2/4 NOVÉ ZPŮSOBY JAK PORAZIT SSL - SSLSTRIP

Protokol SSL

SSL neboli Secure Socket Layer je protokol, resp. vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizaci komunikujících stran. Běžně jej využívá každý, např. v rámci internet bankingu. Ze SSL se stala fráze, která je běžným uživatelům webových služeb opakována tak často, že si začnou myslet, že je to pravda. Konečně, příkladů je více než dost, pro ukázku vyjádření dvou velmi známých bank v ČR.

SSL v tiskových prohlášeních

„Bezpečnost komunikace s bankou přes internet: Pro základní bezpečnost Vašich příkazů musí být zajištěno, aby příkaz k bankovní operaci nebyl nikým modifikován. K tomuto účelu ***je použito silné 128bitové šifrování komunikace s bankou po internetu pomocí technologie SSL****. Pro navázání šifrované komunikace* ***je navíc použit certifikát serveru banky vydaný důvěryhodnou certifikační autoritou****, který zajistí, že*

skutečně komunikujete s bankou a ne s někým, kdo se za aplikaci internetového bankovnictví pouze vydává.“

*„**Pro zajištění bezpečnosti vybudovala banka certifikační autoritu** a využila svou infrastrukturu veřejných klíčů (PKI). ... **Veškerá komunikace probíhá v SSL** (Secure Socket Layer) a ... Tím jsou **uspokojeny veškeré požadavky na bezpečnou komunikaci**, tj. identifikaci, autentikaci, autorizaci, důvěrnost, integritu a nepopiratelnost.“*

Podobná vyjádření jsou samozřejmě silným zjednodušením reálné skutečnosti, ale přiznejme si, že je to stále víc, než ví 90 % obyvatel ČR, resp. kteréhokoliv rozvinutého státu na světě.

Bezpečnost není černobílá

Na SSL, tedy obecně na problematiku šifrování komunikace, je potřeba pohlížet z více směrů. V pozitivním slova smyslu je jejím cílem zajistit, aby data putovala v zabezpečeném kanálu, nešla tak číst cizím subjektem, natož provádět nepozorované modifikace. Bezpečnost je ale složitá soustava, kde věci nejsou bílé a černé. Jde třeba o to, jestli šifrovaná data neobsahují něco nebezpečného. Nemyslím teď jen zájem většiny vlád a silových složek číst a odposlouchávat libovolnou komunikaci na světě a vše odůvodňovat hrozbou terorismu. Mnohem častěji jde o nevhodné použití této technologie, zejména včasné ukončení SSL komunikace. Je dost nebezpečné to dělat až na straně aplikačního serveru. Máte IDS? Aplikační firewally? V tuto chvíli jsou vám k ničemu. Tento příklad uvádím pouze proto, aby bylo jasnější, že šifrování je v mnoha případech hezká idea, která bývá nevhodně přenesena do elektronického světa.

Podmínky realizace dále popisovaných útoků

V rámci dalšího uvažování o nových hrozbách pomineme ty staré. Tedy např. to, že SSL je jen rámec, který využívá různé šifrovací algoritmy, různé délky klíčů atd. Takže lze používat SSL, kde bude využit např. DES a 40bitový klíč. To určitě není dobře. Ne kvůli SSL, ale kvůli DES, který je již několik let považován za „mrtvý". Útoky, které budeme dále rozebírat, počítají s tím, že si server s klientem pokusí domluvit to nejlepší dostupné šifrování a nejdelší možný klíč. Dále pro upřesnění budeme předpokládat, že většinu technik nelze používat bez kombinace s jinými a že každý útok musí být zařazen do nějakého prostředí. V našem případě se tedy zaměříme na útoky, které využívají MITM - Man In The Middle techniky. Ty nejjednodušší mohou být na LAN síti, zejména ARP spoofing, případně na WIFI. Ale nelze zcela zatracovat ani kombinace, kdy útočník navede svou oběť např. kvůli překlepu URL na svou stránku. Např. paypal.com je legitimní stránka, ale paypall.com nebo paypaI.com jen počítají s tím, že uživatel nezná dobře anglický jazyk nabo může přehlédnout dvě podobně vypadající písmena.

