BAKI UNİVERSİTETİNİN XƏBƏRLƏRİ
№1 Fizika-riyaziyyat elmləri seriyası 2007

İNFORMATİKA

WEB SERVERLƏRİN MÜHAFİZƏSİ ÜÇÜN PROKSİ PROQRAMININ YARADILMASI

R.F.FƏRƏCULLAYEV
İnformasiya Texnologiyaları İnstitutu

Məqalədə web-serverlərin mühafizəsi üçün xüsusi web-proksi proqramının yaradılmasına baxılmışdır. Əksər hallarda HTTP (hipermətnlərin göndərilməsi protokolu) protokolu üzrə həyata keçirilən hücumları digər səviyyələrdə tanımaq mümkün olmur. Belə hücumların qarşısını almaq üçün yaradılmış hazır proqram məhsulları – WAF (WEB proqram fayrvolları) nəzərdən keçirilmişdir. Məqalədə, həmçinin HTTP sorğularının strukturu araşdırılmışdır. HTTP hücumlarının qarşısının alınması üçün xüsusi proksi proqramının yaradılması və trafikə nəzarət edilməsi təklif olunmuşdur. Proksi proqramının iş prinsipi təbii alqoritmik dildə göstərilmişdir.

Giriş. Web-serverlərin təhlükəsizliyinin təmin olunması üçün müəyyən metodlar, proqram və aparat vasitələri işlənib hazırlanmışdır. Web-serverlərin təhlükəsizliyi bir neçə səviyyəyə ayrılır [1]:

- Web-serverin fiziki təhlükəsizliyi.

İstənilən şəxsin web-serverin arxasında əyləşib ona istədiyi müdaxilələri etməməsi üçün web-serverin təhlükəsizliyi fiziki səviyyədə təmin olunmalıdır.

- Web-serverin əməliyyat sisteminin təhlükəsizliyi.

Web-serverin təhlükəsizliyi, həmçinin onun istifadə etdiyi əməliyyat sistemindən asılıdır. Web-server üçün təhlükəsiz fəaliyyət göstərə bilən şəbəkə əməliyyat sistemi seçilməlidir.

- Web-proqramların və resursların təhlükəsizliyi.

Web-serverin təhlükəsizliyinin sonuncu – üçüncü səviyyəsi isə web-serverdə yerləşdiriləcək web-proqramların təhlükəsizliyidir. Web-serverdə yerləşdiriləcək web-proqramların açacağı təhlükəsizlik boşluqlarını əvvəlcədən demək mümkün deyil. Əməliyyat sisteminin sazlanması ilə web-proqramların yarada biləcəyi təhlükələrə qarşı yalnız müəyyən tədbirlər görmək olar. Bu, web-serverdə yer-

ləşdirilmiş bütün proqramların təhlükəsizliyinin təmin edilməsi demək deyil.

Web-proqramın fəaliyyətini dayandırmaq üçün bədniyyətlilər müxtəlif səviyyəli hücumlardan istifadə edirlər: DoS, DDoS, ARP Spoofing, Injection attacks, Hidden field consumptions və s. Bu hücumları da müxtəlif səviyyələrə bölmək olar:

- Daxili hücumlar.

ARP spoofing, viruslar, troyanlar və s.

- Paket səviyyəsində edilən hücumlar

Məsələn: yarımçıq açılmış TCP/IP sessiyası, Amplification, SYN sel hücumları və s.

- Sistemə soxulmalar.

Əməliyyat sisteminin və yaxud da bu və ya digər protokolun məntiqi səhvlərindən istifadə edərək, trafiki izləməklə əldə edilən informasiyadan istifadə edərək sistemə icazəsiz soxulma cəhdləri.

- HTTP səviyyəsində edilən hücumlar.

Bu sorğular isə adi HTTP sorğusu şəklində formalaşdırılırlar. Onların emal edilməsi zamanı bədniyyətlilər web-serverin fayl sisteminə, verilənlər bazasına və digər resurslara müraciət hüququ qazanırlar.

