

**ЎЗБЕКСТАН РЕСПУБЛИКАСЫ БАЙЛАНЫС,
ИНФОРМАТИЗАЦИЯЛАЎ ҲАМ ТЕЛЕКОММУНИКАЦИЯ
ТЕХНОЛОГИЯЛАРЫ МЭМЛЕКЕТЛИК КОМИТЕТИ**

**ТАШКЕНТ ИНФОРМАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАРЫ
УНИВЕРСИТЕТИ НӨКИС ФИЛИАЛЫ**

Информациялық технологиялар кафедрасы

**Компьютер инжиниринги факультетинин
«Информатика ҳәм информациялық технологиялар» бағдарының
4-курс студенти Бекниязов Динисламның**

ПИТКЕРИЎ ҚӘНИГЕЛИК ЖУМЫСЫ

Темасы: “Клиент-сервер технологиясында жаратылган системаларда серверге жиберилген сораўнамаларда қәуипсизлик шараларын көриў усыллары”

Илимий басшы: _____

**«Матбуот тарқатувчи»
сыңар кәрханасы
директоры Салимова Б.
асс. А.Рысназаров**

Кафедра баслығы: _____

т.и.к. Арзымбетов Т.З

Мазмуны:

Кирисиў	3
§1. Клиент-сервер түріндеги системалар хам олардың информациялық қәўипсизликлери	
§2. Веб серверлерге жиберилетуғын сораўларды жаратыў тиллери.....	
§3. Веб серверлерди хәм мағлыўматларды қәўипли сораўнамалардан қорғаў усыллары	
Жуўмақлаў.....	
Әдебиятлар	

КИРИСИҰ

Жаңа информациялық технологиялардың раўажланыўы хәмде жамийетти компьютерлестириў, жаратылған информациялық системаларда текте информация кәўипсизлиги шараларын көриў шәрт болып қалмай бул оның тийкарғы характеристикаларының бирине айланбақта. Информацияларды қайта ислеўде кәўипсизлик факторы биринши дәрежеде болған системаларда ушырасады.

Хәзирги ўақытта көпшилик информациялық системалар «клиент-сервер» архитектурасы тийкарында исленип хәмде жумыс ислеп турғанын, солардың ишинде басым көпшилиги Веб серверлерге жайластырылып, Веб технологиялар тийкарында ислеп шығылғанлығын есапқа алсақ онда олардың информациялық кәўипсизликлери мәселелери актуал болып турғанлығын көремиз. Веб технологиялар жәрдемінде жаратылған системаларды бузыўға сырттан болатуғын ҳарекетлердин көпшиликлери системалардың клиент тәрәпинен жиберилетуғын сораўнамалар тийкарында шөлкемлестирилиўи, программалаўшылардың бундай системаларда интерфейслерди жаратыўда жеткиликли билим хәмде тәжирийбеге ийе болыўы керек болады.

Питкериўши қанийгелик жумыста Клиент-сервер технологиясында жаратылған системаларда Веб серверге жиберилген сораўнамаларда кәўипсизлик шараларын көриў усыллары , бузыўшылардың қоллаатуғын усыллары анализленип, оларға қарсы информацияларды қорғаў усыллары қаралып шығылған.

Жумыстың биринши параграфында Клиент-сервер түриндеги системалар, олардың жумыс ислеў технологиялары, оларға сырттан болатуғын хўжимлер хам олардың информациялық кәўипсизликлери үйренилди.

Екинши параграфында клиент сервер технологиясында жумыс ислеуши системаларды жаратыўдың бири болған Web серверлерде

программаластырыў тили РНР үйренилип хәм оның мүмкиншиликлери каралып өтиледі.

Үшинши параграфында Веб серверлерди хәм мағлыўматларды қәўипли сораўнамалардан қорғаў усыллары анализленип, анық жағдайларда қаўиплердиң алдын алыўшы программалық кодлар ислеп шығылады.

§1. КЛИЕНТ-СЕРВЕР ТҮРИНДЕГИ СИСТЕМАЛАР ҲАМ ОЛАРДЫҢ ИНФОРМАЦИЯЛЫҚ ҚӘУИПСИЗЛИКЛЕРИ

Информациялық системаның эффективлиги көбинше оның архитектурасына байланысly болады. Ҳәзирги ўақытта олардың ишинде перспективли болып клиент-сервер архитектурасы болып табылады.

Клиент-сервер архитектурасының реал үш масштаблы басқышы бар болып, олар төмендегише:

монопол системалар- өз ишине бир клиент ҳәм бир серверди алады (десктоп-системалар);

корпоратив системалар- өз ишине бир серверге жалғанатуғын көплеген клиентлерди алады (LAN, локаль тармақлар);

интернет-системалар, өз ишине көплеген серверлер менен ислесиўши көплеген клиентлерди алады (WAN, глобаль системалар).

Клиент-сервер архитектурасындағы клиент ҳәм серверге өзлерине тән шәртлер қойылады. Булл шәртлерди қанаатландырыўшы программалар мағлыўматларды



Клиент процессоры Сервер процессоры

Бөлистирилген қайта ислеў дегенде клиенттиң сораўына байланысly сервер тәрепте программаның орыланыўы түсиниледи.

Программалық тәмийнаттың серверлик тәрепи мағлыўматларды сақлаў ҳәм қайта ислениўине, ал клиент тәрепи серверге керекли сораўнаманы жиберий ушын ислетиледи.

Клиент-серверлик системалар төмендегише артықмашылыққа ийе:

Клиент-серверлик усыл модуллик болып, серверлик программалық компонентлер ықшам ҳәм автоном болады;

Программаның хәр-бир компоненти пайдаланыўшы режиминде өз алдына қорғалған процессте орынланады, сервердиң бузылыўы (сбой) операцион системаның басқа компонентлерине кесент етпейди;

Компонентлердиң автономлылығы олардың бир компьютерде бир-неше процессорларда (симметриялық көп процессорлық қайта ислеў) ямаса тармақтың бир-неше компьютерлеринде (бөлистирилген есаплаўлар) орынланыўын тәмийнлейди.

Бул артықмашылықлардан басқа клиент тәрәпинде пайдаланыўшыға мағлыўматларды алыў, экранда көрсетиў, редакторлаў, серверге жиберилиўши сораўнамалар ушын қолай интерфейс жаратылыў мүмкиншилиги. Соның менен бирге буннан басқа, клиент тәрәпте мағлыўматлардың қандайда бөлегин қайта ислеўдиде шөлкемлестириўге болады. Бул сервер ресурсларына түсетуғын аўырмашылықты хәмде тармақтағы мағлыўмат алмасыў трафигин азайтады.

Енди серверлердиң атқаратуғын роллери, функциялары хәмде қолланылыўы бойынша: исши топар сервери (Сервер рабочей группы), домен контроллеры сервери, Прокси сервер, Электрон почта сервери, Веб сервер, Терминаллық серверлер, Мағлыўматлар базалары сервери, Файл сервери, Программалар сервери, Брандмауэрлар (файрволл), ДНСР серверлер, FTP серверлер болып классификацияларға бөлинеди.

Хәзирги күнде кең қолланылатуғын хәмде питкерий қәнийгелик жумыстың мақсети болған Web серверлер менен танысып өтемиз.

Веб-сервер (web-server) – бул веб-сайтларға клиент тәрәпинен берилген сораўнамаларды (HTTP-сораўлар) қабыл етиўде хәмде қайта ислеўде әмелге асырыўшы сервер. Клиент сыпатында әдетте хәр қыйлы веб-браузерлер қолланылады. Жуўап ретинде веб-сервер клиентке HTTP-жуўап, көпшилик жағдайларда , HTML-бетлер ге қосымша, файллар, сүүретлер, х.т.б. мағлыўматларды береди.

Булардан басқада веб-серверлер тармақ хызметлерине, мағлыұматлар базаларына, файлларға рұхсатларды хәмде электрон саўданың басқада программаларына сораўнамаларды шөлкемлестириўде қолланатуғын CGI, JSP, ASP хәм PHP уқсаған скриптлердиде орынлайды.

Пайдаланыўшының клиенті болған веб-браузер арқалы керекли ресурсқа берилген URL-адреслер бойынша сораўнамалар, хәмде сервердиң жуўап ретинде берген мағлыұматлары HTTP протоколы арқалы әмелге асырылады.

HTTP (англ. HyperText Transfer Protocol - протокол передачи гипертекстлерди узатыў протоколы) – булл әмелий басқыштағы мағлыұматларды узатыў протоколы болып, оның тийкарғы хызмети «клиент-сервер» технологиясында тармақ хәм пайдаланыўшының арасындағы байланысты жаратыў болып есапланады.

Веб-серверди биринши жаратыўшы Европаның ядролық изертлеўлер лабораториясының программылық тәмийнат бойынша мәслахатшы болып жұмыс ислеген Тим Бернерс-Ли болып есапланады. Ол өзиниң жұмысларын жеңиллестириў ушын Дүнья жүзлик торға тийкар болған «Энквайр» (англ. enquire - сораўшы) программасын жаратты.

1989-жыл ол CERN ның ишки тармағын жаратыўда бир-бири менен гипербағытлар арқалы байланысқан хўжжетлерди бериўши глобаллық гипербағытлы проектти усынды. Бул проекттиң әмелге асыўы CERN нын алымлары ушын информацияларды алмасыў, излеў хәм алымларды бирлестириўди жеңиллестиретуғыны анық еди. Оның ушын ол хәм жәрдемшилери URI хәм URL идентификаторларын, http протоколын хәмде HTML тилин ислеп шықты. Жоқарыдағы бул технологиялар хәзирги Интернетте кеңнен қолланылады хәмде оларсыз ислеп болмайды.

Бернерс-Ли проектти орынлаў нәтийжесинде дүньяда биринши «httpd» деп аталыўшы веб-серверди хәмде NeXT компьютери ушын WorldWideWeb деп аталыўшы веб-браузерди ислеп шықты.

1991-1993 жыллар аралығында Бернерс-Ли, бул технологияларды жетилистирип кейин жәмийетке усунды. Кейин браузерди ол "Nexus" деп атын өзгертти.

Дүньядағы биринши веб-сервер хэм веб-браузер NeXTSTEP персонал компьютеринде жумыс иледи. Ол компьютер хэзирги ўақытта CERN музейине қойылған.

