosti dat a pravosti karty;
- Combined DDA/Application Cryptogram Generation (CDA[10]) – karta generuje dynamický podpis transakčních dat, tím ujišťuje terminál stejně jako u DDA a zároveň o tom, že informace přenášené mezi kartou a terminálem nebyly pozměněny.

Při online transakci dochází k ověření pravosti karty, když online systém vydavatele ověří AC (typu ARQC), který je generován kartou z důležitých transakčních dat za účelem prokázání pravosti dat i karty samotné.

Při offline transakci terminál uschová AC (typu TC) generovaný kartou, aby vydavatel mohl ověřit platnost provedené transakce.

EMV autentizační metody dat/karet:

	SDA	**DDA**	**CDA**	**Online autentizace**
Detekce neplatných statických dat	X	X	X	X
Detekce pozměněných statických dat	X	X	X	X
Detekce zneužití odposlechnutých dat		X	X	X
Detekce útoku vložením pozměněných dat			X	X
Dostupné při offline transakcích	X	X	X	

Seznam metod ověření držitele karty byl uveden výše. Z bezpečnostního hlediska je snaha o eliminaci používání pouhého podpisu držitele, případně jsou

[10] Definuje až EMV 4.1

používány kombinované metody zadání PIN[11] i podpisu (jejich použití u některých typů karet není povolené a navíc to zpomaluje vybavení transakce, takže se v praxi příliš nepoužívá). Zbývá tedy Online PIN a Offline PIN.

Online PIN metoda spočívá v zašifrování zadaného PINu terminálem a odeslání kontrolní hodnoty s ARQC vydavatelské bance k ověření.

Offline PIN je naopak zaslán příkazem VERIFY kartě, která hodnotu ověří oproti referenční hodnotě bezpečně uložené na kartě a informuje terminál o výsledku operace v odpovědi na příkaz VERIFY (zároveň jsou v kartě nastaveny odpovídající stavové informace, které budou terminálu při transakci předány).

Offline PIN může být dle EMV ověřen kartou dvěma způsoby – při tzv. plaintext PIN je kartě předáván v otevřené podobě a při Enciphered PIN je šifrován RSA šifrou.

Offline PIN byl zaveden proto, aby bylo možné realizovat čistě offline transakci a došlo ke snížení telekomunikačních nákladů. Aby toto nebylo na úkor bezpečnosti zákazníka, poskytuje karta prostředky pro risk management a rovněž terminály se chrání zpracováním některých transakcí výhradně online viz terminal risk management výše.

Zajímavá část risk managementu karty je tzv. velocity checking, aneb kontrola počtu prováděných transakcí offline. Karty obsahují jak vnitřní čítač každé započaté transakce, tak počtu transakcí provedených offline, popřípadě i sumární částky offline transakcí.

Jsou zde konfigurovány limity těchto počtů a tyto ještě mohou být dolní/měkké a horní/tvrdé. Při překročení dolního limitu počtu offline transakcí bude karta generovat ARQC. Pokud však terminál není schopen online zpracování a zašle kartě druhou žádost o AC TC, karta transakci (nebude-li další důvod, proč požadoval její risk management online zpracování) schválí. Naopak, pokud je překročen horní limit offline transakcí, karta bude trvat na online zpracování a v případě nemožnosti online zpracování danou transakci zamítne.

Obecně je snaha tyto limity nastavovat na co nejmenší hodnotu. Nastavením na nulu vzniká karta použitelná jen v online terminálech, ale je zde prakticky plná kontrola nad stavem účtu/kreditu.

Vhodným nastavením podmínek v seznamu CVM lze dosáhnout, že například veškeré hotovostní transakce budou prováděny online. Naopak při použití na mýtných branách se preferuje rychlost provedení transakce (nejlépe offline) a protože případné škody při zneužití jsou zde minimální, není při použití karty držitel ověřován vůbec (používat metoda No CVM) a dokonce karty mají možnost v tomto případě ignorovat offline velocity checking.

Aplikace na kartách mohou obsahovat krom omezení počtu chybně zadaných PIN také další ochranné prvky proti zneužití. Může být limitován počet započa-

[11] Délka PIN může být až 12 znaků, ale ne každý terminál je bude pravděpodobně schopen korektně zpracovat.

tých transakcí (tj. např. vložení do čtečky v bankomatu), ale také počet issuer skriptů zaslaných do karty (nezávisle na správnosti MAC) i samotný počet použití symetrických šifrovacích klíčů. Tyto limity nastavuje vydavatel a asociace většinou pouze doporučují hodnoty těchto limitů.

Na platby čipovými kartami vrhá stín přetrvávající existence magnetického proužku, resp. možnost stále ještě platit kartami s magnetickým proužkem. Přítomnost magnetického proužku na čipových kartách je vyžadována, aby bylo možné nové karty používat v zemích, kde ještě čipové terminály ještě nejsou běžně k dispozici. Jako ochrana před výrobou falešného proužku dle dat načtených z čipu karty se nastavují některé údaje na proužek s pozměněnými hodnotami. A naopak na proužku je zaznamenán kód informující obsluhu terminálu, že karta má být použita v čipovém terminálu.

Vydavatelé platebních karet se snaží zvyšovat bezpečnost klientů dalšími doplňkovými bezpečnostními funkcemi – např. možností změny PIN, ale také třeba možností „zamykání“[12] karty prostřednictvím zaslání speciální SMS do střediska vydavatele, které online platby autorizuje.

Šifrovací klíče

Karty obsahují sadu 128bitových TripleDES klíčů – pro komunikaci s vydavatelem, pro šifrování dat a pro zabezpečení dat. Při DDA/CDA k nim ještě přibývá RSA klíč pro ověření pravosti karty. Minimální bezpečné délky klíčů použitých pro SDA jsou předepsány asociacemi/EMV včetně časového omezení platnosti (např. 1024b do 31. 12. 2009, 1152b do 31. 12. 2012, 1408b do 31. 12. 2014 atd.). Na straně banky jsou klíče bezpečně uloženy v HSM modulech a jejich periodická obměna a případný transport se řídí přísnými pravidly key managementu.

Software pro práci s kartami

Pro testovací účely je plně k dispozici, kromě software poskytovaného asociacemi a výrobci karet vydavatelům, například tento software:

- **PC/SC-Lite** – Projekt Movement for the Use of Smart Cards in a Linux Environment (M.U.S.C.L.E.) implementuje API v C (v podobě knihovny *libpcsclite.so*), kterou emuluje ASPI *winscard* z platformy Windows(R).
- **pcscd** – Middleware poskytující v podobě Linux deamonu rozhraní podobné winscard.
- **OpenCT** – Implementace ovladačů a middleware pro čtečky smart karet, vytvořen OpenSC vývojáři, lze použít pro PC/SC Lite.

[12] Toto blokování účtu nemá vliv na transakce prováděné offline

- **OpenSC** – Knihovny a utility nad kartami, převážně kryptografické, implementuje např. PKCS#11, PKCS#15 atp.
- **pcsc-perl** – Perl PC/SC interface v podobě modulů Chipcard::PCSC a Chipcard::PCSC::Card.

Kam spěje technologicky vývoj platebních karet

Jednoznačný trend je zvyšování bezpečnosti platebních aplikací. Snaha je o nahrazování karet s SDA, kartami které podporují CDA nebo aspoň DDA a délky klíčů používaných pro asymetrické šifry se průběžně zvětšují.

Z hlediska praktického je snaha o zjednodušení použití klienty, omezení nutnosti dávat kartu z ruky a zkrácení času potřebného pro provedení transakce. V Jižní Koreji a v USA je tak například již několik milionů bezkontaktních platebních karet. Visa i MasterCard zde navazují na standard bezkontaktních smart karet ISO/IEC 14443 (A/B) s operačním rozsahem < 10 cm a přenosovou rychlostí mezi 106 a 424 Kb/s. JCB rověž používá proprietární bezkontaktní řešení s obdobným operačním rozsahem.

Pilotní projekt asociace MasterCard v USA přenáší z čipu data odpovídající obsahu magnetickém proužku, pilot Visa se snaží provádět plnohodnotnou EMV transakci. Řešení MasterCardu v USA je založeno jednoduchosti integrace do stávajících POS pouhým přídavným zařízením a krátkém dosahu signálu. Z hlediska bezpečnosti však toto řešení zde v Evropě neobstojí.

V současné době EMVCo definuje specifikaci Contactless EMV a jak VISA tak i MasterCard připravují vlastní implementace bezkontaktních transakcí se zvláštním důrazem na minimalizaci komunikace mezi čipem a terminálem. Lze očekávat potřebu kryptografického koprocesoru v čipech spojeného s použitím CDA, což předčí bezpečnost stávajících kontaktních čipových karet. Tyto čipy mají být dostupné formou plastový karet, v hodinkách a v mobilních telefonech.

Dalším směrem vývoje, kterým se začínají platební asociace zabývat, je možnost použití stávajících platebních aplikací na kartách pro autentizaci uživatelů a realizaci plateb přes Internet. Cílem je dosáhnout jednotného standardizovaného řešení e-commerce, které zvýší množství realizovaných plateb na dálku a to za současného snížení nákladů obchodníků na zřízení bezpečné platební aplikace na svých serverech. Připravované řešení Chip Authentication Program má být dokonce použitelné dokonce i pro telefonické bankovnictví. Sice pravděpodobně bude vyžadovat přenosné HW zařízení pro komunikaci PC s kartou, ale i tak se zřejmě máme na co těšit.

Reference a použitá literatura

[1] *PC/SC Workgroup.* http://www.pcscworkgroup.com

[2] ISO/IEC 7816 *Identification cards – Integrated circuit cards – Cards with contacts.*

[3] EMVCo, LLC. *EMV 4.1 specifications (book 1–4)*, červen 2004. http://www.emvco.com

[4] EMVCo, LLC. *EMV Contactless Specifications for Payments Systems.* http://www.emvco.com

[5] Krhovják, J., Matyáš, V. *Platební systémy a specifikace EMV*, DSM 6/2006.

[6] Okrouhlý, J. *Platební čipové karty v Plaster Bank*, DSM 2/2007.

[7] *pcsc–perl.* http://ludovic.rousseau.free.fr/softwares/pcsc-perl/

[8] *Chip and SPIN!* http://www.chipandspin.co.uk/

[9] *Visa approved VSDC Chip Cards as of March 2007.* http://partnernetwork.visa.com/cd/testing/pdfs/Cards.pdf

[10] *Technical Requirements for Visa Smart Debit and Credit.* (VSCD version 1.3.1) http://partnernetwork.visa.com/cd/vsdc/pdf/VSDC_CP.pdf (neaktuální, ale přístupné)

[11] *Visa Integrated Circuit Card Specifications.* (VIS) http://partnernetwork.visa.com/cd/vis/main.jsp (vyžaduje povolení)

[12] *Contactless Payment Specification.* http://partnernetwork.visa.com/cd/contactless/main.jsp (vyžaduje povolení)

[13] *MasterCard specifikace* http://www.mastercardonline.com (vyžaduje členství)

[14] *Výrobci čipů certifikovaných pro platební aplikace:* Atmel, Infineon, STMicroelectronics, Philips, Renesas Technology corp., Samsung Electronics, Toshiba.

[15] *Výrobci karet certifikovaných pro platební aplikace:* Axalto, Austria Card Ltd., Gemplus, Geisecke & Devrient, Infineon Technologies, Oberthur Card Systems, Philips Semiconductors, Sagem, Samsung Electronics.

History of Unix

Ladislav Lhotka

E-mail: lhotka@cesnet.cz

1 Introduction

The Unix operating system is nowadays, at the venerable age of 38, stronger than ever. It runs on a wide variety of data processing and communication devices, from mobile phones through notebooks, desktops and servers to Internet routers and supercomputers. Therefore, it may seem rather surprising that its conception was purely coincidental, unplanned and the early development was carried out by a small group of programmers literally under guerrilla conditions. Yet it is by no means the only case of a revolutionary software concept or program created by gifted and unsupervised individuals far away from business plans and project consortia.

However, the entire history of Unix is quite fascinating and instructive. In this paper I will focus on the most important punctuated equilibria of Unix evolution, which was driven not only by academic and commercial interests but also, at least during the periods of growth, by a community of volunteer hackers. (Note that the word *hacker* is used here in its original positive meaning to denote a person "who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorising about programming". [3]).

Unix is now a trademark of The Open Group.[1] Formally, only systems that are certified by this consortium are entitled to use this name. In this paper, I use the term more loosely, meaning the entire family of Unix and Unix-like systems.

2 Romantic Origins

Since 1964, AT&T Bell Laboratories participated, together with General Electric and MIT, in the *Multics* project. Its aim was to develop a new operating system for the GE computers that would serve the needs of the whole spectrum of users. Multics is now generally perceived as a failed project, but the fact is that it brought into the field of computing a number of revolutionary ideas

[1] http://www.opengroup.org

including hierarchical file system, interactive user sessions, dynamic linking, hot-swap hardware reconfiguration and integrated data security.

Nonetheless, given the limited hardware capabilities and primitive development tools of that time, the Multics project turned out to be overly ambitious and in late 1960s it became clear that it will not deliver a usable system any time soon. As a consequence, Bell Labs gradually withdrew from the project and the top managers even considered ceasing their computer services entirely.

These decisions left behind a group of frustrated Multics developers at Bell Labs' Murray Hill Computer Center – Ken Thompson, Dennis Ritchie, Doug McIlroy and Joe Ossanna. They were already too spoilt by the user friendliness of their Multics prototypes to simply return to card punchers and start feeding the GE 645 with batch jobs. On the other hand, without a clear future and day-to-day duties they had plenty of time for brainstorming their ideas and hacking on whatever they wanted.

One of such pet projects was Space Travel, a sophisticated interactive game that Thompson originally wrote for Multics. After finding an obsolete and little-used PDP-7, he and Ritchie ported Space Travel to this machine. However, since they started from zero (using a cross-assembler on GE 645), they had to implement everything from floating-point arithmetic to process control and a file system. These at first rudimentary system services had been continually improved and in 1970 Brian Kernighan coined the name Unix (the original spelling was UNICS) that was, of course, "a somewhat treacherous pun on Multics" [5].

In 1970, after several unsuccessful attempts, the Unix pioneers finally found tactics that convinced the Bell Labs management to support their work and purchase a new PDP-11 machine. Since the term "operating system" was still taboo in Bell Labs, they proposed to develop a system for editing and typesetting documents tailored to the needs of the AT&T Patent Department. This text processing system – precursor to the troff/nroff suite – was indeed implemented, deployed and used, but, more importantly, provided an excuse for its developers to continue their work on the supporting software, the Unix operating system.

The hardware resources of the PDP-11 were still quite laughable by today's measures: 64 KB of *virtual* address space and 0.5 MB disk. Having such a restricted playground, the Unix developers were forced into having parsimony as their primary priority. This imperative necessarily shaped the system design and led to the famous Unix principle of writing simple programs with short names that do just one thing and make as little noise as possible.

While none of us would probably be able to work on that system, the ancient Unix already had many of the features that are still used in modern Unixes, in particular:

1. The hierarchical file system used (simpler) *inodes* containing file type and size, access permissions and the list of physical blocks occupied by the file.

However, the slash-separated path names did not exist yet so that one had to move through the hierarchy by changing directories one by one. Also, it was impossible to create new directories while the system was running.

2. Hardware devices such as terminals, disks and printers were represented by device files, hence the abstraction of everything being a file.

3. The user interacted with the system through a command interpreter or *shell*, which was a user program rather than part of the kernel.

4. *Multiple processes* were supported, although at first only one for each connected terminal.

The initial period of Unix at Bell Labs culminated in 1973 by rewriting the operating system in the new C programming language. This was an extremely daring decision – till that time all operating systems had been written in assembly languages – but the performance penalty was soon outweighed by the opportunity for quickly spreading the fresh know-how and attracting new developers, mainly in the academic domain.

3 Berkeley Era

Shortly after rewriting Unix in C, in October 1973, Thompson and Ritchie presented their operating system at the Fourth ACM Symposium on Operating Systems Principles [6] and attracted the interest of Professor Bob Fabry of the University of California at Berkeley. In January 1974, a tape with Fourth Edition Unix was delivered to Berkeley and graduate student Keith Standiford embarked on an uneasy task to install it on PDP-11/45. Thompson would help him with debugging remotely via a 300-baud modem.

During the first period in Berkeley, Unix was a time-sharing system in a way that was not particularly appreciated by its users, since the PDP-11/45 was a "dual-boot" machine – each day it ran eight hours under Unix and the remaining sixteen hours under the native DEC RSTS. As the number of Unix users steadily grew (especially after the INGRES database was ported to Unix), it was soon perceived as a severe limitation and UCB consequently invested in new hardware – PDP-11/40 and later 11/70.

A great impulse for further development of Unix at Berkeley was Thompson's decision to spend his sabbatical year 1975/76 at UCB where he previously got his master degree. Apart from porting the latest Sixth Edition to the 11/70, he also implemented a Pascal interpreter that was later enhanced and improved by two graduate students, Bill Joy and Chuck Haley.

This new Pascal on Unix became very popular among computer science students at Berkeley and later at other places, too. In order to be able to satisfy

the requests for this system, Joy put together the first Berkeley Software Distribution (BSD) in 1977 and sent away over 30 copies during the next year. Next year, the second distribution with many additions and improvements (2BSD) was released, followed by a number of further releases till 1995.

Berkeley then soon took over from Bell Labs the role of the leader of the Unix development and coordinator of the user and programmer community. In 1975, the Unix Users' Group, later renamed to USENIX, was founded.

In 1979, Bob Fabry succeeded in obtaining a contract from DARPA to implement the TCP/IP protocols in BSD Unix. With the aim of coordinating the work on this 18-month contract (it was later extended by another two years), Fabry set up the *Computer Science Research Group (CSRG)* that also became the official authority behind all subsequent BSD Unix releases. The TCP/IP implementation, together with the suite of "r" programs (*rsh*, *rcp* and others) first appeared in the interim 4.1a release and soon after in the official 4.2BSD release. Arguably, this high-performance TCP/IP stack was the crucial factor that decided about the success of BSD Unix and Unix in general.

Bill Joy, Kirk McKusick, Ozalp Babaoglu and other UCB developers kept on adding new functionality and improving performance of the BSD Unix. The most significant contributions, apart from TCP/IP, were:

- Virtual memory subsystem that was initially developed for the 32-bit VAX computer and included in the 3BSD distribution in 1979.
- Fast file system, also known as Unix File System (UFS), that offered considerably better performance than the original Thompson's file system and also introduced new features such as symbolic links.
- Job control – ability of a user to start multiple processes from a single terminal – and the C shell that supported it.

The Berkeley branch of Unix continues in three open source operating systems, namely NetBSD, FreeBSD and OpenBSD.

4 Command Shells

From the viewpoint of computer users, the ability to enter commands directly to the keyboard and see the result more or less immediately was perhaps the single most attractive feature of the early Unix (despite the fact that the first terminals were teletypes ASR-33). The interactivity was based on time sharing and inexpensive process spawning, and implemented by means of the *command shell*. On Unix, as well as virtually all operating systems that have been created since then, one instance of a shell program is run for each active terminal. Its

role is to read user commands, execute other programs as child processes, wait for their termination and finally mediate the program output back to the user.

The first such program was named after its author – the *Thompson shell*. While being relatively simple (for example, shell programming was not possible), since the beginning it allowed for input and output redirection from the terminal line to a file using the '`<`' and '`>`' characters. Later versions then introduced a more powerful redirection mechanism – pipes – that opened a new dimension for combining programs. For example, this trivial pipe

```
ls \| wc -l
```

gives the number of files in the current directory. Both programs in this pipe (*ls* and *wc*) run concurrently, essentially as coroutines [2], with the output of the former being connected to the input of the latter. In order to leverage the benefits of pipes, the traditional Unix programming style recommends to use the standard input and output, whenever possible, as the source/destination of program data.

In 1979, the users of Seventh Edition Unix got a much improved shell written by English mathematician Steve Bourne. The main advantage over the Thompson shell was the ability to execute *shell scripts*, i.e., sequences of commands recorded in a file. Bourne, who was a member of the ALGOL68 team, extended the shell commands into a full-fledged programming language by introducing the essential branching and looping constructs: `if-then-else-fi`, `for-do-done`, `while-do-done` and `case-esac`. The Bourne shell is still the default login shell on many Unix systems and has been used for writing myriads of small scripts and also some rather large programs such as databases.

At about the same time, BSD Unix came up with the *C shell* written mainly by Bill Joy. This shell introduced programming constructs reminiscent, as its name suggests, of the C language. However, the C shell, unlike the Bourne shell, has never been widely used for scripting and programming. On the other hand, C shell offered a much more comfortable environment for the simple work from the command line: control of multiple concurrent processes and editing of command line arguments. This made C shell quite popular among Unix users, especially in the academic circles.

Later, a number of alternative shells were written, but two of them deserve a special mention: *Korn shell* written by David Korn of Bell Labs in early 1980s and *Bash* (Bourne-again shell) written originally by Brian Fox for the GNU project. Both shells are backward compatible with the Bourne shell but add and extend the user-friendly features of the C shell. Bash is now probably the most widely used shell among Unix users.

5 The vi Editor

One of the programs that is still included with virtually every Unix or Unix-like system is the *vi* editor. Its author is none other than Bill Joy who wrote it probably in 1976 and 1977. However, the history of its predecessors is interesting as well, also because it uncovers the early European traces in the Unix history.

The first Unix text editor was Thompson's *ed*. A version of this program is still included in most Unixes so that everyone can get an idea of the level of user-friendliness it offered to Unix pioneers. However, we should remember that at that time the only terminal was a line-oriented teletype and therefore the *ed* way of editing text was about the only option.

By mid 1970s, video display terminals became more common and people started thinking about truly interactive text editing with immediate feedback. George Coulouris of Queen Mary College, University of London, modified *ed* into *em* – editor for mortals. He utilised the raw mode of the Unix terminal driver that allowed for sending each keypress immediately to the operating system and displaying it on the screen.

By the way, QMC seems to be the very first institution in Europe that used Unix, namely Fourth Edition in 1973. The first appearance of Unix (Fifth Edition) in continental Europe occurred probably two years later in the International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria.

Coulouris showed *em* to Bill Joy when he was visiting Berkeley in summer 1976. Joy got immediately interested, as he was just struggling with *ed* when rewriting the Pascal compiler. However, it turned out that every keystroke in *em* caused the operating system on PDP-11/70 to swap, so Joy and his friend Chuck Haley started hacking on the program. Their effort finally led to the *ex* editor that was included in the first BSD distribution.

Still in 1976, UCB purchased the new ADM-3A video display terminals that offered screen-addressable cursors. Joy had one of them at home connected by a 300-baud modem and finally developed, over many sleepless nights, the full-screen *vi* editor.

This editor was so much better than its line-oriented predecessors that soon quite a few people, even in Bell Labs, were using it. However, since they worked on different terminals with different control sequences (even though the primitive operations were essentially the same), it was tedious to port *vi* to all of them. So Joy decided to separate the screen management into a small interpreter that reads the capabilities and control sequences of the particular terminal from a disk file, */etc/termcap*. This way of handling video terminal input/output then became a standard library that was used by most full-screen programs.

6 Crisis

In 1982, Bill Joy left Berkeley and founded, together with three graduate students from Stanford University (Andy Bechtolsheim, Vinod Khosla and Scott McNealy) a start-up company named Sun Microsystems. This event started the commercial period of the Unix history. Sun's vision was to build a networked personal Unix machine (workstation) and they had in their hands all the necessary components: BSD Unix with its excellent TCP/IP and also great hardware – Motorola 68k-based workstation originally developed by Bechtolsheim for the Stanford University Network (hence the name SUN).

Also in 1982, the antitrust lawsuit against AT&T that started in 1974 finally came to an end. As a consequence, AT&T was divided into several companies specialising in narrower fields of the telecommunication business (local exchanges, long-distance operation, manufacturing, research&development). On the other hand, the antitrust settlement also lifted the restrictions of the first antitrust suit in 1956 that banned AT&T from selling computer products. Due to this restriction, AT&T had no objections against Ken Thompson distributing first Unix editions on tapes for free to anyone who was interested. In the new situation, AT&T couldn't fail to see the business opportunity and promptly made Unix into a standard software product. The first commercial release was Unix System III in 1982, which was essentially based on Seventh Edition Unix. In 1983, Unix System V was released that also included some components from the BSD branch, for example the *vi* editor.

In the second half of 1980s, a number of companies attempted to repeat the success of Sun and developed their own Unix workstations based either on 68k processors or various RISC chips. The most successful competitors to Sun have been Silicon Graphics, Digital Equipment, IBM and later HP. The fierce competition in the Unix workstation market had the positive effect in spreading Unix, both internationally and also to new vertical markets. However, in the last analysis it almost killed Unix as such.

The first negative effect of Unix commercialisation was the break-up of the hacker community that fuelled the early Unix development. While the BSD distributions stayed freely available, they only supported PDP and VAX minicomputers that became, as an approach to computing, gradually obsolete and gave way to the new networked workstations.

However, the major problem of the Unix market was its fragmentation. As one of the fundamental marketing principles is to differentiate own products from the competition, the workstation vendors introduced many new proprietary features that made various Unixes incompatible in terms of the user interface, system administration tools and programming interfaces.

Meanwhile, Intel-based personal computers became more powerful. The 32-bit 80386 processor already had all the ingredients necessary for hosting Unix –

large flat memory space (up to 4 GB) with memory management, protected mode and multitasking support. Indeed, first versions of Unix for 386 machines started to appear. The most successful was *Xenix*, originally developed by Microsoft from Seventh Edition Unix and later sold to Santa Cruz Operation (SCO) who ported it to the 386 processor.

Microsoft itself found a considerably better business strategy in addressing large masses of non-technical users that were ostensibly overlooked by the Unix vendors. Microsoft marketing succeeded in reaching top managers who decided about large PC installations despite the mumbling of their IT staff about technical inferiority of the Intel chips and MS DOS.

After Microsoft consolidated its dominance in the low-end PC market, it started an assault on Unix with the development of Windows NT in early 1990s. It was quite clear that the relatively weak Unix vendors were not able to compete with the giant and Unix lifetime seemed to be coming to an end.

As a matter of fact, the Unix community was well aware of the risks of fragmentation. Already in 1984, a group of European companies formed the X/Open consortium that tried to identify and define as open standards the main aspects of information technology and chose Unix as the basis. In 1988, the family of POSIX standards was published under the auspices of IEEE. They defined an application programming interface that was ultimately accepted by all important Unix players. These standards were certainly important but nevertheless unable to stop the vendors from new rounds of Unix fragmentation.

In 1987, AT&T started cooperating with Sun and jointly developed System V Release 4. This system integrated the popular features from BSD (sockets, UFS) and SunOS (Network File System), and its 386 version was binary compatible with Xenix. While this looked as Unix reunification, the remaining vendors (DEC, HP, IBM, Siemens and others) felt threatened by the AT&T/Sun coalition and formed the *Open Software Foundation (OSF)* with the aim of creating a BSD-based open Unix standard and implementing it on their machines. The first implementation – OSF/1 – appeared in 1990, but only DEC really delivered it as a product for their workstations. All in all, OSF was a rather expensive failure and contributed to the bitter end of DEC, one of the most influential companies in the history of computing.

The last and perhaps most regrettable incident in these so-called *Unix wars* was the lawsuit Unix System Laboratories (an AT&T spin-off whose task was to develop and sell Unix) versus University of California. In 1989, the CSRG group at Berkeley decided to factor out their own TCP/IP stack from the BSD distribution. This allowed them to distribute their source code without forcing the recipient to pay per-binary royalties to AT&T. The result was the *Networking Release 1* distribution that was offered with a very liberal *BSD licence* that later became one of the most popular open source licences.

Encouraged by the success of the Networking Release, Keith Bostic of CSRG embarked on a seemingly impossible task – rewrite from scratch all parts of the BSD Unix that were "contaminated" by the AT&T code. He invested much effort in recruiting new developers at Usenix meetings and other events and was actually the first to effectively use Internet volunteers. After a year and a half, he had new clean-room implementations of all substantial utilities and libraries. After a detailed inventory, it turned out that only six kernel files of AT&T origin remained and would have to be reimplemented in order to get a freely redistributable system. CSRG released this code (without the six files) as Networking Release 2 in June 1991.

Before long, Berkeley alumni Bill and Lynne Jolitz wrote the missing six kernel files and contributed them to UCB. Bill Jolitz also joined a new start-up, Berkeley Software Design Inc. (BSDI) that wanted to make BSD Unix (including Jolitzes' work) into a commercial product. However, Bill Jolitz soon had a conflict with BSDI management, left the company and later released the very first freely redistributable Unix implementation – 386/BSD, under the same BSD licence as the Networking Releases. This is the code base that the NetBSD, FreeBSD and OpenBSD projects stem from.

In the meantime, BSDI started their marketing campaign offering their BSD/386 (later renamed to BSD/OS in order to avoid confusion with 386/BSD) system for a price seriously undercutting that of System V Release 4. This upset Unix System Laboratories and they filed a lawsuit against the Regents of the University of California and BSDI for code copying and theft of trade secrets. UCB responded with a counter-suit against USL and it looked like the stage was set for a very long and complicated legal battle. Fortunately, in 1992 Novell acquired USL together with the Unix assets and its CEO, Ray Noorda, started to actively seek settlement that was finally reached in early 1994. At that time, Windows NT was already gaining momentum and nobody would bet a penny on Unix.

7 GNU Is Not Unix

The word "hacker" originated in another prominent computer science group, the *Artificial Intelligence Laboratory* at the Massachusetts Institute of Technology. For about fifteen years it had been a thriving place with many bright students and researchers that invented some of the groundbreaking concepts and programs, for example Lisp or time sharing, and participated in the early (pre-Unix) phases of ARPANET. While the principal subject of their research was artificial intelligence – vision, robotics and language – they also spent a lot of time improving, developing and experimenting with operating systems and programming languages. They also eagerly awaited the arrival of Multics and, after

realising that it was not going to happen, wrote their own operating system, the *Incompatible Time-sharing System (ITS)* for PDP-10. This really looks like a déjà vu, doesn't it?

Since 1971, one of the hackers at the MIT AI Lab team was Richard M. Stallman, who helped with the development of ITS, experimented with full-screen text editing and enjoyed the spirit of freedom and cooperation the group was based on. However, in the beginning on 1980s several leading hackers of the AI Lab joined the *Symbolics* start-up that was developing and selling Lisp-based graphical workstations. Together with the fact that DEC discontinued the development of their PDP line of computers, this led to the end of the hacker community at the AI Lab.

Stallman was seriously frustrated by the new situation where proprietary programs without source code slowly filled the niche previously dominated by free source code sharing. Being a highly skilled programmer, he could also easily get a job in that company or another, but he decided instead to follow his moral maxims and leverage his talent in trying to restore the hacker environment. He started the *GNU (GNU is Not Unix)* project whose aim was to develop an Unix-like operating system consisting entirely of *free software*. In Stallman's interpretation, free software must satisfy the following requirements:

- Anyone can run the program, for any purpose.
- Users of the program are free to modify it to suit their needs or correct bugs. Of course, this requires access to the source code.
- The (modified) program can be freely copied and given away to other users.

Specifically, the English word "free" refers here to freedom, not zero price.

In 1984, Stallman quit his job at MIT and concentrated on putting together the GNU system. Partly, he could reuse components that were already free programs, for example the X Window System or TEX typesetting system but still many essential pieces were missing and had to be developed from scratch. Stallman first wrote GNU Emacs, a customisable and programmable text editor, and continued with the development environment: GNU C Compiler, GNU debugger and the make utility.

Stallman earned a decent living (at least to his standards that are similar to those of Mother Therese) by selling tapes with his software and this relative success led him to the idea of launching the Free Software Foundation (FSF) that could hire more programmers to work on the operating system. Indeed, in the second half of 1980s, several programmers were paid by FSF and developed a number of important components of the GNU system, such as the GNU C Library (by Roland McGrath) or Bash shell (by Brian Fox).

Stallman and FSF also formulated a new software licence, the *GNU GPL (General Public License)*. Its main purpose is to guarantee that free programs remain free and cannot be included in derivative proprietary programs. To achieve this goal, they used the copyright laws with an interesting twist known as *copyleft*: the programs are copyrighted as usual but accompanied with a licence that explicitly allows actions that standard software licences usually forbid – copy and modify the program and redistribute it to other users. The GPL also demands that every user be provided (upon demand) with the complete source code for the program. Finally, all derivatives of a program covered by GPL are required to carry the same licence.

The latter requirement, dubbed the viral or infectious property of the GPL, is often criticised, not only by proprietary software vendors but also by the BSD camp, and Stallman is labelled as a communist, crusader or threat to the intellectual property. While I personally do not share all his views, I do believe such uncompromising and idealistic personalities are important for reminding us about the pitfalls of excessive pragmatism. Just as dissidents like Václav Havel were during the communist era.

Anyway, by 1991 Stallman's vision of an entirely free operating system was almost fulfilled, only one crucial component was still missing – the kernel. He wanted to adopt the Mach microkernel developed at Carnegie Mellon University because of the much-advertised advantage of microkernels, namely the ability to write most of the operating system functions as user space programs. However, the reality turned out to be much more complicated and the development of HURD, as the kernel is called, progressed at a pace of a handicapped snail – even now, in 2007, it is still work in progress not ready for production use.

8 Linux

While the appearance of Unix in 1969 was unplanned and surprising, the rise of Linux in the first half of 1990s looks like a sheer miracle. The free 386/BSD system became available at approximately the same time, but Bill and Lynne Jolitz capitalised on the efforts of numerous programmers who had been contributing to BSD Unix for a decade. Also, the GNU HURD team still struggled, after five or six years of work, with delivering just a barely usable kernel. And then comes Linus Torvalds, an unknown undergraduate student of the University of Helsinki, and writes from scratch in about three months a kernel functional enough to get lots of other developers and testers on board.

I first learnt about Linux from an article [1] in the March 1993 issue of *UnixWorld* and immediately gave it a try. At that time, Linux already had shared libraries, TCP/IP, C development environment, GNU Emacs, X Window System and TEX, and that was essentially all I needed. The system was pretty

stable and noticeably faster than both MS Windows 3.1 and other PC Unixes I tested (the latter didn't include 386/BSD though). After going through the adventure of compiling my first kernel, version 0.99, patch level 6, I was firmly hooked.

Linux was not designed as a microkernel but rather as a monolithic kernel which was considered a heresy by the contemporary computer science. In the famous Tanenbaum-Torvalds debate that took place in the `comp.os.minix` USENET group[2], Professor Andrew S. Tanenbaum of the Free University Amsterdam, a highly respected authority on operating systems design, criticised rather harshly the architecture of Linux, also saying that Torvalds wouldn't get a high grade if he were Tanenbaum's student(!). However, Linus was quite stubborn (and also supported by other people including Ken Thompson in the debate) in claiming that the microkernel architecture – a very small kernel-mode part and user-space daemons implementing most of the kernel functionality and cooperating via messages – makes the system complicated and slow. In fact, in the following years many performance enhancements and features from the microkernel world were implemented in Linux. Nowadays, Linux is still a monolithic kernel, but one that is highly modular and portable.

The developers of Linux also invented, perhaps inadvertently, a novel and effective way of managing large numbers of independent volunteer developers contributing their patches exclusively via the Internet. Eric Raymond described and analysed this method in his paper *The Cathedral & the Bazaar* [4]. The idea of "bazaar development" without a detailed specification and design looks like a road to inferior and unmanageable software, but the Linux team has already proved over the years that it is a sound and viable approach. After all, software really isn't a cathedral where the architectonic mistakes cannot be easily undone: it is often more productive to rewrite a broken program than spend too much time on designing an a priori specification that will often later turn out to be broken anyway.

9 Conclusions

For me, the most important lesson of the Unix history is the power of freely exchanged source code. Together with the communication and community-building facilities of the modern Internet, the open source paradigm put together a mighty, albeit slightly chaotic, programming force. However, open source is useful not only for program development but also for the entire software ecosystem, which includes maintenance, support and local adjustments. Of course, not everyone wants to tinker with the source code but those that do have access to all information immediately at their hands without having to pay for training

[2] http://en.wikipedia.org/wiki/Tanenbaum-Torvalds_debate

or signing non-disclosure agreements. This does not mean there is less business opportunity around open source software, just the contrary. The point is though that the profit is generated not *by* (selling) the software but *because of* software and its general availability.

Unix certainly has its own share of problems that stand in the way of a more widespread acceptance, especially among ordinary office and home users. Historically, Unix developers have always targeted technical users and cared very little about straightforward interfaces, ease of use and ergonomic design. Fortunately, there are several efforts underway that may succeed in changing this situation. One project that deserves special attention is *One Laptop Per Child*.[3] This project already developed an inexpensive Linux-based laptop with many innovative hardware and software features tailored specifically to the abilities, interests and psychology of small children. The OLPC project aims at delivering these laptops to millions of kids, at first mainly in the developing countries. Their success could become a nice interim happy-end in the history of Unix.

References

[1] Todd C. Klaus: Checking out Linux. *UnixWorld* 10(3):66–74.

[2] Donald E. Knuth: *The Art of Computer Programming*, Volume I (3rd Edition). Addison-Wesley, 1997. 650 p.

[3] Eric S. Raymond: *The New Hacker's Dictionary* (3rd Edition). MIT Press, 1996. 547 p.

[4] Eric S. Raymond: *The Cathedral & the Bazaar.* O'Reilly and Associates, 2001. 256 p.

[5] Dennis M. Ritchie: The evolution of the Unix time-sharing system. Proceedings of *Language Design and Programming Methodology*, Sydney, Australia, September 1979. Lecture notes in Computer Science #79, Springer Verlag, 1979, p. 25–36.

[6] Dennis M. Ritchie and Ken Thompson. The Unix time-sharing system. Proceedings of *Fourth ACM Symposium on Operating Systems Principles*, Yorktown Heights, NY, October 15–17, 1973. ACM Press, 1973, p. 27 (abstract).

[3] http://www.laptop.org

Identity and Authorization multi-organisation contexts

Peter Sylvester

E-mail: peter.sylvester@edelweb.fr

The topic area of this article is Interoperability in Security Applications and Technical Solutions for Federated Networks and Web Services Security and Identity and Authorisation Management.

The article describes the findings and results of work that has been done for the organisations of the Social Security together with the Direction Securité Sociale (DSS) of the French Ministry of Health and the Caisse Nationale d'Assurance Viellesse (CNAV). The objective of the work was to elaborate an interoperability standard [1] allowing iclient server applications of information systems of different organisations to interoperate.

In order to address a larger audience the work has been done in coordination with the French Direction Générale de la Modernisation de l'Etat (DGME) because the work is related to a more global activity of the DGME concerning general reference specifications for IT systems of the French administration to ensure interoperability [2]. On the European level, there is the European Interoperability Framework EIF [3].

We will describe an approach for cooperation that does not involve the installation of common infrastructures for end-to-end authentication and authorisation (including the management). Instead, it uses a delegated approach using organisation level proxies and gateway services that map local authentication and authorisation decisions from a client organisation to the local authentication and authorisation environment of the server organisation. Similar approaches have been used in the past in an ad-hoc way in electronic mail and other areas, e.g. with gateway, domain or proxy certificates. The proposed solution heavily uses standardised base technologies.

The problem space

Given the ever growing penetration of information systems into practically all areas of an enterprise, there is a growing need to federate identities and the associated authorisation management. We can at least distinguish three different

contexts, one of them being the topic of this article. We give a short overview of the others in order to establish a clear scope.

One well known environment where identity management plays an important role is where a single organisation tries to centralise the definition of identities and their management under a single sign on (SSO) approach. There is an obviously huge market, and vendors have been addressing this since long time.

Today we can see here a large attempt of standardisation in order to enhance interoperability; this is manifested in the OASIS SAML activities. [4] These SAML technology permits a clear separation of applications and security infrastructures dealing with control of identities. Nevertheless, there are important limits, in particular, when organisations are very large or very heterogeneous with different security requirements for sub-populations. Furthermore, when the organisation structures change to more local responsibility, this approach can become difficult to implement.

Another completely opposite scenario is addressed by the Liberty Alliance [5]. Here, an individual has many different identities, but there is no governing organisation that can create a global identity, this is not even desirable. It is the individual that wants to share information between different organisations. Again, and not surprisingly, the SAML frameworks and products can work quite well, and there seems to be a real market. Besides the commercial sector, there are examples in the public sector. In France, a particular application is the context of e-administration relation with citizens. The approach is used in order to permit a citizen to use a one-stop approach in various administrations' activities, e.g., when changing a postal address. In France, for historical reasons, administrations (and any other organisation) cannot easily share information about citizens since there is not single shareable identifier. Thus, to permit a citizen to use a one-stop service, a citizen can federates several identities of different administrations, only authenticate to one, and can easily propagate the changes to other administrations [6].

The third scenario is the one that we treat in this article: On one side organisations permit qualified persons to access some information of another organisation, and, on the other side, organisations propose access to some information to qualified entities of another organisation.

This problem occurs in private and public contexts, in particular it applies to cooperation among different part of public administrations or of para-public organisations. In particular, in Europe with its number of highly independent national structures. The need for such a communication exist as well for the public sector (A2A) as well as for B2B or A2B contexts, anywhere, where employees act on behalf of the organisation or enterprise, i.e., where the employer is responsible for the acts of the persons.

