Securitate web (Web Security)

Motivatie:

- Necesitatea transmiterii pe Internet a documentelor private.

Metode:

- Utilizarea criptografiei. **SSL** (***S**ecure **S**ockets **L**ayer*) este un acronim asociat unui protocol web dezvoltat de Netscape pentru a transmite documente private prin Internet.
- SSL utilizeaza un sistem criptografic cu doua chei pentru a cripta datele — una publica, cunoscuta de oricine, si una privata, secreta, cunoscuta numai de destinatar.
- Foarte raspandit, majoritatea browserelor Web suporta SSL si multe site-uri utilizeaza protocolul pentru a obtine informatii confidentiale, cum ar fi numere de card de credit. Prin conventie, URL-urile care au nevoie de o conexiune SSL incep cu *https:* in loc de *http:*.
- SSL creeaza o conexiune securizata intre un client si un server, peste care pot fi trimise datele in siguranta.

Securitate web (2)

- Un alt protocol de transmitere a datelor in siguranta este Secure HTTP (S-HTTP).
- S-HTTP este proiectat pentru a transmite mesaje individuale in siguranta.
- SSL si S-HTTP, prin urmare, poi fi percepute mai degraba ca tehnologii complementare decat concurente.
- Criptarea nu garanteaza securitatea web!

- Pentru a avea o conexiune SSL de incredere se pun intrebarile:
  - Ce trebuie sa cunoasca serverul despre client?
  - Ce trebuie sa cunoasca clientul despre server?

Caracterizare generala SSL

Cunostintele serverului despre clienti:

- Ce spune protocolul SSL serverului? Cu exceptia cazului in care sunt utilizate certificate client-side, nu spune absolut nimic!
- SSL realizeaza o legatura securizata (secure pipe). Cineva este la celalalt capat, dar nu se cunoaste cine.

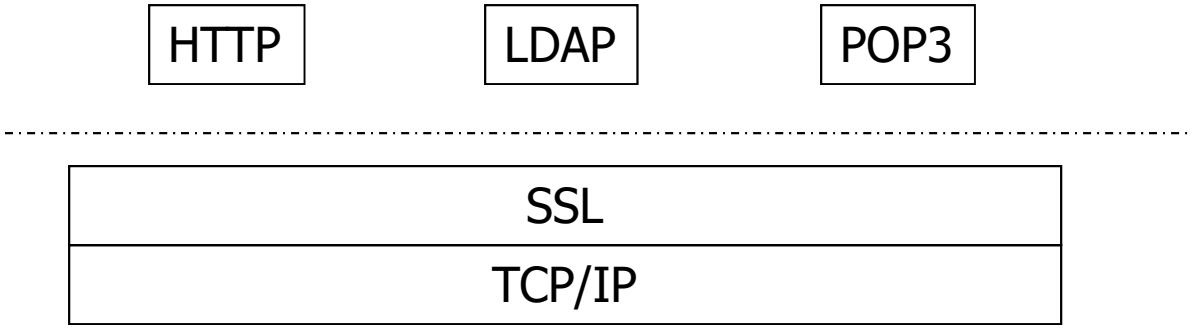
Ce se intampla?

- Teoretic, la platile cu card de credit se atasaza un ordin semnat
- Acelasi lucru incearca sa faca Netscape cu SSL: un mod de asociere a unui certificat client care este legat de cartea de credit si care nu are numarul cartii de credit in certificat

SSL

- Original, SSL a fost dezvoltat de Netscape, pentru a asigura securitatea datelor transportate si rutate de HTTP, LDAP, POP3.

- SSL utilizeaza TCP pentru a furniza o conexiune sigura si autentificata intre cele doua puncte ale retelei (clientul si serverul).



Localizare SSL

SSL - Obiective

Obiectivele principale ale protocolului SSL sunt:

- Autentificarea clientului si serverului unul fata de celalalt. SSL permite utilizarea tehnicilor standard de criptare (cu cheie publica) pentru a permite autentificarea celor doua parti. Desi cele mai frecvente aplicatii constau in autentificarea unui serviciu client pe baza unui certificat, SSL poate folosi aceste metode si pentru a autentifica clientul.