Тим Бернерс-Ли биринши веб-сайтты <http://info.cern.ch> адресине жайластырды, хэм ол хэзир архивте сақланбақта.



http://info.cern.ch - home of the first website

From here you can:

- [Browse the first website](#)
- [Browse the first website using the line-mode browser simulator](#)
- [Learn about the birth of the web](#)
- [Learn about CERN, the physics laboratory where the web was born](#)



Енди биз веб-сервердиң тийкарғы хэм қосымша функцияларын карап өтейик:

-Тармақтың TCP/IP протоколлын қолланып HTTP стандартындағы протокол арқалы веб-браузерлерден сораўнамалар қабыл етиў;

- HTTP протоколы менен излеў хэм файллар, басқада хўжетлерди браузерге жиберий;

- mailto FTP, Telnet х.т.б. уқсаған сораўнамаларды қайта ислеў хэм хызмет көрсетиў;

- CGI стандарт интерфейси арқалы әмелий программаларды орынлатып оның нәтийжелерин жиберий хэм қабыллап алыў;

- Навигациялық сўретли карталарды (Image map) ислеў хэм хызмет көрсетиў;

- Java қосымталарды жүклеу;
- Серверди администрациялау хәм оператив басқару;
- Пайдаланыушыларды авторизация хәм олардың аудентификациясы;
- Пайдаланыушылардың хәр-түрли ресурсларға мүрәжатиңиздиң дизимге алыу журналын жүритиу;
- Веб-бетлердиң жұмысларын автоматластыру;
- Динамикалық түрде генерацияланыушы бетлердиң ислеуин шөлкемлестириу;
- Клиентлер менен қорғалған HTTPS протоколында ислесиуди тәмийнлеу.

Булардан басқада веб-серверлерде излеуге хәмде мағлыұматлар базалары менен байланысты шөлкемлестириуге уқсаған әмеллерди орынлаушы хәр-түрли программалар өз жұмысын атқарады.

Веб-серверлер оларға орнатылған программалық тәмийнатларға қарап характерленеди. Бунда орнатыу, конфигурациясын туурылау, серверди басқару, администрациялау, серверге жайластырылатуғын информацияны басқару, сол информацияны қорғау, рухсатты қадағалау, қосымшаларды ислеу функциялары хәмде өнімдарлықлары есапқа алынады.

Бир уақыттың өзінде бир-неше клиентлердиң бир-неше ресурсларға мүрәжетлерине хызмет етиуин есапқа алсақ, онда веб-серверлерге орнатылатуғын программалық тәмийнатлардың анакурлым үлкенлигин көремиз.

Веб серверлердиң кеңнен қолланылыуы, оның интернети комерциялық активлигиниң артыуына алып келди. Бул өз гезегінде веб-сервердиң информациялық қәуипсизлигин асырып бару кереклигин көрсетпекте.

Биз информация қәуипсизлиги усыллары хаққында сөз етиуимизден алдын информациялық системаларға болатуғын қәуип-қәтерлер тұралы түсиникке ийе болуымыз керек.

ИС ның қәуіпсізлігі дегенімізде оның нормал іслеуіне кесент етіуіші тосынарлы ямаса қастан ісленген хәрекетлерден қорғалғанлығы түсиниледи. Басқаша айтқанда ИС ға сырттан болатуғын гуманлы хәрекетлерге қарсы турыу қәбилети болып табылады.

ИС ның нормал хәрекетін істен шығарыушы ямаса информацияға санкцияланбаған рухсатты алған адамды әдетте бузыушы, айырым жерлерде «компьютер қарақшылары» (хакер) депте атайды.

Информация бузыушылары анық конфиденциаль информация дерегін тауып оны аз шығын менен нызамға қарсы өзлестіриу хәрекетінде болады.

Информацияға бундай жол менен ерисиу ямаса информацияны бузыу мақсетінде өз жеке мәплери бар хаккерлер, бизнес конкурентлер хәмде информацияның анықлығын бурмалаушылар хәр-қыйлы усыллар хәм кураллардан пайдаланады.

Қастан информацияны бузыу хәм информация қәуіпсізлігін сақлаушылар ортасындағы гүрестің нәтийжеси – бул ким артығырақ билиу хәм хәрекет етиуге байланыслы болады.

Жоқарыда айтылып өтилген қәуіп-қәтерлерге қарсы веб-серверлер ушын көплеген программалық хәмде техникалық тәмийнатлар іслеп шығылған, жәнеде ісленбекте. Техникалық тәмийнат өз алдына программалық тәмийнатқа ийе қурылмалар болып есапланады. Олар өзи арқалы өткерилетуғын информацияларды филтрлеуші сырлы автоном алгоритмлерден туратуғын программалар хәмде ишки техник қурылмалар болып оларда уақыттың өтиуі менен (хәзирги дәуірде жүдә аз уақыт) гөнеген болып қалмақта.

Соның ушында веб-серверлерге қойылатуғын системалардың іслеу алгоритмлерін жазғанда программалаушы хәмме тәрептен қәуіпсізлік шараларын көрип барыуы керек хәмде системаның іслеу дәуірінде қәте-кемшиликлерди дүзетип барыуы керек.

Енди биз өткен дәуірлерде веб-серверлерге болатуғын хұжимлер түрлери, хаккерлердиң қолланған усыллары, оларға қарсы гүресиў усылларына мысаллар көрип өтейик.

Саралаў (Brute Force) – пайдаланыўшының атын хәм поролин, шифрлаў гилтин х.т.б. алыў ушын ислеп шығылған автоматластырылған процесс. Саралаўдың туўры хәм кери түрлери болып, туўры саралаўда бир пайдаланыўшының аты ушын поролдиң хәр-түрли варианлары изленеди, кери саралаўда бир парол сайланып соған пайдаланыўшының атлары маслығы тексериледи.

Бул усылға қарсы гүресиўде поролди қәте киритиўге шеклеў қойыў қолланылады. Мысалы поролди қатара үш реет қәте киритиўден кейин қандайда қосымша шәрт қойыў ямаса сораў аралықларын белгили ўақытқа иркиў керек.

Жеткиликсиз аутентификация (Insufficient Authentication) – бундай қорғалыўдың бослығы, веб-сервер хұжим етиўшиге зәрүрли аутентификациясыз керекли информацияға ямаса сервердиң функцияларына рухсатты берип қояды. Мысалы көплеген веб исленбелерде админстративлик рухсатты шөлкемлестириўши кодларды түбир директорияға (/admin/) жайластырады. Әдетте булл жерге бағдар сервердиң басқа файлларында көрсетилмесесе, әпиўайы стандарт браузер арқалы кириўге болады. Серверге контролды алыў ушын усы директориядағы бетлерге кириў жеткиликли.

Буған қарсы гүресиў ушын сайттағы улыўма аутентификациюдан басқа сайттын админстративлик бөлиминне кириўде пайдаланыўшы қосымша аутентификациюдан өткериўди шөлкемлестириў керек.

Поролларды қәўипсиз емес қайта тиклеў (Weak Password Recovery Validation) – Веб сервер бузыўшыға басқа пайдаланыўшының пароллерин қайта тиклеў хызмети арқалы берип қояды. Паролди қайта тиклеўде көплеген серверлер қорғалыў шәртлерине жетерли емес сырлылықты пайдаланады.

Паролларды қайта тиклеуде әмелге асырылатуғын жұмыслады төмендегіше жаратқан дурысырақ болады. Пайдаланыушы «паролди тиклеу» түймесин басқанда, ол оннан логинди хәмде дизимнен өткен ўақытта көрсеткен почтасын сораытуғын бетке өтиўи керек. Соннан кейин пайдаланыушының почтасына паролди тиклеуши бетке генерияцияланған бағыт жибериледи. Бул жағдайда текте почта ийеси ғана паролди өзгерте алады.

Сессия идентификаторының белгили мәниси (Credential/Session Prediction) – бул басқа пайдаланыушыларға ашылған сессияларды тутып алып пайдаланыуға мүмкиншилик береді. Көплеген серверлер сессия идентификаторын өзлериниң алгоритмлери арқалы ислеп береді. Бунда көпшилик жағдайларда хазирги ўақыт хәм пайдаланыушының белгилери қолланылады.

Бундай жағдайларды келтирип шығармау ушын Веб-серверди дурыс конфигурациялау керек, жәнеде исенимлирек болыуы ушын сессияларды жаратқанда өзимиздиң меншик механизмлеримизди ислеп шығыуымыз керек.

Сессияларда таймаут жоқлығы (Insufficient Session Expiration) – егер сессияның идентификаторы ушын хәрекет етиу ўақты шекленбеген ямаса ўақыт көп берилген жағдайларда бузыушы авторизацияның алдынғы мағлыұматлары арқалы рухсатты алыуы мүмкин. Бундай жағдайлар бир компьютерде бир неше пайдаланыушылар ислеген жағдайларда болады.

Программалаушы булл жағдайды есапқа алып сессияның хәрекет етиу ўақтың шекеуи керек.

Сессия фиксациясы (Session Fixation) – бузыушы бул класстағы хужимди орынлау ушын пайдаланыушының сессиясы идентификаторына өзиниң мәнислерин қосып жибереди. Бундай хужимлерге cookie ислетилген системалар қарсылығы төмен болады.

Веб-беттиң мәнислерин алмастырыу (Content Spoofing) – бул техниканы пайдаланған бузыушы пайдаланыушыны серверден сораған бетти

басқа деректен билдирмей жиберіуіге (дәслепки авторизация болатуғын бетти) хәрекет етеди.

Бундай хұжимлерден сақланыуы үшін фреймлардан пайдаланбау хәмде баслысы параметрлар арқалы қандайда керекли файлларға көрсетилген жолларды бермеу керек.

Сценарияларды сайтлар арасында орынлатыуы (Cross-site Scripting, XSS) – бул хұжим етиуішиге пайдаланыушының браузерине орынлатыуға сервер арқалы код жиберіу мүмкиншилигин береді. Бундай кодлар әдетте HTML/JavaScript, ActiveX, Java, Flash тиллери арқалы жазылады.

Сценарияларды сайтлар арасында орынлатыуы арқалы әмелге асырылатуғын хұжимлердің турақлы (сақланған) хәм турақсыз (қайтарылған) еки түри бар болып, екинши түри клиентке қайтарылатуғын бир HTTP сорауда әмелге асырылып, бириншиси бир неше сораулар натийжесинде әмелге асырылады.

Хұжимнің турақсыз түринде пайдаланыушы бузыушы тәрәпинен арнаулы жаратылған бағытқа (ссылкаға) өтиуі шөлкемлестиреди. Нәтийжеде URL ямаса сораунамаға жайластырылған бузыушы код пайдаланыушының браузерине берилип, өзинің жумысын ислеп баслайды.

Ал турақлы түринде бузыушы тәрәпинен жаратылған код серверге енгизилип онда қандайда уақыт аралығында сақланады. Бундай хұжимлер көбинше форумлар, чатлар хәмде Веб-интерфейске ийе почталар арқалы қолланылады. Бул жағдайда пайдаланыушы хұжимге ушырау үшін оның бағытты басыуы шәрт емес, сайтқа кириуінің өзи жеткиликли.