První verze MITM proti SSL - statická forma

Nyní krátký exkurz do historie. Protože je SSL tak často více méně marketingovým způsobem zneužíváno, bylo a je terčem zájmu bezpečnostních specialistů i hackerů. Příkladů je více, ale blíže probereme pouze použití programu Ettercap.[1] Jedná se o víceúčelový nástroj na sniffing síťového provozu, doplněný o techniky vybrané techniky MITM, včetně arpspoof.[2] Jeho první verze umožňovaly podvrhávat statický certifikát, který na uživatele asi nepůsobil moc důvěryhodně. Vše bylo statické - jméno certifikační autority,

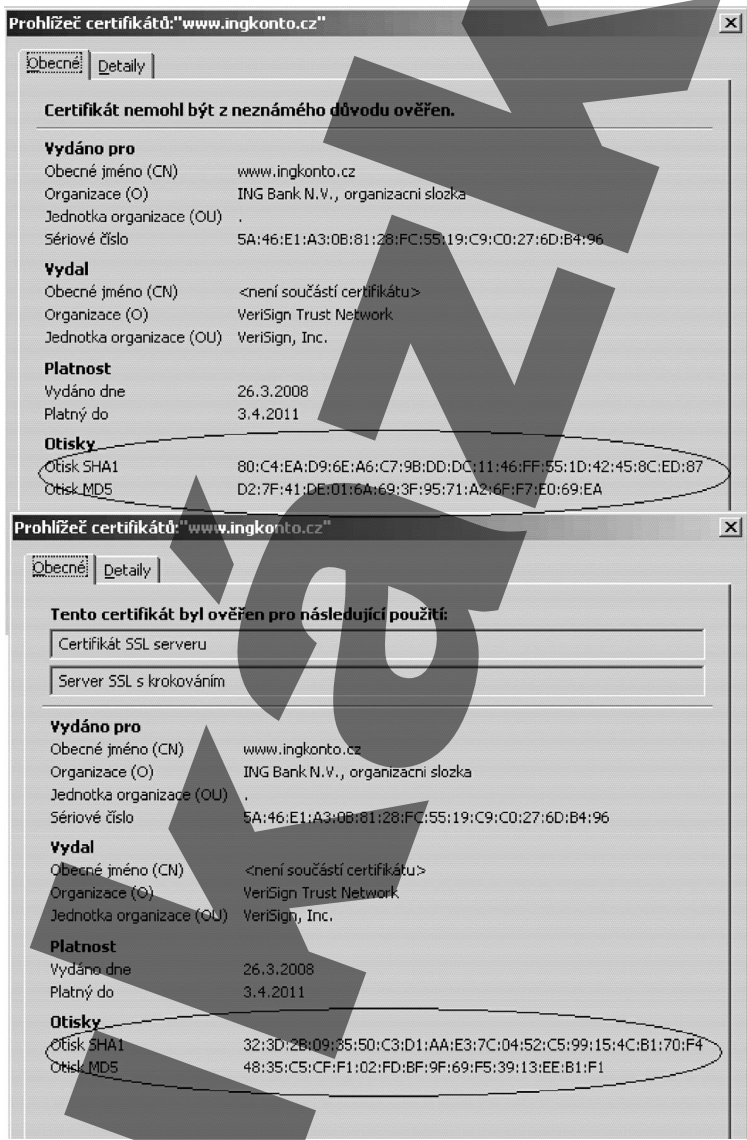
[1] Autoři Alberto Ornaghi a Marco Valleri.
[2] Vlastní technika je blíže popsána v kapitole o hackingu.

jméno webové stránky, divný čas atd. I na něj ale mnoho lidí kliklo.