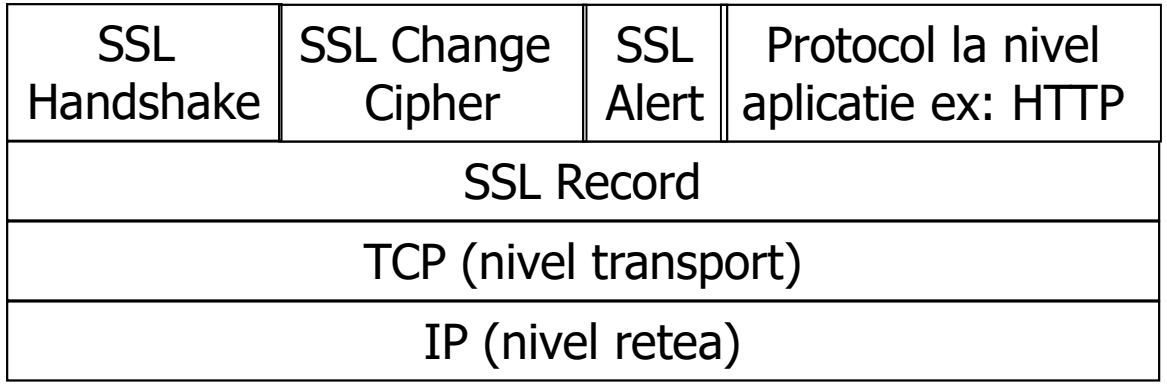
- Asigurarea integritatii datelor: in timpul unei sesiuni datele nu pot fi falsificate.

- Asigurarea confidentialitatii datelor: datele de transport dintre client si browser trebuie protejate de interceptare si citire. Acest lucru este necesar atat pentru datele asociate cu protocolul insusi (securizarea traficului in timpul negocierii) cat si pentru datele aplicatiei care sunt transmise in timpul sesiunii. Privit din acest punct de vedere SSL este mai mult un set de protocoale.

SSL - Arhitectura

SSL poate fi vazut ca fiind divizat pe 2 niveluri:

- Un protocol care asigura securitatea si integritatea datelor SSL Record Protocol (Protocolul de inregistrare SSL)
- Un protocolul care stabileste conexiunile SSL. La acest nivel se utilizeaza alte 3 protocoale: Protocolul SSL Handshake (Protocolul dialogului de confirmare), Protocolul SSL Change Cipher (Protocolul de schimbare cifru SSL) si Protocolul SSL Alert (Protocolul de alerta SSL).



Stiva SSL

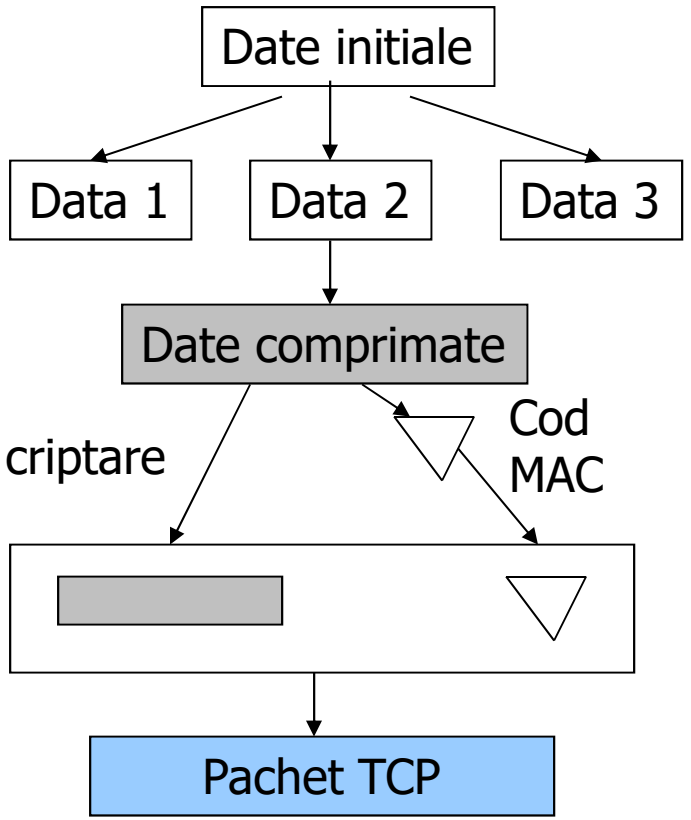
SSL – protocoale componente

- Scopul protocolului SSL Record este preluarea unui mesaj al aplicatiei si transmiterea lui in retea, prin protocolul TCP.
- SSL Record serveste ca baza pentru protocoalele de la nivelurile superioare.
- Protocolul SSL Record este responsabil de criptarea si integritatea datelor.
- SSL Record este folosit pentru incapsularea datelor trimise de alte protocoale.
- SSL Record este implicat in sarcinile de verificarea a datelor.
- Protocoalele de pe nivelul superior se ocupa de managementul sesiunilor, managementul parametrilor de criptare si de transferul mesajelor SSL intre client si server.