Булардан басқада хұжимди шөлкемлестириу үшін системаның әззи жерлеринен пайдаланыудың усыллары көп болып, көплеген хұжимлер web исленбелердің информация алмасыу сораунамалары идентификаторлары арқалы әмелге асырылады.

Сонлықтан Веб бетлерде сораунамалар жаратыуы тиллери, усыллары менен танысып өтемиз.

§2. ВЕБ СЕРВЕРЛЕРГЕ ЖИБЕРИЛЕТУҒЫН СОРАҰЛАРДЫ ЖАРАТЫҰ ТИЛЛЕРИ.

Веб исленбелерде сервер менен пайдаланыўшы арасындағы интерактивликти жаратыўда тийкарынан HTML тилиндеги гипербағытлар хэм формалардан пайдаланылады. Үлкен қолемдеги Веб исленбелерде, веб-серверлерге қойылған мекеме информациялық системаларында программаластырыў кодларын текте HTML ди пайдаланып жазыў үлкен қыйыншылықлар туўдырыўы себепли кейинги ўақытлары HTML кодларды қайта ислеўши бир-неше программалық тиллер хэм оларда программада қолланыў усыллары исленбекте. Бундай тиллер Веб-серверлерге орнатылып, оның нәтийжелери клиентке браузерлер түсинетуғын кодлар көринисинде жиберилип, клиенттен келген сораўнамалар HTML тилинің мүмкиншилиги шегарасында болады. Бундай тиллердің хәзирги ўақытта кең тарқалғаны болып PHP тили саналады.

PHP программаластырыў тилинде өзгериўшилер доллар (\$) белгиси менен басланады. Өзгериўши аты регистрге сезиўшең болады. \$username хэм \$Username атлары еки өз алдына өзгериўши атларын аңлатады. Өзгериўшиниң атаўда өзгериўши аты хәрип ямаса асты сызық белгиси, олардан соң белги –хәриплер топلامы ямаса асты сызық белгиси келиўи мүмкин. Мысалы,

```
<?php
$username='Iygilik';
$Username='Sitaev';
echo "$username $Username";
?>
```

Жоқарыдағы фрагментте еки өз алдына өзгериўши жаратылады. оннан соң нәтийжелерди шығарыў ушын echo операторынан пайдаланылады. PHP синтаксиси тәрәпинен қарағанда болар хәммеси абсолют туўры болады. Келтирилген сценарий нәтийжеси сыпатында Iygilik Sitaev шығарылады.

Регистрге сезгирлик бир тәрәптен карағанда жетерлише болып көринбейди, бірақ әмелиятта бул түрдеги шегараланыў программаның оқылыўын көп аңсатластырады.

Өзгериўши аты сыпатында кирилл хәриплерин де ислетиў мүмкин. Бірақ әмелиятта бундай жол тутыў айырым себеплерге көре мүмкин емес. Айырым халатларда программа оқылыўын көп қыйынластырады хәм түрли системаларда интерпретациялаў мүмкин болмай қалады.

Гилт сөзлер: басқа тиллерге уқсас PHP диң хәм бөлекленген сөзлер топلامы бар. PHP да өзгериўшилерге регитрның сезгирлигига байланыслы болсада, функция атлары хәм бөлекленген сөзлер ушын бул қағыйда белгиленбеген. Мысалы, while, WHILE, While хәм wHILe сөзлери арасында хеш кандай өзгешелик жоқ. 1- таблицада PHP диң бөлекленген гилт сөзлери санап өтилген.

1- таблица. PHP гилт сөзлери

and	break	case
class	continue	default
do	else	elseif
extends	false	for
foreach	function	global
if	include	list
new	not	or
require	return	static
swith	this	true
var	virtual	xor
while		

Мағлыўматлар типии: басқа тиллерге усаған PHP да хәм стандарт типлери бар:

- пүтин;
- қозғалыўшы точка типии

- логикалық
- массив
- класслық
- NULL типі

Бірақ қатаң типлестірілген тілдерден, мысалы С++ дан PHP диң тийкарғы өзгешелигі, өзгеріушілер тәрийпленбейди яки контекста қолланылыуына қарап анықланады.

```
$var=2; // $var – пүтин сан 2
```

```
$var="2"; // $var – 2 мәнисин сақлаушы қатар
```

```
$var=array[2]; //$var – 2 ге тең бир элементли массив
```

Өзгеріуші типі хәр сапар оған мәнис өзлестіріу уақтында анықланады. Мәнис өзлестірмеген өзгеріуші *байланыспаған* есапланады (unboard variable). Байланыспаған өзгеріуші NULL типине тийисли болады хәм текте NULL мәнисине ийе болады. Егер аңлатпада пүтин сан ислетилсе NULL мәнисі 0 ге, егер қатар болса ол жағдайда бос қатарға айландырылады.

isset() функциясы жәрдемінде өзгеріушіге анық бир мәнис өзлестірілгенлиги тексериледи. Параметр сыпатында бул функцияға өзгеріуші аты өзлестіриледи. isset()

функциясы логикалық мәнис береді. Мысалы:

```
isset($holiday)
```

функциясын шақырыу менен \$holiday өзгеріушісине мәнис өзлестірілген болса, NULL мәнисинен өзгешелиги true мәнисі алынады, кери жағдайда false алынады.

Төменде хәр бир типти толығырақ қарастырып шығайық.

Пүтин тип: PHP да текте бир integer пүтин типі бар болып, С++дағы long типі менен сәйкес келеді. Пүтин санды онлық, сегизлик, хәм он алтылық форма көринисінде жазыу мүмкин. 0x менен басланатуғын мәнислер он алтылық санлар, 0 менен ямаса-сегизлик. Мысалы,

```
$number1=1234; // пүтин сан
```

```
$number2=-123; // манфий сан
```

```
$number3=062; // 50 санының сегизлик көриниси
```

```
$number4=0xff; // он алтылық (255)
```

Пүтин сан өлшеми қолланылыу платформасына байланысly болып, көпшилик халларда 32 битке тең болады. Егер пүтин санлар үстінде әмеллер орынланып атырғанда пүтин сан аралығынан шығып кетсе, өз-өзинен қозғалыушы точкалы тип double ге айландырылады. Мысалы,

```
$a=1/2; // $a 0.5 ге тең
```

```
$a=2147483648; // $a -1 ге емес, яки 2147483648
```

Қозғалыушы точкалы тип: PHP дағы double типі C++ дағы типке аналог есапланады. Қозғалыушы точкалы типті әпиуайы болған экспоненционал ямаса аралас форма көринисинде точка (.) қойыу менен тәрийп қылыу мүмкин.

Басқаша сөз бенен айтқанда, .345 хәм 345. аңлатпалары PHP да абсолют коррект есапланады. Төменде қозғалыушы точкалы санға мысал келтирилген.

```
$a=12.34;
```

```
$a=12.3E4;
```

```
$a=12e-34;
```

Логикалық тип: Логикалық өзгеріуші еки мәнистен биреуін – шын (true) ямаса жалған (false) ды қабыл қылыуи мүмкин.

Логикалық өзгеріуші мәнисин true хәм false гилт сөзлер арқалы беріу мүмкин.

```
$boolValue=true; // $boolValue мәниси true ге тең. Бунда регистр хеш қандай рол ойнамайды.
```

Логикалық мәнислер логикалық әмеллердің нәтийжеси есапланады.

Мысалы,

```
if ($a>$b) { echo "a>b" }
```

Логикалық тип integer типине усаған әпиуайы болып, контекстте басқа типлер де ислетилиуи мүмкин.

Егер логикалық контекстте пүтин мәнісли аңлатпалар ислетилсе, ол жағдайда бул аңлатпа мәніси нолге тең болса, false мәніс , кери жағдайда true.

Төменде PHP диң санлар менен ислеўши стандарт функцияларына мысал келтирилген.

```
<?php  
echo log(exp(1))."<br>";  
echo sin(pi()/2);  
?>
```

Әлбетте PHP да стандарт функциялардан пайдаланыў менен ислетпейди. Функциялар алгоритмлик композицияларда күшли қурал болып хызмет қылады.

Қатарлық типлер: PHP да қатар бул- байтлар избе-излиги болып саналады. PHP да Unicode ни қоллаўшы қурал жоқ. Қатарлық константаларды (литераллар) биреўлик ямаса екеўлик қос тырнақ ишинде анықлаў мүмкин.

Арнаулы символлар (мысалы, / хәм //) қатарлық литералларда бир қос тырнақ ишине алынған болса, ажыратылмайды. Бул стандарт өзгериўшилерге де тийисли. Бундай константалар шығарыў потоқында хәр қандай өзгеристе де шығарылады. Еки қаўыс ишине алынған қатарлық константалар басқаша көринис алады. Айырым халларда көрилген өзгериўшилер PHP интерпретаторы тәрәпинен корректив ажыратылады хәм олардың орнына мәніслери шығарылады.

Мысалы, төмендеги қатар

```
'The age is: $age'
```

Егер \$age ниң мәніси 12 ге тең болса, ол жағдайда төмендеги қатар шығарылады.

```
The age is: 12
```

Төменде және бир неше мысаллар келтирилген

```
echo 'Бул қатар'; //Бул қатар ды шығарады
```

```
echo ‘Бул “қатар” \’қатарда\’; // бул “қатар” ‘қатарда’ ны шығарады
echo ‘a\nb’; //a\nb ны шығарады
```

Биринши қатарда келтирилген фрагментке түсіндириу келтириу шәрт емес. Екинши қатардағы мысалда қатар биреулик қостырнақ ишине алынған. Сонлықтан ушын бул қатар ишинде қорқпастан екеулик қостырнақты ислетиу мүмкин, бирақ биреулик қостырнақты маскировка қылыуға тууры келеди. Үшиншиси биреулик қостырнақ ишине алынған қатарда арнаулы белгин шығарылып атыр. Көрип турғаныңыздай, шығарылыуы керек болған белги орнына \n қатарды көшириу белгиси шығарылып атыр.

Егер қатар екеулик қостырнақ ишине алынған болса, ол жағдайда төмендеги арнаулы белгилерди ислетиу мүмкин:

\n	қатарды көшириу
\r	каретканы қайтарыу
\t	табуляция белгиси
\\	кери слеш
\\$	доллар белгиси
\”	қос қос тырнақ

Қалған барлық ҳалларда кери слеш ҳеш нәрсени аңлатпайды ҳәм әпиуайы белги сыпатында қатнасады.