Druhá verze MITM proti SSL - dynamická forma

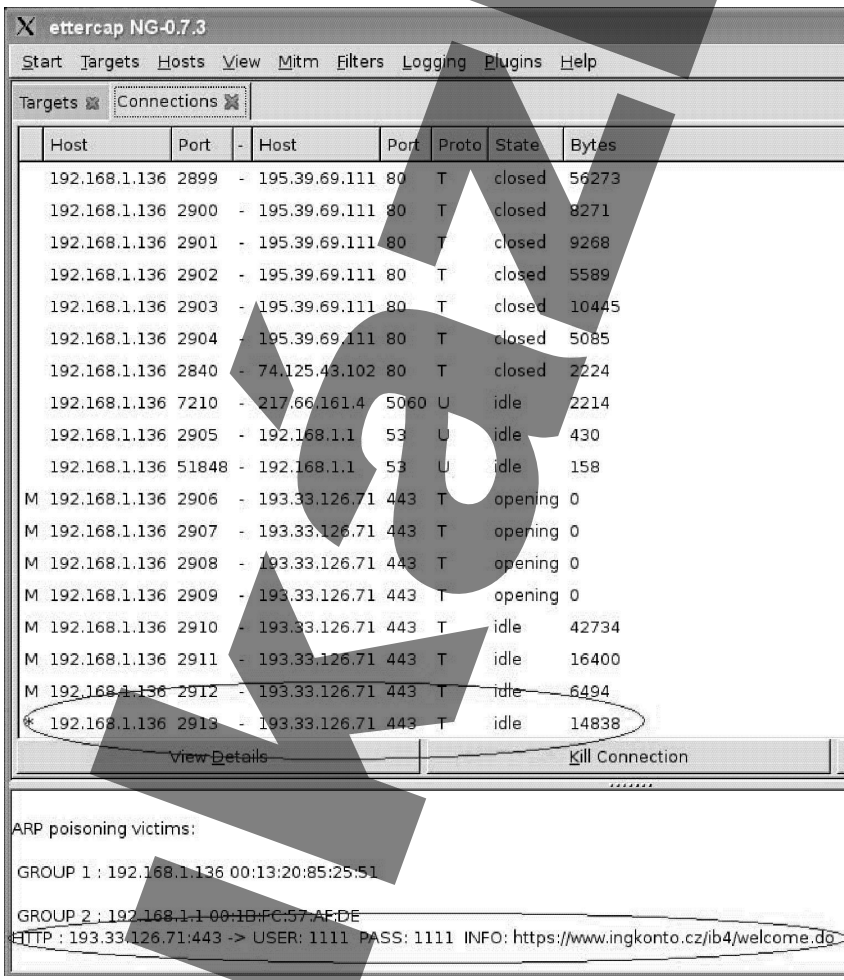
V dalších verzích Ettercap se tato funkcionalita o dost změnila. Výsledkem byla „celkem slušná kopie“, kdy se „on the fly“, tedy průběžně, zkopíroval původní certifikát, ale místo „hrátek“ s MD5 kolizí se jen zfalšoval tzv. fingerprint. Výsledek byl velmi elegantní a výrazně se zvětšilo procento uživatelů, kteří nakonec klikli na „ano“ a dodanému certifikátu začali důvěřovat. Originál se samozřejmě liší, ale podvržený certifikát má v pořádku jméno organizace, certifikační autoritu, platnost. Jen na začátku je uvedeno „Certifikát nemohl být z neznámeného důvodu ověřen“, místo „Tento certifikát byl ověřen pro následující použití“. Použitý mechanismus je blíže popsán v manuálu programu Ettercap.

Rozdíl v certifikátech pouze v poli „Otisky“



V případě, že uživatel neodolal a kliknul - vždyť on(a) se skutečně potřebuje přihlásit do banky, hacker už mohl v klidu sledovat, jak mu oběť sděluje své jméno a heslo.

