Дәріс 14. Операциялық жүйелердің қауіпсіздігі

14.1. Зиян келтірушілер мен зиянды бағдарламалар
Жүйеге кіру қаупі
Зиянкелтірушілер
Зиянды бағдарламалар
Қарсы шаралар
Интрузияларды анықтау
Аутентификация
Қол жеткізуді басқару
Брандмауэрлер
14.2. Буферлік толып кету
Буферлік толып кету түріндегі шабуылдар
Компиляция уақытын қорғау
Бағдарламалау тілін таңдау
Қауіпсіз бағдарламалау әдістері
Тілді кеңейту және қауіпсіз кітапханаларды қолдану
Стек қорғау механизмдері
Жұмыс уақытын қорғау
Орындалатын модульдердің мекенжай кеңістігін қорғау
Мекенжай кеңістігін рандомизациялау
Қауіпсіздік беттері
14.3. Қол жеткізуді басқару
Файлдық жүйеге кіруді басқару
Қол жеткізуді басқару стратегиялары
14.4. Операциялық жүйелерді қорғауды күшейту
Амалдық жүйені орнату және жаңартуларды қолдану
Қажет емес қызметтерді, қолданбаларды және хаттамаларды жою
Пайдаланушыларды, топтарды және аутентификацияны конфигурациялау
Ресурстарды басқару конфигурациясы
Қорғауды басқарудың қосымша құралдарын орнату
Жүйені қорғауды тестілеу
14.6. Қауіпсіздікті қолдау
Хаттама
Деректердің сақтық көшірмесін жасау және мұрағаттау

14.1. Зиян келтірушілер мен зиянды бағдарламалар

Операциялық жүйе әр процеске бірқатар артықшылықтарды қосады. Бұл привилегиялар процесске қандай ресурстар қол жетімді болатынын анықтайды, соның ішінде жедел жад аймақтары, файлдар және артықшылықты жүйелік командаларды анықтайды. Әдетте, пайдаланушы атынан орындалатын процесс операциялық жүйеге берілген пайдаланушы үшін артықшылықтарға ие.

Әдеттегі жүйеде артықшылықтың ең жоғары деңгейі әкімшілік, супервайзер немесе түбірлік қатынас деп аталады. Атап айтқанда, түбірлік қол жетімділік амалдық жүйенің барлық функциялары мен қызметтеріне қол жеткізуді қамтамасыз етеді. Түбірлік қол жетімділікпен процесс жүйені толығымен басқарады және бағдарламалар мен файлдарды енгізе немесе өзгерте алады, өзге процестерді басқара алады, желілік трафикті жібере және қабылдай алады, артықшылықтарды өзгерте алады.

Кез-келген операциялық жүйені жобалау кезінде қауіпсіздікті қамтамасыз етудің басты мәселесі-зиянды пайдаланушының немесе зиянды бағдарламалық жасақтаманың үзіндісінің жүйеде рұқсат етілмеген артықшылықтарға ие болу әрекеттерін қалай болдырмауға немесе кем дегенде анықтауға болады, атап айтқанда, түбірлік қол жетімділікті анықтау болып табылады.

Жүйеге кіру қаупі

Жүйеге кіру қаупі екі жалпы санатқа бөлінеді: зиян келтірушілер мен зиянды бағдарламалар.

Зиян келтірушілер

Қауіпсіздіктің ең көп таралған қауіптерінің бірі – шабуылдаушы (ал екіншісі-вирустар), көбінесе хакер деп аталады. [5] Зиян келтірушілердің төмендегідей кластары бар.

- **Жалған бейне.** Компьютерді пайдалануға рұқсат етілмеген және жүйеге кіруді басқару элементтеріне енетін және заңды пайдаланушының есептік жазбасын теріс пайдаланатын субъект.
- **Заң бұзушы**. Оған кіруге рұқсат етілмеген деректерге, бағдарламаларға немесе ресурстарға қол жеткізетін заңды пайдаланушы немесе мұндай қол жеткізуге рұқсат етілген, бірақ өз артықшылықтарын теріс пайдаланатын пайдаланушы.
- **Заңсыз пайдаланушы.** Тексеруді айналып өту және қол жеткізуді басқару элементтерін айналып өту, бақылау тексерісі үшін деректер жинауды басу мақсатында жүйені қызықты бақылаушы және оны пайдаланатын субъект.

Жалған адам бөтен адам болуы мүмкін, бұзушы әдетте ұйым мүшесі, ал жасырын пайдаланушы бөтен адам да, ұйым мүшесі де болуы мүмкін. Зиян келтірушілердің шабуылдары зиянсыздан ауырға дейін кең ауқымда өзгереді. Зиянсыз шабуылдар жасайтындардың қатарына интернетті және басқа желілерді зерттеуге тырысатын көптеген адамдар кіреді. Ал ауыр шабуылдарға артықшылықты деректерді оқуға, деректердің рұқсат етілмеген модификацияларын орындауға немесе жүйенің қалыпты жұмысын мүлдем бұзуға тырысатын субъектілер жатады.

Шабуылдаушының мақсаты-жүйеге қол жеткізу немесе жүйеде оған қол жетімді артықшылықтарды кеңейту. Бастапқы шабуылдардың көпшілігі жүйеде немесе бағдарламалық жасақтамада зиянды пайдаланушыға жүйеде саңылау ашатын кодты орындауға мүмкіндік беретін осалдықтарды пайдаланады. Зиянкестер жүйеге белгілі бір артықшылықтармен орындалатын бағдарламада буфердің толып кету түріне шабуыл жасау арқылы қол жеткізе алады.

Сонымен қатар, шабуылдаушы қорғалуы керек ақпаратты алуға тырысады. Кейбір жағдайларда бұл ақпарат пайдаланушы құпия сөзі түрінде беріледі. Кез-келген пайдаланушының паролін біле отырып, шабуылдаушы жүйеге кіріп, заңды пайдаланушыға берілген барлық артықшылықтарды жүзеге асыра алады.

Зиянды бағдарламалар

Компьютерлік жүйелерге төнетін қауіптердің ең күрделі түрлері компьютерлік жүйелердегі осалдықтарды теріс пайдаланатын бағдарламалар болуы мүмкін. Мұндай қауіптер зиянды бағдарламалар деп аталады (malicious software немесе malware). Бұл мәтінде біз қолданбалы және қызметтік бағдарламаларға, мысалы, редакторлар мен компиляторларға, сондай-ақ ядро деңгейіндегі бағдарламаларға қауіптерді қарастырамыз.

