ОЖҚ. Дәріс №1. Қосымша қауіпсіздік шараларын қолдануды талап ететін сыртқы жағдайлар. Қауіптер. Қаскүнемдер.

Кейбір адамдар үшін "қауіпсіздік" және "қорғау" ұғымдары бір-бірін алмастырады. Соған қарамастан, бір жағынан техникалық, әкімшілік, құқықтық және саяси мәселелер кіретін тиісті өкілеттіктері жоқ адамдар тарапынан оқу және өзгерту үшін файлдардың қол жетімсіздігін қамтамасыз етудің негізгі мәселелерін және екінші жағынан қауіпсіздікті қамтамасыз ету үшін қолданылатын операциялық жүйенің нақты тетіктерін бір — бірінен ажырату пайдалы. Түсінбеушіліктерді болдырмау үшін, яғни жалпы жоспар мәселелеріне сілтеме жасау үшін қауіпсіздік (security) термині қолданылады, ал компьютерде бар ақпаратты қорғау үшін қолданылатын операциялық жүйенің нақты механизмдеріне сілтеме жасау үшін қорғаныс механизмдері (protection mechanisms) термині қолданылады. Дегенмен, олардың арасында нақты белгіленген шекара жоқ. Зерттелетін мәселенің табиғатын зерттеу үшін алдымен бар қауіптер мен бұзушылардың әрекеттері жағдайында қауіпсіздік мәселелеріне жүгінеміз. Содан кейін қауіпсіздікке қол жеткізуге көмектесетін қорғаныс механизмдері мен модельдерін қарастырыңыз.

Қауіпсіздік туралы көптеген еңбектерде ақпараттық жүйелердің қауіпсіздігі үш компонентке бөлінеді: құпиялылық, тұтастық және қол жетімділік. Бірге, барлық үш компонент жиі CIA деп аталады (сенімділік, тұтастық, қол жетімділік). Олар 1-кестеде көрсетілген және біз басқа CIA (ЦРУ, АҚШ-тың Орталық барлау басқармасы) міндеттеріне ұқсас бұзушылар мен тыңшылардан қорғануымызға қажетті қауіпсіздік қасиеттерінің негізін құрайды.

1-кесте. Қауіпсіздік міндеттері мен қауіптері

Міндеттері	**Қауіп-қатерлер**
Құпиялылық (конфиденциальность)	Деректердің қорғалмауы (незащищенность данных)
Біртұтастық (целостность)	Деректері жалған жасау (подделка данных)
Қол жетімділік (доступность)	Қызмет көрсетуден бас тарту (оказ от обслуживания)

Бірінші қасиет – **құпиялылық** (confidentiality) — деректердің құпиялылығын сақтауға бағытталған. Дәлірек айтқанда, егер белгілі бір деректердің иесі бұл деректер қатаң анықталған адамдар шеңберіне қол жетімді болуы мүмкін деп шешсе, жүйе бұған құқығы жоқ адамдарға қол жеткізе алмайтындығына кепілдік беруі керек. Кем дегенде, иесі кім және не көре алатындығын анықтай алуы керек, ал жүйе жеке файлдарға қатысты осы талаптардың орындалуын қамтамасыз етуі керек.

Екінші қасиет – **біртұтастық** (integrity) — қажетті құқықтары жоқ пайдаланушылар кез-келген деректерді иелерінің рұқсатынсыз өзгерте алмауы керек дегенді білдіреді. Бұл тұрғыда деректерді өзгерту оларға өзгертулер енгізуді ғана емес, оларды жоюды немесе оларға жалған деректерді қосуды да қамтиды. Егер жүйе оған енгізілген деректердің иесі оларды өзгерту туралы шешім қабылдағанға

дейін өзгермейтініне кепілдік бере алмаса, онда ол деректер қоймасы рөлін жоғалтады.

Үшінші қасиет – **қолжетімділік** (**availability**) — ешкім жүйенің жұмысын бұзып, оны істен шығара алмайды дегенді білдіреді. Қызмет көрсетуден бас тартуға әкелетін шабуылдар (**denial of service**, (**DOS**)) жиі кездеседі. Мысалы, егер компьютер интернет-сервер ретінде жұмыс істесе, онда оны үнемі сұраулардан бас тарту оның орталық процессорының барлық жұмыс уақытын кіріс сұраныстарын оқып-үйренуге және қабылдамауға алаңдатып, жұмыс қабілеттілігінен айыруы мүмкін. Егер кіріс веб-бетті оқу сұранысын өңдеу үшін 100 мкс қажет болса, секундына 10 000 сұраныс жібере алатын кез-келген адам серверді өшіре алады. Жеңе үшін, шабуыл, бағытталған бұзу құпиялылығын және тұтастығын деректерді таңдауға болады әбден қолайлы модельдері және технологиясы, ал қарсы шабуылдарға туғызатын жерге қызмет көрсету, айтарлықтай қиын. Кейінірек барлық мүмкін сценарийлер үшін үш негізгі қасиет жеткіліксіз деп шешілді және түпнұсқалылық (аутентификация), сәйкестендіру (есеп беру), атқыламау (**nonrepudiability**), жақындық (құпиялылық) және т.б. сияқты қосымша қасиеттер қосылды, әрине, барлық осы қасиеттерге ие болу жақсы болар еді. Бірақ сонымен бірге, үш бастапқы қасиет қауіпсіздік сарапшыларының көпшілігінің (құрметті) ақыл-ойы мен жүрегінде ерекше орын алады.

Жүйелерге бұзушылар үнемі қауіп төндіреді. Мысалы, әсіресе хакер егер деректерді алмасу хаттамасында шифрлау қолданылмаса жергілікті желі трафигіне қосылып, ақпараттың құпиялылығын бұзуы мүмкін. Сондай-ақ, шабуылдаушы дерекқорды басқару жүйесін бұзып, дерекқордың тұтастығын бұзып, кейбір жазбаларды жоя немесе өзгерте алады. Ақырында, техникалық қызмет көрсетуден бас тартуға себеп болатын шабуылдар бір немесе бірнеше компьютерлік жүйелердің қол жетімділігін бұзуы мүмкін. Бөтен адам жүйеге шабуыл жасай алатын көптеген әдістер бар. Олардың кейбіреулері осы тарауда сәл кейінірек қарастырылады.

Қазіргі уақытта көптеген шабуылдарды өте жақсы құралдар мен қызметтер қолдайды. Бұл құралдардың кейбірін қара шляпалар хакерлері (black-hat hackers), ал кейбірін ақ шляпалар (white hats) жасайды. Ескі батыстықтарға ұқсас, сандық әлемдегі жағымсыз кейіпкерлер қара шляпалар киіп, трояндық аттарға секіреді (Троян хорсес), ал жақсы хакерлер ақ шляпалар киіп, бағдарламаларды өз көлеңкесінен тезірек жасайды. Айтпақшы, танымал ақпараттық басылымдарда "хакер" деген жалпы терминді тек қара шляпаларға қатысты қолдану әдетке айналған. Бірақ компьютерлік әлемде хакер – ұлы программалаушыға бекітілген құрметті атақ. Олардың кейбіреулері, әрине, қолайсыз істермен айналысады, бірақ олардың көпшілігі өте лайықты адамдар. Біз бұл ұғымды өзінің бастапқы мағынасында қолданамыз, ал өздеріне тиесілі емес компьютерлік жүйелерді бұзуға тырысатын адамдар крэкерлер немесе бұзушылар (взломщики, crackers) немесе қара шляпалар деп аталады.

Бұзу құралдарына оралайық. Бір таңқаларлығы, олардың көпшілігін ақ шляпалар жасаған. Шындығында, қаскүнемдер оларды қолдана алады (және қолдана алады), бірақ бастапқыда бұл құралдар компьютерлік жүйелер мен желілердің

қауіпсіздігін тексерудің ыңғайлы құралы болды. Мысалы, nmap сияқты құрал бұзушыларға **портты сканерлеу** (portscan) арқылы компьютерлік жүйе ұсынатын Желілік қызметтерді анықтауға көмектеседі. Nmap ұсынатын қарапайым сканерлеу технологиясының бірі–компьютерлік жүйеде мүмкін болатын әрбір порт нөміріне TCP қосылымын жасау. Сәтті қосылу жүйеде осы портты тыңдайтын қызметтің болуын көрсетеді. Сонымен қатар, көптеген қызметтер кеңінен танымал порт нөмірлерін қолданатындықтан, қауіпсіздікті тексеретін адам (немесе крекер) машинада қандай қызметтер жұмыс істейтінін анықтай алады. Басқаша айтқанда, nmap-бұзушылар үшін де, қорғаушылар үшін де пайдалы құрал, яғни **қосарланған тағайындау** (қосарланған қызмет, dual use) деп аталатын қасиеті бар. Dsniff жалпы атауы бар тағы бір құралдар жиынтығы желілік трафикті бақылаудың және желілік пакеттерді қайта бағыттаудың әртүрлі тәсілдерін ұсынады. "Төмен орбиталық ион зеңбірегі" — Low Orbit Ion Cannon (LOIC) – бұл алыс галактикалардағы жауларды жоятын ғылыми-фантастикалық кітаптардағы қару ғана емес, сонымен қатар қызмет көрсетуден бас тартуға әкелетін шабуылдарды іске қосу құралы. Metasploit ортасымен бірге көптеген түрлі мақсаттарға қарсы зиянды коды бар жүздеген қолайлы құралдармен алдын-ала орнатылған, бұзу ешқашан оңай болған емес. Бұл құралдардың барлығы қосарланған тағайындауға ие екені түсінікті. Бірақ оларды абсолюттік жауыздық (жамандық) деуге болмас, мысалы, пышақтар мен балталарды еске алайық.

Сонымен қатар киберқылмыскерлер көптеген қызметтерді ұсынады (көбінесе онлайн): зиянды бағдарламаларды тарату, ақшаны жылыстату, трафикті қайта бағыттау, ешқандай сұрақсыз хостинг және тағы басқалар. Интернеттегі қылмыстық әрекеттердің негізгі бөлігі ботнеттер деп аталатын инфрақұрылымдарға негізделген, олар мыңдаған (кейде миллиондаған) жұқтырған компьютерлерден тұрады, олар көбінесе бұл туралы білмейтін қарапайым компьютерлер болып табылады. Бұзушылар арнайы машиналарды жұқтырудың көптеген жолдары бар. Мысалы, олар танымал бағдарламалардың ақысыз, бірақ зиянды кодпен толтырылған нұсқаларын ұсына алады. Ащы шындық, қымбат бағдарламалардың ақысыз (бұзылған) нұсқалары туралы уәде көптеген пайдаланушыларға әсер етпейді. Өкінішке орай, мұндай бағдарламаларды орнату хакерлерге машинаға толық қол жеткізуге мүмкіндік береді. Бұл сіздің үйіңіздің кілттерін бейтаныс адамдарға беру сияқты.

Компьютер крекердің бақылауына түскенде, ол бот (bot) немесе зомби (zombie) деп аталады. Әдетте пайдаланушы мұны байқамайды. Қазіргі уақытта ботнеттер жүздеген мың зомбиден тұрады, олар көптеген қылмыстық істердің жұмысшысы болып табылады. Бірнеше жүз мың дербес компьютерлер банктік деректемелерді ұрлау немесе спам жіберу үшін жеткілікті, және сіз өзіңіздің қару-жарағыңызды күдікті мақсатқа жетелеген миллиондаған зомбилермен қандай шайқасты ұйымдастыруға болатындығын ойластырған жөн. Кейде шабуылдың салдары компьютерлік жүйелерден әлдеқайда асып түседі және физикалық әлемге тікелей зиян келтіреді. Бір мысал: Брисбенге жақын орналасқан Квинсленд, Австралиядағы

Maroochy Shire рельефінің қалдықтарын басқару жүйелеріне шабуыл жасау. Кәріз жүйелерін монтаждау жөніндегі компанияның бұрынғы қызметкері Maroochy Shire кеңесі оның қызметтерінен бас тартып, олармен сөйлесуге шешім қабылдағанына наразы болды. Ол ағынды суларды тазарту жүйесін бақылауға алып, саябақтарда, өзендерде және жағалау суларында (барлық балықтар дереу қайтыс болған жерде), сондай-ақ басқа жерлерде миллиондаған литр ағынды суларды ағызуды ұйымдастырды. Жалпы, белгілі бір елге немесе этникалық топқа наразы немесе бүкіл әлемге ашуланған және зиянның сипаты туралы немесе олардың әрекеттерінің құрбаны кім болатыны туралы ойламай-ақ, барынша жойылуды қалайтын адамдар бар. Әдетте, мұндай адамдар жауларының компьютерлеріне шабуыл жасау жақсы нәрсе деп санайды, бірақ шабуылдардың өздері дәл бағыттала алмайды.

Әдетте Stuxnet деп аталатын кибершабуылдарды қолдану Иранның Натанзе қаласындағы (Natanz) уран байыту қондырғысында центрифугалардың физикалық жойылуына әкелді және бұл Иранның ядролық бағдарламасының айтарлықтай баяулауына себеп болды дейді. Бұл шабуыл үшін жауапкершілікті ешкім мойнына алмаса да, кейде бұл асқынулардың көзі Иранға қарсы бір немесе бірнеше елдердің құпия қызметтері деп аталады. Қауіпсіздіктің тағы бір аспектісі-құпиялылық (құпиялылық): жеке пайдаланушыларды жеке ақпаратқа байланысты теріс пайдаланудан қорғау. Бұл құқықтық және моральдық жоспардың көптеген мәселелерін білдіреді.

