Varovanje domače strani

15.6.2001
Mario Medved
mario.medved@trinet.si

Uvod

Moja naloga je bila vzpostaviti sistem za urejanje zavarovanih domačih strani. Obstaja precej različnih sistemov zaščite, ki temeljijo predvsem na sposobnostih Web strežnikov, ki imajo vgrajene svoje sisteme za zaščito. Na žalost pa takim sistemom ne moremo preveč zaupati, saj je praksa pokazala, da so ti sistemi nezanesljivi. Ne mine dan, da se ne bi na internetu razkrila nova varnostna luknja Web strežnikov [1]. Do tega razvoja dogodkov je pripeljala predvsem silovita tekma med konkurenti (Netscape Web Server, Microsoft Information Server, Apache Web Server...), ki z nenehnimi nepreverjenimi novostmi in izboljšavami zasipavajo uporabnika, upajoč da se bodo ti odločili za njihov izdelek. Zato sem se odločil, da bomo naredili sistem, ki bo nadziral celoten pretok podatkov in ne bo ničesar prepustil naključju, kaj šele Microsoftu. V nadaljevanju bom na kratko opisal delovanje, uporabo in namestitev sistema, ki je v celoti izdelan z Delphi 5 programskim orodjem [2].

Kako deluje

Sistem je sestavljen iz dveh delov: iz strežnika, ki je nameščen na Web strežniku in urejevalnika (Editor.exe), ki pripravi podatke za naš strežnik. Strežnik je ISAPI (Internet Server Application Programming Interface) vmesnik [3], kar pomeni, da ga podpira večina Web strežnikov. Z lahkoto se prevede na klasičen CGI (Common Gateway Interface) vmesnik[3], ki lahko teče na poljubnem Web strežniku. Glavna prednost ISAPI arhitekture je v tem, da je za več uporabnikov, ki dostopajo do strežnika, v spominu naložena samo 1 kopija programa, pri CGI arhitekturi, se pa za vsakega uporabnika požene eden program.
Z urejevalnikom najprej izberemo že narejeno domačo stran, ki jo program pretvori in zašifrira v obliko, ki jo razume le naš strežnik (za šifriranje se uporablja IDEA[4]). Tukaj se pokaže že prva prednost našega sistema, ki ne zahteva posega v originalno, nezavarovano domačo stran. Večina zaščit tega seveda ne omogoča. Za tem določimo uporabnike, ki lahko dostopajo do naše strani, ter njihove pravice. Potem nam preostane samo še klik na gumb za prenos datotek do strežnika in naša zavarovana stran je že postavljena.
Se pravi, da imamo na strežniku pripravljene zašifrirane podatke, ki so postavljeni v zaščiteni ISAPI direktorij (Web strežniki imajo ponavadi ISAPI datoteke spravljene v zaščitenih direktorijih, do katerih zunanji obiskovalec nikoli ne bi smeli imeti dostop). Torej v primeru zelo malo verjetnega prevzema Web strežnika zašifrirani podatki nimajo nikakršne vrednosti za napadalce. To je tudi ogromna prednost mojega sistema, saj je ponavadi Web strežnik nekje na drugem koncu države ali sveta

in nikoli ne vemo kateri "administrator" bo brskal po naših podatkih, kljub temu da večina ponudnikov obljublja strogo zasebnost naših podatkov.
Vsak, ki želi z brskalnikom dostopati do naših podatkov, mora najprej našemu strežniku povedati uporabniško ime in geslo. Če je to pravilno potem nam strežnik pošlje našo glavno, začetno domačo stran. Vsi podatki, ki jih dobimo s strežnika so opremljeni z našim geslom in uro prijave (oba sta seveda zašifrirana), tako da nam ni treba za vsak podatek ali dostop znova vpisovati gesla. Seveda pa to geslo po petnajstih minutah nedejavnosti poteče. Torej pri pregledovanju zavarovane domače strani se za vsak html dokument, sliko ali datoteko avtomatsko opravlja identifikacija uporabnika, preverjanje njegovega nivoja dostopa, odšifriranje datoteke na strežniku in njen prenos do uporabnika. Edini način, kako dostopati do domače strani je preko strežnika, ki nas brez pravilnega gesla ne bo spustil nikamor.
Moj sistem za varovanje domačih strani je pilotski projekt, ki dobro deluje le na strežnikih, do katerih imamo neomejen dostop in na intranetih. Za optimalno delovanje na oddaljenih strežnikih, bi za shranjevanje podatkov moral uporabiti MySQL (SQL - Structured Query Language) ali XML (Extended Markup Language).

Navodilo za uporabo

Program za prevajanje in administracijo (Editor.exe) sem poskušal narediti čimbolj uporabniku prijazno. Na desni imamo administracijo uporabnikov, na levi pa urejanje datotek. Na preprost način dodajamo, spreminajmo, brišemo uporabniške skupine, ki jim lahko določimo ime in nivo dostopa. Podobno delamo z uporabniki, ki jih po želji lahko uvrstimo v skupine ali pa tudi ne.