Зиянды бағдарламаларды екі санатқа бөлуге болады: негізгі бағдарламаны қажет ететіндер және тәуелсіз бағдарламалар. Бірінші санаттағы зиянды бағдарламалар паразиттік бағдарламалар деп аталады және олар белгілі бір қолданбалы, қызметтік немесе жүйелік бағдарламаларға қарамастан өмір сүре алмайтын бағдарламалардың бөліктері болып табылады. Бұған вирустар, логикалық бомбалар және саңылаулар мысалдар болып табылады. Екінші санаттағы зиянды бағдарламалар-бұл амалдық жүйеде жоспарлауға болатын тәуелсіз бағдарламалар. Құрттар мен роботтар бағдарламалары мысалдар болып табылады.

Сондай-ақ, көбеймейтін және көбеюге қабілетті бағдарламалық қауіптерді ажыратуға болады. Бірінші санатқа арнайы триггермен белсендірілген бағдарламалар немесе бағдарлама бөліктері кіреді. Оларға тән мысалдар-логикалық бомбалар, саңылаулар және робот-бағдарламалар. Екінші санатқа тәуелсіз бағдарламалар немесе олардың үзінділері жатады, олар орындалған кезде олардың бір немесе бірнеше көшірмелерін жасай

алады, оларды кейіннен сол немесе басқа жүйеде іске қосады. Вирустар мен құрт бағдарламалары мысалдар бола алады.

Қарсы шаралар

Интрузияларды анықтау

RFC 4949 құжатында (Internet Security Glossary – Интернет қауіпсіздігі бойынша арнайы терминдер сөздігі) кіруді анықтау келесідей анықталады: жүйелік ресурстарға рұқсатсыз қол жеткізу әрекеттерін нақты немесе оған жақын уақытта анықтау және ескерту мақсатында жүйелік оқиғаларға мониторинг пен талдауды жүзеге асыратын қауіпсіздік қызметі. Интрузияны анықтау жүйелері (Intrusion detection systems – IDS) келесідей жіктелуі мүмкін.

- **Хост IDS .** Бір хосттың сипаттамаларына және ондағы оқиғаларға күдікті әрекеттер тұрғысынан ағымдағы бақылауды жүзеге асырады.
- **IDS желісі.** Күдікті әрекеттерді анықтау мақсатында нақты желілік сегменттердің немесе құрылғылардың желілік трафигіне мониторинг жүргізеді және желілік, көліктік және қолданбалы хаттамаларды талдайды. IDS келесі логикалық компоненттерден тұрады.
- **Датчиктер.** Деректерді жинауға жауапты. Датчиктің кірісі жүйенің кез-келген бөлігі болуы мүмкін, ол басып кіру туралы куәлікті қамтуы мүмкін. Сенсорға арналған кіріс түрлеріне желілік пакеттер, Журнал файлдары және жүйелік қоңырауларды бақылау кіреді. Датчиктер деректерді жинап, оны анализаторға бағыттайды.
- **Анализаторлар.** Бір немесе бірнеше датчиктерден немесе басқа анализаторлардан кірістерді қабылдайды және болған кірісті анықтауға жауап береді. Нәтижелер шабуыл болды деген қорытындыны растайтын дәлелдерді қамтуы мүмкін.
- **Қолданушы интерфейсі.** IDS үшін мұндай интерфейс пайдаланушыға жүйенің шығуын көруге немесе оның әрекетін басқаруға мүмкіндік береді. Кейбір жүйелерде пайдаланушы интерфейсін басқару, басқару немесе консоль компонентімен теңестіруге болады.

Интрузияны анықтау жүйелері әдетте зиян келтірушілердің де, зиянды бағдарламалардың да зиянды әрекеттерін анықтауға арналған.

Аутентификация

Компьютерлік қауіпсіздік контексттерінің көпшілігінде пайдаланушының аутентификациясы негізгі стандартты блок және бірінші қорғаныс желісі болып табылады. Пайдаланушының аутентификациясы қол жетімділікті басқару мен пайдаланушыларды есепке алудың көптеген түрлеріне негіз болады. RFC 49 құжатында пайдаланушының аутентификациясы жүйелік объект талап ететін немесе талап ететін сәйкестендіруді тексеру процесі ретінде анықталады. Аутентификация процесі келесі қадамдардан тұрады:

1. **Идентификация.** Қорғаныс жүйесіне идентификатор беру. Идентификаторларды мұқият тағайындау керек, өйткені пайдаланушылардың аутентификацияланған тұлғалары басқа қызметтердің, соның ішінде кіруді басқару қызметтерінің жұмыс істеуіне негіз болады.
2. **Верификация.** Объект пен идентификатор арасындағы байланысты растайтын аутентификациялық ақпаратты ұсыну немесе қалыптастыру. Берілген пайдаланушының идентификаторымен байланысты аутентификация ақпаратының әдеттегі элементі құпия сақталған құпия сөз болып табылады. Пайдаланушы идентификаторы мен құпия сөздің тіркесімі әкімшілерге пайдаланушыға кіру құқығын орнатуға және оның жүйедегі әрекеттерін тексеруге мүмкіндік береді.

Негізінде, сәйкестендіру пайдаланушы жүйе талап ететін сәйкестендіру ақпаратын ұсынатын құрал ретінде қызмет етеді. Пайдаланушының аутентификациясы ұсынылған сәйкестендіру ақпаратының дұрыстығын анықтайтын құрал ретінде қызмет етеді.

Пайдаланушының жеке басын растау үшін келесідей жеке және белгілі бір комбинацияда қолдануға болатын төрт жалпы құрал бар:

1. Субъектіге белгілі зат. Мысал ретінде парольді айтуға болады, жеке пайдаланушы идентификаторы (PIN) және алдын-ала дайындалған бірқатар сұрақтарға жауап беріледі.

2. Субъектіге тиесілі нәрсе. Электрондық кілт карталары, смарт карталар және физикалық кілттер мысал бола алады. Аутентификацияның бұл түрі токен деп аталады.

3. Субъектіні білдіретін нәрсе (статикалық биометрия). Мысал ретінде саусақ іздері, торлы қабық және бет арқылы тұлғаны тану болып табылады.

