УДК 004

ВЕБ-САЙТТЫҢ ҚАУІПСІЗДІГІ: SQL ИНЪЕКЦИЯСЫ

Әділғазина Ақмарал Еркінбекқызы
akma_adilgazina@mail.ru
7М06111- «Ақпараттық қауіпсіздіктің әдістері мен технологиялары», 2-курс магистранті
Л.Н.Гумилев атындағы Еуразия ұлттық университеті, Нұр-Сұлтан қ., Қазақстан.
Ғылыми жетекші : Ташенова Жулдыз Мусагуловна - PhD, доцент м.а

Аннотация. Бұл мақала SQL инъекциясының анықтамаларының қолданыстағы әдістеріне және негізгі себептеріне шолу болып табылады. Қарастырылған материал негізінде мәліметтерді мәтін формаларына енгізу деңгейінде ақпаратты қорғау әдістері ұсынылады.

Кілт сөздер: ақпарат; қауіпсіздік; SQL; SQL инъекциясы.

Веб-сайттар мәліметтер базасында орналастырылған құрылымдық ақпаратқа сілтеме жасайды. Ақпаратты енгізу және көрсету үшін қолданушылар формалар арқылы деректермен өзара әрекеттеседі. Мысалы, аутентификация кезінде логин мен пароль енгізіледі, критерийлер серверде тексеріледі. Деректер базасында пайдаланушылар туралы ақпарат және қол жетімділіктің белгіленген деңгейі бар. Кейде сайт жасаушылар ақпаратты енгізу өрісінде сервердің жұмысын бұзатын мәтін болуы мүмкін жағдайды ескермейді. Сервер алынған мәліметтерді берілген команда ретінде өңдейді. Шабуылшы осалдықты пайдалана алады, мысалы, аутентификация алу арқылы әкімші құқығымен пайдаланушы деректері - бұл мәселе SQL инъекциясы деп аталады. Пайдаланушы аты мен құпия сөзді енгізгеннен кейін, мәтіндік форманың деректері серверге жіберіліп, сұраудың орнына келесі түрде ұсынылады[1]:

select * from Users where login=’person’ AND password= ‘key’,

person және *key* — сайттағы мәліметтер өрісіне енгізілген мәтін.

Егер қолданушы базасында логин мен пароль жұбы анықталған болса, қосымша қолданушыға авторизация береді. Бірақ пайдаланушы атын «person» сөз тіркесімен ауыстыру арқылы соңғы сұрау бастапқы пәрменді жоққа шығарады:

select * from Users where login=’***person*\’ -- ’** AND password= ‘key’.

Ал егер админ деген қолданушы болса, сайт сізге паролін тексермей, оның аккаунтында жұмыс істеуге мүмкіндік береді. Екі «-» сызықшасынан кейін база командаңың қалған бөлігін елемейді, өйткені түсініктеме SQL синтаксисінде осылай белгіленеді. «;» Таңбасы бір SQL командасын екіншісінен бөледі, сондықтан бірнеше командалар орындалуы мүмкін.

Мәліметтер базасында SQL инъекциясының осалдығын тудыратын қате сұраныстардың нұсқаларын қарастырайық.

Осалдықтың себептері

Мұндай осалдықтар келесі себептерге байланысты туындайды [2, б. 3-4]:

- SQL сұраныстарының динамикалық құрылысы. Программист үшін ең оңай нұсқа - қолданушы енгізген деректерді қосымша сүзгісіз дайын сұраныс құрылымына ауыстыру, мысалы:

select * from users where login =’request.getParameter(“person”)‘.

- Ерекше жағдайды өңдеу дұрыс емес. Егер сіз қатесі бар SQL командасын жіберуге тырыссаңыз, SQL сервері ол туралы ақпаратты пайдаланушыға қайта жібереді. Шабуылшы осы хабарламаларды басшылыққа ала отырып, дұрыс команданы енгізеді.
- Арнайы таңбалардың дұрыс өңделмеуі. Мысалы, MySQL мәліметтер қорында түсініктеме «-» немесе «#» таңбаларын қолдану арқылы көрсетіледі [3]. Сондықтан қосымшада мұндай таңбалар жұмыс істеуі керек.
- Түрді дұрыс өңдеу. Апострофты (') таңбаны өңдеу мәтіндік деректердің осалдығын жабады. Алайда мәселе сандық типтерде қалады. Шабуылдаушының команданы қосу мүмкіндігі бар, мысалы:

select * from money where sum = 1 union select 1, person, key from users.

