SSL protokol

SSL protokol (1) - princip a přínosy

Internet dnes využívají miliony uživatelů a používají jej i pro obchod nebo jiné aplikace, u kterých by mělo být zajištěno soukromí. Protokol SSL zajišťuje toto soukromí a spolehlivost pro komunikující aplikace, chrání data před odposloucháváním, zfalšováním a paděláním.

Internet dnes využívají miliony uživatelů a používají jej i pro obchod nebo jiné aplikace, u kterých by mělo být zajištěno soukromí. Protokol SSL zajišťuje toto soukromí a spolehlivost pro komunikující aplikace, chrání data před odposloucháváním, zfalšováním a paděláním.

Potřeba bezpečného přenosu

V minulých letech došlo k významnému nárůstu počítačových komunikací. Internet dnes využívají miliony uživatelů a nepoužívají jej pouze pro stahování informací, ale i pro obchod nebo pro jiné aplikace, u kterých by mělo být zajištěno soukromí. To samozřejmě přináší bezpečnostní problémy, protože přirozenou vlastností Internetu je to, že prakticky každý může přečíst vše co je přes Internet přenášeno. To dělá obchodní transakce nebezpečnými, neboť si kdokoliv může zjistit informace typu detaily o kreditní kartě.

Potenciální úskalí při nákupu přes Internet jsou tyto:

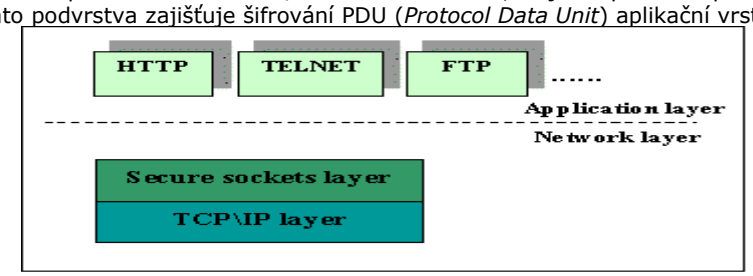
- jak může klient bezpečně předat informace o své platební kartě?
- jak si může být server jist pravostí klienta?
- jak si klient může být jist serverem?

Tyto problémy, které bránily masivnímu komerčnímu využití Internetu, řeší technologie, navržená společností Netscape ve spolupráci s několika dalšími významnými výrobci. Jde o protokol nazývaný SSL, zajišťující zabezpečenou komunikaci v prostředí Internetu.

Princip protokolu SSL

Protokol SSL zajišťuje soukromí a spolehlivost pro komunikující aplikace, chrání data před odposloucháváním, zfalšováním a paděláním.

Na následujícím obrázku je pozice protokolu SSL v TPC/IP modelu. Je vidět, že jde o přidanou podvrstvu mezi TPC/IP a aplikací. Tato podvrstva zajišťuje šifrování PDU (*Protocol Data Unit*) aplikační vrstvy.



Obr. 1 - Pozice protokolu SSL v TPC/IP modelu

SSL je protokol dělený na vrstvy. Dvě jsou hlavní:

- SSL Handshake Protocol
- SSL Record Protocol

SSL Record Protocol je zodpovědný za enkapsulaci (zabalení) dat protokolů vyšší vrstvy (např. HTTP, Telnet, FTP..., ale i ostatní části protokolu SSL).

SSL Handshake Protocol je zodpovědný za vytvoření bezpečné komunikace mezi klientem a serverem. To je dáno na základě ověření a odsouhlasení šifrovacího algoritmu a klíčů.

Hlavní přínosy SSL

Bezpečnost šifrování - primárním přínosem protokolu SSL je ustavení bezpečného spojení mezi dvěma komunikujícími uzly. Poté co jsou iniciačním algoritmem vyměněny bezpečné klíče, je používáno symetrické šifrování.

Spolehlivost - přenos zprávy obsahuje kontrolu integrity dat prostřednictvím entity nazývané MAC (Message Authentication Code).

Interoperabilita - různé aplikace různých programátorů by měly být schopny úspěšné výměny parametrů bez znalosti kódu aplikace druhé strany.

Rozšiřitelnost - struktura SSL umožňuje implementaci nových metod šifrování a výměny veřejných klíčů.

Relativní efektivita - šifrovací operace jsou dost náročné na vytížení procesoru; SSL se snaží tuto zátěž kompenzovat přídavnými funkcemi jako je např. komprimace dat nebo kešování spojení (umožní omezení počtu spojení iniciovaných vždy od začátku).

Základní slovníček z kryptografie (1)

Jako nedílnou součást našeho nového seriálu o zajištění bezpečného přenosu dat po Internetu pomocí protokolu SSL uvádíme i tento základní slovníček z oboru kryptografie.

Jako nedílnou součást našeho nového seriálu o zajištění bezpečného přenosu dat po Internetu pomocí protokolu SSL uvádíme i tento základní slovníček z oboru kryptografie.

Šifrování a dešifrování (Encryption & Decryption)

Šifrování je proces, který z bezpečnostních důvodů kóduje srozumitelnou informaci do nečitelné formy.

Dešifrování je opačný proces – kódovaná informace je převedena do srozumitelné formy.

Šifrovací klíč (Cryptographic Key)

Šifrovací klíč je sada instrukcí, které řídí postup šifrovacího nebo dešifrovacího algoritmu. Obvykle jsou šifrovací a dešifrovací algoritmy obecně známé a šifrovacím klíčem je řetězec znaků, který je uchováván jako „sdílené tajemství" a dělá komunikaci bezpečnou.

Symetrická kryptografie (Symmetric Cryptography)

Pojmem symetrická kryptografie jsou označovány algoritmy, které používají pro šifrování i dešifrování stejný klíč. To znamená, že pro symetrickou kryptografii je používán pro oba účely pouze jeden klíč. Příkladem algoritmů tohoto typu jsou DES (a jeho modifikace jako je 3DES), RC4 nebo IDEA.

Asymetrická kryptografie (Asymmetric Cryptography)

Pojmem asymetrická kryptografie jsou označovány algoritmy, které používají pro šifrování a dešifrování různé algoritmy. To znamená, že potřebuje dva různé klíče. Jeden pro šifrování a druhý pro dešifrování. Klíče jsou označovány jako veřejný a

privátní. Veřejný klíč je poskytnut protější straně, ta s jeho pomocí zašifruje data a přijímající strana je dešifruje klíčem jiným – privátním. Nejpoužívanějším současným algoritmem tohoto typu je RSA.

Hešovací funkce (Hash Function)

Hešování je vytvoření kontrolní informace. V podstatě jde o vytvoření jakéhosi razítka, které prokazuje pravost obsahu.

Jednocestná hešovací funkce (One-way hash function)

Jde o algoritmus, který z textu zprávy vytvoří řetězec pevné délky, nazývaný mesage digest (výtah ze zprávy). Ten je obvykle používaný z důvodu bezpečnosti nebo řízení datového toku. Jednocestnost znamená to, že z řetězce nelze získat zpět původní informaci. Algoritmus je používán pro vytvoření digitálního podpisu.

Příjemce si může porovnat připojený řetězec s řetězcem, který je obsažen v zašifrované části. Lze tím prověřit, že zpráva nebyla zfalšována. Proces je označován jako „hashcheck".

MD5

MD5 je jednocestná hešovací funkce vytvořená profesorem Ronaldem L. Rivestem v roce 1991.

Algoritmus MD5 vytváří z libovolně dlouhé vstupu 128 bitů dlouhý mesage digest (výtah). Je odhadováno, že je prakticky nemožné vytvořit dvě zprávy, které by měly stejný message digest nebo vytvořit zpětně zprávu ze známého digestu. Smyslem algoritmu MD5 je vytvoření elektronického podpisu (ověření integrity dat).