4. Субъект жасаған нәрсе (динамикалық биометрия). Мысал ретінде дауыстық тембр, қолжазба және пернетақтадағы теру жылдамдығы бойынша жеке тұлғаны тану болып табылады.

Егер сіз осы әдістердің барлығын тиісті түрде қолдансаңыз, олар пайдаланушының қауіпсіз аутентификациясын қамтамасыз ете алады. Дегенмен, бұл әдістердің әрқайсысының өзіндік кемшіліктері бар.

Қол жеткізуді басқару

Әрбір жүйелік ресурсқа кім және не жеке қол жеткізе алатынын және әр жағдайда қол жеткізудің қандай түріне рұқсат етілетінін анықтайтын қауіпсіздік стратегиясын жүзеге асырады.

Қол жеткізуді басқару механизмі пайдаланушы (немесе пайдаланушы атынан әрекет ететін процесс) мен қолданбалар, операциялық жүйелер, брандмауэрлер, маршрутизаторлар, файлдар және дерекқорлар сияқты жүйелік ресурстар арасында делдал ретінде қызмет етеді. Алдымен жүйе кіру мүмкіндігін іздейтін пайдаланушының аутентификациясын жасай алады. Әдетте, аутентификация мүмкіндігі пайдаланушыға жүйеге мүлдем кіруге рұқсат етілгенін анықтайды, содан кейін пайдаланушы сұраған нақты кіруге рұқсат етілгенін анықтайды. Қауіпсіздік әкімшісі аутентификация дерекқорын жүргізеді, онда осы пайдаланушыға белгілі бір ресурстарға қол жеткізуге рұқсат етілетін түрі көрсетіледі. Қол жеткізуді басқару функциясы осы пайдаланушыға рұқсат беру керек пе, жоқ па, соны анықтау үшін осы дерекқорға жүгінеді. А тексеру функциясы ағымдағы бақылауды жүзеге асырады және пайдаланушының жүйелік ресурстарға қол жеткізу әрекеттерін есепке алады.

Брандмауэрлер

Брандмауэрлер жергілікті жүйені немесе тіпті бүкіл жүйелер желісін қауіпсіздіктің желілік қауіптерінен қорғаудың тиімді құралы бола алады, сонымен бірге ғаламдық желілер мен интернет арқылы сыртқы әлемге қол жеткізуге мүмкіндік береді. Дәстүр бойынша, желіге қосылған компьютерлерде сақталған құпия деректер файлдарын қорғау үшін желі арқылы сыртқы компьютерлерге қосылатын және оған арнайы енгізілген сақтық шараларын қолданатын арнайы арнайы компьютер брандмауэр ретінде қызмет етеді. Ол сыртқы желіге, әсіресе интернет қосылымдары мен коммутациялық байланыс желілеріне қызмет көрсету үшін қолданылады. Аппараттық және бағдарламалық жасақтамада жүзеге асырылатын және бір жұмыс станциясымен немесе компьютермен байланысқан жеке брандмауэрлер де кең таралған.

Брандмауэр дизайны келесі мақсаттарды көздейді.

1. Барлық трафик ішінен сыртқа және артқа брандмауэр арқылы берілуі керек. Бұған брандмауэр арқылы кіруден басқа барлық жергілікті желіге кіруді физикалық түрде бұғаттау арқылы қол жеткізіледі.
2. Брандмауэр арқылы тек жергілікті қорғаныс ережелерімен рұқсат етілген желілік график өтуі мүмкін.

3. Брандмауэрдің өзі енуге арналған. Бұл қауіпсіз операциялық жүйемен қорғалған жүйені қолдануды білдіреді. Брандмауэрді орналастыру үшін мемлекеттік мекемелерде жиі қолданылатын сенімді есептеу жүйелері қолайлы.

14.2. БУФЕРЛІК ТОЛЫП КЕТУ

Негізгі және виртуалды жад қауіпсіздікті бұзу қаупі бар жүйелік ресурстар болып табылады, сондықтан оларды қорғау үшін тиісті қарсы шаралар қабылдау қажет. Қауіпсіздіктің ең айқын талабы-жедел жадтағы процестердің мазмұнына рұқсатсыз кірудің алдын алу. Сонымен, егер процеске бөлінген жадтың бір бөлігі ортақ деп жарияланбаса, онда оның мазмұны басқа процестер үшін қол жетімді болмауы керек. Егер процеске бөлінген жадтың бір бөлігі басқа тағайындалған процестермен ортақ деп жарияланса, онда операциялық жүйенің қауіпсіздік қызметі жадтың осы бөлігіне тек уәкілетті процестерге қол жеткізуге кепілдік беруі керек.

Буферлік толып кету түріндегі шабуылдар

Буферлік толып кету, әйтпесе буфердің шекарасынан шығуы деп аталады, NIST (ұлттық стандарттар және технологиялар институты – АҚШ ұлттық стандарттар және технологиялар институты) ұйымының Glossary of key Information Security Terms (ақпараттық қауіпсіздік бойынша негізгі терминдер сөздігі) құжатында келесідей анықталады.

Буфердің толып кетуі-бұл интерфейстің күйі, онда буферде оның өлшемінен көп кіріс орналастырылады, сондықтан олар басқа деректердің орнына қайта жазылады. Шабуылдаушы жүйені қабілетсіз ету немесе жүйені өз бақылауына алуға мүмкіндік беретін арнайы дайындалған кодты енгізу үшін осындай жағдайды теріс пайдаланады.

Буфердің толып кетуі бағдарламалау қатесінің нәтижесінде орын алуы мүмкін, бұл процесс деректерді тұрақты өлшемді буферден тыс сақтауға тырысады, сондықтан көршілес жад ұяшықтарын қайта жазады. Бұл ұяшықтарда басқа бағдарламаның айнымалылары немесе параметрлері немесе бағдарламаны басқару ағынынан алынған деректер, соның ішінде қайтарылған мекенжайлар мен алдыңғы стек жақтауларына көрсеткіштер болуы мүмкін. Буферді стекке, үйіндіге немесе процесс деректерінің бөліміне бөлуге болады. Мұндай қатенің салдарына бағдарламада қолданылатын деректердің бүлінуі, бағдарламада басқарудың көзделмеген берілуі, жадқа қол жетімділіктің бұзылуы, сондай-ақ бағдарламаның ең ықтимал аяқталуы жатады. Егер буфердің толып кетуі жүйеге шабуылдың бір бөлігі ретінде әдейі жасалса, онда басқару шабуылдаушы таңдаған кез-келген кодқа берілуі мүмкін, нәтижесінде ол шабуыл процесінің артықшылықтарымен еркін кодты орындауға мүмкіндік алады. Буферлік толып кету шабуылдары қауіпсіздікті бұзатын шабуылдардың ең басым және қауіпті түрлерінің бірі болып табылады.