What needs to be developed is a solution allowing qualified persons determined by one organisation to access to an application of another organisation

through the use of a local application, in other words, to implement client/server applications between consenting but independent organisations.

There is a need for many-to-many relationships as shown in the following illustration. As one can imagine, there is are several scalability problems.

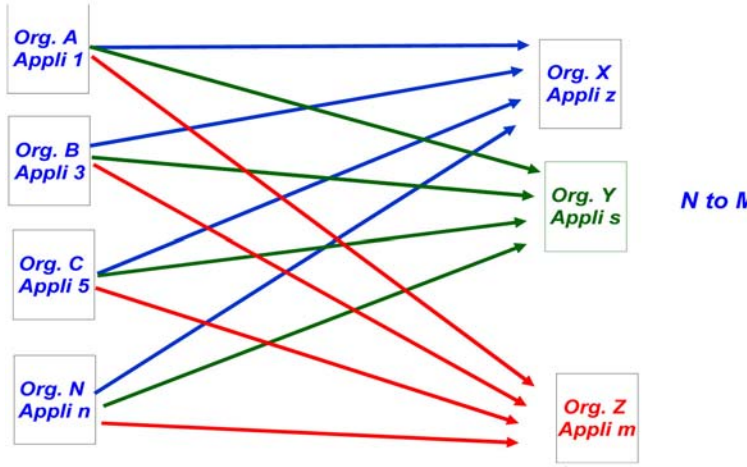


Figure 1 Global communication requirement

Requirements and constraints

In this sub-chapter we list some requirement and constraints and mention approaches which are difficult to implement, and thus excluded.

Organisations have different local and incompatible infrastructures concerning not only the operational information system but also concerning authentication and access and authorisation control. In the past, ad hoc approaches using PKI together with simple access control directories and SSO approaches have been partially implemented to address needs for a a more unified security infrastructure. These approaches are not been extremely successful for various reasons. The proposed PKIs are rather cost intensive and not easy to deploy, and do not solve the problem auf authorisation management.

The impact of the new communication to the existing security infrastructure of each organisation must be as small as possible, i.e., to the largest possible degree, the existing techniques for authentications, roles and access management should be used, and, the required changes should not have an impact to existing usages, and the number of local modifications of the authorisation schemes

should not depend of the number of producer organisations that need to be accessed.

Each organisation is responsible for the attribution of rights and duties to its employees or agents. Thus, authentication and authorisation management need to be controllable by the client organisation.

The distance (physical, organisational, legal) between organisations is often very large and makes management performed by another organisation (even a neutral third party) very difficult and expensive, in particular treatment of changes, and sometimes any delegation is simply impossible. In other words, the overall management of entities and rights must be done in a decentralized, and as a consequence, be done inside the client organisation or under its control.

Naming, roles, rights, attributes or application profiles are specific in each organisation and not necessarily compatible. For example, in one organisation, there may be a hierarchy of global rights, in another and another access rights are based on geographical distribution. In addition, the size of the organisation is an important factor. A global authorisation scheme that tries to combine all the differences is complex and unrealistic to implement.

Another aspect is that some organisations do not want to or must not reveal the identities of employees or agents, this does not mean that anonymous accesses are required, but rather there is need for depersonalisation, i.e., the only the client organisation can (and must be able to) determine from some opaque identifier who has performed a particular transaction.

As an consequence, approaches that would require first some kind of global harmonisation or unification of the identity management is most likely doomed to fail.

When a member of an organisation accesses to another organisation's information system, the organisation takes the responsibility for proper authorisation and they are accountable for misuse. Organisations must respect many constraints regarding the information to be shared, often the concrete possibilities to share information need to be formally agreed and documented, in particular since the information are related to physical persons. The solution must permit the establishment of a service contract, allow easy implementation through a defined set of technologies permitting each partner to remain "master at home" and to assume his responsibilities in a controllable way.

The core idea of a transactions are that the consumer organisation manages its users and access rights and for each request it provides an assertion or attestation which is propagated to the producing organisation together with the request.

A sufficient level of a security must be guaranteed using proper local and strong authentication and authorisation and by a priori trust combined with the possibility of a posteriori control through traces.

An important aspect is that transactions are traceable. Both organisations specify how traces of transactions are performed. It is important that both organisations handle this independently in order not only to allow confrontation in case of problems, but also to allow each organisation to implement its own analysing mechanisms.

Functional decomposition

The intended interactions between organisations can be characterized as follows:

There is a producer – consumer relationship between the information systems of the organisations, the data produced and handled by the information system of one organisation are of interest to another organisation and is accessed and treated by some application in that consuming organisation.

This motivates the first type of an interaction protocol which is based on web services. But for certain (less critical) applications the producer is a simple web interface or portal accessed by a standard browser. The organisations establish an explicit trust through a contract that fixed the rights and duties of each organisation. Depending on the degree of sensibility the contract description may include precise definitions about how the participating perform the identity management. In other words, and to compare it with some better techniques, at least, the contract resembles the engagement of a PKI policy statement, but in this case, it is an agreement. The equivalence of a practice statement may be almost anything between some minimal provisions concerning the communication between the organisation, e.g., addresses of services and certificates, or totally contractual describing even the details of attribution of authorisations.

The overall architecture has three levels:

- The first level consists of all components of preparation (including legal aspects) and a semi-automatic parameter provisioning.
- The second level is the operational system that treat the transaction.
- The third level are all supporting services including the local authentication and authorisation infrastructures, a secure journal infrastructure, dedicated networks and PKIs allow a secure network infrastructure among the organisations.

For the configuration level, the cooperation of each pair of organisation is governed by a formal contract which is also the expression of the mutual trust between the organisations. This contract not only set the legal rules, but also has a formal technical part which can be used without manual interpretation to parametrize the systems. In this way all non-local configuration parameters are fully described in the collaboration agreement, and can therefore be processed

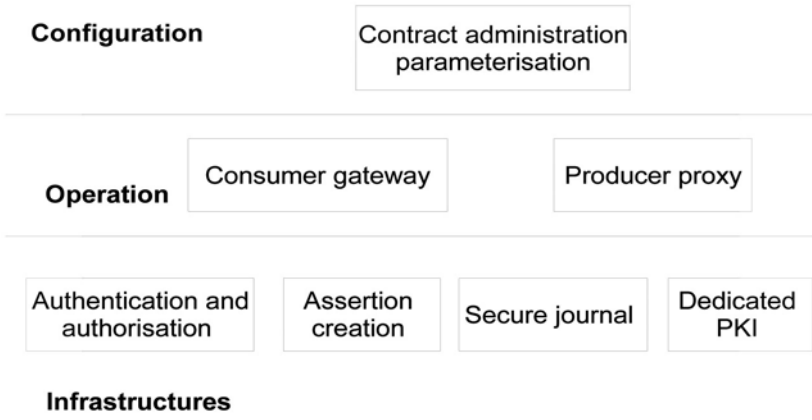


Figure 2 Component Architecture

automatically. This technical annex is therefore a machine readable description of the provided and desired services. We have proposed to evaluate the possibilities the ebXML [7] collaboration establishment procedure and formats, i.e. the establishment of collaboration protocol profiles and agreements to provide all technical parameters of a collaboration. In particular, the usage of an existing standard or proposition allows to use tools for the establishment of the contract and, to some extent, the glue code to parametrize the systems. Although the complexity of the approach may be larger than required, there is the interest of interoperability and adoption.

For the operation level, two usage scenarios have been selected. The producer scenario where the service is provided via a web portal accessible to several client organisations has the following characteristics: The user (or its browser) does not directly access the service. Instead, an outgoing proxy is used that establishes or controls the required authorisations and forwards this together with the request to the target institution. There is an obvious performance problem when a web page contains many visual elements like icons an images. Since, conceptually, each request, i.e., a URL requires an independent and specific authorisation decision. It must therefore be possible to allow the proxy to bypass the strict authorisation rules. Also, a wild card logic for authorisation decisions are necessary to implement.

The outgoing proxy is a completely generic solution for each client organisation independent of any producer service and without any specific local application logic. It can therefore be regarded as a complete part of the client organisations infrastructure.

The producer scenario for web services is very different since there is always a local application that accesses a producer application via a web service.

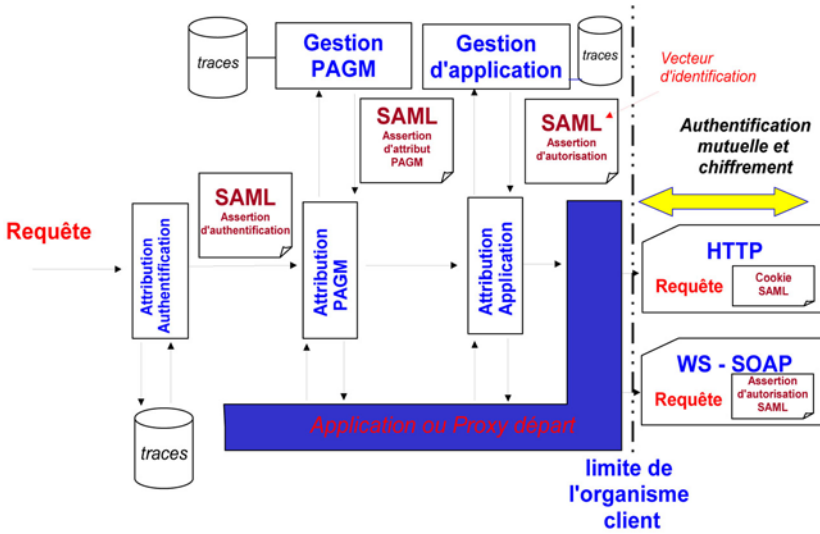


Figure 3 Consumer architecture details

The client application architecture and related security components must ensure proper authorisations for the users. It is highly desirable for this context to have a multi-tiers architecture, where the (potentially insecure) application determines authentication and authorisation via specialized services similar as with some SSO implementations, and, in addition, have a generic web service proxy which controls whether the outgoing requests are accompanied by correct authorisation statements.

For the parts, the solution is in both cases a reverse proxy that implements the journal and verifies the authorisation decision and establishes a security context required for the producer environment, i.e. The authorisation statements are mapped to appropriate capabilities of a security context in the producer environment.

The transactions can be this summarised as follows: The client accesses to a service via a local mirror or gateway (in REST terminology). The client is authenticated using whatever means are used in his organisation. The mirror service (aka gateway or proxy) takes local authentication information to establishes an SAML authentication statement. Next, an attribute assertion is created that asserts a generic role to calling entity, and finally, an authorisation statement permitting the client user to access the target application or service. The assertion and the user requests are transmitted via a gateway to the target server application. On the server side, the authorisation statement is transformed into whatever the target organisation needs to access the service.

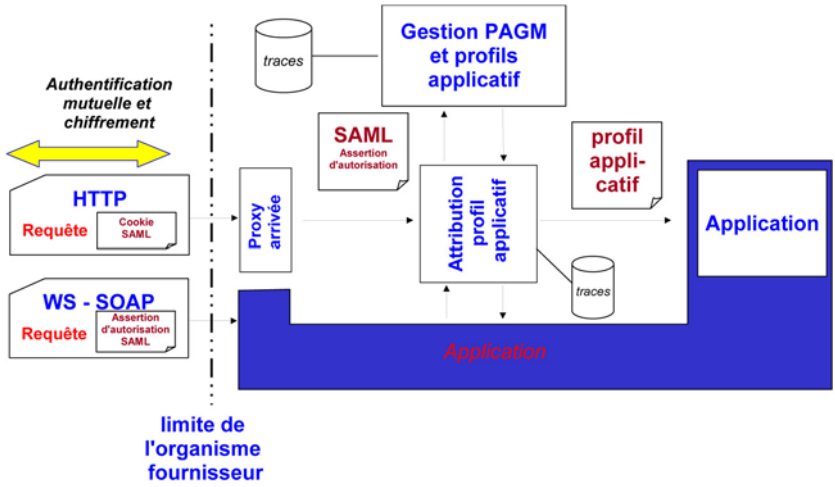


Figure 4 Producer architecture details

For the infrastructure level we have several components.

- The solution interfaces to existing authentication and identity management infrastructure.
- In both scenarios the generic outgoing proxy alos known as local mirror or gateway (in REST terminology).and the incoming (reverse) proxy are responsible for journalising the requests. Although, or, in fact, because the communication is based on mutual trust, the design of the communication standard includes the usage of a secure journal system in order to allow a posteriori control and/or auditing. The design of technical details of the implementation such a system are obviously out of scope, but such a system must meet some obvious service requirements implied by the contracts between the organisation. Technically it is expected that a standardized protocol for secure archiving will be used to interface with the trace system.
- A generic module for the creation and verification of the identification vector. SAML assertion formats are proposed, and thus, the solution can either use existing open source tool kit, or, in the case or local SSO systems be partially based on assertions of the SSO infrastructure (in case that SAML is used there).
- A small dedicated PKI to secure the communication between organisations (not the users).

Since the organisations only talk through proxies, there is no need for an end to end network communication. This not simplifies the network infrastructure between the organisations, but also the security infrastructure between the organisations. Since the services involve access to sensible information concerning persons, organisations are required to protect the communication (at least in France). It has been decided that the gateways and proxies talk to each other using TLS which client and server authentication in order to have full control over the communication channel. Today, it would be more difficult to use an IPSEC solution where the application (the proxies) are largely unaware about whether the connection is protected or not.

Roles, profiles and "PAGM"s

As already indicated, the authorisation schemes used in the participating environments are very different. Even in a simple example where all organisation use an RBAC approach, it is difficult to make a global solution, since the roles are defined independently in each organisation, and, in general there is no simple way to federate them, i.e. , to obtain a common set of roles. One has to solve the $n * m$ problem, i.e. the mapping of roles of n client organisations to application profiles of m producers, i.e. It is desirable to have less then $n * m$ different attributes.

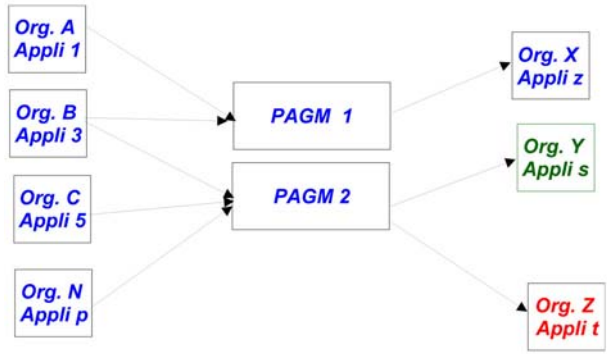


Figure 5 Roles and PAGMs

The proposed solution is to study carefully the role definitions and application profiles, and try to determine a common set of attributes whose definitions are influenced by both existing roles and application profiles, but also, and this is helpful, by legal requirements for qualifications of the actors, i.e. the employees or agents. It is under the responsibility to assign this qualification profiles for which we have invented the abbreviation PAGM (profil générique applicatif

métier) in order not to reuse or misuse any other terminology). The client organisation has to map local roles or qualifications to such a PAGM, or, in an extreme case, assign PAGMs directly to agents, and it is the task of the producer to map this to application profiles. Very often, the consumer organisation simply assigns one simulated user representing the combination of PAGM and client organisation and give the appropriate rights to that user.

The identification vector

As already indicated, the authorisation decision made by the client organisation is transported to the producer organisation. This statement, called identification vector. Its abstract structure and mapping to SAML is outline in the illustration.

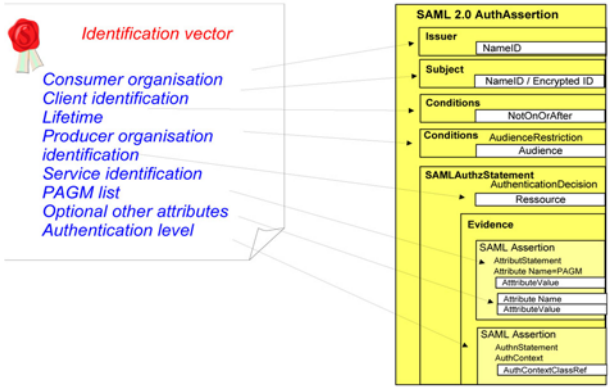


Figure 6 Identification vector structure

The vector contains in an abstract way the following information:

- The client organisation's identification
- An identification of the client
- A validity period
- The producer organisation's identification
- The producer service (a prefix of an URL)
- A list of PAGMs that the client owns for the desired service
- A place holder for optional other attributes
- An indication about the strength of the authentication.

These information are represented using three embedded SAML assertions, an authorisation statement for the service embedding an attribute statement for the PAGM embedding an authentication statement.

Additional remarks

The approach has been specified in detail for the group of all French Public Social Security Organisations, including the transformations necessary in the various local security infrastructures, and published to a wider audience for comments by the French Direction General pour la Modernisation de l'Etat. Remarks and comments have been addressed in the specifications. The experimental implementations will concern real information, including access to information about 80 million persons treated by the French social Security and Retirement services, in order to get feedback from real users. At the time of writing this article, several client organisations that use different local security infrastructures have been fully specified and are being test against the two producer scenarios. They and are expected to be fully operational end of 2007.

The study was conducted in two steps by EdelWeb. During spring 2005, general specifications and principles were developed. In winter 2005/2006 detailed specifications as well as for the generic parts as for the specific parts for some initial applications and customer environments, and development and integration guidelines have been produced. The participating organisations are currently developing of an experimental solution for some real applications. The experiments cover first the outgoing gateway and the mapping to the local authentication environments and the incoming proxy and the mapping to application contexts. In a second step, the collaboration agreement management tools, i.e. the feasibility of the ebXML collaboration process, and the traces system will be studied.

The specifications use or reuse to a very large degree existing techniques and standards, it is therefore expected that that a large part of the necessary implementation will either be available using existing building blocks, in particular concerning the gateway to proxy communication and the transport of SAML assertions. For a web service context, this is essentially one of the approaches used by the Liberty Alliance project. For the direct web interface the logic is slightly different. Some SSO solution providers provide interfaces which are be easily used in the mirror gateway or in the reverse proxy.

References

[1] *Direction de la Sécurité Sociale. Standard d'interopérabilité inter-organisme.* V1.0, 13 july 2005.
http://www.edelweb.fr/iops/Standard_Interoperabilite-V1.0.pdf

[2] *DGME, Référentiel Général d'Interopérabilité version 0.90.* 13 april 2006.

[3] *European Communities, IDABC. European Interoperability Framework, version 1.* 2004. http://ec.europa.eu/idabc/servlets/Doc?id=19528

[4] *OASIS Security Services (SAML) TC.* http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[5] *The liberty Alliance Project.* http://www.projectliberty.org/

[6] *Service-Public.* http://www.service-public.gouv.fr/

[7] *OASIS ebXML.* http://www.ebxml.org/

Identity Management – Compliance and Cost Control

Ralf Knöringer

E-mail: ralf.knoeringer@siemens.com

An Integrated view of identities as the basis for compliance, cross-enterprise business processes and efficient administration

Introduction

Speed, flexibility, agility, early detection of risks and opportunities, rapid response to the resulting challenges – these are the marks of a successful enterprise, especially in tough economic times.

Today's enterprises have to adapt their business processes to the market in real time to keep up with changing market conditions and new customer requirements. Such changes to business processes are now the rule rather than the exception. On the other hand, legal restrictions are growing tighter and tighter. It is necessary to control accounting, protect the private sphere and safeguard intellectual property. The failure to observe rules and requirements by law, i.e. the lack of compliance, can also have direct consequences for management under criminal law.

Confronting these developments is an IT infrastructure that is traditionallyoriented toward functions. Individual systems fulfill defined, individual tasks. In this conventional IT, new requirements generally mean greater complexity.

This is especially true when it comes to integrating applications – something that is indispensable for implementing business processes. This is because business processes link applications within the enterprise and with partners, suppliers and customers and information flows along the process chain. The existing systems provide functions that are used by the processes. The dividing line between internal and external processes and thus between IT within the enterprise and e-business is growing increasingly blurred.

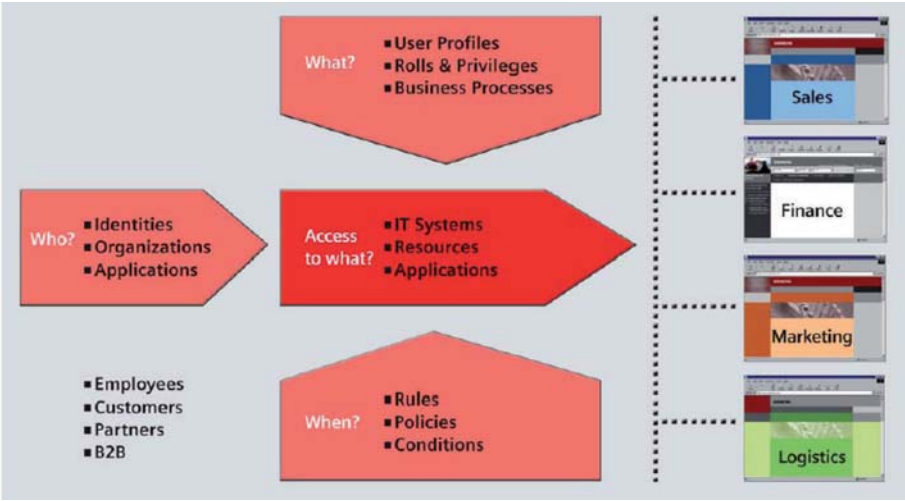


Figure 1

1 Digital identities: The basis for trust in business transactions

In the meantime, new standards for integrating these applications (keyword Web Services) and new approaches for the architecture have emerged. The primary challenge now is handling the identities of the people involved. Their numbers are growing, because it is no longer just the employees of an enterprise who are allowed to access the applications – partners, suppliers and customers are to be, and must be, integrated in the processes.

These persons are represented "digitally" in the systems. Their digital identity is therefore the basic information that must be available throughout the process and across all the systems involved. It is the key to answering the central questions in business processes, security and compliance:

- Who is allowed to access what information and how?
- Who did what and when with what information?

The main issue here is the control of access permissions. To answer these two questions, you have to be able to answer the question as to who is behind the digital identities. Clearly, access permissions cannot be controlled if users are administered completely independently of each other in different systems and are described differently, i.e. have different digital identities that are not linked.

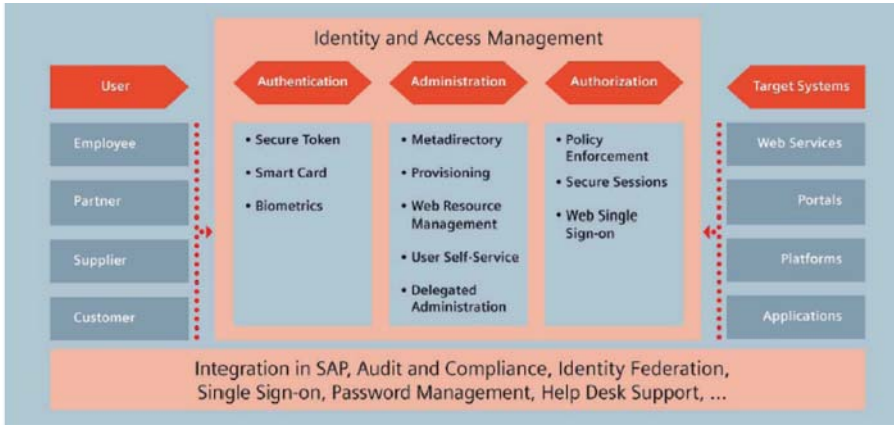


Figure 2 The Identity and Access Management components

As long as there is no consistent view of an identity across the systems, it is not possible to implement new and secure business processes flexibly, nor can compliance requirements be satisfied. IT that is geared to business processes and intended to implement business requirements cannot work without Identity and Access Management. With its secure DirX Identity and Smart Card solutions, Siemens offers the framework for getting IT ready to meet these requirements.

Complete solutions for Identity and Access Management within and across enterprises, smart card-based solutions for secure authentication, solutions for network and system security and for security analysis and consulting help create an IT environment that is able to meet present and future needs.

Processes instead of functions

The technological answer to the question of implementing processes in the future is supplied by Web Services. Over the past few years, Web Services have matured from an idea into a technology that can be used productively and with established standards. In the meantime, areas such as security and reliability have also been addressed above and beyond the basic standards.

The idea of SOAs (service oriented architectures) goes beyond the technological approach of Web Services and can be implemented with Web Services or with other technical approaches. Basically, the idea is very simple. Instead of solving every problem with a separate application that provides functions specifically for that problem, new applications use the functions – or services – of applications that already exist or, if required, new ones.

This makes particular sense when it comes to implementing business processes that extend per se across several systems, areas of an enterprise or even companies. Every new application generates new and resource-intensive integration requirements in the conventional IT environment, resulting in a further increase in the infrastructure's complexity. SOAs help prevent this.

However, apart from technical questions relating to the application infrastructure and system management, which must now look at the entire process and not just the individual server, the challenge is to give users access to precisely the information they should and are allowed to have. That cannot work if every system that provides services has an autonomous **Identity and Access Management** solution.

End-to-end Identity and Access Management is thus a basic requirement for using service-oriented architectures. There are various solution approaches here – from close linking of the systems by means of provisioning and synchronization of information on identities to a loose network in the form of an identity federation that relies on identity data from other connected systems.

Identity and Access Management from Siemens comprises the components for managing identity information, its integration across different systems and directories and authentication and even monitoring of access (AAAA: authentication, authorization, administration, auditing). Especially in a situation where there are new and innovative approaches, but an existing IT and application infrastructure must be used, a broad portfolio of technical solution approaches and implementation expertise, such as that offered by Siemens, are indispensable. The focus here is on providing extensive support for the systems that already exist in IT infrastructures, for example SAP.

Integrated Identity Management for SAP NetWeaver®, mySAP ERP™ and other SAP systems for heterogeneous environments

Certified integration of SAP NetWeaver and the Siemens solutions for Identity and Access Management enables uniform authorization management for applications on the basis of the SAP NetWeaver application infrastructure.

Users in the various SAP NetWeaver modules can be administered globally. The organizational and role concepts of SAP NetWeaver are used to provision other systems. The SAP NetWeaver Portal and the SAP NetWeaver applications provided by it thus become an enterprise-wide access platform. Identity and Access Management from Siemens integrates these elements with non-SAP applications and at the same time increases the efficiency of Identity and Access Management in the SAP Net-Weaver environment.

The Siemens Identity Management solutions thus forge a link between SAP R/3 and mySAP ERP, the SAP NetWeaver infrastructure and the SAP NetWeaver Portal and other internal and external systems. The close cooperation between Siemens and SAP guarantees optimum integration.

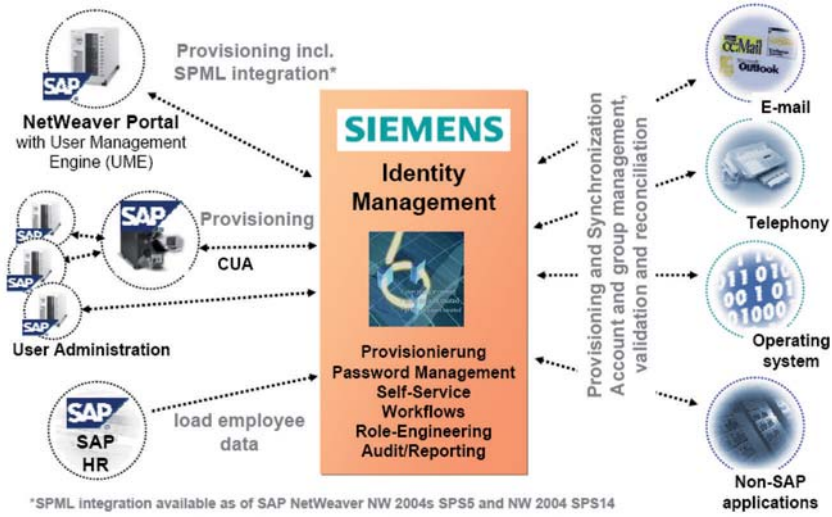


Figure 3 Optimal interworking: Integration of Siemens DirX with SAP systems

Portals as a user interface

Portal are already in widespread use as cross-domain, processoriented interfaces. Whether as employee portals for access to internal applications, partner or supplier portals for collaboration along the supply chain, or customer portals to improve customer relationships – portals are always the interface to a series of applications.

One feature is that users should only have to sign on once in order to be able to use the various applications. Once more, this involves identities and controlling access on the basis of these identities – not to mention farther-reaching approaches such as personalization of the portal's content.

Here, too, there are various solution approaches. The only futureproof and flexible one is the sharing of "universally" available identity information by all applications that can be accessed from the portal. In the highly scalable DirX Directory Server and DirX Identity with its metadirectory service for the automatic synchronization of information in different directory services, Siemens

has a leading technological solution in this field. Based on this functionality, DirX Identity's role-based provisioning enables an automatic provision of user information and the assignment of rights in various systems.

DirX Access also supports Web Access Management and enables single sign-on. This feature authenticates all access via the portal centrally. DirX Access then controls what user is allowed to access what application and how by means of guidelines, policies and rules. The identity data used by DirX Access can in turn be administered by DirX as the directory service. Audit and report functions supplementary support realizing corporate security policies and regulatory compliance requirements.

A further innovative approach for integrating identity information is the identity federation. Here the user logs on to a system (authentication). The other systems trust this authenticating system and receive information on the user's identity. This can be an ID or simply information on the user's role – for example as a clerk responsible for checking invoices. The individual applications can then control what this user is allowed to access.

Siemens boasts leading solutions that are closely integrated with each other in each of these areas, from directory service, provisioning and metadirectory services to Web Access Management, with the Siemens solutions for Identity and Access Management.

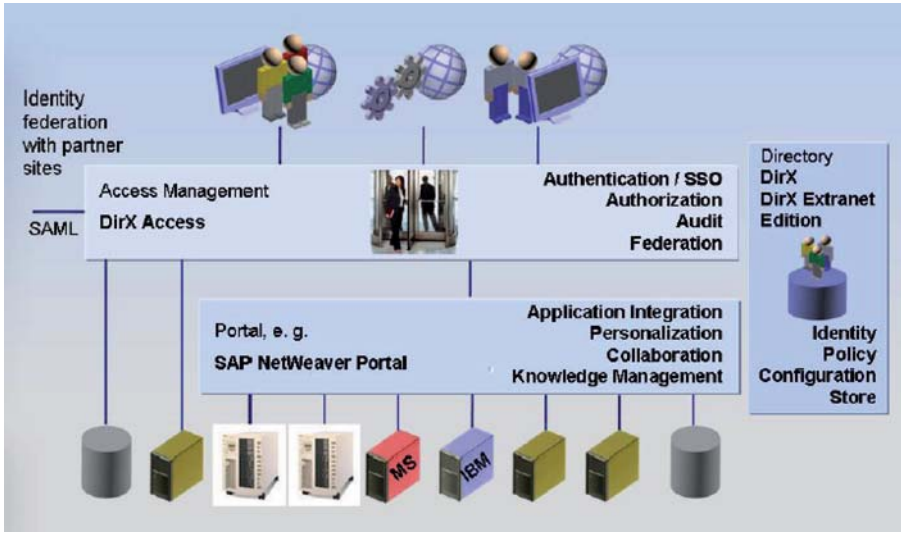


Figure 4 Portals and identity federation

Compliance as a necessity

An often underestimated problem in connection with Identity and Access Management is compliance, i.e. the clear and demonstrable observation of legal regulations. The debate here is currently centering on US standards such as HIPAA in the health sector or the Sarbanes-Oxley Act (SOX) for accounting. These are also of importance for many European enterprises.

However, a point that is overlooked is that European and German regulations, such as the KonTraG (Corporate Control and Transparency Act), the BDSG (German Data Protection Act), the European Data Protection Directive, as well as regulations on risk management in the German Law on Limited Liability Companies and Stock Corporation Law and the strict guidelines on risk management under Basel II, form a closely meshed network of compliance requirements.

Only Identity Management can ensure that an enterprise does not become entangled in this web. This is because a consistent view of "who" is necessary to ensure that the question "Who is allowed to do what where and who did what where?" can be answered. If an employee has different digital identities in a number of systems, it is difficult to obtain a complete overview of the authorizations assigned to the employee and his or her compliance-related actions.

However, apart from the integrated view of identities, defined and stringent processes for managing identities and access permissions are also required. Users must not be created and given access authorizations in ad hoc fashion. Every change to user information and every assignment of rights must be structured and documented.

Only Identity Management can ensure that an enterprise does not become entangled in this web. This is because a consistent view of "who" is necessary to ensure that the question "Who is allowed to do what where and who did what where?" can be answered. If an employee has different digital identities in a number of systems, it is difficult to obtain a complete overview of the authorizations assigned to the employee and his or her compliance-related actions. However, apart from the integrated view of identities, defined and stringent processes for managing identities and access permissions are also required. Users must not be created and given access authorizations in ad hoc fashion. Every change to user information and every assignment of rights must be structured and documented.

DirX Identity, the provisioning solution from Siemens, does precisely that. Internal IT processes are optimized and standardized by means of self-service functions, delegated administration and application and approval workflows. Creation of users in different systems and their assignment to roles and groups are controlled centrally and always carried out in the same way.

Together with the logging of actions by DirX Identity, as well as with DirX Access, this creates the foundation for compliance. The logs enable user actions to be reconstructed. Access can be controlled and monitored centrally with the Web Access Management solution DirX Access. Siemens solutions for Identity and Access Management hence answer the most relevant questions for providing proof of compliance:

- Who has which rights to access which systems?
- Who has granted these rights and why?
- Who accessed which systems and resources and when?
- Who administered which systems and when?

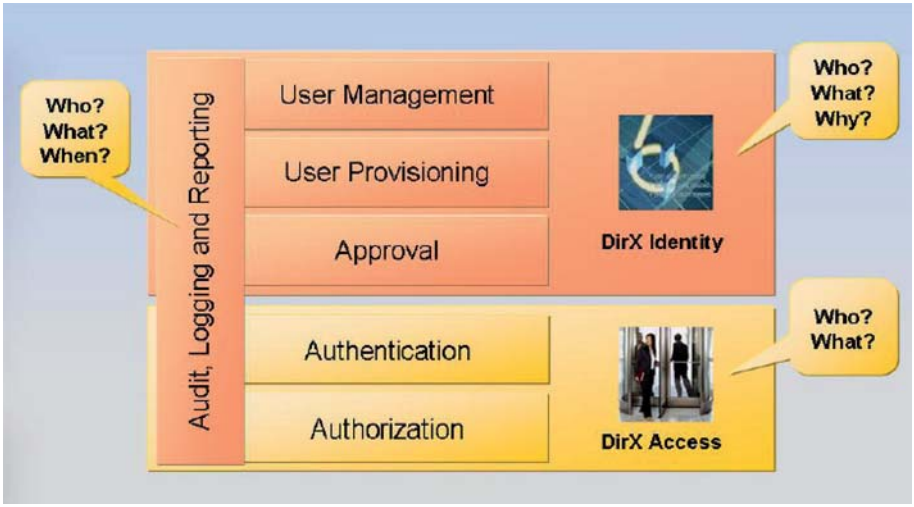


Figure 5

Particularly in applications that are open to partners, suppliers or customers, security plays a major role independently of the issue of compliance. Firewall systems for filtering packets are not enough, since user access to information can only be controlled at the level of the application itself. Here again, DirX Access is the right solution as an authentication and authorization instance. The rules and policies defined for users are enforced on the Web and application servers, with the result that, for example, an end user sees completely different information than a sales partner who looks after a number of customers, for example.

An overview of users and their access permissions is ensured by central management of all access. However, access and enforcement of the rules and policies are carried out locally. Web Access Management with DirX Access does not therefore create any bottleneck, but unites security with scalability.

Cost cutting as an opportunity

Despite the growing requirement for flexibility of IT, security and compliance, cost pressure remains high. IT must therefore not only address the implementation of business processes, but also the optimization of its internal processes. Local user management, a large number of different passwords and complicated security concepts in the various applications are major cost drivers in IT.

With DirX Identity, the processes for creating users and assigning them to groups can be largely automated. Changes can come from HR systems such as SAP and then be automatically replicated by DirX Identity on other systems, such as Microsoft Active Directory, Oracle databases, DirX directories or telecommunications systems. Administrative processes across IT and telecommunications systems are significantly simplified as a result.

These functions are complemented by the ability of DirX Access to delegate administration and creation of workflows for administrative tasks. Moreover, Web Access Management offers the potential for huge savings by simplifying user management for Web-based applications. The security for many applications is increased uniformly and without modifications in the underlying systems.

However, the greatest short-term savings are in password management. Simple interfaces for the resetting of passwords by users or for synchronizing them are not a solution to this problem on their own. They standardize passwords and increase user convenience, but are only part of a multi-level solution. Web single sign-on solutions can be supplemented by password synchronization for existing applications.

The greatest security is achieved by using smart card-based authentication mechanisms that offer users more convenience combined with far greater security. Siemens is the only provider in the Identity and Access Management arena to offer provisioning and mart card solutions from a single source.

Summary – Only integrated solutions will do the job

Complete security infrastructures can be created thanks to the integration in the Siemens portfolio – from network security and smart card infrastructures for reliable authentication to Identity and Access Management as the basis for secure

business processes and portals, compliance and cutting the costs of administering users and access permissions.

Isolated solutions that satisfy individual standards are not sufficient if the infrastructure for Identity and Access Management is not in place. The standardized processes for managing identities and assigning access authorizations that can be created with the products from the DirX family are the basis for meeting the compliance requirements in the pharmaceutical sector. Particularly in environments where compliance and digital signatures are mandatory, proven, close integration with smart card infrastructures, such as that offered by Siemens, is of great importance.

The advantages at a glance:

- Productivity and security are significantly increased by prompt granting and revocation of rights.
- Administration costs are cut noticeably.
- The workload on the help desk is reduced by intelligent password management.
- Automation frees up people to concentrate on their core tasks.
- Increasing number of users of new Web-based applications, for example, are handled more quickly.
- Partners and customers can be integrated in internal services via identity federation.
- Compliance with enterprise security policies and regulatory compliance is supported.
- Management of user data and assignment of rights are more ransparent.
- An integrated identity management solution for SAP NetWeaver, ySAP ERP and other SAP systems in heterogeneous environments s provided.

Identity Management – ORION implementation

Jiří Bořík

E-mail: borik@civ.zcu.cz

Abstract

This presentation discusses two steps crucial to the implementation of an Identity Management system – data source consolidation, and building an advanced provisioning system. As a part of an actual solution (ORION management), Sun Java System Identity Manager project implementation is presented.

1 Orion computing environment

This article summarizes recent activities in the Identity Management area in the ORION computing environment. Since the early nineties, ORION has been providing centralized management and services to all users at the University of West Bohemia. Currently, nearly 20 thousand ORION accounts form one of the largest centrally managed academic user environments in the Czech Republic. The current ORION management system consists of various parts ranging from classic and effective components to modern Identity Management packages (Sun Java System Identity Manager).

An increasing number of interconnected systems and applications effected data transfers between these different systems, and therefore a large number of data duplications. All the relationship network was complicated and unclear and often undocumented. Many of these relations were only occasionally manually maintained administrators. Here are some of the frequent problems:

- reduced functionality of existing managing tools – insufficient relationship with primary data sources, no delegation support for administrators, continuous problem of sleeping accounts etc.,
- absence of sophisticated tools for centralized access management,
- high user dynamics – 5 000 yearly amendments,
- difficulties with detecting and fixing problems,

- high-cost implementation of any new integrated system (unclear or confusing relationship description, heterogeneous technology, necessary customer modification).

This all then required more advanced managing techniques and tools.

Goals of consolidating data sources

The main goal of this stage is to create a quality and credible source of identity data. Usually, in common environments there are many sources of identities not completely or never integrated in one credible source. The result of this situation is conflicting data duplication and then hence – defective data.

In the beginning we had this situation in the ORION environment. Data source analysis indicated that there were at least four main identity sources allowed to create a new identity record:

- students agenda (IS/STAG) – register of students and faculties and information connecting with teaching process
- HR system (Magion) – register of employees and their affiliation to the departments
- ORION computing system management (Moira) – register of ORION users
- identification card system (JIS) – electronic ID cards for students and staff, access and payment system information

Many more other systems included supplementary identity attributes for general use.

For that reason we had to launch project CRO (Centralized Personnel Registry) with the goal to eradicate these conflicts and create a credible source of data for use in all information systems at the University.

CRO created an effective communication basis for data exchange between the main systems. Communication by the XML messages has been established through Message Oriented Middleware driven by Message Broker (see picture 1). Any type of message may have one or more recipients, it depends on the Message Broker configuration. This method simply allows the changing of the message routing for adding a new cooperating system. On the other hand, any new system must have its own developed adapter for connecting to MOM. This adapter allows conversion from an XML message to data changing commands optimized for the local system and vice versa.