Viditelná hesla v případě MITM proti SSL v programu Ettercap

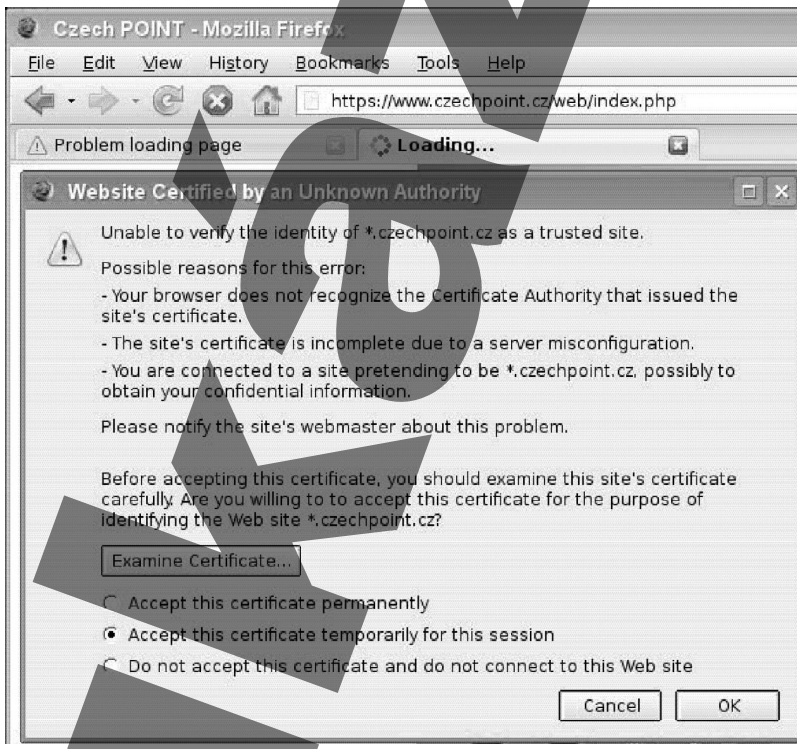


Změna pozitivního modelu za negativní - webové prohlížeče

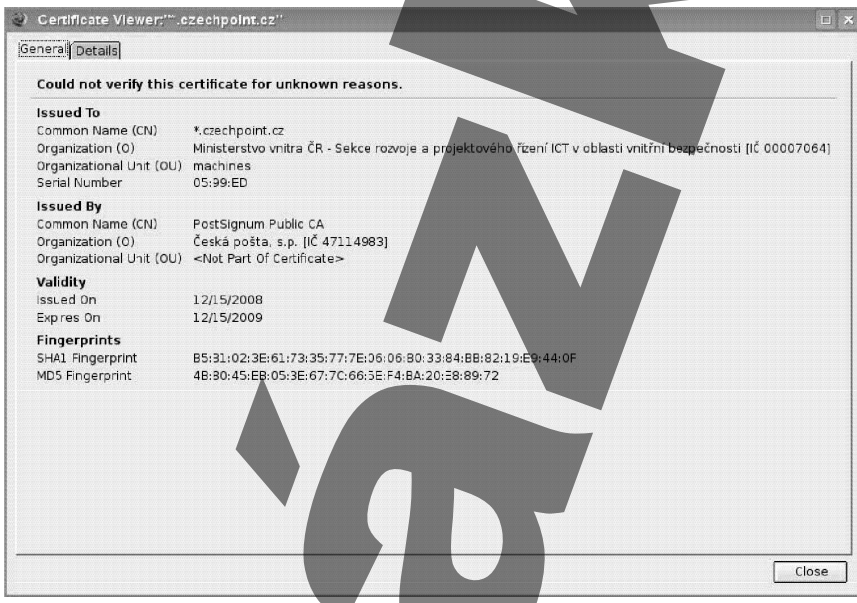
Demonstrovaný útok fungoval téměř ideálně, dokud se nezačalo pomalu měnit zabezpečení webových prohlížečů. Boj se zlozvyky je vcelku složitý, ale zde stačilo poměrně málo - změnit pozitivní model vyhodnocení SSL certifikátu za negativní. V čem je hlavní rozdíl? Je téměř jisté, že i když uživatel vidí výjimku, je zvyklý klikat na „OK“ nebo „Ano“.

Je to rozdíl mezi Firefox 2 a Firefox 3 nebo IE6 a IE7. Pro lepší názornost je uveden příklad státní instituce, která má bezpečnou komunikaci téměř v popisu práce.

Neznámý certifikát a pozitivní model

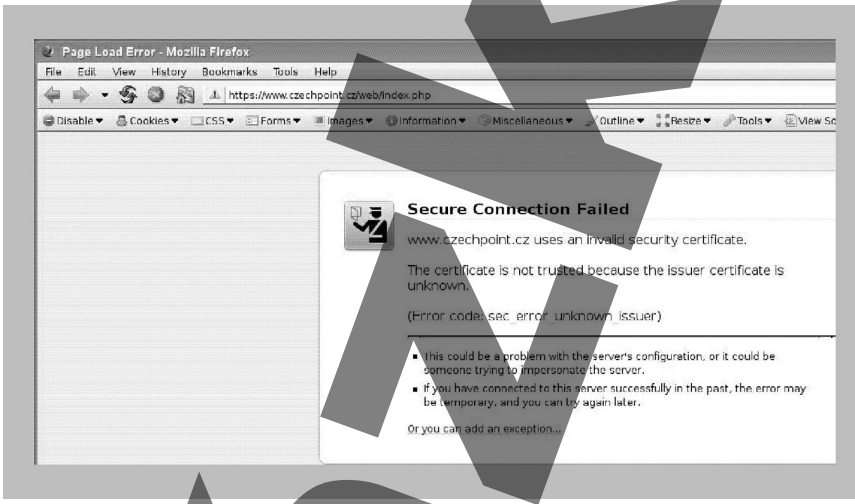


Reakce na certifikát podepsaný neznámou certifikační autoritou. Kdo by neznal PostSignum Public CA. Potvrdit výše uvedené ve FF2 šlo velmi snadno.



Reakce FF3 na stejnou stránku je naprosto rozdílná.

Neznámý certifikát a negativní model



Velmi se to podobá známému „na stránce vznikla chyba, prosím, obraťte se na svého administrátora“. Při používání pozitivního modelu byl uživatel sice varován, ale potřeboval přesně jeden klik, aby problém „vyřešil“. V případě negativního modelu potřebuje pokračovat v dalších krocích:

4 kliknutí navíc

1) Kliknout na odkaz „Or you can add an exeption“ nebo jeho jazykový ekvivalent.
2) Znovu kliknout na tlačítko „Add Exception“ (a možná si přečíst ještě jednou varování, co se vlastně děje).
3) Kliknout na další tlačítko „Get Certificate“ (stále není konec).
4) Teprve teď, když přeskočí možnost prohlédnout certifikát v detailu, může akceptovat výjimku (trvale nebo dočasně).