Осал бағдарламаларды олардың бастапқы кодын және бағдарламалардың орындалуын бақылау арқылы анықтауға болады, өйткені кірістердің шамадан тыс көлемі өңделеді немесе қауіпсіздікті автоматты түрде тестілеу сияқты әдістерді қолдана отырып, ықтимал осал бағдарламаларды автоматты түрде анықтау үшін ерікті түрде енгізілген кірістерді қолдануды білдіреді.

Компиляция уақытын қорғау

Стектегі буфердің толып кетуін анықтау және оны пайдалану қиын емес. Бұл соңғы жиырма жылдағы жүйелердің көптеген бұзылуларын анық көрсетеді. Сондықтан жүйелерді мұндай шабуылдардан олардың алдын алу немесе кем дегенде анықтау арқылы қорғаудың шұғыл қажеттілігі бар. Стектің толып кетуіне қарсы шараларды екі санатқа бөлуге болады.

1. Бағдарламалардың шабуылға төзімділігін арттыруға бағытталған компиляция уақытын қорғау.

2. Орындалатын бағдарламалардағы шабуылдарды анықтауға және тоқтатуға бағытталған жұмыс уақытын қорғау.

Соңғы екі онжылдықта қолайлы қорғаныс құралдары пайда болғанына қарамастан, қазіргі уақытта осал бағдарламалар мен жүйелердің өте үлкен базасы оларды орналастыруға кедергі келтіреді. Осы жерден операциялық жүйелерде және олардың

жаңартуларында орналастырылуы мүмкін және бар осал жүйелерді қорғауды қамтамасыз ететін жұмыс уақытын қорғау құралдарына қызығушылық туындайды.

Бағдарламалау тілін таңдау

Мүмкіндіктердің бірі-бағдарламаны қазіргі заманғы жоғары деңгейлі бағдарламалау тілінде жазу, онда айнымалылардың түрлері және олардың үстіндегі рұқсат етілген операциялар қатаң анықталған. Мұндай тілдер буфердің толып кету шабуылына қарсы тұрады, өйткені олардың компиляторлары кодты рұқсат етілген шекарадан шығудың автоматты тексерулерімен толықтырады, сондықтан бағдарламашыға бұл кодты қолмен жазудың қажеті жоқ. Бірақ бұл бағдарламалау тілдерінің икемділігі мен қауіпсіздігі үшін пайдаланылған ресурстар компиляция кезінде де, жұмыс уақытында да төленуі керек, мұнда әр түрлі тексерулердің қосымша кодын орындау қажет, соның ішінде буфер өлшемдерімен шектелген шектеулермен байланысты.

Қауіпсіз бағдарламалау әдістері

Егер С сияқты бағдарламалау тілдері қолданылса, бағдарламашылар олардың мекен-жайлары мен көрсеткіштерімен жұмыс істеу қабілетін ескеріп, жедел жадқа тікелей қол жеткізуді қамтамасыз етуі керек, бұл үшін ақы төлеу керек. С тілі жүйелік бағдарламалауға арналған, ал оған жазылған бағдарламалар қазіргі кездегіге қарағанда әлдеқайда кіші және шектеулі жүйелерде орындалады. С әзірлеушілері қауіпсіздікке қарағанда пайдаланылатын кеңістіктің тиімділігі мен өнімділігіне көп көңіл бөлді.

Мұндай жүйелерді қорғау үшін бағдарламашы бастапқы кодты зерттеп, барлық қауіпті бағдарламалық жасақтама конструкцияларын қауіпсіз түрде қайта жазуы керек. Кейбір жағдайларда бұл процесс буфердің толып кетуіне байланысты жүйелердің осалдықтарын пайдалану әрекеттерінің жылдам өсу қарқынын ескере отырып басталды.

Тілді кеңейту және қауіпсіз кітапханаларды қолдану

Массивтер мен көрсеткіштердің қауіпті қолданылуына байланысты С-де туындауы мүмкін қиындықтарды ескере отырып, мұндай жағдайларда компиляторларды автоматты түрде шетелге шығуды тексеретін етіп жетілдіру мақсатында бірқатар ұсыныстар енгізілді. Бірақ егер статикалық бөлінген массивтер үшін мұны істеу қиын болмаса, онда динамикалық бөлінген жад үшін бұл әлдеқайда қиын, өйткені компиляция кезінде оның көлемі туралы ақпарат жоқ. Бұл тапсырманы орындау үшін индекстердің семантикасын қол жетімді жадтың шекаралары туралы мәліметтерді қамтитындай етіп кеңейту керек, сонымен қатар шекаралық мәндердің дұрыс орнатылуын қамтамасыз ету үшін кітапхана функцияларын пайдалану қажет. Мұндай әдістер қауіпсіздік шараларын қабылдауды қажет ететін барлық бағдарламалар мен кітапханалардың өзгертілген компиляторының көмегімен қайта құрастыруды қарастырады. Бұл операциялық жүйенің және онымен байланысты утилиталардың жаңа шығарылымында мүмкін болғанымен, үшінші тарап қосымшаларында әлі де белгілі бір қиындықтар болады.

Стек қорғау механизмдері

Бағдарламаларды буфердің толып кету түріндегі классикалық шабуылдардан қорғаудың тиімді әдісі-оның зақымдануының кез келген дәлелін анықтау үшін стек жақтауын дайындау және кейіннен тексеру үшін функцияның кіру және шығу кодын енгізу. Егер кадрдың кез-келген модификациясы анықталса, шабуылдың жалғасуына жол бермеу үшін бағдарламаның орындалуы дереу тоқтатылады. Осындай қорғауды қамтамасыз ететін бірнеше тәсілдер бар.