Protocolul SSL Record

Crearea pachetelor SSL Record:

- Fragmentare
- Incapsulare
- Creare obiect (record)
- Criptare obiect
- Transmitere



Calcul MAC

Fragmentarea se face in blocuri de cel mult 2^{14} octeti. Calcularea MAC este definita astfel:

Hash(MAC_write_secret+pad_2+hash(MAC_write_secret+pad_1+seq_num+ SSL.Compressed.type+SSL.Compressed.length+SSL.Compressed.fragment))

in care:

- \+ = operator de concatenare
- MAC_write_secret = cheie secreta simetrica
- hash = algoritm pentru hash: MD5 sau SHA-1
- pad_1= octetul 36H repetat de 48 de ori pentru MD5 sau de 40 de ori pentru SHA-1
- pad2 = octetul 5CH repetat de 48 de ori pentru MD5 sau de 40 de ori pentru SHA-1
- seq_num = numarul de secventa al mesajului
- SSL.Compressed.type = protocolul de nivel superior care proceseaza fragmentul
- SSL.Compressed.length = lungimea fragmentului
- SSL.Compressed.fragment= fragmentul dupa compresie (daca nu se foloseste compresia, textul clar)

Blocul care rezulta prin adaugarea MAC la fragmentul compresat este criptat cu un algoritm de criptare simetrica, de exemplu: IDEA, DES, 3DES, Fortezza.

Protocoalele Alert si Change Cipher

- Protocolul Alert este folosit de catre una dintre parti pentru a transporta mesajele in timpul sesiunii.

- Fiecare mesaj din protocolul Alert este alcatuit din 2 octeti:
  - Primul octet contine o valoare 1- “warning” sau 2- “fatal”, care determina importanta mesajului trimis. Trimiterea unui mesaj avand statutul de “fatal” de oricare dintre parti va duce la terminarea imediata a sesiuniii SSL.
  - Al doilea octet al mesajului contine unul din codurile de eroare, care poate aparea in timpul unei sesiuni de comunicare.

- Protocolul Change Cipher (Protocolul de schimbare de cifru) este folosit de catre una dintre parti pentru a transporta mesajele in timpul sesiunii
- Este cel mai simplu dintre protocoalele membre SSL.
- Contine un singur mesaj cu valoarea 1. Acest tip de mesaj este trimis de la client catre server si invers.
- Dupa schimbul de mesaje se stabileste o sesiune.
- Acest mesaj si orice alte mesaje sunt transferate folosind protocolul SSL Record.

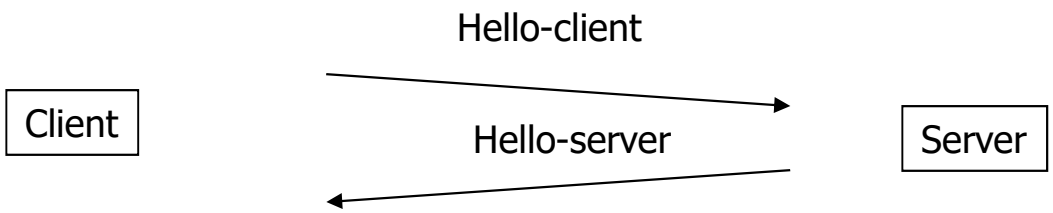
SSL - Protocolul Handshake

- Este cea mai complexa componenta a protocoalelor componente SSL.
- Este utilizat pentru a initia o sesiune intre un server si un client.
- Sunt negociate mai multe componente, cum ar fi algoritmul si cheile folosite pentru criptarea datelor.
- Folosind acest protocol este posibila autentificarea partilor una fata de cealalta si negocierea parametrilor sesiunii.

