ЛЕКЦИЯ 12
Веб-қосымшаларға бағытталған кибершабуылдар және оларға қарсы әрекет ету

Лекция мақсаты

Веб-қосымшаларға тән негізгі қауіптерді, олардың пайда болу себептерін, белгілерін және қорғаныс қағидаларын түсіндіру; SQL Injection, XSS, CSRF және авторизация мәселелерін біртұтас қауіп моделі аясында талдау.

Веб-қосымшаның қауіп моделі

Веб-қосымшаның қауіп моделі — қандай активтер қорғалатынын, дерек қайдан келетінін, қандай сенім шекараларынан өтетінін және қандай қауіптер іске асуы мүмкін екенін жүйелі сипаттау. Браузерден немесе сыртқы API-ден келген дерек сенімді деп қабылданбайды. Сервер деректің форматын ғана емес, пайдаланушының кім екенін, оның нақты операцияға құқығын және бизнес-ережені орындалуын тексеруі тиіс.

Мысал. Тұрғындар порталында өтініш нөмірі дұрыс форматта болуы мүмкін, бірақ бұл сол пайдаланушыға басқа тұрғынның өтінішін көруге құқық береді деген сөз емес. Сондықтан енгізуді тексеру мен авторизация — екі бөлек бақылау.

SQL Injection

SQL Injection — сырттан келген дерек SQL-сұраныстың құрылымына рұқсатсыз ықпал еткенде пайда болатын осалдық. Негізгі себеп — дерек пен команданың ара-жігін дұрыс бөлмеу. Егер қолданба пайдаланушы енгізген мәтінді SQL жолына тікелей біріктірсе, дерек сұраныс логикасының бөлігіне айналуы мүмкін.

Қорғаныстың негізгі қағидасы — параметрленген сұраныстарды қолдану, деректер қоры тіркелгісіне ең аз құқық беру, енгізуді бизнес-ережеге сәйкес тексеру және қателерді пайдаланушыға артық техникалық мәліметсіз өңдеу. WAF қосымша қабат бола алады, бірақ қауіпсіз кодтауды алмастырмайды.

Мысал. Тұрғынның ЖСН-і бойынша жазбаны іздейтін сервис тек күтілетін форматты қабылдап, мәнді SQL командасынан бөлек параметр ретінде беруі тиіс. Сонда пайдаланушы енгізуі сұраныстың құрылымын өзгерте алмайды.

Cross-Site Scripting (XSS)

XSS — сенімсіз дерек браузерде белсенді код ретінде интерпретацияланған кезде туындайтын веб-осалдық. Мәселе енгізудің өзінде ғана емес, деректі қай контексте шығаратынында: HTML мәтіні, атрибут, URL немесе JavaScript контексті әртүрлі кодтауды талап етеді.

XSS сақталатын, шағылысатын және DOM-негізді болып жіктеледі. Сақталатын жағдайда қауіпті мазмұн серверде сақталып, кейін басқа пайдаланушыларға беріледі; шағылысатын жағдайда сұраныс мәні жауапта бірден көрінеді; DOM-негізді жағдайда қауіпті өңдеу браузердегі JavaScript ішінде жүреді.

Мысал. Тұрғын порталдағы өтінішке түсініктеме жазады. Егер бұл мәтін кейін оператордың браузерінде қауіпсіз кодталмай көрсетілсе, енгізілген мазмұн интерфейстің логикасына әсер етуі мүмкін. Сондықтан пайдаланушы мәтіні дерек ретінде көрсетілуі керек.

CSRF

CSRF — аутентификацияланған пайдаланушының браузерін оның еркінен тыс маңызды сұраныс жіберуге пайдалануға бағытталған шабуыл класы. Браузер белгілі жағдайда сессиялық деректерді сұранысқа автоматты түрде қосатындықтан, сервер сұраныстың шынымен пайдаланушы ниетінен туғанын ажыратуы керек.

Қорғанысқа CSRF-токен, SameSite cookie параметрлері, маңызды операцияларды қайта растау және Origin/Referer сияқты контекстік белгілерді орынды тексеру жатады. GET сұранысы жүйе күйін өзгертпеуі тиіс.

Мысал. Порталда байланыс телефонын өзгерту операциясы тек белсенді сессияға сүйенсе, сыртқы бет пайдаланушы браузерін қажетсіз сұраныс жіберуге итермелеуі мүмкін. Қосымша бірегей токенді және операцияны растауды талап етсе, тәуекел төмендейді.