Web-serverlərin fiziki səviyyədə, əməliyyat sistemi səviyyəsində, kanal və protokol səviyyələrində təhlükəsizliyini təmin etmək üçün hazır metod və vasitələr var. Sadaladığımız səviyyələrdən də görünür ki, bu və ya digər proqramçı tərəfindən yaradılacaq web-proqramda açılacaq təhlükəsizlik boşluqlarını əvvəlcədən demək mümkün deyil. Hər bir web-proqramın yaradılması zamanı bütün təhlükələrə qarşı eyni tədbirləri görmək vaxt aparan və yorucu işdir. Buna görə də web-serverlərin təhlükəsizliyini təmin etmək üçün həmin serverə ümumilikdə web-proqram fayrvolu (Web Application Firewall – WAF) [2] tətbiq olunur.

Fayrvol proqramları web-serverdə işləyərək onun mərkəzi sistem resursları hesabına fəaliyyət göstərir və istifadəçi sorğularını web-server proqramından əvvəl emal edərək müəyyən hücumların qarşısını almağa kömək edir.

Bununla da eyni bir web-serverdə yerləşdirilmiş web-proqramı HTTP protokolu səviyyəsində gələn hücumlardan mühafizə etmək mümkün olur. Belə WAF-lara misal olaraq Breach Security kompaniyasının BreachGate WebDefend 5 məhsulunu, F5 Networks kompaniyasının Big-IP Application Security Module 9.2.2 məhsulunu, Imperva kompaniyasının SecureSphere Web Application Firewall 4.2 məhsulunu, Net Continium kompaniyasının NC-1100 AF 5.1 məhsulunu misal göstərmək olar.

Məsələnin qoyuluşu. İstifadəçilər web səhifələri əldə etmək üçün web-serverə sorğu göndərirlər. Bu sorğular sadə HTTP protokolu əsasında sadə mətn şəklində göndərilir. HTTP sorğular bir neçə hissədən ibarət olur: Başlıqlar (headers), sorğu sətri (query string) və göndərmə parametrləri (post parameters). Web-server isə bu mətni analiz edərək

onları xüsusi dəyişənlər şəklində web-proqrama təqdim edir. İstifadəçi sorğusunu əldə etmiş web-proqram isə bu informasiya əsasında istifadəçinin tələbini müəyyənləşdirir və istifadəçiyə onun tələb etdiyi HTML səhifəni təqdim edir.

Bədniyyətli istifadəçilər hücumları təşkil etmək üçün xüsusi HTTP sorğular hazırlayaraq web-serverə göndərirlər. Onların sorğusu bütün səviyyələrdə təhlükəsizlik proqramları və qurğuları tərəfindən heç bir şübhə oyatmadan web-proqrama qədər gəlib çıxır. Bu parametrləri əldə etmiş web-proqram əvvəlcədən nəzərdə tutulmamış işləri yerinə yetirir. Məsələn: müəyyən parametrin həqiqi tip dəyişən alması gözləndiyi halda, ona sətr tipli qiymət verilir və s. Bədniyyətlilər HTTP sorğuların bu şəkildə süni təşkil edilməsi ilə web serverin fayl sisteminin və ya verilənlər bazasının strukturunu öyrənirlər. Bundan sonra da onlar, məhz elə bu cür HTTP sorğuların köməkliyi ilə web-serverdə müəyyən dəyişikliklər aparır, informasiya oğruluğuna və ya məhvinə səbəb olurlar.

Təhlükəsizliyin digər səviyyələrində heç bir şühbə doğurmayan belə HTTP sorğuların qarşısını almaq üçün web-serverlərdə xüsusi web-proqram fayrvolu (Web Application Firewall – WAF) quraşdırılır. WAF proqram məhsulları web-serverdə quraşdırılır və buna görə də serverin sistem resurslarının daha çox istehlak olunmasına gətirib çıxarır. Bu isə web-serverdə "xidmətdən imtina" halına zəmin yaradır. Bundan başqa, bədniyyətli istifadəçilər web-serverə birbaşa müraciət etmək imkanına malik olduqları üçün onlar həmin serverə digər səviyyədə hücumlar təşkil edə bilər, bir neçə hücum növünü birləşdirərək web serverə "ağır zərbə" vura bilərlər. Bəzi WAF-lar isə serverin cavabvermə zamanını gecikdirməmək üçün həmin web-serverə güzgü portlar vasitəsilə qoşularaq bu serverə gələn trafiki analiz edir və bu səbəbdən də web-serverə edilən digər səviyyəli hücumlara reaksiya verə bilmirlər.