- Procesul de negociere se desfasoara pe parcursul a patru faze.
  - Faza 1: initierea unei conexiuni logice intre client si server, urmata de negocierea parametrilor
  - Faza 2: inceperea negocierii de autentificare a serverului de catre client
  - Faza 3: verificarea certificatului serverului si a celorlalti parametri transmisi de catre server
  - Faza 4: Confirmarea mesajului primit si verificarea datelor. La succes se incepe transmisia datelor intre client si server.

Schema protocol negociere (1)

Faza 1



Mesajul Hello-client.

Clientul trimite un mesaj de salut serverului care contine date cum ar fi:

- cea mai noua versiune SSL pe care o foloseste
- date aleatoare folosite pentru protectia cheii de sesiune;
- id-ul sesiunii : numarul de identificare al sesiunii. O valoare diferita de 0 indica faptul ca un client doreste sa actualizeze parametrii conexiunii sau sa stabileasca o noua conexiune a acestei sesiuni. O valoare egala cu 0 indica dorinta clientului de a stabili o noua conexiune
- lista cu cifruri : o lista de algoritmi de criptare si metode de schimb ale cheilor suportate de catre client.

Schema protocol negociere (2)

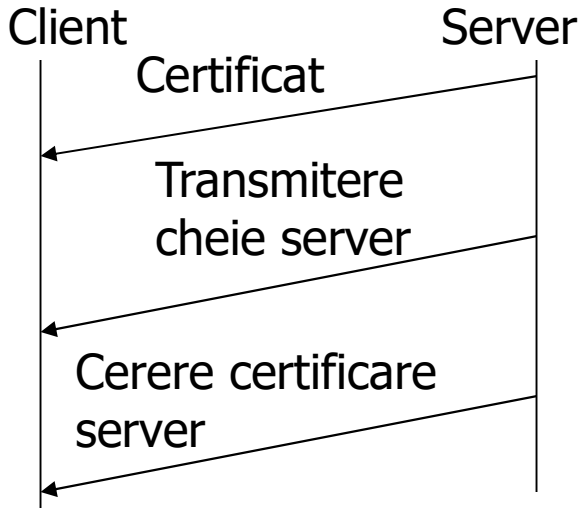
Mesajul Hello-server

Serverul, ca mesaj de raspuns trimite clientului, de asemenea, un mesaj de salut continand aceleasi campuri ca la mesajul primit:

- versiunea: cea mai veche versiune a protocolului SSL suportata de care server;
- date aleatoare (pentru protectia mesajului)
- id-ul sesiunii: daca campul clientului este diferit de 0 se pastreaza valoarea, altfel id-ul de sesiune al serverului contine o valoare pentru o sesiune noua
- Lista de cifruri: o multime de protocoale selectate de server dintre cele propuse de client. Primul element al acestui camp este metoda aleasa pentru schimbul cheilor de criptare, urmatorul element specifica algoritmul de criptare si functiile hash care vor fi folosite.

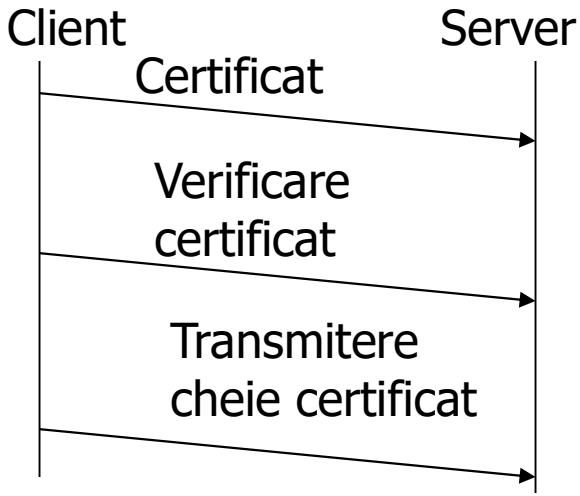
Schema protocol negociere (3)

- In faza 2: Serverul incepe urmatoare faza a negocierii trimitand certificatul sau pentru a fi autentificat de catre client. Acest pas nu este obligatoriu si poate fi omis daca metoda negociata pentru schimbul de chei nu cere transmiterea certificatului. Pasul final al fazei 2 este mesajul de raspuns al serverului. Dupa trimiterea acestui mesaj se asteapta un raspuns.



