ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

Қ.И. САТБАЕВ АТЫНДАҒЫ ҚАЗАҚ ҰЛТТЫҚ ТЕХНИКАЛЫҚ УНИВЕРСИТЕТІ
КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ К.И.САТПАЕВА
KAZAKH NATIONAL TECHNICAL UNIVERSITY NAMED AFTER K. SATPAEV

«Қазақстанның жаңа экономикалық саясатын таратуда жас ғалымдардың орны мен рөлі» ХАЛЫҚАРАЛЫҚ САТБАЕВ ОҚУЛАРЫНЫҢ

ЕҢБЕКТЕРІ

IV том

ТРУДЫ

«Роль и место молодых ученых в реализации новой экономической политики Казахстана» МЕЖДУНАРОДНЫХ САТПАЕВСКИХ ЧТЕНИЙ

Том IV

PROCEEDINGS

INTERNATIONAL SATPAYEV'S READINGS
«Role and position of young scientists in implementation Kazakhstan's New Economic Policy»

IV volume

Алматы 2015 Almaty

ӘОЖ 004.65

Шалабаев Қ.М. студент, Баймағамбетова А.Р. студент, Сағымбекова А.О.
Қ.И.Сәтбаев атындағы Қазақ ұлттық техникалық университеті,
Алматы қ., Қазақстан Республикасы
sagymbekova@mail.ru

WEB-ҚОЛДАНБАЛАР ҚОРҒАНЫСЫН ЖҮЗЕГЕ АСЫРУ

Аңдатпа. Менеджерлер, бухгалтерлер секілді, дерекор қолданушылардың көбісі ақпараттық технологияларға қатысы жоқ адамдар, сондықтан олар үшін дерекқормен жұмыс жасауға арналған бағдарламалар әзірленеді. Осындай қосымшалардың бір түрі ретінде, қолданушыларға ыңғайлы әрі қол жетімді интерфейс ұсынатын веб-қосымшалар болып табылады. Барлық заманауи веб-сайттар интерактивті әрі қолданушылармен әрекеттесу үшін ақпаратты енгізуге арналған енгізу формалары болады. Мысал ретінде логин мен авторизациялауға арналған енгізу өрістері, кері байланыс формасы арқылы мәтіндік хабарламаларды жіберу өрістері және т.б. Веб-сайтта енгізу өрісі болған жағдайда, сервердегі өңдеуші скрипт қолданушылардан қандай да бір ақпаратты күтіп тұрады. Оны өңдегеннен кейін, скрипт бағдарламаланған әрекеттер орындайды, ал авторизация жағдайында, қолданушы сайттың жабық бөліктеріне қатынас құруға рұқсат алады. Ол үшін php скрипттің жұмыс алгоритміне сәйкес тексеру енгізу қажет, нәтижесінде ақпарат сұрыпталып, қолданушылар қате енгізу туралы мәлімет алады.

Мақалада web-қолданбаларға төнетін кеңінен таралған қатерлердің түрлері мен олардың алдын алу және қорғау шаралары туралы мәліметтер қарастырылған. Төнетін қатерлер арасынан қолданушы енгізетін мәліметтерді сүзгілеуден өткізу қорғау шаралары жүзеге асырылған.

Түйін сөздер: web-қолданба, сервер, PHP, форма, қауіп, қорғаныш, мәліметтерді қорғау.

Web-қолданбалардың қауіпсіздігін қамтамасыз ету бұл көптеген аспектілерге ие, комплексті көп деңгейлі мәселе.Бұл мәселелер жеке web – қолданбаларға ғана емес, сонымен қатар жалпы желілік қызметтерге де тән. Web – қолданбалар қауіпсіздігіне web-сервер және оның баптауларының қауіпсіздігі,негізгі функционалдық қызметті іске асыратын арнайы скриптар қауіпсіздігі кіреді. Желілік қызметтер қауіпсіздігіне желілік инфрокұрылымның қауіпсіздігі (бағдарлауыш, желіаралық экран, DNS сервер және т.б.), web-сервер және деректер қоры орналасқан физикалық серверлер қауіпсіздігі, TCP/IP хаттамалар стегінің қауіпсіздігін іске асыру, басқа да іске қосулы web-серверлер қауіпсіздігі және т.б [1].