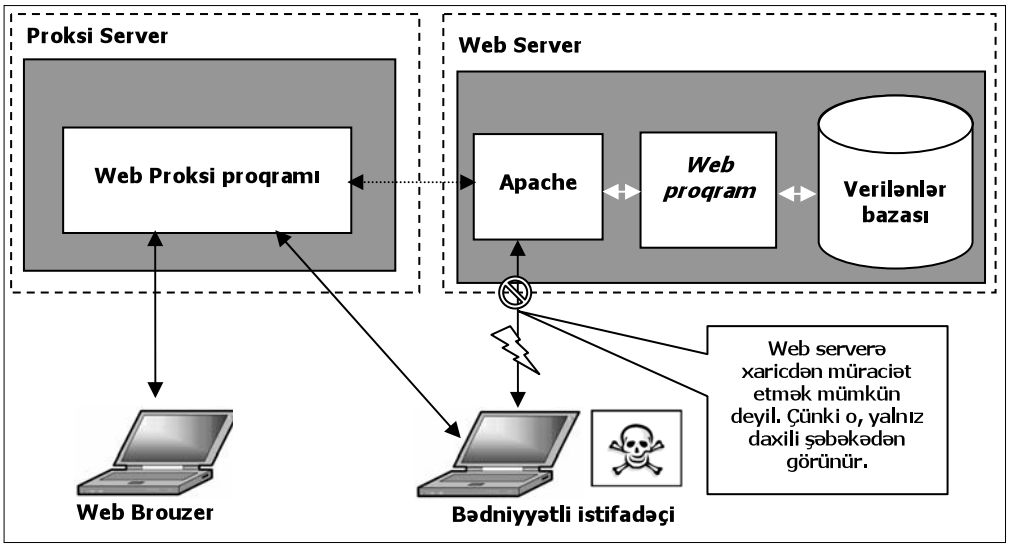
Bunları əsas gətirərək web-serverin təhlükəsizliyinin təmin olunması üçün Proksi [3] texnologiyasından istifadə etmək təklif olunur. Proksilər bir proqram məhsulu və ya qurğu olaraq serverdə quraşdırılır və internet ilə server arasında vasitəçi rolunu oynayır.

Bizim baxdığımız halda Proksi – proqram məhsulu şəklində hazırlanmalı və xüsusi bir serverdə quraşdırılmalıdır.

Proksi proqramı xüsusi serverdə yerləşdirildiyindən, o, gələn sorğuları emal etmək üçün web-serverin sistem resurslarından istifadə etməyəcək. Bununla da web-serverin DoS və DDoS kimi hücumlarına qarşı dözümlülüyü artacaq. Proksi proqramının yerləşdirildiyi xüsusi serveri şərti olaraq Proksi server adlandıraq.

Proksi proqramının və web-serverin bir-birindən ayrı yerləşdirilməsinin digər bir üstün cəhəti isə web-serverə yalnız HTTP protokolu əsasında sorğuların yönələcəyinə əmin olmaqdan ibarətdir. Bu da bədniyyətlilərin birbaşa web-serverə və ya verilənlər bazasına müraciət etmək imkanını aradan qaldırır.