4 je více než 1. Jednoduchá logika a z pohledu bezpečnosti vcelku účinná, i když to někteří uživatelé nemusí chápat.

Neznámý certifikát a negativní model - pokračování

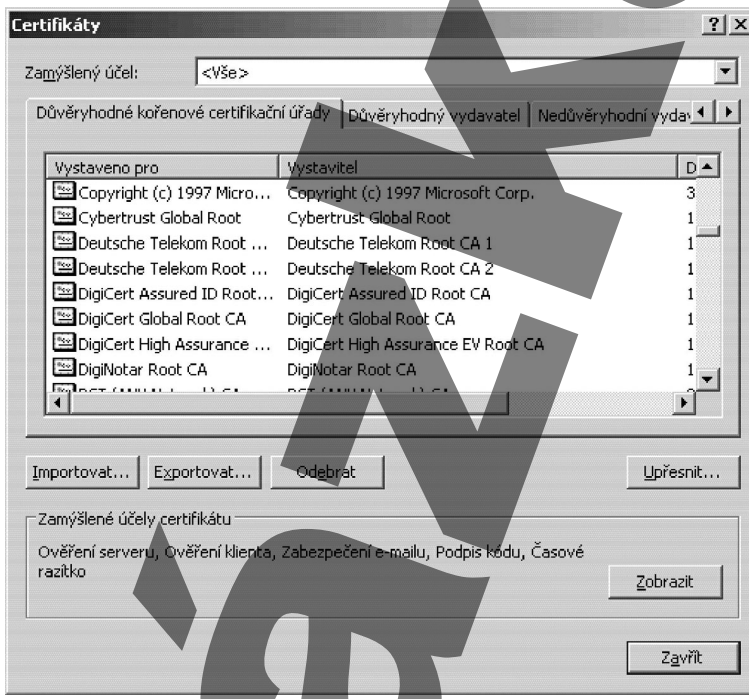
Způsob, jakým se přidávají výjimky do seznamu důvěryhodných certifikátů, není jediným způsobem, jakým se snaží výrobci prohlížečů upozorňovat uživatele, zda-li jsou nebo nejsou na zabezpečené stránce, např. žluté (Mozilla Firefox 2) nebo červené (Internet Explorer 7) zabarvení URL cesty; nebo také bíle (Mozilla Firefox 3 už zase barevný není).

Konečně se dostáváme k avizované novince, tedy dalšímu, způsobu, jak vyzrát nad SSL. Dosavadní starý svět důvěry v bezpečnost a použití SSL „skončil" zhruba v únoru 2009, kdy na BlackHat DC 2009 prezentoval Moxie Marlinspike svůj v podstatě jednoduchý nápad. Proč bojovat s HTTPS, když to jde i bez něj.

Podstata techniky SSLSTRIP

Studoval hlouběji používání SSL certifikátů a jejich řetězení. Běžnou odpovědí na otázku „proč a kdy považuje internetový prohlížeč SSL certifikát za důvěryhodný?" je, že se snaží vyhodnotit vybrané prvky certifikátu, jako např. název organizace a domény. Dále kontroluje pravost podpisu a platnost certifikátu. Nakonec ověřuje skutečnost, jestli zná certifikační autoritu, která certifikát vydala. Když ji má v interním seznamu, ověření končí, protože důvěryhodnost certifikátu serveru se rovná důvěryhodnosti certifikátu CA. Když nemá, jde po řetězci jednotlivých CA tak dlouho, dokud nenajde „kořen". Jedna zajímavá věc, která ale nemá s tímto útokem nic společného - už jste se někdy dívali na seznam CA, která vám dodává výrobce prohlížeče a kterým tedy „musíte" důvěřovat? Jsou jich desítky, možná stovky, závisí na použitém prohlížeči.

Kdy je vlastně certifikát důvěryhodný?