Потенциалды қауіптер:

1. Қолданбалар қызметін бұзу немесе DoS (Denial of Service) шабуылдар
2. Деректер қорынан рұқсатсыз мәліметтер оқу
3. Деректер қорында рұқсатсыз мәліметтерді өзгерту
4. Серверден файлдар оқу

5. Сервердегі файлдарды өзгерту немесе бөгде үрдістерді іске қосу

4 және 5 қауіптер қолданушылардың қатынау құқығына байланысты. Web-қолданбалардағы көптеген шабуылдарды басу скрипты арнайы баптау арқылы жүзеге асады.Негізгі қорғаныш әдісі бағдарламаны қолдану алдында қолданушы енгізген бүкіл мәліметтерді мұқият тенксеруден өткізу. 1 және 2 қауіптерді жүзеге асыратын көп таралған әдіс «SQL injection», яғни скрипке автор тағайындамаған арнайы мақсатты орындауға негізделген SQL сұраныс жіберу.

Кең таралған жағдайдың бірі, қолданушыға басқа қолданушы енгізген ақпаратты көрсету қажеттілігі. Каскүнем енгізген мәлімет (мысалы, JavaScript тілінде жазылған арнайы бағдарлама) сайттың басқа қолданушыларына теріс әсерін тигезбеу қажет.

Бұзу каупін төмендету үшін төмендегі арнайы қорғаныш шараларын орындау қажет:

1. Сервер құрылғыларының жоғары сапалылығы және өнімділігі. Сервер мықты болған сайын салмақты DOS шабуылдарға төзімді болады.

2. Мәліметтерді толық сүзгілеуден өткізу.Бүкіл енгізілген мәліметтерді өңдеп, толық сүзгілеуден өткізіп,қауіпсіз қалыпқа келтіру.

3. Бүкіл жүйелік қателер жайлы хабарламалар өшірулі болуы тиіс, оның орнына арнайы құжатталған CMS хабарламалар болуы тиіс.Өйткені жүйелік қателер жайлы хабарламалар каскүнемге жүйе кемшіліктерін сараптауға мүмкіндік береді.

4. CMS-ті үнемі жаңарту.Сайтты басқару жүйесін әзірлеушілер өз өнімдерінің кемшіліктерін анықтап,оларды жоятын шаралармен толықтырады.

5. Сервер ғана қорғаныста болмай,сайт құрушының дербес компьютерінде қорғаныш шаралары іске асырылуы қажет.

6. Сервердегі мәліметтерді үнемі резервті көшіру.Бұл көшірмелер өзге серверде болуы қажет.

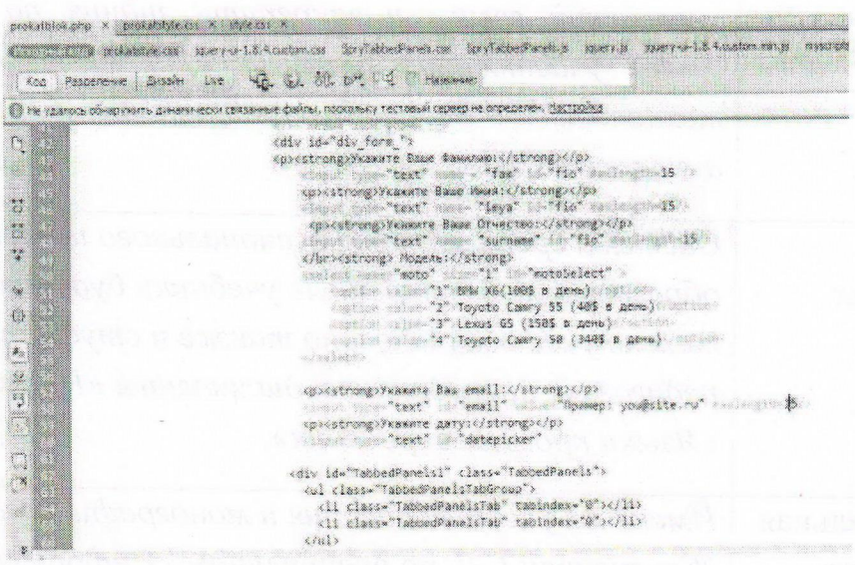
Қазіргі уақытта кең таралған серверлі технологиялардың бірі PHP болып табылады. Ең алдымен серверлі және клиенттік бола алатын ,қолданушы енгізген мәліметтер сенбеу қажет.Өйткені олар зиянкесті скрипт болуы мүмкін [2].