After developing and implementing the CRO technology, the next stage was to consolidate the data from previously unrelated sources. Most of the identity

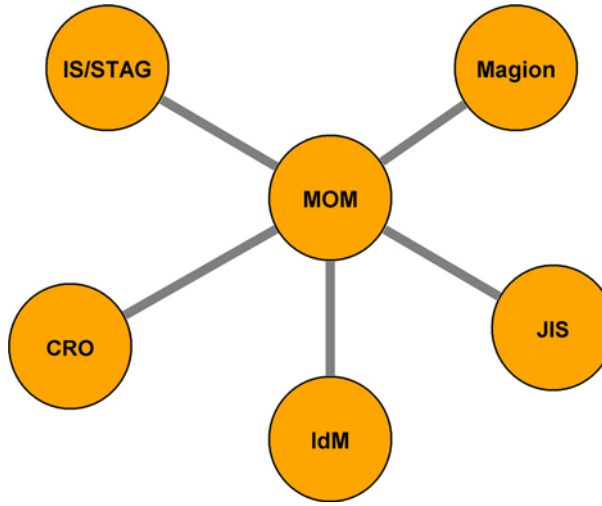


Figure 1 CRO Message Oriented Middleware

records included at least one of these key items – birth certificate number (social security number), personal number or student number. With these key items the data records were automatically coupled. The unrelated records, approx. 5 percent, were processed manually in cooperation with departmental secretaries. All of the data consolidation stage took about six months.

Provisioning

After the data source consolidation we could extend this concept to all systems. But, there are a multitude of systems in ORION, which are purely identity data readers and therefore they didn't need so complex capability of MOM. And on top of that, developing such numbers of adapters would be too expensive, both in labour and costs.

That's why we had to rethink our strategy and decided to use an advanced Identity Management package as a supplement to the system. The features we demanded from the software were:

- Quality provisioning (with least amount of changes to connecting systems)
- Simple workflow system
- Reasonable implementation and license cost
- Ability to connect to other systems using our own sources
- Vendor support, if needed

After some surveys and testing, we decided to use Sun Java System Identity Manager.

Reasons for this choice are as follows:

- Discharge of fundamental requirements
- Reasonable licensing fees
- User references
- Sun offer ORION a free Proof of Concept service
- Results of this PoC

After testing the features of the Sun Identity Manager, we decided to supplement this solution with the Grouper system from Internet2 middleware projects (see picture 2). The complex group management is an important task for us and Grouper is a very suitable tool for this purpose.

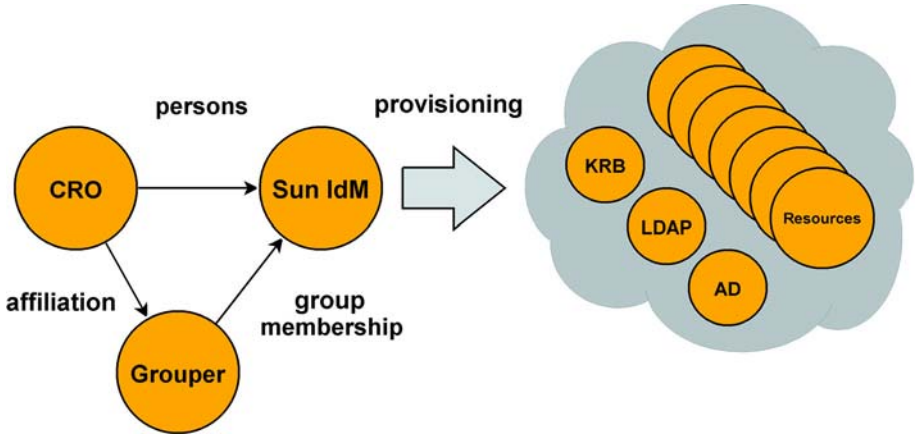


Figure 2 Orion IdM schema

Our current concept effectively split up all functions between appropriate systems and they are as follows:

- Managing primary identities data in CRO,
- complex group managing in Grouper,
- and the other functions – the account life-cycle, role management, provisioning – in Identity Manager.

In our experience this utilizes the best from the applied systems and gives us a complex and effective solution.

Conclusion

It is too early for complex evaluation of the assets in this solution, according to the projected timetable it is expected to be fully operational by August this year. However, at this stage we can make some general conclusions.

First, there is the main question: "to use or not to use" any particular Identity Management package. We need a strong reason to justify the cost of these generally expensive packages. Either we expect that new IdM facilities bring large plusses, or existing problems in this area have to be evident and effectively resolved using IdM. "Trial and error" isn't the right way in this kind of project.

The implementation of Sun IdM took more than one year, ranging from the creation of the basic idea, then choosing the product, proof of concept, to current operational tests. In this project about five staff from the IT department are involved. Concerning some of the labour sources, it is possible to hire an external consultant, but mainly only local people have the in-house knowledge information basis of the existing systems and its relationships.

Presently, we can see in end-systems mainly result of data consolidation. This is not only revision of technology, this is also procedural revision. We have got now clear rules for identity data manipulation and therefore new deployed IdM system can ensure data quality in end-systems. We didn't use yet many of the new function of Identity Manager in operation, but existing project results indicate we go the right way.

Identity a Access Manager řešení jednotné správy uživatelů a uživatelských oprávnění pro heterogenní prostředí

Marta Vohnoutová

E-mail: marta.vohnoutova@siemens.com

Abstrakt

Každá platforma, databáze, skupina aplikací apod. má svou vlastní správu, vlastní seznam uživatelů a uživatelských oprávnění, vlastní bezpečnostní politiku atd. Správa je náročná, ovládání aplikací a přístup k datům klade nároky jak na správce systémů, tak na vlastníky dat i běžné uživatele. Ani pro celkovou bezpečnostní politiku organizace není tento stav vhodný. Proto se implementuje Identity a Access Management.

Příspěvek pojenává o implementaci Identity a Access Managementu v rozsáhlém heterogenním prostředí intranetu státní správy.

Abstrakt

Every platform, database, group of applications etc.has its own administration, own list of users and user access rights, own security policy etc. The result of it is that the administration is demanding, management of applications and data access is difficult and demanding both for administrators, data owners and even for common users. This situation is not suitable even for the security policy of such organization. That is why the Identity and Access Management is implemented.

This ammendment describes the implementation of the Identity and Access Management in heterogenous environment of an intranet of n important state institution.

Klíčová slova: Identita, Autentizace, Autorizace, Audit, Workflow, Řízení identity (Identity Management – IM), Řízení přístupových práv (Access Management – AM), Adresářové služby LDAP (Lightweight Directory Access Protocol),

AD (Active Directory), Role a pravidla, SSO (SingleSignOn), Autentizační API, AAA přístupový portál

Keywords: Identity Authentication, Authorization, Audit, Workflow, Identity Management – IM, Access Management – AM, Lightweight Directory Access Protocol, AD (Active Directory), Roles and Rulesa, SSO (SingleSignOn), Authentization API, AAA access portal

Problémy s údržbou heterogenního prostředí

Ve většině větších firem a organizací existuje heterogenní prostředí. Používají zde různé aplikace na různých operačních systémech, různých databázích, centralizované, decentralizované i lokální, s přístupy přes terminál, klient-server, webové rozhraní.

Každá platforma, případně i skupina operačních systémů apod., databáze, skupina aplikací má svou vlastní správu, vlastní seznam uživatelů a uživatelských oprávnění, vlastní bezpečnostní politiku atd. Správa je náročná, ovládání aplikací a přístup k datům klade nároky jak na správce systémů, tak také na vlastníky dat i běžné uživatele. Ani pro celkovou bezpečnostní politiku organizace není tento stav vhodný.

Podívejme se, jak vypadá například putování nového zaměstnance po organizaci.

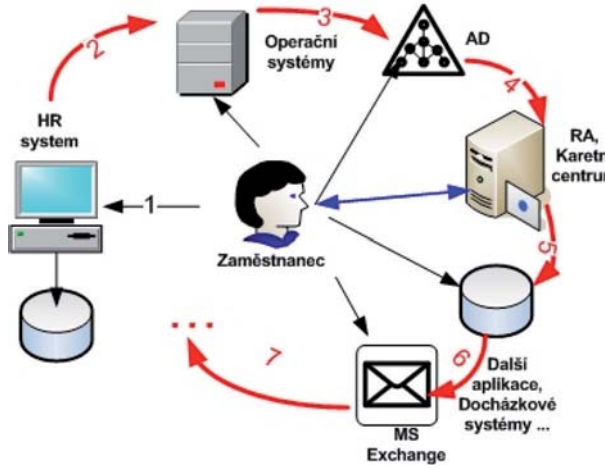
Odborně můžeme říci, že takové „nedůstojné“ kolečko musí nový zaměstnanec absolvovat po organizaci, která nemá implementovánu správu identit tzv. Identity manager.

Identity Manager

Jestliže organizace implementuje Identity Manager pak stačí, aby zaměstnanec navštívil pouze personální oddělení. To mu vydá zaměstnaneckou průkazku a čipovou kartu s instalovanými certifikáty, umožňující mu pohyb po budově i kontrolovaný přístup k aplikacím a datům. Zakládání účtů a distribuci dat po jednotlivých systémech se děje automaticky – a to je právě úkol Identity Managera. Podívejme se na následující obrázek:

Nastoupí nový zaměstnanec, jeho cesta vede na personální oddělení (1), kde ho zaregistrují do svého HR systému. Tím se především rozumí, že pracovníkovi je přiděleno originální zaměstnanecké číslo a je zařazen do organizační struktury.

Celý další proces je automatizován. Aktivace (provisioning) zjistí nový záznam (2) v HR systému a předá ho do Identity Manageru k nastavení účtů



Obr. 1 Povinné kolečko nového zaměstnance po organizaci

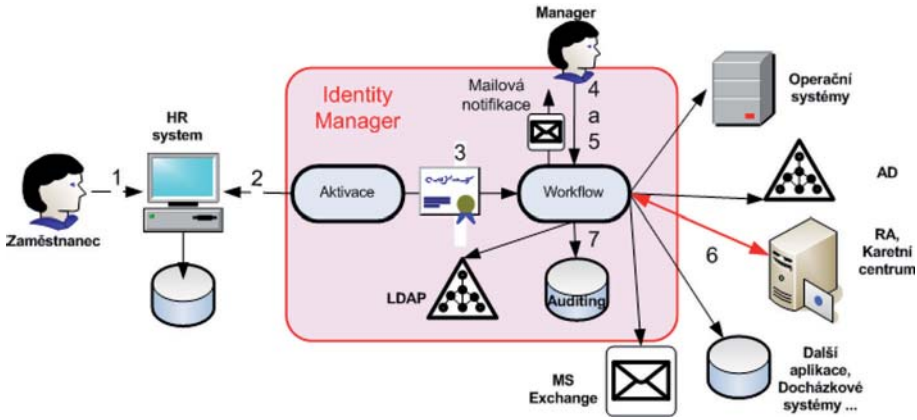
zaměstnance a uživatelských atributů (3). Identity Manager poté inicializuje tzv. workflow se žádostí o schválení pro odpovědné osoby (4) a po schválení (5) aktivuje účty uživatele v operačních systémech, databázích, aplikacích, el. poště apod. (6). Celý proces je auditován (7).

Co je Identity Management

Identity Management je strategie zahrnující různé postupy, procesy a informace sloužící k identifikaci identity během jejího životního cyklu. Touto identitou je jedinec, jeho identita je specifikována množinou příslušných atributů (oprávnění).

K vyřešení Identity Managementu slouží nástroj tzv. **Identity Manager**. Identity Management produktů (dále IM) je na trhu celá řada a jejich kvalita je různá. Hlavními komponentami Identity Managera obvykle jsou:

- Adresářové služby
- Správa elektronických identit
  - Registrace
  - Aktivace (provisioning)
  - Schvalovací workflow
  - Delegování pravomocí



Obr. 2 Po implementaci Identity Manageru stačí, aby nový zaměstnanec zašel pouze na personální oddělení

  - Self-service vybraných činností – uživatel si např. smí sám změnit heslo apod.

- Synchronizace údajů

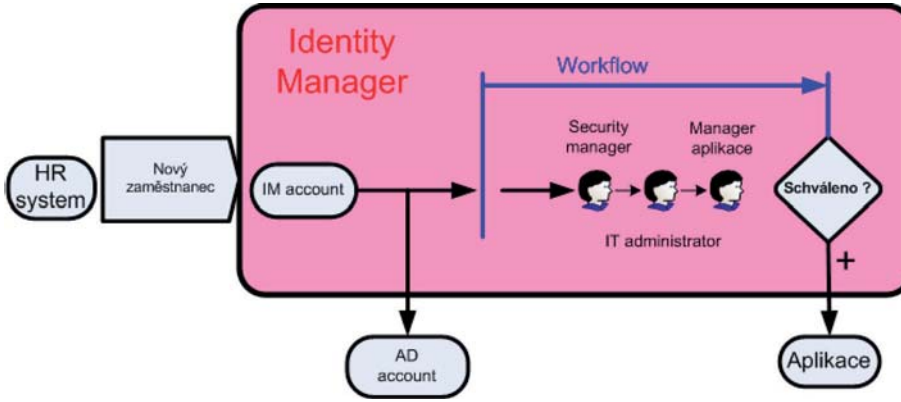
Identity Management centralizuje všechny potřebné údaje o uživatelích (neboli identitách) do jednoho místa. Pomocí Identity Managera lze uživatelské účty snadno vytvořit a/nebo zrušit, čímž přestanou v systémech existovat tzv. „mrtvé duše“, které tam zůstaly po dřívějších zaměstnancích nebo po různém testování apod.

Co je workflow?

Kvalitní Identity Manager obsahuje propracovaný systém tzv. workflow, který je srdcem systému. Česky bychom jej nazvali nejspíše schvalovací proces.

Nastavení workflow není jednoduché, ale jeho správná funkce má za následek, že povolování rolí a přístupových oprávnění provádějí opravdu ti, kdo mají, tedy nadřízení, správci dat apod. a nikoliv ti, kdo rozumí IT technologiím, jak tomu bylo doposud. Workflow oprávněným osobám totiž data ke schválení „přihraje“ takovým způsobem a v takovém tvaru, že IT technologiím opravdu rozumět nepotřebují.

Workflow může také poskytovat data pro informaci, vyjadřovat se k nim apod.



Obr. 3 Příklad workflow

Integrace aplikací do Identity Manageru

Aby mohl Identity Manager s jednotlivými aplikacemi, databázemi a operačními systémy komunikovat, musí být do Identity Manageru nejprve integrovány. Je rozdíl, jestli je integrovaná aplikace celosvětově rozšířená nebo proprietální.

Aplikace celosvětově rozšířené nebo založené na standardech

Tyto aplikace jsou integrovány pomocí předefinovaných konektorů (adaptérů), které jsou součástí dodávky Identity Manageru. Příkladem aplikací a systémů, ke kterým jsou dodávány již hotové konektory, jsou:

- Operační systémy – např. RedHat Linux, Solaris apod.
- Databáze – např. Oracle, MS SQL apod.
- Webové servery – např. WebSphere, MS IIS, apod.
- Rozšířené aplikace – např. SAP

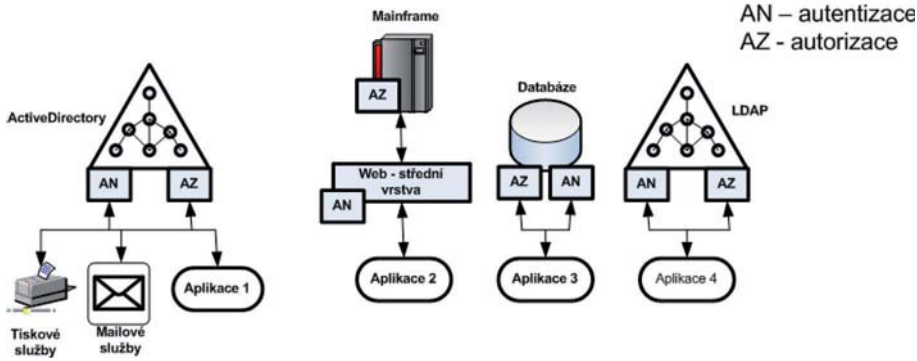
Aplikace proprietální

Proprietální aplikace jsou integrovány pomocí konektorů, které je potřeba nejprve naprogramovat – detailně popsané API je také součástí dodávky Identity Manageru. Příkladem může být např. již zmiňovaný HR systém apod.

Centralizované údaje o uživatelích a jednotnou správu uživatelských účtů tedy máme. Zákonitě nás však napadne, že by bylo vhodné stejným způsobem spravovat i přístupová práva k jednotlivým aplikacím. K tomu slouží další produkt Access Manager.

Access Manager

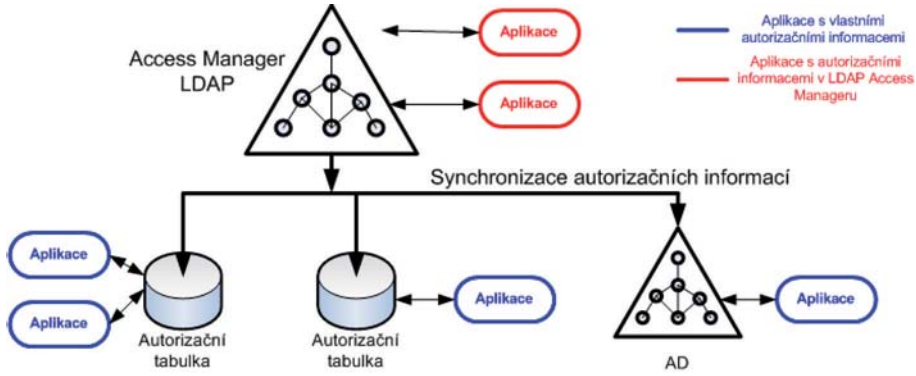
Proces přidělování a správy uživatelských oprávnění je nyní ve většině organizací decentralizován. Je tedy velmi obtížné zjistit, jaká přístupová oprávnění má který uživatel nastavena. Příklad takového stavu je na následujícím obrázku:



Obr. 4 Správa přístupových oprávnění před implementací Access Manageru

Některé aplikace využívají jako zdroj informací o uživatelských oprávněních Active Directory, jiné databáze, různé tabulky či textové soubory.

Po implementaci Access Manageru budeme mít buď pouze jeden zdroj informací o uživatelských oprávněních nebo pokud to nebude proveditelné, vytvoříme navzájem provázanou hierarchii.



Obr. 5 Správa přístupových oprávnění po implementaci Access Manageru

Způsob předávání autorizačních dat aplikacím

Způsob předávání autorizačních oprávnění autentizovaného uživatele aplikaci musí být vždy podroben analýze. Autorizační oprávnění mohou být pak předávány např. ve formě elektronicky podepsané datové struktury, která bude obsahovat seznam oprávnění a rolí uživatele ve vztahu k dané aplikaci.

Většina Access Managerů má také dobře propracované webové prostředí, hodí se proto nejen pro intranety, ale také pro propojení extranetů nebo klientů připojujících se přes Internet.

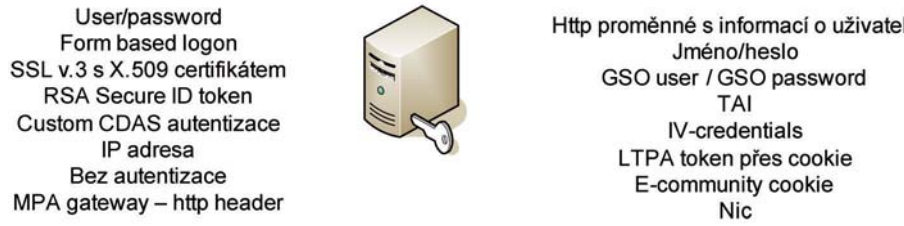
Pokud uvažujete o nasazení např. portálu státní správy, obchodního portálu, portálu pro komunikaci s veřejností nebo obchodními partnery, implementace Access Manageru vám vyřeší většinu problémů s bezpečnou autentizací a autorizací přistupujících klientů.

SingleSignOn

Aby byl celý systém úplný, nesmíme zapomínat sjednotit ani autentizaci uživatele. Vhodné je zavést jednotný způsob autentizace tzv. SingleSignOn.

Access Managery obecně podporují větší množství různých typů autnetizace. Pokud implementujeme Access Management v prostředí intranetu, je možné (i vhodné) zvolit jednotný způsob založený např. na jednotné autentizaci uživatelů do prostředí Windows, nejlépe certifikátem uloženým na čipové kartě. Aplikace pak musíme naučit tuto autentizaci využívat.

Pokud implementujeme Access Manager pro přístup z Internetu či extranetů budeme pravděpodobně využívat širší nabídku autentizačních mechanismů.



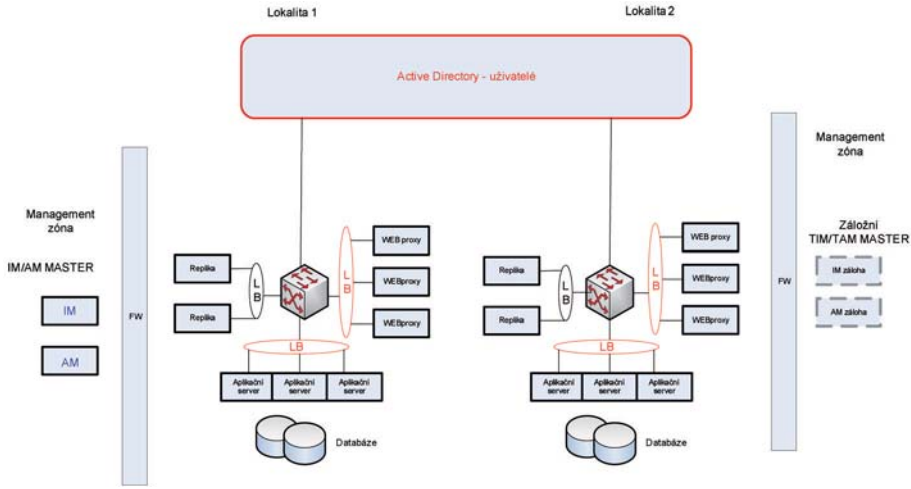
Obr. 6

Na obrázku 6 je znázorněn Access Manager a příklad autentizačích mechanismů:

- levá strana je strana přistupujících klientů (např. Internet)
- pravá strana představuje aplikační servery

Pravá i levá strana podporuje jiné autentizační mechanismy, jedna z úloh Access Manageru je pak převádět jeden způsob autentizace na druhý.

Příklad implementace – síťové řešení



Obr. 7

Na obrázku 7 je uveden zjednodušený příklad síťového řešení. Jedná se o řešení na intranetu, protože se jedná o mission critical řešení, jsou jednotlivé prvky znásobeny a řešení je rozloženo do více lokalit.

Náročnost implementace

Zisk organizace ze správné implementace Identity a Access Manageru bude jistě nemalá. Také auditní kontrola to určitě uvítá. Na druhé straně je však nutné si uvědomit, že implementace je náročná a velmi záleží na kvalitě provedených analýz a na následné disciplíně organizace. Implementace Identity a Access Manageru přinese totiž do fungování organizace nemalé změny, které bude třeba dodržovat.

Uvádí se, že asi 80 % celkové práce na těchto projektech jsou právě analytické práce. Těsná spolupráce pracovníků organizace a především jejího managementu je nutností.

Kdy uvažovat o implementaci?

Identity a Access Manager jsou produkty poměrně drahé a jejich implementace je pracná. Jsou proto vhodné pro větší organizace s heterogenním prostředím vyžadujícím značnou energii na správu. Také organizace spravující důležitá a citlivá data, kde je potřeba mít přístup k těmto datům pod neustálou přísnou kontrolou, jsou vhodnými kandidáty pro implementaci Identity a Access Manageru.

Vypracování celkové bezpečnostní politiky organizace tyto produkty velmi usnadní.

Literatura

[1] IBM RedBooks

[2] Oracle a Microsoft web page

Implementation of workflow for identity management

Jakub Balada

E-mail: Jakub.Balada@siemens.com

Abstrakt

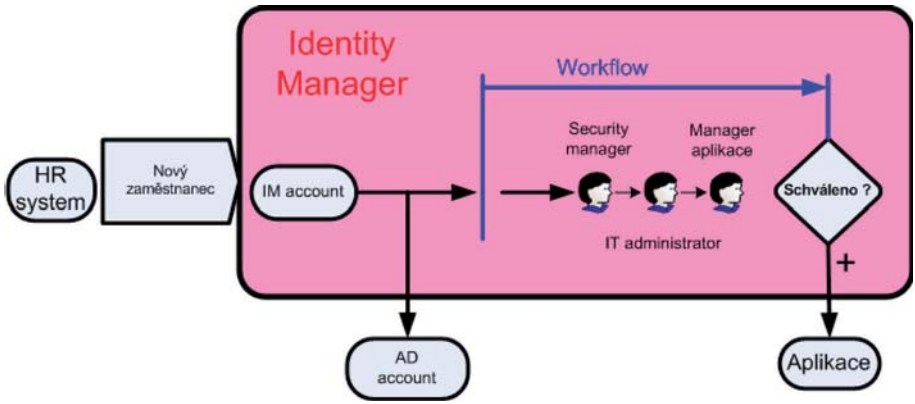
Jedním z velkých přínosů implementace I&A managementu je automatizace organizačních postupů ve společnosti, ať už se jedná o zařazení nového zaměstnance do systému, nastavení jeho přístupových oprávnění nebo například změna pracoviště. Převedení těchto operací z papírové formy pod správu I&A systému vyžaduje hlubokou analýzu organizace společnosti, komplikované jednání s jejími zástupci, navržení a implementace daných workflow a v neposlední řadě naučení všech zaměstnanců novým návykům. O tom všem bude tento příspěvek.

Co si představit pod pojmem workflow v rámci I&A managementu

Nemalou přidanou hodnotou implementace I&A managementu je přesné popsání, nastavení a automatizace postupů v organizaci, které jsou vyvolány nějakou změnou nad identitami. Tato změna může být převzata z HR systému (příchod nového zaměstnance, změna atributu), vyvolána nadřízeným (žádost o přidání rolí) nebo např. správcem aplikace (okamžité odebrání všech rolí). Procesům, které se provádějí v reakci na tuto změnu, můžeme říkat workflow.

Na obrázku 1 vidíme pozici workflow pro nastavení přístupových oprávnění (rolí v aplikacích) v procesu zařazení nového zaměstnance.

Po vytvoření potřebných účtů do systému následuje nastavení přístupových oprávnění nadřízeným. Poté se spustí schvalovací proces, v němž se k daným oprávněním vyjadřují schvalovatelé, kteří jsou vybráni v závislosti na požadavcích. V další části textu si tento proces ukážeme podrobněji. Důležitá je plná automatizace (až na jednotlivá schvalování), která mimo jiné spočívá ve výběru schvalovatelů, generaci mailů, eskalaci požadavků, řešení vícenásobných požadavků a následovně uložení změn. Toto je jeden z hlavních procesů, které využívají workflow, dále si ukážeme ještě další možné případy (změna pracoviště, změna přístupových oprávnění).

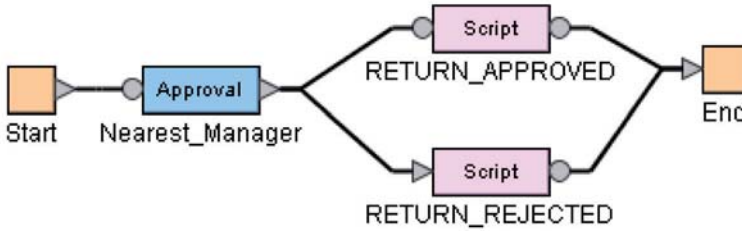


Obr. 1 Umístění workflow pro nastavení oprávnění při příchodu nového zaměstnance

Implementace workflow

Samotná implementace znamená výběr a logické propojení bloků workflow na základě analýzy procesů v organizaci. Tato analýza je základ workflow, proto by na ní měl být kladen hlavní důraz. Zejména účast zodpovědných osob organizace na této analýze je velice důležitá.

Workflow je možné modelovat přímo v rámci Identity managementu. Vyhodnocovací logika se poté většinou implementuje pomocí skriptů, které jsou pod jednotlivými bloky workflow. Na následujícím obrázku je znázorněn elementární případ , představující jednostupňové schválení požadavků.



Obr. 2 Model jednostupňového schvalování požadavků

Po startu workflow, který je jak již jsme si řekli vyvolán nějakou změnou nad identitou, se dostáváme k prvnímu bloku, kterým je approval. Toto je základní stavební prvek workflow, který definuje žádost o schválení. Ta má nadefinovanou sadu vlastnosti, zejména schvalovatele, časový interval na schválení nebo para-

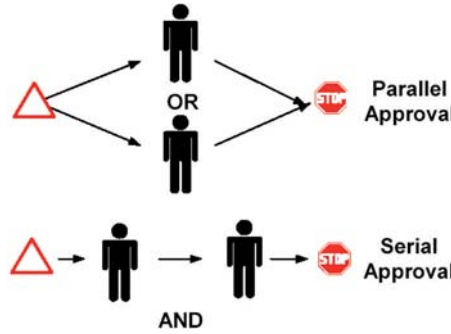
metry pro generaci mailu pro schvalovatele. Dále může mít definovanou eskalaci, což znamená dalšího schvalovatele pro případ, kdy první nereaguje během určené doby.

Po vyhodnocení odpovědi schvalovatele (schválení/zamítnutí), pokračujeme dále po dané větvi. Dostáváme se ke skriptu, který reaguje na daný výsledek schvalovacího bloku. V tomto případě pouze uložíme změny, pokud došlo ke schválení požadavků.

Uspořádání schvalovacího procesu

V praxi se většinou schvalovacího procesu účastní více schvalovatelů, ať už z důvodu bezpečnosti nebo třeba důležitosti jistých osob. Při dvou a více schvalovatelích máme 2 možnosti jejich uspořádání:

- **Paralelní**, kdy stačí souhlas jednoho schvalovatele
- **Sériové**, kde je potřeba souhlas všech zúčastněných schvalovatelů



Obr. 3 Uspořádání schvalovacího procesu

Za zmínku stojí některá pravidla při vyhodnocování:

1. Pokud se v daném skriptu nepodaří vyhodnotit schvalovatele (např. v daném čase není nikdo v pozici správce aplikace) je požadavek schválen automaticky.
2. Pokud se schvalovatel ve workflow opakuje, bere se jeho první vyjádření a dále již není vyzíván.

Reálný příklad workflow

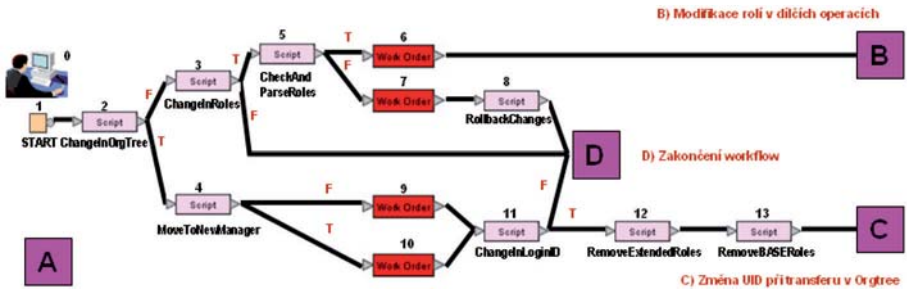
Podívejme se nyní na složitější, v praxi použitelný model workflow. Pro spuštění budeme brát v úvahu 2 akce – změna v HR a požadavky vyvolané nadřízenými. V rámci jednoho workflow budeme řešit 3 procesy – změnu rolí, změnu pracoviště a změnu jiného základního atributu, jako je například telefonní číslo. Všechny 3 procesy mohou být vyvolány z HR, pouze první z nich (změna rolí) může být vyvolána nadřízeným žádostí o přidání/odebrání rolí.

Nyní projdeme kompletní workflow rozdělené do 4 částí. Každý blok workflow je očíslován, některé z nich si podrobně vysvětlíme, některé jednodušší přeskočíme.

A. Zjištění typu procesu

Workflow se spustí (1) změnou nějakého atributu identity. Identitou zde myslíme zaměstnance v organizaci, atributem může být jméno, email, role v organizaci (přístupové oprávnění), pracoviště apod.

První skript (2) zjišťuje, zda se jedná o změnu pracoviště, tedy o přemístění zaměstnance v organizačním stromě. Pokud ano, znamená to v našem případě změnu základních rolí a změnu přihlašovacích údajů (ty jsou v našem případě závislé na lokalitě).



Obr. 4 První část modelu workflow

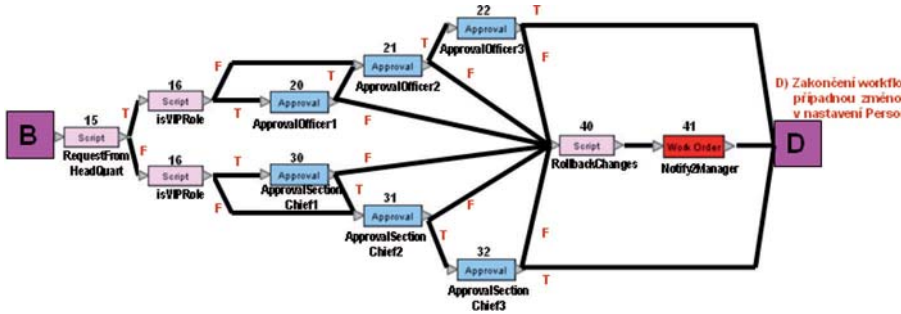
Vydejme se tedy po spodní větvi – změně pracoviště. Nejprve zjistíme (skript 4), zdali se zaměstnanci mění jeho nadřízený. Pokud ano (10), je potřeba oba informovat o tomto přemístění. Dále se automaticky mění přihlašovací jméno zaměstnance (11) a případně se odebírají role (12 a 13). Pokračovat v této větvi budeme ve 3. části C.

Nyní se vraťme na začátek s tím, že se nejedná o změnu lokality. Nacházíme se tedy ve skriptu (3), který zjišťuje, zda se změnily role. Pokud ne, dostáváme se do fáze zakončení workflow (podrobněji ve 4. části), poněvadž ke změně ostatních údajů není potřeba schvalování.

Jedná-li se o změnu rolí (ať už příchodem nového zaměstnance, tedy údaji z HR, nebo žádostí o jejich změnu od nadřízeného), tak se dostáváme ke skriptu (5), který je zkontroluje. Pokud se zde nachází nějaká nesrovnalost (např. žádost o odebrání role, kterou již podřízený nemá), vrátí se všechny role do původního stavu (8) a workflow končí. V opačném případě se pokračuje v části B.

B. Změna rolí

Zde přichází čas na schvalovací kolečko, jehož hlavní stavební kámen – jednostupňový schvalovací proces jsme si vysvětlili výše. Nejprve ale zjistíme (15), zda se jedná o zaměstnance z vedení organizace, což může mít vliv na schvalovací proces. Obě následují větve jsou podobné, pouze se liší ve vyhodnocování schvalovatelů, proto jsou rozděleny.



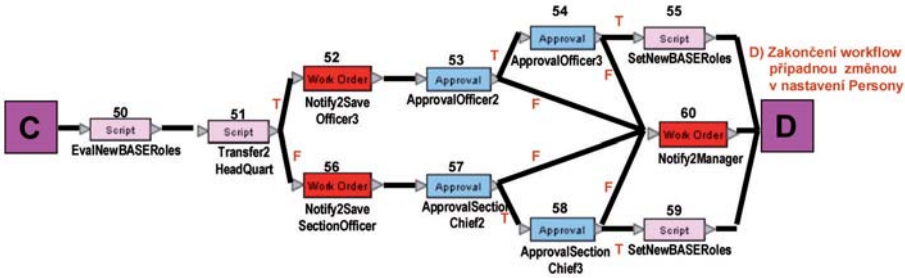
Obr. 5 Druhá část modelu workflow

Následující skript (16) zjišťuje, zda se jedná o významnou roli, říkejme ji VIP (např. možnost zápisu do databáze s finančními údaji). To může mít vliv na počet schvalovatelů, zde uvažujeme o jednom (bezpečnostním) schvalovateli navíc.

Následuje sériové uspořádání 2 nebo 3 schvalovatelů (20, 21 a 22). Jak již jsme si řekli a jak plyne z modelu, ke schválení rolí je potřeba souhlas všech schvalovatelů. V kladném případě se dostáváme ke konci workflow, v opačném k rollbacku (40) a upozornění nadřízeného o zamítnutí změny rolí (41).

C. Změna pracoviště

Nyní se nacházíme v pozici, kdy je změna pracoviště větší (včetně změny nadřízeného) a v našem případě je potřeba nastavit nové základní role a pro zajímavost přesunout domovský adresář a mailbox. To může být potřeba u velkých organizací působících po celé zemi, které mají mailboxy pro zaměstnance na různých místech.



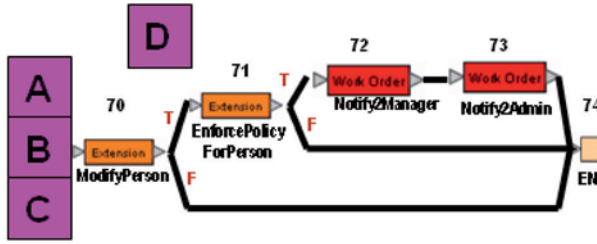
Obr. 6 Třetí část modelu workflow

Nejprve vyhodnotíme nové role, které je potřeba nastavit (50), poté zjistíme zda se jedná o zaměstnance z vedení (51). Obě následující větve jsou opět podobné. Nyní nastává čas pro zásah administrátorů (jediný v celém workflow), jelikož je potřeba přesunout domovský adresář a mailbox. Tato operace bohužel automatizovat nejde, tedy posíláme mail s žádostí o přesun dat a čekáme na vyjádření od administrátorů (52). V tomto bloku je potřeba si uvědomit, že v koncových systémech je stále platné původní nastavení.

Následuje schvalování nových rolí a jejich případné uložení do systému (53, 54 a 55)

D. Konec workflow

Nakonec je potřeba aktualizovat změny u dané identity v databázi (70) a fyzicky vynutit nastavení změněných artibutů účtů patřících identitě (71).



Obr. 7 Čtvrtá část modelu workflow

Poté už následuje jen zpráva o úspěšném uložení změn nadřízenému (72) a administrátorovi dané aplikace (73).

Rozdělení požadavků po aplikacích

Vezměme v úvahu žádost o přidělení rolí, která obsahuje role k různým aplikacím. Jelikož schvalovatelé jsou ve skriptech nadefinováni podle aplikací (v našem případě se jedná o vlastníka aplikace a vlastníka dat), je potřeba workflow pro takovou žádost rozdělit. To znamená, že nadřízený podá jednu žádost o přidělení rolí, ale ve skutečnosti se rozeběhne několik workflow, podle počtu zainteresovaných aplikací.

Jednotlivé role ve skupinách se buď schválí všechny nebo žádná, ale může nastat případ, kdy jedna skupina bude schválena a druhá ne. V tom případě bude nadřízený informován o schválených a nechválených rolích (podle aplikací).

Zvláštní případy

V rámci implementace workflow je potřeba řešit řadu drobných problému.

Jedná se např. o problém při hledání nadřízeného zaměstnance, který se nachází na vrcholu organizačního stromu. Nejde jen o to, že si musí sám sobě žádat o role, ale i o problém při eskalaci schválení požadavku, která jde většinou na nadřízeného.

Dále můžeme zmínit problém při podávání žádosti o role, které nadřízený omylem odešle dvakrát. Workflow se sice chová atomicky, ale už né transakčně. Při zmíněném odeslání požadavků dvakrát se rozeběhnou 2 paralelní workflow (musíme brát v úvahu délku workflow, která může být klidně i 3 dny). Poté se musí řešit případy, kdy v jednom schvalovacím procesu byly role schváleny a v druhém ne.

Podobný problém nastává při paralelním běhu workflow, kdy obě obsahují stejnou roli.

Uživatelské rozhraní

Při implementaci I&A managementu je většinou potřeba vytvořit vlastní uživatelské rozhraní pro zaměstnance organizace. Je to převážně z důvodu workflow, které je vlastně jediné, s čím se běžní zaměstnanci dostanou přímo do styku.

Jedná se především o vytváření žádostí o role, schvalování rolí, výběr delegáta (který může žádat o role pro jeho podřízené) nebo například prohlížení historie.

Všechny tyto operace jsou v principu realizovat přímo v identity managementu, ale většinou je potřeba implementovat tuto nadstavbu. Ať už z důvodu zjednodušení práce s workflow nebo lokalizace do jiného jazyka (musíme si uvědomit, ze systém budou aktivně používat všichni zaměstnanci, kteří mají aspoň jednoho podřízeného), ale především kvůli specifické práci s rolemi.

Tou může být například požadavek na lokalizaci přístupových oprávnění. V tom případě se nenastavuje pouze role, ale je potřeba vybrat i lokalitu, pro

kterou daná role (popř. aplikace) platí. Berme v úvahu lokalizaci na stupni aplikací, tzn. všechny role v dané aplikaci platí pouze v lokalitě, kterou má zaměstnanec nastavenu pro danou aplikaci. Tedy při prvním požadavku o role se musí zažádat o dvojici role-lokalita, což bez speciálního uživatelského rozhraní nejde s příslušným pohodlím provést.

Tato nadstavba se většinou implementuje jako webová aplikace, která komunikuje s Identity managerem pomocí jeho API. Jeho hlavním úkolem je zjednodušit práci se systémem pro koncové uživatele, kteří vyjma administrátorů přijdou do styku pouze s ním.