Schema protocol negociere (4)

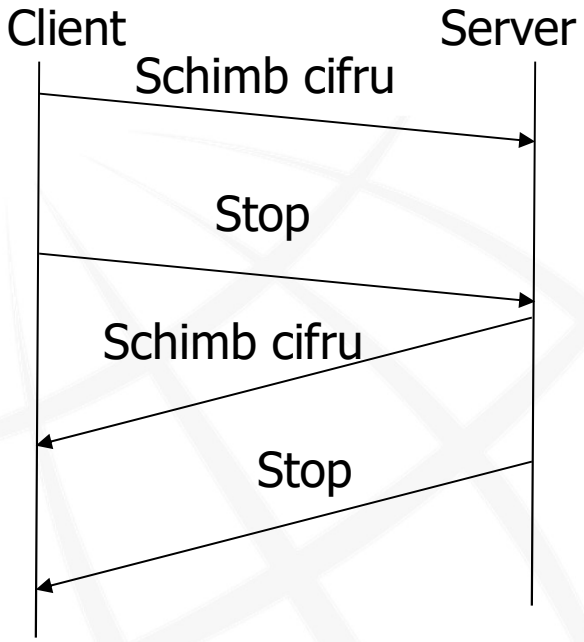
- In faza 3: La finalul fazei 2 clientul a primit mesajul de la server si clientul va trebui sa verifice certificatul serverului, precum si ceilalti parametri transmisi de server odata cu mesajul de salut.



Schema protocol negociere (5)

Faza 4, consta in confirmarea mesajului primit si verificarea datelor.

- Clientul trimite un mesaj cu lista de cifruri si apoi fixeaza parametrii si cheile algoritmilor, clientul trimite mesajul de inchidere care este protejat prin intermediul algoritmului si cheilor secrete. Astfel se confirma daca parametrii negociati si datele sunt corecte.
- Serverul, ca raspuns al mesajului client trimite aceeasi secventa de mesaje. Daca mesajul de final este corect atunci datele transmise, algoritmii negociati si cheia de sesiune sunt corecte. Acestea indica faptul ca sesiunea s-a terminat si este posibila transmiterea datelor aplicatei intre client si server prin SSL. In final, sesiunea TCP intre client si server este inchisa.



Comentarii SSL

- Tehnologie care permite securizarea unei semnaturi fara sa se utilizeze parola;
- Permite o identificare a site-ului la care se conecteaza fara ajutorul unei terte parti
- Toate acestea pot fi facute utilizand certificate SSL client
- Cand se viziteaza o pagina protejata SSL uzual browserul verifica identitatea site-ului prin verificarea certificatului. Site-ul indepartat este capabil sa verifice identitatea utilizand un certificat furnizat in prealabil
- Siteurile pot face ca browserul sa genereze perechea de chei
- Certificatul este pastrat in browser si browserul il va trimite la orice site securizat care cere acest lucru.
- Site-ul la randul sau iti poate verifica calitatea de proprietar al cheii private asociate prezentului certificat.

Web browser security

- Browserele web constituie una dintre cele mai utilizate aplicatii, ca urmare este extrem de importanta cunoasterea vulnerabilitatilor acestora.
- Statistica publicata pe 10 feb 2009. Dau mai jos rezultatele cu mediile pe zi realizata dupa o monitorizare in ultimele 365 de zile a vulnerabilitatilor principalelor browsere.
- Valorile “High severity” includ vulnerabilitatile raportate marcate ca “highly critical” si mai mult. Nivelul “Relative danger” a fost calculat prin adaugarea patratului nivelului la fiecare vulnerabilitate raportata (ne critica=1², extrem de critica=5²).

	Internet explorer	Firefox	Safari	Opera
Vulnerability reports	38	5	2	0
High severity vulnerability reports	1	0	0	0
Vulnerability issues	40	6	3	0
Relative danger	161	19	8	0

Caracteristici ale securitatii la browsere

Componente implicate in securitatea browserelor web:

- Interfata utilizator;
- Erori ale codului browser;
- Continutul activ al paginilor interpretat de catre browser.

Scopul atacatorilor:

- Furt de informatii personale, in special parole pentru informatii financiare;
- Crearea de “roboti” pe calculator ce pot fi utilizati pentru atacuri DoS, trimiterea de spam, gazduirea de phishing web site.

De ce sunt browserele nesigure?

- Sarcina lor este complexa
- Standardele in web sunt multe inca in curs de definire

Tipuri de continut activ

Pentru a creste functionalitatea si a imbunatati designul de foarte multe ori in siteuri web sunt incluse scripturi care executa programe in browser (client side). Din nefericire aceste scripturi sunt o cale prin care atacatorii incarca si executa cod rau intentionat pe masina utilizatorului.