Қауіпсіздік мәселелеріне арналған әдебиеттерде өз қызметіне емес, басқа жұмыстарға араласатын адамдарды крекер (шабуылшылар), шабуылдаушылар (интрудер), кейде жаулар (adversaries) деп атайды. Бірнеше ондаған жыл бұрын компьютерлік жүйелерді бұзу тек достар алдында мақтаныш үшін жүзеге асырылды, бірақ қазір бұл жүйені бұзудың жалғыз немесе тіпті маңызды себебі емес. Ұрлық, хактивизм, вандализм, терроризм, кибер соғыс, тыңшылық, спам тарату, бопсалау, алаяқтық, кейде бұзушылар ұйымның қауіпсіздік жүйесінің нашар күйін көрсеткісі келеді. Хакерлердің де шебер емес еліктеушілерден бастап, кидди сценарийі (сценарий-кидди) деп те аталатын қара шляпаларға дейін, өте жоғары кәсіби крекерлерге дейін жіктемесі бар. Олар қылмыс, үкімет (мысалы, полиция, әскери немесе құпия қызметтер) немесе қауіпсіздік фирмалары немесе бос уақытында бұзақылық жасайтын әуесқойлар болуы мүмкін. Достас емес шет мемлекеттердің әскери құпияларын ұрлаудың алдын алу әрекеті студенттік әзілдер жүйесіне жібермеу әрекетінен түбегейлі ерекшеленетінін түсіну керек. Ақпараттың қауіпсіздігі мен қорғалуын қамтамасыз етуге бағытталған күш - жігердің көлемі, сөзсіз, қарсыластың нақты кім екеніне байланысты болады.

Операциялық жүйелердің қауіпсіздігі. Қорғалған жүйелерді құруға бола ма? Жоғары сенімді есептеу базасы.

Операциялық жүйелердің қауіпсіздігі

Компьютерлік жүйелердің қауіпсіздігін бұзудың көптеген жолдары бар. Мысалы, көптеген адамдар өздерінің PIN-кодтары үшін 0000 комбинациясын

орнатады немесе пароль ретінде " password" сөзін қолданады — оларды есте сақтау оңай, бірақ қауіпсіздіктің жоғары деңгейін қамтамасыз етпейді. Кейбір адамдар есте сақтау мүмкін емес өте күрделі парольдерді де ойлап табады және оларды мониторға немесе пернетақтаға орналастырылған жапсырмаларға жазуға мәжбүр болады. Машинаға физикалық қол жетімділігі бар кез-келген адам (тазалаушылар, хатшылар және барлық келушілерді қоса) ондағы барлық нәрсеге қол жеткізе алады. Басқа да көптеген мысалдар бар, олардың ішінде жоғары лауазымды тұлғалардың құпия ақпараты бар USB дискілерін жоғалту, қоқыс тастамас бұрын дұрыс тазаланбаған өндірістік құпиялары бар ескі қатты дискілер және т. б.

Дегенмен, қауіпсіздікке қатысты ең маңызды оқиғалар күрделі кибершабуылдарға жатады. Бұл кітапта операциялық жүйелерге қатысты шабуылдарға ерекше назар аударылады. Басқаша айтқанда, біз веб-шабуылдарды немесе SQL дерекқорларына шабуылдарды қарастырмаймыз. Оның орнына, біз операциялық жүйелерге бағытталған немесе қауіпсіздік саясатын орындауда (немесе көбінесе орындауға кедергі келтіруде) маңызды рөл атқаратын шабуылдарға назар аударамыз. Жалпы, біз ақпаратты пассивті түрде ұрлауға тырысатын шабуылдар мен компьютерлік бағдарламаның әрекетін бұзуға тырысатын шабуылдарды ажыратамыз.

Пассивті шабуылдың мысалы-желі трафигіне қосылған және деректерді алу үшін шифрды бұзуға тырысатын жау. Белсенді шабуыл кезінде зиянкес зиянды кодты орындауға мәжбүрлей отырып, мысалы, кредит картасының деректемелерін ұрлау үшін пайдаланушының веб-браузерін басқаруды ала алады. Сол сияқты, біз хабарламаны немесе файлды толығымен кілтсіз бастапқы деректерді қалпына келтіруді қиындататын жолмен араластыруға арналған криптографияны және бағдарламаларға қорғаныс механизмін қосатын, бұзушыларға олардың мінез-құлқын өзгертуді қиындататын бағдарламаларды (hardening) нығайтуды ажыратамыз.

Операциялық жүйе көптеген жерлерде криптографияны қолданады: желі арқылы деректерді қауіпсіз беру, файлдарды дискіде қауіпсіз сақтау, пароль файлындағы құпия сөздерді шифрлау және т. б. Бағдарламаларды нығайту да кеңінен қолданылады: бұзушылардың жұмыс істеп тұрған бағдарламаға жаңа кодты енгізуіне жол бермеу үшін, әрбір процестің өзіне қажетті және ол үшін көзделген, жоғары емес артықшылықтардың болуын қамтамасыз ету үшін және т.б.

Қорғалған жүйелерді құруға бола ма? Бүгінгі таңда газет ашып, компьютерлік жүйелерге енген, ақпаратты ұрлаған немесе миллиондаған компьютерлерді бақылауға алған хакерлер туралы тағы бір оқиғаны оқымау мүмкін емес. Бұл жағдайда сіз екі қарапайым және өте қисынды сұрақ қоя аласыз:

1. Қорғалған компьютерлік жүйені құру мүмкін емес пе?
2. Олай болса, неге ол әлі жасалмаған?

Бірінші сұрақтың жауабы: "Теориялық түрде мүмкін". Негізінде, бағдарлама қателіктерден құтылуы мүмкін және егер бұл бағдарлама тым үлкен немесе күрделі болмаса, біз оның қауіпсіздігін қамтамасыз ете аламыз. Өкінішке орай, қазіргі компьютерлік жүйелер өте күрделі, бұл екінші сұрақтың жауабына сәйкес келеді.

Неліктен мүлдем қорғалған жүйелер әлі жасалмаған деген сұрақтың жауабы екі негізгі себепке байланысты. *Біріншіден,* бүгінгі жүйелер қауіпсіз емес, бірақ

пайдаланушылар олардан бас тартқысы келмейді. Мысалы, егер Microsoft Windows жүйесінен басқа, оның вирусқа қарсы иммунитеті бар, бірақ Windows қосымшаларын орындамайтын жаңа өнімі бар деп мәлімдесе, онда барлық жеке пайдаланушылар мен компаниялар бірден Windows-тан бас тартып, жаңа жүйені сатып алуы екіталай. Microsoft корпорациясының қорғалған операциялық жүйесі бар (Fandrich et al., 2006), бірақ ол оны нарыққа шығармайды. *Екінші себеп* соншалықты айқын емес. Қорғалған жүйені құрудың жалғыз белгілі тәсілі-оның қарапайымдылығын сақтау. Функционалдылық мүмкіндіктер -қауіпсіздіктің жауы болып табылады. Көптеген технологиялық компаниялардағы маркетинг бөлімінің батыл жігіттері (дұрыс па, жоқ па) пайдаланушылар көбірек функционалдылықты, одан да кең және сапалы мүмкіндіктерді қалайды деп болжайды.

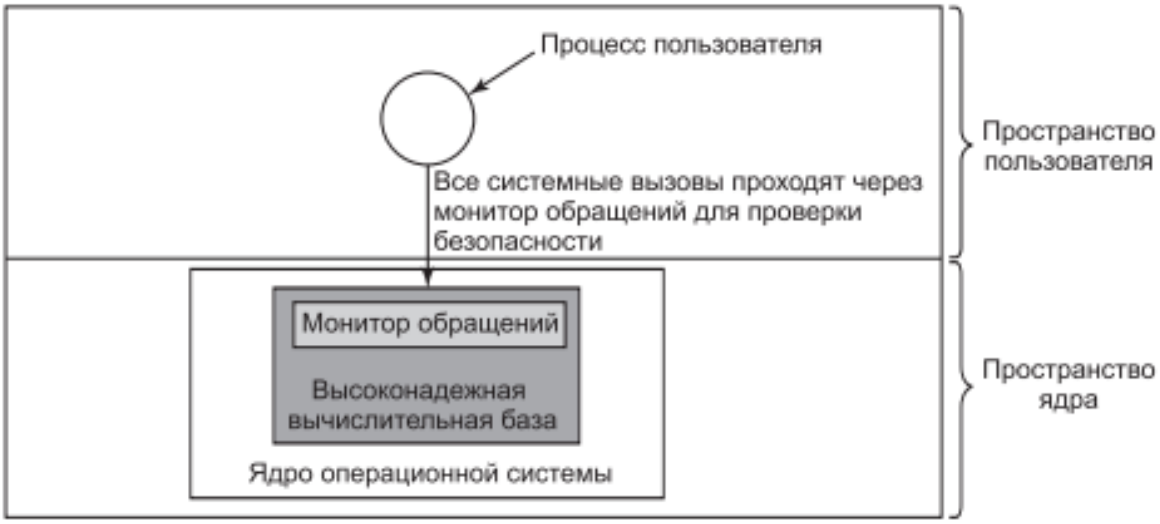
Олар өздерінің бағдарламалық өнімдерінің жүйелік құрылымдарын жасаушылар бұл үшін тиісті нұсқаулар алатынына сенімді. Бірақ мұның бәрі жүйенің күрделілігін, кодтың көбеюін, қауіпсіздік жүйесіндегі ақаулар мен қателіктердің көбеюін білдіреді. Екі қарапайым мысал келтіруге болады. Алғашқы электрондық пошта жүйелерінде хабарламалар ASCII мәтіні ретінде жіберілді. Олар қарапайым болды және оларды мүлдем, яғни толығымен қауіпсіз етуге болады. Пошта бағдарламаларында қателіктер болса да, ASCII хабары компьютерлік жүйені бұзуы мүмкін (кейінірек біз бірнеше ықтимал шабуылдарды талқылаймыз). Содан кейін хабарламалардың мүмкіндіктерін кеңейту және оларға макростар болуы мүмкін Word редакторының файлдары сияқты құжаттардың басқа түрлерін қосу идеясы пайда болды. Мұндай құжатты оқу компьютерде біреудің бағдарламасын іске қосуды білдіреді. Қолданылатын қауіпсіздік механизмдеріне қарамастан, компьютерде біреудің бағдарламасын іске қосу ASCII мәтінін қарауға қарағанда әлдеқайда қауіпті. Пайдаланушыларға пассивті құжаттардан белсенді бағдарламаларға электрондық пошта мазмұнын өзгерту мүмкіндігі қажет болды ма? Мүмкін емес, бірақ бұл өзгерістер біреуге керемет идея болып көрінді, ал ешкім қауіпсіздік үшін салдары туралы ойламады.

Екінші мысал веб-беттерге қатысты. Бүкіләлемдік ғаламторда пассивті HTML парақтары қолданылған кезде, қауіпсіздіктің маңызды мәселелерін ештеңе білдірмеді. Енді көптеген веб-беттерде пайдаланушы мазмұнды көру үшін іске қосатын бағдарламалар (апплеттер мен JavaScript сценарийлері) болған кезде, қауіпсіздік жүйесінде барлық жаңа олқылықтар пайда болады. Олардың біреуін ғана жою керек, оның орнында екіншісі пайда болады. Бүкіләлемдік ғаламтор толығымен статикалық сипатқа ие болған кезде, пайдаланушылар динамикалық толтыруға дауыс беру кезінде екі қолын көтерді ме? Қауіпсіздіктің жақсы жүйесі сәнді қоңыраулар мен ысқырықтардан гөрі маңызды деп санайтын бірқатар ұйымдар бар, бұл ең алдымен әскери адамдарға қатысты. Келесі бөлімдерде бір сөйлемге келтіруге болатын бірқатар тиісті мәселелер қарастырылады. Қауіпсіз жүйені құру үшін сіз операциялық жүйенің өзегіндегі қауіпсіздік моделін қолдануыңыз керек, оны әзірлеушілер оның мәнін түсінуге оңай болар еді және кез-келген қысымға төтеп бере алады, жаңа функционалдылықты қосудан аулақ бола алады.

Жоғары сенімді есептеу базасы компьютерлік қауіпсіздік мамандарының шеңберлерінде қорғалған жүйелер туралы емес, сенімді жүйелер туралы жиі айтылады. Бұл ресми түрде белгіленген қауіпсіздік талаптары бар және осы

талаптарды орындайтын жүйелер. Әрбір сенімді жүйенің негізінде барлық қауіпсіздік шараларын орындау үшін қажетті аппараттық және бағдарламалық қамтамасыз етуден тұратын TCB (Trusted Computing Basse) минималды базасы (сенімді есептеу базасы — жоғары сенімді есептеу базасы) орналасқан. Егер жоғары сенімді есептеу базасы техникалық шарттарға сәйкес жұмыс істесе, жүйенің қауіпсіздігі басқа қолайсыз жағдайларда бұзылмайды.

Әдетте, TCB негізінен аппараттық құралдардан тұрады (қауіпсіздікке әсер етпейтін кіріс құрылғыларынан басқа), операциялық жүйенің негізгі бөлігі және артықшылықты пайдаланушы өкілеттіктері бар көптеген немесе барлық пайдаланушы бағдарламалары (мысалы, UNIX жүйесінің түбірлік каталогында SETUID биті орнатылған бағдарламалар). TCB бөлігі болуы керек операциялық жүйенің функциялары мыналарды қамтиды: процесті құру, процестерді ауыстыру, жад дисплейін басқару, сонымен қатар файлдарды басқару және деректерді енгізу-шығару бөлігі. Сенімді жүйенің дизайнында TCB көбінесе оның мөлшерін азайту және оның дұрыстығын тексеру үшін операциялық жүйенің қалған бөлігінен толығымен бөлінеді. Жоғары сенімді есептеу базасының маңызды бөлігі – үндеу мониторы (1-сурет). Ол қауіпсіздікке қатысты барлық жүйелік қоңырауларды қабылдайды, мысалы, файлдарды ашуға қоңырау шалып, оларды өңдеу керек пе, жоқ па, соны шешеді. Осылайша, үндеу мониторы барлық қауіпсіздік шешімдерін бір жерде орналастыруға мүмкіндік береді, оларды айналып өтуге жол бермейді. Көптеген операциялық жүйелерді ұйымдастыру осы схемадан ерекшеленеді, бұл олардың сенімсіздігінің себептерінің бірі.