Həll üsulu. Web-proqramlar və VBİS internetdən müraciət edilməsi mümkün olmayan, web-serverə paralel serverlərdə yerləşdirilir. Web-proksi proqramı isə web-server adı ilə tanınan və internetdən görünən serverə quraşdırılır. İstifadəçilər tərəfindən göndərilən sorğuları ilk olaraq web-proksi proqramı əldə edir, bu sorğunun hücum olub olmamasını xüsusi alqoritmlər əsasında qiymətləndirir və sorğu normal hesab edilirsə, onu həqiqi web-serverə göndərir. Web-serverdə web-proqramın yaratdığı HTML sənədi əldə etdikdən sonra isə onu geriyə - istifadəçiyə qaytarır.

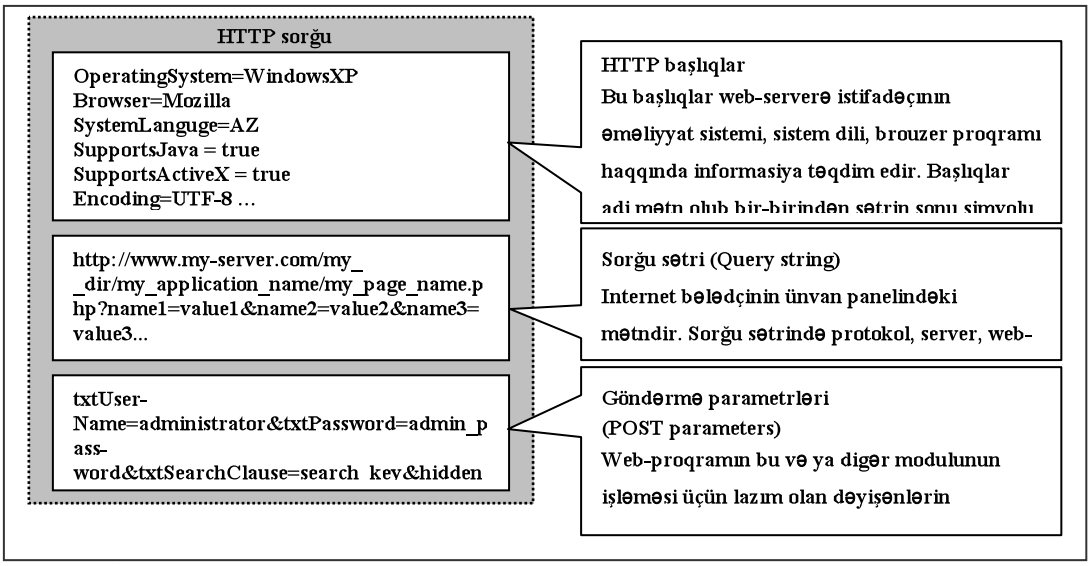


Şəkil 1. Web-serverin web-proksilərlə mühafizə olunması.

Şəkildən göründüyü kimi, adi İnternet istifadəçilərinin və bədniyyətli istifadəçilərin web-serverə daxilolma nöqtəsi eynidir. Onlar yalnız proksi serverdə yerləşdirilmiş web-proksi proqramına müraciət edərək səhifəni tələb edə bilərlər. Bununla da web-serverə yetirilə biləcək digər təhlükələr aradan qaldırılmış oldu. Bundan sonra isə istifadəçilər web-serverə yalnız HTTP sorğularını yönəldə bilərlər.

HTTP sorğuların [4] yarada biləcəyi təhlükəni daha yaxşı başa düşmək üçün HTTP sorğuların strukturuna nəzər salaq (şək. 2).

Şəkildən göründüyü kimi, HTTP sorğular yalnız mətn informasiyasından ibarətdirlər. Hətta böyükhəcmli faylların serverə göndərilməsi zamanı onlar 64 əsaslı mətnlərə çevrilərək göndərmə parametri şəklində server göndərilir. Buna görə də, web-proksi proqramı vasitəsilə istifadəçilərdən gələn HTTP sorğuları mətn informasiyası kimi analiz etmək mümkündür. Hər bir HTTP hücumun tanınması üçün müxtəlif işarələr (signature) mövcuddur və buna görə də web proksi proqramında hər bit HTTP hücumun tanınması və onun yarada biləcəyi təhlükənin qiymətləndirilməsi üçün xüsusi modullar yaradılmalıdır.