- Javascript – uzual si usor de incorporat. Un atac uzual Javascript este redirectionarea utilizatorilor de la un site legitim la unul malitios care poate distribui virusi si colecta informatii personale.
- Controale Java si ActiveX – Spre deosebire de JavaScript, controalele Java si ActiveX sunt programe care se gasesc pe propriul calculator si pot fi descarcate in browser. Controalele Active X pot executa orice pe un calculator. Appleturile Java ruleaza in mediu mai restrans, dar daca nu este de incredere poate crea oportunitati atacatorilor.
- Paginile web contin programe sau referinte la acestea;
- Unele pagini invita utilizatorii: “please install this plug-in”, adica un program;
- Toate formele de continut activ sunt utilitarele obisnuite ale atacatorilor. Daca sunt dezactivate probabil site-ul nu poate fi folosit corect.

Javascript si AJAX

Javascript:

- Nu are legatura cu Java – initial s-a numit LiveScript;
- Sursa celor mai multe brese de securitate in Firefox si IE;
- Nu este clar modelul de securitate;
- Legaturi puternice intre scripturi de atac aflate pe siteuri diferite.

AJAX:

- Ajax – Asynchronous Javascript si HTML;
- Permite realizarea de pagini extrem de interactive;
- Implicatiile de securitate pentru client si server sunt inca destul de neclare, dar este clar ca se creeaza premise.

Active X

Active X – caracteristici:

- Cea mai mare eroare de design la continut activ: ruleaza cu permisiuni complete;
- Peste 1000 de controale
- Translatia: extrem de multe posibilitati de atac

Descarcarea controalelor active X:

- Orice pagina web poate descarca alte controale;
- Translatia: o pagina web poate descarca o bucata de cod arbitrara pentru a o rula la un utilizator;
- Singura protectie este o semnatura digitala in codul descarcat;
- Nu sunt restrictii pentru ceea ce poate face codul.

De ce ActiveX?

- Poate fi utilizat cu succes la Windows Update;
- Numai IE are ActiveX, este singura mare diferenta intre IE si Firefox.

Autentificarea continua

- Autentificarea initiala este realizata prin parola;
- Ea poate fi continuata prin:
  - Cookies: text special formatat trimis de server unui browser la fiecare conectare, browserul retrimite textul serverului pentru autentificare. Problema: Se colecteaza date despre calculatorul utilizatorului (adresa IP, domeniul la care este conectat, tip de browser, preferinte, profil ..)
  - Valori ascunse: informatiile protejate sunt incluse in pagina in campuri ascunse si trimise catre server printr-o metoda HTTP
- Ambele metode au limitari, problema fundamentala vine de la faptul ca ambele sunt trimise de clienti ce nu sunt de incredere

Clienti nesiguri (Untrusted)

- Un site web este interesat in identificarea utilizatorilor;
- Acest lucru stimuleaza utilizatorii in a trisa;
- Scopul site-ului web este de a face imposibila trisarea de catre client
- Un site web nu controleaza software-ul instalat la client

Protejarea informatiei de identificare

- Dupa ce utilizatorul se logheaza se creaza un sir de caractere care contine user-id;
- Criptarea este optionala, dar generarea MAC cu cheie cunoscuta doar de server este obligatorie si rezultatul este transmis clientului
- Cand sirul este trimis la server se valideaza MAC si se decripteaza pentru a vedea cine este clientul
- Doar serverul cunoaste aceste chei, deci numai serverul poate crea siruri de caractere protejate
- Optional: se poate include timestamp, adresa IP, etc

Valorile ascunse in URL sunt vizibile in fisierele log.

Cookies

Caracteristici:

- Folosit uzual
- Permite utilizatorilor sa reintre pe un site (sa se identifice)
- Session cookies – pastreaza informatiile numai pe durata utilizarii browser-ului. La inchiderea acestuia informatia este stearsa
- Persistent cookies – se pastreaza pe propriul calculator, poate fi ajustat intervalul de timp de pastrare a datelor (ex. email account – pagina personalizata la deschidere mail, pagina web favorita..).
- Stocati pe harddiscurile utilizatorului (pot fi copiati usor).