SHA / SHA-1

SHA (Secure Hash Algorithm) je jednocestná hešovací funkce vyvinutá institucí National Institute of Standards and Technology (NIST). Podrobnosti lze zjistit na odkazu http://www.nist.gov .

Algoritmus vytváří ze zprávy 160bit dlouhý řetězec nazývaný message digest. Ten je používán jako digitální podpis zprávy a slouží zejména pro ověření její pravosti. příjemce provede pro přijatou zprávu stejný výpočet a nesouhlasí-li doručený message digest s tím, který příjemce spočítal, zpráva byla cestou upravena.

Digitální podpis (Digital Signature)

Digitální podpis je metoda ověření totožnosti odesílatele elektronické zprávy. Jde o digitální kód, který je připojen ke zprávě. Jako v případě klasického podpisu, digitální podpis garantuje to, že odesílatel je tím, za koho se vydává. Význam elektronického podpisu je zejména pro elektronický obchod. Aby bylo jeho využití efektivní, musí být zajištěna vysoká odolnost proti padělání (falšování).

SSL protokol (2) - používané šifry, spojení a jeho struktura

SSL umožňuje použití různých šifrovacích technologií. Pro šifrování dat jsou používané symetrické metody jako např. DES, RC4 a Triple DES. Asymetrické metody s veřejnými klíči jako jsou RSA, DSS nebo KEA jsou používány pro ověřování komunikujících stran a pro přenos symetrických klíčů, použitých v dalším procesu komunikace. Dále si ujasníme dva pojmy – connection a session.

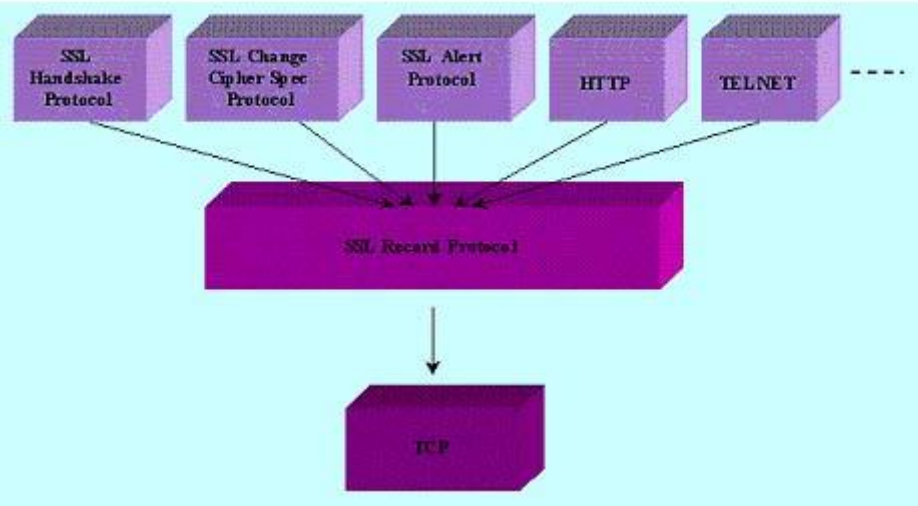
Šifry používané protokolem SSL

SSL umožňuje použití různých šifrovacích technologií. Pro šifrování dat jsou používané symetrické metody jako např. DES, RC4 a Triple DES. Asymetrické metody s veřejnými klíči jako jsou RSA, DSS nebo KEA jsou používány pro ověřování komunikujících stran a pro přenos symetrických klíčů, použitých v dalším procesu komunikace.

Protokol SSL je v hierarchickém modelu používán nad spolehlivým protokolem transportní vrstvy jako je TCP a pod aplikační vrstvou. Nicméně použití SSL není nutně vázáno na TCP/IP prostředí a může fungovat i s jinými protokoly.

Kromě dvou již zmíněných komponent (vrstev), jimiž jsou **SSL Handshake Protocol** a **SSL Record Protocol**, používá SSL další dvě komponenty – **SSL Change Cipher Spec Protocol** a **SSL Alert Protocol**. Handshake, Change Cipher Spec a Alert jsou protokoly používané pro správu SSL komunikace, její navázání a nastavení parametrů zabezpečení.

SSL Record Protocol je nejnižší úroveň protokolové architektury SSL. Ostatní zmíněné části, včetně aplikačních dat běží nad ním.



Obr. 2 - SSL Protocol Stack

Spojení a jeho struktura

Na začátek je potřeba ujasnit si dva pojmy – connection a session. Protože se mi nepodařilo najít významově jednoznačné české ekvivalenty, nebudu tyto pojmy překládat. Ze stejného důvodu vynechám překlad některých pojmů u nichž se domnívám, že by je překlad spíš zamlžil než věci pomohl. Pokusím se zde vyjádřit jaký je mezi session a connection rozdíl.

- Session je ustavení spojení mezi dvěma uzly – tedy jakýsi tunel; v další části bude používán převážně originální pojem, v některých větách použiji význam cesta i když není úplně dokonale výstižný.
- Connection je spojení mezi procesy probíhající v rámci session; i pro tento výraz bude v dalším textu používán originál, pokud bude použit český ekvivalent, bude jím výraz spojení.

Uveďme si ještě jeden pojem, který bude uváděn "počeštěně". Výraz hash bude v dalším textu uváděn jako hešování nebo hešovací funkce. Hešování je vytvoření kontrolní*informace.*

SSL protokol má dva důležité procesy komunikace:

- SSL session

- SSL connection

Struktura session používá následující parametry:

- Session ID – libovolná hodnota identifikující session;
- Peer certificate – X.509.v3 certifikát;
- Kompresní metoda – definice kompresního algoritmu;
- Specifické údaje šifrování – definice šifrovacího a MAC algoritmu použitého pro přenos dat;
- Master secret – heslo použité pro šifrování datového přenosu;
- Příznak nesoucí informaci zda session může být použita pro další connection.

Struktura connection používá následující parametry:

- náhodné číslo generované serverem i klientem;
- SERVER-MAC-WRITE-SECRET;
- CLIENT-MAC-WRITE-SECRET;
- SERVER-WRITE-KEY;
- CLIENT-WRITE-KEY;
- Initialization vectors;
- Sequence number.

Každý connection je svázán pouze s jedním session, ale jedna session může zahrnovat několik různých connection. Strukturu connection definují parametry ověřování (MAC), zatímco strukturu session definují parametry šifrování.
Pro zopakování, toho co bylo uvedeno dříve – MAC je zkratkou Message Authentication Code (ověřovací kód zpráv).

Základní slovníček z kryptografie (2)

Druhá část našeho základního slovníčeku z oboru kryptografie, kterou uvádíme jako nedílnou součást nového seriálu o zajištění bezpečného přenosu dat po Internetu pomocí protokolu SSL.

Druhá část našeho základního slovníčeku z oboru kryptografie, kterou uvádíme jako nedílnou součást nového seriálu o zajištění bezpečného přenosu dat po Internetu pomocí protokolu SSL.

MAC

Zkratka MAC při použití v kryptografii znamená Message Authentication Code, tedy kód pro ověření zprávy. Jde o kousek dat, který je vypočítán hešovací funkcí a před odesláním zprávy připojen na její konec. Používá se pro ověření pravosti obsahu zprávy. Oba účastníci komunikace si odsouhlasí parametry a typ hešovací funkce nebo alespoň tajný klíč pro její použití. Pokaždé když některý z nich přijme zprávu, spočítá její MAC a hodnotu porovná s hodnotou MAC, která je ve zprávě. Rozdíl znamená, že zpráva byla upravena.