1-сурет. Үндеу мониторы

Кейбір заманауи қауіпсіздік зерттеулерінің мақсаттарының бірі-бірнеше миллион кодтық жолдан ондаған мыңға дейін жоғары сенімді есептеу базасын азайту. 1.22-сурет [1] MINIX 3 операциялық жүйесінің құрылымын көрсетеді, ол POSIX үйлесімді жүйе, бірақ Linux немесе FreeBSD-ге қарағанда мүлдем басқа құрылымы бар. MINIX 3 ядросында шамамен 10 000 жол бағдарламалық код жұмыс істейді. Қалған код пайдаланушы процестерінің жиынтығы ретінде іске қосылады. Ядро кодының бір бөлігі, мысалы, файлдық жүйе және процесс менеджері, өте сенімді есептеу базасына кіреді, өйткені бұл код жүйенің қауіпсіздігіне оңай нұқсан

келтіруі мүмкін. Бірақ қалған бөліктер, мысалы, принтер драйвері және дыбыстық карта драйвері, жоғары сенімді есептеу базасына кірмейді және олардың барлық сәтсіздіктері маңызды емес (олар вирустан туындаған болса да), олар жүйенің қауіпсіздігіне нұқсан келтіре алмайды. Жоғары сенімді есептеу базасын екі тапсырыс бойынша азайту арқылы MINIX 3 сияқты жүйелер дәстүрлі дизайн шешімдеріне қарағанда қауіпсіздіктің жоғары деңгейін қамтамасыз етуі мүмкін.

ОЖҚ. Дәріс №2. Ресурстарға қол жеткізуді басқару. Қорғау домендері. Қол жеткізуді басқару тізімдері. Мүмкіндіктер тізімі.

Қауіпсіздік ресурстарына қол жеткізуді басқару. Егер не қорғалуы керек, кімге және не істеуге рұқсат етілетіні туралы нақты модель болса, қауіпсіздікке қол жеткізу оңайырақ болып табылады. Бұл салада өте үлкен жұмыс жасалды, сондықтан осы қысқаша сипаттамада біз тек үстірт шолу жасайтын боламыз. Біз оларды қолданудың бірқатар жалпы модельдері мен тетіктеріне ғана назар аударамыз.

Қорғау домендері. Компьютерлік жүйеде қорғауды қажет ететін көптеген ресурстар немесе объектілер бар. Бұл объектілер құрал-жабдықтар (мысалы, орталық процессорлар, жад парақтары, диск жетектері немесе принтерлер) немесе бағдарламалық жасақтама (мысалы, үдерістер, файлдар, мәліметтер базасы немесе семафорлар) болуы мүмкін.

Әр объектінің оған қол жеткізуге болатын ерекше атауы және осы объектіге қатысты үдерістер орындай алатын операциялардың ақырлы жиынтығы болады. Файлға *read* және *write* операциялары, ал семафорға *up* және *down* операциялары тән. Оларға қол жеткізу құқығы жоқ объектілерге кіру үдерістеріне тыйым салу әдісі қажет екені анық. Сонымен қатар, бұл механизм қажет болған жағдайда рұқсат етілген операцияларды таңдау арқылы үдерістерді шектеуге мүмкіндік беруі керек.

Мысалы, *A* процесіне *F* файлынан деректерді оқу құқығы берілуі мүмкін, бірақ оған бұл файлға жазуға рұқсат етілмейді. Қорғаудың әртүрлі механизмдерін қарастыру үшін домен ұғымын енгізген пайдалы. **Домен** (*domain*) жұптар жиынынан (объект, қол жеткізу құқығы) тұрады. Әр жұп объектіні және сол объектіге қатысты орындалуы мүмкін операциялардың ішкі жиыны болып табылады. **Қол жеткізу құқығы** (*rights*) осы контексте белгілі бір операцияны орындауға берілетін рұқсат түрін анықтайды. Көбінесе домен жеке қолданушымен байланысты болады, бұл пайдаланушы не істей алатындығы немесе, керісінше, не істей алмайтындығы туралы хабарлайды, бірақ ол жеке пайдаланушыға ғана емес, жалпы сипатқа да ие болуы мүмкін. Мысалы, бір жобада жұмыс істейтін бір бағдарламашылар тобының қызметкерлері толығымен бір доменге тиесілі және жоба файлдарына қол жеткізе алады. Объектілерді домендер бойынша бөлу кімге және олардың не туралы білуі керек екеніне байланысты.

Дегенмен, іргелі ұғымдардың бірі – **ең төменгі билік принципі** (*Principle of Least Authority* (**POLA**)) немесе қажетті білім принципі. Жалпы, әр доменде олармен жұмыс істеу үшін минималды объектілер мен артықшылықтар болған кезде және артық ештеңе болмаған кезде қауіпсіздікті сақтау оңайырақ.

2-суретте әр объектіге қатысты әрқайсысында объектілері бар және оқу, жазу және орындау құқықтары бар үш домен көрсетілген (Read, Write, eXecute). *Printer1* объектісі бір уақытта екі доменде болатындығын және олардың әрқайсысында бірдей құқықтарға ие екенін ескеріңіз. *File1* объектісі екі доменде де бар, бірақ олардың әрқайсысында әртүрлі құқықтарға ие.

Кез-келген уақытта әр процесс қандай да бір қорғаныс доменінде жұмыс істейді. Басқаша айтқанда, оған қол жеткізуге болатын объектілердің белгілі бір жиынтығы болады және әр объект үшін белгілі бір құқықтар жиынтығы көзделген. Жұмыс

кезінде процесстер бір доменнен екіншісіне ауыса алады. Домендер арасында ауысу ережелері белгілі бір жүйеге байланысты.

2-сурет. Үш қорғау домені

Қорғау домені идеясын нақтылау үшін UNIX жүйесінен мысал келтіруге болады (Linux, FreeBSD және оларға ұқсас клондарға да қатысты). UNIX-те процестің домені оның UID және GID идентификаторларымен анықталады. Пайдаланушы кірген кезде оның қабығы пароль файлындағы жазбасында бар UID және GID алады және олар оның барлық ұрпақ үдерістеріне мұра болады. Кез-келген комбинацияны (UID, GID) ұсына отырып, сіз барлық объектілердың толық тізімін жасай аласыз (файлдар, соның ішінде арнайы файлдар түрінде ұсынылған енгізу-шығару құрылғылары және т. б. д.), оған қол жеткізудің мүмкін түрін (оқу, жазу, орындау) көрсете отырып, процесс жүгіне алады. Бірдей (UID, GID) комбинациясы бар екі процесс бірдей объектілер жиынтығына бірдей қол жеткізе алады. Дегенмен әр түрлі (UID, GID) мәндерге ие үдерістер әр түрлі файл жиындарына, бұл жиындардың айтарлықтай қабаттасуымен де, қол жеткізе алады.

Сонымен қатар, UNIX-тегі әр процесс екі бөліктен тұрады: тұтынушылық және жүйелік (ядрода орындалатын) режимдерде жұмыс істейтін. Процесс жүйелік шақыруды жүзеге асырған кезде, ол пайдаланушы бөлігінен жүйеге ауысады. Пайдаланушымен салыстырғанда, жүйелік бөлік басқа объектілер жиынтығына қол жеткізе алады. Мысалы, процестің жүйелік бөлігі физикалық жадтағы барлық беттерге, бүкіл дискіге және барлық басқа қорғалған ресурстарға қол жеткізе алады. Осылайша, жүйелік шақыру қорғаныс домендерін ауыстырудың себебі болып табылады.

Процесс *exec* жүйелік жүйелік шақыруды жүзеге асырған кезде, ол SETUID биті немесе SETGID биті орнатылған файлға қатысты, ол жаңа жарамды UID немесе GID идентификаторын алады. Басқа комбинациямен (UID, GID) ол қол жетімді файлдар мен операциялардың басқа жиынтығына ие болады. Орнатылған SETUID битімен немесе SETGID битімен бағдарламаны іске қосу доменнің ауысуына себеп болады, өйткені кіру құқықтары өзгереді. Жүйе объектінің белгілі бір объектіге тиесілігін қалай қадағалайтынын білу маңызды. Концептуалды түрде үлкен матрицаны елестетуге болады, ондағы жолдар домендер, ал бағандар объектілер болады. Әрбір ұяшықта, егер бар болса, домен объектіге қатысты иеленетін құқықтар тізімделеді. 3-суретте 2-сурет үшін құрылған матрица көрсетілген. Осы матрица мен ағымдағы домен нөмірінің көмегімен операциялық жүйе белгілі бір доменнен берілген объектіге кірудің белгілі бір түріне рұқсат етілгенін анықтай алады. Егер сіз *enter*

кіру операциясына рұқсат етілуі мүмкін доменнің өзін объект ретінде елестетсеңіз, домендер арасындағы ауысуды сол кестелік модельге оңай қосуға болады. 4-суретте қайтадан 3-суреттегі матрица көрсетілген, бірақ бұл жағдайда объектілер ретінде үш доменнің өзі пайда болады. *1-домендегі* үдерістер *2-доменге* ауыса алады, бірақ олар енді қайта орала алмайды.

Бұл жағдай UNIX-те орнатылған SETUID битімен бағдарламаның орындалуын модельдейді. Бұл мысалда басқа домендерді ауыстыруға рұқсат етілмейді.

Домен \ Объект	Файл 1	Файл 2	Файл 3	Файл 4	Файл 5	Файл 6	Принтер 1	Плоттер 2
1	Чтение	Чтение Запись						
2			Чтение	Чтение Запись Исполнение	Чтение Запись		Запись	
3						Чтение Запись Исполнение	Запись	Запись

3-сурет. Қорғаныс матрицасы

Домен \ Объект	Файл 1	Файл 2	Файл 3	Файл 4	Файл 5	Файл 6	Принтер 1	Плоттер 2	Домен 1	Домен 2	Домен 3
1	Чтение	Чтение Запись								Enter	
2			Чтение	Чтение Запись Исполнение	Чтение Запись		Запись				
3						Чтение Запись Исполнение	Запись	Запись			

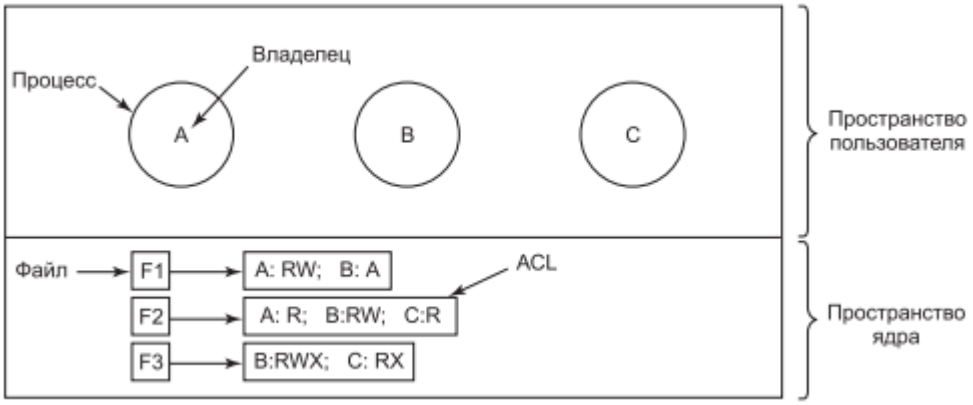
4-сурет. объектілер ретінде домендермен анықталған қорғаныс матрицасы.

Кіруді басқару тізімдері

Іс жүзінде 4-суретте көрсетілген матрица, ол үлкен мөлшерде және таусылған сипатқа байланысты сирек қолданылады. Көптеген домендерде көптеген объектілерге мүлдем қол жетімділік жоқ, сондықтан диск кеңістігі өте үлкен, іс жүзінде бос матрицаны сақтау үшін үлкен ысырап болады. Сондықтан келесі екі әдіс практикалық қолдану тапты: матрицаны жолдар немесе бағандар бойынша сақтау, содан соң тек толтырылған элементтерді сақтау. Бір таңқаларлығы, бұл екі тәсіл бір-бірінен ерекшеленеді. Бүгін (бұл бөлімде) ақпаратты бағандар бойынша, ал келесі бөлімде жолдар бойынша сақтау қарастырылады.

Бірінші әдісте әр объектімен белгілі бір объектіге кіруге рұқсат етілген барлық домендерді, сондай-ақ кіру түрін қамтитын реттелген тізім байланыстырылады. Мұндай тізім 5-суретте көрсетілген. Ол **ACL** (Access Control List – қол жеткізуді басқару тізімі) деп аталады. Мұнда *A, B* және *C* домендеріне жататын үш процесс, сонымен қатар үш файл көрсетілген: *F1, F2* және *F3*. Жағдайды жеңілдету үшін әр доменге тек бір қолданушы сәйкес келеді делік, бұл жағдайда *A, B* және *C* пайдаланушылары. Ақпараттық қауіпсіздік әдебиетінде тұтынушылар көбіне **субъектілер** (*subjects*), **принципиалдар** (*principals*) немесе осы ортада есептік жазбасы бар пайдаланушылар деп аталады (олардың иелерін белгілі бір жерлерден,

объектілерден **(objects**) ажырату үшін, мысалы, оларға файлдарды жатқызуға болады).



5-сурет. Файлдарға қол жеткізуді басқару үшін қол жеткізуді басқару тізімдерін пайдалану

Әр файлда онымен байланысты ACL бар. F1 файлының ACL тізімінде екі жазба бар (нүктелі үтірмен бөлінген). Бірінші жазбада A пайдаланушысының кез-келген процесі осы файлға қатысты оқу және жазу операцияларын жүргізе алады. Екінші жазбада B пайдаланушысының кез-келген процесі осы файлды оқи алады. Пайдаланушы деректеріне қол жеткізудің барлық басқа түрлеріне және басқа пайдаланушыларға қол жеткізудің барлық түрлеріне тыйым салынады. Құқықтар процеске емес, пайдаланушыға берілетініне назар аударыңыз. Қорғау жүйесінің жұмысының нәтижесінде A пайдаланушысының кез-келген процесі *F1* файлына қатысты оқу және жазу операцияларын орындай алады. Мұндай үдерістер қанша болуы маңызды емес. Процестің идентификаторы емес, процестің иесі кім екендігі маңызды.