Стек қорғанысы-белгілі қорғаныс механизмдерінің бірі. Бұл функцияның кірісі мен шығысында қосымша кодты енгізетін GCC компиляторының кеңейтімі. Функция кірісіне қосылған код жергілікті айнымалылар үшін орын бөлмес бұрын, канария мәнін (canary) бұрынғы стек жақтауының меңзер мекенжайынан төмен жазады. Функциядан шығуға қосылған код функциядан әдеттегі шығу операцияларын жалғастырмас бұрын, бұрынғы стек жақтауының көрсеткішін қалпына келтіріп, басқаруды қайтару мекен-жайына жібермес бұрын "канария" мәнінің өзгергенін тексереді. Осылайша, мұндай әрекет бірден

анықталып, бағдарламаның дереу үзілуіне әкеледі. Мұндай қорғаныс механизмі сәтті жұмыс істеуі үшін канарияның мәнін анықталмайтын етіп жасау өте маңызды және ол әр түрлі жүйелерде әр түрлі болуы керек. Егер олай болмаса, шабуылдаушы командалық қабықты іске қосу коды канарияның тиісті мәнін дұрыс жерге орналастырғанына көз жеткізеді. Әдетте, процесті құру кезінде канарияның мәні ретінде кездейсоқ мән таңдалады, ол берілген процестің күйінің бөлігі ретінде сақталады. Содан кейін бұл мән функцияның кірісі мен шығысына қосылған кодпен қолданылады.

Бұл тәсілді қолдану белгілі бір қиындықтарды тудырады. Біріншіден, қорғауды қажет ететін барлық бағдарламалар қайта құрастырылуы керек. Екіншіден, стек жақтауының құрылымы өзгереді және бұл стек жақтауларын талдайтын түзеткіштер сияқты бағдарламаларда асқынулар тудыруы мүмкін.

Жұмыс уақытын қорғау

Бұрын айтылғандай, компиляция уақытын қорғау әдістерінің көпшілігі қолданыстағы жүйелерді қайта құрастыруды қажет етеді. Осы жерден қолданыстағы осал бағдарламаларды қорғауды қамтамасыз ету үшін операциялық жүйелерді жаңарту ретінде орналастыруға болатын жұмыс уақытын қорғау құралдарына қызығушылық туындайды. Мұндай қорғау құралдары процестердің виртуалды мекен-жайларының кеңістігіне бөлінген жад аймағын басқарудағы өзгерістерді қарастырады. Бұл өзгерістер жедел жад аймақтарының қасиеттерін өзгертуге де, шабуылдардың көптеген түрлері бағытталған буферлердің орналасуын болжауды қиындатуға, сондықтан олардың алдын алуға да қызмет етеді.

Орындалатын модульдердің мекенжай кеңістігін қорғау

Буфердің толып кету түріндегі көптеген шабуылдарға машина кодын мақсатты буферге көшіру, содан кейін оған басқару элементтерін беру кіреді. Мұндай шабуылдардан қорғау үшін, мысалы, орындалатын код процестің мекен-жай кеңістігінің басқа жерінде болуы керек деген негізде кодтың орындалуын блоктауға болады.

Бұл мүмкіндікті тиімді қолдау үшін виртуалды жад беттерін процессордың жадыны басқару блогында (memory management unit - MMU) орындалмайтын деп белгілеу қажет. Стек пен үйінді жедел жадтың орындалмайтын аймағына айналғандықтан, қолданыстағы бағдарламаларды буфердің толып кетуіне негізделген шабуылдардың көптеген түрлерінен қорғаудың жоғары дәрежесі қамтамасыз етіледі және бұл тәжірибе операциялық жүйелердің соңғы шығарылымдарының жиынтығында стандартты болды. Дегенмен, орындалатын кодты стекке орналастыруды қажет ететін бағдарламаларды қолдауға байланысты бір қиындық қалады. Бұл, атап айтқанда, динамикалық компиляторларда болуы мүмкін (мысалы, Java атқарушы жүйесінде қолданылады). Стекте орындалатын Код С-да, сондай-ақ Linux-та сигнал өңдегіштерде кірістірілген функцияларды жүзеге асыру үшін қолданылады. Дегенмен, қолданыстағы бағдарламаларды қорғаудың және жүйелердің сенімділігін арттырудың бұл әдісі ең жақсылардың бірі болып саналады.

Мекенжай кеңістігін рандомизациялау

Орындау уақытын осы жерде қарастырылған шабуыл түрлерінен қорғаудың тағы бір әдісі процестің адрестік кеңістігінде негізгі деректер құрылымдарының орналасуымен жұмыс істеуді білдіреді. Естеріңізге сала кетейік, стектегі буфердің толып кету түріндегі классикалық шабуылды жүзеге асыру үшін шабуылдаушы мақсатты буфердің шамамен орналасқан жерін болжай алуы керек. Шабуылдаушы командалық қабықтың іске қосу кодын басқару үшін пайдалану үшін қолайлы қайтару мекенжайын анықтау мақсатында осы орынның шабуылында болжанған мекенжайын пайдаланады. Мұндай болжауды едәуір қиындататын әдістердің бірі-әр процесс үшін стек орналастырылатын мекен-жайды ерікті түрде өзгерту. Қазіргі заманғы процессорларда қол жетімді мекен-жайлар ауқымы өте кең (32 бит) және көптеген бағдарламалар оның аз ғана бөлігін қажет етеді. Демек, мегабайттық мекен-жай кеңістігінде стекке бөлінген жад аймағын жылжыту көптеген бағдарламалардың жұмысына минималды дәрежеде әсер етеді, бірақ сонымен бірге жедел жадтағы мақсатты буфердің орналасу мекен-жайын болжау мүмкін емес.

Мұндай шабуылдардың тағы бір мақсаты-стандартты кітапхана функцияларының орналасуы. Қорғауды айналып өтуге тырысқанда (мысалы, орындалмайтын стектер), буфердің толып кетуіне негізделген шабуылдардың кейбір түрлері стандартты кітапханаларда бар бастапқы кодтың осалдығын пайдаланады, олар әдетте бір бағдарламада бір мекен-жайға жүктеледі. Шабуылдың бұл түрімен күресу шарасы ретінде сіз виртуалды жадта орналасқан мекен-жайлар сияқты бағдарламадағы стандартты кітапханаларды кездейсоқ жүктеу тәртібін анықтайтын қауіпсіздік кеңейтімін пайдалана аласыз.