Бастапқы сұрауға Union көмегімен тағы бір кесте қосылады, бұл жағдайда оны users деп атайды.

- Қауіпсіз ДҚБЖ конфигурациясы. Әдепкі бойынша, ДҚБЖ қосымшасы серверінің тіркелгісінде артықшылықтар бар, бұл дерекқорға шабуылдаушыға қажетсіз командаларды орындауға мүмкіндік береді.

Бағдарламалау тұрғысынан серверді SQL инъекциясынан қорғау мәселесі шешіледі. Әкімші бұл осалдық туралы біліп, тиісті қауіпсіздік шараларын қабылдауы керек.

Қорғау

SQL инъекциясы - бұл кең таралған және жойғыш осалдық, сондықтан қосымшаны және серверді осы шабуылдардан қорғау өте маңызды [4, б. 149-152].

Алдымен сізге кіріс деректерін сүзу қажет. Сүзу серверде жүреді, өйткені ақпаратты қолданбадан ұстап, өзгертуге болады.

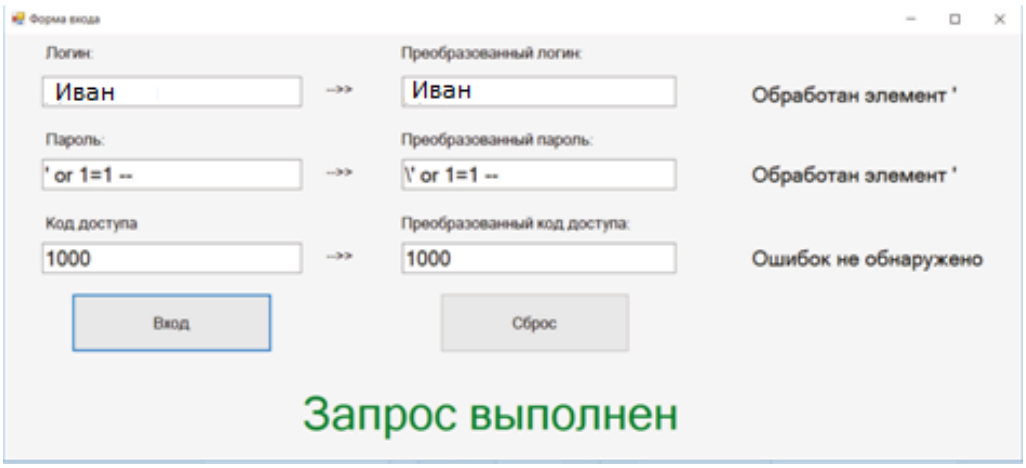
Мәтіндік ақпараттағы арнайы таңбалар, мысалы, апостроф (') немесе алға көлбеу (/), алға көлбеу (/) таңбасымен қосылуы керек. Мысалы, егер пайдаланушы аты өрісінде «Ivan» мәні болса, онда «I / van» мәні мәліметтер базасына жазылады және ешқандай қателіктер болмайды (1-сурет).

Түрді тексеру сандық ақпаратты тексеру үшін қолданылады. Сервер қолданушыға дұрыс қатені қайтарады, ол мәтінді сандық өріске орналастырады (2-сурет).

