

**«ԿՐԹՈՒԹՅԱՆ ԶԱՐԳԱՑՄԱՆ ԵՎ ՆՈՐԱՐԱՐՈՒԹՅՈՒՆՆԵՐԻ
ԱԶԳԱՅԻՆ ԿԵՆՏՐՈՆ» ՀԻՄՆԱԴՐԱՄ**

**ԳԵՎՈՐԳ ՂԱԶԱՐԻ ՔՈՍՅԱՆ
ԵՂԻՍԱԲԵԹ ԾԵՐՈՒՆԻ ԱԼԱՎԵՐԴՅԱՆ**

**ԿԻՐԵՌԱՆՎՏԱՆԳՈՒԹՅԱՆ
ՀԻՄՈՒՆՔՆԵՐ**

ԵՐԵՎԱՆ - 2025

ՀՏԴ 004.056(07)

ԳՄԴ 16.8g7

Ք 845

Քոսյան Գ., Ալավերդյան Ե.

Ք 845 Կիբեռանվտանգության հիմունքներ / Գ. Քոսյան, Ե. Ալավերդյան.-
Եր.: ԿԶՆԱԿ, 2025.- 122 էջ:

Ներկա ուսումնական ձեռնարկը նախատեսված է միջին մասնագիտական կրթության ուսանողների համար, ովքեր ծեռք են բերելու համակարգչային ցանցերի կիբեռանվտանգության ապահովման և սպասարկման տեխնիկի որակավորում: Ձեռնարկը կարող է օգտակար լինել նաև բուհերի ուսանողների և այն ընթերցողների համար, ովքեր ցանկություն ունեն ինքնուրույն յուրացնել կիբեռանվտանգության հիմունքները:

ՀՏԴ 004.056(07)

ԳՄԴ 16.8g7

ISBN 978-9939-977-23-2

© ԿԶՆԱԿ, 2025

Հեղինակների կողմից

«Կիբեռանվտանգության հիմունքներ» ձեռնարկը նախատեսված է մասնագիտական կրթության և ուսուցման (ՄԿՈՒ) ուսումնական հաստատությունների ուսանողների համար, ովքեր կրթության ավարտին ձեռք են բերելու միջին մասնագիտական կրթության 0614.01.5 «Կիբեռանվտանգություն» մասնագիտության 0614.01.01.5 «Տեխնիկ՝ համակարգչային ցանցերի կիբեռանվտանգության ապահովման և սպասարկման» որակավորում: Ձեռնարկը մեծապես միտված է ուսանողների մոտ կիբեռանվտանգության ոլորտի գործնական գիտելիքներ և հմտություններ ձևավորելուն, ինչպիսիք են՝ համակարգչային ցանցերին հատուկ կիբեռոռիսկերը, կիբեռսպառնալիքների տեսակների իմացությունը, դրանց չեզոքացման մեխանիզմները, տվյալների գաղտնիության ապահովման պատմական և արդի համակարգչային դարաշրջանի եղանակներն ու մեթոդները, գաղտնագրային ռազմավարական ալգորիթմների գործողության սկզբունքը, տվյալների հենքերի, վեբ կիրառությունների և ամպային ծառայությունների պաշտպանված շահագործման գործիքակազմը՝ ներառյալ դրանց մշտադիտարկումը և դիմակայունության թեստավորումը: Ձեռնարկը ներառում է կիբեռանվտանգության տեխնիկի որակավորման համար անհրաժեշտ տեսական գիտելիքներ համակարգչային հետախուզության և հետաքննության, տեղեկատվական անվտանգության օրենսդրության և էթիկայի վերաբերյալ, որոնց յուրացումը շրջանավարտներին հնարավորություն կընձեռի կիրառել ձեռք բերված գիտելիքներն ու հմտությունները աշխատաշուկայում, ինչպես նաև ուսումը հաջողությամբ շարունակել բարձրագույն ուսումնական հաստատություններում:

Ձեռնարկը կազմվել է որպես ոլորտի հետաքրքրաշարժ ներածություն՝ միևնույն ժամանակ նաև որպես համապարփակ տեղեկատու: Բովանդակությունը չենք ծանրաբեռնել մաթեմատիկական ապացույցներով՝ հաշվի առնելով, որ ընթերցողները միջին մասնագիտական կրթական ոլորտի ուսանողներ են և, այնուամենայնիվ, բերված հիմնարար գաղափարները և սկզբունքները շարադրված են համապատասխան մաթեմատիկական մոդելների մեկնաբանության հասկանալի և հասանելի բառապաշարով՝ հավատարիմ մնալով ձևակերպումների ճշգրտությանն ու անաղարտությանը: Ներկայացված գրականության ցանկն ամբողջությամբ ընդգրկում է շարադրանքում բերված բոլոր գաղափարները և դրանց գործնական իրականացման սկզբունքները:

Նախաբանում ներկայացված են մարդկության գոյության երկու իրար զուգահեռ տարածությունները և դրանցում առկա անվտանգային ռիսկերի և դրանց հաղթահարման նմանություններն ու տարբերությունները:

Համառոտագիրը ներկայացնում է «Կիբեռանվտանգության տեխնիկի» որակավորմանը միտված մասնագիտական մոդուլները և դրանց առնչվող մասնագիտական գործառույթները:

Գլուխ 1-ը ներկայացնում է կիբեռանվտանգության սահմանումը, հիմնարար հասկացությունները, նպատակները և խնդիրները:

Գլուխ 2-ում հանրամատչելի լեզվով ներկայացված են արդի կիբեռռիսկերը և կիբեռսպառնալիքների վեկտորները՝ ըստ տարակարգերի և դրսևորման ձևերի:

Գլուխ 3-ում ներկայացված են հին աշխարհի՝ ներառյալ հայկական լեռնաշխարհի, պատմական գաղտնագրերի օրինակներ, որոնց կառուցման սկզբունքները փոխառնվել են ժամանակակից գաղտնագրերում՝ ներկայացվելով բանաձևերի և մաթեմատիկական բարդ հնարքների տեսքով:

Գլուխ 4-ը նվիրված է գաղտնագրության համակարգչային դարաշրջանի ռազմավարական ալգորիթմների լրաբանմանը: Չցանկանալով ծանրաբեռնել ձեռնարկը բերված գաղտնագրային ալգորիթմների մաթեմատիկական հիմունքներով և ալգորիթմների մանրակրկիտ նկարագրությամբ (դրանց մեծ մասը բավական աշխատատար են)՝ բերված ալգորիթմների համար նշված են հղումներ QR կոդի տեսքով: Այցելելով հղմամբ՝ ուսանողները և դասավանդողները հնարավորություն ունեն ծանոթանալ գաղտնագրերի և՛ մաթեմատիկական, և՛ ալգորիթմական հիմունքներին: Ալգորիթմների մի մասի մաթեմատիկական ապարատի ըմբռնումը պահանջում է բնագիտական բարձրագույն կրթություն՝ ուստի դրանց համապարփակ մեկնաբանությունը և բերված ինքնաստուգողական աշխատանքների կատարումը թողնում ենք ըստ հայեցողության:

Գլուխ 5-ում ներկայացված են համակարգչային ցանցերին բնորոշ կիբեռռիսկերը և դրանց չեզոքացման մեխանիզմները:

Գլուխ 6-ում բերված են տվյալների հենքերին բնորոշ կիբեռռիսկերը և դրանց չեզոքացման մեխանիզմները՝ ներառյալ տվյալների հասանելիության վերահսկման և պաշտպանության կազմակերպման հիմնարար մոտեցումները:

Գլուխ 7-ում նկարագրվում են վեբ սպառնալիքները և վեբ կիբեռանվտանգության կազմակերպման միջոցներն ու եղանակները:

Գլուխ 8-ը նվիրված է ամպային ծառայություններին բնորոշ կիբեռռիս-կերի և այդ ծառայությունների կիբեռանվտանգության ճանաչողական լու-սաբանմանը:

Գլուխ 9-ում հակիրճ ներկայացված են համակարգչային հետախուզու-թյան հիմնարար մոտեցումները:

Գլուխ 10-ը նվիրված է կիբեռանվտանգության օրենսդրության և էթի-կայի հիմնահարցերին: Հայաստանի Հանրապետության օրենսդրական ակ-տերով ամրագրված են համապատասխան հոդվածներով և դրույթներով, որոնց հղումը ներկայացված է՝ դրանց համապարփակ ուսումնասիրությունը թողնելով ընթերցողներին:

Առանձին հավելվածով բերված են ինքնաստուգման առաջադրանքների տիպեր՝ ձևավորված կիբեռանվտանգության տեխնիկի մասնագիտական որակավորման և կարողունակությունների շրջանակի հաշվի առնմամբ:

Ներածություն

Ժամանակակից աշխարհում մարդկությունն ապրում է երկու իրար զուգահեռ իրականության մեջ. մեկը՝ ֆիզիկական միջավայրում, մյուսը՝ տեղեկատվական տեխնոլոգիաների միջոցով ստեղծված և մեր գիտակցության մեջ օրավուր հաստատվող առցանց միջավայրում, որն այժմ անվանվում է «կիբեռտարածություն»: Կիբեռտարածությունը համակարգիչների և ցանցերի վիրտուալ աշխարհ է, որը ներառում է առցանց գործառույթներ և տվյալների շրջանառություն: «Կիբեռ» նախաձանցը նշանակում է այն էլեկտրոնային միջավայրը, որտեղ տեղի են ունենում վերոհիշյալ ընթացակարգերը: «Կիբեռնետիկա» բառը ծագում է հունարեն *kybernētēs* (κυβερνήτης) բառից, որը նշանակում է ղեկավար, կառավարիչ, անհատ կամ տեխնոլոգիական միավոր, ով առաջնորդում կամ վերահսկում է ինչ-որ ընթացակարգ: «Կիբեռանվտանգությունը» համակարգիչների, ցանցերի, ծրագրային ապահովման և տվյալների պաշտպանությունն է կիբեռգրոհներից, վնասաբեր ծրագրերից և չարտոնված հասանելիությունից: Այն նման է կողպեքների, անվտանգության ահազանգերի, անվտանգության աշխատակիցների կողմից համակարգերի և անձնակազմի պաշտպանությանը, բայց՝ թվային տարածության մեջ:

Կիբեռնետիկան մարդկանց, կենդանիների և ավտոմատացված համակարգերի կառավարման գիտություն է: Այն ուսումնասիրում է, թե ինչպես են համակարգերը տվյալներ հավաքագրում, մշակում, որոշումներ կայացնում, գործում և միևնույն ժամանակ հարմարվում միջավայրի և դրանից ստացվող ազդականների փոփոխություններին՝ պահպանելով կայունություն և գործելու շարունակականություն: Կիբեռնետիկան վերաբերում է նաև նրան, թե ինչպես են համակարգերը «մտածում», «սովորում», «վերահսկում» և «արձագանքում», անկախ նրանից, թե դրանք կենսաբանական են (ինչպես մարդու ուղեղը), մեխանիկական (ինչպես ռոբոտը), թե՞ թվային (ինչպես խելացի ջերմակարգավորիչը կամ համակարգչային ցանցը):

Մեզ հանրաձանոթ համացանցը գործում է կիբեռտարածությունում: Ներկայումս դժվար է պատկերացնել մեկին, ով չիմանա, թե ինչ է համացանցը, կամ՝ ինչպես օգտվել դրանից: Համացանցը վերացրեց աշխարհագրական սահմանները՝ հարմարավետ ու, որը ոչ պակաս կարևոր է՝ մատչելի կապի և տեսակապի հնարավորություն ստեղծեց անխտիր բոլոր օգտագործողների համար՝ անկախ տարիքից և զբաղմունքի տեսակից: Մենք օգտա-

գործում ենք համացանցը սովորելու, աշխատելու, հաղորդակցվելու, առցանց գնումներ կատարելու և նույնիսկ մեքենաներ վարելու համար: Ականատես ենք, թե ինչպես են տեղեկատվական տեխնոլոգիաները կարևոր դեր խաղում մեր կյանքի գրեթե բոլոր ոլորտներում:



Նկ. 1. Համացանց:

Այնուամենայնիվ, եթե համեմատենք մարդկության գոյության երկու միջավայրերը՝ ֆիզիկականը և կիբեռտարածքայինը, կնկատենք, որ ֆիզիկական աշխարհում ֆիզիկական անվտանգության զգացողության և պաշտպանության հակումը մեր գիտակցության մեջ սերմանվում է մեր գիտակցական հասակից և առաջնորդվում է առողջ բանականությամբ, տեսանելի սպառնալիքների ընկալմամբ և սոցիալապես ընդունված անվտանգության գործելակերպով: Մինչդեռ՝ անվտանգության զգացողությունը և բնագործ կիբեռտարածությունում հաճախ թերագնահատվում, անտեսվում և նույնիսկ՝ արհամարհվում է, իսկ անհոգություն նկատվում է անգամ ոլորտի պատասխանատուների կողմից: Սա մեծապես պայմանավորված է կիբեռռիսկերի անտեսանելի և վերացական բնույթով, որոնք միշտ չէ, որ անմիջական կամ ակնհայտ հետևանքներ են ունենում:



Նկ. 2. Կիբեռռիսկեր՝ կիբեռտարածությունում:

Ե՛վ ֆիզիկական հանցագործները, և՛ կիբեռհանցագործները նպատակ ունեն շահագործել մարդկային սովերային հոգեբանությունը և խոցելիությունները անձնական շահույթի համար: Այնուամենայնիվ, այս երկու տեսակի հանցագործների գործունեության եղանակները և գործողությունների ընկալումներն իրարից խիստ տարբեր են: Ֆիզիկական հանցագործները, որպես կանոն, տեսանելի են կամ կարող են հետագայում հայտնաբերվել իրավապահ մարմինների կողմից, և նրանց գործողությունները, ինչպիսիք են՝ կողոպուտը կամ հարձակումը, ճանաչվում են հասարակության կողմից: Մարդիկ, որպես կանոն, գիտակցում են այս ռիսկերը: Ֆիզիկական տարածքներում մարդկանց փոքր տարիքից սովորեցնում են փակել դռները, խուսափել մոֆ նրբանցքներից և զգուշավոր լինել անձանոթների հետ: Այս վարքագծերն ամրապնդվում են իրական աշխարհի փորձով, ավագ սերնդի հոգածությամբ և տեսանելի ազդակներ ունեն: Մարդիկ նաև գիտեն, թե ինչպես արձագանքել, օգնություն խնդրել, զանգահարել ոստիկանություն կամ ֆիզիկապես պաշտպանվել:

Ի տարբերություն դրա, կիբեռհանցագործները գործում են անտեսանելի միջավայրերում՝ հաճախ չթողնելով որևէ ֆիզիկական հետք: Նրանց հարձակումները՝ որոնց անվանում ենք «կիբեռգրոհներ», ինչպիսիք են՝ անձնական տվյալների կողոպուտը կամ վնասաբեր ծրագրերի տարածումը, կարող են աննկատ մնալ օրերով, շաբաթներով կամ նույնիսկ ամիսներով: Մարդիկ կարող են անգամ չգիտակցել, որ իրենց անզգուշության կամ անփութության պատճառով թիրախ են դարձել մինչև այն պահը, երբ վնասն արդեն պատճառվել է, օրինակ՝ պարզվել է, որ իրենց անձնական բանկային քարտի տվյալներն արտահոսել են:

Ֆիզիկական հանցագործությունների և կիբեռհանցագործությունների մեկ այլ սկզբունքային տարբերությունը նրանց դրսևորման ձևերի մեջ է: Ֆիզիկական հանցագործությունները տեղի են ունենում հանցագործի ֆիզիկական ներկայությամբ և մասնակցությամբ: Կան դեպքեր, երբ հանցագործները խղճի խայթ են զգացել և ետ կանգնել հանցանք կատարելուց, կամ էլ զղջացել են դրա համար: Մինչդեռ՝ կիբեռհանցագործությունները կատարվում են առանց զոհի հետ անմիջական շփման, հեռավար և ինքնաշխատ կերպով, անանուն և, որ ամենակարևորն է՝ սառնասրտորեն՝ առանց խղճի խայթի կամ զղջման դրսևորումների՝ մեծամասամբ մնալով չհայտնաբերված ո՛չ գրոհի ընթացքում, ո՛չ էլ դրա ավարտին: Ավելին, մինչդեռ հասարակությունը ընդհանուր առմամբ տեղեկացված է ֆիզիկական հանցագործու-

թյունների դրսևորումներից, ուստի և համեմատաբար զգուշավոր է՝ կիբեռ-հանցագործությունների դրսևորումները լայն քննարկման առարկա չլինելով՝ մտահոգություն չեն առաջացնում:

Ֆիզիկական հանցագործության վայրերն ապահովում են նյութական ապացույցներ (մատնահետքեր, վկաներ, տեսախցիկների գրանցումներ), մինչդեռ կիբեռհանցագործությունների հետաքննությունները հիմնված են թվային հետաքննության և համացանցային հասցեների (IP address) հետևման վրա, որը միշտ չէ, որ ցանկալի արդյունք է բերում՝ դրանց քողարկված կամ անանուն լինելու պատճառով: Արդյունքում, կիբեռհանցագործությունները և դրանց կատարողները ավելի դժվար են նույնականացվում և հետապնդվում:

Քանի որ հասարակությունը դառնում է ավելի թվային, կիբեռհանցագործությունների մշտապես պատահման ընկալումը դառնում է ավելի ու ավելի կարևոր՝ արդյունավետ կիբեռպաշտպանական մեխանիզմներ ներգրավելու և օգտագործողների շրջանում թվային պատասխանատու վարքագիծ մշակելու համար: Առցանց աշխարհն անընդհատ փոփոխվում է, անընդմեջ հայտնվում են նոր առցանց հարթակներ և տեխնոլոգիաներ՝ յուրաքանչյուրն իր խոցելիություններով: Տեխնոլոգիաների արագընթաց զարգացումը՝ զուգորդված կիբեռանվտանգային լուծումների թերացումների և իրազեկության պակասի հետ, անհատներին և կազմակերպություններին ենթարկում է կիբեռռիսկերի և կիբեռգրոհների: Օգտագործողները կարող են չգիտակցել, որ թվացյալ անվնաս գործողությունները՝ կասկածելի հղման վրա սեղմելը, թույլ գաղտնաբառերի օգտագործումը կամ անձնական տվյալների չափազանց շատ տարածումը, կարող են լուրջ կիբեռսպառնալիքների դռներ բացել:

Հասարակության կիբեռիրազեկվածության մակարդակի բարձրացումը և կիբեռանվտանգության հիմնարար հմտությունների զարգացումը այլևս ընտրովի չեն. դրանք կարևոր են ոչ միայն ոլորտի մասնագետների, այլև՝ բոլորի համար: Այս երկակի կյանքում, որ մենք ապրում ենք, հասարակական գիտակցությունը պետք է զուգակցվի կիբեռպաշտպանվածության զգացողության հետ՝ երկու աշխարհներում էլ անվտանգ գոյատևման համար:

ԳԼՈՒԽ 1. Կիբեռանվտանգության ներածություն

Ի՞նչ է կիբեռանվտանգությունը:

Ոչ խիստ սահմանամար՝ կիբեռանվտանգությունը տեխնոլոգիաների, գործընթացների և կառավարման միջոցների ամբողջություն է, որը նախատեսված է տեղեկատվական համակարգերը, ցանցերը, թվային սարքավորումները և նրանցում շրջանառվող տվյալները կիբեռգրոհներից պաշտպանելու համար:

Կիբեռանվտանգությունը նաև տեղեկատվական անվտանգության ծրագրային գործիքակազմի, հակավիրուսային ծրագրերի և ձեռնարկությունների կազմակերպչական քաղաքականության համադրություն է, որի շնորհիվ ապահովվում է տվյալների գաղտնիությունը և դրանց կողոպուտի կանխարգելումը:



Նկ. 3. Կիբեռանվտանգության կազմակերպում:

Կիբեռանվտանգությունը կարևորագույն նշանակություն ունի ազգային ենթակառուցվածքների պաշտպանության և թվային ծառայությունների շարունակականության ապահովման գործում: Այն երաշխավորում է կազմակերպությունների բնականոն աշխատանքը՝ ապահովելով նրանց տվյալների անվտանգությունն ու սեփականությունը և կազմակերպության հանդեպ վստահության կայուն վարկանիշը:

Կիբեռանվտանգությունը կիրառում է գործիքներ, գործընթացներ և վարքագծեր, որոնք օգնում են կանխել կիբեռհանցագործությունները, ապահովել թվային համակարգերի և դրանցում շրջանառվող տվյալների անվտանգությունը՝ ներառյալ

