УДК 004.056

C++ ТІЛІНДЕГІ ПРОГРАММАЛАРДЫҢ ОСАЛДЫҚТАРЫ

Сейдулла Қымбат Дарханқызы
skymbat.01@gmail.com
Л.Н.Гумилев атындағы ЕҰУ ақпарттық технологиялар факультетінің
4-ші курс студенті, Нұр-Сұлтан, Қазақстан
Ғылыми жетекші – Сагиндыков Каким Молдабекович

Программалардың осалдығы-программалық жасақтаманы әзірлеу кезеңінде әзірлеушілер жіберген қателер болып табылады. Олар зиянкестерге программаның функцияларына немесе онда сақталған деректерге заңсыз қол жеткізуге мүмкіндік береді.

Кемшіліктер программаның өмірлік циклінің кез-келген кезеңінде, жобалаудан бастап дайын өнімді шығаруға дейін пайда болуы мүмкін. Кейбір жағдайларда әзірлеушілер жобалау және баптау үшін әдейі бос орындар қалдырады, оларды бекдорлар немесе жарияланбаған мүмкіндіктер ретінде қарастыруға болады. Ал кей жағдайларда, осалдықтардың пайда болуы программалық кодта диверсиялық типтегі ақаулардың пайда болу қаупін арттыратын әртүрлі шығу тегі бар даму құралдарын қолданумен байланысты.

Осалдықтар қосалқы компоненттерді немесе еркін таратылатын кодты (open source) қосу нәтижесінде де пайда болады. Кей жағдайларда басқа әзірлеушілердің коды көбінесе қауіпсіздікті мұқият талдаусыз және тексерусіз "сол күйінде" қолданылады. Құрылған өнімге қосымша құжатсыз функцияларды немесе элементтерді әдейі енгізетін инсайдерлік әзірлеушілер тобының болуын жоққа шығаруға болмайды.

Программалық жасақтама осалдықтарының жіктелуі прграммалық кодты жобалау немесе жазу кезеңінде туындаған қателіктерден туындайды. Пайда болу сатысына байланысты қауіптің бұл түрі жобалау, іске асыру және конфигурацияның осалдықтарына бөлінеді.

Жобалау кезінде жіберілген қателерді анықтау және жою қиын. Бұл алгоритмдердің дәл еместігі, бетбелгілер, әртүрлі модульдер арасындағы интерфейстегі сәйкессіздіктер немесе аппараттық құралдармен өзара әрекеттесу хаттамаларында, оңтайлы емес технологияларды енгізу. Оларды жою өте көп уақытты қажет ететін процесс — өйткені олар айқын емес жағдайларда көрінуі мүмкін-мысалы, трафиктің қарастырылған көлемінен асып кетсе немесе қосымша жабдықтардың көп мөлшерін қосқан кезде, бұл қажетті қауіпсіздік деңгейін қамтамасыз етуді қиындатады және брандмауэрды айналып өту жолдарының пайда болуына әкеледі.

Іске асырудың осалдығы программаны жазу немесе оған қауіпсіздік алгоритмдерін енгізу сатысында пайда болады. Бұл есептеу процесінің дұрыс ұйымдастырылмауы, синтаксистік және логикалық ақаулар. Бұл жағдайда кемшілік буфердің толып кетуіне немесе басқа түрдегі ақаулардың пайда болуына әкелуі мүмкін. Оларды анықтауға көп уақыт кетеді, ал жою машина кодының белгілі бір бөліктерін түзетуді білдіреді.

Аппараттық және программалық жасақтама конфигурациясының қателері өте жиі кездеседі. Олардың жиі кездесетін себептері-сапалы дамудың жеткіліксіздігі және қосымша функциялардың дұрыс жұмыс істеуі үшін сынақтардың болмауы. Бұл санатқа тым қарапайым парольдер мен өзгертілмеген әдепкі есептік жазбаларды жатқызуға болады.

Статистикаға сәйкес, осалдықтар көбінесе танымал және кең таралған өнімдерде — жұмыс үстелі мен мобильді операциялық жүйелерде, браузерлерде кездеседі. Жүйенің осалдығының негізінде көптеген программаларға шабуыл жасалады. Мысалы, CVE Details ақпарт көзінің анализіне қарайтын болсақ 1999 жыл мен 2019 жыл арасындағы осалдылық типіне байланысты қанша осалдылық анықталғанын көре аламыз (Сурет 1.).