Қауіпсіздік беттері

Мұнда қарастырылған шабуыл түрлерінен қорғану үшін қолдануға болатын соңғы әдіс процестердің мекенжай кеңістігіндегі маңызды жад аймақтары арасында қорғаныс беттерін (guard pages) орналастыруды қамтиды. Бұл жағдайда процестің әдеттегіден әлдеқайда көп виртуалды жады бар екендігі қолданылады. Мекенжай кеңістігінің әрбір құрамдас бөлігі үшін пайдаланылатын мекенжай ауқымдары арасында бос орындар орналастырылады. Мұндай бос орындар (қауіпсіздік беттері) жадыны басқару блогында жарамсыз мекенжайлар ретінде белгіленеді және оларға қол жеткізуге кез келген әрекет бірден процестің үзілуіне әкеледі. Осының арқасында процестердің адрестік кеңістігінде іргелес аймақтарды қайта жазуға тырысатын буфердің толып кетуіне (әдетте жаһандық деректерге) жол берілмейді.

14.3. Қол жеткізуді басқару

Қол жеткізуді басқару – бұл операциялық жүйе, файлдық жүйе деңгейінде немесе осындай басқарудың бірдей принциптері қолданылатын екі деңгейде де орындалатын функция.

Файлдық жүйеге кіруді басқару

Жүйеге сәтті кіргеннен кейін пайдаланушыға бір немесе бірқатар хосттар мен қосымшаларға қол жетімділік беріледі. Бірақ бұл әдетте оның дерекқорында сақталған құпия ақпараты бар жүйеде жұмыс істеу үшін жеткіліксіз болып шығады. Пайдаланушының қол жетімділігін басқару процедурасы арқылы оның жеке басын жүйеге орнатуға болады. Тиісті профиль әр пайдаланушымен байланысты болуы мүмкін, онда рұқсат етілген операциялар мен файлдарға қол жеткізу түрлері анықталады, осылайша операциялық жүйе пайдаланушы профиліне негізделген ережелердің орындалуын қамтамасыз ете алады. Бірақ дерекқорды басқару жүйесі жеке жазбаларға немесе тіпті олардың бөліктеріне қол жеткізуді басқаруы керек. Мысалы, әкімшіліктің кез-келген өкіліне компания қызметкерлерінің тізімін алуға рұқсат етілуі мүмкін, бірақ тек сайланған адамдар жалақы туралы мәліметтерге қол жеткізе алады. Бұл жай ғана егжей-тегжейлі деңгей мәселесі емес. Егер операциялық жүйе пайдаланушыға файлға немесе қосымшаға кіруге рұқсат бере алса, содан кейін қауіпсіздікті тексеру жүргізілмесе, онда дерекқорды басқару жүйесі әр кіру әрекеті үшін жеке шешім қабылдауы керек. Мұндай шешім тек пайдаланушының жеке басына ғана емес, сонымен қатар қол жетімді деректердің жекелеген бөліктеріне, тіпті пайдаланушыға берілген ақпаратқа да байланысты болады.

Файлдық жүйеде немесе дерекқорды басқару жүйесінде қолданылатын қол жеткізуді басқарудың жалпы моделі қол жеткізу матрицасы болып табылады (access matrix, 14.1-сурет, а).

Төменде осындай модельдің негізгі элементтері келтірілген.

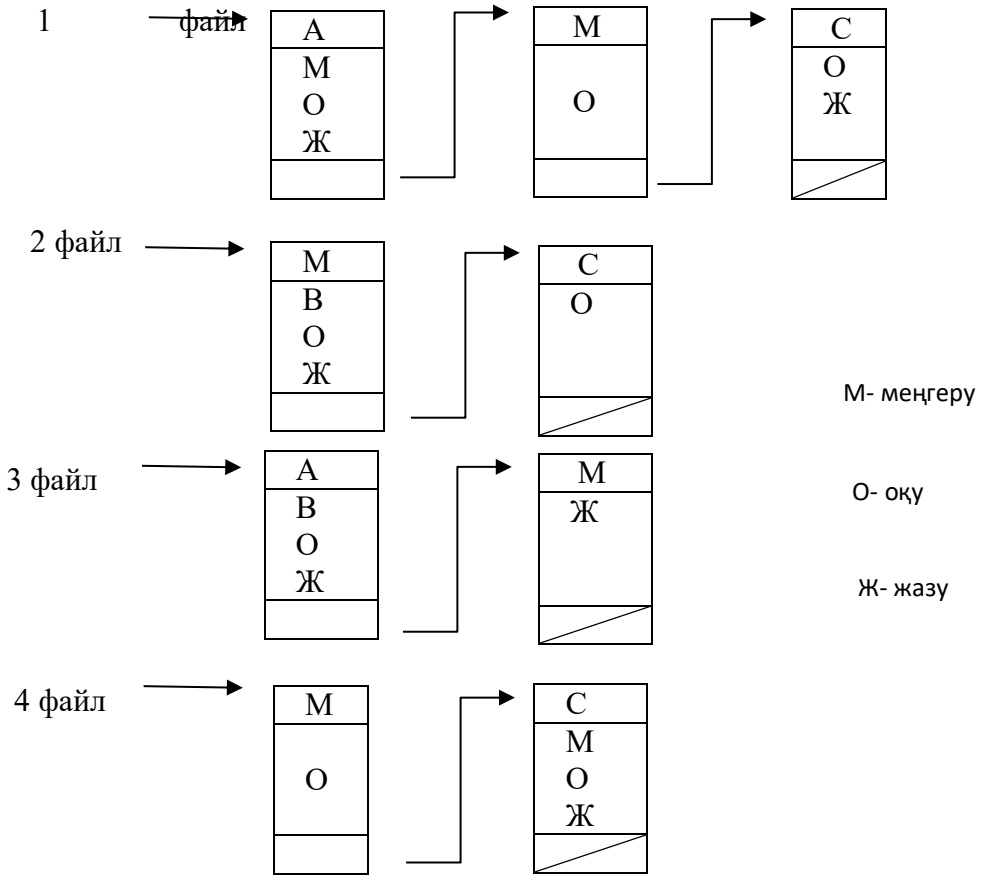
- **Субъект.** Бұл объектілерге қол жеткізе алатын тұлға. Әдетте, субъект ұғымы процесс ұғымына теңестіріледі.
- **Объект.** Мұның бәрі қол жетімділікті басқаруға жатады. Мысал ретінде - файлдар және олардың жеке бөліктері, бағдарламалар, жедел жад бөлімдері және бағдарламалық жасақтаманы айтуға болады.
- **Қол жеткізу құқығы.** Бұл субъектінің объектіге қол жеткізу тәсілі. Мысал ретінде оқу, жазу, орындау және бағдарламалық жасақтама объектілеріндегі функцияларды келтіруге болады.