Kryptografie s veřejným klíčem (Public Key Cryptography)

Kryptografie s veřejným klíčem je typem asymetrické kryptografie používaná na Internetu. Metoda používá dva klíče – jeden veřejný (veřejný klíč) a druhý utajený (privátní klíč). Každá zpráva zašifrovaná veřejným klíčem může být rozšifrována pouze privátním klíčem, každá zpráva zašifrovaná privátním klíčem, může být dešifrována pouze odpovídajícím veřejným klíčem. Uživatel nemůže pouze prostě vytvořit svoji sadu klíčů. K tomu jsou používány internetové úřady (certifikační autority), které vydávají uživatelům certifikáty obsahující sadu legitimních privátních a veřejných klíčů. Uživatel, který zamýšlí použít své vlastní klíče na Internetu, se musí prokázat certifikátem aby potvrdil svoji identitu a platnost klíčů. Význam kryptografie s veřejným klíčem je především v tom, že není nutné aby se účastníci spojení předem domlouvali na utajeném klíči. Ten kdo chce poslat protějšku nějakou zašifrovanou zprávu, zašifruje tuto zprávu jednoduše pomocí jeho veřejného klíče. Pouze oprávněný příjemce je schopen zprávu rozšifrovat svým privátním klíčem – nikdo jiný totiž není zprávu schopen dešifrovat.
Kryptografie s veřejným klíčem je použitelná pro digitální podpisy. Odesílatel, který chce podepsat svoji zprávu ji jednoduše zašifruje svým privátním klíčem. Kdokoliv tuto zprávu přijme, bude ji schopen dešifrovat pouze autorovým veřejným klíčem – tím se prověří odesílatelova totožnost.

DES

Data Encryption Standard (DES) je symetrický šifrovací algoritmus vyvinutý NSA. Je založen na matematické permutaci 56 bitovým klíčem. Dešifrování je prováděno inverzní funkcí se stejným klíčem. Do nedávné doby byla tato metoda považována za bezpečnou. S nárůstem výpočetní kapacity běžně dostupných počítačů je dnes délka klíče 56 bitů nedostatečná a je doporučováno použití 3DES klíčů.

RSA

Šifrovací metoda RSA patří do skupiny asymetrické kryptografie s veřejným klíčem a je dnes používána v mnoha implentacích včetně SSL. Metoda je vhodná jak pro šifrování zpráv, tak pro digitální podpis.
Zkratka RSA vychází ze jmen badatelů, kteří algoritmus navrhli. Byli to Ron Rivest, Adi Shamir a Len Adleman.

Ověření pravosti (Authentication)

Jde o techniku kterou uzel používá pro potvrzení totožnosti jiného uzlu. To zajišťuje, že komunikace je věrohodná. Jednoduchým příkladem je zadání hesla pro přístup ke zdrojům serveru k němuž se uživatel přihlašuje.

Kontrola neporušenosti (Integrity Check)

Tato služba zabezpečuje, že zpráva, kterou příjemce obdržel nebyla cestou upravena a dorazila tak, jak byla odeslána. Potenciální úpravy zahrnují změny, vymazání části zprávy nebo zápis přidané informace. Díky kontrole neporušenosti si příjemce může být jist, že nikdo zprávu nezměnil.

Nonrepudiation

Nonrepudiation zabraňuje tomu aby odesílatel nebo příjemce popřeli zprávu.

Cerifikáty (Certificates)

Certifikát je standardní nástroj, jak provázat veřejný klíč se jménem. Technika je založena na činiteli, jenž má důvěru ostatních (certifikační autorita) a umožňuje dvěma účastníkům, kteří spolu chtějí komunikovat vzájemné ověření totožnosti.
Předpokládejme situaci, kdy účastník B chce poslat účastníkovi A svůj veřejný klíč. Jak může účastník A vědět, že klíč byl skutečně odeslán Béčkem? Certifikáty tento problém řeší. Namísto toho aby B posílal A svůj klíč, pošle mu svůj certifikát a příjemce si může u jeho vydavatele ověřit totožnost Béčka.

Certifikát má tento obsah:

- subjekt – obsahuje identifikační informace, rozlišovací jméno a veřejný klíč;
- vydavatel – obsahuje certifikační autoritu, rozlišovací jméno a podpis;
- doba platnosti – určuje dobu po níž je certifikát platný;

- administrativní informace – může obsahovat informac jako jsou verze, seriové číslo, atd.

SSL protokol (3) - SSL Handshake Protocol

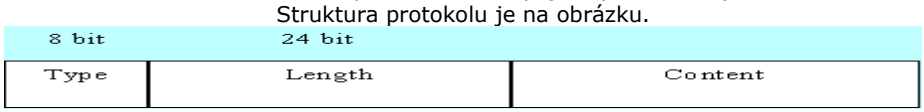
Tento protokol je také nazýván key-exchange protocol, neboli protokol pro výměnu klíčů. Jeho význam spočívá v ustavení bezpečné cesty (session) mezi dvěma účastníky.

Tento protokol je také nazýván key-exchange protocol, neboli protokol pro výměnu klíčů. Jeho význam spočívá v ustavení bezpečné cesty (session) mezi dvěma účastníky.

Činnost protokolu má několik důležitých stavů:

- ověření serveru klientem
- vyjednání společných šifrovacích algoritmů nebo šifer, které umí jak server, tak klient používat
- ověření klienta serverem (jako volitelná možnost!)
- použití šifrování s veřejným klíčem pro výměnu šifrovacích parametrů (sdílená hesla)
- ustavení zabezpečeného SSL spojení (connection)

Struktura protokolu je na obrázku.



Obr. 3 - Struktura protokolu SSL Handshake Protocol

Type: typ zprávy SSL handshake

Length: délka zprávy (v bytech)

Content: parametry přidané ke zprávě

Na dalším obrázku je vidět postup komunikace mezi klientem a serverem.

Obr. 4 - SSL Handshake Protocol - postup komunikace mezi klientem a serverem

Dalo by se následovat popisem jednotlivých stavů, ale tyto detaily nejsou cílem tohoto článku. Pokud by někdo ze čtenářů měl zájem o detaily, lze je vyhledat např. v odkazech uvedených na závěr seriálu.

Zkrácený handshake proces

V případě, že je již vytvořena cesta (session) – nový proces se snaží vytvořit spojení se serverem s nímž již komunikuje jiný proces – lze tuto cestu použít i pro nové spojení. V tomto případě není nutné provádět úplný handshake proces, ale stačí zkrácená verze využívající existující session ID.

SSL protokol (4) - Change Cipher Spec Protocol a Alert Protocol

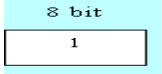
Petr Odvárka Tutoriály 16. května 2002

Tyto protokoly jsou používány v poslední fázi činnosti SSL Handshake protokolu pro přesun z vyčkávacího do provozního stavu a pro předávání informací o chybách objevujících se v průběhu celého spojení.

SSL Change Cipher Spec Protocol

Tento protokol je používán v poslední fázi činnosti SSL Handshake protokolu. Jeho účelem je umožnit účastníkům přesun z vyčkávacího do provozního stavu. To znamená, že účastníci ukončí použití algoritmus výměny klíčů a začnou používat šifrovací a ověřovací (MAC) algoritmy, které byly definovány v předchozích fázích Handshake protokolu.

Zpráva tohoto protokolu má délku 1 byte a je šifrována a komprimována pomocí dohodnutých protokolů.

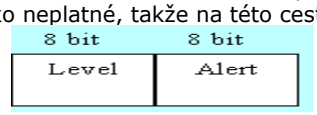


Obr. 5 - SSL Change Cipher Spec Protocol

SSL Alert Protocol

Významem tohoto protokolu je předávání informací o chybách objevujících se v průběhu celého spojení (connection). Výstrahy (alerts) jsou dvou úrovní – fatální a varovné.