Protectia datelor de autentificare:

- La autentificarea continua datele de autentificare sunt de cele mai multe ori necriptate
- Cele mai multe site-uri nu doresc incarcarea suplimentara cu SSL pentru orice
- Credentialele sunt usor de furat
- Apararea uzuala: limitarea duratei de viata; reautentificarea inainte de face lucruri foarte sensibile din punctul de vedere al sigurantei.

Dezavantaje cookies (1)

Identificarea imprecisa:

- Daca pe un calculator sunt folosite mai multe browsere, fiecare va avea cookie-urile sale. Un cookie nu identifica persoana, ci combinatia cont de utilizator, calculator, browser.
- Nu se poate face diferenta dintre doi utilizatori care folosesc acelasi calculator si browser (daca nu folosesc conturi de utilizator locale diferite).

Interceptarea cookie:

- Sesiunile HTTP obisnuite sunt vizibile tuturor calculatoarelor din retea, care pot intercepta pachetele de date. Aceste cookie-uri nu pot contine deci informatii confidentiale. O solutie posibila este HTTPS.
- „Cross-site scripting” (XSS) permite trimiterea unui cookie altor servere, care in mod normal nu ar trebui sa-l primeasca. Browser-ele moderne permit executarea unor fragmente de cod primite de la server; daca cookie-urile pot fi accesate in timpul executiei, ele ar putea fi trimise altor servere decat cele „autorizate”. Criptarea nu ajuta impotriva acestui gen de atac. Metoda este de obicei folosita pe site-uri care permit utilizatorilor sa trimita continut HTML.
- Folosind un fragment de cod, un utilizator rau-voitor poate sa primeasca cookie-uri ale altor utilizatori si cu ajutorul lor se poate conecta ca sa pacaleasca serverul care crede ca altcineva s-a autentificat.

Dezavantaje cookies (2)

„Otravirea” cookie-urilor:

- „Otravirea” cookie-urilor: un atacator trimite unui server un cookie invalid (un cookie primit de la server, dar modificat).
- Cookie-urile ar trebui sa fie retinute si trimise serverului neschimbate; un atacator poate sa le modifice si apoi sa le trimita serverului
- Efect: Daca un cookie retine suma care trebuie platita pentru cumparaturi schimbarea acestei valori ar putea permite cumpararea unor bunuri la un pret mult mai mic.
- Cele mai multe site-uri retin cu ajutorul cookie-urilor doar un identificator de sesiune (un numar unic generat aleator) care identifica sesiunea. El reprezinta un index intr-o tabela interna a serverului, in care se retin valorile cu adevarat importante, cum ar fi pretul unor cumparaturi, etc.

Cookie inter-site:

- Fiecare site ar trebui sa aiba acces doar la cookie-urile proprii
- Erori in programarea browsere-lor pot duce la incalcarea acestei reguli
- Situatie similara trimiterii de cookie-uri modificate, dar nu este atacat serverul, ci un utilizator foloseste un browser vulnerabil.
- Prin ele se poate fura identificatorul de sesiune si atacatorul poate sa se „autentifice” in locul utilizatorului de buna credinta.

Alternative la cookie

In cazul in care browserul este configurat sa nu accepte cookie sunt posibile alternative:

- Adresa IP (sa permita incarcarea paginilor daca cererea vine de la o anumita adresa) este putin precisa, probleme cu NAT.
- URL (query string) ceva mai precisa se bazeaza pe introducerea informatiei in adresa URL. Partea denumita „query string” este folosita de obicei, dar se pot folosi si alte sectiuni. Mecanismul sesiunilor PHP foloseste aceasta metoda daca cookie-urile nu sunt activate.
- Autentificare HTTP Protocolul HTTP include mecanisme care permit accesarea unei pagini Web doar dupa furnizarea unui nume de utilizator si a unei parole, pe care browser-ul le retine si le transmite server-ului la fiecare cerere, fara sa fie introduse de fiecare data; din punctul acestuia de vedere, lucrurile se desfasoara ca si in cazul folosirii cookie-urilor. Transmiterea parolei (si chiar a numelui de utilizator) de fiecare data cand este ceruta o pagina este destul de nesigura: acest trafic poate fi interceptat.

Alternative la cookie (2)

- Obiecte Locale Adobe Flash. Daca un browser foloseste plugin-ul Adobe Flash Player functia de salvare locala a unor obiecte poate fi folosita intr-un mod foarte asemanator cookie-urilor. Aceasta poate fi o optiune atragatoare pentru dezvoltatorii de pagini Web, pentru ca majoritatea utilizatorilor folosesc acest plugin. In plus, configurarile sunt separate de cookie-uri, deci stocarea locala a obiectelor poate fi activata, iar cookie-urile dezactivate. Din pacate are alte probleme de securitate (chiar mai importante).