Závěr

V rámci implementace workflow je nejdůležitější, jak již bylo zmíněno, velice důsledná analýza souvisejících procesů v organizaci. Samotné sestavení workflow není zas tak složité, jak je vidět z reálného příkladu, ale velice pracné. Jednotlivé skripty musí pracovat se spoustou číselníků, nesmí zapomenout na jakoukoliv kombinaci požadavků a hlavně se musí při jejich implementaci myslet na budoucí změnu nebo rozšíření, kterých v podobných projektech není po málu. Typický je příklad vedoucího pracovníka, který při fázi definování workflow požaduje informace o schválení většiny rolí a po nasazení si stěžuje na přehršel informací, které dostává.

Literatura

Schémata převzaty z *Analýzy workflow vypracované společností Siemens IT Solutions and Services* v rámci projektu I&M managementu.

Reverzní proxy neboli Access manager

Libor Dostálek

E-mail: Libor.Dostalek@siemens.com

Abstrakt

Přednáška se věnuje objasnění rozdílů mezi proxy a reverzní proxy. Je rozebírán problém autentizace klientů vůči cílovým serverům. Zmíněny jsou základní autentizační metody: basic, webovými formuláři, uživatelským certifikátem, protokolem Kerberos (včetně SPNEGO) atd.

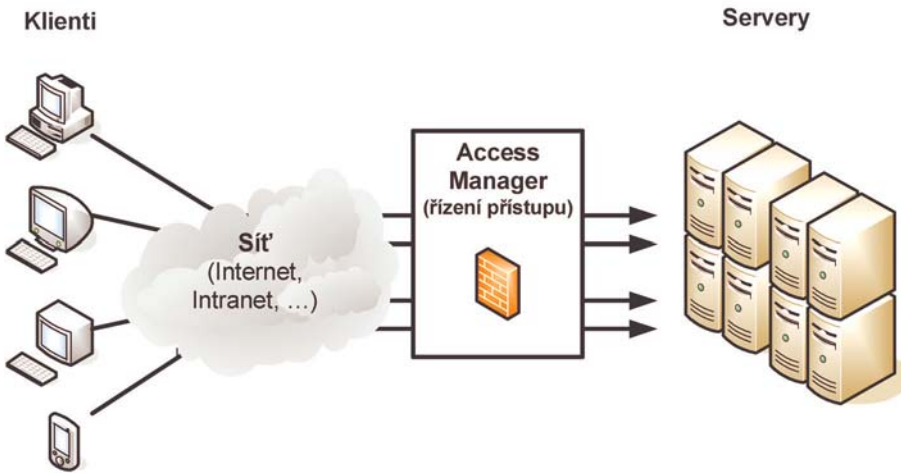
1 Co očekáváme od Access Manageru

Dnes už snad každá firma má na Internetu vystaveno své www.firma.cz, www.firma.com či to úplně in, kterým je www.firma.eu. Kdysi tyto HTTP servery běžely nad statickými stránkami, které vlastnoručně pořizovali sami webmastři editorem vi. Tato doba, kterou pamatují již jen opravdoví pamětníci, dnešním mladým moderním jinochům (tzv. mlamojům) splývá s dobou děrných pásek či magnetických bubnů.

Postupem času se ze statických stránek www.firma.cz přešlo na komplikovaná portálová řešení do jejichž podstaty dnes nevidí už ani jejich autoři natož webmásteři. Ale to vše jakoby ještě nestačilo, neboť mnohé firmy dnes dospěly k názoru, že své agendy (aplikace) rovněž vystaví na Internetu. Výsledkem je, že www.firma.cz se stala jen jakousi slupkou za kterou teprve následuje portál a jednotlivé agenty.

Jelikož funkcí této slupky je řídit přístupy klientů z Internetu a předávat je cílovým serverům, tak tato slupka dostala název Access Manager.

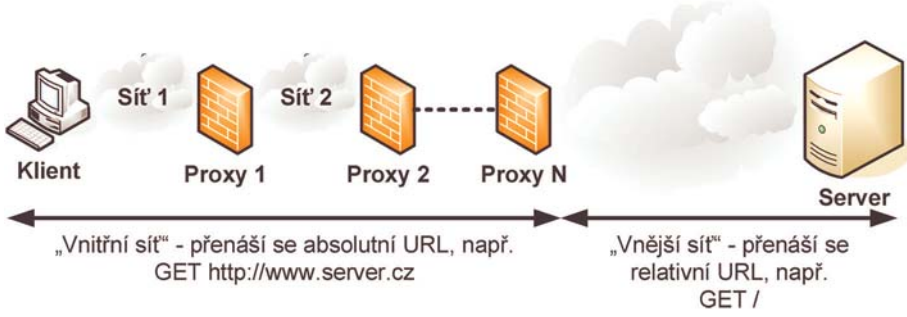
Zajímavé je, že se následně ukázalo, že v případě velkých organizací, které mají na intranetu řadu agend, je Access Manager rovněž velice užitečným nástrojem na intranetu. V takovém případě je přístup zaměstnanců na intranetový portál a jednotlivé podnikové agendy také velice užitečné řídit Access Managerem. Velké organizace pak mají dva Access Managery: jeden pro přístup zákazníků přes Internet a druhý pro přístup zaměstnanců k jejich jednotlivým agendám na intranetu.



2 Reverzní proxy

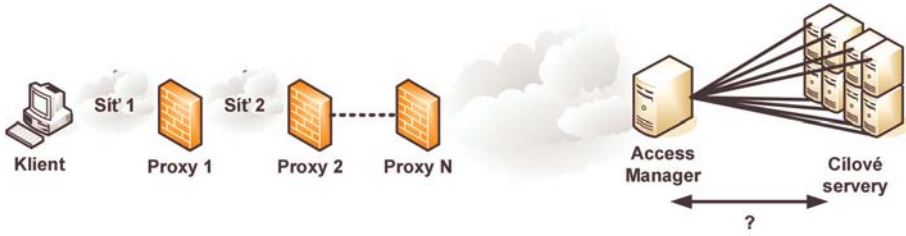
Access Manager je možná výstižné pojmenování, ale jaký je jeho princip? Principem je tzv. reverzní proxy. Proč reverzní?

Klasická proxy je nástroj, který čeká na požadavky klientů a předává je dále směrem k cílovému serveru. Proxy v protokolu HTTP pracuje tak, že klient má ve svém konfiguračním souboru uvedenu IP adresu proxy. Klient pak předá celý požadavek, celé absolutní URL, tak jak leží a běží, této pevně nakonfigurované proxy k vyřízení.



Na cestě k cílovému serveru může být ale další proxy (proxy on proxy). Její IP adresa musí být pevně nakonfigurována v konfiguračním souboru předchozí proxy atd. Nakonec poslední proxy (na obrázku označená N) pak konečně přeloží DNS jméno cílového serveru na IP adresu a naváže s ním spojení protokolem TCP do kterého vloží již relativní URL.

Všimněme si, že na cestě od klienta k cílovému serveru mohla být celá řada proxy, ale všechny leží jakoby na straně klienta. Pokud bychom Access Manager realizovali jako proxy, pak by proxy N musela mít ve svém konfiguračním souboru uvedenu IP adresu Access Manageru – a to je zjevný nesmysl.



Access Manager je tedy jakási trochu jiná proxy – proxy, která leží na straně cílového serveru, proto dostala název reverzní proxy. Reverzní proxy musí předávat požadavky klientů cílovým serverům na základě jiných informací než to dělá klasická proxy. Je zjevné, že tvůrci protokolu HTTP s reverzními proxy nepočítali, a tak budeme muset nějak „naroubovat“ reverzní proxy na stávající protokol HTTP. Výsledkem je příležitost pro lidovou tvořivost jednotlivých vývojářů.

Další důležitou vlastností reverzních proxy je, že z hlediska klienta zastupují cílový server. Z hlediska klienta tak není jasně vidět, je-li jejich požadavek vyřizován přímo reverzní proxy nebo nějakým serverem za ní. Důsledkem je, že klient se bude vždy autentizovat vůči reverzní proxy. Kvalitně nakonfigurovaná reverzní proxy by pak měla tuto autetnizaci korektně předat cílovému serveru. Jinými slovy: s nástupem Access Managerů byla nastolena potřeba Single Sign On jako snad nikdy před tím.

3 Výhybka (junction)

Já vím, že junction je železniční křižovatka, ale mně připadá přiléhavějším slovo výhybka. Protože požadavek klienta si umím představit jako vlak, který dorazil na reverzní proxy, která právě přehazuje výhybku směrem k příslušnému cílovému serveru.

Na bázi čeho může Access Manager takto přehazovat výhybky? V podstatě má k dispozici:

1. IP adresu klienta.
2. Způsob autentizace klienta.
3. Identifikaci klienta v případě, že se klient autentizoval (tj. jedná-li se o neanonymního klienta).

4. Čas, kdy klient přistupuje.
5. Metodu, kterou přistupuje (GET, POST, ...).
6. URL.

Teoreticky může být využita pro přehazování výhybek jakákoliv informace, ale zpravidla se využije část URL. Zjednodušeně můžeme URL pro protokoly HTTP/HTTPS popsat jako:

```
http(s)://www.firma.cz/cesta-1/cesta-2/.../cesta-n/soubor
```

(část URL, následující za případnými znaky # nebo ? není pro další výklad podstatná)

Jistě by bylo možné mít pro každý cílový server jiné DNS jméno (na místo jednotného www.firma.cz), ale nebylo by to příliš praktické. A tak se zpravidla pro „přehazování" volí informace uložená v `cesta-1`, která se pak označuje jako výhybka či junction.

Snadno se to pak pochopí na následujícím příkladu. Mějme na demilitarizované zóně firmy servery s jednotlivými agendami a portálem. Nechť se např. jmenují: katalog.intranet.cz, objednavky.intranet.cz, podatelna.intranet.cz a portal.firma.cz (přitom DNS jména těchto serverů nemusí být ani viditelná z Internetu).

Na homepage firmy jsou pak např. hypertextové odkazy:

- www.firma.cz/catalog, který Access Manager převádí na cílový server na DMZ katalog.intranet.cz.
- www.firma.cz/order, který Access Manager převádí na cílový server na DMZ objednávky.intranet.cz.
- www.firma.cz/registry, který Access Manager převádí na cílový server na DMZ podatelna.intranet.cz.
- www.firma.cz/, který Access Manager převádí na cílový server na DMZ portal.firma.cz.

4 Řízení přístupu (management)

Cílem řízení přístupů na klasické proxy je zpravidla omezovat přístup zaměstnanců na nevhodné servery v Internetu (např. na oblíbeny www.playboy.com).

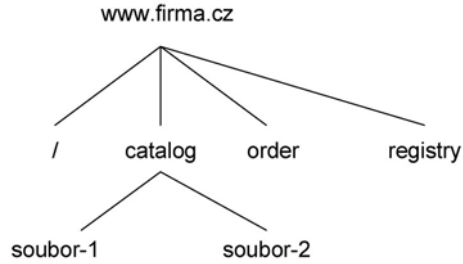
Podobně jako máme řízení přístupu na klasických proxy, tak můžeme i na reverzních proxy řídit přístup, tentokrát ale klientů na cílové servery.

Pro řízení přístupu tak můžeme opět využít jednu nebo kombinaci z následujících informací:

7. IP adresu klienta.
8. Způsob autentizace klienta.
9. Identifikaci klienta v případě, že se jedná o ne-anonymního klienta.
10. Čas, kdy klient přistupuje.
11. Metodu, kterou přistupuje (GET, POST, ...).
12. URL – tentokrát je velice zajímavou právě výhybka.

Můžeme např. nastavit, že na www.firma.cz/order (tj. na výhybku order) mohou přistupovat jen konkrétní autentizovaní uživatelé a to např. jen v konkrétních pracovních hodinách.

Informace na cílových serverech tvoří strom:

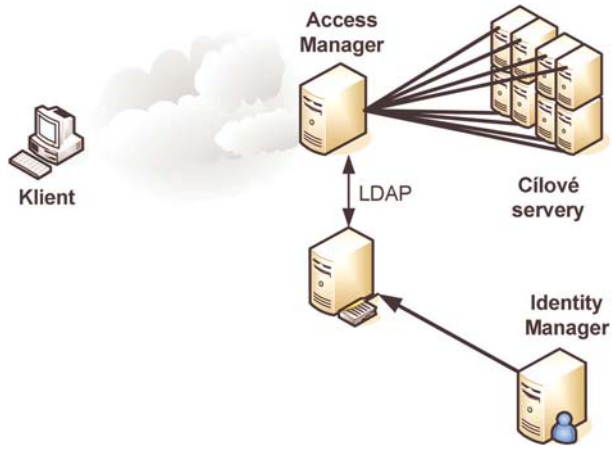


Na každý uzel tohoto stromu lze navěsit konkrétní přístupová oprávnění či jejich další omezení (např. v čase). Jelikož se jedná o stromové uspořádání, tak ideální bází pro tyto informace je protokol LDAP, proto Access Managery, konzultují předávání informací cílovým serverům zpravidla prostřednictvím protokolu LDAP s nějakou bází dat.

Tato báze dat bývá zpravidla plněna z Identity Manageru. Princip pak spočívá v tom, že v Identity manageru jsou každé osobě přiřazeny role. A v Access Manageru jsou pak jednotlivým rolím přiřazena konkrétní přístupová pravidla.

Zpracování pak probíhá v několika krocích:

1. Provede se autentizace klienta, tj. zjistí se identita klienta.
2. Zjistí se jaká role je klientovi přiřazena v aplikaci.
3. Zjistí se jaká přístupová oprávnění jsou přiřazena této roli.
4. Na základě zpracování těchto přístupových oprávnění se klientovi buď umožní skrze Access Manager přistupovat na cílový server nebo se jeho přístup zamítně.



5 Autentizace

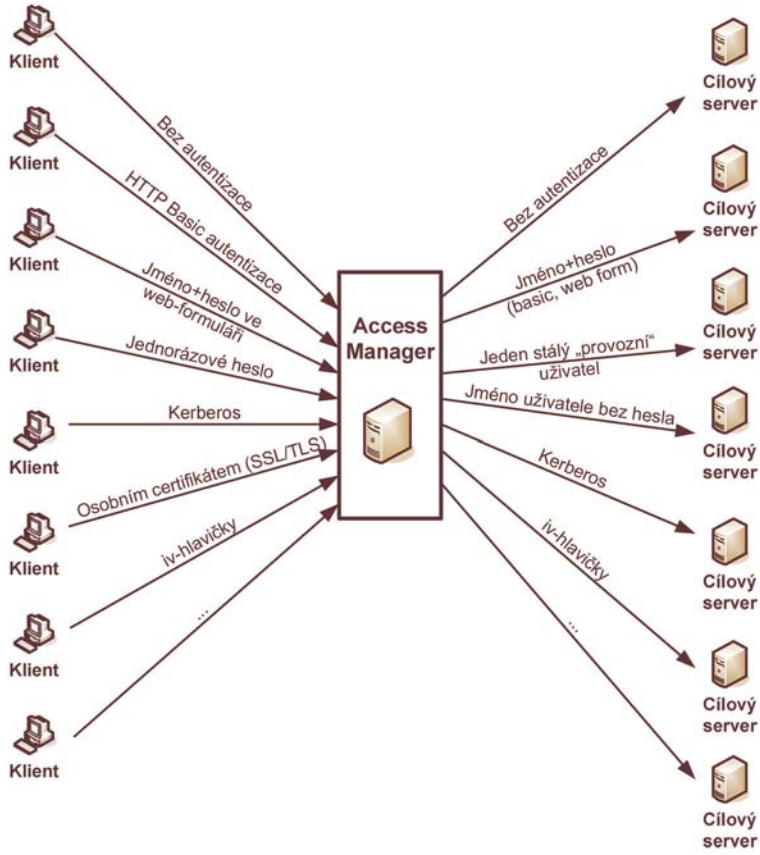
Autentizace se nám rozpadá na:

- Autentizaci klienta vůči Access Manageru.
- Autetnizaci Access Manageru vůči cílovému serveru.

5.1 Autetnizace klienta vůči Access manageru

Zde máme celou škálu možností:

- Bez autetnizace (anonymní přístup).
- Autetnizace stálým heslem a jménem. Zde máme několik možností, kam jméno a heslo umístit:
  - Do HTTP hlavičky Authorization v případě autetnizační metody Basic (viz samostatný odstavec).
  - Do webového formuláře. Tato možnost je zajímavá zejména pokud chceme jméno a heslo zpracovávat přímo aplíkací (nikoliv webovým serverem jako takovým).
- Autentizace jednorázovým heslem, byť významně bezpečnější, je z hlediska protokolu HTTP jen variací na stálé heslo.
- Protokolem Kerberos (viz samostatný odstavec).
- Osobním certifikátem (viz samostatný odstavec).



- Pomocí iv-hlaviček (viz samostatný odstavec).

5.1.1 Metoda Basic

Klient sice může přímo do URL zapsat jméno uživatele a heslo, to je však málo běžné. Běžnější je dialog, kdy klient nezadá žádné autentizační informace a server vrátí chybovou hlášku:

```
HTTP/1.1 401 Unauthorized

WWW-authenticate: autent_metoda realm="řetězec",
                  případné_další_parametry
```

Kde první parametr `autent_metoda` je typ autentizační metody, kterou server vyžaduje. Řetězec v parametru `realm` bude zobrazen klientovi, aby věděl k ja-

kému objektu se má autentizaovat. Konečně některé autentizační metody mohou používat další parametry. Autentizační metoda Basic další parametry nepoužívá. Pokud server podporuje více autentizačních metod, pak pro každou vrátí jednu hlavičku `WWW-authenticate`.

Autentizační metoda Basic je určena autentizaci jménem a heslem. Tato metoda přenáší sítí jméno a heslo v textovém (nezabezpečeném tvaru). Autentizační dialog pak probíhá např. tak, že klient se pošle dotaz na server bez jakékoliv autentizace (C = Client, S = Server):

```
C: GET /soubor HTTP/1.1
C: Host: www.firma.cz
C:
S: HTTP/1.1 401 Unauthorized
S: WWW-authenticate: Basic realm="www.firma.cz"
S:                                          ...další hlavičky
```

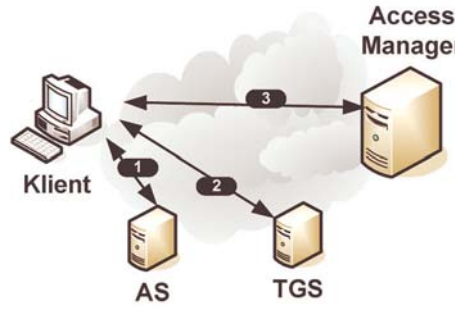
Tj. server dotaz klienta odmítne s tím, že vyžaduje autentizaci. Do své odpovědi ale server přidá hlavičku `WWW-authenticate` ve které nabídne autentizační metodu (v tomto případě Basic) a dodá řetězec, který se má zobrazit uživateli (aby uživateli např. věděl které heslo má zvolit). Klientský software zjistí, že se jedná o autentizační metodu Basic, tj. autentizaci jménem a heslem. Vyzve uživatele k zadání jména a hesla. Výsledek pak vloží do hlavičky Authorization:

```
C: GET /soubor HTTP/1.1
C: Host: www.firma.cz
C: Authorization: Basic RG9zdGFsZWs6aGVzbG8=
C:
S: HTTP/1.1 200 OK
S: ...
```

Tj. klient po obdržení „`HTTP/1.1 401 Unauthorized`“ provedel následnou autentizaci jménem a heslem. Jenže server může nabízet větší množství objektů a ke každému z nich můžeme použít jinou autentizaci. Proto server vrací řetězec „`realm`“, aby prohlížeč mohl uživateli do dialogového okna zobrazit k jakému objektu má zadat jméno a heslo. Z jména a hesla je vytvořen řetězec. Jméno a heslo jsou odděleny dvojtečkou (např. `Dostalek:heslo`). V hlavičce Authorization se nepřenáší řetězec přímo, ale kódován Base64, tj.

```
Base64(Dostalek:heslo)="RG9zdGFsZWs6aGVzbG8="
```

Komukoliv, kdo na cestě od klienta k serveru odchytí hlavičku Authorization, tak stačí řetězec `RG9zdGFsZWs6aGVzbG8=` dekódovat `Base64` (např. programem OpenSSL) a získá heslo.



5.1.2 Autentizace protokolem Kerberos

V případě autentizace protokolem Kerberos se klient nejprve autentizuje vůči Autentizačnímu serveru Kerbera a získá lístek pro vydávání lístků (1) a na jeho základě obdrží od služby TGS lístek pro přístup ke službě „Access Manager“ (2). Tento lístek jej pak opravňuje k využívání (k autetnizaci) vůči službě „Access Manager“ (3).

A nyní jak proběhne autentizace v protokolu HTTP v případě autetnizační meteody Kerberos? Microsoft pro tento případ zavedl typ autentizace SPNEGO (*Simple and Protected Negotiate*). Jeho princip spočívá v tom, že server v hlavičce `WWW-authenticate` nabídne server klientovi autentizační metodu `Negotiate`. Klient pak ve své odpovědi uvede hlavičku `Authorization` opět s dvěma parametry:

- Prvním parametrem je název autentizační metody, tj. v tomto případě řetězec „`Negotiate`“.

- Druhým parametrem je tzv. SPNEGO token, který obsahuje lístek protokolu Kerberos (variantně je podporována i klasická NTLM autentizace).

Vyžití lístků protokolu Kerberos elegantně umožňuje klientům přihlášeným do domény Windows se následně přihlašovat např. na Access manager (webový server) běžící na UNIXu metodou *Single Sign On*. Tj. aby uživatel přihlášený do domény Windows nemusel znovu zadávat přihlašovací informace při přístupu na webové servery (i UNIXové).

Způsob autetnizace protokolem Kerberos/SPNEGO se využívá zejména na intranetech. Dnes již klasickým případem je autetnizace zaměstnanců do domény Aktive Directory pomocí PKI čipové karty. Čím klient získá i příslušné lístky pro protokol Kerberos, kterými se následně autentizuje vůči Access Manageru právě pomocí protokolu Kerberos/SPNEGO.

5.1.3 Autentizace osobním certifikátem (SSL/TLS)

Pokud nad Access Managerem běží SSL/TLS server, pak můžeme využít autentizaci osobním certifikátem.

Mnozí se domnívají, že autentizace osobním certifikátem výrazně zdržuje klienty. Na tomto místě je třeba uvést na pravou míru, že to tak být nemusí. Většina HTTPS serverů je totiž nakonfigurovaná tak, že na počátku komunikace klienta se serverem se zřídí tzv. SSL relace. Při další komunikaci s týmž serverem se pak SSL relace jen obnovuje. Obnovení relace je již výrazně rychlejší neboť není třeba soukromého klíče klienta, který je umístěn např. na čipové kartě.

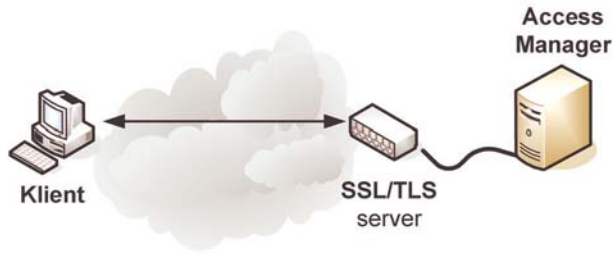
5.1.4 Autetnizace pomocí iv-hlaviček

SSL/TLS servery požírají značné výpočetní zdroje pro kryptografické operace. Řešením je specializovaný kryptografický hardware, který realizuje samotný SSL/TLS server. Takový hardware může být realizován dvěma různými způsoby:

- Jako deska do serveru, pak se jedná o rozšíření samotného serveru. Tento případ nemá na síťovou komunikaci žádný vliv – musíme jen nainstalovat příslušný ovladač pro náš server.

- Jako externí síťové zařízení. Příkladem jsou SSL/TLS servery implementované přímo ve směrovačích CISCO (v CISCO terminologii ukončovače tzv. SSL/TLS tunelu).

Druhý případ je zajímavý tím, že pokud se ukončí SSL/TLS komunikace („SSL/TLS tunel“) před Access Managerem, pak na Access Manager dojde „jen“ HTTP komunikace. Veškeré informace o SSL/TLS relaci zůstanou na externím SSL/TLS serveru.



Aby se tyto informace nezahodily, tak externí SSL/TLS servery doplňují HTTP hlavičky o speciální hlavičky začínající řetězcem „iv-“. Do těchto hlaviček pak ukládají např.:

- Předmět certifikátu klienta.
- Vydavatele certifikátu klienta.
- Případně i celý certifikát klienta.
- IP adresu klienta.
- Identifikaci SSL/TLS relace – to je velice důležité pro případný load balancing mezi baterií Access Managerů.

Server pak přebírá informace z těchto hlaviček a sám je již neprověřuje – důvěřuje předřazenému SSL/TLS serveru.

5.2 Autetnizace Access Manageru vůči cílovému serveru

Pokud klient nebyl anonymní, pak je často velice důležité přenést identitu klienta až na cílový server. Pokud cílový server Access Manageru důvěřuje, pak již klienta nemusí prověřovat, ale pouze zpracuje jeho identitu. Což je právě principem Single Sing On.

V zásadě zde lze rozlišovat následující případy:

- Identita klienta není pro cílový server významná. Cílový server buď slouží všem anonymním klientům nebo důvěřuje Access Manageru, že na něj nepustí žádné klienty, kterým nemá sloužit. V takovém případě Access Manager vystupuje vůči cílovému serveru jako anonymní klient nebo jako stále stejný „provozní uživatel“.
- Identita klienta je pro cílový server důležitá – potřebuje ji pro poskytování svých služeb. V takovém případě identita může být předána:
  - Jako jméno uživatele metodou Basic nebo webovým formulářem. V případě, že Access Manageru důvěřujeme, pak může být identifikace klienta předávána již bez hesla.
  - V iv-hlavičkách, pokud se uživatel autentizoval osobním certifikátem (Access manager pracuje jako externí ukončení SSL/TLS tunelu). Přitom iv-hlavičky mohou být využity i např. pro předání IP adresy kleinta, identifikace SSL/TLS relace a to i v případě, že se jednalo o anonymního klienta protokolu SSL/TLS.
  - Promocí protokolu Kerberos (např. proxy tikety). S touto možností jsme se však v praxi nesetkal.

Literatura

[1] Dostálek, L., Vohnoutová, M. *Velký průvodce PKI a technologií elektronického podpisu.* Praha : Computer Press, 2006.

Implementace Access Managementu s open source produkty

Martin Čížek

E-mail: martin@cizek.com

Abstrakt

Access management je možné implementovat různými způsoby. Některá řešení jsou založena na komplexních komerčních produktech, jiná jsou kombinací menších projektů, kde každý plní svoji funkci. Rozhodneme-li se vydat cestou vlastní integrace dostupných balíků software, vezmeme tím na sebe určitou odpovědnost, ale získáme mnoho možností a velkou flexibilitu. Některými možnostmi implementace access managementu s využitím open source software se zabývá tento příspěvek.

1 Úvod

Příspěvek si klade za cíl dát konkrétní podobu řešení Single Sign On (SSO) a řízení přístupu. Nabízené řešení je jedno z mnoha a nesnaží se být řešením univerzálním.

2 Požadavky na řešení

Hlavní motivaci nasazení centrálně spravovaného řízení přístupu popisují ostatní příspěvky. Také taxonomii integrovaných aplikací a popis klíčových problémů aplikací lze nalézt např. v [mv-iam2007]. Tento příspěvek se zaměřuje na analýzu a implementaci konkrétního případu.

V modelové organizaci example.org [rfc2006] je nasazeno mnoho různých služeb implementovaných jako webové aplikace. Protokol HTTP se používá k přístupu ke všem informačním systémům. Informační systémy jsou provozovány na různých webových serverech a různých platformách. Některé jsou dostupné přímo, některé přes reverzní proxy (viz též [ld-rpam2007]). Většina aplikací je vyvinutá přímo organizací, další jsou open source. Případné proprietární aplikace

lze nakonfigurovat tak, aby přebíraly autentizaci z webového serveru/kontejneru, případně lze toto dojednat s jejich dodavateli.

Popsaná situace představuje určité zjednodušení, nicméně je poměrně častá v organizacích jako jsou univerzity nebo menší a střední firmy. Největším potenciální problém představují špatně konfigurovatelné uzavřené proprietární aplikace. Jejich případ lze individuálně řešit implentací konektorů, které budou převádět jednotné předávání identifikačních a autorizačních údajů na způsob, jemuž rozumí.

Problém ve výše popsané situaci je pochopitelně roztříštěnost autentizačních, autorizačních a auditních mechanizmů. Každý informační systém udržuje svoji databázi přístupových údajů, svoje seznamy rolí a své mapování uživatelů na role. Ve výsledku je celá struktura špatně udržovatelná a dříve nebo později v ní vznikne nepořádek. Dalším faktorem je bezpečnostní riziko – v případě prolomení hůře zabezpečených systémů se útočník dostane k přístupovým údajům, typicky použitelným v ostatních informačních systémech. S tím také souvisí nemožnost jednotného nastavení bezpečnostní politiky jako platnost hesel apod.

Situace má dopad i přímo na koncové uživatele – jednak si přístupy do jednotlivých informačních systémů uživatel obvykle musí zařídit jeden po druhém, jednak se do každého systému musí znovu přihlašovat při každém jeho použití.

Po zhodnocení současné situace lze přejít k vlastní formulaci požadavků:

1. Autentizační databáze nemají být duplikovány pro každou aplikaci. Autentizačních databází ale může být více (např. externí a interní uživatelé).

   V prezentovaném příkladu se počítá s autentizací proti databázi LDAP, v budoucnu však bude vhodné navíc využít Kerberos pro interní uživatele.

2. Jednotná databáze rolí/privilegií uživatelů. Tento požadavek obvykle není jednoduché splnit až na úroveň rolí v rámci konkrétních informačních systémů. Postačí, když bude možné pomocí privilegií řídit alespoň přístup uživatelů k celým aplikacím. Využití rolí z centrální databáze tedy nebude povinností, ale současně řešení musí tuto možnost poskytovat.

   Pro implementaci tohoto požadavku lze opět předpokládat databázi LDAP.

3. Centrální autentizační brána s možností různých autentizačních mechanismů.

4. Mechanizmus držení sessions pro aplikace – po úspěšném použití centrální autentizační brány není nutné po dobu platnosti session na ni znovu přistupovat. Tento požadavek je součástí koncepce SSO.

5. Aplikační servery, na nichž jsou provozovány informační systémy, nemají mít přístup k autentizačním údajům uživatele.

6. Možnost sledovat přístupy uživatelů pro definované skupiny aplikací.

3 Dostupné produkty

Autentizační a autorizační služby jsou poměrně rozšířené, lze například využít HTTP server Apache s mod_ldap a dalšími moduly nebo proxy server Squid s externím autentizátorem a propracovaným systémem ACL. Samozřejmě je žádoucí, abychom si zvolením nějakého produktu nezavřeli dveře k možnostem, které nabízí další software.

Nejspecifičtějším požadavkem je zřejmě udržování sessions s aplikacemi a SSO, proto je vhodné začít odsud. Mezi dostupné možnosti patří:

JA-SIG Central Authentication Service (CAS) [cas] je autentizační systém původně vytvořený na Yale University. Je napsaný v Javě a provozuje se v Servlet Containeru Tomcat (pravděpodobně i jiných). Obsahuje moduly pro různé aplikační servery a bezpečnostní moduly (např. Acegi security).

Princip je podobný Kerberu, místo lístků jsou používány cookies – TGC (Ticket Granting Cookie) a ST (Service Ticket).

Java Open Single Sign-On (JOSSO) [josso] je open source SSO infrastruktura založená na J2EE. Využívá JAAS, klientské moduly existují také pro ASP a PHP. Obsahuje komponentu s reverzní proxy.

Collaborative Single Sign-On (CoSign) [cosign] se skládá ze tří komponent – démona, Webloginu (CGI) a filtrů.

Démon poskytuje hlavní funkčnosti, zejména udržování stavu všech sessions a auditní služby. Weblogin slouží jako centrální přihlašovací služba, dále zajišťuje svázání tzv. login cookie (tj. cookie přihlašovacího serveru) s cookie služby. Cookie služby ověřuje aplikační server proti démonu při všech přístupech. Takto zvolená architektura umožňuje Single Sign-Out. Poslední komponenta, tedy filtry, je umístěna na aplikačních serverech a zajišťuje zabezpečení služeb a případné přesměrování na Weblogin. Podporovány jsou servery Apache 1 a 2.

Pubcookie [pubcookie] se skládá ze samostatného přihlašovacího serveru a modulů pro aplikační servery Tyto umožňují využití existující autentizační služby jako Kerberos, LDAP a NIS k SSO přihlášování do webových aplikací v instituci.

Shibboleth [shibboleth] je open source middleware poskytující SSO. Shibboleth implementuje Osasis SAML [saml].

Stanford WebAuth [webauth] byl vyvinut na Stanford University a je implementován pomocí modulů serveru Apache a CGI skriptů. Je navržen, aby

využil existující infrastrukturu Kerberos, nicméně použití Kerbera k autentizaci uživatelů není podmínkou (lze zvolit jakoukoliv metodu dostupnou serveru Apache).

Architektura WebAuth se skládá z tzv. WebKDC – analogie KDC, login serveru (CGI skriptů) a WAS (WebAuth-enabled Application Servers). Řešení je velmi podobné např. CoSign s rozdílem, že v cookies se ukládádají pověření šifrovaná symetrickými klíči, a není tedy potřeba komunikace mezi WAS a WebKDC k ověření přístupů.

Open Web SSO (OpenSSO) [opensso] je založen na kódu Java System Access Manager. Jedná se o komplexní produkt, podporující centralizované autentizační služby, Federated Identity, JAAS, Kerberos, LDAP. Skládá se ze čtyř modulů – Access Manager, Open Federation Library, Open Federation a J2EE Agents.

Z nabízených možností nakonec padla volba na Webauth, neboť je zakomponován do Apache, což umožní využít možnosti tohoto serveru, navíc Apache je velmi rozšířený a podporovaný. Další důvod je, že za běžného provozu nekomunikují aplikační servery s démonem pro správu sessions (lepší škálovatelnost) a v neposlední řadě je plusem relativní jednoduchost tohoto řešení.

4 Princip řešení WebAuth

Schéma průchodu prvního požadavku systémem je naznačeno na obrázku 1.

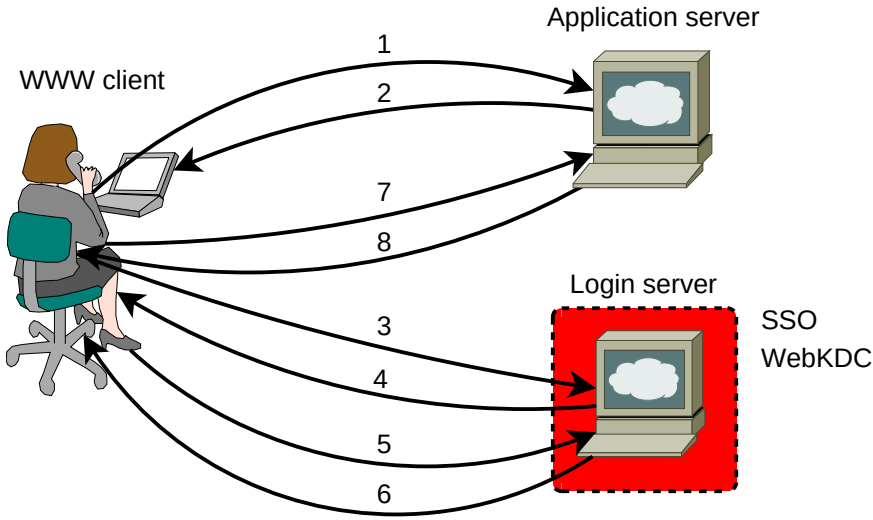
Následující body popisují zjednodušeně situaci, kdy je login server nakonfigurován tak, aby autentizaci prováděl Apache. Pokud je autentizace prováděna přes Kerberos, je situace jen málo odlišná. Způsoby autentizace je možné i kombinovat, více viz domovská stránka projektu.

1. Uživatel přistupuje ke službě bez aplikačního tokenu. Jelikož mod_webauth vyžaduje autentizaci, nepropustí požadavek dále a připraví request token pro id token. Request token je zašifrován session klíčem sdíleným mezi WAS a WebKDC a obsahuje mj. návratovou adresu. WAS získává session klíč z tzv. webkdc service tokenu [wats].

2. WAS pošle klientu přesměrování na login server, přičemž request token je parametrem URL.

3. Klient přistoupí na login stránku serveru a

4. je vyzván k zadání přihlašovacích údajů.



Obr. 1 Průchod požadavku systémem

5. Přihlašovací údaje jsou poslány login serveru, a jsou-li správné požadavek doputuje login skriptu. Ten zjistí identitu uživatele z proměnné REMOTE_USER, ozšifruje requset token, ověří, že je validní, a vyžádá si z WebKDC id token, který je šifrován klíčem session.

6. Uživateli je zobrazena stránka s potvrzením úspěchu a odkazem na původní službu. Odkazu obsahuje id token jako parametr URL.

7. Při dalším přístupu na WAS mod_webauth zpracuje URL, v němž je zakódován id toke a vytvoří aplikační token (šifrován klíčem WAS) a

8. pošle jej uživateli jako cookie s reditectem na původně odkazovanou URL.

5 Postup implementace

Uvedený postup instalace má cesty platné pro distribuci Debian GNU/Linux, ovšem je použitelný obecně. Postup na jiných OS se bude pravděpodobně lišit jen v cestách k některým souborům.

Instalace Nejprve WebAuth nainstalujeme, k tomu obvykle poslouží nástroje operačního systému – postup instalace ze zdrojových kódů lze nalézt na stránkách projektu [webauth]. V distribuci Debian GNU/Linux se jedná o balíky

libapache2-webauth (instaluje se na aplikačních serverech), libapache2-webkdc a webauth-weblogin (instalují se na WebKDC/login serveru).

Kerberos I když Kereberos nebude použit k autentizaci přistupujících uživatelů, je potřeba jej nakonfigurovat pro autentizaci WAS proti WebKDC a WAS proti LDAPu. V Kerberu je nutné zřídit účet služby WebKDC, služby LDAP a účty WAS (kompletní postup včetně instalace Kerbera, LDAPu apod. naleznete na [mc-webauth]). Účty pro WAS by měly začínat předponou `webauth/`. Po vytvoření těchto účtů exportujeme tabulky klíčů (keytabs), přičemž keytab WebKDC umístíme do `/etc/webkdc/keytab` a klíče pro WAS do adresářů `/etc/webauth/keytab` na příslušných aplikačních serverech. Ke klíčům musí mít přístup ke čtení uživatel, pod nímž běží Apache.

Nastavení WebKDC Jelikož nastavení se provádí na serveru Apache, můžeme všechny vlastnosti, které nám tento software nabízí. Samotné nastvení WebKDC může být následující:

```
LoadModule webkdc_module /usr/lib/apache2/modules/mod_webkdc.so
WebKdcServiceTokenLifetime 30d
WebKdcKeyring /var/lib/webkdc/keyring
WebKdcKeytab /etc/webkdc/keytab
WebKdcTokenAcl /etc/webkdc/token.acl
# Pro ladeni
WebKdcDebug on
LogLevel debug

# WebKDC je implementovano jako handler
<Location /webkdc-service>
    SetHandler webkdc
</Location>

<Directory "/usr/share/weblogin">
    AllowOverride All
    Options Indexes FollowSymlinks +ExecCGI
    Order allow,deny
    Allow from all
    # V~produkcnim nastaveni se povoli jen SSL pristup
</Directory>
ScriptAlias /login "/usr/share/weblogin/login.fcgi"
ScriptAlias /logout "/usr/share/weblogin/logout.fcgi"
Alias /images "/usr/share/weblogin/generic/images/"
Alias /help.html "/usr/share/weblogin/generic/help.html"
```

```
# Nastaveni prihlasovani (nevyuzivame-li Kerberos)
<Location /login>
    AuthType Basic
    AuthName "Webkdc auth"
    # Pro testovaci ucely staci Basic autentizace proti souboru,
    # pro nasazeni nahradime s mod_ldap. Lze pouzit imod_auth_kerb
a~# autentizaci SPNEGO.
    AuthUserFile /tmp/testauth
    Require valid-user
</Location>
```

Nakonec ještě musíme zajistit, aby WAS měl možnost získat id token. To je definováno v souboru `/etc/webkdc/token.acl`, uvedeném v konfiguraci výše:

```
# Povoleno kazdemu s krb5 principalem webauth/*@EXAMPLE.ORG
krb5:webauth/*@EXAMPLE.ORG id
```

Nastavení login serveru V přechozím odstavci byl nakonfigurován skript pro logování. Tento skript má také svoji konfiguraci `/etc/webkdc/webkdc.conf`, která může vypadat:

```
our $KEYRING_PATH = ‘ /var/lib/webkdc/keyring‘ ;
our $TEMPLATE_PATH = ‘ /usr/share/weblogin/generic/templates‘ ;
# Kde se nachazi sluzba WebKDC
our $URL = "http://webkdc.example.org/webkdc-service/";
# Rika, ze skript ma identitu uzivatele prebirat ze standardni
# promenne Apache REMOTE_USER. Jinak by skript sam delal
# autentizaci uzivatele proti KDC.
our $HONOR_REMOTE_USER = 1;
```

Nastavení WAS Opět platí, že můžeme využít vlastnosti, které nám Apache nabízí. Informaci o identitě přistupujícího uživatele modul nabízí skriptům v proměnné prostředí `WEBAUTH_USER`, resp. standardní `REMOTE_USER`. Je možná též integrace s AJP pro server Tomcat.