Pokud se objeví fatální výstraha, spojení je okamžitě ukončeno. Ostatní spojení používající stejnou cestu (session) mohou pokračovat, ale session ID bude označeno jako neplatné, takže na této cestě nebude možné navázat žádné nové spojení.



Obr. 6 - SSL Alert Protocol

Level: indikuje fatální nebo varovnou výstrahu

Alert: indikuje specifickou výstrahu

Příklady fatálních výstrah:

- Bad_record_mac - špatná hodnota MAC
- Decompression_failure - délka zprávy po dekomprimaci překročila maximum

- Handshake_failure - chyba při vyjednávání parametrů

Příklady varovných výstrah:

- Certificate_expired - certifikát vypršel
- Certificate_revoked - certifikát byl zrušen tím kdo jej vystavil
- Unsupported_ certificate - nepodporovaný typ certifikátu

Kompletní seznam lze nalézt v doporučené literatuře.

SSL protokol – informační zdroje

Petr Odvárka Tutoriály 21. května 2002

O problematice SSL lze na Internetu možné najít poměrně dost informací. Tento seriál vychází z tutorialu, který je zveřejněn na webu RAD university a z materiálů o technologiích Alteon a Cisco.

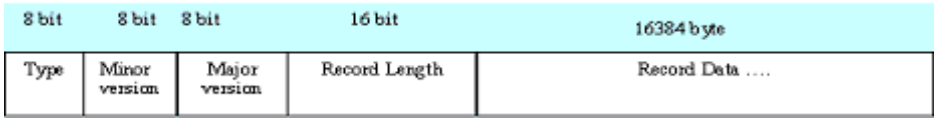
O problematice SSL lze na Internetu možné najít poměrně dost informací. Tento seriál vychází z tutorialu, který je zveřejněn na webu RAD university a z materiálů o technologiích Alteon a Cisco.

http://www.homeport.org/~adam/ssl.html - An overview of SSL version 2.
http://home.netscape.com/eng/ssl3/ssl-toc.html - The SSL Protocol version 3.0, internet draft.
http://developer.netscape.com/docs/manuals/security/sslin/contents.htm - Introduction to SSL.
http://colossus.net/SSL.html - The SSL protocol.
http://www.ultranet.com/~fhirsch/Papers/wwwj/article.html - Introducing SSL and certificates using SLeay.
http://www.faqs.org/rfcs/rfc2246.html - The TLS Protocol version 1.0.
http://www.itl.nist.gov/fipspubs/fip180-1.htm - Secure Hash Standard.
http://www.faqs.org/rfcs/rfc1321.html - MD5.
http://www.nortelnetworks.com/products/01/alteon/isdssl/index.html
http://www.cisco.com/warp/public/cc/pd/si/11000/prodlit/index.shtml

SSL protokol (5) - SSL Record Protocol

Přenášená data jsou v případě protokolu SSL balena do objektu nazývaného record. Record obsahuje hlavičku a data. Dále si ukážeme vývojový digram procesu vzniku datové části recordu.

Přenášená data jsou v případě protokolu SSL balena do objektu nazývaného record. Record obsahuje hlavičku a data.



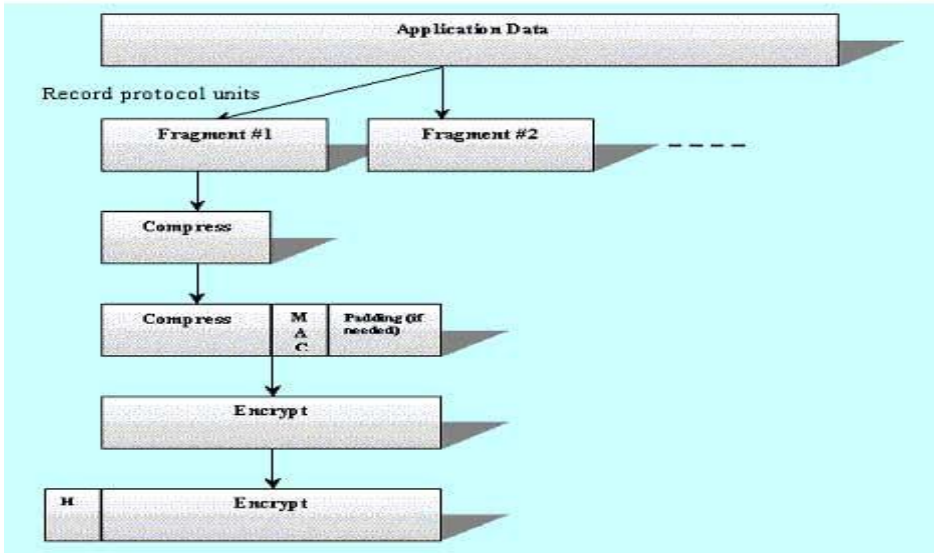
Obr. 7 - Hlavička SSL Record protokolu

Hlavička recordu je dlouhá 5 byte. Obsahuje tato pole:

- *Type* (8 bit) – indikuje datový typ a protokol vyšší vrstvy, který by měl data zpracovat. Typy jsou:
  - change_cipher_spec - změna specifikace šifrování
  - alert - *výstraha*
  - handshake
  - application_data - *aplikační data*
- *Version* (16 bit) – obsahuje informaci o verzi SSL protokolu (major i minor)
- *Length* (16 bit) – obsahuje informaci o délce datového pole.
- SSL Record Data

Datové části se podrobují 4 fázím:

1. fragmentace
2. komprimace (volitelně)
3. aplikace MAC (ověřovací kód zprávy)
4. šifrování



Obr. 8 - Vývojový digram procesu vzniku datové části recordu.

Fragmentace

Datové informace jsou děleny do ryze textových SSL recordů o délce 2^{14} byte nebo menších. Výsledek operace je nazýván *SSLPlaintext*.

Komprimace

V této fázi je na ryze textové recordy *SSLPlaintext* použit komprimační protokol dohodnutý v Handshake fázi. Výsledek je nazýván *SSLCompressed*. Komprimace nesmí způsobit žádnou ztrátu dat. Délka *SSLCompressed* nesmí být větší než výše uvedená maximální délka 2^{14}byte.

Aplikace MAC

V této fázi je k *SSLCompressed* připojena ověřovací informace MAC.

MAC je vypočítána jako hash funkce s následujícími parametry:

MAC-DATA = HASH (MAC-WRITE-SECRET, PAD2, HASH (MAC-WRITE-SECRET, PAD1, SEQUENCE-NUMBER, SSLCompressed.type, SSLCompressed.length, SSLCompressed.fragment))

Hešovací algoritmus byl domluven v Handshake procesu. Význam ostatních hodnot je následující:

- MAC-WRITE-SECRET klíč sdílený klientem a serverem
- PAD1 0x36 byte opakovaný 48 x pro MD5 a 40 x pro SHA
- PAD2 0x5C byte opakovaný 48 x pro MD5 a 40 x pro SHA
- SEQUENCE-NUMBER funguje jako počitadlo; každý účastník má dvě počitadla – jedno pro odeslané a druhé pro přijaté zprávy; počitadla jsou shozena na 0 v případě změny parametrů šifrování (např. změna klíče)

SSL podporuje 2 algoritmy hešování:

- MD5 s délkou výsledku 128-bit
- SHA – *Secure Hash Algorithm*, s délkou výsledku 160-bit

Šifrování

Používány jsou dva šifrovací algoritmy – proudový (stream) a blokový.

Proudový nepotřebuje doplňování velikosti a record může být použit ve formátu v jakém jej předal předchozí proces.

Blokový algoritmus požaduje doplnění délky recordu na požadovanou délku násobku bloku. Celková délka výstupu ze šifrování nesmí překročit 2^{14} byte.