Мысалы, біздің сайтымыздағы автомобильді жалға алуға тапсырыс беру үшін толтырылатын формалар.Аты-жөнді,электронды почтаны және жеке хабар енгізетін өрістер. PHP скрипт формалардан алынған мәліметтерді арнайы файлда сақтацды.Осы файлдан әрі қарай мәліметтер оқылады.Алғашқы көзге ешқандай қатер жоқ сияқты болып көрінеді,алайда каскүнем бұл формаларды өз мақсаттарында қолдана алады.

Ең алдымен аты-жөн және e-mail мекен-жайды енгізітін формада енгізілетін символдар ұзындығына шектеу қою.Бұл қорғаныштың көптеген әдістерінің бірі.Ол үшін енгізу формаларында мысалы ретінде maxlength=15 кодын жазайық.Мысалы:

```
...
<input type=text name=user_email maxlength=15>
...
```

Енгізілетін мәліметтер ұзындығына шектеу қоятын код 1- суретте,ал ұзындықты шектеу нәтижесі 2- суретте келтірілген.

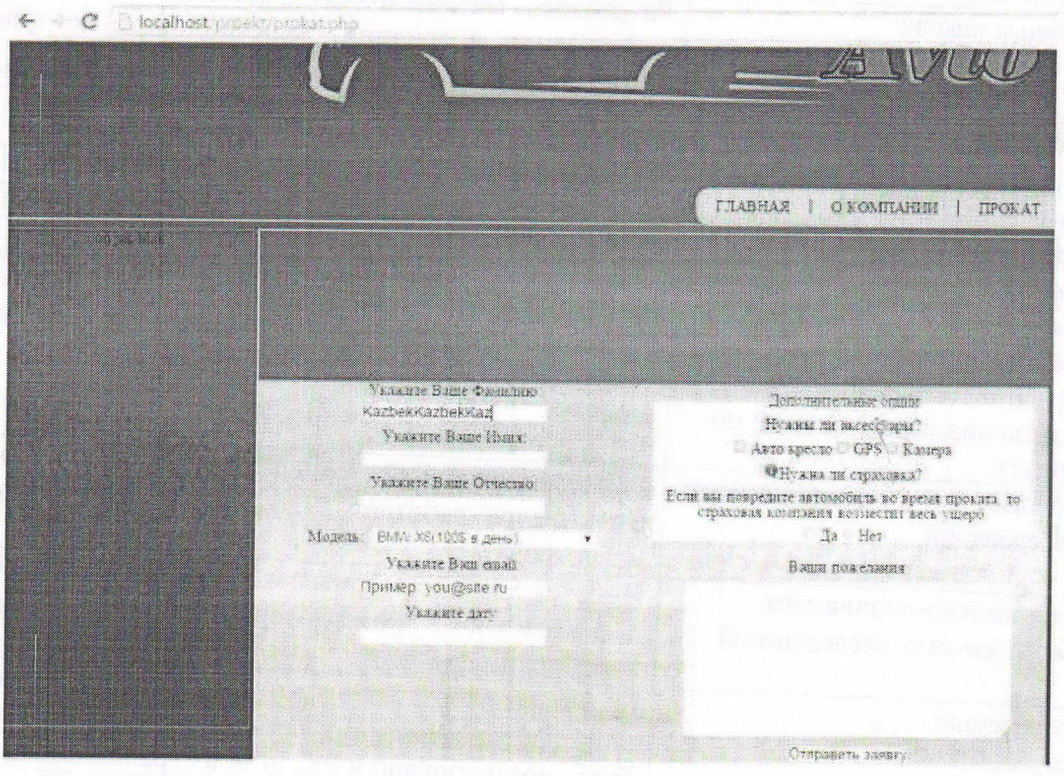


Сурет 1 - Енгізілетін мәліметтер ұзындығына шектеу қоятын код

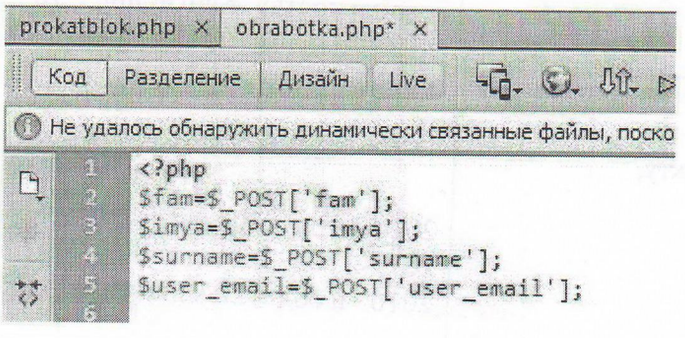
Сонымен мәлімет енгізу өрісінде 15 символдан артық енгізуге болмайды.
PHP скрипт басында келесі кодты жазамыз :

```
<?php
$fam=$_POST['fam'];
$imya=$_POST['imya'];
$surname=$_POST['surname']
$user_email=$_POST['user_email'];...
```

Өңдейтін PHP файлға POST массив ретінде формадан жіберілетін мәліметтер айнымалылары 3-суретте келтірілген.



Сурет 2 - Ұзындықты шектеу кодының нәтижесі