Өзгериуши атын фигуралы қауыс ({}) ишине алып шегаралау мүмкин. Мысалы, ‘ белгисин идентификаторда ислетиу мүмкин болмасада, төмендеги фрагмент нәтийжеси

```
$beer='Heineken';
echo “$beer’s taste is great”;
төмендегише болады
Heinken’s taste is great
Мине және бир мысал
Echo “He drunk some $beer”;
Echo “He drunk some $(beer)s”;
```

Биринши қатарда \$beer өзгеріуішисиниң мәнисі шығарылмайды, себеби s белгиси идентификаторга тийисли болғанлықтан интерпретатор тууры шешимди анықлай алмайды. Екинши қатарды қайта іслеу нәтижесинде beer өзгеріуіши мәнисі нәтижелеуіши қатарға шығарылады.

Қатар үстінде жийе іслетилетуғын функциялар

Функция аты	Параметр типі	Беретуғын мәнисі
Strlen	қатар	Қатардағы белгилер саны
Strcmp	еки қатар	0, егер еки қатар бирдей болса; -1 , егер екинши қатар биринши қатардан лексикографик үлкен болса; 1, егер екинши қатар биринши қатардан лексикографик кишкене болса
Strpos	еки қатар	Жайласуы орны 2-параметр менен берилген 1- қатардағы қатар асты .Егер үшінши параметр(пүтин) берилген болса қатар астының узынлығын билдиреди
Chop	қатар	Оңға қарап өширилген барлық бос орынлардан ибарат қатар
Trim	қатар	Оңға хәм шепке қарап өширилген барлық бос орынлардан ибарат қатар
Ltrim	қатар	Шепке қарап өширилген барлық бос орынлардан ибарат қатар
Strtolower	қатар	Барлық бас хәриплери кишкене хәриплерге өткизилген қатар
Strtoupper	қатар	Барлық кишкене хәриплери бас хәриплерге өткизилген қатар

Массивлер: РНР тилинде массивлер “мәнис /гилт” жуплықларының тәртіпленген топламлары бар. Бул мағлыұматлар типі сондай оптималлаштырылған болып, оны әпиұайы массив, дизим (вектор),

коллекция, стек, нәўбет, терек хәм басқа мағлыўматлар стуруктураси сыпатында ислетиў мүмкин. Және айтып кететуғын нәрсе, PHP тилинде массивлердиң индекси сыпатында текте пүтин санлар емес ал қатарларды да ислетиў мүмкин. Мысалы:

```
ram[fam][1]='Burhanov'; ram[fam][2]='Utewliev';  
ram[ati][1]='Shayaxmet'; ram[ati][2]='Nietbay';
```

Типлерге келтириў: Айтып өткенимиздей, PHP да өзгериўшиниң анық бир типке тийисли екенлиги белгили емес.

Тип сценарий орынланыўы даўамында анықланады хәм контекстте өзгериўшиниң қандай ислетилиўине байланыслы рәўиште өзгерттириў мүмкин.

Соның ушын сол өзгериўшиниң өзінде түрли типлерге тийисли мағлыўматларды сақлаў мүмкин, мысалы қатар хәм сан

```
$id=123;  
$id="abc1";
```

PHP арифметикалық логикалық хәм қатар типине тийисли өзгериўшилерди басқа типлерге автоматик өзгерттириў имканиятын береді. Бунда өзгериўши өз типин өзгерттирмейді.

```
$a=1; // $a- пүтин сан  
$b=1+$a/2; // $b – 1.5 ге тең  
$c="10 студент"+$b ; // $c – 11.5 ге тең
```

Типке келтириўди төмендеги функциялар арқалы әмелге асырыў мүмкин.

- (int), (integer) – пүтин санға келтириў
- (bool), (boolean) – логикалық типке
- (real), (double), (float)- қозғалыўшы точкалы

- (string) – қатарға
- (array) – массивке айландырыу
- (object) – объектга айландырыу

Өзгериуішени типке келтириуі үшін `settype(Var, Type)` функциясынан хам пайдаланыуі мүмкин. Бул функция еки параметрге ийе: өзгериуіши аты хәм тип. Мысалы оны төмендегише қолланыуі мүмкин.

```
$var1='2test';
$var2=false;
Settype($var, 'integer');
Settype($var, 'string');
```

Мағлыұматларды шығарыуі кураллары: PHP интерпретаторы жәрдемінде генерация етилетуғын PHP сценариясына кирген қәлеген мағлыұматлар HTML бетиниң бир бөлеги есапланады. Басқаша айтқанда барлық кириуіши мағлыұматлар HTML код көринисинде болыуы шәрт.

Шығыуышы мағлыұматларды үш усылдан бири арқалы генерация қылыуі мүмкин: `echo`, `print()` хәм `printf()` функциялары жәрдемінде орынланады. `Echo` функциясын қауыс жәрдемінде де қауыссыз да жазыуі мүмкин. Қауыс пенен жазылғанда бул функцияға бир қатарлық параметр бериуі мүмкин. Кери жағдайда бул функция анық узынлықдағы параметрлер дизимин бериуі мүмкин. Мысалы, төмендеги қатар абсолют коррект есапланады:

```
Echo "Oranges are taste <br/>", "Apples are too <br/>";
```

Бирақ төмендега қатар коррект емеслиги көринип турыпты

```
Echo ("Oranges are taste <br/>", "Apples are too <br/>");
```

`Echo` функциясы хеш қандай мәнис қайтармайды.

`Print` функциясы болардан өзгешеликли рәуиште оған текте бир параметр бериуі мүмкин. Буннан басқа бул функция мәнис береді. `echo` функциясына уқсас қауыс жазыуі шәрт емес. Беретуғын мәнис талап қылынған әмел

туұры орынланғаны хаққында мағлыұмат береді. (1 болса хәммеси туұры, хәм 0 де кери жағдай).

Print функциясына да қатарлық параметр бериў мүмкин. Егер узатылатуғын мәнис басқа типке тийисли болса мәнис автомат string ке айландырылады.

Printf функциясы С тилиндеги функция менен абсолют уқсас есапланады. оны дәстүрши шығыўшы мағлыұматларды пүтинлигише басқарып болмайтуғын ўақытта пайдаланыў қолай есапланады.

Резюмлар

- Басқа тиллерден өзгешеликли рәўиште РНР өзгериўшилери анық анықланбайды. Олардың типі өзгериўши контекстте ислетилиўине қарап автоматик анықланады. Орынланыў процессинде өзгериўши типі өзгериўи мүмкин.

- РНР тилинде стандарт функциялар топламы бар: integer, double, boolean, string. Түрлі типтеги мағлыұматлар менен ислеўде түрлі стандарт функциялардан пайдаланыў мүмкин.

- Шәртли рәўиште РНР интерпретаторы талап қылынған типке көре өзгериўшиниң типин наанық өзгертиреди. Бундай келтириўди анық орынлаў да мүмкин. Бул мәселени арнаўлы функциялар арқалы да шешиў мүмкин.

- Басқа тиллерге усаған РНР да мағлыұматларды шығарыўдың бир неше усулларына ийе. Бирақ соны естен шығармаслық керек, РНР сценарийлери HTML-бетниң динамик генерациясына байланыстырылып қойылады.

Әмеллер: Әмеллер бир ямаса бир неше өзгериўшилер манипуляциясы ушын қолланылады. Әпиўайы мысал сыпатында арифметикада қосыў әмели мысал болады. РНР тили әмеллери тийкарынан С тили әмеллерине уқсас.

Әмеллердиң бир неше группалары бар.

- Өзгериўшилер үстинде әпиўайы математик әмеллерди орынлаў ушын арифметикалық әмеллер
- Өзлестириў әмеллери, айырым өзгериўшилерин мәнисин өзгертиў ушын
- Бўлевой аңлатпаларды есаплайтуғын логикалық әмеллер
- Санды екилик көринисинде авңлататуғын разрядлы әмеллер
- Конкатенация (бул әмел библиотека функциялары менен биргеликте әмелге асырылады)
- Қәтеликти баҳалаў
- Шәртли әмеллер

Төменде 4 таблицада РНР тили тийкарғы әмеллери олардың дәрежесине карап санап өтилген:

4 таблица. РНР тили тийкарғы әмеллери

Әмеллер	Жазыў
()	дәрежени анықлаў
[]	индекс бойынша массив элементине мүрәжат
!	логикалық бийкарлаў
~	разрядлы бийкарлаў
++,--	инкремент, декремент
@	қәтеликти кемейтириў
* / %	көбейтиў, бөлиў, қалдықлы бөлиў
+ - .	қосыў, алыў, конкатенация
<< >>	разрядлы шепке хәрекет, оңға хәрекет
< <= > >=	кишкене, кишкене ямаса тең, үлкен, үлкен ямаса тең
= = != = = = != =	тең, тең емес
&	разрядлы хәм
^	разрядлы ямасадан басқа
	разрядлы ямаса
&&	логикалық хәм

	логикалық ямаса
?:	шәртлі әмел
= += -= *= /= .=	өзлестіріу
and	логикалық хәм
xor	логикалық ямаса дан басқа
or	логикалық ямаса

Арифметикалық әмеллер: PHP тили C- ге ұқсас программаластырыу тиллерінде бар болған арифметикалық әмеллер топламы бар. Егер +, -, хәм * операндлери пүтин санлардан ибарат болса нәтийже хәм пүтин саннан ибарат болады. Егер операндлардан бири қозғалыушы точкалы болса, ол жағдайда нәтийже де хәқыйқый саннан ибарат болады. Бөлиу әмели усыған ұқсас интерпретация қылынады. Егер пүтин санлар үстиндеги әмеллер орынланыуы нәтийжесінде толып жүз берсе, ол жағдайда нәтийже сан екилик анықлықдағыға (double) айландырылады. 5 таблицада PHP тили арифметикалық әмеллери келтирилген.

5 таблица. Арифметикалық әмеллер

Әмеллер	Жазыу
$a+b$	a хәм b ның қосындысы
$a-b$	a хәм b лардың айырмасы
$a-b$	a хәм b лардың айырмасы
$a*b$	a хәм b лардың көбеймеси
a/b	a хәм b лардың тийиндиси
$a\%b$	a хәм b ның қалдықлы тийиндиси

Өзлестіріу операторлары: Улыуа жағдайда $a++$ аңлатпасы “ a өзгеріушиси есаплауда нәтийжени өзлестирмекте” деп түсинеди. Төмендеги $a++$ типіндеги буйрықларды төмендегише түсиніу керек. Биринши a өзгеріушисиниң мәниси алынады, кейин оған 1 саны қосылады, нәтийже оператив ядта сақланады. оннан соң нәтийже ядтан шығарып алынады, нәтийжеде мәниси 1 ге асқан a өзгеріушиге өзлестіриледі.