SSL verze 3.0 používá následující algoritmy:

Blokové šifrování	Proudové šifrování
IDEA - 128 bit RC2 - 40 bit DES - 40 bit DES - 56 bit 3DES - 168 bit Fortezza - 80 bit	RC4 - 40 bit RC4 - 128 bit

SSL protokol (6) - příklady ochrany SSL před útoky (I)

Petr Odvárka Tutoriály 24. května 2002

V další části našeho seriálu následují příklady, které ukazují sílu bezpečnosti protokolu SSL a vysvětlují, jak protokol chrání proti různým útokům. Dnes se podíváme na ochranu před podvrhem identity a ochranu před zkreslením (Garbling Attacks).

V další části našeho seriálu následují příklady, které ukazují sílu bezpečnosti protokolu SSL a vysvětlují, jak protokol chrání proti různým útokům. Dnes se podíváme na ochranu před podvrhem identity a ochranu před zkreslením (Garbling Attacks).

SSL verze 2.0 má množství trhlin, které snižují úroveň jeho bezpečnosti. Verze 3.0 tyto trhliny odstraňuje.

Popis situace

Alice a Bob jsou dva komunikující účastníci označení A a B, Mallet je útočník a je označen jako M.

Jestliže je zpráva X zašifrována s použitím klíče, je označena jako klíč[X].

Výraz MAC[X] označuje ověřovací údaj pro zprávu X.

Hash[X] označuje heš zprávy X.

Garble[X] označuje zkreslenou (porušenou) zprávu X.

Ochrana před podvrhem identity

Řekněme, že si Alice přeje kontaktovat Boba a Bob má certifikát obsahující jeho veřejný klíč:

A->B: Ahoj

B->A: Dobrý den, já jsem Bob a tady je můj certifikát.

Nicméně kdokoliv může poslouchat komunikaci mezi Bobem a Alicí, což znamená, že např. Mallet může udělat následující:

A->B: Ahoj.

M->A: Dobrý den, já jsem Bob a tady je můj certifikát.

Ale protože Mallet nemá Bobův privátní klíč, nebude schopen se prokázat jako Bob:

A->B: Ahoj.

B->A: Dobrý den, já jsem Bob a tady je můj certifikát.

A->B: Jo, v pořádku

B->A: Alice, jsem to skutečně já – a potvrzuje to hešem zprávy:

bobs-private-key[Hash[Alice, jsem to skutečně já]].

To, co Bob právě udělal se nazývá elektronický (digitální) podpis. Pouze Bobův veřejný klíč může rozšifrovat druhou část zprávy. Alice používající veřejný klíč z Bobova certifikátu, může rozšifrovat druhou část zprávy, použít heš na první část a výsledek porovnat s hodnotou rozšifrované zprávy. Bob s Alicí si nejprve musí dohodnout hešovací funkci – to zde není ukázáno. Protože Mallet nezná Bobův privátní klíč, je jeho potenciální podvrh odhalen.

Ochrana před zkreslením (Garbling Attacks)

Alice si ověřila, že jde skutečně o Boba a nyní si s ním chce vyměnit klíč pro symetrické šifrování použité v další komunikaci (nejpoužívanější algoritmus je DES a jeho varianty).

A->B: Dobrá Bobe, tady máš klíč bob's-public-key[secret-key]

posílá klíč pro další komunikaci zabalený Bobovým veřejným klíčem

B->A: secret-key[bob's message]

posílá zprávu zašifrovanou

Nicméně Mallet může vnést do procesu nový prvek a udělat tohle:

A->B: Ahoj.

M->B: Ahoj.

B->A: Dobrý den, já jsem Bob a tady je můj certifikát.

M->A: Dobrý den, já jsem Bob a tady je můj certifikát.
A->B: Jo, v pořádku
M->B: Jo, v pořádku
B->A: Alice, jsem to skutečně já – a potvrzuje to hešem zprávy:
bobs-private-key[Hash[Alice, jsem to skutečně já]].
M->A: Alice, jsem to skutečně já – a potvrzuje to hešem zprávy:
bobs-private-key[Hash[Alice, jsem to skutečně já]].
A->B: Dobrá Bobe, tady máš klíč bob's-public-key[secret-key]
M->B: Dobrá Bobe, tady máš klíč bob's-public-key[secret-key]
B->A: secret-key[bob's message]
M->A: Garble[secret-key[bob's message]]

Alice neví, že Mallet posunuje informaci mezi ní a Bobem, takže když Mallet vidí, že byl vyměněn veřejný klíč, zkreslí komunikaci a doufá, že Alice, která teď věří Bobovým zprávám, bude věřit zkreslené zprávě a odpoví na ni. Mallet nezná klíč a nemůže vytvořit platnou zprávu, ale existuje šance, že se mu to podaří.

Aby došlo k odhalení tohoto způsobu útoku, je použit MAC (ověřovací kód zprávy):

A->B: Dobrá Bobe, tady máš klíč bob's-public-key[secret-key]
secret-key[nějaká zpráva, MAC[nějaká zpráva]]

Nyní mohou Bob i Alice potvrdit pravost svých zpráv výpočtem MAC, porovnáním přijaté hodnoty MAC s výsledkem vlastního výpočtu. Mallet může zkoušet hádat hodnoty MAC pro své změněné zprávy, ale s dobrou hešovací funkcí jej jeho šance prakticky nulová.

SSL protokol (7) - příklady ochrany SSL před útoky (II)

V druhé části příkladů, ukazujících sílu bezpečnosti protokolu SSL se podíváme na ochranu před odpovídáním na zprávy, útokem vyříznutím a vložením, útokem přestavením specifikace šifrování, útokem přestavením verze, slovníkovými útoky a útokem na krátké bloky.

V druhé části příkladů, ukazujících sílu bezpečnosti protokolu SSL se podíváme na ochranu před odpovídáním na zprávy, útokem vyříznutím a vložením, útokem přestavením specifikace šifrování, útokem přestavením verze, slovníkovými útoky a útokem na krátké bloky.

Ochrana před odpovídáním na zprávy (Preventing Replaying Messages)

Mallet se nechce vzdát a odpovídá záznamy předchozích komunikací. To může způsobit poměrně dost problémů. Nicméně, jestliže je do zprávy přidán a kontrolován nějaký náhodný element (např. čas odeslání zprávy), odpovídání zaznamenanou zprávou lze odhalit. Mimoto je dobrou ochranou proti tomuto typu útoku použití sekvenčních čísel pro výpočet MAC. Účastník spojení zprávu nepřijme pokud je sekvenční číslo mimo očekávanou hodnotu počitadla.

Ochrana před útokem vyříznutím a vložením (Cut and Paste Attacks)

Cut-and-paste attack znamená, že útočník uprostřed spojení vyřízne kus šifrované zprávy z jednoho paketu a vloží jej do jiného paketu, takže se přijímající účastník neúmyslně vzdá rozšifrovaného textu.

SSL 3.0 zabraňuje tomuto typu útoku použitím silných ověřovacích algoritmů před tím, než každý šifrovaný paket odešle. To zajišťuje efektivní ochranu před nepřátelskou modifikací.

Ochrana před útokem přestavením specifikace šifrování

SSL v2.0 umožňoval změnit během komunikace úroveň zabezpečení. Díky tomu mohli útočníci vynutit změnu zabezpečení na slabší úroveň. Je to známo jako *CipherSuite rollback attack*. SSL v3.0 odstranila tuto slabinu tím, že není možné měnit úroveň zabezpečení.

Ochrana před útokem přestavením verze

Při tomto typu útoku útočník mění verzi zadanou v client_hello message na verzi SSL 2.0. To přinutí server zvolit slabší verzi SSL protokolu – 2.0 a pak je prolomení mnohem jednodušší.