F2 файлының ACL тізімінде үш жазба бар: *A, B* және *C* пайдаланушылары файлды оқи алады,ал B пайдаланушысы да оған жаза алады. Қол жеткізудің басқа түрлеріне рұқсат етілмейді. *F3* файлы орындалатын бағдарлама екені анық, өйткені *A* және *B* пайдаланушылары бұл файлды оқи да, орындай да алады. *B* пайдаланушысына оған жазба жүргізуге рұқсат етіледі. Бұл мысал ACL тізімдері арқылы қорғаудың жалпы формасын көрсетеді. Іс жүзінде неғұрлым күрделі жүйелер жиі қолданылады. Бастау үшін, мұнда тек үш кіру құқығы көрсетілген: оқу, жазу және орындау. Олардан басқа, басқа қол жетімділік құқықтары болуы мүмкін. Құқықтардың бір бөлігі жалпы сипатқа ие болуы мүмкін, яғни барлық объектілерге таралуы мүмкін, ал бір бөлігі объектіге нақты байланысты болуы мүмкін. Жалпы сипаттағы құқықтардың мысалдары **объектіні жою** (*destroy object*) және **объектіні көшіру** (*copy object*) құқығы болуы мүмкін, олар түріне қарамастан кез-келген объектіге тиесілі болуы мүмкін. Нысанға ерекше қатысы бар құқықтарға пошта жәшігі объектісіне **хабарлама қосу** құқығы (*append message*) және каталог объектісі үшін **алфавиттік ретпен сұрыптау** құқығы (*sort alphabetically*) кіреді.

Осы уақытқа дейін ACL тізіміндегі жазбалар жеке пайдаланушыларға тиесілі болды. Көптеген жүйелер пайдаланушылар **тобы** (**group**) тұжырымдамасын қолдайды. Топтардың атаулары бар және оларды ACL тізімдеріне де қосуға болады. Топтардың семантикасында екі мүмкін нұсқа бар. Кейбір жүйелерде әр процесте

UID пайдаланушысының идентификаторы және GID тобының идентификаторы бар. Мұндай жүйелерде ACL тізімдерінде көрініс жазбалары бар

UID1, GID1: Құқық 1; UID2, GID2: Құқық 2;...

Нысанға кіру туралы сұрау түскен жағдайда, осы сұрауды берген адамның UID және GID тексерісі жасалады. Егер бұл идентификаторлар ACL тізімінде болса, онда тізімде көрсетілген құқықтар беріледі. Егер комбинациялар(UID, GID) тізімде болмаса, кіруге рұқсат етілмейді.

Негізінде, топтарды қолдану **рөл** (*role*) ұғымын енгізеді. Tana жүйелік әкімші болып табылатын есептеу орталығын қарастырсақ, ол *sysadm* тобына кіреді. Бірақ компанияда қызметкерлерге арналған клубтар бар делік және Тана–көгершін әуесқойлары клубының мүшесі болсын. Клуб мүшелері *pigfan* тобына жатады және көгершін деректер базасын жүргізу үшін компанияның компьютерлеріне қол жеткізе алады. ACL тізімінің бөлігі 2-кестеде көрсетілген көрініске ие болуы мүмкін.

2-кесте. Қол жеткізуді басқарудың екі тізімі

Файл	**Қолжетімділікті басқару тізімі**
Password	tana, sysadm: RW
Pigeon data	bill, pigfan: RW; tana, pigfan: RW;

Егер Тана осы файлдардың біріне кіруге тырысса, нәтиже оның қай топқа кіргеніне байланысты болады. Тіркеу кезінде жүйе өзінің қай тобын пайдаланғысы келетінін немесе топтарға кіруді бөлек сақтау үшін әр түрлі атаулар және/ немесе парольдер болуы мүмкін екенін сұрауы мүмкін. Бұл схеманың мәні-Тана өзінің Көгершін хоббиімен айналысқан кезде пароль файлына қол жеткізе алмауы. Ол жүйеде жүйелік әкімші ретінде тіркелген жағдайда ғана пароль файлына қол жеткізе алады.

Кейбір жағдайларда пайдаланушыға белгілі бір файлдарға қол жетімділік берілуі мүмкін, ол қазіргі уақытта топқа жатады. Мұны кез-келген нәрсені білдіретін **топтық таңбаны** (*wildcard*) пайдалану арқылы ұйымдастыруға болады. Мысалы,

tana,* :RW

жазбасы пароль файлындағы Таняға қазіргі уақытта қай топқа жататынына қарамастан қол жеткізуге мүмкіндік береді.

Тағы бір мүмкіндік – кез-келген топқа кіретін және белгілі бір қол жетімділік құқығы бар пайдаланушы осы құқықтарды алады. Артықшылығы – бір уақытта бірнеше топқа кіретін пайдаланушы тіркелу кезінде қай топты пайдалану керектігін көрсетпеуі керек. Олардың барлығы оның бүкіл жұмысы барысында ескеріледі. Бұл тәсілдің кемшілігі–бұл инкапсуляцияның аз дәрежесін қамтамасыз етеді.

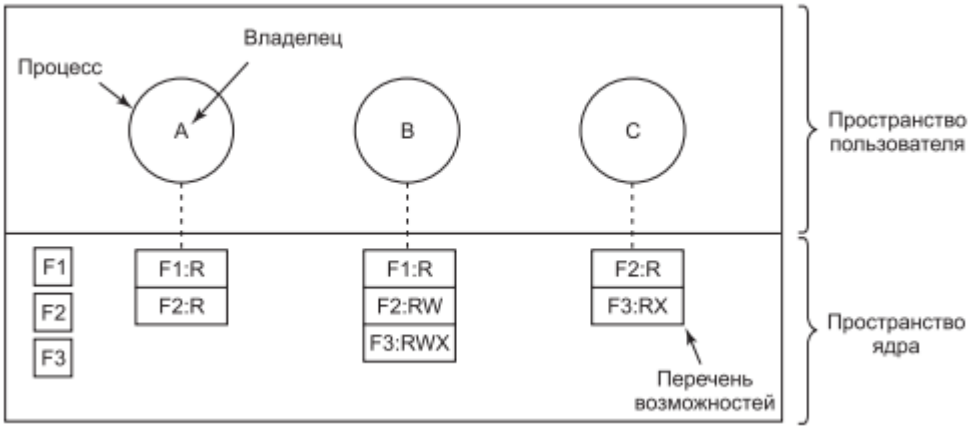
Көгершін клубының жиналысы кезінде пароль файлын өңдей алады. Топтар мен топтық таңбаларды пайдалану белгілі бір пайдаланушының файлға кіруін іріктеп бұғаттауға мүмкіндік береді. Мысалы,

virgil, *: (none); *, *: RW

жазбасы оқуға арналған файлды оқу және жазу құқын немесе мүмкіндігін Вирджил (Virgil) атты тұтынушыдан басқа барлық пайдаланушыға береді. Бұл жазба рет-

ретімен жазылғандықтан жұмыс істейді және олардан бірінші сәйкес келеді, ал кейінгі жазбалар тіпті талданбайды.

5-суретте көрсетілген матрицаны, сондай-ақ жолдармен жазуға болады. Бұл әдісті қолданған кезде әр процеске қол жеткізуге болатын объектілердің тізімі, сондай-ақ әр объектімен қандай операцияларға рұқсат етілгені туралы ақпарат, басқаша айтқанда, оның қорғау домені байланысты болады. Мұндай тізім **мүмкіндіктер тізімі** (*capability list, C-list*) деп аталады, ал оның элементтері – **мүмкіндіктер** (*capabilities*) (Dennis and Van Horn, 1966; Fabry, 1974).
6-суретте үш процестің жиынтығы және олардың мүмкіндіктер тізімі көрсетілген.



6-сурет. Мүмкіндіктерді пайдалану кезінде әр процесс олардың тізімін алады

Мүмкіндіктер тізімінің әр элементі иесіне белгілі бір объектіге қатысты белгілі бір құқықтар береді. Мысалы, 6-суретте *A* пайдаланушысы иелік ететін процесс *F1* және *F2* файлдарын оқи алады. Әдетте, мүмкіндіктер тізімінің элементі файл идентификаторынан (немесе жалпы жағдайда объекттен) және әртүрлі құқықтарға арналған бит массивінен тұрады. UNIX отбасылық (семейства) операциялық жүйелерінде файл идентификаторы ретінде оның ***i***-ші түйінінің нөмері пайдаланылуы мүмкін. Мүмкіндіктер тізімі өздері объектілер болып табылады және оларды басқа мүмкіндіктер тізімдері көрсете алады және, осылайша, қосалқы домендерді (субдомендерді) пайдалануды жеңілдетеді.

ОЖҚ. Дәріс №3. Операциялық жүйелердің қауіпсіздігі. Қорғалған жүйелерді құруға бола ма? Жоғары сенімді есептеу базасы.

Операциялық жүйелердің қауіпсіздігі. Компьютерлік жүйелер қауіпсіздігінің беделін түсірудің көптеген жолдары бар. Мысалы, адамдардың көпшілігі өздерінің PIN-кодтары үшін 0000 комбинациясын орнатады немесе пароль ретінде "password" сөзін қолданады – бұларды есте сақтау оңай, бірақ олар қауіпсіздіктің жоғары деңгейін қамтамасыз етпейді. Керісінше, кейбір адамдар есте сақтау мүмкін емес өте күрделі парольдерді де ойлап табады және оларды мониторға немесе пернетақтаға орналастырылған жапсырмаларға жазуға мәжбүр болады. Машинаға физикалық қол жетімділігі бар кез-келген адам (тазалаушылар, хатшылар және барлық келушілерді қоса есептегенде) ондағы *барлық* нәрсеге қол жеткізе алады. Басқа да көптеген мысалдар бар, олардың ішінде жоғары лауазымды тұлғалардың құпия ақпараты бар USB дискілерін жоғалтуы, өндірістік құпиялары бар, бірақ қоқысқа тастамас бұрын дұрыс тазаланбаған ескі қатты дискілер және т.б.

Дегенмен, қауіпсіздікке қатысты ең маңызды оқиғалар күрделі кибершабуылдарға жатады. Кітапта операциялық жүйелерге қатысты шабуылдарға ерекше назар аударылады. Басқаша айтқанда, біз веб-шабуылдарды немесе SQL дерекқорларына жасалатын шабуылдарды қарастырмаймыз. Оның орнына, біз операциялық жүйелерге бағытталған немесе қауіпсіздік саясатын орындауда (немесе көбінесе орындауға кедергі келтіруде) маңызды рөл атқаратын шабуылдарға назар аударамыз.

Жалпы, біз ақпаратты пассивті түрде ұрлауға тырысатын шабуылдар мен компьютерлік программаның әрекетін бұзуға тырысатын шабуылдарды ажыратып қарастырамыз. Пассивті шабуылдың мысалы – желі трафигіне қосылған және деректерді алу үшін шифрды бұзуға (егер ол бар болса) тырысатын жау. Белсенді шабуыл кезінде қаскүнем зиянды кодты орындауға мәжбүрлей отырып, мысалы, кредит картасының деректемелерін ұрлау үшін, пайдаланушының веб-браузерін басқаруды өзіне ала алады. Сол сияқты, біз толығымен хабарламаны немесе файлды кілтсіз бастапқы деректерді қалпына келтіруді қиындататын жолмен араластыруға арналған **криптографияны** және бағдарламаларға қорғаныс механизмін қосатын, бұзып кірушілерге олардың іс-әрекеттерін өзгертуді қиындататын **программаларды нығайтуды** (*укрепление,* ***hardening***) ажыратамыз.

Операциялық жүйе криптографияны көптеген жерлерде қолданады: желі арқылы деректерді қауіпсіз жіберу, файлдарды дискіде қауіпсіз сақтау, пароль файлындағы құпия сөздерді шифрлау және т. б. Программаларды нығайту да кеңінен қолданылады: бұзып кірушілердің жұмыс істеп тұрған программаға жаңа кодты енгізуіне жол бермеу үшін, әрбір процестің өзіне қажетті және ол үшін көзделген, және одан жоғары емес артықшылықтардың болуын қамтамасыз ету үшін және т.б.

Қорғалған жүйелерді құруға бола ма? Бүгінгі таңда газет ашып, компьютерлік жүйелерге енген, ақпаратты ұрлаған немесе миллиондаған компьютерлерді бақылауға алған хакерлер туралы тағы бір оқиғаны оқымай қалу мүмкін емес. Бұл жағдайда сіз екі қарапайым және өте қисынды сұрақ қоя аласыз:

3. Қорғалған компьютерлік жүйені құруға болмай ма?
4. Егер олай болса, неге ол әлі жасалмаған?

Бірінші сұрақтың жауабы: "Теориялық түрде мүмкін". Негізінде, программа қателіктерден құтылуы мүмкін және егер бұл программа тым үлкен немесе күрделі болмаса, біз оның қорғалғанына көз жеткізе аламыз. Өкінішке орай, қазіргі компьютерлік жүйелер өте күрделі, бұл екінші сұрақтың жауабына сәйкес келеді.

Неліктен мүлдем қорғалған жүйелер әлі жасалмаған деген сұрақтың жауабы екі негізгі себепке байланысты. *Біріншіден,* бүгінгі жүйелер қорғалмаған, бірақ пайдаланушылардың олардан бас тартқысы келмейді. Мысалы, егер Microsoft Windows жүйесінен басқа, оның вирусқа қарсы иммунитеті бар, бірақ Windows қосымшаларын орындамайтын жаңа өнімі SecureOC бар деп мәлімдесе, онда барлық жеке пайдаланушылар мен компаниялар бірден Windows-тан бас тартып, жаңа жүйені сатып алуы екіталай. Microsoft корпорациясының қорғалған операциялық жүйесі бар (Fandrich et al., 2006), бірақ ол оны нарыққа шығармайды. *Екінші себеп* соншалықты айқын емес. Қорғалған жүйені құрудың жалғыз белгілі тәсілі – оның қарапайымдылығын сақтап қалу. Функционалдылық мүмкіндіктер – қауіпсіздіктің жауы болып табылады. Көптеген технологиялық компаниялардағы маркетинг бөлімінің батыл жігіттері (дұрыс па, жоқ па) пайдаланушылар көбірек, одан да кеңірек, сапалырақ функционалдық мүмкіндіктерді қалайды деп болжайды. Олар өздерінің бағдарламалық өнімдерінің жүйелік құрылымдарын жасаушылар бұл үшін тиісті нұсқаулар алатынына сенімді. Бірақ мұның бәрі жүйенің күрделілігінің арта түсуін, кодтың көбірек санын, қауіпсіздік жүйесіндегі ақаулар мен қателіктердің көбеюін білдіреді.

