УДК 004

ВЕБ-ҚОСЫМШАЛАРДЫҢ ОСАЛДЫҒЫН АНЫҚТАУ ЖӘНЕ ЖОЮ ҚҰРАЛДАРЫ

Рахымжанова Айнұр Ғалымқызы
ainur.galimkyzy@gmail.com
Л.Н.Гумилев атындағы ЕҰУ ақпарттық технологиялар факультетінің
4-ші курс студенті, Нұр-Сұлтан, Қазақстан
Ғылыми жетекші – Сагиндыков Каким Молдабекович

Қазіргі уақытта веб-қосымшалар үлкен танымалдыққа ие, өйткені олар қарапайым қосымшаларда жоқ көптеген маңызды артықшылықтарды ұсынады. Бағдарламалық жасақтама саласында болып жатқан жайттарды қарастыра отырып, веб-қосымшалардың өте үлкен дамуын байқауға болады. Бүгінгі күнде компаниялардың көп бөлігі қарапайым қосымшалардан веб-қосымшаларға көшуде, өйткені олар веб-қосымшалар технология тарихына күшті серпін береді деп есептейді. Веб-қосымшалардың артықшылықтары:

- Веб-қосымшаларды орнату және жаңарту арзан және оңайырақ.
- Веб-қосымшалар соңғы пайдаланушы үшін жан-жақты және практикалық болып есептеледі.
- Веб-қосымшалар деректерді сақтауды ұйымдастыруды жеңілдетеді.

Веб қосымшалар - бұл клиент-сервер архитектурасына негізделген бағдарламалық жасақтаманың ерекше түрі. Веб-қосымшаның логикасы сервер мен клиент арасында бөлінеді, деректерді сақтау негізінен серверде жүзеге асырылады, ал ақпарат алмасу желі арқылы іске асады. Бұл тәсілдің артықшылықтарының бірі-тұтынушылар пайдаланушының нақты операциялық жүйесіне тәуелді емес, сондықтан веб-қосымшалар платформааралық қызмет болып табылады.

Алайда, веб-қосымшалар ақпараттық қауіпсіздікті қамтамасыз етумен байланысты көптеген мәселелерге ие, өйткені оларды әзірлеу көбіне қатаң мерзімдерде жүзеге асырылады және қосымша желі арқылы қолданушыларға да, хакерлерге де қол жетімді. Осалдықтар зиянкестерге құпия ақпаратты ұрлауға, деректерге рұқсатсыз өзгерістер жүргізуге, қосымшаның қолжетімділігін бұзуға мүмкіндік береді. Қазіргі уақытта веб-қосымшалардың қауіпсіздігін қамтамасыз ету проблемасы өте өзекті: барлық анықталатын осалдықтардың 60%-дан астамы веб-қосымшаларға тиесілі. Веб - қосымшалардың қауіпсіздік деңгейі біртіндеп өсіп келеді, бірақ әлі де төмен. Оған соңғы статистикалар арқылы, мысал келтіруге болады:

- 10 веб-қосымшаның 9-ында қылмыскерлер қолданушыларға шабуыл жасай алады. Оның ішінде-клиенттерді олардың бақылауындағы ресурсқа қайта бағыттау, фишингтік шабуылдар көмегімен есептік деректерді ұрлау, компьютерді зиянды БҚ-мен зақымдау.
- Қосымшаға рұқсатсыз кіру 39%-ға мүмкін. Сонымен қатар, жүйені толық бақылау веб-қосымшалардың 16%-ында, ал жүйелердің 8% - ында веб-қосымшаның серверін толық бақылау ұйымның жергілікті желісіне шабуыл жасауға мүмкіндік берді.

● Маңызды деректердің ағып кету қаупі веб-қосымшалардың 68% - ында кездеседі. "Ағып кеткен" деректер арасында бірінші орында дербес (47% ағып кету), ал екінші орында — есептік (31%) алады.

● Осалдықтардың 82% - ы қосымшаның кодында болды.

● Орташа алғанда бір веб-қосымшаға келетін осалдықтар саны 2018 жылмен салыстырғанда бір жарым есе азайды. Орташа алғанда, бір жүйеге 22 осалдық келеді, олардың төртеуі тәуекелдің жоғары деңгейіне ие.