Сурет 3 - Өңдейтін PHP файлға POST массив ретінде формадан жіберілетін мәліметтер айнымалыларының коды

Яғни айнымалылардың мәні POST массивке сай өрістерден тікелей аламыз . Код барлық айнымалыларға әсер етеді.. Бұл жағдайда хабар жіберетін форманың мәлімет беру әдісін method="post" – қа өзгертеміз, мысалы [3] :

...
<form action="obrabotka.php" id="my_form" method="post">
...

Үнсіз GET әдісінен мәліметерді серверге жіберу POST әдісінің айырмашылығы, мәліметтер браузердегі мекен-жай қатары арқылы емес ,мәліметтер дестесі арқылы беріледі. Яғни POST-ты Windows стандартты жеткізу бағдарламары мен Делфи сұраныс жіберу түрі арқылы жасанды құруға болады. :

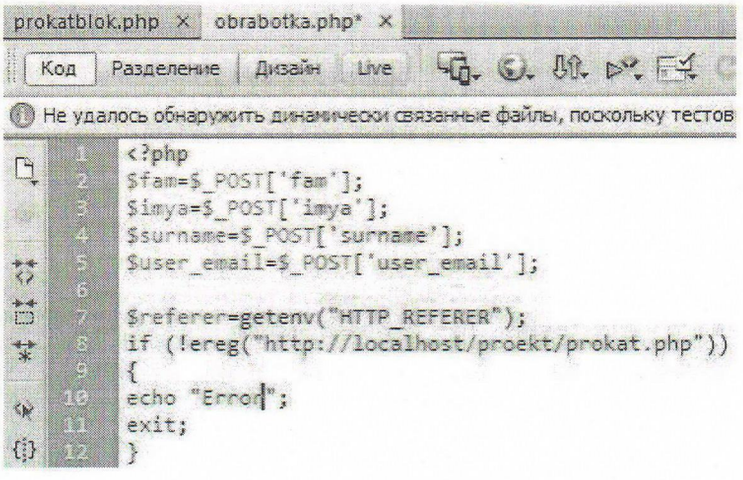
...
POST /guest.php HTTP/1.0
user_email= ya_tebla_vzlomal...
...

Егер қаскүнем жоғарыдай кейіпте мәлімет жіберетін болса,оны төмендегідей тоқтатуға болады :

```
$referer=getenv("HTTP_REFERER");
if (!ereg("http://localhost/proekt/prokat.php"))
{
echo "Error";
exit;
}...
```

Мысалда көрініп тұрғандай, ашық тұрған браузердің web-беттерінен (біздің домен - http://localhost/proekt/prokat.php) сұраныстың жіберілгені тексіріледі. Егер қателік табылмаған жағдайда керек іс-әрекет орындалады, кері жағдайда қателік жайында "Error" хабарын шығарады және скрипт өз жұмысын аяқтайды: exit.