Samotné nastavení WebAuth může být:

```
LoadModule webauth_module /usr/lib/apache2/modules/mod_webauth.so
WebAuthKeyring /var/lib/webauth/keyring
WebAuthKeytab /etc/webauth/keytab
WebAuthServiceTokenCache /var/lib/webauth/service_token_cache
# Pro testovani, abychom mohli sniffovat komunikaci. V produkci
# se poouzije vyhradne SSL.
```

```
WebAuthRequireSSL off
WebAuthSSLRedirect off

WebAuthLoginURL "http://webkdc.example.org/login/"
WebAuthWebKdcURL "http://webkdc.example.org/webkdc-service/"
# Jmeno sluzby sluzby WebKDC pro ziskani service ticketu
WebAuthWebKdcPrincipal service/webkdc

<Location />
    AuthType WebAuth
    Require valid-user
</Location>
```

Otestování základní konfigurace Při přístupu na aplikační server bychom nyní měli být přesměrování na login server, který by měl požadovat basic autentizaci. Je-li úspěšná, měla by se zobrazit stránka s informací o úspěchu a odkazem, který vede zpět na aplikační server. Součástí odkazu je token, na jehož základě později WAS pošle klientu cookie s app tokenem.

Autorizace pomocí LDAP K zajištění autorizace využijeme podporu LDAP dodávanou v modulu WebAuthLdap. Modul se instaluje do aplikačního serveru s WAS. Autorizace je prováděna ověřením uživatelovy přítomnosti v alespoň jedné z požadovaných rolí. Příklad definice role ve formátu LDIF:

```
dn: cn=APP1,ou=groups,dc=example,dc=org
objectclass: top
objectclass: groupOfNames
cn: APP1
member: uid=tester, ou=people, dc=example,dc=org
```

Konfigurace pro naznačenou strukturu rolí je např.:

```
LoadModule webauthldap_module
           /usr/lib/apache2/modules/mod_webauthldap.so
# Pro prihlaseni do LDAPu pouzijeme klic WAS
WebAuthLdapKeytab /etc/webauth/keytab
WebAuthLdapTktCache /var/lib/webauth/krb5cc_ldap
WebAuthLdapHost ldap.example.org
# Kde se bude vyhledavat
WebAuthLdapBase ou=groups,dc=example,dc=org
# V~jakem atributu je obsazeno jmeno privilege group
WebAuthLdapAuthorizationAttribute cn
# Filtr, kterym se vybiraji privilege groups pro uzivatele,
```

```
# retezec USER je nahrazen jmenem pristupujiciho uzivatele.
WebAuthLdapFilter member=uid=USER,ou=people,dc=example,dc=org
# Vicehodnotove hodnoty LDAP atributu exportovane do promennych
# prostredi budou oddeleny carkou.
WebAuthLdapSeparator ,
# Hodnoty atributu cn (nazvy privilege groups) budou skriptum
# dostupne v~promenne Apache WEBAUTH_LDAP_CN (oddelene carkou):
WebAuthLdapAttribute cn

# Vlastni zabezpeceni:
<Location />
    AuthType WebAuth
    Require privgroup APP1
</Location>
```

Nastavení proxy Pro integraci jiných aplikačních serverů než Apache, resp. Tomcatu s AJP konektorem je možné použít reverzní proxy. Použití proxy přináší i další výhody – usnadňuje centralizaci auditu požadavků, centralizuje konfiguraci a zjednodušuje život správcům cílových aplikačních serverů. Proxy samozřejmě mohou být používány v kombinaci s virtuálními servery. To navíc umožňuje jemnější (per virtual host) konfiguraci některých atributů, jako např. `WebAuthLdapFilter`, který nemůže být zadán na úrovni adresáře.

Následující příklad vyžaduje mod_proxy a mod_headers:

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName app1.example.org
    ProxyRequests Off
    <Location />
        AuthType WebAuth
        Require privgroup APP1
        # Cilovy server je pochopitelne potreba zabezpecit
        # pred primym pristupem.
        ProxyPass        http://realapp1.example.org:8080/
        ProxyPassReverse http://realapp1.example.org:8080/
        # Identitu uzivatele a~seznam jeho roli cilovemu serveru
        # predavame v HTTP hlavickach:
        RequestHeader set "X-WEBAUTH-USER"  "%{WEBAUTH_USER}e"
        RequestHeader set "X-WEBAUTH-PRIVS" "%{WEBAUTH_LDAP_CN}e"
    </Location>
</VirtualHost>