Oba účastníci vyjednávání mají možnost zjistit zda protějšek podporuje vezi 2.0 i 3.0. Klient je schopen vložit indikátory o tom, že podporuje verzi 3.0 do RSA šifry. Server je schopen na základě těchto indikátorů rozpoznat, že klient verzi 3.0 podporuje. Indikace v rámci RSA je dostatečná, protože jediná cesta k přestavení verze na 2.0 vede právě přes RSA – verze 2.0 totiž jiný algoritmus nepodporuje.

Ochrana před slovníkovými útoky (Dictionary attacks)

Tento útok funguje v situaci kdy útočník zná určitou část originální zprávy. V tomto případě se může pokusit šifrovat text s jakýmkoliv možným klíčem dokud neobjeví správný výraz. Objeví-li jej, celá zpráva může být s tímto klíčem rozšifrována. SSL se chrání před tímto typem útoku silným šifrovacím algoritmem jako jsou IDEA (128 bit) nebo 3DES (168 bit).

Útok na krátké bloky (Short-block attacks)

Když poslední blok zprávy obsahuje 1 byte ryzího textu a zbytková část bloku je doplněna na velikost, lze tento blok relativně snadné rozšifrovat. Ve skutečnosti tento typ útoku SSL nehrozí, neboť nepoužívá tak krátké bloky.

SSL protokol (8) - příklady použití a jednotlivé verze protokolu

V dnešním pokračování se podíváme na několik příkladů pro použití protokolu SSL, jednotlivé verze protokolu SSL - rozdíly mezi verzemi SSL v2.0 a SSL v3.0 a protokol TLS (Transaction Layer Security), který je také někdy označován jako SSL 3.1.

V dnešním pokračování se podíváme na několik příkladů pro použití protokolu SSL, jednotlivé verze protokolu SSL - rozdíly mezi verzemi SSL v2.0 a SSL v3.0 a protokol TLS (Transaction Layer Security), který je také někdy označován jako SSL 3.1.

Několik příkladů pro použití protokolu SSL

Organizace, která chce prostřednictvím Internetu poskytnout zabezpečenou komunikaci, může použít protokol SSL.

Bankovní systémy používají tento protokol umožňující klientům prohlížet důvěrná data pomocí klasických internetových prohlížečů a provádět manipulaci s účty z domu.

Stejnou myšlenku lze použít na vzdělávací organizace. Studenti mohou prohlížet své personální údaje – např. výsledky testů nebo zkoušek.

Velké množství e-commerce aplikací na Internetu používá ochranu pomocí SSL.

Nejznámějším využitím SSL je zabezpečení přístupu k webovým serverům protokolem HTTP. Protokol HTTP zabezpečený pomocí SSL se nazývá HTTPS. Na straně serveru používá namísto portu 80 port 473.

Verze protokolu SSL

Protokol SSL byl vyvinut společností Netscape Communications. Verze, která byla uvedena pro používání, byla označena jako 2.0. Měla poměrně dost slabých míst a byla snadno napadnutelná (viz. výše). První významná úprava bezpečnostních vlastností

byla ve verzi 3.0. Později, jako další oprava SSL protokolu provedená v rámci IETF, vznikl protokol TLS (Transaction Layer Security), který je také někdy označován jako SSL 3.1. SSL 3.0 a TSL jsou velmi podobné a mají několik drobných rozdílů.
Rozdíly mezi SSL v2.0 a SSL v3.0
V kapitole Příklady ochrany SSL před různými útoky jsou uvedeny některé možné způsoby napadení zabezpečeného spojení. Většina z nich byla identifikována a popsána právě při používání SSL 2.0 a pokud je verze 2.0 nebyla schopna ošetřit, vedlo to k vylepšením uvedeným v novější verzi – SSL v3.0.
Kromě ochrany před výše uvedenými způsoby napadení zavádí SSL 3.0 některé funkční změny. Např. zodpovědnost za volbu šifrovacích a kompresních algoritmů. Ve verzi 2.0 byla na straně klienta, od verze 3.0 je na straně serveru.
TLS
TLS (Transport Layer Security) je protokol vytvořený v rámci IETF jako internetový standard pro nahrazení SSL v3.0. TLS zajišťuje komunikacím v prostředí Internetu soukromí a umožňuje klient/server aplikacím předejít odposlechu, falšování nebo padělání zpráv.
Protokol TLS je založen na specifikaci protokolu SSL 3.0 publikované firmou Netscape. Rozdíly mezi TLS 1.0 a SSL 3.0 nejsou dramatické, ale jsou natolik významné, že spolu protokoly nespolupracují (ačkoliv TLS 1.0 obsahuje mechanismus pro zpětnou kompatibilitu se SSL 3.0).
Rozdíly mezi SSL v3.0 a TLS v1.0
Při použití blokového šifrovacího algoritmu je nutné datové bloky doplnit na násobky určité velikosti.
V TLS protokolu může doplnění končit v jakékoliv délce, která je násobkem délky bloku (do velikosti 255 byte). Například – jestliže mají data před šifrováním délku 79 byte a délka šifrovacího bloku je 8, doplněk může být dlouhý 1, 9, 17, atd. až 249 byte.
V dřívějších verzích SSL protokolu musí být doplněk nejkratší možné velikosti. V případě uvedeného příkladu je to tedy 1 byte. Použití proměnné délky doplňku stěžuje útok pomocí analýzy délky zpráv.
TLS podporuje všechny výstrahy definované v SSL 3.0 s výjimkou zprávy no_certificate. TLS podporuje všechny mechanismy výměny klíčů i šifrování používané SSL 3.0 s výjimkou výměny klíčů typu Fortrezza a symetrického šifrovacího algoritmu. Určitý rozdíl je i ve způsobu výpočtu MAC, ale výsledná úroveň zabezpečení je shodná pro oba protokoly. Drobné rozdíly jsou ve zprávách typu certificate_verify a ukončení procesu.
Poté co byl protokol TLS publikován jako internetový standard, začal být obecně přijímán pro ověřování a šifrování komunikace mezi klienty a servery.
Vzhledem k popsaným rozdílům mezi SSL 3.0 a TLS 1.0 (SSL 3.1), lze toto pojednání považovat jako dostatečné i v ohledu k popisu principů na nichž funguje protokol TLS.

SSL protokol (9) - vliv na výkon serverů a řešení problému
Je jasné, že server, který používá zabezpečenou komunikaci – např. HTTPS, bude více vytížen než server plnící obdobné funkce bez zabezpečení (HTTP). Jak je tato přidaná zátěž významná? Poměrně dost.
Vliv protokolu SSL na výkon serverů
Je jasné, že server, který používá zabezpečenou komunikaci – např. HTTPS, bude více vytížen než server plnící obdobné funkce bez zabezpečení (HTTP). Jak je tato přidaná zátěž významná? Poměrně dost.
Existují studie, které prokazují, že HTTPS protokol snižuje výkon serveru až na zlomky výkonu s protokolem HTTP! Viz. graf, jehož zdrojem je Networkshop Inc.
Jak tento problém řešit? Možností je několik a každé z nich je vhodné pro určité prostředí. Této problematice se budeme věnovat v následující kapitole.
Řešení problematiky snížení výkonu serverů
Zvýšení výpočetního výkonu serveru
Nejjednodušší, ale nejproblematičtější řešení. Nelze jej škálovat. Řešení lze použít pro omezený počet současně připojených uživatelů, nižší než v následujícím případě.
Vybavení serveru hardwarovým SSL akcelerátorem
Hardwarový akcelerátor zajišťuje převzetí určitých matematických operací spojených se šifrováním. Toto řešení je vhodné do prostředí, kde je pouze jeden server, příp. maximálně jeden server pro daný účel.
Použití externího akcelerátoru
Externí akcelerátory jsou specializovaná zařízení, která zajišťují vytvoření bezpečného, šifrovaného spojení s uživatelem (protokol HTTPS). Komunikace mezi SSL akcelerátorem a servery probíhá transparentně (protokol HTTP). Protože je komunikace směrem na servery transparentní, lze zde použít metody přepínání na L7 zajišťující redundanci nebo load balancing aplikací (viz. seriál).
Principielně se nabízejí dvě varianty samostatných SSL akcelerátorů. Buď jako průchozí zařízení – tj. jeden port je směrem k uživateli a druhý směrem k serverům nebo jako paralelní zařízení s určitou logikou řízení toku.
Použití "průchozích" SSL akcelerátorů