Қол жеткізу құқығы матрицасының бір өлшемі деректерге қол жеткізуге тырысуы мүмкін анықталған субъектілерден тұрады. Кіру құқықтары матрицасының екінші өлшемі қол жетімді объектілерден тұрады. Бөлшектердің ең жоғары деңгейінде объектілер жеке деректер өрістері бола алады. Бұл матрицадағы объектілер неғұрлым үлкейтілген топтар болуы мүмкін.

Іс жүзінде кіру құқығының матрицасы әдетте сирек кездеседі және ыдыраудың екі әдісінің бірімен жүзеге асырылады. Атап айтқанда, кіру құқығының матрицасы қол жеткізу тізімдерін алу үшін бағандарға бөлінуі мүмкін (access control list, 14.1-сурет, б). Осылайша, әр объект үшін пайдаланушылар мен оларға берілген кіру құқықтарын тізімдейтін кіру тізімі бар.

Объектілер

Субъектілер	1 файл	2 файл	3 файл	4 файл
А пайдаланушысы	Меңгеру Оқу Жазу		Меңгеру Оқу Жазу	
В пайдаланушысы	Оқу	Меңгеру Оқу Жазу	Жазу	Оқу
С пайдаланушысы	Оқу Жазу	Оқу		Меңгеру Оқу Жазу



Сурет 14.1. Қол жеткізуді басқару құрылымдарының мысалы

Жолдарға кіру матрицасының ыдырауы соңында рұқсат тізімдерін береді (capability tickets, 14.1-сурет, в). Рұқсат пайдаланушыға рұқсат етілген нысандар мен операцияларды анықтайды. Әрбір пайдаланушының бірқатар рұқсаттары бар және оларға басқа пайдаланушылардан сұрау немесе оларды басқа пайдаланушыларға беру құқығы берілуі мүмкін. Желідегі деректерге қол жеткізуді басқару мәселелері пайдаланушылардың қол жеткізуін басқарумен қатар қарастырылуы керек. Сонымен, егер кейбір пайдаланушыларға белгілі бір деректер элементтеріне қол жеткізуге рұқсат етілсе, онда қорғау мақсатында бұл деректер элементтерін рұқсат етілген пайдаланушыларға беру кезінде оларды қорғау үшін шифрлау қажет болуы мүмкін.

Қол жеткізуді басқару стратегиялары

- Қол жеткізуді басқару стратегиялары қол жеткізудің нақты түрлерін, қандай жағдайда және кімге рұқсат етілетінін белгілейді. Мұндай стратегиялар әдетте келесі санаттарға бөлінеді.
- **Дискрециялық қол жеткізуді басқару (discretionary access control – DAC**). Сұрау салушы тұлғаның жеке басы және Сұрау салушы субъектіге не кіруге рұқсат етілгенін (немесе рұқсат етілмегенін) белгілейтін қол жеткізу (авторизация) қағидалары негізінде жүзеге асырылады.
- **Мәжбүрлеп қол жеткізуді басқару (mandatory access control-ХАА).** Жүйелік ресурстардың құпиялылық дәрежесін немесе маңыздылығын білдіретін құпиялылық белгісін (грифін) қандай жүйелік объектілердің белгілі бір ресурстарға қол жеткізуге құқығы бар екенін білдіретін құпияларға рұқсатпен салыстыру негізінде жүзеге асырылады.
- **Рөлдік қол жетімділікті басқару (role-based access control – RBAC).** Ол жүйеде пайдаланушылардың рөлдері, сондай-ақ берілген рөлдері бар пайдаланушыларға қол жетімділіктің нақты түрлерін белгілейтін ережелер негізінде жүзеге асырылады.
- **Атрибуттық қатынасты басқару (attribute-based access control – Avas).** Ол пайдаланушының атрибуттары, қол жеткізуді қажет ететін ресурстар, сондай-ақ ағымдағы сыртқы шарттар негізінде жүзеге асырылады.

DAC-қол жеткізуді басқаруды жүзеге асырудың дәстүрлі әдісі. Бұл әдіс файлға қол жеткізуді басқаруды қарастырған кезде енгізілген. ХАА тұжырымдамасы әскери сипаттағы ақпаратты қорғауға қойылатын талаптардан туындады. Rbac және Avas стратегиялары барған сайын танымал бола бастады.

14.4. Операциялық жүйелерді қорғауды күшейту

Қорғауды күшейтудің алғашқы маңызды қадамы-барлық басқа қосымшалар мен қызметтер сүйенетін негізгі операциялық жүйені қорғау. Қауіпсіздіктің астына берік негіз қалау үшін тиісті түрде орнатылған, соңғы түзетулермен жаңартылған және конфигурацияланған операциялық жүйе қажет. Өкінішке орай, көптеген операциялық жүйелер үшін стандартты конфигурацияда көбінесе қауіпсіздікке емес, қолдану мен жұмыс істеудің қарапайымдылығына баса назар аударылады. Әр ұйымның қауіпсіздікке деген қажеттілігі әр түрлі болғандықтан, оның өзіндік профилі бар, демек, қорғаныс жүйесінің конфигурациясы бар. Жүйенің қауіпсіздігіне қойылатын нақты талаптар жаңа талқыланғаннан кейін жоспарлау сатысында қойылуы керек.

Іске асырудың егжей-тегжейіндегі айырмашылықтарға қарамастан, операциялық жүйелерді қорғауды күшейтудің жалпы тәсілі бірдей болып қалады. Ең кең таралған операциялық жүйелердің көпшілігінде конфигурация бойынша нұсқаулықтар мен

әрекеттер тізімі бар, сондықтан оларға жүгінген жөн, бірақ әр ұйымның және оның жүйелерінің нақты қажеттіліктерін әрқашан ескеру қажет. Кейбір жағдайларда автоматтандырылған құралдар жүйелік конфигурацияны қорғауды күшейтуге қосымша көмек көрсете алады.

[177] операциялық жүйені қорғауды күшейту үшін келесі негізгі қадамдарды ұсынады.