Екі қарапайым мысал келтіруге болады. Алғашқы электрондық пошта жүйелерінде хабарламалар ASCII-мәтіні ретінде жіберілді. Олар қарапайым болды және оларды мүлдем, яғни толығымен қауіпсіз етуге болатын еді. Пошта бағдарламаларында қателіктер болса да, ASCII-де жазылған хабарламаның компьютерлік жүйені бұзуы күмәнді шаруа болатын (кейінірек біз бірнеше ықтимал шабуылдарды талқылаймыз). Содан кейін хабарламалардың мүмкіндіктерін кеңейту және оларға макростар болуы мүмкін Word редакторының файлдары сияқты құжаттардың басқа түрлерін қосу идеясы пайда болды. Мұндай құжатты оқу сіздің компьютеріңізде біреудің бағдарламасын іске қосуды білдіреді. Қолданылатын қауіпсіздік механизмдеріне қарамастан, компьютерде біреудің бағдарламасын іске қосу ASCII мәтінін қарауға қарағанда әлдеқайда қауіпті. Пайдаланушыларға пассивті құжаттардан белсенді бағдарламаларға электрондық пошта мазмұнын өзгерту мүмкіндігі қажет болды ма? Мүмкін емес, бірақ бұл өзгерістер біреуге керемет идея болып көрінді, ал ешкім қауіпсіздік үшін салдары туралы ойламады.

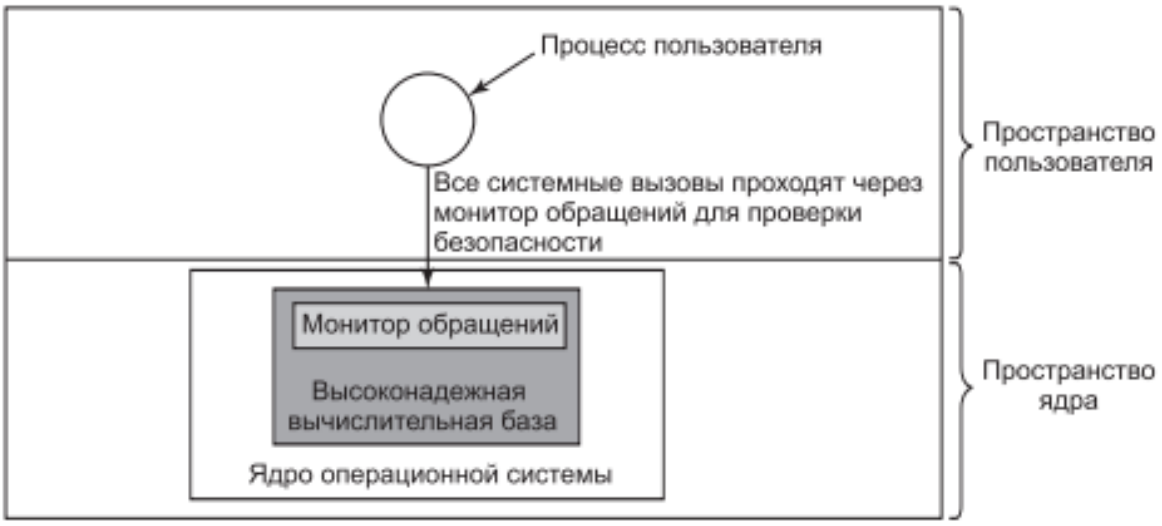
Екінші мысал веб-беттерге қатысты. Бүкіләлемдік ғаламторда пассивті HTML парақтары қолданылған кезде, қауіпсіздікке қатысты маңызды мәселелер туындайтыны ештеңе білдірмейді. Енді көптеген веб-беттерде пайдаланушы мазмұнды көру үшін іске қосатын бағдарламалар (апплеттер мен JavaScript сценарийлері) болған кезде, қауіпсіздік жүйесінде барлық жаңа олқылықтар пайда болады. Олардың біреуін жойғанда, екіншісі пайда болып жатты. Бүкіләлемдік ғаламтор толығымен статикалық сипатқа ие болған кезде, пайдаланушылар екі қолын көтеріп динамикалық толтыруға дауыс берген жағдай болды ма? Қауіпсіздіктің жақсы жүйесі сәнді қоңыраулар мен ысқырықтардан гөрі маңызды деп санайтын бірқатар ұйымдар бар, бұл ең алдымен әскери адамдарға қатысты.

Келесі бөлімдерде бәрін бір сөйлемге келтіруге болатын бірқатар тиісті мәселелер қарастырылады. Қорғалған жүйе құру үшін операциялық жүйенің ядросындағы, оның мәнін әзірлеушілердің түсінуіне оңай болатын қрапайым және кез-келген қысымға төтеп бере алатын, жаңа функционалдылық мүмкіндіктерді қосудан ада ететін қауіпсіздік моделін қолдану керек.

Жоғары сенімді есептеу базасы. Компьютерлік қауіпсіздік мамандарының ортасында қорғалған жүйелер туралы емес, сенімді жүйелер (**trusted systems**) жайында жиі айтылады. Бұл ресми түрде рәсімделген қауіпсіздік талаптары бар және осы талаптарды орындайтын жүйелер. Әрбір сенімді жүйенің негізінде барлық қауіпсіздік шараларын зорлап орындау үшін қажетті аппараттық және бағдарламалық қамтамасыз етуден тұратын жоғары сенімді есептеу базасы (высоконадежная вычислительная база – **TCB (Trusted Computing Basse)**) минималды базасы орналасқан. Егер жоғары сенімді есептеу базасы техникалық шарттарға сәйкес жұмыс істесе, жүйенің қауіпсіздігі басқа қолайсыз жағдайлардың бәрінде де бұзылмайды.

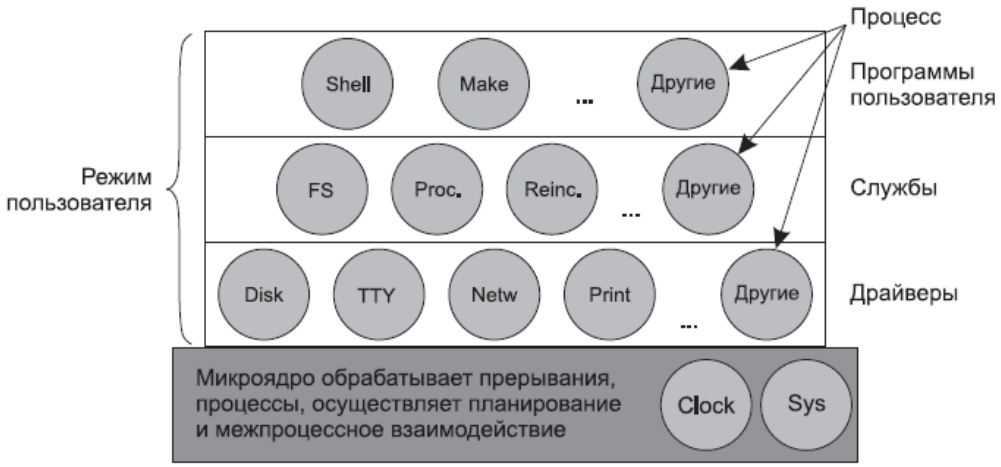
Әдетте, TCB негізінен аппараттық құралдардан (қауіпсіздікке әсер етпейтін енгізу-шығару құрылғыларынан басқа) тұрады, операциялық жүйенің ядросы бөлігінен және артықшылықтары бар пайдаланушы өкілеттіктері бар көптеген немесе барлық тұтынушы (мысалы, UNIX жүйесінің түбірлік каталогындағы SETUID биті орнатылған) программаларынан тұрады. Операциялық жүйенің TCB-ның бөлігі болуға тиіс функцияларына мыналар: үдерісті (процесті) құру, үдерістерді ауыстыру, жадты кескіндеуді басқару және, сонымен қатар, файлдар мен деректерді енгізу-шығаруды басқарудың белгілі бір бөлігі жатады. TCB көбінесе сенімді жүйенің конструкциясында, оның мөлшерін азайту және оның дұрыстығын тексеру үшін, операциялық жүйенің қалған құрылымынан толықтай бөлек ұйымдастырылады.

Жоғары сенімді есептеу базасының маңызды бөлігін қатынасу мониторы құрайды (1-сурет). Ол қауіпсіздікке қатысты барлық жүйелік шақыруларды (үзілімдерді) қабылдайды, мысалы, файлдарды ашуға жасалған жүйелік шақыруға қатысты, оларды өңдеу керек пе, жоқ па, соны шешеді. Осылайша, қатынасу мониторы барлық қауіпсіздік шешімдерін, оларды айналып өтуге жол бермей, бір жерде орналастыруға мүмкіндік береді. Көптеген операциялық жүйелерді ұйымдастыру осы схемадан басқаша боғандықтан, олардың сенімсіздігі себептерінің бірі болып табылады.



1-сурет. Қатынасу мониторы

Қауіпсіздік саласындағы кейбір заманауи зерттеулер мақсаттарының бірі – жоғары сенімді есептеу базасының көлемін бірнеше миллион жолдық кодтан ондаған мыңға дейін азайту. 2-суретте POSIX-үйлесімді жүйе, бірақ Linux немесе FreeBSD-ге қарағанда мүлдем басқа құрылымы бар, MINIX 3 операциялық жүйесінің құрылымы көрсетілген. MINIX 3 ядросында шамамен 10 000 жол бағдарламалық код жұмыс істейді. Қалған код пайдаланушы үдерістерінің жиынтығы ретінде іске қосылады. Ядро кодының бір бөлігі, мысалы, файлдық жүйе және үдеріс менеджері, жоғары сенімді есептеу базасына кіреді, өйткені бұл код жүйенің қауіпсіздігіне оңай нұқсан келтіруі мүмкін. Бірақ қалған бөліктер, мысалы, принтер драйвері және дыбыстық карта драйвері, жоғары сенімді есептеу базасына кірмейді және олардың барлық сәтсіздіктері маңызды емес (олар вирустан туындаған болса да), олар жүйенің қауіпсіздігіне ешқандай нұқсан келтіре алмайды. Жоғары сенімді есептеу базасын екі дәрежеге азайту арқылы MINIX 3 сияқты жүйелер дәстүрлі дизайн шешімдеріне қарағанда қауіпсіздіктің жоғары деңгейін қамтамасыз етуі мүмкін.



2-сурет. MINIX 3 жүйесінің жеңілдетілген құрылымы ([1], 1.22-сурет)

ОЖҚ. Дәріс №4. Криптография негіздері. Құпия кілтпен шифрлау. Ашық кілтпен шифрлау. Бір жақты функциялар. Сандық қолтаңбалар. Криптографиялық процессорлар

Криптография қауіпсіздікті қамтамасыз етуде өте маңызды рөл атқарады. Көптеген адамдар газет криптограммаларымен-белгілі бір жүйеде әр әріп басқа әріппен алмастырылатын кішкентай жұмбақтармен таныс. Қазіргі криптографияға олардың хот-догтар сияқты керемет тамақ дайындаумен бірдей байланысы бар. Бұл бөлімде компьютерлік дәуірдің криптографиясына қысқаша шолу жасалады. Жоғарыда айтылғандай, криптография көптеген жерлерде операциялық жүйелерде қолданылады. Мысалы, кейбір файлдық жүйелер дискідегі барлық деректерді шифрлай алады, IPSec сияқты протоколдар барлық желілік пакеттерді шифрлауға және/немесе қол қоюға мүмкіндік береді, ал көптеген операциялық жүйелер бұзушыларға оларды қалпына келтіруге мүмкіндік бермеу үшін құпия сөздерді шифрлайды. Сонымен қатар, 9.6 — бөлімде шифрлаудың қауіпсіздіктің басқа маңызды аспектісіндегі рөлі-аутентификация қарастырылады. Біз осы жүйелер қолданатын негізгі элементтерді қарастырамыз. Бірақ криптография мәселелерін байыпты қарау бұл кітаптың міндеттеріне кірмейді. Компьютерлік қауіпсіздік туралы көптеген керемет кітаптар осы тақырыпты егжей-тегжейлі қарастыруға арналған. Қызығушылық танытқандарға, мысалы, Kaufman et al. (2002), Gollman (2011) кітаптарын ұсынуға болады. Әрі қарай, олармен ешқашан кездеспеген оқырмандар үшін криптография туралы өте қысқаша шолу жасалады.

Криптографияның мақсаты — ашық мәтінді (plaintext) кодтау-хабарламаны немесе файлды оны шифрланған мәтінге (ciphertext) айналдыру. Оны қайтадан ашық мәтінге айналдыру туралы тек оған құқығы бар адамдар біледі. Барлық адамдар үшін шифрланған мәтін тек түсініксіз биттер жиынтығы болады. Жаңадан бастаушылар үшін бұл таңқаларлық, бірақ шифрлау үшін қолданылатын алгоритмдер (функциялар) әрқашан ашық болуы керек. Оларды құпия сақтауға тырысу ешқашан жұмыс істемейді және құпияны сақтауға тырысатын адамдарда жалған қауіпсіздік сезімін тудырады. Коммерцияда бұл тактика белгісіздікке байланысты қауіпсіздік деп аталады (security by obscurity) және оны тек әуесқойлар қолданады. Бір қызығы, көптеген трансұлттық корпорациялар осы санатқа жатады, олардың қызметкерлері бұл мәселені жақсы білуі керек. Іске нақты көзқараспен қауіпсіздік кілт деп аталатын алгоритмдердің параметрлеріне байланысты болады.