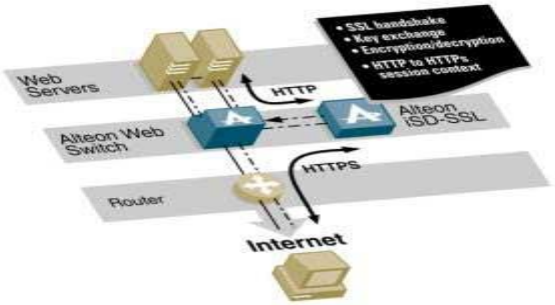
Obr. 9 - Použití "průchozích" SSL akcelerátorů
Průchozí SSL akcelerátory mají na první pohled jednu velkou výhodu – je jí relativní jednoduchost použití, neboť prvek kromě SSL akcelerace zajistí i přepínání podle obsahu (content switching). Lze tak zajistit určitou škálovatelnost aplikací za prvkem. Může být použito několik serverů s rozkládáním zátěže a redundancí. Komplikovaná je naopak redundance a rozkládání zátěže v prvku samotném. Druhou slabinou může být komplikovaná konfigurace a zabezpečení komunikací. Na prvku musí být vytvořeny sady filtrů omezujících průchod protokolů prvkem. O čem zde píši bude jasnější z dalšího textu, věnovaného "paralelním" SSL akcelerátorům.
… pokračování

SSL protokol (10) - použití paralelních akcelerátorů
Paralelně zapojované SSL akcelarátory jsou přísně specializovaná zařízení, která dělají jediné – šifrují a dešifrují. Lze je popsat jako jednoúčelová zástupná (proxy) zařízení.
Paralelně zapojované SSL akcelarátory jsou přísně specializovaná zařízení, která dělají jediné – šifrují a dešifrují. Lze je popsat jako jednoúčelová zástupná (proxy) zařízení.

Použití "paralelních" SSL akcelerátorů

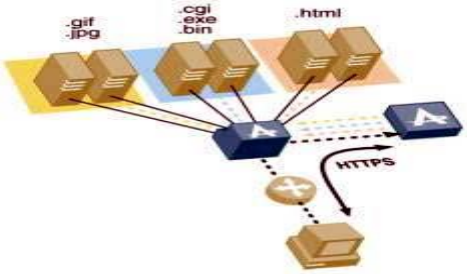
Paralelně zapojované SSL akcelarátory jsou přísně specializovaná zařízení, která dělají jediné – šifrují a dešifrují. Lze je popsat jako jednoúčelová zástupná (proxy) zařízení.

Zatímco v případě "průchozích" akcelerátorů jsou servery adresovány přímo, v případě paralelních SSL akcelerátorů jsou servery adresovány prostřednictvím akcelerátoru. Nejlépe je to vidět na následujícím obrázku, kde je SSL akcelerátor použit ve spojení s content switchem. Uživatel, který se servery komunikuje zabezpečeným kanálem vlastně zabezpečeně nekomunikuje se servery, ale s SSL akcelerátorem. Mezi ním a servery probíhá transparentní komunikace. Content switch má tedy možnost zajišťovat redundanci a rozkládání zátěže. Pokud přes něj jdou zašifrovaná data, nemůže koukat do obsahu a tudíž rozkládání zátěže a z ni redundanci nelze dělat!



Obr. 10 - Použití "paralelních" SSL akcelerátorů

Na následujícím obrázku je zobrazena výhoda kombinace content switche a SSL akcelerátoru zmíněná v předchozím textu.



Obr. 11 - Kombinace content switche a SSL akcelerátoru

Škálování výkonu a zabezpečení dostupnosti lze v tomto řešení provádět velice elegantně. Příště si popíšeme možný vývoj topologie s technologií Alteon.

SSL protokol (11) - aplikace technologie Alteon

V dalším pokračování seriálu se podíváme na postup aplikace technologie Alteon při řešení problému snížení výkonu serverů.

V dalším pokračování seriálu se podíváme na postup aplikace technologie Alteon při řešení problému snížení výkonu serverů.

Postup aplikace technologie Alteon

Aplikace jednoho SSL akcelerátoru

Uživatel komunikuje s SSL akcelerátorem – ten ve vztahu k němu vystupuje jako server. Přes SSL akcelerátor může procházet pouze šifrovaná komunikace! To má poměrně významný dopad na zabezpečení. Není možné aby přes něj prošel nezabezpečený datový tok. Nezabezpečené datové toky (HTTP, FTP...) musí být směrovány přímo na server.

Obr. 12 - Aplikace jednoho SSL akcelerátoru

Postup komunikace:

- uživatel naváže HTTPS spojení s SSL akcelerátorem (server není zveřejněn přímo – adresa – *např.* secure.svetsiti.cz - není směřována na server, ale na SSL akcelerátor);
- SSL akcelerátor naváže HTTP spojení se serverem;
- uživatel pošle požadavek v HTTPS na SSL akcelerátor, ten zajistí dešifrování a předá jej v HTTP na server;
- server odpoví akcelerátoru v HTTP, ten odpověď zašifruje a v HTTPS odešle uživateli.

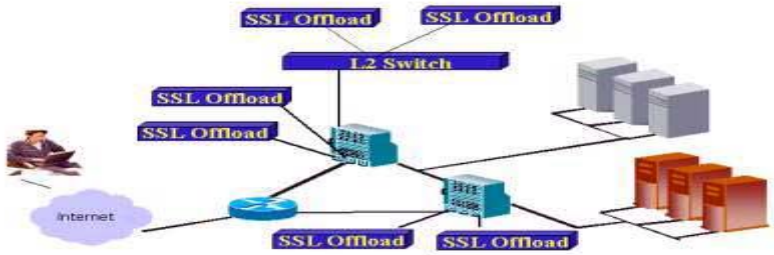
Přidání content přepínače

Přidáním content přepínače se do systému vnáší škálovatelnost a funkce rozkládání zátěže na servery. V obrázku je popsán postup komunikace. Ten je s některými odlišnostmi obdobou toho, jak komunikace probíhá v předchozím případě.

Obr. 13 - Přidání content přepínače

- Klient posílá požadavek HTTPS;
- Content switch přehazuje požadavek na portu 443 na SSL akcelerátor (příp. na jejich skupinu);
- iSD-SSL zajistí navázání SSL spojení (hand shake);
- iSD-SSL iniciuje HTTP spojení (port 80) se serverem;
- Switch vybírá reálný server na základě pravidel definovaných pro rozkládání zátěže nebo redundanci;
- Server odpovídá na HTTP požadavky SSL akcelerátoru;
- SSL akcelerátor šifruje data od serveru a prostřednictvím HTTPS je posílá klientovi.

Doplnění na plně redundantní prostředí



Obr. 14 - Doplnění na plně redundantní prostředí

Domnívám se, že uvedený obrázek je potřebné okomentovat pouze několika větami:

- v případě potřeby zvýšení výkonu pro SSL akcelaraci je jednoduše přidán další SSL akcelerátor;
- redundace content switchů lze doplnit redundantním připojením na skupiny akcelerátorů; ty mohou být připojeny buď přímo na content switche nebo pomocí L2 přepínačů;
- samozřejmě lze zajistit i redundantní připojení k Internetu.