- Амалдық жүйені және оған арналған барлық жаңартуларды орнатыңыз.
- Амалдық жүйені төмендегі қадамдарды орындау арқылы қауіпсіздік қажеттіліктерін дұрыс қанағаттандыру үшін күшейтіңіз және конфигурациялаңыз.
- Қажетсіз қызметтерді, қолданбаларды және хаттамаларды жою.
- Пайдаланушыларды, топтарды және өкілеттіктерді конфигурациялау.
- Ресурстарды басқару құралдарын конфигурациялау.
- Қосымша қорғаныс құралдарын, брандмауэрлерді және кіруді анықтау жүйелерін орнату және конфигурациялау.

Негізгі амалдық жүйенің қорғанысын тексеріп, оны қорғауды күшейту үшін жасалған қадамдар оның қауіпсіздік қажеттіліктерін дұрыс қанағаттандыратынына көз жеткізіңіз.

14.5. Қауіпсіздікті сақтау

Жүйе дұрыс салынғаннан, қорғалғаннан және орналастырылғаннан кейін оның қауіпсіздігін үздіксіз сақтау процесі басталады. [177] бұл қауіпсіздікті сақтау процесі келесі қосымша қадамдарды қамтуы керек деп болжайды.

- Хаттамаланатын ақпаратты ағымдағы бақылау және талдау.
- Тұрақты сақтық көшірме жасау.
- Қауіпсіздік ережелерін бұзушылықтарды жою.
- Жүйені қорғауды жүйелі түрде тестілеу.

Сапа барлық жауапты бағдарламалық жасақтаманы жаңарту үшін бағдарламалық жасақтаманы сүйемелдеу үшін, сондай-ақ қажет болған жағдайда ағымдағы бақылау және конфигурацияны қайта қарау үшін тиісті процестерді орындау.

Хаттама

[177] атап өткендей, "хаттама жасау қауіпсіздіктің негізі болып табылады". Хаттама-бұл жүйеде туындаған ақаулар туралы ғана хабарлауға қабілетті бақылау құралы. Бірақ жүйе бұзылған немесе істен шыққан жағдайда тиімді хаттама жасау оның әкімшілеріне не болғанын тезірек және дәлірек анықтауға көмектеседі, сондықтан олардың күш-жігерін туындаған ақауларды түзетуге және жоюға тиімді бағыттайды. Тіркеу журналдарында жүйемен, желімен және қосымшалармен жасалуы мүмкін дұрыс деректер жазылуы керек. Әрі қарай, бұл деректер тиісті түрде талдануы керек. Протоколдау кезінде жиналған деректердің әртүрлілігі жүйені жоспарлау кезеңінде анықталуы керек, өйткені ол қорғаныс талаптарына және сервердің ақпараттық жабылуына байланысты. Хаттамалау кезінде ақпараттың едәуір көлемін жасауға болады, сондықтан оларды сақтау үшін жеткілікті орын бөлу өте маңызды. Сондай-ақ, тіркеу журналдарын автоматты түрде айналдыру және мұрағаттаудың қолайлы жүйесі конфигурациялануы керек, бұл хаттамаланатын ақпараттың барлық көлемін ретке келтіруді қамтамасыз етеді. Қолайсыз оқиғаларды анықтау үшін тіркеу журналдарын қолмен талдау көп уақытты қажет етеді және сенімсіз. Оның орнына кейбір автоматтандырылған талдауды қолданған дұрыс, өйткені бұл жүйеде күдікті әрекеттерді анықтауға мүмкіндік береді.

Деректердің сақтық көшірмесін жасау және мұрағаттау

Жүйеде деректердің үнемі сақтық көшірмесін жасау-бұл жүйелік және пайдаланушы деректерінің тұтастығын сақтауға көмектесетін тағы бір маңызды құрал. Деректерді сақтаудың заңды немесе өндірістік талаптары да бар. Сақтық көшірме - бұл бірнеше сағаттан бірнеше аптаға дейін салыстырмалы түрде қысқа уақыт ішінде жоғалған немесе бүлінген деректерді қалпына келтіруге мүмкіндік беретін белгілі бір уақыт аралығында деректердің көшірмелерін жасау процесі. Мұрағаттау (archive) - ескі деректерге қол

жеткізудің заңды немесе өндірістік талаптарын қанағаттандыру үшін айлар немесе жылдар бойы деректердің көшірмелерін ұзақ уақыт бойы сақтау процесі. Бұл процестер әр түрлі қажеттіліктерді қанағаттандырса да, бір-бірімен байланысты және ұйымдастырылуы сирек емес. Сақтық көшірме жасау мен мұрағаттауға қатысты қажеттіліктер мен стратегиялар жүйені жоспарлау кезеңінде анықталуы керек. Осы кезеңде қабылданған негізгі шешімдерге мыналар жатады: сақтық көшірмелерді қалай сақтау керек – желіде немесе желіден тыс, жергілікті немесе қашықтағы жерде. Мәмілеге келу шешімдеріне қауіпсіздіктің жоғарылауымен және әртүрлі қауіптерден сенімді қорғаныспен салыстырғанда іске асырудың қарапайымдылығы мен шығындар арасындағы таңдау кіреді.

ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1. Garg, R.; Verma, G.Operating Systems [OP]: An Introduction - Softcover Publisher: Mercury Learning & Information, 2017. 290 p.

2. https://gifer.com/ru/7h0m

3. https://3dnews.ru/1034959

4. Darrell Hajek, Cesar Herrera, Flor Narciso Principles of Operating Systems. Independently Published (24 April 2020) 176 pages.

5. Andrew S. Tanenbaum and Herbert Bos. Modern Operating Systems. 4/E.1136 pages, Pearson India, 2016.

6. Silberschatz Abraham, Galvin Peter Baer and Gadne Greg. Operating system concepts.

7. Amdahl GM (1967) Validity of the single-processor approach to achieve large scale computing capabilities. AFIPS Joint Spring Conference Proceedings 30 (Atlantic City, NJ, Apr. 18–20), AFIPS Press, Reston VA, pp 483–485.

8. https://studfile.net/.

9. https://habr.com/ru/post/40227/.

10. wikimedia.org

11. wordpress.com

12. blackandwhitecomputer.blog

13. http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf.

14. encyclopedia2.thefreedictionary.com

15. linustechtips.com

16. youtube.com/watch?v=w3K1JklY6D4