Şəkil 2. HTTP sorğunun strukturu.

Web-proksinin iş prinsipi isə aşağıdakı kimidir:

1. Web-proksi web-serverdə yerləşdirilir və əvvəlcədən təyin olunmuş portda – adətən 80-ci portda HTTP sorğulara qulaq asır.
2. İstifadəçilərdən sorğunu qəbul etdiyi zaman sessiyanı gözləmə rejimində saxlayaraq sorğunu analiz etmək üçün alt modullara ötürür.
3. Sorğu alt modullar tərəfindən normal qiymətləndirilərsə, web-proqramın yerləşdirildiyi serverə yeni bir HTTP sessiya açılır və sorğu web-proqrama göndərilir və cavab gözləmək üçün bu sessiya da açıq saxlanılır.
4. Web-proqram HTTP sorğunu qəbul edən kimi onu emal edir və nəticəni elə həmin sessiya vasitəsilə web-proksiyə qaytarır.
5. Web-proksi açıq saxladığı birinci sessiya vasitəsilə istifadəçiyə web-proqramın yaratdığı HTML səhifəni göndərir.

Təhlükəsizliyin daha da artırılması və DoS hücumlarının zəiflədilməsi üçün web-serverə yeni bir sessiya açıldıqda xüsusi bir taymer ilə web-serverin cavab verəcəyi vaxtı da ölçmək lazımdır. Əgər sorğunun emalı təyin olunmuş vaxtdan çox davam edərsə, gələcəkdə belə sorğuları web-serverə buraxmayaraq, serverin "xidmətdən imtina" hallarına qarşı dözümlülüyünü artırmaq olar.

Nəticə. Məqalədə HTTP sorğuların strukturu araşdırıldı və bu sorğular vasitəsilə bədniyyətlilərin web-serverə vura biləcəyi təhlükələrə nəzər salındı. HTTP hücumlarla mübarizə aparmaq üçün mövcud WAF proqramlar nəzərdən keçirildi və onların mənfi cəhətləri göstərildi.

HTTP sorğuların qarşısını almaq üçün yeni model – proksi modeli təklif olundu və bu modelin iş prinsipi təbii alqoritmik dildə göstərildi.

ƏDƏBIYYAT

1. Kevin Borders, Atul Prakash. Web Tap: Detecting Covert Web Traffic. ISBN:1-58113-961-6. Washington DC, USA © 2004
2. I. Bar-Gad. Web application firewalls protect data. http://www.networkworld.com/news/tech/2002/0603tech.html © March 2002.
3. Yen-Jen Chang, Yung-Ching Weng, Feipei Lai. Enhanced Object Management for High Performance Web Proxies. ACM Press New York, USA © 2004
4. http://methods.ringing.org/query.html. 16 November 2005.

СОЗДАНИЕ ПРОГРАММЫ ПРОКСИ ДЛЯ ЗАЩИТЫ ВЕБ СЕРВЕРОВ

Р.Ф.ФАРАДЖУЛЛАЕВ

РЕЗЮМЕ

Рассматриваются создание веб прокси для защиты веб сервера. В основном случае не возможно уловить атаки HTTP (Протокол передачи гипертекстов) на других уровнях. Рассматриваются приложения WAF (Файрволы для веб приложений) – предназначенные для ликвидации таких родов атак. В статье рассматриваются структура HTTP запросов. Предлагается создание специальной прокси программы и надсмотр трафика для предотвращения HTTP атак. Рабочий принцип программы прокси показан на реально-алгоритмическом языке.

CREATING PROXY APPLICATION TO SECURE WEB SERVERS

R.F.FARADJULLAYEV

SUMMARY

Reviewed creating a special web proxy for securing web servers. In most cases it is not possible to catch HTTP attacks (hyper text transfer protocol) at other levels. Reviewed WAF (Web Application Firewalls) - intended for liquidation of such kinds of attacks. IN this article also reviewed structure of HTTP queries. Suggested creating special web proxy application and watching the traffic for preventing HTTP attacks. The working principle of this web proxy shown on the natural algorithmic language.