# Konfigurace ostatnich virtualnich serveru je analogicka.
```

Jak příklad naznačuje, na cílovém serveru by měly být údaje o přistupujícím uživateli a jeho privilegiích dostupné v HTTP hlavičkách.

6 Shrnutí

Zhodnotili jsme několik open source produktů pro implementaci SSO (WebISO), řízení přístupu a integrace aplikačních serverů do takové struktury. Jako příklad implementace jsme nakonfigurovali prostředí s produktem Stanford WebAuth, webovým serverem Apache a dalšími podpůrnými balíky. Ačkoliv toto řešení není jediné možné, mělo by demonstrovat princip většiny ostatních způsobů implementace.

Literatura

[mv-iam2007] Vohnoutová, M. Identity a Access Manager, In *EurOpen Jaro 2007*.

[ld-rpam2007] Dostálek, L. Reverzní proxy neboli Access manager, In *EurOpen Jaro 2007*.

[cas] JA-SIG Central Authentication Service
http://www.ja-sig.org/products/cas/

[josso] Java Open Single Sign-On Project
http://www.josso.org/

[cosign] CoSign – Collaborative single sign-on
http://www.umich.edu/∼umweb/software/cosign/

[pubcookie] Pubcookie: open-source software for intra-institutional web authentication
http://www.pubcookie.org/

[shibboleth] Shibboleth Project – Internet2 Middleware
http://shibboleth.internet2.edu/

[webauth] Stanford WebAuth
http://www.stanford.edu/services/webauth/

[opensso] Open Web SSO
https://opensso.dev.java.net/

[rfc2006] RFC 2006 – Reserved Top Level DNS Names
http://www.rfc-editor.org/rfc/rfc2606.txt

[saml] Oasis SAML
http://www.oasis-open.org/committees/
tc_home.php?wg_abbrev=security

[mc-webauth] Stránka o zprovoznění WebAuth
http://www.cizek.com/cs/linux/im-am/webauth

[wats] WebAuth V3 Technical Specification (Protocol)
http://www.stanford.edu/services/webauth/protocol.html

Trusted Archive Authority – Long Term Trusted Archive Services

Aleksej Jerman Blazic

E-mail: aljosa@setcce.si

Abstract

With the transition to paperless environment the e-business and e-government processes demand to prove the existence of data at a specific point of time and to demonstrate the integrity of the data since that time during long term periods is becoming of utmost importance. Through the course of time the true value of electronic content may simply evaporate due to technological progress and incompetence to prove the authenticity of data stored. This paper presents an approach and a solution to the problem of long-term trusted preservation of electronic data for dematerialized business processes. The basic system design and the architecture of a long-term trusted electronic archive service is introducing a protocol and an evidence record syntax that is being standardized within the IETF Long Term Archive and Notarization Service Working Group. System environment for long term archiving is based on PKI-enabled services for creation of long term integrity proofs. Furthermore, the solution contains additional class of services that meet the requirements for long term validity of digitally signed and legally binding documentation.

1 Introduction

The existence of information in electronic form is undermined by continual change and progress on a number of organizational and technological fronts. The original environment where digital data is created may change or simply cease to exist. Used formats may become obsolete and completely unreadable. Evidences for data integrity provision may evaporate during the time and disprove data authenticity. Legal e-business or e-government environments where electronic data have the most important roles are unable to adopt full dematerialization of these processes as long as time exposes threats to trustworthiness and applicability of information preserved in electronic form.

The nature of digital objects requires constant and perpetual maintenance as they depend on elaborate hardware and software systems with defined data

and information models, and on technology standards that are upgraded or replaced regularly. The ability to rely on digital records is an issue of increasing concern with the rise of formal e-commerce and e-government and their proliferation. How such records are understood, used, preserved, and verified over time is highly contingent upon the juridical-administrative, procedural, provenance, documentary, and technological contexts.

The current technical community is looking for standardized solutions to mitigate such issues. One of the approaches is the introduction of a Trusted Archive Service (TAS) that will address the relevant multi dimensional issues: data management, long-term integrity and authenticity, storage media lifetime, disaster planning, advances in cryptosystems or computational capabilities, changes in software technology, legal issues, etc. A long-term Trusted Archive Service provides a stable environment for preservation of digital data over long periods of time through a regimen of technical and procedural mechanisms. Such services must periodically perform activities that assure content readability and media accessibility and preserve data integrity and non-reputability of data existence. Furthermore, the Trusted Archive Service is provides necessary mechanisms for maintaining the evidences on the validity of digital signatures as their validity is decomposing through time due to cryptographic and formal constrains. Therefore, a primary goal of a long-term Trusted Archive Service is to support the credible assertion of a claim that is currently asserted, at points well into the future meaning that the service provides enough evidence to demonstrate the existence of archived data at a given time and the integrity of the archived data since that time.

2 Long Term Stability of Electronic Records

In ever changing environments, digital objects may not only loose their practical and content value but may also suffer from the lack of integrity and authenticity or non-repudiation proof. Preservation of digital objects is hence a multidimensional conceptual and technological challenge. Different motivations associated with the record keeping have already resulted in technological strategies for storage, processes, maintenance, readability, etc. The main challenges of the past research on long-term electronic archiving were focused towards definition of a system environment for data storage, accessibility and interpretation. An example of such initiatives is the Open Archive Information System (OAIS), currently recognized as ISO standard, which determines a stable framework for long-term document preservation, while functional requirements for long term archiving are summarized in (governmental perspective of) the Model Requirements for the Management of Electronic Records (MoReq).

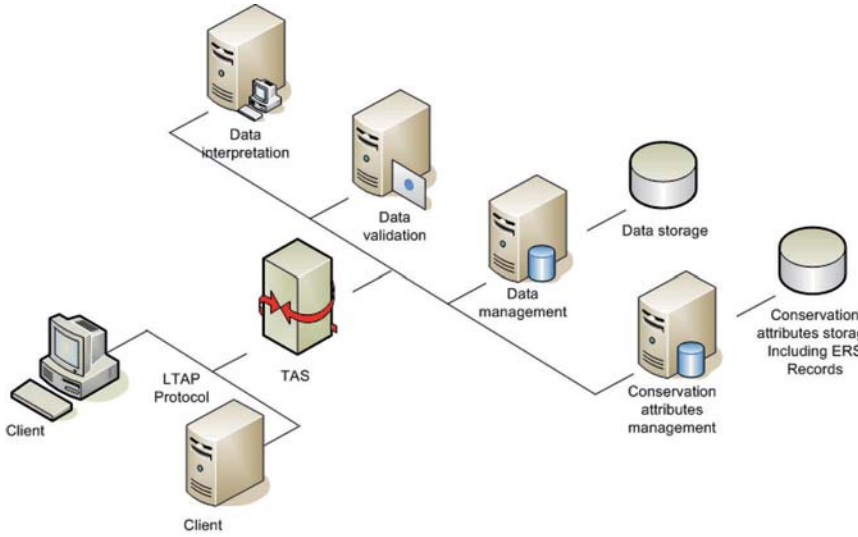


Figure 1 General structure of the TAS service for long term data preservation

The current managerial and storage capabilities of electronic data, long term readability strategies, etc., are therefore well understood and integrated in the up-to-date information systems. The present concerns of long term record keeping are focused on the problems of digital objects integrity and proof of existence. Specific techniques are required that can prove that a data object (electronic document) existed at a certain time in the past and was not altered since after. Such issue is becoming especially important with the proliferation of (legally binding) documents that need to be archived on (formally) defined periods of time. To prove authenticity and ownership of documents, such records may contain digital signatures or time-stamps. The value of such evidence may simply evaporate through time for many reasons: non-existing revocation information on issued digital certificate guaranteeing the authenticity of the signatory (due to termination of a certificate authority), expiration or revocation of a certificate (due to key loss or damage associated with a digital signature, or weaknesses of used cryptographic algorithms), etc. This implies the need of a service with reliable mechanisms that will provide the necessary evidences of security assertions validity well into the future.

Long-term Trusted Archive Service provides long-term operation regardless of the means used. Rather than complete architectural solution such service is characterized and defined as an aggregator of standardized techniques which provide a user with the single interface that meets its principal demand for long term data integrity provision of the archived documents. In other words, the TAS service simply combines techniques for data storage, data management and

data transformation with the evidence and integrity service functionalities. The TAS service delivers a complete solution for record keeping on a long term basis as an important functional and technological element of dematerialized business and governmental processes.

Characteristics of trusted electronic archive are documented and summarized in interaction, storage and management, integrity and evidence provision by use of service specific protocol. Hence the latest work in the field of electronic data preservation deals mainly with means that deliver proof and evidence of data existed and processes executed in the past using data integrity and time evidence techniques. Technical realization of such requirements is currently interpreted through standardization attempts of IETF LTANS WG. Techniques defined, developed and integrated for these purposes are the Long Term Archive Protocols and the Evidence Record Syntax.

3 Requirments and Operation of the TAS Service

Requirements for archive system may differ between service users, mainly with strong relation to the characteristics of business performed. The TAS service plays a key role of final resorts for electronic records regardless of their type and content. The key functionality delivered by the TAS service is retrieval and storage of data that may come in any form, raw, signed or encrypted. Using supporting infrastructure the TAS service then performs perpetual maintenance to assure data access and interpretation and to demonstrate data integrity on a long term basis.

3.1 Data submission, retrieval and deletion

A TAS service is designed to retrieve different types of data. Such data may come in different forms and formats, while the TAS service concept recognizes two general data types: unsigned (raw) and signed data. Both data types are usually provided in plain or encrypted form (e.g. for confidential purposes). Different data types have impact on the TAS service performing. Signed data require additional preservation related information to prolong the validity of signatures applied, while the general preservation concepts and procedures remain the same.

A TAS service handles archived data in a way that stored data are identified and retrieved at any time using distinguishing labels (identifiers). Using TAS service functions is strongly related to user interaction with the service. Hence a mechanism to enable authentication and authorization is usually needed and may be borrowed from underlying protocols.

It is important to enable formalization of all processes performed between users and services in a way that attestation is delivered on objects inserted in the archive. Such proofs are usually not required immediately but needed at least to define liabilities in the archiving process.

3.2 Management of archived data objects

A long-term archive service permits clients to request the following basic operations:

- specify an archive period for submitted data objects
- extend or shorten defined archive period for an archived data object
- specify metadata associated with an archived data object
- specify an archive policy under which the submitted data should be handled

Some characteristics of archive process may be modified on user's request such as possibility to extend or shorten the archive period after the initial submission. It is also possible to express an archive period in terms of time, an event or a combination of time and event. Users are also able to specify metadata that, that can be used to enable retrievers to render the data correctly, to locate data in an archive or to place data in a particular context.

3.3 Provide evidence records that support demonstration of data integrity

The TAS service is capable of providing evidence that can be used to demonstrate the integrity of data for which it is responsible from the time it received the data until the expiration of the archive period. This is achieved by providing evidence records that support the long-term non-repudiation of data existence and demonstrate integrity since a specific point in time. Such evidence record contains sufficient information to enable the validity of an archived data object's characteristics to be demonstrated (to an arbitrator). Evidence records are structured in such a way, that modification to an archived data object or its evidence record is always detectable, including modifications made by administrators of a TAS.

3.4 Archive policy

Archive policy determines the characteristics of the service implementation. Such policy contains several components including archive data maintenance

policy, authorization policy, service policy, assurance policy, etc. A maintenance policy defines rules such as preservation activity triggers, default archive period and default handling upon expiration of archive period. Maintenance policies should include mechanism-specific details describing the TAS operations.

An authorization policy defines the entities permitted to exercise services provided by the TAS service, including information on who is permitted to submit, retrieve or manage specific archived data objects. A service policy defines the types of services provided by a TAS service, including acceptable data types, description of requests that may be accepted and deletion procedures.

3.5 Data confidentiality

As a service that may be accessed through open infrastructure, it is expected that such a service provides means to ensure confidentiality of archived data objects, including confidentiality between the user and the TAS service. Hence, the TAS service incorporates means for accepting encrypted data such that future archive activities apply to the original, unencrypted data. Encryption or other methods of providing confidentiality must not pose a risk to the associated evidence record.

3.6 Transfer data and evidence from one service to another

TAS services are driven mainly by business factors and hence may cease to exist or may for example change archive policy. There are several reasons for user changing services and a TAS service provides means to transfer archived data together with the evidence records to another service regardless of its nature of operation. Usually such a service must enable acceptance of data together with previously generated evidence record. In general, the TAS services operate in a way that evidence records span over multiple providers in the course of time without losing the value of evidence.

3.7 Operations on groups of data objects

Data grouping occurs for logical (e.g. semantic), business process related or any other reason. A user of a TAS service has complete control over data grouping, i.e. comprise a group, while retrievers are able to retrieve one, some or all members of a group of data objects. In such circumstances a TAS service ensures that evidence is always present for a single instance of data or in other words, non-repudiation proof is available for each archived data object separately.

4 Designing the TAS Service

Architectural components of the TAS service are defined according to the service mechanisms for data object preservation on a long term basis including data retrieval, validation and evidence creation and management. The interaction with the service is performed by use of a Long-term Archive Protocol (LTAP). The protocol is characterized by the types and structures of messages exchanged between the user and the service. The LTAP messages carrying archive processing information (and payload) represent the formal interaction between the client and the server.

Data management is the next building block as required mechanisms for the archive object logic and handling. The management takes care about the physical data storage. Solution and mechanisms for this part of the service are already well known and documented and commonly implemented as Document or Data Management Systems (DMS).

Evidence record generation and management is the focal element of the TAS service providing attestations and demonstration of data authenticity and integrity It may combine certification services for evidence creation (e.g. time-stamping) and processing of objects that are associated with security attributes such as digital signatures (CRL, OCSP, etc.). Evidence record use PKI enabled techniques to protect archive data and deliver information on presence on the timeline.

All logical components of the TAS service may come in different arrangements and may be combined as a single entity. When designing the TAS service and the respective arrangement of the logical components, it is of utmost importance the logic of individual entity to be understood correctly. This is why some of the blocks are present as a complex entities consisting of several logic (sub-)components.

The Trusted Archive Service as presented extends the existing architectures (OAIS, MoRey, DOD 505, etc.) with specific mechanisms that meets integrity and non-repudiation proof criteria as required by most known specifications for document and records archiving. LTANS WG initiative of IETEF defines the missing technological elements that address the TAS requirements.

4.1 Archive Data Object

Archive Data Objects (ADO) are constructed upon request for data archiving. These objects are maintained through the complete archival lifecycle by the TAS service. They are presented, transferred, shredded but never modified in any way, except for supporting information such as managerial attributes, evidence records, etc. Logical object structure defines data to be preserved and data generated and maintained for the preservation process, the later are

simply known as data Conservation Attributes (CA). The data to be preserved must remain in the original form, while the role of CA is to provide trustworthy information on archive data existence, integrity, authenticity and validity (e.g. digital signatures).

Through the preservation process some additional information is usally collected or generated for the purpose of ADO object management. Figure 2 presents the logical structure of the ADO object, which may be physically distributed over different systems, e.g. the system for data or document management (in the role of document physical repository), the system for evidence management (in the role of evidence information generation and maintenance) and the system for data validation and certification (in the role of digital signatures verification), etc.

Some meta information may be associated with the archive data and may deliver descriptive information of data and associated attributes like digital signatures. Additional meta information is collected during the archival procedure by the TAS service. Such example is complementary information to digital signatures (digital certificates, certificate revocation lists, etc.) associated to archive data.

For particular document archival process, some specific information is required based on the legislation or simply requested by a user with specific needs. In cases when required information is not delivered as meta information, the TAS service collects archive meta data from a user or alternative resource.

Evidence information is generated by the TAS service or retrieved from supporting service (e.g. time-stamp) upon request by the TAS service. Once constructed, the archive object is a matter of continual maintenance by the TAS service for the long term integrity and time existence provision. Such maintenance is seen through provision of newly generated evidence records replacing the former proofs (due to e.g. cryptography weakness) for the complete duration of object archiving.

4.2 Long Term Archive Protocol

The Long-term Archive Protocol is a transport protocol carrying messages exchanged between the TAS service and the client. The messages are interpreted in a way that enable the logical data structure and all needed information (or references to the information) to be built as an archive object with the archive data itself (or reference to archive data).

The logical structure of the LTAP messages addresses the archive data, the archiving process information and references together with the request for information processing. By using LTAP protocol a user is able to deliver enough information to a TAS service for building data objects and for performing operations on the archive objects. Structure of the LTAP message is as follows:

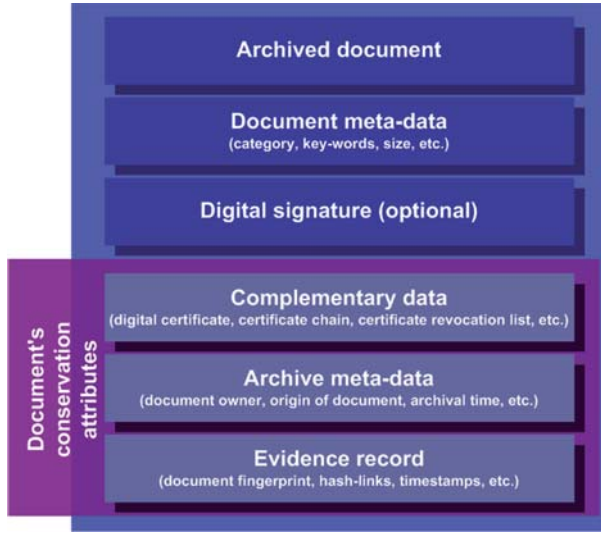


Figure 2 Structure of the Archive Data Object

- request information or request status information,
- raw data or references in case of status request,
- metadata providing additional information about the raw data (or any other data directly associated to an archive data object such as digital signatures),
- authorization and authentication information of the entities participating in the procedure e.g. the service server and the user,
- other information, required for supporting function like billing or charging.

The interaction with the TAS is performed by use of an interface that allows clients to define an archive service they are interested in (e.g. with grouping functionality) more particularly:

- submit data to the TAS (and request creation of evidence records for data) – submit
- check status of submitted data (that are being processed by the TAS) – status
- extract, transfer or simply retrieve data (archive data and evidence data) from the TAS – export

- delete data and/or evidence records from the TAS – delete
- verify the integrity and authenticity of data archived by the TAS – verify

The LTAP protocol is designed in order to provide the formalization of interaction between a user and the TAS service. Result of this interaction is the attestation of procedures performed by the TAS service (e.g. archive data). The protocol itself is asynchronous by nature. The LTAP exchange pattern is made of a request and two different types of responses. The initial response that appears after the issued request is a technical acknowledgement from the TAS service that the request has been received and accepted or rejected (for authorization reasons, for example). The second is a statement from the TAS service containing an indication of the outcome of the requested operation (e.g. request to archive a data object). This result (called an attestation) is in general considered as a document with long term validity as it allows the user to use it as a reference for the operation and as reference for the data that have been preserved by the TAS.

4.3 Evidence Record Syntax

An evidence record is a unit of data, which is used to prove the existence of an archive data object or object group at a certain time. Evidence Record Syntax (ERS) in the TAS context defines the data structure used to hold the information of data object time existence and integrity proof over long periods of time. Integrity protection is in general achieved using one-way hash algorithms and digital signature techniques. By integrating time component from a trusted time source a trustworthy proof is provided.

The hash and signature algorithms used in the TAS service may become weak and insecure during the preservation time. Generally recognized integrity and time existence mechanism are the time-stamps techniques as defined in Time-Stamp Protocol/Service. These time-stamps, issued by Time Stamping Authorities (TSA), are signed confirmations that data existed at a certain time. As time-stamps are actually digital signatures with a time component delivered by a trusted third party, like Trusted Time Authority (TTA), they suffer from the same time limit restrictions as digital signatures.

The ERS syntax broadens and generalizes time-stamping approach for data of any format. This implies as well handling of large amounts of data objects (grouping). The ERS syntax specifies data object which contains archive time-stamps and some additional management and grouping related data. Archive time-stamp is defined as a time-stamp and lists of hash values that allow verifying the existence of several data objects at a certain time. This ERS record is a logical part of an archive object and may be maintained separately form the preserved data.

The ERS syntax is based on evidence information delivered in a form of a time-stamp. For preservation purposes this time-stamp has to be extended into an archive time-stamp. Stamps can refer to a single object or to a group of objects (according to the grouping requirement). Grouping method is based on building hash trees. In such trees, leaves are the hash values of the objects and the root hash is produced by time-stamping procedure as shown in Figure 3. A hash tree is composed in a way that a deletion of a single or several objects does not influence other values.

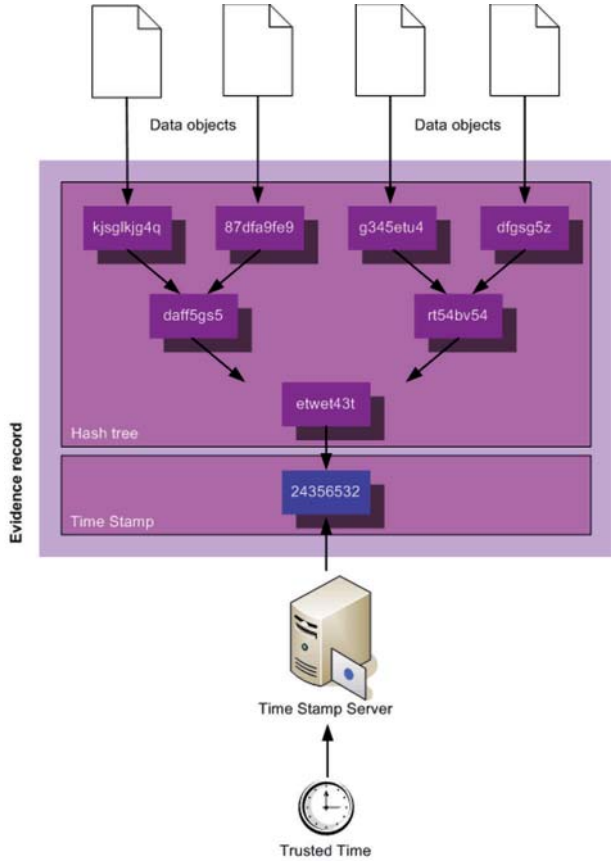


Figure 3 Evidence Record Syntax logical structure and hash grouping

The hash tree generated in this way together with the singular time-stamp provided over the root value has to be maintained over the complete archive period, implying that in case the cryptographic algorithms used to build the

tree and the archive time-stamp become weak they have to be replaced. To provide a continual demonstration of integrity in the process of the tree and the time-stamp renewal a complete data object or object group as well as preceding evidence record has to be processed by use of stronger algorithms. The ERS syntax recognizes two ways of the archive time-stamp renewal, the simple renewal and the complex renewal method.

In the case of simple renewal procedure, the archive time-stamp is hashed and time-stamped by a new archive time-stamp. It is not necessary to access the initially archived data objects itself. This simple form of renewal is sufficient, if only the hash algorithm or the public key algorithm of the archive time-stamp lose its security suitability or there is a risk that time-stamp certificates get invalid (due to time validity limitation).

The simplified procedure is not sufficient if the hash algorithm of the hash tree and of the archive time-stamp becomes insecure. In this case the object or object group, hash tree and archive time-stamp has to be hashed and time-stamped again by a new archive time-stamp. One option to avoid hash tree re-time-stamping is the use of stronger algorithms in the initial phase of archiving. This approach does not provide any guarantee as the weakness of the hash algorithm can always be found in its mathematical concept. If the limitations of the hash algorithm are only related to the length of computed hashes, then the longer hashes may be used to solve the problem for the longer periods.

5 Trusted Archive Service Implementation

The electronic data stability over long periods of time has become an important issue and challenge for researchers and standardization bodies, as electronic documents and electronic signatures are of increasing importance for e-business and e-government processes. Business and governments are introducing electronic forms and electronic methods for doing business in B2B, B2C, G2C, etc., scenarios. Preservation of electronic data is still multidimensional challenge as it concerns social and business life as well. One dimension of this challenge has been addressed recently by the IETF working group on long-term archive service that has defined the requirements for this service (LTANS).

The TAS services have already been pushed into the practice. An example of use of standardized techniques to provide long term stability and validity of electronic records is the preservation of electronic invoices for telecom and mobile operators and internet providers. Such records are usually supported with digital signatures to provide legal validity and hence must be preserved following technical and formal requirements. Electronic invoices also presents typical grouping requirements as such documents are generated in batches. Obtaining a single time stamp for e.g. 1 million records is a prerequisite for business process

optimization as time-stamping is usually offered as external commercial service. Furthermore, providing integrity and authenticity proof for e.g. 10 years is also a critical factor. As PKI-enabled infrastructures are already well integrated in daily business (e.g. digitally signed electronic invoices), the TAS service exploits such functionalities to meet legal and technical requirements on preserving electronic heritage of contemporary business models.

References

[1] *Assistant Secretary Of Defense For Command, Control, Communications And Intelligence, 2002: Design Criteria Standard for Electronic Records Management Software Applications – DoD-5015.2-STD.* Department Of Defense Office of DASD.

[2] Brandner, R., Pordesch, U., Gondrom, T. *Evidence Record Syntax (ERS).* Internet draft, draft-ietf-ltans-ers-12.txt. IETF, 2006.

[3] *Consultative Committee for Space Data Systems, 2002: Space data and information transfer systems – Reference Model for an Open Archival Information System (OAIS) – Blue Book, Issue 1.* CCSD,.

[4] *European Commission, IDA Programme, 2001: Model Requirements For The Management Of Electronic Records.* CECA-CEE-CEEA.

[5] *European Electronic Signature Standardization Initiative, 2001: Trusted Archival Services (Phase #3, Final Report).* EESSI.

[6] *International Standards Organization, 2003: Space data and information transfer systems – Open archival information system – Reference model – ISO 14721:2003.* ISO.

[7] Blazic, J., Sylvester, P., Wallace, C. *Long-term Archive Protocol (LTAP).* draft-ietf-ltans-ltap-04.txt. IETF, 2005.

[8] Wallace, Pordesch, U., Brandner, R. *Long-Term Archive Service Requirements.* Internet draft, draft-ietf-ltans-reqs-05.txt. IETF, 2005.

Long Term Archiving Implementation – Slovenian Experience with Long Term Archiving

Aleksej Jerman Blazic

E-mail: aljosa@setcce.si

Abstract

Preservation of electronic documents is a multi dimensional challenge. The first attempts to standardize processes related to electronic archiving date back in seventies. These attempts mainly addressed the conceptual and organizational problems of electronic data preservation, which are today well understood and implemented form standardization and technological point of view. However, legislative recognition stays behind organizational and technological attempts. In the recent years a significant progress has been made on national level to remove the final barrier of business process dematerialization by recognizing electronic form on long term basis and by enabling transformation from paper to electronic form.

1 Introduction

Business as well as governmental processes today heavily depends on information technology. Documents are created, processed and exchanged in electronic fashion regardless of their content and format. It is estimated that over 80 percent of organization information is contained in form of a document, be it contract, invoice or a simple power point presentation. For its undermined value over long terms of time the content is usually transformed form original electronic form to paper. Paper (or other solid material) is recognized to sustain environmental changes in efficient way and deliver integrity proofs in a simplified manner.

However information based on paper limits its use. Next to processing restrains, paper based preservation imposes significant reduction of business or governmental process optimization. Searching and retrieving paper based documentation may be counted in minutes, even hours or days. Space used for large volume paper archives demands significant investment in physical infrastructure, which is usually difficult to upgrade when requirements rise. The list of disadvantages of paper based preservation is endless, while a simple fact is unavoidable: most of the information is already generated in electronic form. The

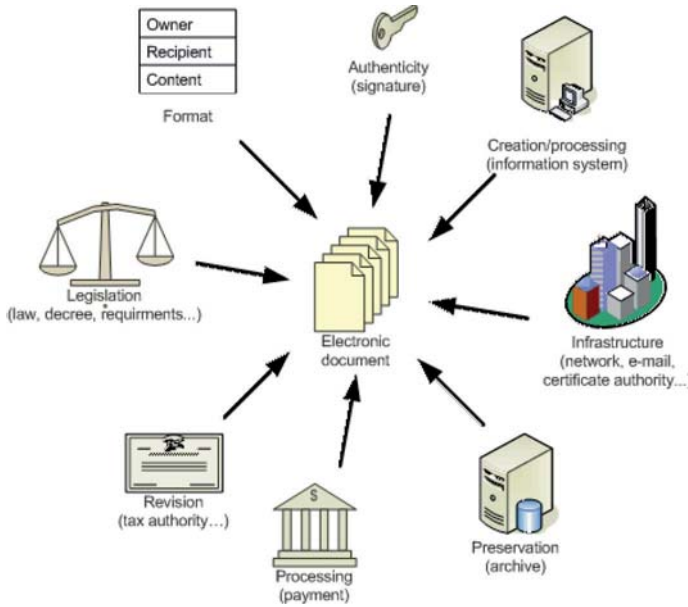


Figure 1 Multidimensional challenge of electronic format recognition and implementation

sole reason for using paper in today's advanced communication and information infrastructure is its indubitable recognition by legislation, even across borders.

Transition to electronic form hence requires the harmonization of technological concepts, organizational principles and legislation framework. Most of the EU based countries have already made an important step towards electronic business and electronic government recognition. However as transition to paperless environments can not happen in a single action, the existence of both forms, electronic and paper at the same time needs to be address for the purpose of incremental dematerialization.

2 Foundations

The initial attempt to use electronic form in business processes was strongly driven by industry for over 40 years. First electronic documents were used mainly by associations of industry partners such as transport or retail industry. Concepts of Electronic Data Interchange (EDI), Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT), etc., were defined and implemented since seventies. Being adopted by closed environments with

well known and authenticated communicating parties, such concepts presented a stable and secure environment for conducting electronic business and delivered several advantages over paper based business processes.

With the aim to enable a wider community to exploit advantages of electronic business, new and open technological concepts needed to be defined and legislation framework set up. Harmonizing common European market and boosting the use of electronic services for business and government process is one of the major focuses of European Commission. Directives 1999/93/EC on a community framework for electronic signature (Directive on electronic signature) and 2000/31/EC on certain legal aspects of information society services, in particular electronic commerce, in the internal market (Directive on electronic commerce) present the foundation for implementation of national regulation on electronic commerce in European countries.

Slovenian legislation followed Directive 1999/93/EC with the implementation of national Electronic Commerce and Electronic Signature law (ECES)[1]. By recognizing electronic form and electronic signature as legally accepted means, ECES presents the foundation for wider electronic business and government implementation. However, harmonization with the rest of legal framework is needed for overall recognition of electronic form, which was done with updates to related acts such as Official Procedure law (OP), Value Added Tax law (VAT) and Notary Law.

The general principle of ECES is nondiscrimination of electronic form against physical form. For this purpose ECES defines:

- electronic means,
- electronic messages,
- electronic systems,
- electronic communication and
- electronic signatures.

Following conditions defined by the law electronic form is recognized equally to physical form as well as handwritten signature is equal to a signature produced using electronic means.

While most of the legal implementations recognize the use of electronic form, they do not define coexistence and transformation between paper and electronic form. Occurrences of such happen mostly in document exchange processes and processes related to preservation of formal documents. Coexistence is addressed by Document and Archive Material Preservation and Archives law (DAMPA),

[1] ZEPEP was implemented in 2000 and had three annexes up to date.

defining requirements and rules form organizational and technological perspective on preservation of paper and electronic records. Furthermore it defines conditions and requirements for coexistence of both forms and transition form one form to another.

DAMPA was implemented in year 2006 to define archive and document material and the role and function of private and public archives. Upgrade based on initial archive law (Archive material preservation and archives law) broadens the impact of preservation with the recognition of digital form and transformation between the forms. In technology context DAMPA is supported by Common technology requirements defining long term specific digital format, storage media and integrity protection.

3 Electronic Preservation

Electronic preservation delivers several advantages over paper based archives. Using electronic means and infrastructures availability of information may be increased by several factors. Access to electronic records is simplified, optimized also in the context of remote access. Indexing and searching is increased in terms of volume and decreased in terms of time, while revision control is maintained in far more complex manner. From security point of view electronic archives delivers multitude of security layers form distributed locations, multiple copies to advanced access control and audit trails.

Electronic preservation is addressed by organizational, technical and legislation challenges. On organization and technical level electronic preservation must deliver answers such as:

- resistance of electronic media on long term basis,
- challenges of environment change due to technology progress,
- demonstration of authenticity and integrity.

Such technology topics have been intensively researched in the past, mostly independently. Today electronic preservation is achieved using combination of technologies that provide a stable environment for electronic documents on a long term basis. From legislation perspective, organization and technology concepts have been integrated in a transparent way – technology formats and standards are only referenced as guidance to end user solutions. Furthermore, governmental institutions and independent organizations provide a variety of services delivering certification and accreditation on solutions and services offered on the market, easing the selection and regulation compliancy testing for the end user.

Information technology for electronic preservation is available on the open market and has come to a stage being implemented as a supportive infrastructure, meaning that most of the legal information is kept in original paper based form. Legislation approaches do not affect the current state of technology progress. Rather than that they recognize well understood concepts and formats proven to work even in most demanding environments. Legislation provides a stable framework where organization and technological concepts are recognized and approved, therefore removing the final barrier of business and government process dematerialization.

4 Formal Recognition of Electronic Preservation

Legislative recognition of electronic preservation follows organizational principles:

- durability and reproduction,
- availability and accessibility,
- integrity and authenticity.

Several stages of implementation create an overall picture of electronic preservation. General requirements are summarized in law (DAMPA). Specific requirements are defined through decree, such as operation of electronic archives, demonstration of authenticity, organization of archives, etc. while references to standards and technical recommendations are implemented through Common Technical Requirements (CTR).

Long term preservation is the general issue from legal as well as technical point of view. Usually the original environment of electronic documents ceases to exist due to continual technology progress. Long term preservation is therefore defined to more than 5 years time. In this context transformation to stable format is required.

General condition for stable document format is usually wider community recognition or standardization on a global level. Open standards are less affected by technology progress as they are usually supported by larger community and publicly available (e.g. for revision).

4.1 Preservation infrastructure

Following general legislation requirements preservation infrastructure is composed of elaborated pieces of hardware and software. With the aim to harmo-

nize internal market, solutions and services are subjects to inspection. Such inspections occur in two scenarios only:

- as a registration or accreditation procedure of solutions and services provided on the market,
- as an audit procedure in case of disputes or other related events.

Preservation infrastructure is defined by the general service, i.e. storage and supporting services such as:

- transformation,
- integrity demonstration,
- encryption/decryption,
- organization,
- indexing,
- ...

Implementing trusted archive solution or services may not need to follow the complete technology concept. Rather than that, preservation needs to be tuned to business process' characteristics, like form of documents used (paper or electronic). The final technology composition may therefore vary form user to user, following only general guidelines. Technology set up is reflected by internal preservation procedures.

4.2 Procedures

Paper based as well as electronic preservation is strictly followed by internal procedures and rules, which define:

- internal organization and personnel,
- preservation physical infrastructure,
- documentation capture and transformation,
- short term preservation,
- selection, transformation and long term preservation,
- deleting and shredding,
- continuous performance,

- supervision and audit,
- implementation, transition and mass capture,
- internal rules update.

Rules and procedures are defined and implemented by individual organization:

- definition of internal procedures and rules or adopting external sample rules;
- supervision of preservation processes;
- modification and completion of internal rules (when required).

For the purpose of easing the implementation general and required elements of internal rules are defined by state institutions or shared between organizations. Procedures and rules implemented are expected to be reasonable, accurate and interpretable. They must be adapted and tuned to organization (needs) in a manageable fashion. Supervision of rules is performed by state institutions and occurs on periodic basis.

4.3 Implementation

Implementation of electronic archive is performed in a series of procedural and technology steps. The general implementation stages are:

- identification of documented material for preservation;
- identification of technology standards;
- definition of internal preservation procedures rules;
- implementation of technology solutions and/or services for document preservation.

Technology vise an electronic archive solution must provide an environment that follows trustworthy principles in all stages of the preservation process, starting with the capture. Rules characterizing the initial step of capturing are defined for both, paper based and electronic based documents. In both cases, integrity of captured information is to be preserved together with archived data.

Transformation of data to be archived may occur for at least two reasons:

- transformation from paper to electronic form,
- transformation to long term (digital) form.

Long term form is expected to successfully resist technology progress and is hence referenced by Common Technology Requirements. An example of such form are PDF/A or XML formats. Transformation to long term formats must occur on a periodic basis – new form replacing outdated and unstable format. Such transformation needs to be performed in controlled environment eliminating potential errors and manipulations in the process.

In the preservation process, additional sets of data (meta data) are usually generated for organizational and managerial purposes. After (successful) document transformation and meta data collection, evidence records (e.g. time stamping) are applied to archive data and archive meta data. For digitally signed electronic documents additional measures are usually taken to prolong the validity of digital signatures. All procedures in relation to integrity, authenticity and validity must be documented and present the integral part of archiving solutions.

In the final stage of the preservation process data deleting occurs. In many occasions, digital data shredding can not imitate paper form. Several measures need to be taken into account due to the facts such as data replication, meta information, etc. When data shredding occurs, all archived data, including meta data, needs to be deleted. Shredding procedure is also a part of internal rules.

5 Summary

Transition form paper to electronic form needs to take into account all aspects including electronic preservation. Technology approaches are supplemented by legislation frameworks. In general electronic archiving must take into account all aspects including technology definition, transformation process, operation requirements and risk assessments. To sustain on the long term basis, preservation procedures must be well documented.

Slovenian experience demonstrates an effective approach on coexistence of both, paper and electronic form as an important step in the process of business and government process dematerialization. In the evolution of law several important stages were envisaged and steps taken. With the year 2000 the equalization of physical and paper form was defined. Five years later electronic form prevalence occurred with a simple decision on predominance of Journal of Republic of Slovenia in electronic form. As of year 2006 preservation in electronic form is generally recognized also for data originating in paper form. The final step towards process dematerialization is performed through certification and accreditation procedures establishing a common understanding of electronic preservation of archive and documented material and creating an open market for suppliers and service providers.

References

[1] *Directive 1999/93/EC on a community framework for electronic signature (Directive on electronic signature),* 1999, European Commission.

[2] *Directive 2000/31/EC on certain legal aspects of information society services, in particular electronic commerce, in the internal market (Directive on electronic commerce),* 2000, European Commission.

[3] *Document and Archive Material Preservation and Archives law (DAMPA),* 2006, Government of Republic of Slovenia.

[4] *Electronic Commerce and Electronic Signature law (ECES),* 2000, Government of Republic of Slovenia

[5] *Notary Law,* 2004, Government of Republic of Slovenia.

[6] *Official Procedure law (OP),* 2004, Government of Republic of Slovenia.

[7] *Value Added Tax law (VAT),* 2003, Government of Republic of Slovenia.

The French Amdinistration's profile for XAdES

Peter Sylvester

E-mail: peter.sylvester@edelweb.fr

Since 1999 we have the European Directive for electronic Signatures. In order to specify details of formats and technology, European standardisation bodies worked in the context of the European Electronic Signature Standardisation Initiative. The ETSI has created two ranges of specifications based on the two major signature technologies, namely Cryptographic Message Syntax (CMS) which is standardized in the SMIME group of the IETF and XML-DSIG of the W3C. The document that build on XML-DSIG is called XAdES. It is an equivalent of a document CAdES for CMS, and also enhances XML-DSIG with some of the basic feature of CMS, e.g.e. attributes or structures for counter signatures.

In the context of the French administration's activity to define a general reference for interoperability and security, there are rules that require the support of signatures of XML documents which are supposed to conform with the EU directive. In 2006, EdelWeb was contracted by the Direction Générale pour la Modernisation de lEtat (DGME) to develop a specification.

Starting from a preliminarily version of a text which was produced as an extract of a product documentation of some French vendor of XML signature solutions, public comments were solicited. A small group of solution developers, important users and participants from the administration was formed. As a result of the initial comments, a complete new document was drafted which has no resemblance with the initial ons. The document was discussed within the group. Comments, errors, suggestion were treated. The document is in its final state and awaiting publication by the administration (DGME).

An addition document was produced by the Direction Centrale de la Securité de Systèmes d'Information (DCSSI) that defines which cryptographic algorithms are appropriate to use in this particular context.

One of the first crucial tasks was to determine the approach and the scope of the document. It was necessary to make important decisions in controversial discussions.

XAdES goes far beyond the elementary need to fulfil the requirements of the EU directive. It defines different formats in a hierarchy. Although the practical

implementation of all the formats is not a big technical problem, it is by no means clear to what business or legal requirement they correspond.

One of the first decisions was to determine which features of XAdES should be used and for what purpose. Obviously, the purposes of creation and verification of a signature are required. It was proposed to use also the XAdES features supposed to permit re-verification at a later date. This feature was rejected by consensus since this relates to long-term work flows and document procedures, and this was clearly out of scope of the work.

As a consequence only the formats BES and EPES are relevant. Since nothing is said about the other formats, the document is not a complete profile for XAdES but contains a selection of XAdES features.

Another important decision was to exclude anything about the usage of cryptographic algorithms or details of certificates. As already mentioned, the DCSSI produced in parallel a document concerning algorithms. This document may evolve following the evolution of cryptographic algorithms, contrary to our technical specification which is supposed to remain rather stable. Concerning certificates, the DGME also has a set of documents describing rules for certification authorities, policies and practices, these documents, also known as PRIS, are also only referenced.

Besides the treatment of XAdES, there are two other smaller chapter concerning features of XML and XML-DSIG. They seemed necessary because of some potential ambiguities in the general rules defined by the RGI document (référence générale d'interopérabilité) of the DGME. The confusion comes from the usage of the IETF terminology of 'MUST, MAY, SHOULD', etc. and usage of phrases like 'MUST use' instead of 'MUST support'. The concrete example: "One MUST use the XML version 1.1" which is strictly incompatible with the XML specification itself. Thus, awaiting a correction or clarification of the RGI document, it was preferred to add clarifying text in our document.

A particular feature of the RGI and our document is the terms "optional" or "may" are never used.

For XML-DSIG, there are a few remarks concerning the canonisation algorithms and the different formats of XML-DSIG. We do not actually restrict any of the possibilities of XML-DSIG concerning multiple signatures or different formats. The reason is that this part is related to document formats and not to signature formats. One of the consequences is that our document is not sufficient to guarantee interoperability of document creating and verification tools, application specific document formats and additional rules about multiple signature need to be defined. In other words, the scope limit requires that a verifying application is capable to (with the help of a user or automatically) to determine which signature is to be verified and to provide the input data to a verification tool. Similar, after creation of a signature, it is the responsibility of the applica-

tion to join the signature to the XML document. (or not, in case of a detached signature).

Concerning the usage of signature keys, only keys certified by X.509 are required. This is a consequence of XAdES but also a direct requirement for the administration. X.509 certificates must conform to the rules specified in the PRIS.

XAdES extends the signature verification to address a requirement of the EU directive. It is required that the signature identifies the signer. A common interpretation is that a signature verifying certificate is not sufficient because one can imagine a substitution by a rogue CA that bind the known public key to another entity. Whether this theoretical problem can exist, i.e., whether the is a scenario where an attacked can have any benefit, is totally unknown. The proposed solution is therefore to include the verifying certificate in the signed portion of the document.

XAdES permit two ways to do that. The usage of a SigningCertficate attribute was selected and the inclusion of KeyInfo discouraged. The reason is essentially that KeyInfo may change, i.e., from a reference to a certificate to a certificate. We also require that a non XML-DSIG verification must be possible. The semantics of the SigningCertficate attribute is unknown to a normal verifier, therefore the KeyInfo must provide sufficient information permitting the verification.

A few additional elements of XAdES are recommended. They concern the date of signing, the location, roles, and engagements. No semantics is required by any signature creation or verification tool, the values of these fields are uninterpreted texts.

It is recommended to reference signed document policies leaving out of scope of our specification any description of what that can be and how it can be treated.

Electronic notary services

Peter Sylvester

E-mail: peter.sylvester@edelweb.fr

Since many years the term notarisation or notary service is used more and more in the security area of information systems in particular in the context of what is called dematerialisation or paperless document work flows. The terms notarisation and de materialisation are in fact French and the inituitive semantics may be misleading.

Furthermore, the definition of what these terms denote is vague, and there are confusions. For notarisation, the term does not designate the activities of a notary or public notary which are more and more supported by IT technology, although this profession may operate such services.

During the definition of the IETF working group LTANS (long term archive and notarisation services) it was even claimed that notary or notary services only designate the activities of that profession, and cannot be used in other context. By choosing an English pronunciation of a French word, the request was simple rejected.

For the context in which we are interested, the term notary service exist since at least 10 years to designate particular IT services performed by a (technically) third party. The first draft of RFC 3029, initiated by Entrust was actually 'draft-adams-notary' before becoming dcs and later DVCS. This technical specification is an example for services that we are trying to generalize.

Another example are the OASIS SAML specifications which also have a similar basic model. What can we say about that model? Briefly, a service performs some action related, requested by or concerning a client, and produces a document called attestation or assertion. In general, this document is intended to be presented to other participants in a particular work flow, at least potentially.

In the sequel, we will outline an approach for a general framework, motivated by the transformation of the paper world to an electronic support of documents and present some details in an abstract way and hope to stimulate some discussion. We mention several some examples of use cases that have been experimented by the author and other people during the last decade.

A paper world example

In order to explain the context, we start with an example from the paper world that involves authentic document. This example is a document created in a professional company context. The illustration shows a layout of a typical one page document. The content could be an invoice or an engagement of the company. What we are interested in is to explain the other parts on the paper, let us call them decorations and what are they good for.

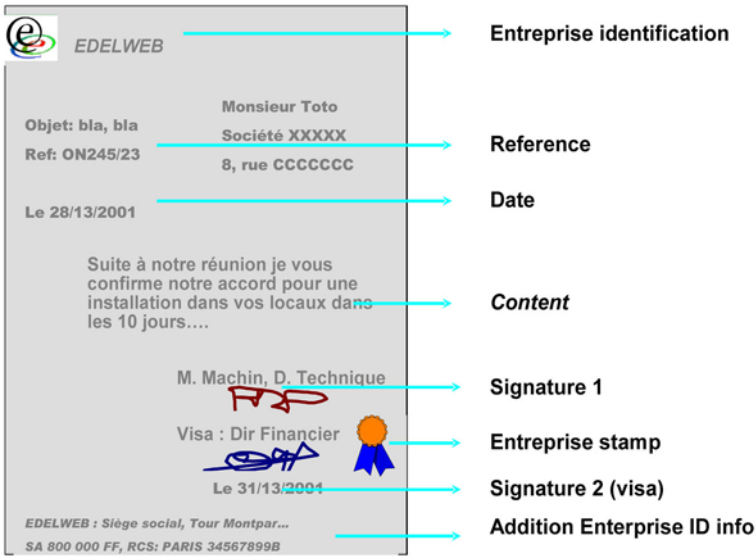


Figure 1 Formal enterprise document

The most important point for such a document is allowing any receiver or the pretended author to get a good feeling concerning the authenticity of the document, an understanding about the engagements and concerned or responsible persons. We like to avoid the term proof, guarantee, proof element etc. since they imply some potentially undesirable meaning or false analogies.

We find in this document various identification, a company logo, a reference number, signature dates, stamps and legal information. All of these information contribute to the verifiable authenticity of the document. Even the quality of the paper plays an role. These decorations are in other words security measures. During the last few thousands of years, they have been developed slowly, addressing new threats.

A paper that contains the company logo and the legal information of the company is normally only accessible to few persons. The reference number and

a stamp indicate that a copy of the document has been made and stored correctly. Signatures and counter signatures show who has seen the document and for what purpose. When such a document is finished, something else has happened in the environment, in particular, one or more copies have been made, the document has been registered etc. This contributes to non-repudiation.

Requirements for the paperless support

How can we translate this into the paper less world? One of the difficulties that we are faced with is that the terms used in both world don't correspond, a well known example being the terms signature or document.

Electronic signatures contribute to the integrity of the document, this is not the same situation case for signatures on paper. In the paper the document (i.e., the information) is bound once and in a static way to its representation. It may happen that one has to re-edit certain long term documents. In the electronic form the binding to a concrete support and representation are more dynamic. Since we no longer have paper, we also don't have a paper with a logo any more.

It should be obvious that an electronic signature alone can probably not be sufficient in this case; we are not in a situation where a private person just writes a simple letter and signs it.

What we have to do is to define procedures or work flows for the creation of the document where the data are presented to several services, some action is performed, and the outcome is asserted and included in the document. This is not at all different from the paper world.

On the document level, we need to have a structure that carries assertions on the document. To a certain degree, this is very close to CMS data structures with counter signatures, or the higher level features of CadES or XadES but they are specific and lack a common philosophy. Another example is the approach of SAML assertions that can be embedded within each other but a safer linkage to documents (e.g. via hashes) seems missing.

If we have structured document, we can try to add a semantic layer, we leave this outside of the scope here.

For the individual attestation level, it seems useful to have a common framework for a layout of an attestations and for a protocol to request and obtain them. Again, we already have several approaches developed by different standardisation bodies. The SAML approach or interaction in ebXML are encouraging, the IETF security area has a few important base formats to be taken care of. The attestation framework and protocol framework should be done in a abstract way allowing concrete binding to different lower level technologies, e.g; SOAP, SMIME, XML, CMS, HTTPS. Here we are faced which the problem that these

bindings have a great influence to the document formats and create difficulties such as the need of encapsulation of certificates or time stamps in XML-DSIG or XadES.

We are now ready to define actual services, and the specific protocol payloads for services like time stamping, certificate and signature validation, archiving, document conformance, company seals, Since at least some of the services exist already, we are faced with a compatibility or migration problem, and, most likely, with reluctance and resistance of producers of existing technology, and with the encapsulation problem mentioned in the last paragraph.

These technical prerequisites should be defined in such a way that a large number of organisational scenarios can be covered, and, furthermore, it should be easy to adapt the frameworks to new activities. We do not fully believe in the Internet doctrine that everything above TCP belongs immediately to the application layer, since this has created in the past very monolithic applications and few common protocol layers or many different presentation formats. After several decades one can fold out mid-level layers or define new ones, e.g. HTTP or SOAP.

We can here see three levels requirements for interoperability, we cite a definition the European Interoperability framework:

Organisational Interoperability is about streamlining administrative processes and information architecture to the institutional goals we want to achieve – and to facilitate the interplay of technical and organizational concerns. It requires the identification of 'business interfaces', and coordination throughout Member States and the European Union.

Technical Interoperability is about knitting together IT-systems and software, defining and using open interfaces, standards and protocols. It relies on cooperation as well as on technical infrastructures.

Semantic Interoperability is about ensuring that the meaning of the information we exchange is contained and understood by the involved people, applications, and institutions. It needs the know how of sector institutions and publication of its specifications.

Abstract protocol outline

Since we are technician, we describe here a basic abstract protocol between a client and a server. The protocol is used in our IETF LTAP specification protocol for a trusted archive service and motivated by analysing other proposals like DVCS, another initial LTAP protocol, and more generally EDI, X.400 etc.

The services that we want to access may be operated in different modes, e.g., an online request with an immediate response or requests for which a response requires an important amount of time. An example for the latter is an

archive service. It is of great advantage to assume that the protocol is essentially asynchronous.

We think that an approach which is borrowed from EDI, X.400 or ebXMl that consist of sending a request and permitting two types of answers, is appropriate. A client send as request and receives first a technical acknowledge and then a acceptance of the request and the requested attestation (or a rejection). We must provide rules of how to repeat request in the case of a possible loss, and as a consequence may need to ensure idempotent operations. There is nothing new here, this is well known in the network level for TCP over IP but easily overlooked.

We can thus observe the following states in a client and server:

- The client and server are in the idle state.
- The client has initiated an operation. The server may have received the received or not. Since the client cannot assume that the has received the request, it client retry the operation after a time-out.
- The server has received the request and has send the technical acknowledge, and starts performing some associated operation. The server ensures idempotence if necessary, on multiple occurrences of an identical request, it sends the same response. The server at that point may still loose all knowledge of the operation, e.g. in case of a power failure.
- The client has received the technical acknowledge. Depending on a lower layer binding the client either waits for the final outcome of switches to a polling mode to obtain the final answer using some identifier provide by the server. It may be necessary to repeat the initial request in case the server has not all knowledge of the transaction and thus, responds with an error.
- The server has finished its work, and has send a definitive answer for an operation. The server assumes the responsibility of the operation. This means in particular, that the server has secured in an appropriate way the knowledge of the operation or associated data.
- The client has received a definitive answer and goes to idle state.
- A server may delete knowledge of a transaction after some time, in particular for negative responses. For positive answers, this also depends on the nature of the operation.

In case that the service can be provided immediately, i.e. When the server responds immediately with a definitive answer, it is obvious how the state changes

combine. We may want to define an operation to abandon a transaction; it is not clear to us whether this should be a generic feature.

This transaction approach permits us to implement cost effective solutions by allowing operations to fail and loss of data which can be recoverable in a higher organisational or semantic context.

Appropriate security measures need to used depending on the nature of the transaction and the participants. There is nothing very special here, except maybe one thing which is also motivated by a requirements of the EU directive for Electronic Signature. When an authentication method is used to protect the request or response, we do not want the method provide the identification of the participants. In other words, we want that the requests and responses explicitly contain identifiers of participating entities, and the authentication methods may be used to verify them, and of course, the identities are protected by the authentication method. This is for example not the case in basic CMS SignedData or XML-DSIG. We do not have particular requirements for the lower layer, allowing the usage of HTTP(S), electronic mail, or web services, etc.

We can now define the protocol data structures. The requests must contain two parts:

- an information identifying the request, its nature, and the participants, and, depending on the nature of the request.
- Associated data which we think that we can structure them in a global way. There may be some raw data or a reference to them. In addition, there may be meta data associated.

The response may be of two kinds:

- a structured error information which should allow in a global way to detect classes of errors similar to the first digit in a SMTP reply. Thus, clients can react appropriately without even knowing all details of an error.
- A response related to a request which contains the identification part of the request and, depending on the operation, additional data provided by the server, in particular, to link it to data.

Since we want to build structure with assertions, the requests and responses must contain elements for linking, i.e. identifier and cryptographic hashes in order to permit global authenticity checking. We do noy require that cryptographic links need to be safe for a very long time.

We have to bind these abstract information to concrete data structures, and here we are faced with two competing religions, i.e. ASN.1 and XML. Fortunately, there exist now specifications of correspondence, i.e., it is possible to encode ASN.1 data structure using XML Encoding rules, which results in the

possibility to define a corresponding XSD schema, and in the other sense, we have the possibility to derive an ASN.1 syntax from an XSD schema. What we believe is important , is to provide presentations in both styles and to attempt to have them compatible or easily convertible. Of course, conversions may create problems in the authentication layers.

Data structures and work flows

We have to combine attestations and some data. Depending on the nature of the data, we have a broad range of possibilities. They all have advantages and disadvantages. We list some examples.

- Encapsulation into a mime-multipart document. This may be useful as a transport feature but not necessarily as a storage format.
- Encapsulation of the elements as separate files in a zip-format similar as with the odt format of OpenOffice.
- Stacks of attributes in CMS SignedData assuming that assertions are represented as attributes.
- In XML, techniques like in XadES.
- Separate files linked together by some other meta information.

The goal of such formats is to have a common structure that can be eventually processed automatically by a work flow engine based on some document policy.

In this way we can be able to define rules for work flow engine which ensure that the necessary services are obtained. Since some of services are external to the main work flow, we actually have to define rules that allow secure interoperability between work flows. One first feature can be immediately derived from the protocol transaction model. After sending a request, the work flow may split into two flows, where one waits for the outcome, and the other may already prepare subsequent operations with the hope of a successful outcome.

Use case examples

We start with an electronic form of a formal company document. The creator's work flow systems ensures the conformance with defined document rules. We see here that there is an interest to involve a notary service at the final step, i.e., before the document released to its destination. This notary service includes in particular a feature to validate signatures, at least, it may call such a service.

The final attestation resembles a seal from the enterprise which confirms the proper execution of the work flow.

When such a document is received in a partners work flow, again, a similar service is invoked, of course, based of an different trust description. In particular, this service does not necessarily need to verify all the various signatures of the document, it may be sufficient to verify the other company's seal, and some attestations from third parties like time stamps authorities and archive services.

This approach permits to solve the problem of algorithm agility and availability of cryptographic algorithms. If the destination is not able to verify directly the signatures because an implementation of the required algorithms is not available, a notary service can be involved which in turn delivers a verifiable attestation. We know at least one concrete scenario and products which handles this for electronic documents exchanged between Russia and Poland.

The second example describes a concrete architecture defined in 1998 by the author and developed in a prototype which concerns electronic signature verification and a method to ensure confidentiality of the data with the possibility of datarecovery by the involved organisations. In 1998, in France it was required to obtain an authorization for products that involve high level cryptographic protection which we were able to obtain.

The solution involves a server based on RFC 3029 (DVCS) which, besides of signature verification, implements public key certificate validation of key encryption keys. As a result of the verification, the notary server not only has verified that is is still allowed to encrypt data for the recipient, but it also provides a set of additional public key encryption certificates which the client must also use to encrypt the data (i.e., the random key for the symmetric algorithm). This allows for example to implement checking of data in a outgoing enterprise mail gateway. A server typically has to add at least two certificates, i.e., one for the sender's organisation, one for the recipient's organisation, Additionally, there can be one for a government authority.

When receiving a document using the appropriate client software, a similar operation is performed, i.e. The decrypting program asks its notary service to verify the public key certificate of the decrypting user. The response of the server is then compared with the actual document structure, i.e., whether it contains the required number of recipients and whether the symmetric encryption key was correctly encrypted with for the required recipients. In case of mismatch, the decryption is either refused, or the document is enhanced with the required recipients.

Another use case also involves DVCS (we have not many other notary services so far): This was experimented in the OpenEvidence project and involves an equivalence of certified or recommended mail. The approach consists of enhancing the signed notification features of SMIME. The protocol consists of

three events: creation of a signed document, reception and sending a receipt, consumption of the receipt.

At each step, a DVCS server is called to obtain a time stamp, or, in case of the initial step, and archive attestation. Since DVCS time stamp are qualified in the sense that that can contain additional information, we can have the intended recipients of the data, i.e. their email addresses. The DVCS attestation are included as signed attributes with the data and the receipt. When the final receipt is received, the client thus informs its local server about the final outcome of the transaction. The server infrastructure has this a complete knowledge about the transactions. The logs of the servers can be analysed, lost transactions can be detected. etc.

Final remarks

We invite the readers to participate in the work of the IETF working group LTANS. This article is supposed to initiate the definition of concrete specifications.

We also hope to raise the need of such services and better treatment in standardisation bodies in order enhance interoperability and long term stability.

Jak je to se sílou algoritmů pro výpočet hash

Michal Hojsík

E-mail: Michal.Hojsik@siemens.com

Abstrakt

Od augusta roku 2004, keď tím čínskych vedcov objavil kolízie pre série hašovacích funkcií MD4, MD5, HAVAL-128 a RIPEMD, získala táto oblasť veľkú pozornosť. Boli objavené nové algoritmy na generovanie kolízií, popísané kolízie pre certifikáty X509, navrhnuté a následne zlomené veľké množstvá vylepšení týchto funkcií. Pozornosti vedcov ale neušla ani dnes najrozšírenejšia funkcia SHA1. Veľké úsilie sa taktiež venuje vývoju nových hašovacích funkcií. Príspevok obsahuje popis súčastnej situácie v oblasti hašovacích funkcií a detailnejšie popisuje praktické následky známych objavov na certifikátech X509. V skratke sa taktiež venuje budúcnosti hašovacích funkcií.

Úvod

V auguste roku 2004 sa počas rump session konferencie Crypto 2004 odohrala významná udalosť. Profesorka Wangová popísala 2 správy, ktoré majú rovnakú MD5 hash [1]. Pred oznámením tohto prevratného výsledku sa na poli hešovacích funkcií dlhší čas nič nedialo. Po ňom začalo z pohľadu hešovacích funkcií búrlivé obdobie. Krátko po zverejnení Wangovej kolidujúcich správ boli popísané ich prvé praktické využitia. Časom sa objavili nové a rýchlejšie techniky hľadania kolízií MD5 a zároveň aj nové praktické využitia. Ale nezostalo iba pri MD5. Pozornosť kryptológov sa zamerala taktiež na jedinú prakticky používanú náhradu za MD5, a to funkciu SHA-1.

V tomto príspevku sa pozrieme na niektoré výsledky z oblasti hešovacích funkcií a ich praktické využitia pri tvorbe kolidujúcich certifikátov.

Stručný úvod do hešovacích funkcií

V kryptológii pojem hešovacej funkcie označuje funkciu h, ktorá ľubovoľne dlhému vstupu (slovo ľubovoľne znamená napríklad vstupu kratšiemu ako 2^{64} bitov) priradí výstup konštantnej dĺžky, a ktorá splňuje nasledujúce vlastnosti:

1. jednocestnosť
2. bezkolíznosť

Pojmom jednocestnosť označujeme vlastnosť, že pre ľubovoľnú správu M je jednoduché vypočítať funkčnú hodnotu $h(M)$, ale pre zadanú hodnotu x je *výpočtovo nezvládnuteľné* nájsť takú správu M, pre ktorú by platilo $h(M) = x$. Pretože potenciálnych vstupných správ je veľké množstvo a hodnôt iba málo, všetci tušíme, že také M určite existuje. Jednocestnosť požaduje, aby neexistoval žiaden efektívny spôsob, ako také M nájsť.

Bezkolíznosť sa všeobecne delí na bezkolíznosť prvého a druhého rádu.

Bezkolíznosť prvého rádu znamená, že pre danú funkciu h nie sme schopní efektívne nájsť ľubovoľné dve správy M a M' také, že $h(M) = h(M')$. Bezkolíznosť druhého rádu znamená neschopnosť pre danú správu M efektívne nájsť inú správu M', pre ktorú platí $h(M) = h(M')$. Táto vlastnosť sa označuje taktiež ako neschopnosť nájdenia druhého vzoru.

Predstavme si, že naša hešovacia funkcia spracováva správy do dĺžky 2^{64} bitov a jej výstup má 512 bitov (prípad SHA-512). Máme teda približne $2^{(2^{64})}$ možných vstupov ale iba 2^{512} výstupov. Pre každú hodnotu x teda existuje obrovské množstvo správ M, pre ktoré platí $h(M) = x$. Takisto pre každú správu M existuje veľké množstvo správ M' s rovnakou hodnotou hash funkcie. Uvedené vlastnosti jednocestnosti a bezkolíznosti nám určujú požiadavky na výpočtovú zložitosť hľadania týchto správ.

Je dôležité uvedomiť si zásadný rozdiel medzi schopnosťou nájdenia dvoch správ s rovnakou hodnotou hash a schopnosťou *k danej správe* M nájsť inú správu s rovnakou hodnotou hash funkcie, teda rozdiel medzi kolíziou prvého a kolíziou druhého rádu. Požiadavka bezkolíznosti prvého rádu je pritom očividne silnejšia ako požiadavka na bezkolíznosť druhého rádu.

V prípade hľadania kolízie prvého rádu môžeme využiť známy narodeninový paradox, ktorý si teraz v skratke pripomenieme. Predstavme si, že máme skupinu 24 ľudí. Narodeninový paradox hovorí, že pravdepodobnosť, že sa v tejto skupine nachádzajú 2 osoby majúce narodeniny v ten istý deň je viac ako 50 %. Pretože táto pravdepodobnosť je pre tak malú skupinu väčšia, než by väčšina z nás očakávala, označujeme to za paradox. Aj keď rok má iba 365 dní, nám stačí skupina 24 osôb. Je to dané tým, že my nehľadáme niekoho, kto má narodeniny v ten istý deň ako konkrétna osoba (tu by sme potrebovali skupinu aspoň 183 osôb),

ale hľadáme ľubovoľnú dvojicu s touto vlastnosťou. No a možných dvojíc je už aj v tak malej skupine dostatok.

Trochu obecnejšie: pokiaľ náhodne vyberáme z N prvkov s opakovaním, je počet ťahov pred prvým opakovaním už vytiahnutého prvku približne rovný hodnote 1,26 krát odmocnina z N.

Aplikujme tento výsledok na teóriu hešovacích funkcií. Predpokladajme, že hľadáme kolíziu pre hešovaciu funkciu s dĺžkou výstupu 160 bitov, teda napríklad pre funkciu SHA-1. Narodeninový paradox nám udáva, že pokiaľ vypočítame hodnotu funkcie pre 2^{80} rôznych správ, s nadpolovičnou pravdepodobnosťou medzi nimi budú 2 správy s rovnakou hodnotou hash. Pokiaľ teda požadujeme hešovaciu funkciu so zložitosťou nájdenia kolízie prvého rádu aspoň 2^{80} (bezpečnosť ekvivalentná používaniu symetrickej šifry s 80 bitovým kľúčom), potrebujeme aby naša funkcia mala dĺžku výstupu aspoň 160 bitov (MD5 má dĺžku výstupu 128 bitov).

Pozrime sa teraz na aplikácie požadovaných vlastností jednocestnosti a bezkolíznosti.

Jednocestnosť hešovacích funkcií sa v kryptológii využíva vo veľkom množstve kryptografických protokolov. V počítačovom svete je asi najznámejšie využitie jednocestnosti pri overovaní užívateľského hesla. Pri tom najjednoduchšom protokole je v systéme uložený hash užívateľského hesla a pri prihlásení sa porovnáva hash práve zadaného hesla s hodnotou uloženou na disku. Jednocestnosť v tomto prípade chráni heslá pred zvedavým administrátorom, ktorý môže získať zoznam hešov užívateľských hesiel. Vďaka jednocestnosti ale zo znamu nezíska samotné heslá (takto jednoduchý protokol nás ale samozrejme neochráni proti slovníkovému útoku).

Najrozšírenejším využitím bezkolíznosti (prvého aj druhého rádu) je elektronický podpis. Pri jeho použití nikdy nepodpisujeme samotnú správu M, ale iba jej hash $h(M)$. Má to niekoľko dôvodov. Za prvé správa M môže byť priveľmi dlhá a podpis by teda zabral priveľa procesorového času (dnešné asymetrické algoritmy používané v algoritmoch pre elektronický podpis prevádzajú veľké množstvo modulárnych operácii, predovšetkým časovo náročné umocňovanie) a za druhé podpis správy by bol rovnako dlhý ako správa samotná. Pokiaľ ale podpisujeme iba hodnotu $h(M)$, máme pre ľubovoľne dlhú správu konštantne dlhý (lepšie povedané konštantne krátky) podpis. Hash správy sa v tomto prípade používa ako jednoznačný identifikátor podpisovanej správy a požiadavka na bezkolíznosť je zahrnutá práve v slove jednoznačný.

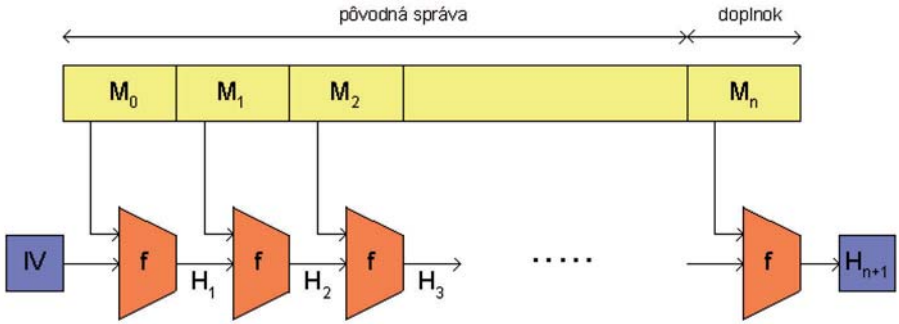
Pri podpise správy teda v skutočnosti podpisujeme iba jej hash hodnotu. Z toho vyplýva, že ak by niekto našiel inú správu s rovnakou hash hodnotou, bude náš podpis platný aj pre túto „falošnú“ správu. Preto od hešovacích funkcií používaných pre elektronický podpis vyžadujeme bezkolíznosť.

Na záver úvodu si ešte priblížime základný koncept dnešných hešovacích funkcií.

Konštrukcia hešovacích funkcií

Všetky dnes hromadne používané hešovacie funkcie sú založené na takzvanej iteratívnej konštrukcii. To znamená, že tieto hešovacie funkcie spracovávajú správu postupne – po blokoch. Základným kameňom takýchto funkcií je takzvaná kompresná funkcia. Pomocou nej sa postupne spracováva hešovaná správa a vytvára sa takzvaný kontext.

Pozrime sa na tento proces detailne. Pred samotným hešovaním je správa doplnená tak, aby jej celková dĺžka bola násobkom čísla daného popisom hešovacej funkcie. Predpokladajme, že požadujeme bitovú dĺžku deliteľnú číslom 512 ako je tomu v prípade funkcií MD4, MD5, SHA-0, SHA-1 a SHA-256 (doplnok je tvorený tak, že obsahuje informáciu o pôvodnej dĺžke danej správy). Následne je správa rozdelená na bloky, v našom prípade dlhé 512 bitov. Označme ich M_i, $i = 0, \ldots, n$ a priebežný kontext hešovacej funkcie označme H_i. Proces hešovania si popíšme pomocou obrázku 1.



Obr. 1 Iteratívna hešovacia funkcia

Na začiatku za kontext H_0 označíme takzvaný inicializačný vektor, ktorý je daný popisom hešovacej funkcie. Následne v i-tom kroku vytvoríme nový kontext H_{i+1} tak, že použijeme kompresnú funkciu na predošlý kontext H_i spolu s časťou hešovanej správy M_i,

$$H_{i+1} = f(H_i, M_i).$$

Po prevedení kompresnej funkcie na posledný blok správy M_n dostávame posledný kontext H_{n+1} z ktorého je následne predpísaným spôsobom odvodený výsledný hash správy M.

Všimnime si, že vďaka iteratívnosti tejto konštrukcie je výsledný hash správy tvorenej blokmi $M_2, \ldots, M_n$ (vynechali sme prvý blok) s inicializačným vektorom H_1 rovnaký ako hash celej správy s inicializačným vektorom $H_0 = IV$.

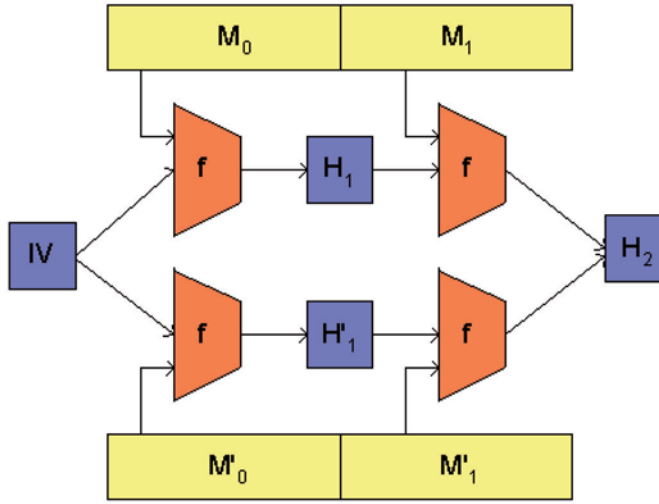
Kolízia hešovacej funkcie prvého rádu znamená nájdenie dvoch správ M a M' pre ktoré platí $h(M) = h(M')$.

Prípad MD5

Ako sme už spomenuli v úvode, v auguste roku 2004 oznámila profesorka Wangová schopnosť nachádzať kolízie prvého rádu vo funkcii MD5. Na konferencii ale zverejnila iba 2 páry kolidujúcich správ. Aj to omylom (chcela zverejniť iba jeden pár, ale urobila chybu pri prepise konštanty IV zo štandardu a tak pridala druhý, „správny“ pár). Detailmi ďalšieho vývoja útokov na MD5 sa zaoberať nebudeme. Pripomeňme si ale, že tri štvrte roku po zverejnení kolidujúcich správ bolo možné pomocou takzvanej metódy tunelov objavenej českým kryptológom Vlastimilom Klímom generovať kolízie za menej ako 1 minútu na notebooku, pričom Wangovej to trvalo približne 10 hodín na superpočítači.

Pozrime sa trochu detailnejšie na výsledky dosiahnuté v roku 2004 a 2005.

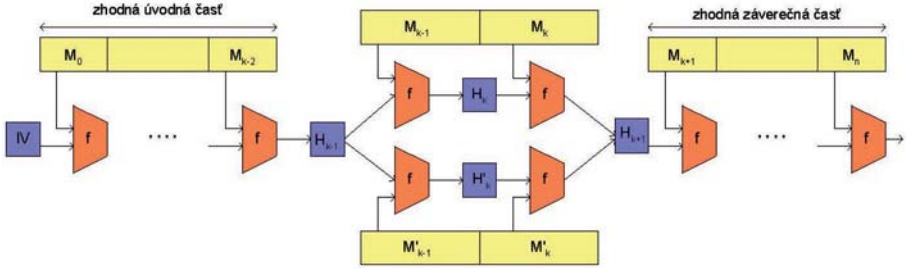
Tím profesorky Wangovej našiel metódu ako pre ľubovoľný inicializačný vektor IV konštruovať správy M a M' pre ktoré platí $h(IV, M) = h(IV, M')$, pričom správy M a M' pozostávajú každá z 2 blokov. Grafické znázornenie kolízie pre správy $M = (M_1, M_2)$ a $M' = (M'_1, M'_2)$ je na obrázku číslo 2.



Obr. 2 Schéma kolície pre MD5

Za spomenutie stojí, že správy M a M' sa líšia iba nepatrne a to tak, že $M'_1 = M_1 + C$ a $M'_2 = M_2 - C$, kde C je 512 bitová konštanta obsahujúca iba 3 nenulové bity v binárnom zápise. Kolízia ale nenastáva pre každé správy s touto diferenciou. Správy M a M' je potrebné konštruovať špeciálnym spôsobom. Práve zložitosti tejto konštrukcie sa týkali všetky následné urýchlenia čínskeho útoku.

Vďaka možnosti hľadania kolízii pre ľubovoľnú hodnotu IV je možné vytvárať dlhšie kolidujúce správy M a M' nasledujúcim spôsobom. Na začiatku oboch správ sa môže nachádzať ľubovoľne dlhý, avšak pre obe správy rovnaký úsek, $M_i = M'_i$, $i = 1, \ldots, k$. Pri výpočte hodnoty hash sa teda bude zhodovať kontext H_k pre správu M s kontextom H'_k pre správu M', $H_k = H'_k$. Následne pomocou jednej z publikovaných metód určíme 4 bloky A_1, A_2 a B_1 a B_2 také, že pre kontext H_k (v tomto prípade chápaný ako nový inicializačný vektor) bude platiť, že $h(H_k, A_1A_2) = h(H'_k, B_1B_2)$. Bloky A_1 a A_2 pripojíme k správe M a bloky B_1 a B_2 k správe M'. Na koniec oboch správ ešte môžeme pripojiť ľubovoľne dlhú, pre obe správy zhodnú záverečnú časť. Výsledné správy, líšiace sa na pozíciách k a $k+1$, budú mať rovnakú hodnotu hash funkcie. Tento postup ilustruje obrázok číslo 3.



Obr. 3 Tvorba kolidujúcich správ

Využitie kolízii pre MD5 v certifikátoch, prvá metóda

Ako sme v úvode spomenuli, bezpečnosť schém pre elektronický podpis je závislá taktiež na bezkolíznosti použitej hešovacej funkcie. Po objavení metódy na hľadanie kolízií pre MD5 bolo len otázkou času, kedy sa objavia dve kolidujúce zmysluplné správy. Stalo sa tak v roku 2005 a boli to dva certifikáty s rovnakou hash hodnotou ale rozdielnymi údajmi. Pretože ich hash bol rovnaký, bol rovnaký aj podpis certifikačnej autority používajúcej podpisový algoritmus MD5 RSA 2048.

Zamyslime sa spôsobom, ako by sa dali takéto certifikáty pomocou vyššie popísanej metódy vytvoriť. V prvom rade si musíme uvedomiť, že oba certifikáty sa môžu líšiť iba v dvoch blokoch, inak musia byť úplne identické. Otázka teda je, kam tento rozdiel vniesť. Náš prvý nápad by mohlo byť vytvorenie dvoch certifikátov s rozdielnym predmetom. Následne by sme si mohli nechať podpísať od certifikačnej autority používajúcej algoritmus MD5 certifikát na naše meno. Pomocou kolidujúceho certifikátu by sme potom veselo podpisovali elektronické

faktúry, pretože podpis certifikačnej autority pod týmto certifikátom (okopírovaný z prvého certifikátu) by bol platný a teda príjemca faktúry by certifikát overil. Ak by ale prišlo na platenie, ukázalo by sa, že certifikát s daným ID bol v skutočnosti vystavený na iné meno a teda podpisy pod faktúrami nie sú platné.

Problém je v tom, že hodnoty rozdielnych blokov dané algoritmom pre hľadanie kolízií neovplyvníme. Museli by sme teda vystaviť certifikáty na mená obsahujúce zdanlivo náhodných dvakrát 512 bitov, ktoré sa líši iba v 6 bitoch. Takéto certifikáty by však v praxi asi neboli použiteľné.

Riešením toho problému je ukryť rozdielne kolidujúce bloky do vhodne zvoleného modulu verejného kľúča. Takto dostaneme 2 certifikáty vystavené na rovnakú osobu ale s rozdielnymi verejnými kľúčmi. Možnosť zneužitia takýchto certifikátov je podobná ako v predošlom prípade. Detaily tejto konštrukcie tu popisovať nebudeme a záujemcov odkážeme na článok web stránku projektu [2].

Využitie kolízii pre MD5 v certifikátoch, druhá metóda

Po chvíľke upokojenia v oblasti výskumu kolízii MD5 prišla koncom predošlého roku správa, že výskumný tým ktorý publikoval kolidujúce certifikáty objavil metódu takzvaných chosen-prefix kolízii pre funkciu MD5, teda kolízii pre voliteľný prefix správ [3].

Hlavným rozdielom oproti doterajším metódam je to, že kým doposiaľ sme pre zvolenú hodnotu IV vedeli konštruovať správy M a M' s $h(IV, M) = h(IV, M')$, nová metóda dokáže hľadať kolidujúce správy M a M' pre rozdielne vstupné inicializačné vektory IV a IV'. Na prvý pohľad malý rozdiel, ktorý má však veľké dôsledky. Pri konštrukcii kolízie si teda zvolíme hodnoty IV a IV' a pomocou nového algoritmu nájdeme správy M a M', pre ktoré bude platiť rovnosť $h(IV, M) = h(IV', M')$.

Popíšme si rozdiel oproti predchádzajúcim metódam pomocou nasledujúceho obrázku. Označme voliteľnú časť správ plnou čiarou a dopočítanú časť správ čiarou prerušovanou. Na obrázku naľavo popisujúcom pôvodnú metódu vidíme, že sa nám pre obe správy zhoduje úvodná časť (na obrázku 4 označená písnemon A), nasleduje nekontrolovateľná časť rozdielnych blokov (označená písmenom F) a na záver máme opäť zhodnú časť pre obe správy (označená písmenom Z). Obrázok napravo ilustruje novú metódu. Správy nám opäť môžu začínať rovnakým úsekom (na obrázku označené A), ale následne sa líšia ľubovoľným – voliteľným spôsobom (na obrázku označené D). Pri výpočte hodnoty hešovacej funkcie v tomto bode dostávame pre obe správy rozdielny kontext. H_k pre správu M a H'_k pre správu M'. Tento kontext použijeme ako IV a IV' pre náš nový algoritmus, ktorý nám určí správy N a N' pre ktoré platí $h(IV, N) = h(IV', N')$. Stačí teda pripojiť bloky správy N za bloky správy M a bloky správy N' za správu M' a dostávame 2 správy s rovnakou hodnotou hash funkcie (na obrázku označené F). Na záver oboch správ môžeme ešte rovnako ako v predošlom prípade pripojiť

ľubovoľný, pre obe správy rovnaký záverečný úsek (označená písmenom Z). Je zrejmé, že nová metóda nám dáva oveľa viac možností pri konštrukcii zmysluplných kolízií.

Obr. 4 Schémy tvorby kolidujúcich správ

Rovnaký tým vedcov zároveň pripravil kolidujúce certifikáty využívajúce túto novú metódu. Hlavný rozdiel oproti predošlým kolidujúcim certifikátom spočíva v tom, že máme možnosť rozdielne voliť úvodnú časť certifikátu – predmet, sériové číslo, atď. Pritom je iba na nás ako a kde sa budú tieto certifikáty líšiť. Následne pomocou nového algoritmu vypočítame kolidujúce správy pre dané inicializačné vektory určené počiatočnými časťami certifikátov. Vypočítané (a teda neovplyvniteľné) rozdiely zanesieme rovnako ako v predošlom prípade do vhodne zvolených verejných kľúčov.

Dostávame teda 2 certifikáty s rozdielnymi položkami v certifikáte a rozdielnym verejným kľúčom, ale rovnakou hodnotou hešovacej funkcie, a teda aj rovnakým podpisom od certifikačnej autority.

Detaily novej metódy bude možné po uverejnení nájsť v článku [4].

Bezkolíznosť druhého rádu u MD5

V tomto príspevku sme si popísali 2 techniky na vytvorenie 2 certifikátov s rovnakou hodnotou hešovacej funkcie MD5. V oboch popísaných prípadoch išlo o kolízie prvého rádu.

Pokiaľ sme teda od augusta roku 2004 podpísali nejaký dokument, alebo súbor pomocou algoritmu používajúceho MD5, môže sa nám stať, že sa sme zároveň „omylom“ podpísali aj niečo iné.

Doposiaľ ale nie je známy žiadny spôsob umožňujúci vytvárať kolízie druhého rádu pre funkciu MD5. To znamená, že pre pevne danú správu M zatiaľ nie sme schopní nájsť inú správu s rovnakou hodnotou hash. Táto úloha je omnoho zložitejšia ako úloha nájsť kolíziu prvého rádu a nepredpokladá sa, že by sa v blízkej dobe našlo jej efektívne riešenie.

Všetky správy podpísané pomocou algoritmu používajúceho funkciu MD5 vytvorené pred objavením metód na výpočet kolízií teda zostávajú neohrozené.

Prípad SHA-1 a SHA-0

Vo februári roku 2005 profesorka Wangová opäť prekvapila. Tentoraz publikovala svoje výsledky týkajúce sa hľadania kolízií hešovacej funkcie SHA-1 [5]. Ich praktické využitie je pre ich stále veľkú výpočtovú zložitosť nemožné, ale pôvodnú zložitosť nájdenia kolízie prvého rádu znížila svojím postupom z hodnoty 2^{80} danej narodeninovým paradoxom na hodnotu 2^{69}.

Po Wangovej článkoch vydal americký úrad pre štandardy a technológie NIST ktorý zodpovedá za štandardy o hešovacích funkciách odporúčania k používaniu hešovacích funkcií.

NIST doporučuje okamžité ukončenie používania funkcie MD5 v certifikátoch ale naďalej prehlasuje SHA-1 za bezpečnú. Pokiaľ je to možné, doporučuje používať novú triedu funkcií SHA-2 a predpokladá, že tieto funkcie definitívne nahradia SHA-1 do roku 2010, ktorá bude dovtedy podľa NISTu bezpečná.

Od roku 2004 sa taktiež objavilo niekoľko takzvaných generických útokov na hešovacie funkcie založené na iteratívnom princípe. Tieto výsledky sa teda týkajú každej hešovacej funkcie bez ohľadu na konštrukciu jej kompresnej funkcie, a sú teda aplikovateľné ako na MD5, tak aj SHA-256 [6, 7].

V prípade SHA-0 (dĺžka výstupu 160 bitov) uverejnila Wangová metódu hľadania kolízii so zložitosťou menej ako 2^{39}, čo je oveľa menej ako 2^{80} z narodeninového paradoxu [8].

Z praktického hľadiska to znamená, že funkcie MD5 ani SHA-0 by nemali byť používané v žiadnej aplikácii, ktorej bezpečnosť je ohrozená schopnosťou nachádzať kolízie prvého druhu. Na druhú stranu medzi kryptológmi sa už dávno predtým vedelo o istých slabinách funkcií triedy MD a funkcia SHA-0 nikdy ani nemala byť používaná.

Záver

Na záver príspevku sa ešte v skratke pozrieme na ďalší predpokladaný vývoj v oblasti hešovacích funkcií.

Ako sme už spomenuli, v súčasnosti najrozšírenejšou hešovaciou funkciou je pravdepodobne funkcia SHA-1. Medzi kryptológmi sa jej ale neprikladá dlhá životnosť, a je iba otázkou pár rokov (v horšom prípade pár mesiacov, podľa NISTu minimálne 3 roky), kedy bude existovať efektívna metóda hľadania kolízií prvého rádu pre SHA-1. NIST odporúča nahradiť SHA-1 funkciami z triedy SHA-2. Tieto funkcie sú ale založené na rovnakých technikách ako funkcia SHA-1 a teda trpia rovnakými problémami ako ostatné podobné funkcie. Naviac návrhové kritéria týchto funkcií navrhnutých americkou NSA neboli zverejnené, čo taktiež vyvoláva istý stupeň nedôvery.

Vo svete vzniklo niekoľko nových návrhov na hešovacie funkcie budúcnosti. Spomeňme napríklad prácu Belgického kryptológa Arjena Lenstru na hešovacích

funkciách VSH založených na zložitosti výpočtu modulárnej odmocniny [9], alebo nám geograficky bližšiu iniciatívu Vlastimíla Klímu na konštrukciu hešovacích funkcií pomocou takzvaných špeciálnych blokových šifier [10]. Súdiac ale podľa súčasného stavu, definovanie štandardu nových tried hešovacích funkcií zaberie kryptologickej obci ešte veľmi dlhú dobu.

Súčasné odporúčania sú ale jasné. Pre aplikácie ktoré MD5 používajú, je potrebné zvážiť, nakoľko ich ohrozuje možnosť nájdenia kolízií prvého rádu. Pre nové aplikácie odporúčame používať rozšírenia SHA-1 – funkcie SHA-256, SHA-384 s SHA-512, ale samotnú SHA-1 už nie.

Literatura

[1] Wang, ., Feng, D., Lai, X., Yu, H. *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD.* rump session CRYPTO 2004, Cryptology ePrint Archive, Report 2004/199. http://eprint.iacr.org/2004/199

[2] http://www.win.tue.nl/∼bdeweger/CollidingCertificates

[3] http://www.win.tue.nl/hashclash/TargetCollidingCertificates

[4] Stevens, M., Lenstra, A., Weger, de. B. *Target Collisions for MD5 and Colliding X.509 Certificates for Different Identities.* prijaté na EuroCrypt 2007.

[5] Wang, X., Yin, Y. L., Yu, H. *Finding collisions in the full SHA1.* Advances in Cryptology – CRYPTO 2005, Lecture Notes in Computer Science Vol. 3 621, pp. 17–36, Springer-Verlag, 2005.

[6] Joux, A. *Multicollisions in Iterated Hash Functions.* Advances in Cryptology – CRYPTO 2004, Lecture Notes in Computer Science Vol. 3 152, pp. 306–316, Springer-Verlag, 2004.

[7] Kelsey, J., Schneier, B. *Second Preimages on n-Bit Hash Functions for Much Less than 2^n*, Advances in Cryptology – EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3 494, pp. 474–490, Springer-Verlag, 2005.

[8] Wang, X., Yin, Y. L., Yi, Y. *Efficient collision search attacks on SHA-0.* Advances in Cryptology – CRYPTO 2005, Lecture Notes in Computer Science Vol. 3 621, pp. 17–36, Springer-Verlag, 2005.

[9] Contini, S., Lenstra, A. K., Steinfeld, R. *VSH, an Efficient and Provable Collision Resistant Hash Function.* Cryptology ePrint Archive, Report 2005/193. http://eprint.iacr.org/2005/193

[10] http://cryptography.hyperlink.cz/SNMAC/SNMAC_CZ.html

Jak opravdu anonymně vystupovat na Internetu

Petr Břehovský

E-mail: breh@breh.cz

WARNING: Normal Internet/Email Technology is the Most Comprehensive Surveillance System Ever Invented.

André Bacard

. . . a zákonnou povinností každého ISP by mělo být upozornit na to své zákazníky.

1. Jaké stopy zanecháváme a kdo všechno má tyto informace k dispozici
2. Běžná opatření
3. Anonymní remailery
4. Proxy servery
5. Tor a další anonymizery služeb s nízkou latencí
6. Darknets, DeepWeb a dark Internet
7. Zdroje a další užitečné odkazy

Jaké stopy zanecháváme a kdo všechno má tyto informace k dispozici

Z pohledu běžného uživatele není technologie Internetu nijak problematická. Umožňuje vzájemnou komunikaci, získávání informací, nakupování, elektronické bankovnictví a další užitečné věci. Někdo si možná uvědomuje, že je vhodné některá citlivá data šifrovat, aby je nemohl přečíst ten, komu nejsou určena. Šifrováním se však v tomto příspěvku zabývat téměř nebudeme. Konec konců v dnešním světě pravděpodobně nemusí být až tak důležité „co“, ale spíše „kdo s kým“, neboli kam se zrovna připojujeme, co si prohlížíme a s kým komunikujeme. A tohle „kdo s kým“ je s rostoucí uživatelskou přívětivostí technologií na

první pohled zcela pod kontrolou uživatele. Vždyť adresář je bezpečně uložen v e-mail klientu, nebo WEB prohlížeči. Pod povrchem uživatelského rozhraní je však všechno jinak. Komunikace probíhá ve velké většině případů IP protokolem, do kterého jsou zapouzdřeny TCP segmenty a UDP datagramy.

Na následujícím obrázku je uvedena struktura těchto paketů a jsou zvýrazněna pole, která jsou z hlediska „kdo s kým“ kritická, a pole, která pomáhají identifikovat typ operačního systému uživatele (více informací o pasivní identifikaci OS lze najít například zde: http://lcamtuf.coredump.cx/p0f.shtml).

IP paket

Verze	**Délka hlavičky**	**TOS**	**Celková délka**
Identifikace		**Návěští**	**Offset fragmentu**
TTL	**Protokol**	**Checksum hlavičky**	
IP adresa zdroje			
Cílová IP adresa			
Volby			
Data			

UDP datagram

Zdrojový port	**Cílový port**
Délka	**UDP checksum**
Data	

TCP segment

Zdrojový port		**Cílový port**	
Sekvenční číslo			
ACK			
Délka hlavičky	**Rezerva**	**Návěští**	**Velikost okna**
TCP checksum		**Ukazatel na urgentní data**	
Volby			
Data			

Je tedy zřejmé, že uvedené pakety poskytují informace o tom jaké zařízení s kterým dalším zařízením komunikuje, a kterou službu používá. Také není příliš

obtížné zjistit, komu dané adresy a provozované služby patří. A to buď z veřejných zdrojů (WHOIS, Google, apod.), nebo v databázích telekomunikačních firem.

Co se týče elektronické pošty, jsou v hlavičkách e-mailů zaznamenávány adresy odesílatele, příjemce i jména serverů, přes které dopis procházel:

```
Return-Path: <apache@staff.ipex.cz>
Received: from adriana.gin.cz (mx.ipex.cz [212.71.175.4])
        by icicle.breh.cz (8.12.8/breh) with ESMTP id l0OFtQSx010932
        for <breh@breh.cz>; Wed, 24 Jan 2007 16:55:26 +0100
Received: from staff.ipex.cz (farm1-dg.ipex.cz [212.71.175.26])
        by adriana.gin.cz (Postfix) with ESMTP id 16E4EDC057
        for <breh@breh.cz>; Wed, 24 Jan 2007 17:00:26 +0100 (CET)
Received: by staff.ipex.cz (Postfix, from userid 48)
        id 088CC100C203; Wed, 24 Jan 2007 17:00:23 +0100 (CET)
To: breh@breh.cz
From: info@ipex.cz
```

Tyto informace jsou uchovávány nejen v lozích klientů a serverů, ale některé také na routerech poskytovatelů připojení (ISP, zaměstnavatel) a poskytovatelů infrastruktury, na zařízeních zaměstnavatele a samozřejmě putují po síti, takže k nim má přístup každý, kdo je připojen do toho správného segmentu sítě.

Navíc existují metody analýzy datových toků pomocí kterých se lze pokusit na základě paternů nalezených v komunikaci odvodit přenášené informace, nebo alespoň jejich typ. A to i v případě šifrovaných dat.

Úvahu o tom, kdo všechno má, nebo může mít tato data k dispozici a k čemu je může použít nechť provede každý sám.

Běžná opatření

V první řadě je vhodné analyzovat data, která náš počítač po připojení do sítě opouštějí i ta, která na něj přicházejí. Ruku na srdce. Víme vůbec co všechno je z našeho počítače do sítě odesíláno? Kromě dat o kterých víme, to může být například automatické odesílání registrací aplikací, automatická kontrola dostupnosti nových aktualizací, automatické odesílání popisů chybových stavů (bug report) atd. O datech odesílaných addwarem, a podloudně nainstalovanými trojskými koni ani nemluvě.

Nevyhovuje ani implicitní konfigurace prohlížeče. Pokud si chceme zachovat alespoň minimální soukromí, měli bychom například v prohlížeči Mozilla nastavit následující:

Zapnout filtrování popup oken
Zakázat Javu
Neodesílat skutečnou e-mail adresu ftp serverům

Neakceptovat cookies, nebo alespoň zapnout varování a analyzovat je
Zakázat animace
Neukládat data z formulářů
Neukládat hesla
Neinstalovat Flash plugin

Kontrolu nad daty plynoucími z našeho počítače a přicházejícími ze sítě můžeme zvýšit instalací lokálního proxyserveru. Vhodným se zdá například Privoxy (http://www.privoxy.org/), kerý dovoluje filtrování toků dat, manipulaci s cookies, řízení přístupu, blokování bannerů, adware, popup oken apod.

Samozřejmostí je firewall, který je nastaven tak, aby blokoval veškerá data přicházející zvenčí a odchozí data přesměrovával na proxyserver.

A jako poslední krok nelze nedoporučit kontrolu komunikace síťovým analyzátorem.

Anonymní remailery

Popis technologií zvyšujících soukromí začneme remailery, které jsou schopné ho zaručit v maximální míře. Je to možné díky dávkové povaze zpracování e-mailů. Jak uvidíme v dalších kapitolách, je stejný úkol v případě interaktivních protokolů mnohem složitější.

Základní princip anonymizace elektronické zprávy spočívá v tom, že zprávu odešleme na speciální poštovní server (remailer) ve formě žádosti o doručení na cílovou adresu. Remailer (nebo systém remailerů) naši zprávu přepošle adresátovi s tím, že změní odchozí adresu (naši e-mail adresu) na adresu vygenerovanou a obě adresy uloží do interní databáze. Příjemce zprávy tedy dostane e-mail přicházející z adresy vygenerované remailerem. Pokud na tuto adresu odpoví, odpověď dojde remaileru, ten nahradí generovanou adresu, skutečnou adresou (najde ji uloženou v databázi) a dopis přepošle zpět původnímu odesílateli.

Odesílatel	**REMAILER**	**Příjemce**
From: `breh@breh.cz` To: `remailer@rem.dom` :: `europen@europen.cz` text	Databáze ... `breh@breh.cz` = `856784@rem.dom`	From: `856784@rem.dom` To: `europen@europen.cz` text

Protože jsou žádosti o přeposlání obvykle šifrovány, ISP neví komu e-mail ve skutečnosti odesíláme, a protože je odchozí adresa generována remailerem, příjemce neví od koho dopis ve skutečnosti pochází.

Anonymní remailery dělíme na pseudoanonymní (princip popsaný výše) a anonymní (opravdu anonymní). Pseudoanonymní remailery (nym servery) jsou

založeny na důvěře provozovateli serveru. Tzn. musíme věřit technickým dovednostem a personální integritě provozovatele, který musí zajistit bezpečnost (nedostupnost) logů a záznamů v databázi remaileru. Velmi poučná je z tohoto hlediska historie remaileru http://anon.penet.fi (http://en.wikipedia.org/wiki/Penet_remailer). V případě pseudoanonymního remailerů, musíme také počítat s tím, že veškeré naše transakce s remailerem budou logovány na zařízeních mezi námi a remailerem, logovány samotným remailerem a linkovány na náš přístupový bod do Internetu (telefonní číslo, port přepínače ISP, port DSLAMu apod.).

Oproti tomu anonymní remailery realizují anonymitu principiálně, takže nezávisí na lidském faktoru, nebo slabosti jednotlivého prvku architektury. Rozlišujeme remailery tří typů:

- Typ I – Cypherpunk
- Typ II – Mixmaster
- Typ III – Mixminion

Remailery typu I akceptují zprávy zpravidla šifrované pomocí PGP, nebo GPG, z jejichž hlaviček odstraňují všechny informace, které slouží k identifikaci odesílatele. Cílová adresa (adresa příjemce je specifikována uvnitř zašifrované zprávy). Zpráva může být odesílatelem směrována přes několik remailerů aby byla snížena pravděpodobnost odhalení odesílatele, přičemž dochází k dalšímu šifrování zpráv (přidání další vrstvy, tentokrát s cílovou adresou následujícího remaileru v řetězci). Na odeslanou zprávu nelze odpovědět, protože původní adresa odesílatele se nikde neuchovává.

Remailery typu II odesílají zprávy rozdělené na pakety, které putují v pozměněném pořadí. Je tak ještě ztíženo trasování zprávy. Je možné uměle zadržovat zprávy na jednotlivých serverech, takže je velmi složité odhadnout zda uživatel A odeslal zprávu uživateli B i když jsou oba pod kontrolou (analýza časů odeslaných a přijatých e-mailů). Oproti typu I však k odeslání zprávy potřebujeme specializovaného poštovního klienta.

Typ III se skládá z množství serverů (mixes), které přijímají zprávy rozdělené na pakety konstantní velikosti, přehazují jejich pořadí a odesílají je dále směrem k příjemci. Každý paket putuje přes síť serverů jinou cestou, a ani jeden ze serverů tak nezná vazbu: původní odesílatel – konečný příjemce. Jednotlivé pakety jsou na každém serveru rozšifrovány, poté je identifikován další server v řetězci a před odesláním znovu šifrovány veřejným klíčem dalšího serveru. Mixminion navíc umožňuje odesílat zprávy anonymním příjemcům (odesílatel není schopen identifikovat příjemce zprávy) a je navržena integrace s nym servery.

Proxy servery

V případě interaktivních protokolů je ochrana soukromí realizována pomocí anonymizujících proxy serverů. Myšlenka je podobná jako u nym serveů. Nenapojujeme se přímo na server poskytující službu, ale na proxy server, který spojení s cílovým serverem zprostředkuje. Cílový server tak nezná IP adresu klienta (vidí pouze že se na něho připojuje proxy server) a poskytovatel připojení vidí, že se uživatel připojuje na proxy server, a není tedy schopen určit cílovou službu (pouze v případě, že nepřečte z datového toku http příkaz CONNECT). Existuje řada anonymizujících proxy serverů pro HTTP/HTTPS, (http://www.findproxy.org/). Aby byla práce s nimi pohodlnější, můžeme použít Switchproxy (http://addons.mozilla.org/cs/firefox/addon/125), rozšíření pro Mozillu, které umožňuje snadnou správu proxy serverů i jejich řetězení.

Bezpečnost tohoto řešení je však přibližně na stejné úrovni jako pseudoanonymní remailer, řada veřejných proxy serverů podporuje pouze http a použití jiných protokolů je značně problematické, ne-li nemožné. Tyto nedostatky odstraňuje TheOnionRouter (Tor http://tor.eff.org/index.html.en)

Tor a další anonymizery služeb s nízkou latencí

Tor je síť serverů která umožňuje zachovat soukromí během přístupu k WEB serverům, publikování dokumentů, komunikaci pomocí IRC, instant messagingu, ssh a dalších aplikacích komunikujících pomocí TCP protokolu. Tok dat je směrován od klienta k serveru a zpět skrz síť serverů tak, aby ISP na straně klienta nebyl schopen určit s kterým serverem klient komunikuje a aby server nedovedl tohoto klienta identifikovat. Tor je v současné době pravděpodobně nejdokonalejĽí síť tohoto typu, která poskytuje soukromí pro služby s nízkou latencí. Požadavek na nízkou latenci přenášených služeb umožňuje určité typy útoků (http://www.cs.colorado.edu/department/publications/reports/docs/CU-CS-1025-07.pdf), ale nebezpečí jejich dopadu na běžného uživatele, který chce pouze zvýšit své soukromí při práci s Internetem je minimální.

Tor navíc dovoluje definovat takzvané skryté služby, které umožňují publikovat informace (například na WEB serveru) aniž by kdokoli mohl určit, kde se daný WEB server ve skutečnosti nachází.

Ve stádiu vývoje je síť I2P (http://www.i2p.net/) jejíž srovnání s ostatními sítěmi tohoto typu lze nalézt na http://www.i2p.net/how_networkcomparisons.

Darknets, DeepWeb a dark Internet

Obecně jsou darknets sítě, určené pro sdílení informací mezi účastníky, kteří si přejí zachovat anonymitu. Většinou se jedná o sítě, které jsou dostupné pouze

pro konkrétní skupinu lidí, tzn. „zbytek světa„ do nich nemá přístup. Jedná se vlastně o Internet uvnitř Internetu. Příkladem implementace takové sítě může být Freenet (http://freenetproject.org/).

DeepWeb jsou v podstatě WEB servery, na které nevedou linky a jejichž obsah lze jen těžko indexovat vyhledávači. Toho lze dosáhnout například vygenerováním obsahu stránek na základě formy dotazu z dat uložených v databázích, zveřejněním pouze netextových dokumentů, zaheslováním přístupu ke stránce apod.

Dark Internet je adresní prostor IP protokolu, který není dostupný z Internetu. Nejedná se o privátní adresní prostor tak jak ho známe z RFC 1918, ale o běžné adresy, které nejsou dostupné z důvodu chybné konfigurace routerů, výpadků technologie, nebo zákeřných záměrů.

S rostoucí dostupností síťových technologií (např. 802.11) začínají vznikat sítě, které nevyužívají síťovou infrastrukturu Internetu, ani infrastrukturu telekomunikačních operátorů ale jsou vybudovány na infrastruktuře vlastní.

Zdroje a další užitečné odkazy

Citát v úvodu http://www.andrebacard.com/remail.html

Pasivní identifikace OS: http://lcamtuf.coredump.cx/p0f.shtml

Implementace Toru na LiveCD: http://theory.kaos.to/projects/AnonymOS.pdf

Privoxy http://www.privoxy.org/

Penet remailer http://en.wikipedia.org/wiki/Penet_remailer

Mixminion http://mixminion.net/
http://peculiarplace.com/mixminion-message-sender/ (Windows GUI)

Anonymizující WEB proxy http://www.findproxy.org/

Switchproxy https://addons.mozilla.org/cs/firefox/addon/125

Tor a i2p http://tor.eff.org/index.html.en http://www.i2p.net/

DarkNets http://en.wikipedia.org/wiki/Darknet

Freenet http://freenetproject.org/

Archiv dokumentů o anonymní komunikaci
http://freehaven.net/anonbib/topic.html#Anonymous_20communication

Kovářova kobyla...

Radoslav Bodó

E-mail: bodik@civ.zcu.cz

Abstrakt

V dnešní době rychlého a hojně rozšířeného internetu jsou hackerské útoky na denním pořádku. Díky kvalitním vyhledávačům je navíc velmi snadné najít si různé informace o zdokumentovaných technikách, hotových nástrojích a nezřídka i potenciálních obětech útoku. V tomto příspěvku ukážeme případovou studii forenzní analýzy napadeného počítače s OS Linux.

1 Úvod

Bezpečnostní incidenty začínají v praxi několika způsoby. Buď přijde oznámení cizího subjektu o útocích nějakého stroje, nebo hlášení uživatele o podezřelém chování, nebo přijde hlášení od některého z IDS[1] systémů domovské sítě. Podle důležitosti daného zařízení provede bezpečnostní technik průzkum (automatický nebo osobní), na jehož základě je vydáno patřičné rozhodnutí o dalších akcích vedoucích k vyřešení události. Tomuto průzkumu a interpretaci nalezených údajů se říká *forenzní analýza*.

2 Získání dat

Nejdříve je nutné vyhledat uživatele daného PC a vyzpovídat ho, kdy a po jakých akcích se závadné chování objevilo, nebo alespoň na jaké období by mohl mít podezření. Dále se musí zjistit k jakým účelům byla stanice používána a případně porovnat uvedená data se statistikami datového provozu v domácí síti. Ta mohou odhalit IP útočníka nebo dalších obětí.

V tomto případě bylo hlášeno podezřelé chování uživatelem a v IDS Snort[8] domovské sítě byl nalezen záznam o úspěšném spuštění programu `bin/id` v rámci komunikace s webovým serverem, který na daném stroji běžel.

[1] Intrusion Detecnion System

```
---- cut IDS event ---
IPv4: 147.228.xxx.200 -> 195.34.xxx.96
     hlen=5 TOS=0 dlen=1496 ID=2773 flags=0 offset=0 TTL=63 chksum=59931
TCP:  port=80 -> dport: 2608  flags=***A**** seq=226365842
...
...
d><font face=Verdana size=-2 color=red><b>   
Linux mistik 2.6.14.6-grsec #1 SMP Thu May 11 11:57:34 CEST 2006 i686
      GNU/Linux<br>
   -<br>   linux-gnu<br>
   Apache/2.0.54 (Debian GNU/Linux) PHP/4.3.10-18<br>
   uid=33(www-data) gid=33(www-data) groups=33(www-data)<br>
   /sbin/apache   ( drwxrwxrwx )
</b></font></font></td></tr></
...
textarea name=report cols=121 rows=15>uid=0(root) gid=0(root)
groups=33(www-data).</textare
...
---- cut IDS event ---
```

Jedná se o záznam pravidla 498[2] a potenciálním útočníkem bylo ip 195.34.xxx.96. Toto ukazuje, že stroj poslal identifikaci uživatele, pod kterým byla spuštěna webová služba. V praxi to bývá jeden z prvních příkazů, který útočník na vzdáleném systému spouští (druhým nejčastějším příkazem je `,uname -a‘`). Dále, že stroj napaden byl, pravděpodobně pomocí webové služby. Nicméně ani jedno není zaručené.

Pokud je PC stále zapnuto, je vhodné udělat alespoň základní průzkum na běžícím vzorku užitím standardních nástrojů `ps`, `ls`, `netstat`, `strace`, `lsof`, ... a pokusit se vyhledat nesrovnalosti s tím, co by uživatel v systému očekával. Je vhodné donést si tyto nástroje s sebou na externím médiu (případně i staticky kompilované), uložit seznam otevřených síťových spojení, socketů a souborů, zavedených jaderných modulů a v extrémních případech udělat otisk paměti[3].

Při základním průzkumu je cílem najít aktivní známky zneužití systému, najít běžící procesy útočníka či automatický malware. Tento průzkum zahrnuje kontrolu běžících procesů např. podle názvu (cron vs. crond; netscan; ...), nesrovnalostí v `/proc`[3], zběžnou kontrolu systémových souborů, prostředí, logů, atd. Záleží čistě na zkušenosti a zvědavosti vyšetřovatele. Nakonec je možné použít některý specializovanější nástroj pro hledání rootkitů (`chkrootkit`, `rkhunter`, `rpscan`, ...). Při tomto prvotním průzkumu je důležité nedělat žádné operace zápisu na disk, abychom nepoškodili důkazní materiál, jež bude analyzován později.

Někdy se též doporučuje co nejdříve odpojit počítač od zbytku sítě (jednoduchým vytažením kabelu). Nicméně je dobré míti na paměti, že to může

[2] http://www.snort.org/pub-bin/sigs.cgi?sid=498

[3] Symlink `/proc/<PID>/@exe` ukazuje na binárku jiného jména než ukazuje `/proc/<PID>/cmdline`

aktivovat v malwaru tzv. deadman-switch – proceduru, jež zajistí úklid proti prozrazení [1]. Stejně tak ji může vyvolat standardní vypnutí systému. Proto se někdy doporučuje ukončit běh stroje před snímáním duplikátů (viz níže) hrubou silou, to však může vést ke ztrátě dosud nezapsaných dat.

Následně je nutno vytvořit forenzní duplikát disku/systému pro hlubší analýzu. To proto, aby mohl uživatel dále pokračovat v práci (může jít i o planý poplach), nebo aby mohl přejít rovnou k reinstalaci systému a dále abychom samotným vyšetřováním nepoškodili hledané důkazy.

K vytvoření duplikátu je potřeba nějaké LiveCD[4] (Knoppix, Helix, ...) a úložiště kam data uskladnit (jiné PC popřípadě externí disk). Po spuštění LiveCD, se pomocí `dd` a `nc` (ev. `ssh`) vytvoří kopie lokálních disků a zapíše se informace o rozložení dat na nich uložených. Je vhodné (nikoli však nutné) sejmout obrazy jednotlivých oddílů zvlášť (kvůli snazší manipulaci) a pořídit si i kontrolní součty zkoumaných dat. Z kopií je možné nakonec uložená data číst a provádět libovolné analýzy.

```
sklad$ nc -l -p 1234 > /tmp/mistik.hda1

mistiKnopix$ dd if=/dev/hda1 | nc sklad 1234
mistiKnopix$ cat <hda1>/etc/fstab >sklad> mistik.fstab
mistiKnopix$ fdisk -l /dev/hda >sklad> mistik.fdisk
mistiKnopix$ md5sum /dev/hda1 >sklad> mistik.hda6.md5

sklad$/dukazy/mistik#  mount -o ro,loop,noexec -t ext3 mistik.hda5 mounts/usr
```

3 Odhalení průnikové cesty

Výchozími body pro forenzní analýzu jsou: typ bezpečnostního incidentu, zachovaný stav OS, očekávaný stav OS a dochované záznamy o činnosti – logy. Použité vyšetřovací nástroje závisí na zvyklostech vyšetřujícího, ale nejčastější jsou jimi klasické utility: `ls`, `od`, `grep`, `find`, `md5sum`, `ldd`, `strings`, `dd`, `file`, `lsattr`, `objdump`, ..., případně specializované nástroje pro hledání dat v neobsazeném prostoru na disku (Coroner's Toolkit, Sleuth Autopsy)[1].

Jako první se doporučuje prohlídka systémových logů, avšak s vědomím, že informace v nich nemusí být pravdivé. Další kontroly se měly zabývat následujícími položkami:

- systémová nastavení (`etc`)
- spouštěcí skripty (rc, inittab, ...)
- kontrola záměny systémových nástrojů (`ls`, `ps`, `netstat`, ...)
- suid a sgid programy a adresáře nejlépe pomocí ověřených kontrolních součtů z distribučního média

[4] Lokální systém mohl být pozměněn tak, aby útočníka schovával

- důležitá místa jako `/tmp`, `/bin`, ...
- moduly jádra operačního systému
- moduly autentizačnícho subsystému
- profily a historie příkazů všech uživatelů
- skryté či nezměnitelné (immutable) soubory
- soubory podezřelých vlastníků (numerické hodnoty)
- soubory a adresáře vlastněné provozovanými službami
- ...

Podle výše uvedeného je prvním krokem v popisovaném případu prozkoumání logu webového serveru a speciálně vyhledání záznamů související s IP nalezenou v IDS (plus následné prohledání souborů souvisejících se službou www na typické útoky php-injection[5], sql-injection a soubory touto službou/uživatelem vytvořené). V protokolu jsou nalezeny požadavky na soubor, který do systému zřejmě nepatřil (na forenzním duplikátu navíc nebyl vůbec nalezen).

```
 ---- cut var/log/apache2/access.log ---
 access.log:195.34.xxx.96 - - [13/Jan/2007:17:34:23 +0100]
"POST /ftp/incoming/z.php HTTP/1.0" 200 34840
 access.log:195.34.xxx.96 - - [13/Jan/2007:17:34:49 +0100]
"POST /ftp/incoming/z.php HTTP/1.0" 200 34974
 ...
 access.log:195.34.xxx.96 - - [13/Jan/2007:17:59:59 +0100]
"GET /ftp/incoming/z.php HTTP/1.0" 200 31099
 ---- cut var/log/apache2/access.log ---
```

Jinými slovy z toho vyplývá, že útočník dostal pravděpodobně nějak na cílový stroj vlastní skript/program, který mohl vzdáleně volat (úspěšně viz `200 OK HTTP/1.1`). Z uvedených záznamů je patrné, že byl umístěn někde v adresáři, který souvisel s další službou na stroji provozovanou – FTP serverem. Jeho nastavení ukázalo, že adresář byl přístupný pro čtení i zápis anonymním uživatelům.