Уязвимости по типу

Год	# уязвимостей	DoS	Выполнение кода	Переполнение	Повреждение памяти	SQL-инъекция	XSS	Обход каталогов	Разделение HTTP-ответа	Обойти что-то	Получить информацию	Получите привилегии	CSRF	Включение файла	# эксплойтов
1999 г.	894	177	112	172			2	7		25	16	103			2
2000 г.	1020	257	208	206		2	4	20		48	19	139			
2001 г.	1677	403	403	297		7	34	123		83	36	220		2	2
2002 г.	2156	498	553	435	2	41 год	200	103		127	74	199	2	14	1
2003 г.	1527	381	477	371	2	49	129	60	1	62	69	144		16	5
2004 г.	2451	580	614	410	3	148	291	111	12	145	96	134	5	38	5
2005 г.	4935	838	1627	657	21 год	604	786	202	15	289	261	221	11	100	14
2006 г.	6610	893	2719	663	91	967	1302	322	8	267	271	184	18	849	30
2007 г.	6520	1101	2601	954	95	706	884	339	14	267	324	242	69	700	44 год
2008 г.	5632	894	2310	699	128	1101	807	363	7	288	270	188	83	170	74
2009 г.	5736	1035	2185	700	188	963	851	322	9	337	302	223	115	138	738
2010 г.	4652	1102	1714 г.	680	342	520	605	275	8	234	282	238	86	73	1493
2011 г.	4155	1221	1334	770	351	294	467	108	7	197	409	206	58	17	557
2012 г.	5297	1425	1459	843	423	243	758	122	13	344	389	250	166	14	624
2013	5191	1455	1186	859	366	156	650	110	7	352	511	274	123	1	205
2014 г.	7946	1598	1574	848	420	305	1105	204	12	457	2106	239	264	2	401
2015 г.	6484	1791	1826 г.	1083	749	218	778	150	12	577	748	367	248	5	127
2016 г.	6447	2028 г.	1494	1324	717	94	497	99	15	444	843	600	87	7	1
2017 г.	14714	3154	3004	2495	745	508	1518	279	11	629	1639	459	327	18	6
2018 г.	16556	1853 г.	3041	2368	400	517	2042 г.	531	11	708	1424	247	461	31 год	4
2019 г.	12174	919	2277	1247	296	410	1593	280	4	495	900	129	398	40	
Общий	122774	23603	32718	18081	5339	7853	15303	4130	166	6375	10989	5006	2521	2235	4333
% Из всех		19,2	26,6	14,7	4.3	6.4	12,5	3,4	0,1	5.2	9.0	4.1	2.1	1,8	

Сурет 1. Осалдық түріне байланысты 1999 жылдан 2019 жылға дейінгі кезеңдегі осалдықтар саны туралы деректер

CVE (Common Vulnerabilities and Exposures) - ақпараттық қауіпсіздіктің жалпыға белгілі осалдықтарының дерекқорына сәйкес – 1-сурет бойынша буфердің толып кету осалдықтары, сондай-ақ деректерді енгізу мен жолдарды пішімдеудің дұрыс өңделмеуі барлық анықталған осалдықтардың ішінде ең танымал болып табылады деп қорытынды жасауға болады.

Компьютердің жадына тікелей қол жеткізуді (көрсеткіштердің еркін арифметикасы, жадты бөлу және босату) және типтерді келтіруді қолдайтын, бірақ автоматты түрде тексерілмеген C және C++ сияқты абстракция деңгейі төмен программалау тілдері жадқа қол жеткізу тұрғысынан қауіпсіз емес болып табылады.

Veracode, 2013 жылдың 1 қазанынан 2015 жылдың 31 наурызына дейін 208670 қосымшаны талдаған, қауіпсіздік фирмасының талдауына сәйкес - сценарий тілдері қосымшалардағы қауіпсіздік қателерін көбірек тудырады, деп жариялаған. Осы талдау негізінде, C ++ тілінен - 26 кемшіліктер/МБ (8,8 сыни кемшіліктер/МБ) анықталған болатын. Бұл анализ CWE санаттарының негізінде зардап шеккен қосымшалар пайызын көрсетті.