- Alte metode de a stoca date local Unele browsere suporta un mecanism prin care paginile Web isi pot stoca local unele date printr-un script. IE poate fi folosit pentru a stoca date intr-o pagina Web salvata pe hard local, intr-un document XML, sau in sectiunile „Favorites” ori „History” ale browser-ului.

- window.name Daca Javascript este activat, proprietatea name a obiectului window poate fi folosita pentru a stoca local date, deoarece aceasta ramane neschimbata la incarcarea succesiva a unor pagini Web. Metoda este mai putin cunoscuta si folosita.

Furt cookie

- Furt cookie. In general cookie-urile sunt trimise doar serverelor care le-au creat, pentru a le „fura” browser-ul trebuie sa le trimita altor servere.
- Scripturile JavaScript au de obicei acces la toate cookie-urile stocate de browser, si pot sa le trimita oriunde. De ex, cel care detine domeniul example.com poate scrie pe un alt site o legatura :
  - Click here!
- Cand un utilizator apasa pe acest link, browserul inlocuieste document.cookie cu lista cookie-urilor active pe acel site, care ajung astfel la serverul example.com, si asa are acces la cookie-urile utilizatorilor sitului pe care a pus link-ul.
- Atac imposibil de prevenit de browser, pentru ca scriptul vine chiar de la serverul care a creat cookie-urile, si totul pare a fi autorizat de acel server.
- Solutie: administratorii siturilor care permit utilizatorilor sa le modifice continutul sa implementeze metode pentru respingerea acestui gen de scripturi.
- Flagul HttpOnly nu le fac vizibile la programele client precum JavaScript. Ex:
  - Set-Cookie: RMID=732423sdfs73242; expires=Fri, 31-Dec-2010 23:59:59 GMT; path=/; domain=.example.net; HttpOnly
- Cand browserul primeste un astfel de cookie, el trebuie sa il foloseasca in mod obisnuit pentru schimburile HTTP urmatoare si sa nu-l faca vizibil scripturilor.

Server-side Security

Serverele sunt tinte atragatoare cel putin din urmatoarele motive:

- Stergerea informatiilor stocate sau degradarea lor
- Furt de date (de regula date sensibile cum sunt cele de conturi bancare);
- Distribuire “malware” la clienti care au incredere in acel site

Apararea standard:

- Verificarea tuturor intrarilor
- Constientizarea faptului ca nimic din ceea ce clientul trimite nu poate fi considerat de incredere
- Verificarea periodica a site-ului (audit)

Scripturi server-side

- Majoritatea site-urilor web utilizeaza scripturi server-side: CGI, ASP, PHP, Ruby, server-side include etc.;
- Fiecare din aceste scripturi este un serviciu de retea diferit;
- Pentru ca un site sa fie securizat toate scripturile lui trebuie sa fie securizate;
- Ce context de securitate produc scripturile la rulare? Cum face serverul protectia fisierelor sensibile in timpul executarii scripturilor cu functionare dubioasa?
- Ultima problema atacata este o problema particulara cu scripturi server side, cum ar fi PHP;
- Protectie partiala: utilizare de ceva similar cu suexec; rularea scripturilor ca un utilizator diferit si rularea proceselor web server ca utilizator web server implicit.

Injection Attacks; Curatire site

- Adesea, intrarile furnizate de utilizatori sunt utilizate pentru a construi un fisier sau pentru o interogare SQL.

- SQL injection attack este o tehnica care exploateaza vulnerabilitatile la nivel de database layer a unei aplicatii. Vulnerabilitatea se manifesta cand intrarile utilizator sunt incorect filtrate pentru caracterele escape.

- Utilizatorii rau intentionati pot trimite date false. (ex: in script care trimite email colecteaza username si executa: /usr/bin/sendmail username. Utilizatorul furnizeaza: foo; rm –rf / ca un username. Codul ce se executa este: /usr/bin/sendmail foo; rm –rf /

- Rezultat: stergere continut sistem de fisiere

Curatire site:

- Stergerea scripturilor implicite nesecurizate (ex: nph-test-cgi la Apache).