SSL protokol (12) - zapojení prvků SCA 11000 fy Cisco

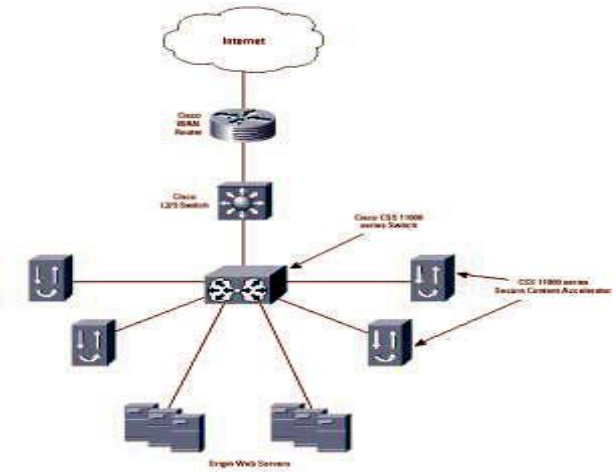
V závěrečné díle seiálu se podíváme na řešení společnost Cisco, která se na trhu prosazuje s nejširší škálou síťových produktů a má v nabídce samozřejmě i komplexní řešení CDN (Content Delivery Networks) prvků, mezi něž SSL Akcelerátory patří.

Společnost Cisco, která se na trhu prosazuje s nejširší škálou síťových produktů má v nabídce samozřejmě i komplexní řešení CDN (Content Delivery Networks) prvků, mezi něž SSL Akcelerátory patří.

Zapojení prvků SCA 11000 fy Cisco

Společnost Cisco, která se na trhu prosazuje s nejširší škálou síťových produktů má v nabídce samozřejmě i komplexní řešení CDN (Content Delivery Networks) prvků, mezi něž SSL Akcelerátory patří.

Prvky řady Cisco SCA (Secure Content Accelerator) 11000 ve spojení s prvkem řady CSS (Content service switch) 11000, zajišťuje řešení podobné tomu, které bylo uvedeno u řešení postaveného na prvcích Alteon.



Obr. 15 - Zapojení prvků SCA 11000 fy Cisco

Závěr

Nejdůležitější poznatky z tohoto textu jsou následující:

- je nutné používat SSL ve verzi min. 3.0 (tedy min. do doby než budou i v ní objeveny zásadní chyby a bude provedena významnější náhrada než verze 3.1 – tedy protokol TSL);
- protokol SSL může kromě šifrování i komprimovat data, čímž se zajímavě zefektivňuje využití linek;
- k ustavení spojení stačí pouze jeden certifikát a to certifikát serveru; certifikát klienta není požadován, ale může zvýšit úroveň důvěry serveru klientovi.

SSL protokol je poměrně významným konzumentem výkonu procesoru, existuje několik úrovní možného řešení; pro volbu toho nejvhodnějšího je nutné mít představu o dalším rozvoji aplikačního prostředí; pro využití bez dalšího rozvoje je vhodný SSL akcelerátor ve formě adaptéru do serveru, pro větší řešení je nutné jít do škálovatelného řešení založeného na externích SSL akcelerátorech v kombinaci s přepínáním podle obsahu (content switching).

Společnost Infinity a.s., v níž tento článek vznikl, je jako jedna z mála v republice vybavena demo a servisním skladem SSL akcelerátorů a content switchů; může tedy zákazníkům nabídnout, kromě teoretických úvah, i praktické nasazení technologie s testováním v reálném prostředí!

Doplnění síťového slovníčku z bezpečnostní problematiky

Jako doplnění seriálu o zajištění bezpečného přenosu dat po Internetu pomocí protokolu SSL přinášíme další část sítového slovníčku z bezpečnostní problematiky.

Jako doplnění seriálu o zajištění bezpečného přenosu dat po Internetu pomocí protokolu SSL. přinášíme další část sítového slovníčku z bezpečnostní problematiky

IPSec (IP Security)

Sada protokolů vyvinutých IETF pro bezpečnou výměnu IP paketů; IPSec je jednou z nejpoužívanějších technologií pro vytváření VPN (virtuální privátní sítě).

IPSec podporuje dva režimy – trasportní a tunelovací; transportní šifruje pouze datovou část paketu a hlavičky nechává nedotčené; tunelovací šifruje paket jako celek, včetně původní IP hlavičky a výsledek balí do nového paketu s novou hlavičkou.

(http://www.ipsec.com/)

VRRP (Virtual Router Redundancy Protocol)

Proprietární protokol společnosti Nortel Networks pro zajištění dostupnosti odchozí brány její redundancí na dvou a více směrovačích; protokol funguje na základě spolupráce min. dvou spolupracujících směrovačů, první z nich má IP adresu odchozí brány jako standardní interfejs, druhý z nich je schopen IP adresu odchozí brány převzít v případě výpadku primárního směrovače; v případě použití VRRP protokolu je použita virtuální MAC adresa – ne MAC adresa odpovídající interfejsu; pravidelným informováním se o stavu (heartbeat pakety) se směrovače vzájemně ujišťují o stavu; pokud přestane primární směrovač odpovídat, převezme adresu odchozí brány směrovač záložní.

(http://www.nortelnetworks.com/solutions/lan/collateral/ppvrrp.pdf)

HSRP (Hot Standby Router Protocol)

Proprietární protokol společnosti Cisco Systems pro zajištění dostupnosti odchozí brány její redundancí na dvou a více směrovačích; protokol funguje na základě spolupráce min. dvou směrovačů, které mají vytvořenu virtuální (tzv. démon) IP a MAC adresu, již sdílí; pravidelným informováním se o stavu (heartbeat pakety) se vzájemně ujišťují o stavu; pokud přestane primární směrovač odpovídat, převezme virtuální adresu směrovač záložní.

(http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/cs009.htm)

L2TP (Layer 2 Tunneling Protocol)

Tunelovací protokol založený na spojení funkcí a vlastností protokolů L2F společnosti Cisco Systems a PPTP společnosti Microsoft.

PPPoE Point-to-Point Protocol Over Ethernet

PPPoE je technologie zajišťující vytváření Point-to-Point spojení prostřednictvím připojení na Ethernet; vlastnost je využívaná zejména na DSL linkách, v prostředí kabelových modemů a bezdrátového připojení na Internet; základní definice protokolu je v RFC 2516.

(http://www.carricksolutions.com/pppoe.htm#1)

RC4

Šifrovací algoritmus vyvinutý Ronaldem Rivestem v roce 1987; algoritmus používá klíč s délkou 1 do 2048 bitů; nejčastěji jsou používány hodnoty 40 a 128 bitů.

(http://www.ncat.edu/~grogans/algorithm_history_and_descriptio.htm)

SHA-1 (Secure Hash Algorithm)

Algoritmus vyvinutý v NIST (National Institute of Standards and Technology) pro podporu elektronického podpisu; výsledkem algoritmu je 160ti bitový řetězec sloužící jako jednoznačné potvrzení pravosti obsahu; změnou jakéhokoliv byte původního obsahu se mění i výsledný řetězec (message digest).

(http://www.secure-hash-algorithm-md5-sha-1.co.uk/)

MD5

Algoritmus vyvinutý Ronaldem Rivestem pro podporu elektronického podpisu; výsledkem algoritmu je 128 bitový řetězec sloužící jako jednoznačné potvrzení pravosti obsahu; změnou jakéhokoliv byte původního obsahu se mění i výsledný řetězec (message digest); díky tomu, že je výsledný řetězec kratší, je proces jeho výpočtu rychleší než v případě algoritmu SHA-1