PHP да қолайлық үшін бір неше ишки өзлестіріу операторлары анықланған болып, олар өзлестіріуден басқа ұазыйпаларды да орынлайды. Импанияты барынша бул операторлардан көбірек пайдаланыу керек, олар интерпретаторда буйрықлардың эффектив орынланыуын тәмийнлейди. Ишки операторлар хәм оларга эквивалент әмеллер 6 таблицада келтирилген.

6 таблица. PHP ишки операторлары

Аңлатпа	Эквивалент
$a+=b$	$a=a+b$
$a-=b$	$a=a-b$
$a*=b$	$a=a*b$
$a/=b$	$a=a/b$
$a.=b$	$a=a.b$
$a\%=b$	$a=a\%b$
$a\&=b$	$a=a\&b$
$a!=b$	$a=!b$
$a\sim=b$	$a=\sim b$
$a<<=b$	$a=a<<b$
$a>>=b$	$a=a>>b$

Логикалық әмеллер: Логикалық әмеллер 7 таблицада келтирилген.

$\&\&$ хәм `and`, $\|\|$ хәм `or` операторлары текте дәрежеси менен өзгешелик қылады.

Разрядлы әмеллер: Разрядлы операторлар логикалық орынлау үшін қолайланған, сондай ақ айырым өз алдына операнд битлери үстинде `1` бит мәніси `true` менен сәйкес келеди, `0` мәніси `false`.

7 таблица. PHP дың логикалық әмеллери

Аңлатпа	Әмел аты	Нәтийже
$a \&\& b$	Логикалық хәм	true, егер a хәм b true болса
$a \ \ b$	Логикалық ямаса	false, егер a хәм b false болса
$!a$	Логикалық жоқ	true, егер a false болса; false егер a true

\$a and	Логикалық хәм	&& ге аналог
\$a or \$b	Логикалық ямаса	ге аналог
\$a xor \$	Логикалық ямасаға кери	true егер a хәм b лардың мәнислери хәр қыйлы болса

Инкрементлеў хәм декрементлеў: Инкрементлеў (декрементлеў) – бул өзгериўши мәнисин 1 ге асырыў (кемейтириў) ди билдиреди.

Инкрементлеў/декрементлеў префиксли хәм постфиксли формалары- өзгериўши атынан алдын ямаса кейин қойылған әмел белгисинен парықланады (8 таблица) .

8 таблица. РНР тилинде инкрементлеў хәм декрементлеў

Аңлатпа	Нәтийже
\$a++	\$a ны береди хәм оннан соң \$a ны 1 ге асырады
\$a- -	Уқсас бирақ 1 ге кемейеди
++\$a	\$a ның мәниси асады хәм оннан соң \$a ның мәниси алынады
-- \$a	Уқсас бирақ 1 ге кемейеди

Басқа әмеллер: () Әмеллер орынланыў приоритетин өзгерттириў әмели . Төртмүйеш қаўыс ишине алынған аңлатпа биринши орынланады.

[]

Бул әмелди ислетип массив индекси жәрдемінде оның элементи мәнисин алыў мүмкин.

@

Қәтеликти кемейтириў әмели. Егер қатарды орынлаў даўамында бул белги менен қәтелик жүз берсе, хабар диагностикасы ислемейди. Мысалы, @echo(5/0) қатар интерпретация хеш қандай нәтийже бермейди. @ белгисиз ислетилген тап сол қатар қәтелик ҳаққында хабар береди.

Қатарларды конкатенациялаў (бирлестириў) әмели, бир қатарды басқа қатар ақырына жазып қояды. Мысалы, echo(“My name is ”.”Michael”); буйрығы нәтийжеси My name is Michael болады.

Басқарыушы операторлар: РНР диң қәлеген сценарийси аңлатпадан ибарат. Аңлатпа қурамына өзлестириу әмели, функцияны шақырыу, сондай ақ басқарыушы операторлар (шәртли операторлар, цикллар, тармақланыушы блок хәм басқалар). Қағыйда бойынша хәр бир аңлатпа точка үтир(;) менен тамамланады. Аңлатпалар группаластырылған болыуы мүмкин хәм олар қурамалы аңлатпаны дүзеди.

f-else-elseif операторы: Бар конструкция қәлеген программаластырыу тилиниң тийкарғы қуралларынан бири есапланады. Бул басқарыушы операторлар берилген шәртке көре анық бир фрагмент кодын орынлауы ямаса таслап кетиуи мүмкин. if операторының улыуа көриниси төменде келтирилген.

```
if ( 1 шәрт ) {  
  1 код блогы  
} elseif ( 2 шәрт ) {  
  ...  
} elseif (n шәрт) {  
  N код блогы  
} else {  
  Алтернатив блок коды  
}
```

1 шәрт шын болғанда (нәтийже true болса), хәш болмағанда *1 код блогы* орынланады. Кери жағдайда *2 шәрт* орынланады хәм х.к. Егер барлық шәртли аңлатпалар жалған болса, ол жағдайда *else* операторы жәрдемінде анықланған *алтернатив блок коды* орынланады. Мысалы, төмендегише.

```
If ( $a > $b ) {  
  Print “a is bigger then b”;  
} elseif ( $a = $b ) {  
  Print “a is equal b”;  
} else {
```

```
Print "b is bigger than a";
```

```
}
```

PHP тилинде сондай ақ, тернар әмели ?: төмендеги синтаксиси бойынша ислетиледи

```
((1 аңлатпа)?(2 аңлатпа) : (3 аңлатпа))
```

Бул конструкция төмендегише интерпретация қылынады. Егер 1 аңлатпа шын болса, барлық аңлатпалар мәніси 2 аңлатпаға тең болады. Кери ҳалда(сондай ақ 1 аңлатпа жалған болғанда да) барлық аңлатпа мәніси 3 аңлатпаға тең болады.

While циклы:

Цикл көриниси төмендегише

```
While (шәрт) {
```

```
Код блогы
```

```
}
```

Код блогы шәрт жалған болғанға шекем орынланады. Шәрт барлық ўақытта цикл басында тексериледи. Егер шәрт биринши итерациядан false мәніске ийе болса, ол жағдайда цикл бир мәрте де орынланбайды.

Төменде 1 ден 10 ға шекем болған санларды шығарыў ушын while циклына мысал келтирилген.

```
<?php
```

```
$i=1;
```

```
While ($i<=10) {
```

```
Echo $i++;
```

```
}
```

```
?>
```

do-while циклы: бул цикл да жоқарыдағы циклға уқсас болып, бунда шәрт цикл денеси орынланғаннан соң тексериледи. Басқаша қылып айтқанда цикл операторлары ҳеш болмағанда бир рет орынланады.

```
Do {
```

```
Блок коды;
```

```
} while (шәрт);
```

do-while циклының ислетилиўине мысал.

```
$i=10;
```

```
do {
```

```
print $i;
```

```
}
```

```
while ($i>20);
```

Келтирилген мысалда шәрт орынланбағанда \$i өзгериўшиси ҳеш болмағанда бир мәрте орынланады.

For циклы: PHP да курамалы болған циклларды for операторы жәрдемінде орынлаў мүмкин. Буның ушын төмендеги цикл ислетиледи.

```
For (1 аңлатпа; 2 аңлатпа; 3 аңлатпа) {
```

```
Код блогы;
```

```
}
```

1 аңлатпа биринши итерацияда ҳеш қандай шәрт тексерместен орынланады. Әдетте бул аңлатпа өзгериўши - счётчикти инициализация (жаратыў) ушын ислетиледи. Цикл басында барлық ўақытта *2 аңлатпа* тексериледи. Егер шын болса, код блогы орынланады, кери жағдайда цикл орынланыўы тоқтатылады. Хәр-бир итерация ақырында *3 аңлатпа* орынланады. Хәр бир аңлатпа бос болыўы мүмкин. Кең тарқалған мысалларды келтиремиз. For операторы жәрдемінде түрли усыллар менен 1 дан 10 ға шекем болған санларды шығарыў.

1 мысал(классикалық усыл)

```
for ($i=1; i<=10; i++) {
```

```
echo $i;
```

```
}
```

2 мысал

```
for ($i=1; $i++) {
```

```
if ($i>10) break;
```

```
echo $i;  
}
```

3 мысал.

```
$i=1;  
for (; ;) {  
if ($i>10) break;  
echo $i;  
$i++;  
}
```

4 мысал.

```
For ($i=1; $i<=10; echo $i; $i++) { }
```

Келтирилген мысаллардың түрлі көриністеріне карамай “классикалық” усылды қолланыў мәслахат бериледи.

Foreach операторы: бул оператор PHP диң 4 версиясында пайда болды. Ол берилген әмелди массивтиң барлық элементлери үстинде орынлайды. Соның ушын бул пунктқа және бир мәрте қайтыўға туўры келеди. Foreach операторының қолланыў ушын төмендеги синтаксис ислетиледи.

```
Foreach (массив as $value) {  
}  
foreach (массив as $key=>$value) {  
}
```

Биринши жағдайда хәр бир итерацияда массивтиң берилген элементлерине \$value өзгериўшиси меншиклестиреди, массивтиң берилген көрсеткиши бир элемент алдынға жылжыйды. (тап сондай кейинги итерацияда берилген өзгериўши массивтиң кейинги элементине меншиклеседи). Екинши жағдайда да тап сондай, бирақ массивтиң гилт мәнисине \$key өзгериўшисине меншиклеседи. Мысалы, төмендегише:

```
<? php  
$arr=array('one', 'two', 'three');
```

```
foreach($arr as $value) {
echo 'Value: ' . $value . ' <br>';
}
?>
```

Келтирилген мысалда массив элементлериниң мәнислери шығарылады.

Switch операторы: Берилген оператор if тиң инструкциясына жүдә уқсас. Айырым ҳалларда бирдей, сонлықтан сол өзгериўшиниң мәнисин бир неше басқа мәнислер менен салыстырыўға туўры келеди. Ол жағдайда with операторынан пайдаланыў жақсырақ болады.

```
<?php
switch ($value) {
case 'apple':
    echo 'This is an apple!';
case 'pear':
    echo 'This is an pear!';
default:
    echo 'Hm . . Who knows . . . ';
} ?>
```

Егер \$value өзгериўши мәниси apple ямаса pear ге тең болса, сәйкес код блоклары орынланады.

Егер мәнис санап өтилгенлерден биреўинеде тең болмаса, default гилт сөзи арқалы берилген блок коды орынланады.

Break хәм continue операторлары: Break операторы цикл орынланыўын тоқтатыў ушын хәм басқа for, foreach, while, do...while, with операторлары жәрдемінде көрилген конструкциялардың орынланыўын тоқтатады.