Авторизация және объект деңгейіндегі қолжетімділік

Авторизация — аутентификациядан өткен субъектінің нақты ресурсқа немесе операцияға құқығын тексеру. Пайдаланушы жүйеге дұрыс кіргенінің өзі барлық объектілерге қол жеткізуге рұқсат бермейді. Сервер әр сұраныста субъект, ресурс және әрекет арасындағы байланысты тексеруі тиіс.

Мысал. /application/125 және /application/126 екі өтінішті білдірсін. Пайдаланушы URL-дегі идентификаторды өзгерте алғанымен, сервер 126-өтініш осы пайдаланушыға тиесілі болмаса оны қайтармауы керек.

Көпқабатты қорғаныс

Веб-қорғаныс бір ғана механизмге сүйенбеуі керек. Қауіпсіз кодтау, аутентификация мен авторизация, TLS, қауіпсіз конфигурация, WAF, журналдау, SIEM мониторингі және жаңарту процестері бірін-бірі толықтырады. Бір қабаттағы қате келесі қабат арқылы шектелуі мүмкін.

Мысал. WAF күмәнді сұранысты тоқтатуы мүмкін, бірақ негізгі қолданбада параметрленген SQL сұраныстары бәрібір қолданылуы тиіс. Журналдау қайталанған күмәнді әрекеттерді кейін корреляциялауға мүмкіндік береді.

Негізгі жіктеу және салыстыру

Қауіп	Негізгі себеп	Нысана	Негізгі қорғаныс
SQL Injection	Дерек пен SQL командасын араластыру	Деректер қоры	Параметрленген сұраныс, ең аз құқық
XSS	Сенімсіз деректі браузерде қауіпсіз емес шығару	Пайдаланушы браузері	Контекстік кодтау, қауіпсіз DOM
CSRF	Сессиямен келген сұраныстың ниетін растаудың болмауы	Маңызды операция	CSRF-токен, SameSite, қайта растау
Авторизация қатесі	Ресурсқа құқықты серверде тексермеу	Бөгде объект	Әр сұраныста объект деңгейіндегі тексеру

Өзара байланыстар схемасы



Сурет 1. Веб-қосымшаны көпқабатты қорғау логикасы

Кешенді практикалық мысал

Тұрғындар порталында аутентификация, өтініш беру, түсініктеме жазу, құжат жүктеу және профильді өзгерту функциялары бар. Әр функция әртүрлі қауіппен байланысты: іздеу мен деректер қоры сұраныстары үшін SQL Injection, пайдаланушы мәтінін көрсету үшін XSS, профильді өзгерту үшін CSRF, өтінішті қарау үшін объект деңгейіндегі авторизация маңызды.

Қорғаныс кезінде барлық енгізу серверде тексеріледі, SQL параметрленеді, браузерге шығарылатын мәтін контекстке сәйкес кодталады, маңызды күй өзгертетін операциялар CSRF-токенмен қорғалады, ал әр объектіге қолжетімділік пайдаланушы құқығымен салыстырылады. Оқиғалар журналға жазылып, күмәнді қайталанулар мониторинг жүйесіне беріледі.

Қорытынды

Веб-қосымша қауіпсіздігі жеке осалдықтарды жаттаумен шектелмейді. Әр қауіп деректер ағыны, сенім шекарасы және серверлік бақылаулармен байланыста қарастырылады. Қорғаныстың сенімділігі қауіпсіз кодтау, құқықтарды тексеру, конфигурация, журналдау және мониторинг қабаттарының үйлесуіне тәуелді.

Бақылау сұрақтары

1. SQL Injection қандай жағдайда пайда болады?
2. Параметрленген сұраныс не үшін қажет?
3. XSS түрлерінің айырмашылығы қандай?
4. XSS кезінде контекстік кодтау неге маңызды?
5. CSRF қандай браузерлік қасиетке сүйенеді?
6. CSRF-токеннің қызметі қандай?
7. Аутентификация мен авторизацияның айырмашылығы неде?
8. Объект деңгейіндегі авторизация қалай тексеріледі?
9. WAF неге қауіпсіз кодтауды алмастырмайды?
10. Көпқабатты веб-қорғаныс қалай құрылады?

Әдебиеттер тізімі:

1. Менщиков А.А., Заколдаев Д.А., Воробьева А.А. Введение в кибербезопасность и киберустойчивость. — СПб.: Университет ИТМО, 2024.

2. Лукьянец С.Е. Гибридные системы кибербезопасности: искусственный интеллект и анализ сетевого трафика. — 2025.