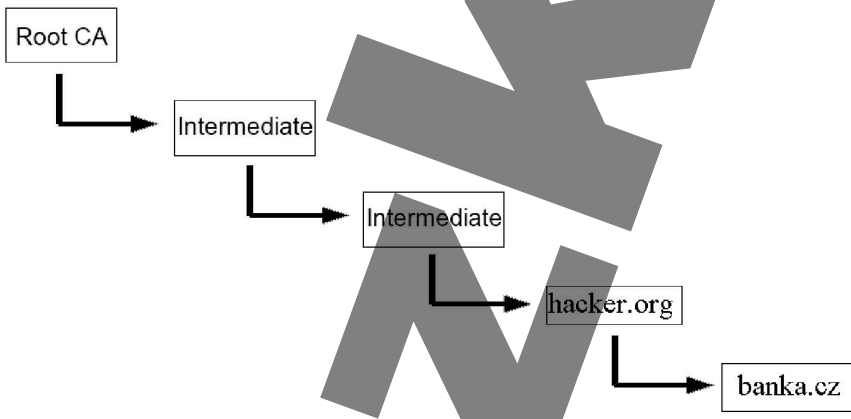


Kdo rozhoduje za nás?

Nemám nic proti Deutsche Telekom, ale Telefonice O2 nebo jiným operátorům také nedůvěřuji, tak proč za mě Microsoft rozhodl, že zrovna tento ano?

V čem je tedy uvedený příklad ověřování platnosti, resp. důvěryhodnosti špatný? Zaměřuje se jen na ověření podpisu a podmínkou je, aby byl „řetěz ověřování“ kompletní. Je to jeden z příkladů, jak špatně může dopadnout implementace matematicky velmi silného PKI do reálného světa. Co se stane, když se řetěz pro ověřování malinko prodlouží?

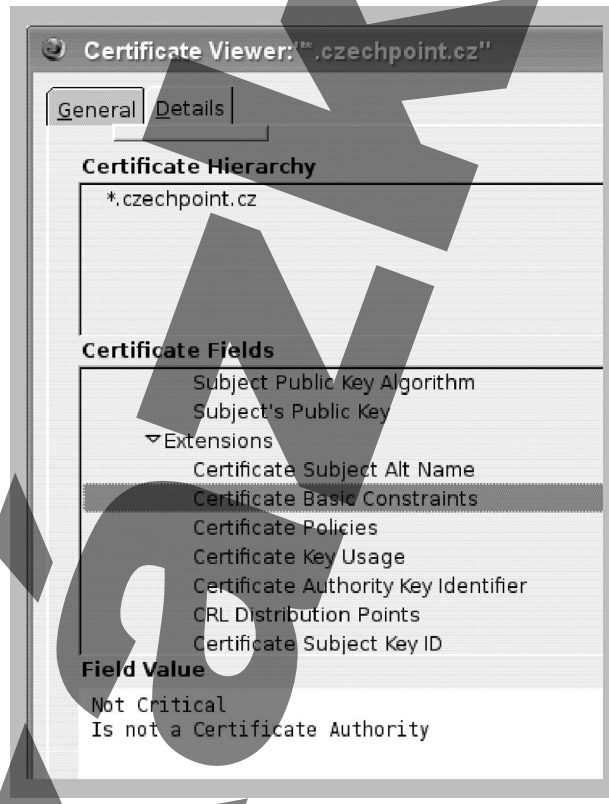
Řetězení certifikátů - naivní model pro ověření důvěryhodnosti



Nestane se vůbec nic. Vždyť přece platí, že:
1) všechny podpisy jsou platné,
2) certifikát není expirovaný,
3) posloupnost řetězce je netknutá,
4) kořenová CA je v seznamu naimportovaných certifikátů prohlížeče.

Jenže rozdíl je, že hacker.org právě vytvořil platný certifikát pro banka.cz - přitom není banka.cz. Kde se stala chyba? Většina autorit neřeší nastavování příznaku „basicConstraints: CA=FALSE“, resp. v grafickém znázornění.

Co dokáže jedno „bezvýznamné“ pole v X.509 certifikátu



Totéž platí pro většinu prohlížečů. IE, Firefox, Opera, Konqueror a další toto pole a jeho hodnotu prostě neřeší. Každý tak může vytvořit certifikát pro libovolnou další doménu.

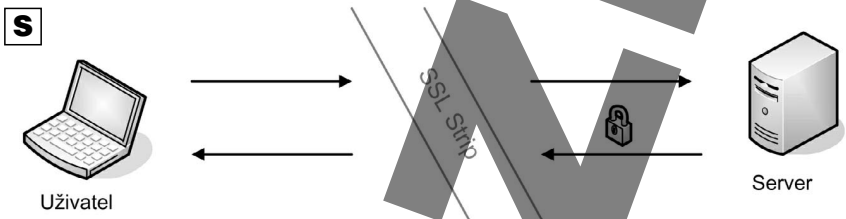
Sázka na lidské chování

Výše uvedený autor tedy přišel s nástrojem, který je známý jako SSLSTRIP, aby uvedenou zranitelnost demonstroval. Při jeho použití se navíc vychází z jedné mnohokrát ověřené pravdy. Kdy jste naposledy napsali