Continue операторы цикл ишинде қолланылады хәм берилген итерацияның қалған бөлегин тоқтатыў хәм кейинги итерацияға өтиў ушын ислетиледи.

PHP жәрдеминде MySQL мағлыұматлар базасына байланысыұ:

PHP да `mysql_connect` функциясы сценариясын MySQL менен байланыстырады.

```
mysql_connect ( "localhost" , "username" ,  
"password" );
```

Бул функция үш лазым болмаған параметрге ийе. Бул параметрлерди қозғамаслық қағыйдасы бойынша ислетиұ ушын функция аргументлери көрсетилмейди.

```
$db=mysql_connect();
```

Анық базаның өзине байланысыұ ушын `mysql_select_db()` функциясынан пайдаланылады.

```
mysql_select_db("cars");
```

MySQL серверине сораұлар `mysql_query` функциясы жәрдеминде бериледи. Сораұ дәслеп қандайда бир қатар өзгериұшисине өзлестирилген болыұы керек.

```
$query="select *from mashina"
```

```
$result=mysql_query($query);
```

Көпшилик ҳалларда нәтийжелеұши мағлыұматларда қатарлар санын билиұ керек болады. Буның ушын `mysql_num_rows` функциясынан пайдаланылады.

```
$num_rows=mysql_num_rows($result);
```

Мағлыұматлар топламынан өз алдына қатарларды түрли усыллар жәрдеминде ажратып алыұ мүмкин. Мысалы `mysql_fetch_array` функциясы жәрдеминде, кейинги қатардың хеш-кодын береди. Егер жазыұ басқа бар болмаса, `false` мәнисин береди.

Төмендеги мысалды көремиз:

```
<html>
```

```
<body>
```

```
<?php
```

```
$db = mysql_connect("localhost", "гость");
```

```

mysql_select_db("Xodim",$db);
$result = mysql_query("SELECT * FROM xodim",$db);
if ($myrow = mysql_fetch_array($result))
{
echo "<table border=1>\n";
echo "<tr><td>Name</td><td>Holat</td></tr>\n";
do
{
printf("<tr><td>%s          %s</td><td>%s</tr>\n",          $myrow["first"],
$myrow["last"],
$myrow["address"]);
}
while ($myrow = mysql_fetch_array($result));
echo "</table>\n";
}
else
{
echo "Қатар жоқ";
}
?>
</body>
</html>

```

Биз мағлыұматлар базасынан мағлыұматларды алыұды үйрендик, енди усы жумыстың кериси, мағлыұматларды мағлыұматлар базасына жазыұды көрип өтейик. Буның ушын әпиұайы форма жаратамыз.

```

<html>
<body>
<form method="post" action="<?php echo $PHP_SELF?>">
Аты :<input type="Text" name="first"><br>

```

```

Фамилияңыз :<input type="Text" name="last"><br>
Адресиңиз :<input type="Text" name="address"><br>
Жумыс орныңыз:<input type="Text" name="ish"><br>
<input type="Submit" name="submit" value="Мағлыўматларды жиберийў">
</form>
</body>
</html>

```

Енди формага мағлыўмат кирилгенлигин тексерийўши дастур дүземиз. Бул ҳәзирше әпиўайы тексерийў болып, мағлыўматлар жиберетуғын өзгериўшилер \$HTTP_POST_VARS өзгериўшиси жәрдемінде экранға шығарылады.

```

<html>
<body>
<?php
if ($submit) {
// process form
while (list($name, $value) = each($HTTP_POST_VARS)) {
echo "$name = $value<br>\n";
}
} else{
// display form
?>
<form method="post" action="<?php echo $PHP_SELF?>">
Аты:<input type="Text" name="first"><br>
Фамилиясы:<input type="Text" name="last"><br>
Адреси:<input type="Text" name="address"><br>
Жумыс орны:<input type="Text" name="ish"><br>
<input type="Submit" name="submit" value="Мағлыўматларды жиберийў">
</form>
<?php
} // end if

```

```

?>
</body>
</html>
Енди киритилген мағлыўматларды базаға жазамыз.
<html>
<body>
<?php
if ($submit) {
// киритиў формасы
$db = mysql_connect("localhost", "гость");
mysql_select_db("mydb",$db);
$sql = "INSERT INTO isshi(ati,last,address,jumis) VALUES
('$ati','$last','$address','$ish')";
$result = mysql_query($sql);
echo "Мағлыўматлар киритилди.\n";
} else{

?>
<form method="post" action="<?php echo $PHP_SELF?>">
Аты:<input type="Text" name="first"><br>
Фамилиясы:<input type="Text" name="last"><br>
Адреси:<input type="Text" name="address"><br>
Жумысы:<input type="Text" name="ish"><br>
<input type="Submit" name="submit" value="Мағлыўматларды жибериў">
</form>
<?php
}
// end if
?>
</body>
</html>

```

§3. ВЕБ СЕРВЕРЛЕРДИ ХӘМ МАҒЛЫҰМАТЛАРДЫ ҚӘҰИПЛИ СОРАҰНАМАЛАРДАН ҚОРҒАҰ УСЫЛЛАРЫ.

Хәзирги ўақытта PHP алдындағыдай динамикалық Web – серверди ислеп шығыўда кең қолланылып атыр. Статистикалық (static) Web – бетлер барлық қолланыўшыларға бирдей хәмде интерактивлик өз-ара хәрекет қолайлықлары менен тәмийнлемейди. Өз гезегинде, динамикалық (dynamic) бетлер усындай ўазыйпалар менен тәмийнлейди. Қолланыўшының киргизген информацияларына қарап браузерде хәр қыйлы бетлер көриниўи мумкин. Мысалы: айырым Web – сервердиң ресурсларына кириўден алдын, қолланыўшы өз атын ямаса паролин киргизиўи керек болады. Соның менен бирге тармақ ылайыклы профилге тийкарланып сол қолланыўшыға сазланады. Қолланыўшы интерактив каталогтан товардың керекли турин хәм тек ғана керекли информацияны алыўи мумкин.

Динамикалық Web-бетлер HTML-форманың жәрдемінде қолланылудан керекли информацияларды алады. Форманың майданына киргизилген мағлыўматлар қайта исленеди, усы қайта ислеудин нәтийжесине байланыслы қолланыўшыға хәр қыйлы Web-бетлер берилиўи мумкин. Соның менен бирге берилген қайта ислеўлер хәр қыйлы бетлерди шығарыў ушын шәртли операторларда сақланыўи ямаса қолланыўи мумкин.

Бул бөлимде формаларды дузиу ушын қолланылатуғын HTML дескрипторлары көрип шығылмайды. Сиз усылар менен таныссыз деп есапланады. Соның менен бирге бул бөлимде Web-тармақтың қәуипсизлигин тәмийнлеудин сораўлары, формаларды көриниске келтиретуғын PHP тилин қолланыў хәм қолланыўшының информацияларын қайта ислеўлер көрип шығылады.

Web-қосымшалар бузыўшылардың хужимине жудә көп мәрте ушрайды. Көпшилик Web-қосымшалар ашық серверлерде жайласады хәм тоарлардың хызметлер топламын ямаса қәлеген кириўшиге информацияларды береді. Информацияларды өз кириўшилеринен

алатуғынлығы себепли, динамикалық Web-тармақлар айрықша нәзік екен. Көпшилик қолланыушыларымыз информатсион ресурслардан жақсы нийетте пайдаланыушы нормал адамлар болыуына қарамастан, олардың ишинде жаман нийеттегилерде бар. Адетке көре оларды төмендеги категорияларға бөлиуге болады:

Зыянкеслер, кредит карточкалардың номерлери бар файлларды хәм басқа конфиденциал, мәселен пароллари бар информацияларды табыуға хәрекет қылатуғынлар.

Бузыушылар, Web-тармақларды бузыуға хәрекет етиушилер. Олар Web-тармақты бузыу ямаса өзиниң шеберлигин көрсетиу ушын зыян келтириуди өзлерине ермек деп есаплайды.

Қолланыушыға зыян келтириуге хәрекет етиуши қастыяншылар, Web-серверлерге оларға кириушилерге зыян келтириуши қуралларды жайластырады.

Кейинги тийкарғы ескертиулер Web-қосымшаның сақланыу дәрежесин тәуир-ақ көтериуге рухсат береди, бирақ егерде ол ҳақыйқаттанда керекли хәм жасырын информацияларды қайта ислеу ушын қолланылыуы керек болса, тийисли китапларға ямаса экспертлерге хабарласқан жақсы.

Web-қосымша орнатылған компьютерлердин қәуипсизлигин тәмийнлеу. Бул системалық администратордың жууапкершилиги көлемине киреди.

Информацияларға кириудин шегараланыуы. Берилип атырған информацияның көлеми берилип атырған хызметтиң ямаса шешилип атырған мысалдың көлеминен улкен болмауы керек. Информатсияны Web арқалы арқайын шығарып алмайтуғындай етип сақлау керек.

Қолланыушыдан информацияны алып атырғандағы сақлық. Сизиң қосымшаңыздың қарыйдарлық бөлиміндеги тийкарғы мағлыұматларды жақсылап тексеру керек.

Web-серверди ең қорғанылған тәртипте функциялауға сазлау, бул қосымша ғамхорлықты талап етеди, егер информация жудә керекли жасырын

болса, кеткен күшлер соған ылайық. Бул сораулар анағұрлым кеңрек кейинде көріледі. Web-тармақ орнатылған компьютерди қауіпсіздікпен тәміяленуі.

Web-қосымшаны қорғаудың, бірінші жолы болып ол орнатылған компьютерди қорғаныс менен тәміяленуі болып табылады. Системалық администратор әдетте бул мәселені шешуіге жууапкер болып табылады. Соның ишінде брандмауэр орнатуға, шифрлау/парольди ашу режимин активлестіруіге, сканерлеудің тасландықтарынан пайдаланылуға хәм т.б. болады. Көплеген жағдайларда Web-қосымшаны іслеп шығушылардың бірду-бірүиде системалық администратор болмайды. Көплеген жағдайларда Web-қосымшаны уй системалық администратор болмайды. Бірақ қалеген жағдайда қауіпсіздікті тәміяленуі үшін жақынлаудың пухта анализин бире системалық администратор менен, бирде Web-қосымшаны күшейтетуғын ресурсларды беретуғын Web-хостинглық компания менен өткеруі керек.