Егер P — кәдімгі мәтіні бар файл болса, KE — шифрлау кілті, C — шифрланған мәтін және E — шифрлау алгоритмі (яғни функция), C = E(P, KE). Бұл шифрлау анықтамасы. Бұдан шығатыны, шифрланған мәтін белгілі E шифрлау алгоритмін параметрлермен қолдану арқылы алынады, оның ішінде ашық мәтін P және құпия шифрлау кілті, KE болады. Ашық алгоритмді және құпиялылық мазмұнын тек кілттерде қолдануды қамтитын идея Керкгофф принципі деп аталады (Kerckhoffs' Principle). Оны XIX ғасырдағы голландиялық криптограф Август Керкгофс тұжырымдады. Бүгінгі таңда барлық маңызды криптографтар бұл идеяны ұстанады. Алдыңғы формулаға ұқсас, P = D(C, KD), мұндағы D — шифрлау алгоритмі, ал KD — шифрлау кілті. Осы формулаға сәйкес, C Шифр мәтінінен кәдімгі p мәтінін алу үшін, егер KD шифрлау кілті болса, d алгоритмін C және KD параметрлері ретінде

іске қосу керек. Әр түрлі компоненттер арасындағы қатынастар 13-суретте көрсетілген.

13-сурет. Ашық және шифрланған мәтін арасындағы қатынастар

Құпия кілтпен шифрлау

Жоғарыда айтылғандарды нақтылау үшін шифрлау алгоритмін қарастырыңыз, онда әр әріп басқа әріппен ауыстырылады, мысалы, барлық а әріптері Q әріптерімен, барлық В әріптері W әріптерімен, барлық С әріптері Е әріптерімен және т. б. осы мысалда көрсетілгендей ауыстырылады:

ашық мәтін: ABCDEFGHIJKLMNOPQRSTUVWXYZ

шифрланған мәтін: QWERTYUIOPASDFGHJKLZXCVBNM

Мұндай жалпы схема моноалфавиттік алмастыру (monoalphabetic substitution) деп аталады, мұнда кілт-толық алфавиттік жиынтыққа сәйкес келетін 26 әріптен тұратын жол. Бұл мысалда qwertyuiopasdfghjklzxcvbnm шифрлау кілті бар. Осы кілттің көмегімен ATTACK ашық мәтіні QZZQEA шифрланған мәтініне айналады. Шифрлау кілті шифрланған мәтіннен ашық мәтінді қалай алуға болатындығы туралы хабарлайды. Бұл мысалда kxvmcnophqrszyijadlegwbuft шифрлау кілті бар, өйткені шифрланған мәтіндегі а ашық мәтіндегі K, B — X және т. б. сәйкес келеді. Бір қарағанда, бұл өте қауіпсіз жүйе сияқты көрінуі мүмкін, өйткені криптографтар жалпы схеманы (әріптік ауыстыру) білетініне қарамастан, олар 26-ның қайсысы екенін білмейді! ≈ 4 * 1026 мүмкін кілттер қолданылады. Дегенмен, таңқаларлық аз мөлшерде Шифр мәтінін алғаннан кейін, олар шифрды оңай бұза алады. Әдетте аутопсия ұлттық тілдердің статикалық қасиеттерін ескере отырып жүзеге асырылады. Мысалы, ағылшын тілінде е әрпі жиі кездеседі, одан кейін t, o, a, n, i және т. б. Биграммалар деп аталатын ең көп таралған екі әріптен тұратын комбинациялар-th, in, er, re және т.б. бұл ақпаратты пайдалану кезінде шифрды бұзу қиын емес. Осы сияқты көптеген криптографиялық жүйелер шифрлау кілті болған кезде шифрлау кілтін оңай анықтауға мүмкіндік беретін қасиетке ие. Мұндай жүйелер құпия кілтпен шифрлау (құпия-кілт криптографиясы) немесе симметриялы кілтпен шифрлау (symmetric-key cryptography) деп аталады. Моноалфавитті алмастырумен шифрлаудың барлық абсолютті жарамсыздығымен, симметриялы кілті бар басқа Алгоритмдер белгілі, олар жеткілікті ұзындықта салыстырмалы түрде жоғары қарсылыққа ие. Нақты қауіпсіздікті қамтамасыз ету үшін 2256 ≈ 1,2 · 1077-ге тең кілттерді санау кеңістігін қамтамасыз ететін кем дегенде 256 биттік кілт қолданылуы керек. Ұзын кілттер әуесқойларға қарсы тұра алады, бірақ мемлекеттік құрылым мамандарының күш-жігеріне қарсы емес.

Ашық кілтпен шифрлау

Құпия кілт жүйелерінің тиімділігі хабарламаны шифрлау үшін қажетті есептеулердің қолайлы көлеміне байланысты, бірақ олардың кемшілігі бар: жіберуші де, алушы да ортақ құпия кілтке ие болуы керек. Кілтті беру үшін оларға физикалық байланыс қажет болуы мүмкін. Бұл мәселені шешу үшін ашық Кілттерді шифрлау қолданылады (қоғамдық-кілт криптографиясы) (Диффи және Хеллман, 1976). Бұл жүйелер шифрлау және шифрлау үшін әртүрлі кілттерді қолданатын қасиетке ие және жақсы таңдалған шифрлау кілті болған кезде тиісті шифрлау кілтін ашу мүмкін емес. Мұндай жағдайларда шифрлау кілтін ашық етіп жасауға болады, тек шифрлау кілтін құпия ұстауға болады. Ашық кілтпен шифрлау туралы алғашқы идеяны алу үшін келесі екі сұрақты қарастырыңыз:

1. 314159265358979 * 314159265358979 қанша болады? 2. 391257150641938709059482850824 1 санының квадрат түбірі қандай? Алтыншы сынып оқушыларының көпшілігі, егер сіз оларға қағаз бен қарындаш беріп, дұрыс жауап беру үшін сироп қосылған үлкен пломбир уәде етсеңіз, олар бірінші сұраққа бір-екі сағат ішінде жауап бере алады. Көптеген ересектер қарындаш, қағаз және өмір бойы 50% салық жеңілдіктері туралы уәде алып, екінші мәселені калькуляторсыз, компьютерсіз немесе басқа көмексіз шеше алмайды. Квадрат түбірін салу және алу бір-біріне қатысты кері операциялар болғанымен, олар есептеу қиындықтарында айтарлықтай ерекшеленеді. Асимметрияның ұқсас формалары ашық Кілттерді шифрлау үшін негіз болады. Шифрлау кезінде қарапайым операция қолданылады, бірақ кілтсіз шифрлау сізге көп уақытты қажет ететін операцияны қажет етеді. Ашық Кілттерді шифрлау жүйесінде RSA үлкен сандарды көбейту оларды көбейткіштерге бөлуден гөрі оңайырақ болады, әсіресе модульдік арифметика қолданылған кезде және қолданылған барлық Үлкен сандар жүздеген сандардан тұрады (Rivest et al., 1978). Криптографиялық әлемде бұл жүйе кеңінен қолданылады. Дискретті логарифмдерге негізделген жүйелер де қолданылады (El Gamal, 1985). Ашық Кілттерді шифрлау жүйелерінің басты мәселесі-олар симметриялы шифрлау жүйелеріне қарағанда мың есе баяу жұмыс істейді. Ашық кілттерді шифрлаудың жұмыс әдісі - барлығы жұпты алады (ашық кілт, жеке кілт) және ашық кілт жарияланады. Ашық кілт — шифрлау кілті, ал жабық кілт-шифрлау кілті. Әдетте, кілттер автоматты түрде жасалады, мүмкін пайдаланушы таңдаған парольді алгоритмге берілетін бастапқы Сан ретінде қолданады. Пайдаланушыға құпия хабарлама жіберу үшін корреспондент осы хабарламаның мәтінін алушының ашық кілтімен шифрлайды. Тек алушының жеке кілті болғандықтан, ол тек хабарламаны шеше алады.

Бір жақты функциялар

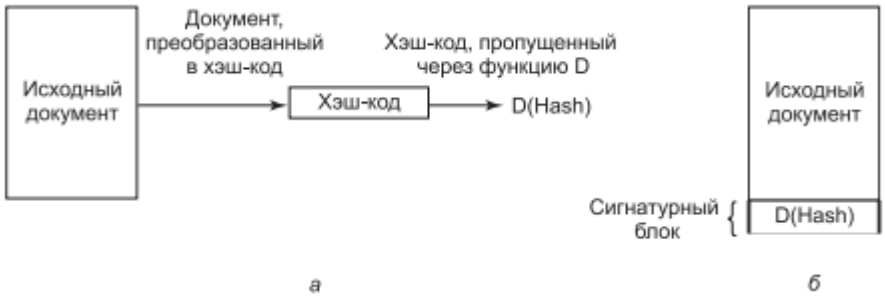
Бұдан әрі қарастырылатын көптеген жағдайлар бар, оларда F функциясы бар, ол берілген f және оның x параметрінде y = f(x) оңай есептеуге мүмкіндік береді, бірақ есептеулер арқылы тек f(x) берілген кезде x мәнін табуға мүмкіндік бермейді. Мұндай функция, әдетте, биттердің тізбегін күрделі түрде бұрмалайды. Бастапқыда ол y мәнін x мәніне тағайындай алады. Содан кейін ол циклді қолдана алады, ол x-де қанша рет болса, сонша рет орындалады, әр өту кезінде y биттері өту нөміріне байланысты белгілі бір жолмен реттеледі. Сонымен қатар, әр өту кезінде әр түрлі

тұрақтылар қосылады және тұтастай алғанда биттер толығымен араласады. Мұндай функциялар криптографиялық хэш функциялары деп аталады (cryptographic hash функциясы).

Сандық қолтаңбалар

Құжаттың цифрлық қолтаңбасының қажеттілігі жиі туындайды. Мысалы, банкке электронды пошта арқылы оған акцияларды сатып алуға тапсырма беретін банк клиентін елестетіп көріңіз. Тапсырманы жібергеннен және орындағаннан кейін бір сағаттан кейін акциялар құлады. Енді клиент тапсырманы электрондық пошта арқылы жібергенін жоққа шығарады. Әрине, банк электронды тапсырманы ұсына алады, бірақ клиент комиссия алу үшін банк оны жалған деп мәлімдей алады. Сот олардың қайсысы шындықты айтып жатқанын қалай біледі?

Цифрлық қолтаңбалар электрондық хабарламаларға және басқа да цифрлық құжаттарға кейінірек жіберуші олардан бас тарта алмайтындай етіп қол қоюға мүмкіндік береді. Жалпы әдістердің бірі-құжатты бір жақты криптографиялық хэш алгоритмі арқылы бастапқы өткізіп жіберу, оны өзгерту өте қиын. Хэш функциясы, әдетте, бастапқы құжаттың көлеміне тәуелсіз белгіленген ұзындықтың нәтижесін береді. Ең танымал хэш функциясы-SHA-1 (Secure Hash Algorithm), 20 байттық нәтиже шығарады (NIST, 1995). SHA-1 — SHA-256 және SHA-512 соңғы нұсқалары сәйкесінше 32 және 64 байттық нәтиже береді, бірақ олар әлі де кең таралмады. Келесі қадам бұрын сипатталған ашық кілттерді шифрлауды қолдануды қамтиды. Құжат иесі d(hash) алу үшін өзінің жеке кілтін хэшке қолданады. Қолтаңба блогы деп аталатын бұл мән құжатқа бекітіліп, алушыға жіберіледі (14-сурет). Кейде d функциясын хэшке қолдану хэш шифры деп аталады, бірақ іс жүзінде бұл Шифр емес, өйткені хэш шифрланбаған. Бұл жай математикалық хэш түрлендіру.



14-сурет. а — қолтаңба блогын есептеу; б-алушыға келетін нәрсе

Құжат пен хэшті алған кезде алушы алдымен SHA-1 алгоритмін немесе келісілген криптографиялық хэш функциясын қолдана отырып, құжаттың хэшін есептейді. Содан кейін алушы қол қою блогына E(D(hash) алу үшін жіберушінің ашық кілтін қолданады. Нәтижесінде, ол өзара жою арқылы шифрланған хэшті "шифрлайды" және оны бұрынғы түрінде алады. Егер есептелген хэш қолтаңба блогындағы хэшке сәйкес келмесе, онда құжат, қолтаңба блогы немесе екеуі де жалған болады(немесе кездейсоқ өзгертілген). Мұндай схеманың мәні мынада: баяу ашық кілтпен шифрлау тек хэшке қатысты қолданылады, бұл деректердің салыстырмалы түрде аз бөлігі. Айта кету керек, бұл әдіс барлық x үшін ғана жұмыс істейді

$E(D(x)) = x$.

Барлық шифрлау функциялары үшін бұл қасиеттің болуына алдын — ала кепілдік берілмейді, өйткені олардан талап етілгендердің бәрі шартты сақтау болып табылады

D(E(x)) = x,

мұндағы E-шифрлау функциясы, ал D-шифрлау функциясы. Қолтаңбаның қасиетін алу үшін оларды қолдану тәртібі ешқандай рөл атқармауы керек, яғни D және E коммутативті функциялар болуы керек. Бақытымызға орай, RSA алгоритмінде мұндай қасиет бар. Осы электрондық қолтаңба схемасын пайдалану үшін алушы жіберушінің ашық кілтін білуі керек. Кейбір пайдаланушылар өздері қолданатын ашық кілттерді өздерінің веб-парақтарына орналастырады. Басқалары шабуылдаушылар веб-бетті бұзып, олардың кілтін абайлап өзгертеді деп қорқады. Ашық кілттерді тарату үшін оларға басқа механизм қажет. Хабарлама жіберушілер үшін кең таралған әдістердің бірі-Пайдаланушының аты мен ашық кілті бар және сенім тудыратын үшінші тараптың сандық қолтаңбасы бар сертификатты (certificate) хабарламаға бекіту. Пайдаланушы осы үшінші тараптың ашық кілтін алғаннан кейін, олардың сертификаттарын жасау үшін осы үшінші сенімді Тараптың қызметтерін пайдаланатын барлық жіберушілердің сертификаттарын қабылдай алады. Сертификаттарға қол қоятын сенімді үшінші тарап сертификаттау орталығы (Certification Authority (CA)) деп аталады. Алайда, пайдаланушы CA қол қойған сертификатты тексеру үшін оған осы орталықтың ашық кілті қажет. Ол қайдан келуі керек және пайдаланушы оның түпнұсқалығын қалай тексере алады? Мұны жалпы қабылданған тәртіпте жасау үшін ашық кілттерді басқарудың толық схемасы қажет, оны ашық кілттердің инфрақұрылымы деп атайды (Public Key Infrastructure (PKI)). Веб-браузерлер үшін бұл мәселе ерекше түрде шешіледі: барлық браузерлер шамамен 40 Сертификаттау орталығының алдын-ала орнатылған кілттерімен келеді. Бұрын цифрлық қолтаңбалар үшін ашық кілттерді шифрлауды қолдану қарастырылған, бірақ ашық Кілттерді шифрлау қолданылмайтын схемалар да бар екенін атап өткен жөн.