přímo do URL? Většina uživatelů internetu prostě jde na správnou doménu, která je implicitně publikována pomocí HTTP (tedy nešifrovaná) a odtud teprve pomocí tlačítka nebo odkazu „bezpečný vstup na účet“ nebo podobně přechází na zabezpečenou verzi. Případně počítá s tím, že dojde k automatickému přesměrování (známé HTTP kód 302). Většina stránek je napsána tak, že najetím na tlačítko není vidět správný odkaz na URL. Kód může být v AJAX nebo jiném rozšíření, které je dnes běžné. Uživatel tedy nemůže vědět, resp. vidět, kam jde. Na druhou stranu, i když je odkaz zobrazován, je to téměř nepodstatné, protože jej většina uživatelů stejně nesleduje.

Jak SSLSTRIP pracuje?

Jak SSLSTRIP funguje? Jedná se o transparentní HTTP proxy. V případě, že zachytí pokus o HTTPS komunikaci, nahradí směrem k uživateli HTTPS za HTTP, ale se serverem dále komunikuje pomocí HTTPS - tam to ani jinak nejde. Výsledkem je, že server nemůže rozpoznat rozdíl a uživatel pouze nevidí v URL cestě jedno písmenko.



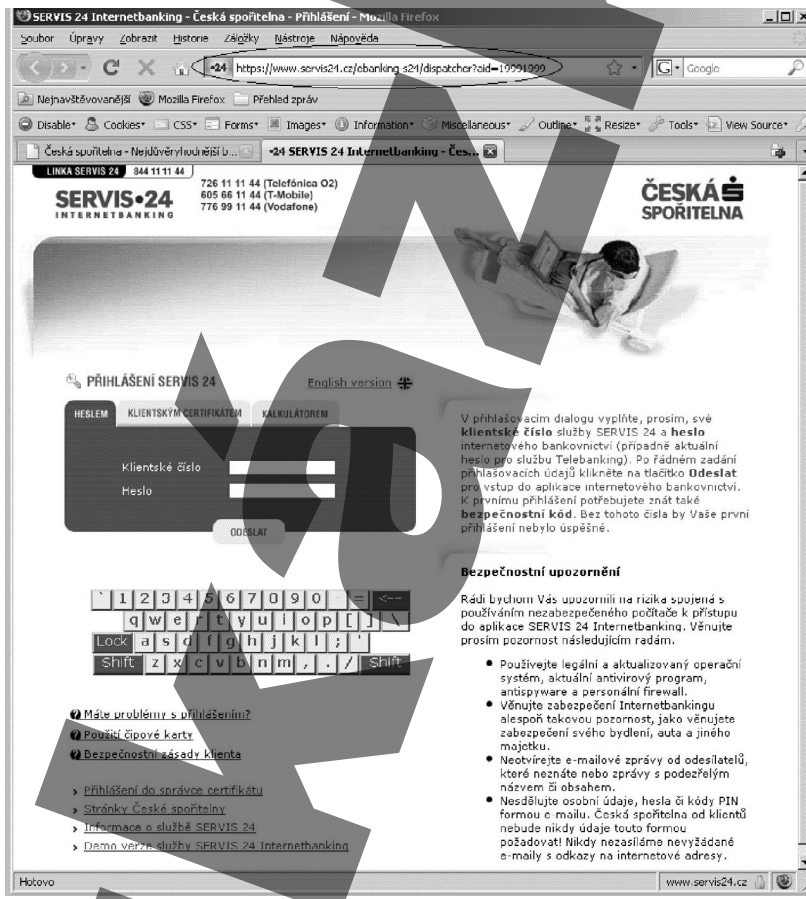
Jak to vypadá ve skutečnosti? Konfigurace je jednoduchá - manuál k programu to shrnuje ve 4 bodech (návod je jen pro Linux):

1) Povolte forwarding:
   echo „1“ > /proc/sys/net/ipv4/ip_forward
2) Nastavte si pravidla na iptables firewallu - všechen HTTP provoz bude přesměrován na zvolený port (default TCP/10000):
   iptables -t nat -A PREROUTING -p tcp —destination-port 80 -j REDIRECT —to-port <yourListenPort>
3) Spustte SSLSTRIP s možnostmi, jaké chcete:
4) Začněte provádět MITM - arpspoof:
   arpspoof -i <yourNetworkdDevice> -t <yourTarget> <theRoutersIpAddress>