1. Za urejanje datotek moramo najprej imeti narejeno domačo stran.
2. Stran, ki želimo da se prva prikaže uporabniku po uspešni prijavi, se mora imenovati "Logon.htm", stran, ki jo želimo uporabiti za prikaz ob napačnem geslu "Error.htm" in stran, ki se bo prikazala ob prijavi pa "Login.htm". Vse strani lahko poljubno spreminjamo, le pri "Login.htm" je treba paziti na imena spremenljivk, ki se morajo ujemati z imeni v privzeti datoteki "Login.htm".
3. V nastavitvah "Settings" moramo nastaviti lokacijo naše domače strani (to naredimo tako, da poiščemo našo "Logon.html" datoteko) in lokacijo našega serverja (se pravi lokacija našega strežnika, ki se imenuje "Iserver.dll"). Direktorij v katerem se nahaja "Iserver.dll" mora biti izbran kot ISAPI direktorij pri nastavitvah našega Web strežnika. Za nastavitev Web strežnika glej naslednje poglavje.
4. Potem nam preostane samo še klik na gumb za pripravo strani "Encode" ter gumb "Transfer" za prenos datotek na ISAPI direktorij, kjer so dostopne Web strežniku in tudi našemu "Iserver.dll" strežniku. Če Web strežnika nimamo na našem računalniku (ali v lokalnem omrežju), potem moramo datoteke, ki se nahajajo v istem direktoriju kot naš program (v poddirektoriju "Output"), prek FTP-ja prenesti na strežnikov ISAPI direktorij.
5. Za dostop do naše zavarovane strani poženemo katerikoli internetni brskalnik in vtipkamo naslov našega strežnika + ISAPI direktorij + "Iserver.dll". Potem nam bo "Iserver.dll" poslal zahtevo po prijavi na domačo stran.

6. V urejevalniku lahko posameznim datotekam določimo še nivo dostopa (kateri uporabniki smejo dostopati do nje). Do datoteke določenega nivoja lahko dostopajo samo uporabniki z manjšim ali enakim nivojem. Se pravi, da imamo na prvem nivoju administratorje (nivo 0), na zadnjem pa nepriljubljene profesorje (nivo 9). Datoteke imajo nivo označen s pomočjo končnice, kar pomeni, da imamo lahko na strežniku isto datoteko z več nivoji (saj imajo različne končnice), torej imamo lahko za uporabnike različnih nivojev različne podatke, strani ali slike.

Nastavitev

1. Najprej je treba naložiti BDE (Borland Database Engine), ter v njem definirati podatkovno bazo z imenom "dbWeb", katere parameter "PATH" kaže na lokacijo naših podatkovnih datotek, ki se nahajajo v istem direktoriju kot urejevalnik (Editor.exe). V teh Paradoxovih datotekah so shranjeni podatki o uporabnikih in uporabniških skupinah. Te datoteke zaradi lažje uporabe še niso zašifrirane, toda podpora za to je že vgrajena.
2. Potem namestimo katerikoli Web strežnik (priporočam OmniHTTPd Web Server[5]), ki podpira ISAPI.
3. Na Web strežniku nastavimo ISAPI direktorij, v katerega posnamemo naš strežnik, se pravi datoteko "iserver.dll".
4. Potem lahko normalno zaženemo naš urejevalnik, kjer nastavimo lokacije datotek in začnemo delati.

POZOR: Vse nastavitve morajo biti opravljene z Adminstratorskimi pravicami, saj le tako lahko BDE in Web strežnik shranita podatke v Windowsov register.

Zaključek

Sistem se dobro obnese v intranetih in lokalnih omrežjih, saj omogoča zelo hitro in preprosto vzpostavitev portala do zavarovanih podatkov. Mogoče bo postal predstavitveni del večjega PACS (Picture Archiving and Communication System) sistema, ki se bo razvijal za Klinični Center. Možni napadi na sistem so zaradi arhitekture sistema mogoči le preko prevzema nadzora nad Web strežnikom in dostopa do "iserver.dll" datoteke, v kateri se seveda nahaja algoritem in geslo, ki povesta kako odšifrirati podatkovne datoteke. Uporabo tako imenovanega "reverse engineering-a" lahko zelo otežimo s pomočjo posebnih programov (packers, compressors) za kodiranje datotek[6], preprečiti pa ga seveda ne moremo. Drugi način napada je seveda prestrezanje gesla uporabnika, kar lahko otežimo z uporabo SSL-ja (Secured Socket Layer).

V strežnik "Iserver.dll" je mogoče vgraditi tudi opazovanje prometa in izvajanje posebnih ukazov (strežnik nam lahko pošlje e-mail ob določenem dogodku in podobno...).

Varnost podatkov na internetu je popolnoma v domeni velikih podjetij, ki postajajo vse močnejša. Oracle, Microsoft in podobni samo povečujejo svoj tržni delež in nas s tržnimi in malo manj tržnimi pristopi silijo v nakup njihovih izdelkov. Lep primer je Microsoftova ".NET" filozofija, ki bo združila vse Microsoftove izdelke v

zaokroženo celoto in prisilila uporabnike, da ne bodo niti vedeli za obstoj drugih ponudnikov[7]. Ta trend je že mogoče zaznati pri velikih slovenskih podjetjih, ki ne želijo slišati za nič drugega kot Microsoft, kljub temu da je večina njihovih izdelkov neprimerno slabših od konkurence. Za prihodnost lahko upam, samo da bodo uporabniki dovolj obveščeni in se ne bodo pustili zavajati korporacijam.

Reference

[1] Internet Security
- http://www.cert.org
- http://www.loc.gov/global/internet/security.html

[2] Delphi
- http://www.borland.com/delphi/
- http://www.delphi-jedi.org

[3] ISAPI & CGI
- http://msdn.microsoft.com
- http://hoohoo.ncsa.uiuc.edu/cgi/

[4] IDEA
- http://community.roxen.com/developers/idocs/rfc/rfc3058.html
- http://axion.physics.ubc.ca/crypt.html

[5] OmniHTTPd Web Server
- http://www.omnicron.ca/httpd

[6] Packers / Compressors
- http://www.mesa-sys.com/~codomain/pack.htm

[7] Market leaders
- http://www.microsoft.com
- http://www.oracle.com

Dodatek

Urejevalnik v vsej svoji farbovitosti.