Криптографиялық процессорлар

Барлық шифрлау жүйелері үшін кілттер қажет. Егер кілттер бұзылған болса, онда оларды пайдалануға негізделген барлық қауіпсіздік жүйесі бұзылған. Сондықтан кілттерді қауіпсіз сақтау ерекше маңызды. Бірақ қауіпті жүйеде кілттерді қауіпсіз сақтауды қалай ұйымдастыруға болады? Өнеркәсіптің ұсыныстарының бірі-кілттерді сақтауға арналған өзгермейтін жады бар криптографиялық процессор болып табылатын сенімді платформа модулі (сенімді платформа модулі (TPM) деп аталатын Чип. TPM ашық мәтінді блоктарды шифрлау немесе жедел жадтағы шифрланған мәтін блоктарын шифрлау сияқты криптографиялық операцияларды орындай алады. Ол сонымен қатар сандық қолтаңбаларды тексере алады. Осы операциялардың барлығын мамандандырылған жабдықпен орындау арқылы жұмыс жылдамдығы және оларды кеңінен қолдану ықтималдығы едәуір артады. Көптеген компьютерлер қазірдің өзінде TPM криптопроцессорларымен жабдықталған және болашақта мұндай компьютерлердің саны едәуір артады. TPM-бұл өте даулы құрылғы, өйткені әр түрлі тараптар TPM-ді кім басқаратыны және оны кім және кім қорғайтыны туралы әртүрлі идеяларға ие. Бұл тұжырымдаманың үлкен

жақтаушысы-Palladium, NGSCB және BitLocker-ді қоса алғанда, оны қолданудың барлық технологияларын жасаған Microsoft корпорациясы. Осы корпорация мамандарының пікірінше, Операциялық жүйе криптопроцессорды басқарады және оны, мысалы, қатты дискінің мазмұнын шифрлау үшін пайдаланады. Бірақ ол оны тыйым салынған бағдарламалық жасақтаманы іске қосудың алдын алу үшін қолданғысы келеді. Тыйым салынған бағдарламалық жасақтама қарақшылық (яғни заңсыз көшірілген) бағдарлама немесе операциялық жүйе іске қосуға рұқсат бермейтін бағдарлама болуы мүмкін. Егер TPM жүйені іске қосу процесіне тартылса, ол өндіруші TPM ішіне орналастырған құпия кілтпен қол қойылған амалдық жүйені ғана іске қоса алады. Оны тек операциялық жүйенің таңдаулы провайдерлері қолдана алады (мысалы, Microsoft).

ОЖҚ. Дәріс №5. Аутентификация. Әлсіз парольдер. Unix-тегі құпия сөзді қорғау. Бір реттік парольдер.

Әрбір сенімді (**secured**) компьютерлік жүйе кіру кезінде барлық пайдаланушылардан аутентификацияны талап етуі керек. Өйткені, егер Операциялық жүйе пайдаланушының кім екеніне сенімді бола алмаса, ол қандай файлдарға және басқа ресурстарға қол жеткізе алатындығын біле алмайды. Аутентификация тақырыбы тым маңызды емес болып көрінуі мүмкін, бірақ бұл күткеннен әлдеқайда қиын. Пайдаланушының аутентификациясы "Онтогенез филогенезді қайталайды"бөліміндегі 1-тарауда қарастырылған нәрселерге қатысты. ENIAC сияқты әмбебап машиналардың алғашқы модельдерінде жүйеге кіру процедурасын айтпағанда, Операциялық жүйе болған жоқ. Кейінгі пакеттік жүйелер мен уақытты бөлу жүйелері, әдетте, тапсырмалар мен пайдаланушыларды аутентификациялау үшін кіру процедураларына ие болды. Алғашқы шағын компьютерлерде (мысалы, PDP-1 және PDP-8) кіру процедурасы болған жоқ, бірақ UNIX операциялық жүйесінің PDP-11 шағын компьютеріне таралуымен бұл процедура қайтадан сұранысқа ие болды. Алғашқы дербес компьютерлерде (мысалы, Apple II және IBM PC бастапқы нұсқасы) кіру процедурасы болған жоқ, бірақ ол Linux және Windows 8 сияқты жеке компьютерлерге арналған күрделі операциялық жүйелерде пайда болды (бірақ қысқа мерзімді пайдаланушылар оны өшіре алады). Сонымен, қазіргі уақытта көптеген адамдар интернет-банкингті пайдалану, электронды сатып алу, музыка жүктеу және басқа да әрекеттерді орындау мақсатында қашықтағы компьютерлер жүйесіне кіреді (жанама түрде). Барлық осы әрекеттер аутентификацияланған кіруді қажет етеді. Аутентификацияның маңыздылығына көз жеткізгеннен кейін оны жүзеге асырудың қолайлы әдісін іздеуге көшу керек. Пайдаланушыларды кіруге тырысқанда аутентификацияның көптеген әдістері үш негізгі қағидаға негізделеді, атап айтқанда: пайдаланушыға белгілі бір нәрсеге сүйену; қолданушыда бар нәрсеге сүйену; ол білдіретін нәрсеге сүйену. Кейде олардың екеуі қосымша қауіпсіздік шараларын қажет етеді. Осы принциптердің негізінде өзіндік күрделілігі мен қауіпсіздік қасиеттері бар аутентификацияның әртүрлі схемалары жасалады. Олардың барлығы кезек бойынша келесі бөлімдерде қаралатын болады. Пайдаланушыдан тіркеу аты мен парольді енгізуді талап ететін аутентификация нысаны кеңінен қолданылды. Құпия сөзді қорғауды түсіну және жүзеге асыру оңайырақ. Ең қарапайым іске асыру жұптардың негізгі тізілімін (тіркеу аты, пароль) қолдау болып табылады. Енгізілген тіркеу атауы тізілімде ізделеді және енгізілген пароль сақталған парольмен салыстырылады. Егер олар сәйкес келсе, кіруге рұқсат етіледі, егер жоқ болса, ол қабылданбайды. Монитордың жанында орналасқан қызықты көздерден жасыру үшін әрдайым енгізілетін құпия сөз экранда көрсетілмейді. Windows жүйесінде әр терілген таңба Жұлдызшамен көрсетіледі. Unix жүйесінде парольді енгізу кезінде ештеңе көрсетілмейді. Бұл екі схема әртүрлі қасиеттерге ие. Windows жүйесінде қолданылатын Схема ұмытылған пайдаланушыларға терілген таңбалардың санын бақылауды жеңілдетеді, бірақ сонымен бірге құпия сөздің ұзындығын ашады. Қауіпсіздік тұрғысынан үнсіздік-бұл алтын. Қателіктер қауіпсіздік деңгейіне айтарлықтай әсер етуі мүмкін, тағы бір сала

9.15-суретте көрсетілген. Жүйе хабарламаларды жоғарғы регистрде көрсетіп, пайдаланушы төменгі регистрде енгізген кезде сәтті кіру 15, а-суретте көрсетілген. 15, б-суретте крекердің а жүйесіне кіруге сәтсіз әрекеті көрсетілген, ал 9.15, в-суретте хакердің В жүйесіне кіру әрекеті сәтсіз аяқталды.

15-сурет. Жүйеге кіру: а-сәтті; б-атын енгізгеннен кейін кіруден бас тарту; в-аты мен паролін енгізгеннен кейін кіруден бас тарту

15-суреттегі В жүйесі қате тіркеу атауын көргеннен кейін кіруден бас тартады. Бұл қате деп саналады, өйткені бұзушыға тиісті атау табылғанға дейін Тіркеу атауын таңдауды жалғастыруға мүмкіндік береді. 15-суретте бұзушыдан әрқашан пароль сұралады және ол енгізген тіркеу аты сәйкес келе ме, жоқ па, оған хабарланбайды. Ол тек тіркеу аты мен парольдің тіркесімі жарамсыз екенін біледі. Кіру процедураларынан басқа, көптеген Ноутбуктер жоғалған немесе ұрланған жағдайда олардың мазмұнын қорғау үшін тіркеу аты мен парольді қажет етеді. Әрине, бұл ештеңеден жақсы, бірақ көп емес. Кез-келген ноутбук қуатты қосып, BIOS орнату бағдарламасына Del немесе F8 пернесін немесе амалдық жүйені іске қоспас бұрын BIOS параметрлерін шақыруға байланысты басқа пернелерді (әдетте экранда көрсетілетін ақпарат) басып тұрып кіре алады. Орнату бағдарламасында ол қатты дискіні іске қоспас бұрын компьютерді USB флэш-дискісінен бастауға нұсқау беру арқылы қолданылатын іске қосу құрылғыларының тізбегін өзгерте алады. Содан кейін табылған ноутбук толық операциялық жүйесі бар USB флэш-дискісін салып, компьютерді осы дискіден іске қосады. Амалдық жүйені іске қосқаннан кейін қатты дискіні орнатуға болады (UNIX-те) немесе d: құрылғысы ретінде қол жетімді (Windows-та). Осындай жағдайдың алдын алу үшін көптеген BIOS жүйелері қолданушыға BIOS теңшеу бағдарламасын парольмен қорғауға мүмкіндік береді, осылайша тек пайдаланушы жүктеу кезінде құрылғылардың сауалнама тізбегін өзгерте алады. Егер сіз ноутбукты қолдансаңыз, кітап оқуды тоқтатып, BIOS-қа құпия сөзді орнатып, оқуға оралыңыз.

Әлсіз парольдер. Көбінесе бұзушылар жүйеге кіріп, жоспарланған компьютерге (мысалы, Интернет арқылы) қосылып, қолданыстағы комбинациялардың бірін тапқанша көптеген комбинацияларды (тіркеу аты, пароль) сұрыптайды. Көптеген адамдар қандай-да бір түрде тіркеу атауында өздерінің нақты атын қолданады. Мысалы, Ellen Ann Smith үшін ellen, smith, ellen smith, ellen-smith, ellen тиісті нұсқалары болуы мүмкін. smith, esmith, easmith және eas. "Сіздің жаңа туған нәрестеңізге арналған 4096 есім" сияқты кітаптардың бірімен және фамилиялармен толтырылған телефон кітабымен қаруланған крекер жүйеге шабуыл жасайтын ел үшін ықтимал тіркеу атауларының компьютерленген тізімін оңай жасай алады (ellen_smith АҚШ-та немесе Ұлыбританияда бірдей жұмыс істей алады, бірақ Жапонияда емес). Әрине, тек тіркеу атауын шешу жеткіліксіз. Әлі бәсекелік пароль. Мұны істеу қиын ба? Сіз ойлағаннан әлдеқайда оңай. UNIX жүйелеріндегі

парольдердің қауіпсіздігі бойынша классикалық жұмысты Моррис пен Томпсон жасаған (Моррис және Томпсон, 1979). Олар Ықтимал парольдердің тізімін жасады: атаулар мен фамилиялар, көше атаулары, қала атаулары, орташа сөздіктердегі сөздер (сонымен қатар кері жазудағы сөздер), тіркеу нөмірлері және т.б. содан кейін олар сәйкестіктерді іздеудегі жүйелік пароль файлымен салыстырды. Барлық парольдердің 86% - дан астамы олардың тізімінде табылды. Егер біреу жоғары білікті пайдаланушылар сәтті парольдерді таңдайды деп ойласа, мен сендіре аламын: олай емес. 2012 жылы 6,4 миллион LinkedIn құпия сөздері бұзылғаннан кейін Интернетке кірген кезде, көптеген нәтижелерді талдау қызықты болды. Ең танымал пароль "пароль" сөзі болды. Ең танымал екінші пароль "123456" болды (ең танымал ондыққа "1234", "12345" және "12345678"кірді). Олардың тұрақтылығы туралы айтудың қажеті жоқ. Шын мәнінде, хакерлер ықтимал пайдаланушы аттарының (логиндердің) тізімін және ықтимал парольдердің тізімін оңай құрастырып, оларды компьютерлердің ең қол жетімді санына сәйкестендіретін бағдарламаны іске қоса алады. Ұқсас зерттеулер IOActive-де 2013 жылдың наурыз айында жүргізілді. Үйдегі маршрутизаторлар мен приставкалардың ұзақ тізімі олардың осалдығын қарапайым бұзылулардан анықтау үшін сканерленді. Көптеген пайдаланушы аттары мен парольдерді қолдануға тырысудың орнына, олар өндірушілер орнатқан бір ғана танымал логин мен парольді қолданып көрді деп болжауға болады. Пайдаланушылар бұл мәндерді бірден өзгертеді деп болжалды, бірақ көбісі бұлай жасамағаны белгілі болды. Зерттеушілер жүздеген мың ұқсас құрылғылардың ықтимал осалдығын анықтады. Ирандық ядролық объектіге Stuxnet-шабуыл кезінде бұзушылар центрифугаларды басқаратын Siemens компьютерлерінде жылдар бойы Интернетте айналып келе жатқан әдепкі парольдердің бірін қолданғанын пайдаланып, одан да алаңдатарлық жағдай деп санауға болады. Бүкіләлемдік ғаламтордың танымалдылығының артуы мәселені одан әрі қиындатты. Бір парольдің орнына, қазір көптеген адамдарда он немесе одан да көп нәрсе бар. Барлық осы құпия сөздерді есте сақтау өте қиын болғандықтан, адамдар қарапайым, тұрақсыз парольді алып, оны көптеген веб-сайттарда қолдануға тырысады (Флоренсио және Герли, 2007; Gaw and Felten, 2006). Құпия сөздерді табу оңай деп қорқудың қажеті бар ма? Әрине, тұрарлық. 1998 жылы San Jose Mercury News Беркли қаласының тұрғыны Питер Шипли бірнеше компьютерлерді бір коммутатордың барлық 10 000 нөміріне (мысалы, (415) 7700xxxx) қоңырау шалған автокөлік қоңырау құрылғысы ретінде қолданғанын, телефон компанияларын алдау үшін нөмірлерді кездейсоқ сұрыптап, компьютерлердің осындай қолданылуына кедергі келтіріп, осы әрекеттерді анықтауға тырысқанын хабарлады. 2,6 миллион қоңырау шалғаннан кейін, Шипли шығанағы аймағында 20000 компьютерді тапты, олардың 200-інде қорғаныс жоқ. Оның айтуынша, тұрақты крекер басқа адамдардың компьютерлерінің 75% - ына ене алады (Деннинг, 1999). Бірақ мұның бәрі ежелгі уақытқа оралу болды, өйткені компьютер 2,6 миллион телефон нөміріне қоңырау шалуы керек еді. Хакерлер тек Калифорнияда ғана емес. Австралиялық крекер дәл осылай жасауға тырысты. Ол бұзған жүйелердің ішінде Сауд Арабиясындағы Citibank компьютері болды, оған несие карталарының нөмірлері мен несие лимиттері туралы ақпарат (бір жағдайда — 5 миллион доллар), сондай-ақ транзакциялар туралы жазбалар (оның ішінде қоғамдық үйге бару үшін кем дегенде