HTTP_REFERER айнымалысы қолданушы браузерімен құрылады, сондықтан біз оны тексеруіміз қажет. Оның және POST сұраныстың жасанды түрін жасау қиынға соқпайды. Домен жолын тексеретін және дұрыс емес жол енгізілген жағдайдағы қателік жайлы хабар шығару коды 4-суретте келтірілген.

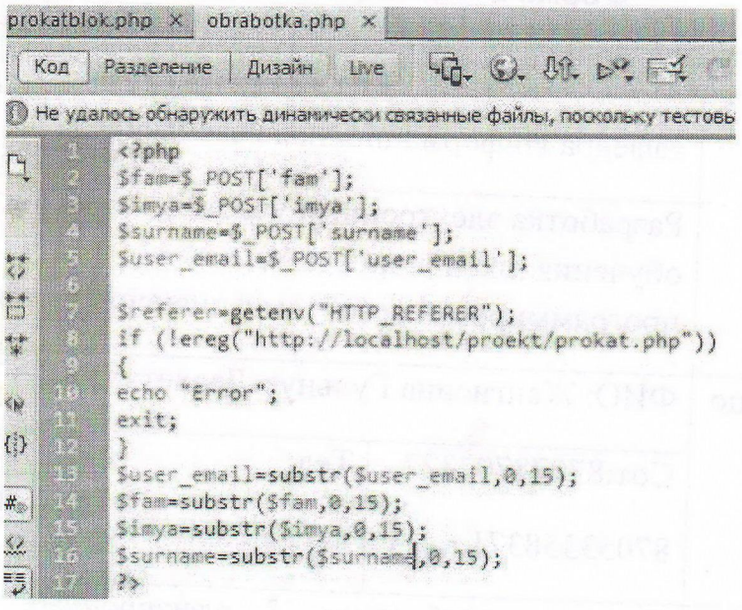


Сурет 4 - Домен жолын тексеретін және дұрыс емес жол енгізілген жағдайдағы қателік жайлы хабар шығару коды

Қаскүнем біз орнатқан қорғаныш шараларын бұзып өтіп,кез келген жерден серверге кез келген ұзындықтағы хабар жіберетін жағдайда, HTTP_REFERER тексеру скриптынан кейін , қатаң түрдегі хабарды немесе қатарды кесетін код жазу керек:

...
$user_email=substr($user_email,0,15);
...

Енгізілген мәліметтерді анықталған ұзындықта кесетін код 5-суретте көрсетілген.



Сурет 5 - Енгізілген мәліметтерді анықталған ұзындықта кесетін код.