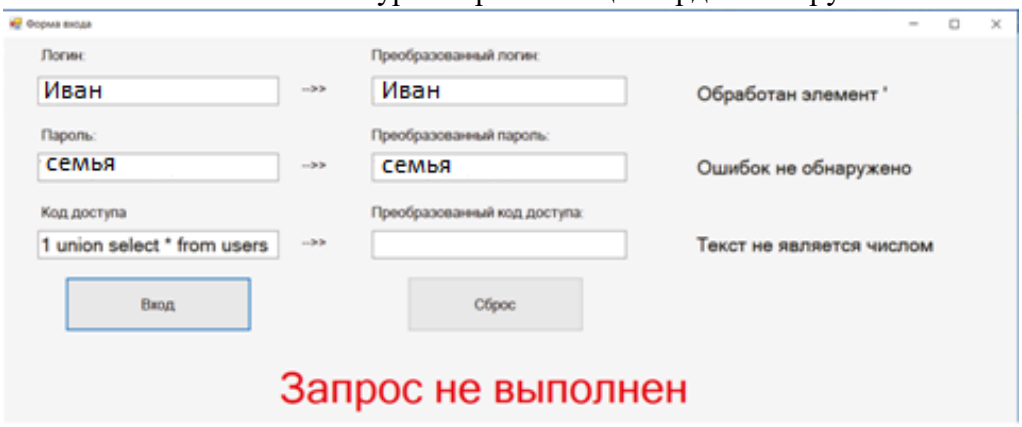
Мақалада деректерді енгізудің дұрыстығын тексеруге арналған бағдарламаның тізімі келтірілген (3-сурет).

Ақпаратты қосымшаға енгізу үшін таңбалардың санын шектеп, оны серверде тексерген жөн. Егер өріс толы болса, сұрау қабылданбайды.

Сондай-ақ, SQL серверінен қате шығаруды өшіру ұсынылады, бұл жағдайда шабуылдаушы командаларды «соқыр» болжауға мәжбүр болады. Сервер тіркелгісінің құқықтарын шектей отырып, мәліметтер базасын пайдаланушылардың құқықтарын конфигурациялау ұсынылады.



1-Сурет. Арнайы таңбаларды тексеру



2-Сурет. Сандық өріс арқылы қол жеткізудің сәтсіз әрекеті

```
private bool CheckOfString(TextBox TB, TextBox TB2, Label L, string type)
{
    TB2.Text = "";
    bool flag = true;
    if (TB.Text.Length == 0) { L.Text = "Пустая строка, запрос не выполнен";
        return false; }
    if (type == "text") {
        for (int i = 0; i < TB.Text.Length; i++) {
            if (TB.Text[i] == '\'')
            { TB2.Text += "\\\'";
                flag = false;
                L.Text = "Обработан элемент \'"; }
            else if (TB.Text[i] == '\"') {
                TB2.Text += "\\\"";
                flag = false;
                L.Text = "Обработан элемент \""; }
            else if (TB.Text[i] == '\\'){
                TB2.Text += "\\\\";
                flag = false;
                L.Text = "Обработан элемент \\"; }
            else { TB2.Text += TB.Text[i]; }
        }
    } else if (type == "int")
    { try { Convert.ToInt32(TB.Text);
            TB2.Text = TB.Text; }
        catch (FormatException)
        { L.Text = "Текст не является числом";
            flag = false;
            return false; }
    } if (flag) L.Text = "Ошибок не обнаружено";
    return true; }
```

3-Сурет. Енгізілген деректерді тексеруге және өңдеуге арналған код тізімі

Қорытынды

SQL инъекциясы веб-сайттың проблемаларына әкеледі. Кіретін ақпаратты дұрыс өңдеу арқылы құпия деректерге рұқсатсыз қол жеткізуді болдырмау маңызды. SQL инъекциясының осалдығын жабу ақпараттық қауіпсіздік пен веб-қосымшалардың дұрыс

жұмыс істеуін қамтамасыз етудің маңызды міндеті болып табылады. Бұл есептерді бағдарламалау құралдарының көмегімен шешуге болады, шешімнің мысалы мақалада қарастырылған.

Қолданылған әдебиеттер тізімі

1. Jones M. Fight against SQL injection attacks [Электронды ресурс] // IBM.
Қатынау режимі: https://www.ibm.com/developerworks/security/library/se-sqlinjection-attacks/index.html.
2. Егоров М. Выявление и эксплуатация SQL-инъекций в приложениях //
Защита информации. INSIDE. 2011. № 2. 2–8б.
3. Евтеев Д. SQL Injection от А до Я [Электронды ресурс] // Қатынау режимі: https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/PT-devteev-Advanced-SQL-Injection.pdf.
4. Бирюков А. А. Информационная безопасность: защита и нападение. М. :
ДМК Пресс, 2012. 474 б.