Информацияға кируіди шеклеу

Web-тармақта сақланатуғын информацияларға кируі шекленген болуы керек. Хеш қандай гумансыз, қолланыушылар көриу үшін таярланған Web-бетлер ашық хәм Web-сервердың тийисли көпшилик киретуғын каталогинде болуы шөрт. Соның менен бирге қолланыушы сақланып турған файллардың атларын көриудің зөрүрлиги жоқ. Сиз, файллардың дизими берилген кавалог көрсетилген Web-тармаққа кирген шығарсыз. Уыума жағдайда бундай кылмау керек, себеби кируішилер хәр қандай алып тасланған файлды көре алады. Файллардың дизими, URL-адресте мақсетли файлдың аты көрсетимеген болса, ал жасырын турде көрсетилип атырған файл каталогте жоқ болған жағдайда көрсетиледи. Әдетте Web-сервер бундай жағдайда белгили файлларды шығаруы үшін, егерде аты айқын көрсетилмеген болса усындай тәртіпте қурылады. Қарсы жағдайда қолланыушы файллардың дизимин көре алады, бул болса оғада кеуилдегидей емес. Жақсысы Web-серверди сондай тәртіпте қуруу керек,

каталогқа кириуге хәрекет қылған уақытта, каталогқа кириуге мумкиншилик жоқлығы ҳаққында хабар шығыуы керек.

You don't have permission to access/secretdirectory on this server.
(Кириүқадаған. Каталогкекириүгеҳаққыңызжоқ./secretdirectory усы серверде).

ApacheWeb-серверин httpd.conf конфигурацион файлдын Indexes директивасы жәрдемінде тәртиплеуге болады.

OptionsIndexes// Каталог файлларын көрсетиуге рухсат береді.

Options-Indexes// Каталог файлларын көрсетиүди қадағанлайды.

Басқа Web-серверлерди тәртиплеуде тийисли документацияларға карауға туура келеді. Оннан қалса, файлларға аңсат шешилетуғын атларды коймау керек. Мысалы, егер Web-қосымшада қолланыушының пароли менен файл ислетилип атырған болса, оған passwords.php атын берюү жудә жақсы идея емес. Оның орнына бундай файлға тусиниксиз ямаса алдаушы ат берюү керек. Мәселен, VegetableRecipe.php. Бул, файлларға анық атларды берюү керек деген қағыйдаға тууры келмейди, деп қарсы шығыуы мумкин. Бирақ бул жағдай болып табылады. Айырым қастыяншылар URL-адрес браузерине www.yoursite.com/password.html киритип, буннан не шығады екен деп кутеди.

Web-тармақ оғада көплеген информациялар көпшиликке арналмаған. Мысалы, мағлыұматлар базасы ашық каталогта берилмеуи керек. Тийкарынан бундай информацияны Internet аркалы алыуға, кириуге болмайтуғын басқа компьютерде сақланған жақсы.

Қолланыушыдан информацияны алыудағы сақлық

Қолланыушылар тосаттан ямаса қасттан формада қәуипли (кеуилсиз) информацияны киргизиуи мумкин. Соның ушын оны сақлаудан ямаса қолланыудан алдын әдеплиликке хәм қәуипли символлар яки қатарлар жоқлығын тексерюү керек. Киргизилген информация (хәтте тосаттан) сценарийге ямаса мағлыұматлар базасына әдеуир зыян келтириуи мумкин. Әсиресе қәуипли болып HTML дескрипторы (мысалы, <script>) табылады. Себеби оларды Web-тармақларда қәуипли сценарийлерди жайластырыуда қолланыуға болады. Егерде қолланыушы аркалы киргизилген информацияны

тексерместен мағлыұматлар базасына киргизсек хэм сақлап қалсақ, толы проблемаларға дус келиұимиз мумкин, әсиресе бул жағдайда, егер Web-тармақтың басқа кириұшилери бул информацияны кейинликте қолланса (хэм жиберсе). Форма майданларына киргизилген мағлыұматларды тексеретуғын сораұлар, кеңрек төмендеги “Мағлыұматларды тексериұ” бөлимінде көриледи.

Қәуипсиз Web-серверди қолланыұ

Web-қосымша менен кириұшилердинөз-ара хәрекетин путкиллей есаплаұға болмайды. Егер Web-тармақта жайласқан файл қосымшаның клиент бөлимине (браузер) берилсе, мағлыұматлар интернет арқалы берилиип атырған ұақытта сырттағы басқа бир адам файлдың мазмунын оқып алыұы, толығы менен мумкин. Көплеген Web-тармақларда бул проблема емес. Бирақ егерде берилип атырған мағдыұматлардың ишиндекредит карточкалардың номерлери ямаса пароллери бар болса, олардың конфиденцияллығын тамийнлеұ сораұлары дарриұ алдынғы планға өтеди. Мағлыұматлардың қорғаныұлық дәрежесин қәуипсиз Web-серверди қолланып көтериұге болады. Қәуипсиз Web-серверлерде мағлыұматларды қорғаұ ушын SSL(SecureSocketsLayer) протоколы кең қолланылады. Оны қолланып атырған ұақытта информация кодланады хэм соннан кейин Web арқалы бериледи. Клиентлик программалық тәмийнлеұ қайтып декодлаұды орынлайды. Хеш қандай гумансыз, қәуипсиз Web-серверди кушейтиұ, қосымша кушлерди талап етеди, бирақ айиырым қосымшалар буларсыз өтпейди.

SSL протоколды қолланыұдың өз-ара хәрекетинде URL-адрес префикс https-тан басланады, ал http-дан емес.

Хәр бир Web-серверди өзинше қорғаұға болады. Қорғанылған SSL протоколын қолланыұ режимин қостырыұ хәққындағы информацияны, тийисли документацияда излеұге туұры келеди.

Мысалы: Apache серверин пайдаланғанымызда зәрур болған мағлыұматларды төмендеги еки Web-тармақларынан тауып алыуыңызға болады: www.modssl.org хәм www.apache-ssl.org.

Егер сиз Web-сервер IIS компания Microsoft-дан қолланыуды қарар еткен болсаңыз, онда SSL протоколы баянламасы хәққында мағлыұматларын www.microsoft.com Web-тармағынан тауып алыуыңызға болады.

Статистикалық Web-бети көриниси

Статистикалық Web-бети ең әпыұайы хәм кәуипсиз есапланады.

Егер Web-тармағында ислеп атырған уақтында сондай страницалар толық жетерли болса, онда PHP тили пайдаланылғанда хеш қандай зәрурлик болмайды. Бирақ сол уақытта статистикалық Web-бетлер менен өз-ара қатнасықта қолланылыуы мумкин.

Хар турли Web-бетлер көриниси ушын PHP-тилин пайдаланыуымызға болады, сондай-ақ, статистиканы да. HTML-кодын жуумақлауда echo функциясын қолланыуға болады.

Егерде PHP сценариясы қолланыуы керек болған бетинде тек ғана HTML дескрипторы қолланылса, онда белгили файлды талап етилген жерине төмендегише көринисте қосыуымыз керек: `include ("файл_аты");`

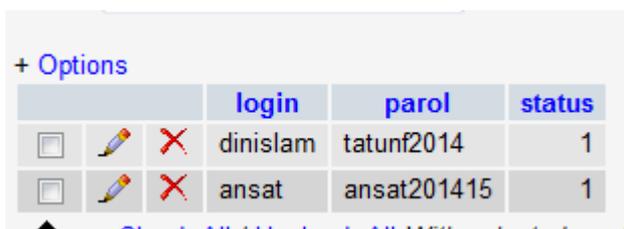
Егер қандайда бир себептен PHP сценариясында статистикалық Web-бетин өзгертиу керек болса, онда кейингисин орынлаң.

PHP дескрипторына ылайықлы (тийисли, керекли) файлды басына хәм ақырына қосып қойың. (`<?php` хәм `?>`), кейин усы файлдың басына echo функциясын қосың хәм барлық HTML кодын бирдей тырнақша менен жуумақлаң.

Енди биз PHP тили жәрдемінде жазылған информациялық системаларда кәуипсизлик шараларын мысаллар жәрдемінде қарастырып өтейик.

Көплеген тийисли парол хәм логин менен кириуши веб системаларда рухсат етилген пайдаланыушылар бир-неше болғанлықтан, парол хәм логинлер мағлыұматлар базасының белгили бир таблицасында ямаса

таблицаардың белгилі бір майданларында сақланыуы әдеттегі жағдай болып саналады. Көбінесе бұл құпия белгілерді MySQL таблицасында тууырдан тууы сақлау жағдайлары ушырасады. Мысалы:

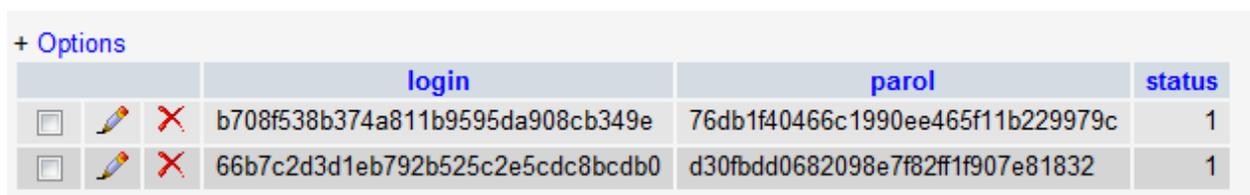


+ Options			login	parol	status
<input type="checkbox"/>			dinislam	tatunf2014	1
<input type="checkbox"/>			ansat	ansat201415	1

Егерде бузыушы жоқарыдағы жоллардың бири менен бизиң мағлыұматлар базамызға рұхсат алалған болса, онда системамыздың құпия паролларын аңсат жол менен қолға киргизеди.

Буның алдын алыу ушын табицаға бул жазыуларды шифрлап жайластырсақ мақсетке мууапық болар еди. Шифрлау алгоритмлерин өзлеримиз жасап алсақта болады. Ямаса PHP тилинде қатарларды шифрлаушы функция бар болып, ол md5 деп аталады. Бунда қатарларды шифрлаудда «MD5 RSA Data Security, Inc.» алгоритми қолланылып он алтылық санау системасының 32 ханалы саны алынады. Бул функцияға кері функция PHP тилинде жоқ болып, шифрды ашыуда бузыушылар тәрепинен исленген алгоритмлер хәм таяр программалар интернетте табылып, олар ислеу тезлиги көби менен 2 секундта шифрды аша алады. Онда бул функцияны пайдаланыуда программалаушы тәрепинен қосымша усыл қолланыуы керек болатуғынлығы көрсетиледи.

Жоқарыдағы парол хәм логинди бир рет шифрласақ онда табицаға төмендегише киритиуимиз керек болады.