бір аударым) алуға мүмкіндік берді. Оның әріптесі, сонымен қатар банк жүйесіне еніп, 4000 несие картасының нөмірін жинады (Деннинг, 1999). Мұндай ақпарат мақсатсыз пайдаланылған кезде, банк клиент жария етуге тиіс деп мәлімдей отырып, өз қатесінің мүмкіндігін сөзсіз және батыл түрде жоққа шығарады. Интернет хакерлер үшін нағыз сыйлық болды. Ол исключил все алғанда, олардың жұмыс. Енді кейбір телефон нөмірлеріне қоңырау шалудың қажеті жоқ. Қоңырау келесі рәсімге айналды. Хакер IP-мекен-жайлар жиынтығы бойынша пинг сұрауларын (желілік пакеттер) жіберетін сценарий жаза алады. Егер ол қандай-да бір жауап алса, содан кейін сценарий TCP қосылымын машинада іске қосылуы мүмкін барлық қызметтерге орнатуға тырысты. Жоғарыда айтылғандай, портты сканерлеу деп аталатын және сценарийді нөлден жазудың орнына, крекер Nmap сияқты мамандандырылған құралдарды қолдана алады, бұл кеңейтілген портты сканерлеу технологиясының кең спектрін ұсынады. Алайын, қандай қызмет, қай машинада іске қосылған взломщик еді кірісуге шабуыл. Мысалы, егер крекер құпия сөзді қорғауды зерттегісі келсе, ол осы аутентификация әдісін қолданған қызметтерге, мысалы, telnet серверіне немесе тіпті веб-серверге қосылуы керек еді. Біз әдепкі пароль немесе бір немесе басқа тұрақсыз пароль бұзушыларға көптеген есептік жазбалардан, кейде тіпті толық әкімшілік құқықтармен егін жинауға мүмкіндік беретінін көрдік.

Unix-тегі құпия сөзді қорғау. Кейбір ескірген операциялық жүйелерде парольдер дискіде шифрланбаған түрде сақталды, бірақ әдеттегі жүйелік қорғаныс механизмдері арқылы қорғалды. Дискідегі барлық құпия сөздерді шифрланбаған түрде сақтау біршама қиындық тудырды, өйткені көптеген адамдар оларға қол жеткізе алды. Оларға жүйелік әкімшілер, машина операторлары, қызметкерлер, бағдарламашылар, менеджерлер және тіпті кейбір хатшылар кіруі мүмкін. UNIX-те сәтті шешім қолданылады. Кіру бағдарламасы Пайдаланушыдан оның аты мен паролін енгізуді сұрайды. Пароль дереу" шифрланған", оны бекітілген деректер блогын шифрлау кілті ретінде пайдалану арқылы. Шын мәнінде, кіріс ретінде парольмен және шығыс ретінде парольмен бір жақты функция іске қосылады. Бұл процесс нақты шифрлау емес, бірақ оны шифрлау деп атау оңайырақ. Содан кейін кіру бағдарламасы ASCII жолдарының қарапайым жиынтығы болып табылатын құпия сөз файлын оқиды, әр пайдаланушы үшін біреуі тіркеу аты бар жолды тапқанша. Егер осы жолдағы шифрланған пароль тек есептелген шифрланған парольге сәйкес келсе, кіруге рұқсат етіледі, ал егер сәйкес келмесе, ол қабылданбайды. Бұл схеманың артықшылығы-ешкім, тіпті артықшылықты қолданушы да құпия сөздерді таба алмайды, өйткені олар жүйеде шифрланбаған түрде сақталмайды. Иллюстрация үшін біз қазір шифрланған пароль пароль өрісінде сақталады делік. Кейінірек біз UNIX-тің қазіргі нұсқаларында бұл енді жасалмайтынын көреміз. Егер крекер шифрланған парольді ала алса, схема келесі шабуылға ұшырауы мүмкін. Алдымен крекер Моррис пен Томпсон сияқты мүмкін парольдердің сөздігін жасайды. Олар белгілі алгоритмді қолдана отырып алдын-ала шифрланған. Бұл процесс қанша уақытқа созылатыны маңызды емес, өйткені ол бұзылуға тырысқанға дейін де жүреді. Енді парольдер мен шифрланған парольдердің тізімімен қаруланған крекер соққы береді. Ол жалпыға қол жетімді

пароль файлын оқиды және одан барлық шифрланған парольдерді шығарады. Бұл парольдер оның тізіміндегі шифрланған парольдермен салыстырылады. Әр сәйкестікте тіркеу аты және шифрланбаған пароль белгілі болады. Қабықта жұмыс істейтін қарапайым сценарий бұл процесті автоматтандырады және оны бірнеше секунд ішінде жасауға болады. Сценарийді қалыпты іске қосу арқылы сіз ондаған парольдерді ала аласыз. Мұндай шабуылдың мүмкіндігін түсінген Моррис пен Томпсон шабуылды іс жүзінде пайдасыз ететін технологияны сипаттады. Идея әр парольмен тұз (тұз) деп аталатын N-биттік кездейсоқ санды байланыстыру болды. Кездейсоқ Сан парольдің әр өзгеруімен өзгереді. Ол құпия сөз файлында шифрланбаған түрде сақталады және оны кез-келген адам оқи алады. Шифрланған парольді пароль файлында сақтаудың орнына, алдымен пароль мен кездейсоқ Сан біріктіріліп, содан кейін бірге шифрланады. Алынған шифрланған нәтиже пароль файлында сақталады. 16-сурет бес қолданушыға арналған пароль файлын көрсетеді: Bobbie, Tony, Laura, Mark және Deborah. Файлдағы әр пайдаланушыға үтірмен бөлінген үш жазбадан тұратын бір жол бөлінген: тіркеу атауы, тұз және "пароль + тұз"шифрланған тіркесімі. E(Dog, 4238) жазбасы Bobbie-ге тиесілі Dog паролін кездейсоқ тағайындалған 4238 тұзымен біріктірудің және оларды e шифрлау функциясы арқылы өткізудің нәтижесін білдіреді.

Bobbie,4238, e(Dog,4238)
Tony,2918,e(6%%TaeFF,2918)
Laura,6902, e(Shakespeare,6902)
Mark,1694,e(XaB#Bwcz,1694)
Deborah, 1092, e(LordByron,1092)

16-сурет. Шифрланған парольдерді алдын-ала есептеу үшін тұзды пайдалану

Енді ықтимал парольдердің тізімін жасап, оларды шифрлап, нәтижелерін сұрыпталған f файлында сақтағысы келетін крекер үшін Бұл не болатынын көрейік, осылайша әр шифрланған парольді оңай табуға болады. Егер шабуылдаушы пароль Dog сөзі болуы мүмкін деп болжаса, енді Dog-ны шифрлау және нәтижені F файлына қою жеткіліксіз. ол Dog0000, Dog0001, Dog0002 және т. б.сияқты 2n жолдарын шифрлап, осы жолдардың барлығын F файлына енгізуі керек. UNIX жүйесінде бұл әдіс n = 12 көмегімен қолданылады. Қосымша қорғаныс үшін UNIX-тің кейбір заманауи нұсқаларында шифрланған парольдер, әдетте, жеке "көлеңкелі" файлда сақталады, оны пароль файлынан айырмашылығы тек түбірлік пайдаланушы оқи алады. Пароль файлына тұз қосып, оны оқылмайтын етіп өзгерту, жанама (және баяу) оқуды қоспағанда, осы файлға қатысты көптеген шабуылдарға төтеп бере алады.

Бір реттік парольдер. Артықшылықты пайдаланушылардың көпшілігі қарапайым адамдарды айына бір рет құпия сөздерін өзгертуге көндіреді. Бірақ, оларға ешкім ескертулерін қабылдай алады. Жүйеге кірген сайын парольді өзгерту одан да экстремалды болып көрінеді, бұл бір реттік парольдерді (one-time passwords) пайдалануға әкеледі. Мұндай парольдерді пайдаланған кезде пайдаланушы парольдер тізімі бар блокнот алады. Жүйеге кірген сайын тізімдегі келесі пароль қолданылады. Хакер парольді тапса да, оны пайдалана алмайды, өйткені келесі

жолы басқа парольді пайдалану керек болады. Бұл жағдайда пайдаланушы пароль дәптерін жоғалтпауы керек.

Шын мәнінде, Лесли Лампорт ойлап тапқан (Lamport, 1981) қолданушыға бір реттік парольдерді қолдана отырып, қауіпті желі арқылы қауіпсіз кіруді қамтамасыз ететін талғампаз схеманың арқасында сіз осындай блокнотсыз жасай аласыз. Лэмпорт әдісін үйдегі жеке компьютерде жұмыс істейтін қолданушы интернет арқылы серверге кіру үшін қолдана алады, тіпті хакерлер барлық трафикті екі бағытта да қадағалап, көшіре алады. Сонымен қатар, файлдық жүйелерде серверде де, пайдаланушының үй компьютерінде де ешқандай құпияны сақтаудың қажеті жоқ. Кейде бұл әдіс бір жақты хэш тізбегі деп аталады (one-way hash chain). Алгоритм бір жақты функцияға, яғни y = f(x) функциясына негізделген, ол x болған кезде y-ны оңай алуға мүмкіндік беретін қасиетке ие. Кіріс және шығыс бірдей ұзындықта болуы керек, мысалы, 256 бит. Пайдаланушы есте сақтау керек құпия сөзді таңдайды. Ол сонымен қатар алгоритм құра алатын бір реттік парольдер санына сәйкес келетін N бүтін санды таңдайды. Мысалы, N = 4 — ті қарастырайық, бірақ іс жүзінде әлдеқайда үлкен мән қолданылуы керек N. егер құпия пароль s болса, онда бірінші пароль бір жақты функцияны N рет іске қосу арқылы алынады: P1 = F(F(F(F(s))) екінші пароль бір жақты функцияны (n – 1) рет іске қосу арқылы алынады: P2 = f(f(F(s))) үшінші парольді алу үшін f функциясы екі рет, ал төртінші парольді алу үшін — бір рет. Жалпы алғанда, Pi -1 = f(Pi). Мұнда бастысы, егер осы тізбектегі кез-келген пароль болса, оған тиесілі алдыңғы парольді есептеу қиын емес, бірақ келесі парольді есептеу мүмкін емес екенін түсіну. Мысалы, P2 болса, P1 табу қиын емес, бірақ P3 табу мүмкін емес. Сервер P0 санымен іске қосылады, ол f(P1). Бұл мән пайдаланушының тіркеу атымен байланысты пароль файлының жазбасында сақталады, сонымен бірге 1 бүтін санмен бірге келесі P1 паролі талап етілетінін көрсетеді. Пайдаланушы бірінші рет кіргісі келгенде, ол өзінің тіркеу атауын пароль файлындағы 1 бүтін санды жіберу арқылы жауап беретін серверге жібереді. Пайдаланушы машинасы жауап ретінде P1 жібереді, оны жергілікті жерде терілген s мәнінен есептеуге болады. Содан кейін сервер f(P1) есептейді және алынған мәнді пароль файлында (P0) сақталған мәнмен салыстырады. Егер мәндер сәйкес келсе, кіруге рұқсат етіледі, бүтін сан 2-ге дейін артады, ал пароль файлындағы P0 мәні P1 мәніне сәйкес келеді. Келесі кіру кезінде сервер пайдаланушыға 2 санын жібереді, ал пайдаланушы машинасы P2 паролін есептейді. Содан кейін сервер f(P2) есептейді және алынған мәнді пароль файлындағы жазбамен салыстырады. Егер мәндер сәйкес келсе, кіруге рұқсат етіледі, бүтін сан 3-ке дейін артады, ал пароль файлындағы P2 паролі P1 паролінің үстіне жазылады. Бұл схеманың жұмыс істеуіне мүмкіндік беретін қасиет , егер крекер Pi-ді ұстап алса да, одан Pi + 1 мәнін есептеу мүмкіндігі болмайды, ол тек Pi -1 паролінің мәнін есептей алады, ол бұрын қолданылған және ешқандай құндылықты білдірмейді. Барлық n парольдер қолданылған кезде, сервер жаңа құпия кілтпен қайта іске қосылады.