● Әрбір бесінші осалдық-жоғары деңгейдегі тәуекел.

Веб-қосымшаның осалдығы әзірлеушілер қауіпті кодты веб-қосымшаға қосқанда пайда болады. Бұл даму сатысында да, бұрын табылған осалдықтарды қайта өңдеу немесе түзету сатысында да болуы мүмкін. Кемшіліктер көбінесе олардың ауырлығы мен таралуы бойынша жіктеледі.

OWASP (Open Web Application Security Project коммерциялық емес ұйымы) өз зерттеулерінен кейін Интернет пен веб-қызметтерге арналған бағдарламалық қамтамасыз етудегі ең қауіпті, бірақ сонымен бірге жиі кездесетін осалдықтардың тізімін ұсынды. OWASP мәліметтері бойынша, бұл осалдықтарға өзін де, клиенттерін де хакерлерден қорғағысы келетін мемлекеттік және коммерциялық ұйымдар мұқият назар аударуы керек. Бұл осалдықтардың барлығы айтарлықтай кең таралған, тіпті білікті емес хакерлер де оларды пайдалана алады, өйткені интерактивті бұзу құралдарын табу оңай. OWASP Top 10:

1) инъекция (инъекцияның барлық түрлері, соның ішінде SQL, LDAP және т.б.).

2) Cross Site Scripting (XSS).

3) Broken Authentication and Session Management (аутентификация және сеансты басқару архитектурасындағы қателер).

4) Insecure direct Object References (қорғалмаған ресурстар мен нысандар).

5) Cross Site Request Forgery (CSRF).

6) Security Misconfiguration (түрлі құрылымдардың, платформалардың қауіпті конфигурациясы).

7) failure to Restrict URL Access (арнайы артықшылықтарды талап ететін функционалға рұқсатсыз кіру – мысалы, WordPress-тегі блогты басқаруға қол жеткізу үшін URL-де қосарланған "//" көмегімен тексеруді айналып өту).

8) анықталмаған Redirects and Forwards (фишингке, HTTP Response Splitting және XSS-ке әкелетін ашық қайта бағыттаулар).

9) Insecure Cryptographic Storage (маңызды деректерді қауіпсіз сақтау).

10) Insufficient Transport layer Protection (деректерді тасымалдау деңгейінде беру кезінде жеткіліксіз қорғау, мысалы HTTPS орнына HTTP бойынша).

Одан бөлек, JSON немесе XML сұрауларын өңдейтін веб-қосымшалар мен API-лер да шабуылдарға ұшырайды.

JSON (JavaScript Object Notation) - бұл қосымшалар арасындағы байланыс үшін қолданылатын жеңіл деректер алмасу форматы. Ол XML-ге ұқсас, бірақ қарапайым және JavaScript өңдеуге ыңғайлы. Көптеген веб-қосымшалар осы форматты деректерді өзара алмасу, сериялау / сериясыздандыру және маңызды ақпаратты сақтау үшін пайдаланады.

Қарапайым серверлік JSON инъекциясын PHP-де келесідей жасауға болады:

Сервер пайдаланушы деректерін JSON жолы ретінде, оның ішінде есептік жазба түрін сақтайды. Пайдаланушы аты мен пароль пайдаланушының кіруінен тікелей тазалаусыз алынады;

JSON жолы қарапайым байланыстыру арқылы жасалады:

```
$json_string= '{"account":"user","user":'".$_GET['user']."',"pass":'".$_GET['pass']."'}'
```

Зиянкес өзінің пайдаланушы атына деректерді қосады:

```
john%22,%22account%22:%22administrator%22
```

Алынған JSON жолы:

```
{
```

"account":"user",
"user":"john",
"account":"administrator",
"pass":"password"
}

Осылайша, JSON талдаушысы сақталған жолды оқыған кезде (json_decode) екі есептік жазбаны анықтайды және john пайдаланушысына әкімші құқығын бере отырып, соңғысын алады.