```
 ---- cut var/log/auth.log ---
 Dec  5 17:11:49 mistik proftpd[7734]: mistik.zcu.cz
(proxy.lipetsk.ru[195.34.xxx.96]) -
               ANON anonymous: Login successful.
 Jan  8 04:31:56 mistik proftpd[29866]: mistik.zcu.cz
(crawl-66-249-66-10.googlebot.com[66.249.xxx.10]) -
               ANON anonymous: Login successful.
 ---- cut var/log/auth.log ---
```

V tomto okamžiku je téměř jisté, že útočník ovládl stroj pomocí *konfigurační chyby*. Přes FTP nahrál na server program, který byl schopen spustit

[5] Například `egrep -i "http://" /var/log/apache2/*`

pomocí WWW serveru a pod jeho identitou. V nastavení serveru a interpretru PHP nebylo omezeno prakticky vůbec nic (chroot, safe_mode, open_basedir, disabled_functions ...), takže činnost útočného skriptu/programu nebyla ničím limitována a mohla pohodlně využít všech dostupných prostředků a SW vybavení.

Průzkum dalších logů (`wtmp`, `xferlog`, `auth.log`) dokazuje, že útočník nebyl zřejmě pouze jeden a navíc že o této nebezpečné shodě nastavení dokázal informovat i Google kohokoliv, kdo věděl co má hledat (viz obrázky 1 a 2). Víc vodítko v podobě podezřelé IP adresy neposkytlo.

Obr. 1 Vyhledání oběti na Googlu

Následný podrobný průzkum logů a systému odhalil skript `/tmp/back`, který zevnitř stroje otevře shell po tcp na zadanou adresu a tím obejde jak nastavení většiny současných lokálních firewallů, tak i autentizaci potřebnou k přístupu k terminálové službě – tzv. *backconnect*. Dále prohlídka odhalila masivní volání skriptu `r57.php`, který byl na disku nalezen a podroben zkoumání. V poslední řadě pak odhalil informaci, že jeden z útočníků používá prohlížeč Opera v anglickém jazyce.

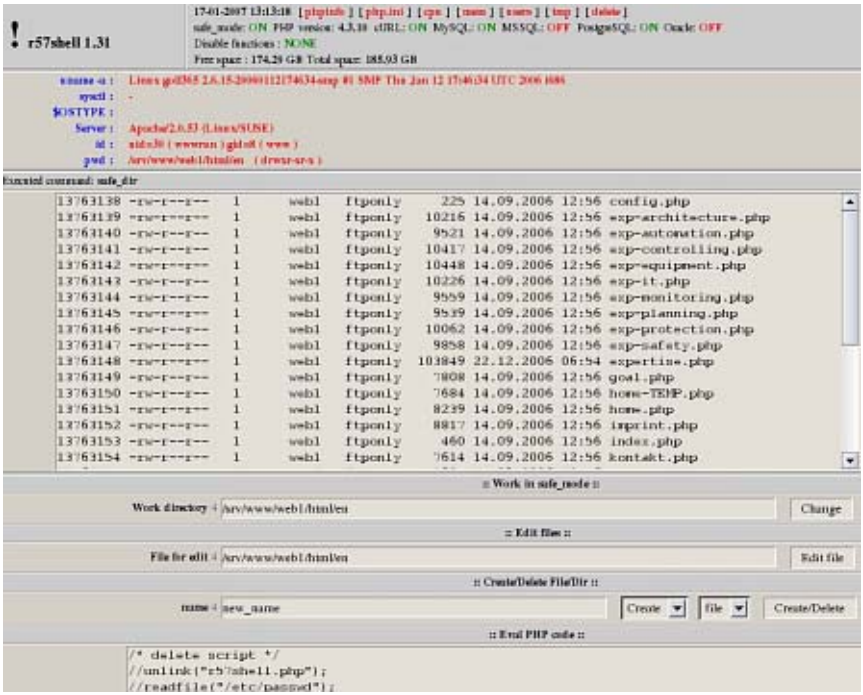
```
---- cut var/log/apache2/access.log ---
error.log.1:[client 195.34.xxx.96] script ‘ /mnt/parta/ftp/incoming/r57.php‘
      not found or unable to stat,
 referer: http://www.google.com/search?q=allinurl:r57.php&
          hl=en&lr=&client=opera&rls=en&hs=22Y&start=20
access.log.2:87.126.xx.17 - - [17/Dec/2006:14:17:38 +0100]
      "GET /ftp/incoming/r57.php HTTP/1.1" 200 32295
---- cut var/log/apache2/access.log ---
```

Obr. 2 Ukázka logu typu apache combined

4 Nalezené nástroje

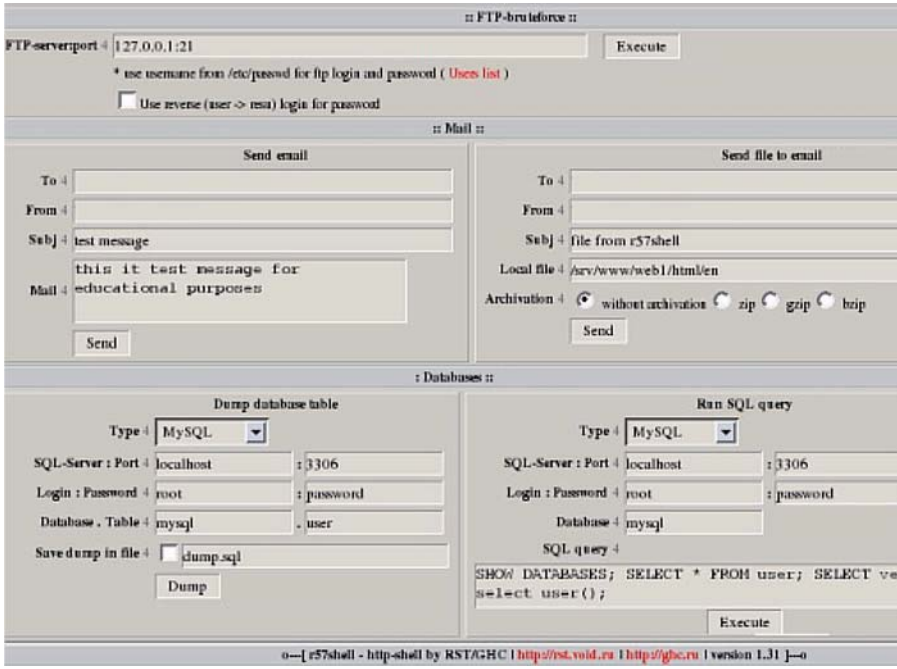
Nalezený skript `r57.php` (dále jen R57) je velmi pěknou ukázkou práce současných hackerů. R57 je jedním z veřejně dostupných útočných PHP skriptů[6], které mají za úkol zjednodušit ovládání napadeného stroje, umožnit prvotní průzkum napadeného systému a urychlit další rozšiřování sama sebe. R57 je rovnou psán dvojjazyčně (anglicky a rusky) a uživateli umožňuje (obrázky 3 a 4):

- listování adresářů
- editování stávajících a nahrávání nových souborů
- pohodlné vyhledávání souborů pomocí předdefinovaných příkazů find
- procházení databází Mysql, MS-SQL, PostgreSQL
- ftp a email klient, lámání ftp hrubou silou
- automatické promazání `/tmp`



Obr. 3 Náhled r57shell.php - 1

[6] C99, PHPshell, mambot, ...



Obr. 4 Náhled r57shell.php - 2

Navíc obsahuje R57 užitečné funkce v podobě zdrojových kódů programů (pro C a Perl) umožňující:

- připojit `/bin/sh` na určený lokální TCP socket – *bindport*
- připojit `/bin/sh` na spojení iniciované *zevnitř* stroje na zadaný cíl – *back-connect*
- vytvořit TCP tunel skrz napadený server – *datapipe*

Tyto utilitky jsou v R57 uloženy v kódování Base64, aby se mohly dobře integrovat do nosného skriptu a jednoduše tak obejít problémy se závorkováním a uvozovkováním a teoreticky může R57 v této podobě nosit i binární programy. V sobě mají implementováno ukrývání svých pravých názvů přepisováním proměnné *argv[0]* a funkce v nosném skriptu umožňují jejich pohodlnou kompilaci a spuštění přímo z webového rozhraní.

Hypoteticky lze vytvořit další skript (za pomoci funkce *datapipe*), který dokáže vytočit krycí spojení přes několik uzlů v internetu, a navíc tyto uzly najít online pomocí Googlu. Tyto dotazy se Google snaží blokovat.

Nicméně ani tvůrci tohoto útočného skriptu nedají nic jenom tak. Ve veřejné variantě R57 jsou další dvě pole, která obsahují javascriptový kód (též v Base64), odmaskované kódy jsou vloženy při zobrazení v prohlížeči do stránky a uživatel skriptu se tak sám nahlásí autorům a to pomocí veřejného webového počítadla. V konečném důsledku mohou autoři sledovat jak napadené servery, tak některé uživatele svého výtvoru.

5 Získání superuživatelských práv

Pokud je napadený stroj důležitou součástí infrastruktury sítě, případně obsahuje-li citlivá data pro napadenou organizaci, bývá zvykem ve vyšetřování dále pokračovat i po prokázání počátečního napadení. Důvodem je odhad možných škod, které útočník napáchal, či odhad důsledků jeho činnosti. Technicky bývá další hledání zaměřeno na důkazy o prolomení *uid=root*, s jehož oprávněním je útočník schopen odcizit veškeré dokumenty na stroji uložené, pokusit se zachytit uživatelská hesla nebo odposlouchávat síťový provoz na lokálním segmentu, ... Tento postup se v zásadě neliší od postupu popisovaném v kapitole 3.

Ve zkoumaném systému byla nalezena zadní vrátka v podobě nastaveného hesla pro uživatele `uid=99 (apache)`, ten nebyl v systému oprávněně a dokazuje to, že měl útočník superuživatelská práva. V popisovaném případu bylo však původně napadnuté `uid=33 (www-data)`, webová služba.

Pomocí příkazu `,find / -user www-data‘` byl nalezen adresář, v němž si útočník nechal další nástroj:

```
 ---- cut ‘ls -lia /sbin/apache‘ ---
 total 19
 30230 drwxrwxrwx 2 root     root     1024 2007-01-14 10:54 .
 20098 drwxr-xr-x 3 root     root     5120 2007-01-15 16:21 ..
 30320 -rw-r--r-- 1 www-data www-data 1362 2007-01-13 18:27 pack.tgz
 30313 -rw-r--r-- 1 www-data www-data 1790 2007-01-14 18:25 passwd
 30370 -rwsrwxrwx 1 root     root     5295 2007-01-14 09:03
s~30369 -rw-r--r-- 1 www-data www-data 1460 2007-01-14 18:25 shadow
 30374 -rw-r--r-- 1 www-data www-data  722 2007-01-14 10:54 shadow.tgz
 ---- cut ‘ls -lia‘ ---
```

Je zde k vidění suid program `/sbin/apache/s`. Atribut suid je označení umožňující uživateli vykonat takový program s dočasně jinými právy, než jaká byla uživateli přidělena (viz uid vs. euid). Pomocí `strings` (vytažení tisknutelných řetězců z binárního souboru) je možné odhadnout jeho účel. Zkoumaný `/sbin/apache/s` je velmi krátký a dle získaného výpisu slouží nejpravděpodobněji ke spuštění jiného programu s právy roota, viz volání *setuid()*, *setgid()* a *system()*. Totéž prokáže i zpětný překlad (disassembling). Zbývá zjistit, jak příslušný program útočník vyrobil, čili je třeba najít důkaz, nebo alespoň nalézt nějakou lokálně zneužitelnou chybu zkoumaného systému. Těch existuje daleko

více než vzdálených a pátrání se v tomto případě zaměřilo na ostatní suid programy a vyhledání jejich publikovaných chyb:

```
---- cut ---
$ find . -type f -perm -04000 -ls
 20124  16 -r-sr-xr-x 1 root  root   15000 ^en 28  2004 ./root/sbin/unix_chkpwd
 30370   6 -rwsrwxrwx 1 root  root    5295 led 14 09:03 ./root/sbin/apache/s
 29829  36 -rwsr-xr-x 1 root  root   35512 srp 12 20:05 ./root/bin/login
...
990617 796 -rwsr-xr-x 1 root  root  809836 pro  8 23:29 ./usr/bin/gpg
...
$ find . -type f -perm -02000 -ls
...
---- cut ---
```

Dle výpisu mohlo jít tedy o zneužití jedné ze dvou chyb v GPG. Instalovaný SW je datovaný na 8.12.2006 a o den později vyšlo DSA-1231[7] oznámení o dvou chybách, které mohou vést ke spuštění podstrčeného kódu pomocí chyby přetečení zásobníku. Nicméně důkaz o zneužití této chyby nebyl nalezen a nebyl nalezen ani publikovaný exploit nebo jeho proof-of-concept.

Nakonec vše prozradil *process accounting* v podobě databáze programu `atop`. Z výpisu je možné vidět spouštění programu `getsuid` identitou www-data, program do systému nepatří a jeho název naznačuje vytvoření programu `/sbin/apache/s`.

```
---- ‘atop -r var/log/atop.log‘ ---
12556  www-data www-data  2007/01/08 15:36:59     --      S~0% apache2
22461  www-data www-data  2007/01/10 18:28:45     --      S~0% apache2
...
    ?  www-data www-data  2007/01/13 17:35:25     NE   0  E   0% <chmod>
    ?  www-data www-data  2007/01/13 17:35:38     NC  11  E   0% <getsuid>
    ?  www-data www-data  2007/01/13 17:35:38     NE   0  E   0% <getsuid>
    ?  www-data www-data  2007/01/13 17:34:06     NE   0  E   0% <id>
    ?  www-data www-data  2007/01/13 17:34:07     NE   0  E   0% <ls>
    ?  www-data www-data  2007/01/13 17:34:24     NE   0  E   0% <ls>
    ?  www-data www-data  2007/01/13 17:34:49     NE   0  E   0% <ls>
---- ‘atop -r var/log/atop.log‘ ---
```

Podle jména je pak jednoduché najít hotový exploit za pomoci vyhledávače:

- http://www.ykzj.org/article.php?articleid=2009
- http://www.securityfocus.com/bid/18874

Tento exploit využívá lokální chybu linuxového jádra v systémovém volání *prctl()* (process control). Chyba umožní vytvoření souboru `core`[8] i v adresáři, ve kterém nemá běžící program právo zápisu. Exploit tedy vytvoří dva procesy,

[7] Debian Security Advisory - http://www.debian.org/security/2006/dsa-1231

[8] coredump - obraz procesu po pádu určený ke zjištění příčiny chyby v debugeru

z nichž jeden nechá spadnout v adresáři `/etc/cron.d` tak, aby cron (běžící s rootovskými právy) dokázal interpretovat text v coredumpu. V padajícím programu je zakompilován řetězec:

```
 ---- cut getuid.c ---
char *payload="\nSHELL=/bin/sh\nPATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/
     usr/sbin:/usr/bin\n
 * * * * *   root   chown root.root /tmp/s ; chmod 4777 /tmp/s ;
     rm -f /etc/cron.d/core\n";
 ---- cut getuid.c ---
```

Program `/tmp/s` byl na forenzním duplikátu nalezen. Nakonec bylo potřeba pouze prokázat použití tohoto exploitu. To se povedlo nalezením smazaného coredumpu (je nalezen blok ve kterém je řetězec obsažen):

```
---- cut ---
sklad @ /dukazy/mistik# grep -iab "/etc/cron.d/core" mistik.hda1
29158161:* * * * *   root   chown root:root /tmp/s ; chmod 4777 /tmp/s ;
     rm -f /etc/cron.d/core
---- cut ---
```

Funkčnost exploitu byla úspěšně odzkoušena ve virtuálním pískovišti[9]. V této chvíli bylo prokázáno, že k získání superuživatelských práv byla zneužita lokální chyba jádra *Linux Kernel PRCTL Core Dump Handling Privilege Escalation Vulnerability* [10].

6 Závěr

K tomuto incidentu došlo díky nedostatečnému zabezpečení provozovaných služeb:

- anonymní přístup na FTP server s možností zápisu do adresáře publikovaného pomocí webové služby,
- nezabezpečené PHP,
- webový server neběžel v odděleném prostředí, které by útočníkovi ztěžovalo průnik do systému (jail/chroot),
- neaktualizované jádro operačního systému.

Stroj byl napaden již v listopadu roku 2006, tedy minimálně dva měsíce před odhalením. V této konfiguraci byl však provozován cca 3 roky, čili přesné datum průniku není možné přesně určit.

[9] testovacím prostředí VMware

K odhalení došlo víceméně náhodou (provozní problémy stroje podnítily hlubší průzkum), také k tomu velmi dopomohl fakt, že útočníci po sobě prakticky nezametali.

Služba IDS Snort, kterou v domovské síti provozujeme, přinesla jeden z prvních praktických výsledků a bude nadále rozšiřována, protože mohla přinést varování o průniku automaticky.

Troufám si tvrdit, že chybu jádra CVE-2006-2451 obsahuje velké procento současných produkčních strojů napříč celým internetem. Proti této a jiným podobným chybám jsou jedinou obranou včasná varování od systémů IDS, správná konfigurace a zabezpečení služeb tak, aby nedošlo k úvodnímu napadení.

Jako poznámku na závěr a zamyšlení, bych si dovolil podotknout, že vyhledávací služby jsou šavle broušené na mnoho stran...

Literatura a odkazy

[1] Kadlec, J. *Forenzní analýza unixových systémů.* http://jose.dump.cz/diploma.html

[2] Dittrich, D. *Basic Steps in Forensic Analysis of Unix Systems.* http://staff.washington.edu/dittrich/misc/forensics/

[3] Stover, S., Dickerson, M. *Using memory dumps in digital forensics.* ;login:, vol. 30, no. 6, pp. 43–48.

[4] Brunette, G. M. Jr. *Hiding within the trees.* ;login:, vol. 29, no. 1, pp. 48–54.

[5] Loza, B. *Under Attack.* ;login: 2005, vol. 30, num. 3.

[6] Loza, B. *Finding trojans for fun and profit.* ;login:, vol. 30, no. 5, pp. 19–22.

[7] Vyskočil, M. LiveCD a jejich použití, *Sborník příspěvků z XXVIII. konference EurOpen.CZ.* 2006.

[8] Snort. *Intrusion Detection system.* http://www.snort.org/

[9] BASE. *Basic Analysis and Security Engine.* http://sourceforge.net/projects/secureideas

[10] *Linux Kernel PRCTL Core Dump Handling Privilege Escalation Vulnerability.* http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2451 http://www.securityfocus.com/bid/18874

. . . už nechodí bosa

Michal Švamberg

E-mail: svamberg@civ.zcu.cz

Abstrakt

Žádný operační systém není bezpečný. Je pouze otázkou času, peněz a pohnutek, kdy bude prolomen. Nejčastěji jsou napadány koncové stanice, které mají často nízkou nebo žádnou ochranu. A to stále navzdory faktu, že je dostatek kvalitních programů, které umožňují zabezpečení na vysoké úrovni.

Zabezpečení koncových stanic, ale i serverů, se provádí většinou až po úspěšném útoku. Nejinak tomu bylo v tomto případě. Se seriózním zmapováním dostupných programů a jejich otestováním se začalo až s křížkem po funuse. Dále se dozvíte o možnostech zlepšení ochrany na linuxových operačních systémech a také získáte několik rad, poznatků a zkušeností z praxe.

Legislativní opatření

Pokud chcete zlepšit zabezpečení výpočetní techniky, vrhněte se napřed na to administrativně. Nařiďte, jaké služby a kde mají běžet, co je možné a zakázané mít na pracovních stanicích. Určete osobu, za kterou budou ostatní chodit v případě, že budou mít podezření na nějaké nekalosti.

Nezapomeňte udělat osvětu, co může taková nekalost znamenat. Nejčastěji lze zaznamenat problém na výkonu stanice, který rapidně poklesne. Uživatelé snížení výkonu začnou přisuzovat systému a raději si zvyknou, než aby někoho volali nebo se museli pustit k přeinstalování.

Napadený stroj je samozřejmě nejlepší přeinstalovat. V žádném jiném případě nezískáte jistotu, že jste se útočníka zbavili. Je to relativně tvrdé opatření, které by měl bezpečnostní technik vždy vyžadovat. K obnově systému lze využít zálohy, ale pozor na dobře zazálohovaného útočníka.

Dodržovaná legislativní opatření jsou daleko účinější než opatření technická, jsou více systémová a ve výsledku potřebují jen minimální údržbu. Bohužel jejich zavední nebývá přijato s největším nadšením.

Technická opatření

Programů, kterými lze zajistit na linuxových systémech vyšší bezpečnost nebo odhalit útok je nepřeberné množství. Přesto jsem se pokusil vybrat ty, které jsou běžně v distribucích a proto je jednodušší jejich nasazení.

Většinou každá distribuce má svůj návod, jak vhodně nastavit systém tak, aby byl bezpečný. Velmi dobrý je návod od Debianu[1] a také od Gentoo[2]. K prvnímu seznámení je také vhodná reference card z LinuxSecurity[3]. V dokumentačním projektu linuxu lze najít sice postarší dokument o bezpečnosti[4], přesto v něm lze najít lecos poučného.

Provozování většiny technických opatření je velmi časově náročné a tudíž nákladné. Je třeba správně určit nejzranitelnější části provozovaného systému a ty následně chránit vhodnými prostředky.

Existuje rozdělení do tří skupin, na *proaktivní*, *pasivní* a *reaktivní* ochranu. Proaktivní ochrana se snaží vyhledat a upozornit na části systému, které by mohly být zneužity. Pasivní ochrana je senzor, který na možný útok upozorní. Reaktivní ochrana navíc provede nějaká protiopatření, nejčastěji přenastavení firewallu.

Proaktivní nástroje doporučuji používat, protože stejný přístup k nim mají také útočníci a často je zneužívají ke hledání slabých míst. Při jejich použití se také dozvíte mnoho zajimavých věcí o svém stroji, které jste možná ani vědět nechtěli.

Na úrovni systému

Základním prvkem zabezpečeného systému je vzdálené logování a analýza logů. Pro sběr logů je nejvhodnější `syslog-ng`[5], a analýzu zajistí `logcheck`[6]. Nevýhodou jsou celkem časté emaily (několik denně) a také nikdy nekončící ladění pravidel analyzátoru.

Pro process accounting[7], tedy monitorování spuštěných příkazů a jejich návratových hodnot, je potřeba mít podporu v jádře. Při odhalování podezřelých aktivit se však bude hodit. Accounting umožňuje pouze ukládání spuštěných příkazů, nikoliv jejich parametrů. Jednoduchý process accounting bez nutnosti podpory jádra dokáže zajistit `atop`[8], avšak není zcela spolehlivý.

Velmi doporučovanou volbou je běh služeb v `chroot(8)`, který lze spravovat nástoji jako je `makejail`[9] nebo `jailer`[10]. Bohužel `chroot` je relativně snadno prolomitelný, pokud k němu není zapnuta dodatečná ochrana v jádře.

Modifikace jádra

Modifikací standardního linuxového jádra je možné značné zvýšení bezpečnosti. Tyto modifikace zavádějí nespustitelný zásobník a haldu, ošetřují chroot a do-

časné soubory nebo omezují pravomoce superuživatele a zavádějí přístupové seznamy, tzv. Access Control List. Při zavádění těchto modifikací lze očekávat komplikace, proto je napřed vyzkoušejte a seznamte se s nimi.

Nejznámnější a lze říci i nejobsáhlejší je projekt grsecurity[11], jenž obsahuje `pax` (PageExec) pro ochranu před spustitelností zásobníku a zároveň používá RBAC (Role-Based Access Control) a další drobné ochrany systému. Mimo jiné rozšiřuje ochranu chrootu, vybraných systémových volání a ochranu adresářů `/tmp` a `/proc`.

Projekt LIDS[13] se soustředí na ochranu souborů a procesů, implementaci ACL a zlepšení logovacích schopností jádra. Využívá modelu Mandatory Access Control (MAC), který přiřazuje objektům a subjektům bezpečnostní atribut. Na základě srovnání atributů pak bude povolen nebo zamítnut přístup.

SELinux[12] vychází z omezení práv uživatele na nezbytně nutné minimum. Lze říci, že implementuje model MAC. Tento projekt má nyní nejblíže k plné implementaci do jádra a asi nejlepší podporu v distribucích.

Projekt OpenWall[14] se soustředí okolo nespustitelného zásobníku a omezení pro `/proc` a `/tmp` adresáře. RSBAC[15] definuje pravidla a povolení pro různé operace v závislosti na uživatelských rolích.

Asi nejčastěji nasazovanými projekty jsou grsecurity a SELinux, jejich srovnání naleznete v dokumentu[17].

Souborový systém aneb najdi změnu

Na poli systémů detekce průniků (IDS), je několik možností jak hledat změny v souborovém systému. Vychází se z toho, že útočník vždy zanechá v souborovém systému stopy, proto je vhodné hledat změny. Na druhou stranu souborový systém je docela dynamická struktura.

Dříve velmi známý a oblíbený tripwire[19] (po hrátkách s licencemi existuje i open source varianta) lze dnes plně nahradit nástrojem AIDE[20]. Oba nástroje dělají to samé: vytvoří si databázi o souborech v systému a pak pravidelně porovnávají systém s databází. Na změny vás upozorní emailem.

Výhodou AIDE je licence a také jednodušší a přehlednější konfigurace na základě regulárních výrazů. Co do funkčnosti je s Tripwire shodný, v některých parametrech jej převyšuje.

Nevýhodou těchto systémů je, že vás zásobuje emaily, kterým je potřeba se věnovat a téměř vždy něco udělat. Dalším problémem je, kam bezpečně uložit databázi nebo její kontrolní součet, tak aby nemohla být útočníkem podvržena. Je zde také možnost, že bude podvržen i nástroj, kterým integritu souborového systému zjišťujeme.

Připojení k síti

Mezi základní bezpečnostní opatření patří firewall. V linuxovém jádře je implementován netfilter[18], obecně známý spíše pod jménem iptables. Firewall je dobrá ochrana, avšak nesmí být přeceňována.

Mezi nejznámější analyzátory síťového provozu patří Snort[21]. Snort ukládá do databáze informace o komunikacích, na které mu sedí hledané vzory. Snort se nehodí příliš pro odhalování útoku, ale spíše slouží jako pomocník k upřesnění nebo dohledání již dříve podezřelé činnosti.

Nástroj Nessus[22] obsahuje databázi (formou pluginů) o možných zranitelnostech. Po síti se pokouší zjistit, zda dané testy jsou úspěšné či nikoliv. Je to velmi silný nástroj, který vám prozradí věci, které jste ani nevěděli.

Jednoduchým a relativně funkčním opatřením je zpomalit opakované pokusy, tzv. útok hrubou silou. Například omezení maximálně čtyř pokusů za minutu o přihlášení k SSH může vypadat takto:

```
iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent --set
iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m recent \
         --update --seconds 60 --hitcount 4 -j DROP
```

Je vhodné neignorovat používání wrapperů. Wrappry se konfigurují soubory `/etc/hosts.allow` a `/etc/hosts.deny`. Zde lze připsat zajimavé kódy, které vám umožní například lepší monitoring útoků hrubou silou nebo získávat informace o odmítnutých pokusech o připojení[23]:

```
# /etc/hosts.deny
ALL: ALL: SPAWN ( \
          echo -e "\n\
          TCP Wrappers\: Connection refused\n\
          By\: $(uname -n)\n\
          Process\: %d (pid %p)\n\
          User\: %u\n\
          Host\: %c\n\
          Date\: $(date)\n\
        " | /usr/bin/mail -s "Connection to %d blocked" root) &
```

Drobní pomocníci

Tiger[24] je sada skriptů pouštěných z cronu, které se snaží monitorovat systém a hlásit změny, které mohou vést k bezpečnostním problémům. Dokáže hlídat skryté soubory, setuid, síťová spojení, kontrolní součty softwarových balíků a mnoho dalšího. Jednotlivé distribuce mají nespočet malých nástrojů pro zajištění lokální bezpečnosti, Tiger se snaží tyto nástroje nahradit a jejich výhody spojit do jednoho nástroje.

Program upozorní na problém pouze jednou, čímž významně snižuje počet emailů. Pokud se problém vyřešil, oznámí to opět emailem. Velmi doporučuji nasadit, dozvíte se věci, které jste o svém systému netušili.

Ukázka výpisu tigera (`NEW` – nově detekované problémy, `OLD` – vyřešené):

```
# Checking listening processes
NEW: --WARN-- [lin003w] The process ‘portmap‘ is listening on socket 111
              (TCP on every interface) is run by daemon.
NEW: --WARN-- [lin003w] The process ‘portmap‘ is listening on socket 111
              (UDP on every interface) is run by daemon.
OLD: --WARN-- [lin003w] The process ‘vos‘ is listening on socket 33503
              (UDP on every interface) is run by xuser.
```

Nástroj rkhunter[25] si pravidelně stahuje databázi o rootkitech a snaží se je nalézt v systému. Zároveň provádí testy na nejčastěji zneužívané bezpečnostní problémy. Nalezené problémy opět hlásí emailem. Program lze doporučit, je jednoduchý, rozumně konfigurovatelný a upozorní vás na ty největší problémy. Ukázka výpisu rkhunteru:

```
Scanning for hidden files...  [ Warning! ]
Checking for allowed root login... Watch out Root login possible. Possible risk!
-------------------------------------------------------------------


Found warnings:
[06:26:50] WARNING, found:  /dev/.udev (directory)
[06:26:52] Warning: root login possible. Change for your safety the
           ‘ PermitRootLogin‘

-------------------------------------------------------------------
```

Jako náhradu za rkhunter lze použít chkrootkit[26], je ale méně konfigurovatelný a neudržuje si sám aktuální databázi rootkit vzorů a nedělá tolik práce jako rkhunter. Je vhodné mít alespoň jeden z těchto nástrojů nasazený. Ukázka výpisu chkrootkit:

```
/etc/cron.daily/chkrootkit:
The following suspicious files and directories were found:
/usr/lib/iceape/.autoreg
/usr/lib/iceweasel/.autoreg
/usr/lib/epiphany/2.14/extensions/.pyversion
/usr/lib/xulrunner/.autoreg
/lib/init/rw/.ramfs
```

Jeden příklad za všechny

Po zjištění závažného průlomu do jednoho stroje (viz příspěvek *Kovářova kobyla...*) jsme se zabývali zlepšením bezpečnosti. Přibližně po měsíci přišel další útok, na který upozornil logcheck, obsahoval celkem 8 896 záznamů:

```
Feb 20 20:18:02 147.228.53.16 cron[17241]: Error: bad minute; while
reading /etc/cron.d/core
```

O kousek výše nad tímto výpisem uvidíte další podezřelé záznamy:

```
Feb 20 20:16:52 147.228.53.16 crontab[19467]: (maara) LIST (maara)
Feb 20 20:16:55 147.228.53.16 crontab[16419]: (maara) BEGIN EDIT (maara)
Feb 20 20:16:57 147.228.53.16 crontab[16419]: (maara) END EDIT (maara)
Feb 20 20:16:58 147.228.53.16 crontab[216]: (maara) LIST (maara)
Feb 20 20:17:05 147.228.53.16 kernel: grsec: From 77.48.19.29: signal 11 sent to
/home/maara/expl[expl:18408] uid/euid:23190/23190 gid/egid:100/100,
parent /home/maara/expl[expl:21600]
uid/euid:23190/23190 gid/egid:100/100 by /home/maara/expl[expl:21600]
uid/euid:23190/23190 gid/egid:100/100,
parent /bin/tcsh[tcsh:17331] uid/euid:23190/23190 gid/egid:100/100
```

Ty mohly být způsobeny regulérní činností, přesto se vyplatí na uživatele maara si trochu posvítit. A samozřejmě nás to nutí podívat se do kompletního logu. Ovšem nikoliv na lokální stroj, ale na vzdálený logovací serveru. Útočníkovi se podařilo inkriminovanou část z lokálního logu odstranit. V `kern.log` si lze najít již známý záznam z checklogu:

```
kern.log:Feb 20 20:11:36 147.228.53.16 kernel: PAX: bytes at
  PC: e8 00 00 00 00 b8 17 00 00 00 31 db cd 80 58 bb 3d 00 00 00
kern.log:Feb 20 20:11:36 147.228.53.16 kernel: PAX: bytes at
  SP-4: 00000000 00000003 bffffea4 bffffeb4 bffffebb bffffec2 bffffec9 00000000
        00000000 2e000000 7078652f
        2f2e006c 6c707865 652f2e00 006c7078 6f72702f 32312f63 2f393132 69766e65
        006e6f72 00000000
kern.log:Feb 20 20:17:05 147.228.53.16 kernel: grsec: From 77.48.19.29: signal 11
sent to /home/maara/expl[expl:18408] uid/euid:23190/23190 gid/egid:100/100,
parent /home/maara/expl[expl:21600]
uid/euid:23190/23190 gid/egid:100/100 by /home/maara/expl[expl:21600]
uid/euid:23190/23190 gid/egid:100/100,
parent /bin/tcsh[tcsh:17331] uid/euid:23190/23190 gid/egid:100/100
```

V logu `syslog` najdeme ještě podezřelejší a zajimavější řádky, a hlavně příkaz, který `cron` zavolal:

```
syslog:Feb 20 20:18:02 147.228.53.16 cron[17241]: Error: bad minute;
while reading /etc/cron.d/core
syslog:Feb 20 20:18:02 147.228.53.16 cron[17241]: Error: bad time specifier;
while reading /etc/cron.d/core
syslog:Feb 20 20:18:02 147.228.53.16 /USR/SBIN/CRON[21977]: (root) CMD
  (cp /bin/sh /tmp/sh; chown root /tmp/sh; chmod 4755 /tmp/sh;
rm -f /etc/cron.d/core)
```

Utvrdit se o identitě útočníka můžeme přes accounting příkazem `lastcomm` (číst od spodu):

```
expl            F DX  maara    ??         0.00 secs Tue Feb 20 20:17
cron            F     root     ??         0.00 secs Tue Feb 20 20:17
crontab               maara    stderr     0.00 secs Tue Feb 20 20:16
ls                    maara    stderr     0.00 secs Tue Feb 20 20:16
gcc                   maara    stderr     0.00 secs Tue Feb 20 20:16
collect2              maara    stderr     0.02 secs Tue Feb 20 20:16
ld                    maara    stderr     0.03 secs Tue Feb 20 20:16
as                    maara    stderr     0.00 secs Tue Feb 20 20:16
cc1                   maara    stderr     0.05 secs Tue Feb 20 20:16
wget                  maara    stderr     0.01 secs Tue Feb 20 20:16
```

Příznaky u spuštěného příkazu `expl` jsou:

D – program byl ukončen a vygeneroval core dump soubor

X – program byl ukončen signálem SIGTERM

F – program spuštěn forkem bez použití funkce `exec()`.

Posledním krokem bylo pozvání si uživatele na kobereček, kde se bez protestů přiznal.

Reakce na tento útok byla do dvou hodin (prodleva logchecku a čtení emailu). Přibližně další dvě hodiny trvalo dohledání důkazů ve dvou lidech, následně byl stroj odstaven a přeinstalován. Zde měl útočník významně zjednodušenou práci, protože již na stroji vlastnil lokální účet. V logových souborech bylo možné dohledat jiné neúspěšné pokusy o útok, kterým zabránil grsecurity, bohužel na chybu v jádře byl krátký.

Závěr

Tento příspěvek si nedával za cíl dokonale zabezpečení stroje, ale poukázat na zajimavé a možná i neznámé programy. Tyto programy mohou jistě zlepšit zabezpečení vašich počítačů, avšak před potencionálním útočníkem je potřeba se mít stále na pozoru.

Odkazy

[1] Securing Debian Manual
http://www.us.debian.org/doc/user-manuals#securing

[2] Gentoo Security Handbook
http://www.gentoo.org/doc/en/security/

[3] Linux Security Quick Reference Guide
http://www.linuxsecurity.com/docs/QuickRefCard.pdf

[4] Security Quick-Start HOWTO for Linux
http://www.tldp.org/HOWTO/Security-Quickstart-HOWTO/

[5] Domovská stránka projektu `syslog-ng`
http://www.balabit.com/products/syslog_ng/

[6] Domovská stránka analyzátoru `logcheck`
http://logcheck.org/

[7] The GNU Accounting Utilities
http://www.gnu.org/software/acct/

[8] ATOP System & Process Monitor
http://www.atconsultancy.nl/atop/

[9] Domovská stránka nástroje `makejail`
http://www.floc.net/makejail/

[10] Domovská stránka nástroje `jailer`
http://www.balabit.com/downloads/jailer/

[11] Modifikace jádra: grsecurity
http://www.grsecurity.net/

[12] Modifikace jádra: Security-Enhanced Linux
http://www.nsa.gov/selinux/

[13] Modifikace jádra: LIDS – Linux Intrusion Detection System
http://www.lids.org/

[14] Modifikace jádra: OpenWall
http://www.openwall.com

[15] Modifikace jádra: RSBAC – Rule Set Based Access Control
http://www.rsbac.org/

[16] Dobšíček, M., Ballner, R.: *Linux – bezpečnost a exploity.* České Budějovice : Nakladatelství KOOP, 2004. ISBN 80-7232-243-5.

[17] SELinux and grsecurity: A Case Study Comparing Linux Security Kernel Enhancements
`www.cs.virginia.edu/~jcg8f/GrsecuritySELinuxCaseStudy.pdf`

[18] The netfilter.org project
http://www.netfilter.org/

[19] Domovská stránka tripwire
http://www.tripwire.com/products/enterprise/ost/

[20] AIDE – Advanced Intrusion Detection Environment
http://sourceforge.net/projects/aide

[21] Domovská stránka projektu Snort
http://www.snort.org/

[22] Domovská stránka projektu Nessus
http://www.nessus.org/

[23] Using TCP wrappers to secure Linux
http://linuxhelp.blogspot.com/2005/10/ \
using-tcp-wrappers-to-secure-linux.html

[24] Tiger – The Unix security audit and intrusion detection tool
http://www.nongnu.org/tiger/

[25] rkhunter – The Rootkit Hunter project
http://rkhunter.sourceforge.net/

[26] chkrootkit – locally checks for signs of a rootkit
http://www.chkrootkit.org/

Curiculum vitae

Jakub Balada – Jakub.Balada@siemens.com
Siemens IT Solutions and Services
Jakub Balada je studentem 5. ročníku MFF UK, oboru softwarového inženýrství. Ve společnosti Siemens patří do skupiny IT bezpečnosti, kde se v současné době věnuje Identity managementu. Dále se zabývá problematikou PKI čipových karet, hlavně jejich využitím v elektronickém bankovnictví.

Aljoša Jerman Blažič, M.Sc. – aljosa@setcce.si
SETCCE
Aljoša Jerman Blažič is the head of SETCCE, a research and development company in the field of electronic business and information security. He has graduated of Telecommunication science at the Faculty of Electrical Engineering, University of Ljubljana and obtained his Masters degree at the Faculty of Economics, University of Ljubljana. He is a Ph.D. candidate at the same educational institution, preparing a thesis on the topic of Trusted Archiving Services.

His past work was performed as a researcher for Laboratory for Open Systems and Networks at Jozef Stefan Institute on security systems, broadband and mobile communications mainly for European research projects. He moved to applied research and development projects. His current work is performed in the field of research, design and development of advanced electronic business platforms, formal electronic documents, preservation systems for electronic records, security and privacy mechanisms and ambiental intelligence. He is an active contributor to standardization bodies (IETF, ETSI, GZS, ...) and author of technology standards with special focus on formal electronic documents and preservation mechanisms.

Radoslav Bodó – bodik@civ.zcu.cz
CIV ZČU
Absolvent Fakulty Aplikovaných Věd Západočeské univerzity v Plzni v oboru Distribuované systémy. Od roku 2005 pracuje v oddělení Laboratoře počítačových systémů, Centra informatizace a výpočetní techniky jako správce operačních systémů Linux a distribuovaného výpočetního prostředí Orion, se specializací na oblast bezpečnosti IS a služeb na platformě Java.

Jiří Bořík – borik@civ.zcu.cz
CIV ZČU
Ing. Jiří Bořík graduated in 1992 at the University of West Bohemia specializing in Electronic computers. Since 2004, he works with the Centre for Information Technology at the University of West Bohemia. As a member of the Laboratory for Computer Science, he participates in developing the ORION Computing Environment focusing, among others, on identity management solutions.

PaedDr. Bohumír Brom – BOHUMIR.BROM@NACR.CZ

Absolvent VŠ studia historie se zaměřením na učitelství. V letech 1980–1990 působil jako pedagog. Od roku 1990 pracuje v Národním archivu (dříve Státním ústředním archivu) jako archivář, a to v oblasti předarchivní péče. Dlouhodobě se specializuje na obory průmysl, obchod a finance, zejména pokud jde o ústřední instituce státní správy a jejich podřízené organizace.

Petr Břehovský – BREH@BREH.CZ

Ing. Petr Břehovský (*1965) Vystudoval obor Operační systémy a sítě na katedře výpočetní techniky Petrohradského institutu jemné mechaniky a optiky. Pracoval jako správce operačních systémů UN*X a TCP/IP sítí. Zabývá se lektorskou činností v oblasti protokolů TCP/IP, os UN*X a bezpečnosti výpočetních systémů. Širší veřejnosti je znám spoluprací na překladech knih Hacking bez tajemství a Počítačový útok, detekce, obrana a okamžitá náprava. V současné době pracuje v oddělení bezpečnosti informačních technologií firmy Telefónica O2 Czech Republic.

Martin Čízek – MARTIN@CIZEK.COM

Autor se zabývá problematikou integrace a návrhu heterogenních systémů, vývojem lightweight J2EE aplikací, školením technologií a open source produktů. V současné době pracuje jako team leader ve společnosti Itonis, s. r. o., vyvíjející platformu pro multimediální služby dodávané prostřednictvím IP sítí.

Libor Dostálek – LIBOR.DOSTALEK@SIEMENS.COM
Siemens IT Services and Solutions
RNDr. Libor Dostálek (*1957) je vedoucím oddělení IT Security Consulting společnosti Siemens IT Solutions and Services. Podílel se na projektech se zaměřením na poskytování Internetu, elektronické bankovnictví, bezpečnost sítí a IT bezpečnost. Je autorem Velkých průvodců:

- *Velký průvodce protokoly TCP/IP a DNS* (anglicky vyšlo jako dvě publikace: „Understanding TCP/IP“ a „DNS in action“, rusky vyšlo jako „TCP/IP и DNS в теории и на практике. Полное руководство“.
- *Velký průvodce protokoly TCP/IP, bezpečnost* (vyšlo též polsky pod názvem „Bezpieczeństwo protokołu TCP/IP“)
- *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu.*

Michal Hojsík – Michal.Hojsik@siemens.com
Siemens IT Solutions and Services
Mgr. Michal Hojsík (*1982) vyštudoval obor matematické metódy informačnej bezpečnosti na matematicko-fyzikálnej fakulte Univerzity Karlovej v Prahe, kde taktiež pokračuje na doktorskom štúdiu.

V súčasnosti pracuje ako konzultant na bezpečnostnom oddelení firmy Siemens.

Zaujíma sa predovšetkým o symetrickú kryptografiu a kryptoanalýzu.

Ralf Knöringer – ralf.knoeringer@siemens.com
Siemens AG
Mr. Knöringer has more than 15 years practical experience within the international security market place. His special focus areas are Identity & Access Management, Provisioning & Directory solutions.

With his comprehensive knowledge of the Security market place, technology trends and international customer projects Mr. Knöringer is the management contact for Siemens key customers and strategic partners interested in secure user management, entitlement and access management solutions.

Ladislav Lhotka – lhotka@cesnet.cz
CESNET, z. s. p. o.
Ladislav Lhotka (*1959) absolvoval v r. 1983 matematické inženýrství na FJFI ČVUT a v r. 1992 ukončil vědeckou aspiranturu v ÚTIA AV ČR v oboru Technická kybernetika. V první dekádě své profesionální kariéry se věnoval matematickému a simulačnímu modelování ekologických systémů. Po příchodu Internetu do ČR se zapojil do budování akademických sítí a to se mu postupně stalo hlavním zaměstnáním. Od r. 2001 pracuje ve sdružení CESNET, kde v současné době vede aktivitu Programovatelný hardware a podílí se též na spolupráci v rámci mezinárodního projektu GÉANT2. K jeho odborným zájmům patří kromě síťových technologií ještě operační systém Linux, programování v Pythonu a systémy pro zpracování textu (XML, TEX).

Jan Okrouhlý – okrouhly@kp.cz
Hewlett-Packard, s. r. o.
Ing. Jan Okrouhlý (*1972) vystudoval Katedru informatiky a výpočetní techniky Fakulty aplikovaných věd Západočeské Univerzity v Plzni. Po vystudování pracoval v tamním Centru informatiky a výpočetní techniky v Laboratoři počítačových systémů na rozličných IT projektech. Od roku 2005 je zaměstnán jako technologický konzultant v Hewlett-Packard Praha, kde se podílí na vydávání čipových karet.

Martin Pavlis – martin@pavlis.net
Microsoft
Martin Pavlis, MCT, MCSE+S+E, MCSA+S+E, MCP.
Vedení autorizovaných kurzů firmy Microsoft, účast na projektech návrhu a implementace podnikových řešení na platformě Microsoft, realizace odborných seminářů a konferencí. Se zaměřením na tyto technologie: Windows Cluster, Windows Server Systems, Echange Server, ISA Server a další.

Peter Sylvester – Peter.Sylvester@edelweb.fr

More than 30 years of experience in network, system and application security, studies and implementation od security solution for various environments and networks. Co-founder of EdelWeb, a French IT-security company since 1995, after work as IT engineer at INRIA France and GMD Bonn, Active participation of standardisation and open source development, specification, architecture, development et maintenance of software and proof of concept implementations Development of multinational computer networks, participation in 5 EU IST programs. Diplome in Mathematics from University Bonn.

Michal Švamberg – svamberg@civ.zcu.cz
CIV ZČU
Vystudoval obor Distribuované systémy na Západočeské univerzitě v Plzni. Dlouhodobě se věnuje správě operačního systému Linux a jeho nasazení v distribuovaném prostředí Západočeské univerzity v Plzni. Mezi další oblasti patří správa virtuálních serverů Xen, diskového subsystému či distribuovaného souborového systému AFS.

Marta Vohnoutová – Marta.Vohnoutova@siemens.com
Siemens IT Solutions and Services, s. r. o.
Měla na starosti projekty bezpečného připojení k Internetu a budování a monitorování bezdrátových sítí. V PVT, a. s. byla zaměstnána jako Senior konzultant. V současné době pracuje v Siemens IT Solutions and Services, s. r. o., jako Solution architekt. Je spoluautorkou nově vyšlé knihy *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu.* Je držitelem MCSE a MCT pro W2K.