+ Options			login	parol	status
<input type="checkbox"/>			b708f538b374a811b9595da908cb349e	76db1f40466c1990ee465f11b229979c	1
<input type="checkbox"/>			66b7c2d3d1eb792b525c2e5cdc8bcd0	d30fbdd0682098e7f82ff1f907e81832	1

Егерде форма арқалы киритилген логинди «\$log_f», паролди «\$par_f» өзгеріушилерине, базаға киритилген логинди «\$login_b», паролди «\$parol_b» өзгеріушилерине меншиклеп берсек онда рұхсатты беріуде дурыслығын тексеріу төмендегише жазылады:

```
if (md5(log_f) == $login_b and md5(«$par_f) == «$parol_b) { ..... }
```

Лекин бундай шифрларды қайта ашыу қыйын болмауы себепли, басқаша төмендегі ұсылды пайдаланыу тууры болады.

Қатардың белгили бир басланғыш сандағы символларын (мысал үшін 3 символын) бир мәрте шифрлап аламыз, қалған символларды шифрланған қатарға қосып қайта шифрласақ онда төмендегі алгоритм исленеди.

қатар	шифр
din	f88639f78d5d6da6dfb8498fd5e763b1
f88639f78d5d6da6dfb8498fd5e763b1islam	85caf03a3e3054f222e2fe4da5538e81

Бундай шифрланған паролди салыстырып рұхсатты шөлкемлестириу үшін төмендегі код жазылады.

```
if (md5(md5(substr($log_f,0,3)).substr($log_f,3, strlen($log_f)-3)) == $login_b) { ..... }
```

Бул алгоритмнің ислеуин мысал ретинде тексерип көрейик:

Егер қупыя сөз “dinislam” сөзи болып ол формаға дурыс киритилгенде «Сәлем Динислам» хабары шығыуы, қәте киритилгенде қайтадан формаға келиу керек болсын. Программа коды төмендегіше болып:

```
<?
$login_b='85caf03a3e3054f222e2fe4da5538e81';
if (md5(md5(substr($log_f,0,3)).substr($log_f,3, strlen($log_f)-3))<>$login_b){
    echo "<b>Qupiya zo'zdi kiritin': </b>
<form method=post action=index.php><input name=log_f>
<input name=submit type=submit value=Kiriw></form>";}
else {echo 'SALEM DINISLAM';}
?>
```

Браузерге киргенде



Айнасы пайда болады, егер форма майданына «dinislam» сөзи киритилсе:



Айнасы пайда болады.

Бундай усылдан пайдаланыў рухсаттың толық қәўипсилигин тәмийнлениўине кепиллик бермейди. Себеби веб программаластырыўдан хабары бар адам өзиниң компьютеринде PHP код жазып оған тосынарлық сайлап алыў алгоритмлерин пайдаланып, формаға автомат түрде қатар киргизиў мүмкиншилигине ийе болады. Бул пайдаланыўшының серверге байланысыў тезлигине қарап бир неше сораўнаманы аз ўақыт аралығында жиберіў мүмкиншилигин береді. Бул процессти қупыя сөз табыламан дегенше даўам еттире бериўге болады.

Бундай бузыўдың алдын алыў ушын сайтларда көбинесе қосымша қәўипсизлик түринде сораў қойылады. Олардың ишинде исенимлиреги сүўрет көринисинде браўзерде пайда болатуғын текстти керекли майданға териў керек болатуғын усыл болып саналады.

old.ccitt.uz/uz/interactive/reception/

★ 8 Искать в

Murojaat mavzusi: Boshqalar

Murojaat matni:

Fayl qo'shish: **Указать**

Faylning hajmi 2Mb.dan ortmasligi kerak. Agar Siz bittadan ortiq fayl jo'natmoqchi bo'lsangiz, u holda barcha fayllarni arxivga joylagan holda jo'nating

Spamdan himoyalash:

O'ngdagi tasvirdagi belgilarni kiriting

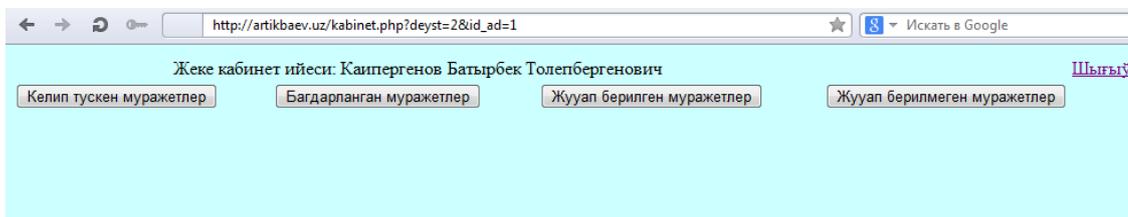
Jo'natish

Бундай қорғалыў ушын дәслеп тосынарлы символларды берийўши хэм оларды сүүретке аландырыўшы программалық код (ПКЖ ның ақырында қосымшада келтирилген) жазылыўы хэм қупыя сөз сервер тәрепинде салыстырылыў ўақтында бул тексте салыстырылыўы шәрт. Бул системаның ислеўине белгили тәсийр көрсетпейди. Бирақ кейинги ўәқытлары жеке компьютерлердинде мүмкиншиликлериниң артыўы және «компьютер көрийўи» технолгиясының имканиятларының жоқарылаўы бул усылдыда егер кушли программист болса бузып өте алыўы мүмкин екенлигин көрсетеди. Бундай бузыўдың алдын алыў ушын сүүрет текстти браўзер айнасында жайласыў майданының еркинлиги, (майдан кеңирек болыўы, сүүрет киширек болыўы) ямаса баскаша айтканда сүүреттиң хәр жерде шығыў (орнын озгертиў) имканиятын пайдаланса болады.

Сораўнамаларды автомат толтырыў (спам) арқалы бузыўдың алдын алыў усылларынан және бири бул, қупыя сөзлерди қәте киритиўдин санын шеклеў болып, бунда система қатарына белгили ўақыт аралығында белгили сандағы қәте киритилген қупыя сөзден кейин блока түсийи боып, бул барлық пайдаланыўшылар ямаса анық бир IP адрес ушын шеклеўлерди қойыў арқалы әмелге асырылады. Буның ушында мағлыўматлар базасында бир

таблица ямаса текст файл ашып пайдаланыў керек болады. Бул усылға жазылған код қосымшада келтирилип өтилген.

Және веб-серверлерде жаратылған системаларда программалаўшылар тәрәпинен жол қойылатуғын эззи қорғалыўдың бири болған рухсат берилгеннен кейинги ислеў имканиятларының программа кодларының бөлек файлларда жазылып керекли жерге `include("файл_аты")` ; функциясы арқалы шақыртылып ислетип кетилиўи ямаса базаға мағлыўматлар жазыў файлларының өз алдына болыў жағдайларында ушырасады. Мысалы Артықбаев Ықластың ПҚЖ ның амелий жумысының алдын ислеген вариантында пайдаланыўшы өзиниң кабинетине парол хәм логин арқалы кириўи жақсы шөлкемлестирген болып, бирақ рухсат етилген пайдаланыўшыға ашық болған хызметлер коды `kabinet.php` файлында берилген. Егерде браузер адреслер қатарына туўрыдан туўры төмендегише адреси терсек онда рухсаттан өтпей ақ рухсат етилмеген жеке кабинетке кириўимизге болады.



Бундай бослықтың алдын алыў ушын хәр-бир керекли өз-алдына жазылған программа коды сақланыўшы файллардың басына, рухсатты тексерийши кодты шақыртып ислетип кетиў мақсетке муўапық болады.

ЖУЎМАҚЛАЎ

Хәзирги дәўирде мәмлекетимизде Интернет тез пәт пенен раўажланып турмысымыздың хәр-бир тараўына кирип келиўи, жәмийеттеги қатнаслардың электрон түрине өтиўи турмысымызда унамлы тәсийрин тийгизиўи менен биргеликте усы информация системаларының кәўипсизлигин сақлаў машқалаларын келтирип шығармақта.

Бул питкерий кәнигелик жумыста Клиент-сервер технологиясында жаратылған системаларда веб серверге жиберилген сораўнамаларда кәўипсизлик шараларын көрий усыллары қаралып төмендеги мәселелер шешилди.

- Клиент-сервер технологиясы хәм олардың түрлери, тәреплердің мүмкиншиликлери үйренилди.
- Веб серверлер, олардың програмалық тәмийнаты хәмде тармақ арқалы оларға болатуғын хұжимлерде қолланылатуғын усыллар анализленди.
- Веб серверлерге тийкарғы хұжимлер ушын жол ашып беретуғын информациялық системаларда жаратылған сораўнамаларды жаратыў тиллериниң бири болған PHP скриптлик тилиниң қолайлықлары үйренилди.
- PHP тилинде шөлкемлестирилген сораўнамалардың программалық кодларын ықшамластырыў хәм сырлылығын асырыў мәселеси қарастырылып анық мысаллар исленди.
- Сораўнамалар арқалы әмелге асырылыўы мүмкин болған анықланған хұжимлер ушын қарсы усылларға программалық кодлар жазылды .

Бул исленген жумыслар арқалы Клиент-сервер технологиясында жаратылған системаларда веб серверге жиберилген сораўнамаларда кәўипсизлик шараларын көрийде жақсы нәтийжелерге ерисиўге болатуғынлығы көрсетилди.

Интернет технологияларының жедел раўажланып баратырғанын есапқа алған халда, Web программаластырыўға болған талапта артып бармақта. Сонлықтан биз жоқарыда көрип өтилгенлер менен тоқтап қалмай бул тараўда тынбай излениўимиз керек.

ПАЙДАЛАНГАН ЭДЕБИЯТЛАР

1. Орлов А.А.
PHP. Полезные приёмы. Издательство: Горячая -Телеком, 2004
2. Кузнецов М.В., Симдянов И.В., Голышев С.В.
PHP5. Практика создания web-сайтов. Изда: БХВ-Петербург, 2005
3. Кухарчик А..PHP. Обучение на примерах. Издательство: Новое знание, 2004. PDF
4. Луис Аргерих, Ванкиу Чой, Джон Коггсхол.
Профессиональное PHP программирование. Издательство: Символ-Плюс, 2003.DJVU
5. Sh.A. Nazirov, R.V. Qobulov.
MySQL va PHP (qo'llanma). Uzbekistan 2008)
6. Юрий Едомский
Техника Web-дизайна – БХВ-Петербург - 2005.
7. Буковецкая О.А.
Дизайн текста: шрифт, эффекты, цвет - ДМК – 2000
8. Рейчел Эндрю
CSS: 100 и 1 совет "Символ-Плюс"-2010
9. И. Шапошников
Web-сайт своими руками -: БХВ – Санкт-Петербург, 2000
10. <http://www.php.ru/>
11. <http://www.intuit.ru/>
12. <http://www.mysql.ru>