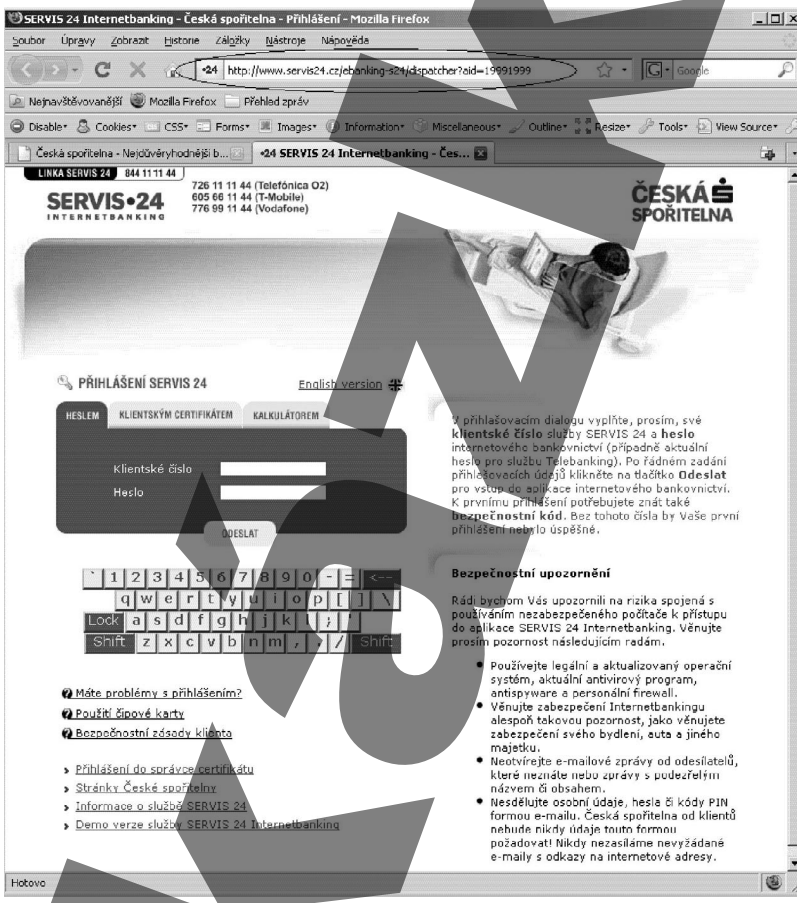
Za příklad jsem vybral SERVIS24 České spořitelny, ale nezáleží na tom - SSL si nevybírá. Je to jako hádanka pro děti, s tím rozdílem, že není 5, ale jen jeden rozdíl.

Když chybí jedno písmenko

Příklad jedné z českých bank - správně



Příklad jedné z českých bank - špatně



V logu proxy je pak zaznemenáno vše potřebné. SSLSTRIP je na začátku, není napsán jako dokonalý nástroj. Je dost věcí, které by v něm šly zlepšit. Dnešní stránky jsou plné složitého kódu, používají cookie a další metody, takže všechny stránky nemusí se současnou verzí tohoto nástroje fungovat správně. Ale

princip a možné dopady jsou naprosto zřejmé. Autor nástroje v rámci konference provedl pár testů. Za 24 hodin byly zachyceny desítky hesel k platebním branám, jako je paypal, desítky čísel kreditních karet a stovky dalších hesel k „zabezpečeným“ službám. Dále demonstroval možnosti, jak tento útok provádět i bez arpspoof, tedy MITM na jiné úrovni.

Kde program SSLSTRIP stáhnout?

Pro všechny, kteří si chtějí útok vyzkoušet, je zde URL, ze které je možné si potřebný nástroj a know-how stáhnout: http://www.thoughtcrime.org/software/sslstrip/.

Doporučení pro koncové uživatele

Doporučení pro běžné uživatele není v danou chvíli snadné. Snad jen ostražitost a zase ostražitost. Také není na škodu znát některé klíčové stránky a psát jejich URL přímo, včetně specifikace protokolu, nebo si stránky v tomto tvaru dát do záložek (bookmark).

V kombinaci s technikami založenými na MD5 kolizích, popsaných také v těchto aktualitách, můžeme doufat, že internet banking a jiné citlivé služby, kde se na bezpečnost klade důraz, nebude pouze adrenalinovým sportem pro běžné uživatele. Je totiž hezké, že vás každá banka varuje, kolik máte věnovat zabezpečení svého počítače a kolik toho musíte znát. Diskuse na téma „přenesení odpovědnosti“ velmi přesahuje rozsah této aktuality. Otázkou spíše je, jsou-li toho běžní uživatelé schopni.