CWE (Common Weakness Enumeration) - бұл әлсіз жерлер мен программалық қамтамасыз етудің осалдығы санаттарының жүйесі. Ол программалық қамтамасыз етудің кемшіліктерін түсіну және осы кемшіліктерді анықтау, жою және болдырмау үшін пайдаланылуы мүмкін автоматтандырылған құралдарды жасау мақсатында қолданылады. C++ тілінде ең жиі кездесетін 10 осалдылық негізінде жасалған анализден, қосымшалардың ең көп зардап шегетін CWE санаттары: қателерді өңдеу(75%), буфердің толып кетуі(56%) мен буферлірді басқару қателіктері(53%) екендігін байқауға болады (Сурет 2).

Language	CWE Category	Apps Affected
C++	Error Handling	75%
	Buffer Overflow	56%
	Buffer Management Errors	53%
	Numeric Errors	50%
	Cryptographic Issues	42%
	Directory Traversal	42%
	Potential Backdoor	37%
	Race Conditions	29%
	Dangerous Functions	28%
	Code Quality	27%

Сурет 2. C++ программалау тілі бойынша осалдықтардың негізгі санаттары

Қазіргі уақытта C++ программалау тілі әзірлеушілер арасында ең көп таралған тілдердің бірі болып табылады, сонымен қатар ол ең осал болып табылатын программалау тілдерінің тізімінде бастапқы орындардың бірін алады. Әзірлеушілер үшін C++ тілінің осалдықтарының жақсы толық базасының болмауы ең қиын мәселелердің бірі болып табылады. Мұндай база болмаса, қауіпсіз программаларды жазу туралы айту, жобалау және әзірлеу процесін бақылау өте қиын.

Осы себепті қауіпсіз программалауды, яғни, осалдықтардың кездейсоқ енгізілуіне жол бермейтін және зиянды программалар мен рұқсат етілмеген қолжетімділікке төзімділікті қамтамасыз ететін программалық қамтамасыз етуді әзірлеу әдістемесін қолдану керек. Қауіпсіз программалаудың міндеті – пайдаланушының деректерін ұрлау, осалдылықтарды туындатпау, қорғау, жүйені сақтау болып табылады.

Сонымен қатар, программалық жасақтамалар үшін статикалық талдаудың рөлін де ұмытпау керек. Статикалық талдау кодты жазу сатысында қателер мен осалдылықтарды анықтауға мүмкіндік береді, бұл болашақта жөндеу уақытын үнемдейді және даму процесін арзандатады. Статикалық талдау әсіресе ендірілген жүйелерді жобалау кезінде пайдалы болуы мүмкін. Мұндай жүйелерде қателер көбінесе жазбаша программалық кодты ғана емес, сонымен қатар құрылғының өзін де қамтуы мүмкін, статикалық талдауды қолдану қателерді, кем дегенде, программалық жасақтама әзірлеушісінің жағында іздеуге уақытты үнемдеп, құрылғының өзін тексеруге көп уақыт бөлуге болады.

Кірістірілген жүйелерге арналған программадағы қателіктердің бағасы өте жоғары, өйткені оларды жаппай өндіріс басталған кезде түзету мүмкін емес немесе өте қиын және қымбат. Кейде бұл тіпті адамдардың өмірі үшін де қауіпті (мысалы машиналар, зымырандар үшін бағдарламалық жасақтама жасау). Сондықтан, ендірілген құрылғылардың коды мүмкіндігінше мұқият тексерілуі керек, әсіресе қателіктер құрбандыққа немесе материалдық шығындарға әкелмеуі үшін.

Қолданылған әдебиеттер тізімі

1. Роберт С. С. Безопасное программирование на языке С и С++. Пер. с англ. – М: ООО “И.Д. Вильямс”, 2015;
2. Безопасность програмного кода. [Свободный отчет компании Эшлон, 2010];
3. Кубрин С.С., Самарин Н.Н. Современное состояние инструментальных средств анализа программного обеспечения на уязвимость., 2013;
4. Область применения анализаторов кода [http://www.controlengrussia.com/programmnye-sredstva/bezopasnost-programmnye-sredstva/appchecker/-]
5. Обнаружение уязвимостей в теории и на практике, или почему не существует идеального статического анализатора. [https://habr.com/company/solarsecurity/blog/420337/]
6. Безопасное программирование. [https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5]
7. https://www.cvedetails.com/vulnerabilities-by-types.php
8. https://news.softpedia.com/news/top-programming-languages-that-generate-software-vulnerabilities-497101.shtml