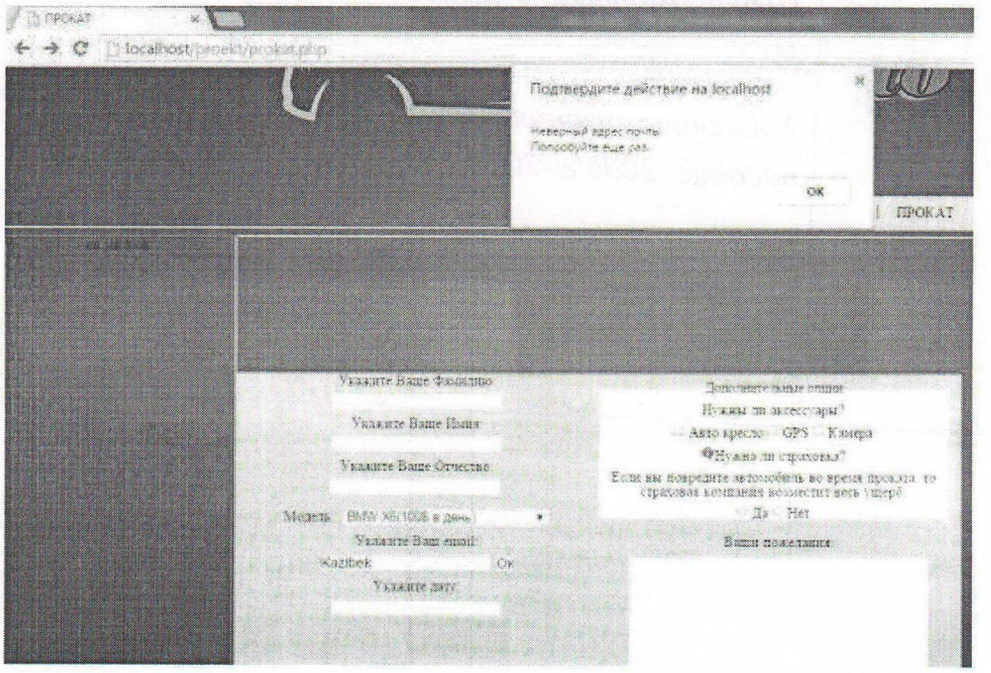
Қолданушы серверге қате хабар жіберер алдында ,оны алдын-ала ескерткенм жөн.Бұл мәселені шешу үшін JavaScript тілінің қолданушы скриптары қажет. Ол үшін енгізілген мәліметтерді жіберу алдында тексеріп, қателіктер анықталған жағдайда арнайы хабар жіберу.Мұны электронды мекен-жайды дұрыс енгізу үшін қолдануға болады [4] :

```
...
<script language="JavaScript">
function checK(f) {
if (f.email.value=='') { alert("Укажите адрес почты."); f.email.focus(); return false }
if (/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,4})+$/.test(f.email.value)) { return true }
alert('Неверный адрес почты.\nПопробуйте еще раз.'); f.email.select()
return false
}
</script>
<form action="obrabotka.php" id="my_form" method="post"onSubmit="return checK(this)">
<p><strong>Укажите Ваш email:</strong></p>
<input type="text" name="user_email" id="email" value="Пример: you@site.ru" maxlength=15 onfocus="if (this.select) this.select()" onclick="if (this.select) this.select()" size=20>
<input type="submit" name="subscribe" value="Ок"/></form>
...
```

Жоғарыдағы кодтың орындалу нәтижесі 6- суретте көрсетілген.

"Ок" батырмасын басқаннан кейін мәліметтер файлға берілмей тұрып checK функциясы арқылы тексеріледі.Егер енгізілген мекен- жай бос немесе қате болған жағдайда ,қолданушы келесі хабар алады: "Укажите адрес почты" немесе "Неверный адрес почты.\nПопробуйте еще раз."Бұл хабарлар арнайы alert() терезесінде шығады ,яғни web-бет қайта жүктелмейді.

Қорыта келе, қолданушы формаға енгізетін мәліметтер тарапынан web-қолданбалардың функционалдық қызметіне көптеген қауіптер төнеді. Енгізілетін мәліметтерді қатаң сүзгілеуден өткізіп, қауіпсізлік шараларын қолдану қажет.



Сурет 6 - Енгізілген электронды мекен-жайды алдын-ала тексеру коды

ӘДЕБИЕТТЕР

1. «Безопасность Web-приложений» http://daxnow.narod.ru/index/0-40
2. «Эволюция атак на веб-приложения и веб-сервисы»http://www.slideshare.net/c3retc3/2-21310126
3. «Приемы сетевой обороны на PHP»http://www.php.su/articles/?cat=security&page=010
4. «PHP-сценарии обработки HTML-форм» http://www.in-internet.narod.ru/teor/ phpform.html

REFERENCES

1. «Bezopasnost WEB-prilozhenii» http://daxnow.narod.ru/index/0-40
2. «Evolyuciya atak na web-prilozheniya i web-servisy» http://www.slideshare.net/c3retc3/2-21310126
3. «Priyomy setevoi oborony na PHP» http://www.php.su/articles/?cat=security&page=010
4. «PHP-scenarii obrabotki HTML-form» http://www.in-internet.narod.ru/teor/phpform.html

Шалабаев Қ.М., Баймагамбетова А.Р., Сағымбекова А.О.

Реализация защиты WEB-приложений

Резюме. Приведена классификация методов атак и методов защиты на Web-приложений. Реализована защита веб-сайта.

Ключевые слова: web-приложения,сервер,PHP,форма,защита,защита информации.

Sagymbekova A.O., Shalabaev K.M., Baimagambetova A.R.

Implementation of WEB-application protection

Summary. A classification of attack methods and methods of protection for Web-based applications. Implemented such protections limiting the length of the password.

Key words: web-application, server,PHP, shape, protection, protection of information.