Веб-қосымшалардың қауіпсіздігін қамтамасыз етудің кең таралған әдістерінің бірі - оларды жою мақсатында веб-қосымшаның осалдығын анықтау. Бұл жұмыста веб-қосымшалардың осалдығын анықтаудың заманауи әдістері талқыланады және олардың мүмкіндіктеріне талдау жүргізіледі. Веб-қосымшалардың осалдықтарын автоматты түрде анықтау әдістерін екі негізгі топқа бөлуге болады:

1) веб-қосымшаның бастапқы кодтарына жүгінбестен, стендте жазылған веб-қосымшаның жұмысын талдайтын әдістер:

- Қауіпсіздік кеңестерінің көмегімен веб-қосымша туралы ақпаратты анықтау және оның осалдығын анықтау әдісі.
- Енуді тестілеу әдісі.

2) веб-қосымшаның бастапқы кодтарын және конфигурациялық баптауларды талдайтын әдістер:

- Веб-қосымшаның бастапқы кодтарын статикалық талдау әдісі.
- Веб-қосымшаның бастапқы кодтарын динамикалық талдау әдісі.

Статикалық талдау әдісін динамикалық тілдерге қолданудың айтарлықтай кемшілігі бар: орындау кезінде бастапқы кодты құруға болатын қосымшаларда статикалық талдау әдісі noninterference қасиетін бұзумен байланысты барлық осалдықтарды анықтауға кепілдік бере алмайды. Динамикалық талдау әдісі, егер сынақ жиынтығы бағдарламаның барлық мүмкін жолдарын қамтымаса, noninterference қасиетінің бұзылуына байланысты барлық осалдықтарды анықтауға кепілдік бере алмайды. Енуді тестілеу әдісі барлық осалдықтардың анықталғанына кепілдік бере алмайды. Осылайша, барлық қарастырылған әдістер осалдықтарды анықтауға толық кепілдік бере алмайтынын көре аламыз. Статикалық және динамикалық талдау әдістері веб-қосымшаға қате енгізілген деректерді жіберумен байланысты осалдықтарды ғана анықтай алады.

Пайымдай келе, талдау негізінде веб-қосымшалардағы осалдықтарды автоматты түрде анықтау әдістерін одан әрі дамытуға мүмкіндік беру үшін әр түрлі әдістердің мүмкіндіктерін интеграциялау керек деген қорытынды жасауға болады.

- Осал тұстарды барынша кең қамту;
- Автоматты талдау қорытындылары бойынша берілген кластардың осалдықтарының болмауын кепілдендіруге (идеал) қол жеткізу үшін осалдықтарды анықтаудың толықтығын бақылау.

Қолданылған әдебиеттер

1. Vartanov S.P., Gerasimov A.Y. Dynamic program analysis for error detection using goal-seeking input data generation. *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2014;26(1):375-394. (In Russ.)

2. Козлов Д. Д., Петухов А. А. "Методы обнаружения уязвимостей в web - приложениях" / Программные системы и инструменты: тематический сборник ф-та ВМиК МГУ им. Ломоносова N 7. П/р Л.Н. Королева. М: Издательский отдел ВМиК МГУ. Изд-во МАКС Пресс, 2006 г.

3. Низамутдинов М. Ф. Тактика защиты и нападения на Web-приложения. — СПб.: БХВ-Петербург, 2005. — 432 с.: ил.

4. Будников Е.А., Борисова С.Н. УЯЗВИМОСТЬ Web-ПРИЛОЖЕНИЙ // Международный студенческий научный вестник. – 2015. – № 3-2.;
5. Королев О.Л. Безопасность веб-приложений / О.Л. Королев, М.А. Лукьянова // Сборник трудов Международной научно-практической конференции «Проблемы информационной безопасности». – Симферополь, 2016. – С. 166- 168.
6. OWASP Top 10 Application Security Risks – 2017. [Электрондық ресурс]. - Кіру режимі: https://www.owasp.org/index.php/Top_10-2017_Top_10 (өтініш берген күні: 04.03.2018).
7. https://habr.com/ru/post/526878/.
8. https://www.ptsecurity.com/ru-ru/research/analytics/web-vulnerabilities-2020/
9. http://sanchiz.net/blog/advantage-of-web-applications