- **տվյալների գաղտնիությունը**, որը ենթադրում է տվյալների քողարկում կամ դրանց հասանելիության տրամադրում բացառապես լիազորված անձանց

- **տվյալների ամբողջականությունը**, որով հավաստիացվում է տվյալների ճշգրտությունը, անխաթարությունը և վավերականությունը դրանց ողջ կենսացիկլի ընթացքում՝ ստեղծումից մինչև հեռացումը
- **տվյալների հասանելիությունը**, որով երաշխավորվում է անհատների և կազմակերպության տվյալների ըստ պահանջի օգտագործելիությունը
- **հաշվետունակությունը**, որով կազմակերպությունները պատասխանատվություն են կրում օգտագործողների անձնական տվյալների հետ գործողությունների համապատասխանության և համաչափության համար:
- **կատարված գործողությունների չհերքումը**, որը ներդրված անվտանգության ինքնաշխատ մեխանիզմ է: Այն հավաստիացնում է, որ տվյալների հասանելիության և դրանց մշակման դերակատարները հետագայում չկարողանան հերքել իրենց մասնակցությունը տվյալ գործողությանը կամ տեղի ունեցած հաղորդակցությանը:

Կիբեռգրոհը ցանկացած դիտավորյալ ջանք է տվյալների, հավելվածների կամ այլ ակտիվների կողոպուտի, բացահայտման, նենգափոխման կամ ոչնչացման նպատակով՝ ցանցին, թվային համակարգին կամ թվային սարքին չարտոնված հասանելիության միջոցով: Կիբեռգրոհի արդյունքում վնաս կրած անհատը, կազմակերպությունը կամ կառավարությունը կոչվում է «կիբեռզոհ»:

Նշենք, որ «զոհի» զգացողությունը համակում է անհատին կիբեռգրոհի հետևանքով պատճառված վնասի, անհարմարության կամ հոգեբանական ճնշվածության արդյունքում, բայց ոչ երբեք գրոհի իրականացման ընթացքում:



Նկ. 4. Կիբեռգրոհներ տեղեկատվական համակարգերի և անհատներ դեմ:

Համակարգչային դարաշրջանում ձեռնարկատիրական գործունեության մեծ մասը, ուսուցումը, հեռավար աշխատանքը, բանկային գործառույթները և այլն, տեղի են ունենում առցանց՝ բավականաչափ հարմարավետ և մատչելի պայմաններում: Այնուամենայնիվ, այսպիսի թվայնացված կենսագործունեությունն իր բոլոր հարմարությունների հետ մեկտեղ լուրջ վտանգներ է ներկայացնում. հաքերները, վիրուսները և կիբեռհանցագործները կարող են՝

- բացահայտել գաղտնաբառեր և օգտագործել դրանք,
- առցանց կողոպտել գումարներ,
- լրտեսել հաղորդակցություններ,
- ոչնչացնել արժեքավոր տվյալներ,
- աննկատ կերպով վերահսկել համակարգերը և պատճենել տվյալները,
- հասանելիություն ձեռք բերելով անհատական կամ ձեռնարկության համակարգիչներին, գաղտնագրել դրանց պարունակությունը և փրկագին պահանջել՝ տվյալների հասանելիությունը վերականգնելու համար:

Անհատ մարդկանց համար նման կիբեռգրոհները կարող են նշանակել տվյալների սեփականության և գաղտնիության կորուստ, մինչդեռ ձեռնարկությունների և կառավարությունների համար դրանք կարող են հանգեցնել խափանումների, վարկանիշի կորստի և անվտանգային լուրջ վնասների:

Հաքերության առաջին դրսևորումները նկատվեցին ARPANET համացանցի ստեղծումից հետո՝ 1969–1970-ական թվականներին և դարձան նոր մշակույթ՝ նոր միջավայրում: Ընդ որում «հաքեր» բառն այն ժամանակ չունեւ բացասական երանգ, այլ՝ ներկայացնում էր մարդկանց, ովքեր հետաքրքրասիրությունից դրդված՝ փորձում էին բացահայտել թվային համակարգերի կազմակերպումը և ճարտարապետությունը՝ ի ցույց դնելով սեփական հնարքները և կարողությունները՝ ամենևին նպատակ չունենալով վնաս պատճառել:

Պակիստանցի եղբայրների կողմից 1986 թվականին ստեղծված Brain վիրուսը պարզապես ցույց էր տալիս իրենց անունները:

Առավել հայտնի հաքերներից մեկը՝ Կևին Միտնիկը, 1995 թվականին ձերբակալվեց ամերիկյան կառավարության կողմից՝ հեռախոսային ցանցերի ազդանշանային համակարգի հետազոտման և դրանցում հանցավոր միջամտության համար. գաղտնալսում, անվճար զանգերի իրականացում և համակարգչային տարատեսակ խարդախություններ:

1988-ին Ռոբերտ Մորիսը ստեղծեց առաջին լայնածավալ ցանցային որդը՝ Morris Worm-ը, որը խափանեց ARPANET ցանցի 10%-ը:

Արդեն 2000-ականներից առաջ եկան առանձնակի խոշոր դրամաշորթություններ: Հաքերները սկսեցին աշխատել կազմակերպված խմբերով՝ վաճառքի հանելով կողոպտված տվյալները սև շուկաներում: Առաջին անգամ խոսվեց նաև կիբեռպատերազմների մասին: Իրանի միջուկային ցենտրիֆուգի դեմ 2010 թվականին կիրառվեց Stuxnet վիրուսը, որն ի թիվս պատճառած այլ վնասների՝ վարակեց 200,000 համակարգիչ:

Համակարգիչների արգելափակումը, բովանդակության հնարավոր գաղտնագրումը և ապագաղտնագրման դիմաց փրկագնի պահանջի (ransomware) հաքերային գրոհների սկիզբն առաջին անգամ դրվեց 2017 թվականին WannaCry վիրուսի ստեղծմամբ: Այդ նույն թվականին տվյալների խոշոր արտահոսք տեղի ունեցավ Equifax ընկերությունում, որի արդյունքում ավելի քան 140 միլիոն մարդու անհատական տվյալներ հասանելի դարձան հաքերներին: Համացանցում շարունակաբար է հայտնվում թարմացվող վիճակագրություն կիբեռգրոհների վերաբերյալ:

Հրապարակայնորեն հասանելի տեղեկատվության հավաքագրման և վերլուծության տեխնոլոգիաներ (Open Source Intelligence, OSINT, Social media intelligence, SOCMINT) առաջ եկան՝ հետախուզական տվյալներ հավաքագրելու համար տարբեր սկզբնաղբյուրներից, ինչպիսիք են՝ լրատվամիջոցները, համացանցը, անձնական և կառավարական գրառումները և մասնագիտական հրապարակումները: Այդ հրապարակումներն օգտագործվում են տարբեր նպատակներով, այդ թվում՝ ազգային անվտանգության, իրավապահ մարմինների, գործարար հետախուզության և կիբեռանվտանգության նկատառումներով: Հաքերները ևս գերծ չմնացին նման հետախուզությունն օգտագործել անհատներին ու կազմակերպություններին թիրախավորելու համար:

Տվյալների հավաքագրման նշված տեխնոլոգիաներին կարելի է ծանոթանալ հետևյալ հղումներով^(*),⁽²⁾:

Կիբեռգրոհների շարունակական վիճակագրությունը միջազգային հանրությանը ստիպեց լուրջ վերաբերվել դրանց՝ որպես քրեական հանցագործությունների նորաձև և նորաճ դրսևորումների և որպես մշտապես հավա-

* 1) <https://www.sans.org/blog/what-is-open-source-intelligence>,

2) <https://brand24.com/blog/what-is-social-media-intelligence>

նական իրադարձությունների: Որպես դատավարական գործառույթ՝ ներմուծվեց այսպես կոչված «համակարգչային հետաքննության» (computer forensics) գաղափարը, որի նպատակն է՝ կիրեռահանցագործությունների բացահայտման դեպքում (որն ի դեպ՝ ոչ միշտ է տեղի ունենում) դրանք վերաճակերպել որպես քաղաքացիական և քրեական օրենսգրքով սահմանված հանցագործություններ՝ վնասի չափի գնահատման և համապատասխան պատժաչափի սահմանման հնարավորությամբ:



Նկ. 5. Համակարգչային հետաքննություն:

Տեղեկատվական իրազեկ հանրությունն այժմ ավելի լրջորեն է վերաբերվում կիրեռահանցագործություններին, քանի որ դրանք կարող են թիրախավորել յուրաքանչյուր անհատի, ով տիրապետում է համացանցին միացվող անհատական համակարգչի կամ խելացի սարքերի՝ առանց տարիքային սահմանափակման:

Կիրեռագրոհների վերաբերյալ տեղեկացվածությունը մեզ օգնում է ճանաչել և հնարավորինս կանխարգելել դրանք: Տիպային վտանգներից են՝

- մտավոր սեփականության կորուստը,
- անհատական և գործարար տվյալների արտահոսքը,
- համակարգչային և ցանցային վիրուսների ներթափանցումը,
- վնասաբեր ծրագրերի ներբեռնումը՝ անզգուշությամբ կամ անտեղյակության հետևանքով,
- հաքերների «պասիվ» գրոհները, երբ գրոհողը պարզապես գաղտնալսում է հաղորդակցությունը՝ առանց միջամտելու դրա ընթացքին,
- հաքերների «ակտիվ» գրոհները, որի դեպքում գրոհողը ոչ միայն գաղտնալսում է հաղորդակցությունը, այլև վնասաբեր գործողություններով միջամտում է դրա ընթացքին,

- էլեկտրոնային կեղծ նամակագրությունները, որոնք տարածվում են համացանցում՝ փորձելով մոլորեցնել արձագանքողներին և ուղղորդել նրանց վնասաբեր կայքեր,
- էլեկտրոնային հասցեների նմանակումը, երբ իրական հասցեներում որոշ նիշեր փոխարինվում են արտաքուստ իրարից աննշան տարբերվող նիշերով, որը, որպես կանոն, վրիպում է օգտագործողների ուշադրությունից,
- նիշքեր կամ հղումներ, որոնց ընթերցելիս կամ հետևելիս ինքնաշխատորեն տեղադրվում են վնասաբեր ծրագրեր,
- թույլ և հեշտությամբ կռահելի գաղտնաբառերի գուշակումը և շահագործումը,
- հաքերների կողմից թվային համակարգերի հասանելիության թույլ մեխանիզմների շրջանցումը,
- օգտագործողների նույնականացման և ինքնության հաստատման թույլ մեխանիզմները կամ դրանց բացակայությունը,
- հին և խոցելի ծրագրակազմի օգտագործումը, որը կարող է հաքերների կողմից ներթափանցման հնարավորություն դառնալ,
- չստուգված արտաքին կրիչների օգտագործումը, որի հետևանքով վնասաբեր ծրագրեր կարող են ներթափանցել տեղեկատվական համակարգ,
- դրամական ռեսուրսների խնայման նպատակով հանրային անլար ցանցերից երկար ժամանակով օգտվելը և այդ ընթացքում հաքերների կողմից թիրախավորվելը և այլն:

Կիբեռգրոհների մեծ մասից կարելի է հաջողությամբ պաշտպանվել՝ դրսևորելով չափազանց պարզ, միևնույն ժամանակ արդյունավետ վարքագիծ, որն անվանում են՝ «կիբեռհիգիենա»՝ նման ավանդական մարդկային հիգիենայի, որն արդեն իսկ արմատավորվել է մեր գիտակցության մեջ:

Կիբեռհիգիենան հուշում է՝

- օգտագործել եզակի և տառա-թվային-կետադրական այբուբենի հիման վրա կառուցված երկար գաղտնաբառեր և երբեք դրանք չփոխանցել ոչ ոքի և ոչ մի պատրվակով,
- չհետևել կասկածելի հղումներին,
- չընթերցել անհայտ անձանցից ստացված էլեկտրոնային նամակների առդիր նիշքերը,

- գործածել բացառապես արտոնված ծրագրակազմ,
- պարբերաբար պահուստավորել (պատճենել) համակարգչի պարունակությունը արտաքին էլեկտրոնային կրիչների վրա՝ պարբերաբար համոզվելով դրանց գործունակության մեջ,
- առցանց հարթակներից և ծառայություններից օգտվելու ավարտին փակել այդ հարթակները և ծառայությունները,
- հանրային անլար ցանցերից օգտվել ծայրահեղ անհրաժեշտության դեպքում և անհրաժեշտ ժամանակով:

Կիբեռանվտանգության խնդիրը նաև տեղեկատվական համակարգերում տվյալների գաղտնիության ապահովման մեխանիզմների ներդնումն է՝ զուգակցելով դրանք տվյալների նենգափոխումների կանխարգելման և տվյալների ամբողջականության վերահսկման տեխնոլոգիաների հետ: Սա իր հերթին ապահովում է տվյալների մշտապես հասանելիությունը օրինական օգտագործողների կողմից՝ միևնույն ժամանակ ապահովելով օգտագործողների գործողությունների հաշվետունակությունը այդ գործողությունների ընթացքի լոգավորման միջոցով, որով կանխվում է կատարված գործողություններից հետագա հրաժարումը:

Այսպիսով, կիբեռանվտանգությունը համընդհանուր ազգային անվտանգության բաղադրիչ է: Պատշաճ կազմակերպման պարագայում այն ապահովում է անձնական և պետական կարևորության տվյալների, մտավոր սեփականության և կրիտիկական ենթակառուցվածքների տեղեկատվական պաշտպանությունը, օգնում է կանխել ֆինանսական կեղծիքները, տվյալների չարտոնված հասանելիությունը, բացահայտումը, շահագործումը կամ նենգափոխումը:

ԳԼՈՒԽ 2. Կիբեռռիսկեր և կիբեռսպառնալիքի վեկտորներ

Ի՞նչ են կիբեռռիսկերը: Կիբեռռիսկերը համակարգիչների, թվային սարքերի և համացանցի օգտագործումից բխող հավանական վտանգներ են, որոնք կարող են հանգեցնել՝

- ֆինանսական վնասի,
- անձնական տվյալների կորստի,
- համակարգերին չարտոնված հասանելիության,
- ձեռնարկության վարկանիշի կորստի և այլն:

Կիբեռռիսկերը մեր առօրյա կյանքի մասն են կազմում՝ պայմանավորված մեր կողմից էլեկտրոնային փոստի, առցանց գնումների, սոցիալական ցանցերի և այլ առցանց հարթակների օգտագործմամբ, ուստի՝ կիբեռռիսկերի ընկալումն ու ճանաչումը այդ հարթակներում պաշտպանված մնալու միակ գրավականն է:

Ինչ է կիբեռսպառնալիքը: Կիբեռսպառնալիքը թվային համակարգի, տեղեկատվական ցանցի կամ թվային տվյալների դեմ հաքերային գրոհների և տարատեսակ վնասաբեր գործողությունների իրականացման հնարավորությունն է: Սա չի նշանակում, որ կիբեռգրոհն իրականում տեղի է ունեցել, այլ այն, որ կա դրա տեղի ունենալու հնարավորություն կամ հավանականություն և որոշակի ռիսկ:

Ի՞նչ է կիբեռսպառնալիքի վեկտորը: Կիբեռսպառնալիքի վեկտորը այն ուղին կամ մեթոդն է, որն օգտագործում է կիբեռգրոհողը՝ թվային համակարգին կամ ցանցին չարտոնված հասանելիություն ձեռք բերելու համար, ինչպես օրինակ՝ տան դուռը կամ պատուհանը, որի միջոցով չարագործը կարող է ներթափանցել կողոպուտի նպատակով:

Անհատական համակարգիչների և համակարգչային ցանցերի դեմ կիբեռսպառնալիքների ամենատարածված վեկտորներից են՝

Վնասաբեր ծրագրերի (malware) տեղադրումը, որի հետևանքով տեղի է ունենում՝

- վիրուսների ներթափանցում,
- տեղեկատվական համակարգի գործունեության արգելափակում,
- կեղծ լուրերի, հաղորդագրությունների և խուճապի տարածում (scareware),

- լրտեսող ծրագրերի տեղադրում՝ օգտանունների, անցաբառերի, գաղտնաբառերի, վարկային հաշվեհամարների և այլն կորզելու նպատակով (spyware),
- համակարգչի օպերացիոն համակարգի կամ այլ ակտիվների հասանելիության ձեռք բերում ադմինիստրատորի մակարդակով (rootkit),
- ինքնաբերաբար վերարտադրվող վնասաբեր կոդերի ինքնաշխատ տարածում տեղեկատվական համակարգում (worms) և այլն:

Սոցիալական ինժեներիա: Այս գրոհները հմտորեն մշակված տեխնոլոգիաներ են՝ հիմնված մարդկային բնավորության շահագործման վրա՝ դրդելով մարդկանց կատարել գործողություններ, որոնք սթափ գիտակցության պարագայում նրանց կողմից երբեք չէին կատարվի: Օրինակներից են՝ անձնական, ձեռնարկատիրական կամ պետական գաղտնի տեղեկատվության տրամադրումը, անզգուշության կամ փոթևոտության հետևանքով վնասաբեր ծրագրերի տեղադրումը, ստացված կեղծ հաղորդագրության հրատապության բնույթով պայմանավորված՝ ձեռնարկության դրամական միջոցներից փոխանցման անխոհեմ կատարումն առանց վերադասի հետ խորհրդակցելու և այլն: Սոցիալական ինժեներիայի դրսևորումներից են՝

- **Ֆիշինգը** (phishing), որի նպատակը կեղծ էլեկտրոնային նամակներ կամ հաղորդագրություններ ուղարկելու ճանապարհով ստացողի հոգեբանության վրա ազդելն է՝ տպավորություն թողնելով, որ օրինական ուղարկողից են: Դրանք սովորաբար դրդում են զոհին այցելել ինչ-որ հղմամբ կամ ընթերցել ինչ-որ առդիր նիշք, որի հետևանքով հնարավոր է դառնում զոհին ուղղորդել վնասաբեր հարթակ կամ կայք և հարկավոր տվյալներ կորզել նրանից՝ օրինական թվացող հարցերի ու պատասխանների առաջադրման միջոցով: Ստորև բերված նկարում պատկերված է օգտագործողի էկրանին հայտնված հաղորդագրության օրինակ՝ անցաբառի անհապաղ փոփոխման հրատապ պահանջով:



Նկ. 6. Ֆիշինգի դրսևորման օրինակ:

Փութկոտության հետևանքով դրա արձագանքման հետևանքը լինելու է հաքերի կողմից անցաբառի կողոպուտը և համապատասխան առցանց ծառայության ամբողջական վերահսկման հնարավորությունը՝ հետագա միջամտությունների իրականացմամբ: Օգտագործողի անունից կարող են ուղարկվել հաղորդագրություններ՝ նենգամիտ և անգամ վարկաբեկիչ բովանդակությամբ, որոնց հերքումը օրինական օգտագործողի համար կլինի անհնար:

- **թիրախային ֆիշինգը** (spear phishing), որն ուղղորդվում է առանձնակի հետաքրքրություն առաջացնող անձանց դեմ: Այդպիսի անձինք սովորաբար բացահայտվում են սոցիալական ցանցերում՝ իրենց տիրապետման տակ գտնվող արժեքավոր տեղեկությունների, ունեցվածք և գրավիչ այլ հատկանիշների ի ցույց դրմամբ՝ առաջացնելով այլոց չարամիտ հետաքրքրությունը:
- **«օվկիանոսային կետերի ֆիշինգը»** (whale phishing), որը թիրախային ֆիշինգի տեսակ է և ուղղորդվում է բարձրաստիճան պաշտոնյաների դեմ, քանի որ վերջիններս առավել տեղեկացված են տեղեկատվական համակարգի կառուցվածքից և ճարտարապետությունից, ուստի և ավելի կարևոր թիրախ են հաքերների համար:
- **գործարար էլեկտրոնային փոստի վարկաբեկումը** (business email compromise, BEC), որի դեպքում կիբերհանցագործը հրատապ նամակագրությամբ ներկայանում է որպես վերադաս, մատակարար, գործընկեր կամ նույնիսկ երկրի (անգամ որևէ այլ երկրի) բարձրաստիճան պաշտոնյա՝ ավելի ազդեցիկ տպավորություն թողնելու և գո-

հին մոլորության մեջ գցելու համար՝ հնարավորություն չտալով վերջինիս սթափորեն գիտակցել կատարվելիք արաքը:



Նկ. 7. Էլեկտրոնային փոստի վարկաբեկման ճանապարհով գումարի շորթում:

Ծառայության մերժման գրոհները (denial-of-service, DoS), որոնք հեղեղում են տեղեկատվական համակարգերի սերվերները մեծաքանակ կեղծ հարցումներով՝ զրկելով սերվերներին օրինական հարցումներին արձագանքելու հնարավորությունից՝ սույնով նվազեցնելով համակարգի կատարողականի ունակությունը: Ծառայության մերժման գրոհը կարող է լինել և՛ ինքնանպատակ, և՛ որպես որևէ հաջորդող գրոհի նախապատրաստում: Այս գրոհի տարատեսակ է նաև բաշխված ծառայության մերժման գրոհը (distributed denial-of-service, DDoS): DoS և DDoS գրոհների միջև տարբերությունն այն է, որ DoS գրոհները մեկնարկվում են մեկ գրոհողի կողմից, իսկ DDoS գրոհները՝ այսպես կոչված՝ «բոտների»՝ հաքերի վերահսկողության տակ գտնվող և վնասաբեր ծրագրերով վարակված թվային սարքերի ցանցի օգտագործմամբ:

Հաշվեհամարի վարկաբեկումը, որի դեպքում հաքերները հասանելիություն են ստանում են օրինական հաշվեհամարներին՝ հետևյալ դրսևորումներով.

- հաշվեհամարի մուտքային տվյալների կողոպուտ ֆիշինգային գրոհների միջոցով,
- այսպես կոչված՝ «մութ ցանցերից» (dark net) վճարի դիմաց թիրախային անձանց գաղտնաբառերի ձեռք բերում,
- գաղտնաբառերի հատարկում՝ դրանց բոլոր տարբերակների փորձարկման ծրագրերի օգտագործմամբ, ինչպիսիք են՝ «Hashcat:»-ը, «John the Ripper»-ը, կամ՝ այլ:

Մարդը՝ արանքում գրոհները (Man-in-the-Middle), որոնց անվանում են նաև գաղտնալսման գրոհ: Այս դեպքում հաքերը գաղտնի կերպով ներդրվում է հաղորդակցվող կողմերի միջև և վերահսկողության տակ վերցնելով հաղորդակցությունը՝ ընթերցում կամ նույնիսկ նենգափոխում է այն; Հաքերը կարող է փոփոխել իր IP հասցեն օգտագործողի IP հասցեի նմանակով և հաղորդակցվել թիրախային սերվերի հետ որպես օրինական հաճախորդ: Այսպիսի գրոհները լայնորեն իրականացվում են հատկապես չպաշտպանված հանրային wifi ցանցերում՝ օգտագործելով այդ տիրույթում օգտագործողների գտնվելու ժամանակը:

Գրոհներ մատակարարման շղթայի դեմ, որոնց դեպքում հաքերները ներթափանցում են տեղեկատվական համակարգ՝ թիրախավորելով ծրագրային ապահովման մատակարարներին կամ ծառայություններ մատուցողներին: Քանի որ մատակարարները որոշակիորեն կապված են իրենց հաճախորդների տեղեկատվական ցանցերի հետ, հաքերները կարող են օգտագործել այդ ցանցերը միաժամանակ մեկից ավելի թիրախների հասանելիության համար:

Կայքերի միջև սկրիպտավորումը (Cross-site scripting, XSS), որի դեպքում հաքերը վնասակար կոդ է ներմուծում օրինական վեբ կայքէջերի կամ վեբ ծրագրի մեջ: Արդյունքում, կայք կամ հավելված այցելելու դեպքում օգտագործողի վեբ զննարկչում ինքնաշխատորեն ծրագրային կոդ է գործարկվում՝ հնարավորություն ստեղծելով կոդոպտել գաղտնի տեղեկատվություն կամ ուղղորդել օգտագործողին վնասաբեր կայքեր:

SQL ներարկումը, որի դեպքում շահագործվում է կառուցվածքավորված հարցումների լեզուն (SQL)՝ կայքի կամ հավելվածի ներքին տվյալների հենքեր նենգամիտ հրամաններ ուղարկելու նպատակով: Հաքերները հրամանները մուտքագրում են որոնման և մուտքագրման պատուհանների միջոցով, որի արդյունքում կեղծ հրամանները՝ որպես արտոնված հրամաններ, փոխանցվում են տվյալների հենք՝ հանգեցնելով տվյալների և գաղտնի տեղեկույթի կոդոպուտի:



Նկ. 8. Կեղծ հրամանների ներարկման կիբեռգործի իրականացում:

Դոմենային անունների թունելավորումը և կեղծումը, որոնց միջոցով հաքերները թաքցնում են տվյալների արտահոսքը և չարամիտ երթևեկը դոմենային անունների համակարգի (Domain Name System, DNS) փաթեթների ներսում՝ շրջանցելով կրակե պատնեշները (firewall) և անվտանգության առկա մեխանիզմները: Դոմենային անունների համակարգը համացանցի «հեռախոսագիրքն» է, որը դոմենային անունները (օրինակ՝ «google.com») թարգմանում է թվային IP հասցեների (օրինակ՝ «172.217.160.142»)՝ համացանցին միացված թվային սարքերի միարժեքորեն նույնականացման համար: Դոմենային անունների թունելավորումը (DNS tunnelling) հնարավորություն է ստեղծում հաքերներին գաղտնի հաղորդակցման կապուղիներ ստեղծել այսպես կոչված՝ «Հրամանների և վերահսկողության» սերվերների (Command and Control Server, C&C Server) հետ՝ կողոպտված տվյալների փոխանցման նպատակով:

Դոմենային անունների կեղծման դեպքում (DNS Spoofing) հաքերը նենգափոխում է DNS գրառումները՝ փոխարինելով կայքի իրական IP հասցեն կեղծ հասցեով: Այն պահից, երբ զոհն այցելում է իրական կայք, անգիտակցաբար հայտնվում է չարամիտ պատճենի հարթակ, որի արդյունքում զոհի կողմում կարող է տվյալների կողոպուտ կամ վնասաբեր ծրագրի տարածում տեղի ունենալ:

Գրոհ՝ առանց նիշքերի ներդրման, որի դեպքում գրոհողներն օգտվելով տեղեկատվական համակարգի խոցելիություններից՝ չարամիտ կոդեր

ներարկում են ուղղակիորեն համակարգչի հիշողության մեջ: Հաքերները հաջողությամբ օգտագործում են նաև օպերացիոն համակարգերում ներկայացված սկրիպտային գործիքները, օրինակ՝ «Microsoft Windows PowerShell»-ը՝ չարամիտ սկրիպտներ գործարկելու համար, որով ոչ միայն գաղտնի տվյալների կողոպուտ, այլև տեղեկատվական համակարգի կարգաբերումների նենգափոխում կարող են իրականացնել:

Առաջադեմ հետևողական կիբեռգործները (Advanced Persistent Threats, APT attacks), որոնց արդյունքում վնասաբեր ծրագրերն աննկատ կերպով ներթափանցում են տեղեկատվական համակարգ և աստիճանաբար տարածվելով և չբացահայտված մնալով՝ երկարաժամկետ հասանելիություն են ձեռք բերում համակարգի ներքին ռեսուրսներին՝ հաստատելով վերահսկողություն և իրականացնելով տվյալների արտահոսք:

APT գործիքները տարբերվում են մյուս կիբեռգործներից նրանով, որ՝

- զգալիորեն ավելի բարդ են՝ իրենց իրականացման տեսանկյունից,
- նպատակ ունեն տարածվել ամբողջական տեղեկատվական համակարգում,
- ունեն բազմակի վնասաբեր ազդեցություն. ներթափանցելով տեղեկատվական համակարգ, տեղակայվում են այնտեղ մինչև հայտնաբերվելը,
- գործում են փուլային սկզբունքով՝ հետևյալ հերթականությամբ.
 - հետախուզություն, որի ընթացքում ընտրվում է թիրախային զոհ և գրոհելու մարտավարություն,
 - ներթափանցում տեղեկատվական համակարգ ֆիշինգի կամ թիրախային ֆիշինգի գործածմամբ,
 - գաղտնի տեղակայում համակարգի ներսում և վնասաբեր ծրագրերի տեղադրում,
 - հրամանների և վերահսկողության (C&C) սերվերի հետ քողարկված կապի հաստատում,
 - տվյալների արտահոսքի կազմակերպում,
 - հերթական արտահոսքի հետքերի վերացում և հետագա գրոհին նախապատրաստություն:



Նկ. 9. Տվյալների արտահոսք:

«Զրոյական օրվա խոցելիությունների» շահագործումը: Որևէ համակարգի խոցելիությունը օգտագործվող ծրագրային ապահովման, սարքավորումների կամ գործընթացի՝ տեղեկատվական անվտանգության տեսանկյունից ոչ պատշաճ կամ թերի իրականացման հետևանք է, որը կարող է հմտորեն շահագործվել հաքերի կողմից:

Խոցելիությունը կոչվում է «Զրոյական օրվա» (Zero-day vulnerability), եթե այն մինչ գրողի պահը անհայտ է եղել անվտանգության պատասխանատուներին և նույնիսկ համայնքին, արտադրողին, մատակարարին և ծրագրային ապահովման թիմին, ինչը նշանակում է, որ թվային արտադրանքն ի սկզբանե նախագծվել է անվտանգային թերացումներով: Հազվագյուտ են այն դեպքերը, երբ արտադրող ձեռնարկությունները ետ են կանչում իրենց արտադրանքը՝ անգամ կիբեռանվտանգային խոցելիությունների բացահայտման պարագայում: Անվտանգային հայտնի դիպվածները, որպես կանոն, ունեն վերականգնման հստակ ընթացք և օրերի քանակ, որի վերաբերյալ առկա է հստակ վիճակագրություն նաև միջազգային փորձից, մինչդեռ՝ նշված տեսակի խոցելիությունները կարող են տևել անգամ տարիներ, մինչև որ արտադրողները տեղեկացվեն դրանց մասին և փորձեն այն վերացնել՝ այդ ընթացքում հնարավորությունն տալով հաքերներին օգտվել այդ հապաղումից: Այստեղից էլ ծագել է անվանումը՝ «գրո օր»:

Թվային արտադրանքների տեղեկատվական անվտանգության թերացումները պայմանավորված են նաև արտադրանքի ինքնարժեքի պահպանման ձգտումով: Գաղտնիք չէ, որ անվտանգային ցանկացած լուծում ֆինանսական լրացուցիչ ներդրում է պահանջում, որը նույնիսկ կարող է բազմապատկել ինքնարժեքը և ուստի՝ գինը շուկայում: Օգտագործողները՝ հասկա-

նախորեն, հնարավորություն չունեն ինչ-որ կերպ «կարկատել» այդ խոցելիությունները, ինչը որ հաճախ սեփական միջոցների հաշվին կատարվում է ձեռնարկությունների կողմից:

Օգտագործողների ստեղնաշարի վերահսկումը (Key logging), որի միջոցով հաքերները գրանցում են ստեղնաշարի սեղմումները և այդ տվյալներն օգտագործում գաղտնաբառեր կամ զգայուն տվյալներ կողոպտելու նպատակով: Նման միջամտությունները սովորաբար իրականացվում են աննկատելիորեն միացված լարերի միջոցով, կամ էլ տեղակայվում են վնասաբեր ծրագրերի ներբեռնման ճանապարհով: Օգտագործողները սովորաբար տեղյակ չեն լինում իրենց գործողությունների նման վերահսկումից: Կան գործատուներ, որոնք օրինական եղանակով օգտագործում են նման գործիքակազմ՝ աշխատակիցների գործունեությունը վերահսկելու համար: Նման դեպքերում աշխատակազմն անպայմանորեն տեղեկացվում է այդ մասին:



Նկ. 10. Օգտագործողի ստեղնաշարի վերահսկողություն:

Տեղեկատվական համակարգերի կիբեռնապառնալիքի ներքին վեկտորները (Insider threats), որոնք առաջանում են կազմակերպությունների ներսում և աշխատակազմի անդամների ոչ պատշաճ վարքագծի հետևանք են.

- անփութություն,
- անուշադրություն,
- աշխատատեղերի կիբեռանվտանգային կարգաբերումների թերացում կամ անտեսում
- տվյալների բաց և հասանելի շրջանառության թողտվություն

- տվյալների գաղտնագրային պաշտպանության լիովին կամ մասնակի անտեսում
- ներքին լրտեսություն
- չհավաստագրված ծրագրային ապահովման կամ թվային համակարգերի շահագործում
- չվերահսկվող աշխատատեղեր
- վրեժխնդրության դրսևորումներ՝ վերադասից դժգոհության պատճառով
- կարևոր տեղեկույթի արտահոսքի կազմակերպում՝ դրամական պարզևի դիմաց
- լիազորությունների և չարտոնված հասանելիության չարաշահումներ
- տեղեկատվական համակարգի ոչ լիարժեք տեսանելիության ապահովում
- իրական ժամանակում ձեռնարկության տեղեկատվական միջավայրի՝ ցանցերի, տվյալների հենքերի ծրագրակազմի, ենթակառուցվածքների և տվյալների շրջանառության վերահսկման անտեսում:

Թվարկված սպառնալիքները կարող են լինել միտումնավոր (չարամիտ գործողություններ, ինչպիսիք են լրտեսությունը, խարդախությունը կամ վնասարարությունը) կամ ոչ միտումնավոր (ֆիշինգային խաբեության զոհեր, անվտանգության ոչ պատշաճ կարգաբերումներ): Ներքին սպառնալիքները, որպես կանոն, դժվար հայտնաբերելի են, քանի որ իրագործողները՝ լինելով աշխատակազմի անդամ, ունեն կազմակերպության տեղեկատվական համակարգի լիազորված հասանելիություն՝ թեկուզ և դերակատարումների և հասանելիության մակարդակների հստակ տարանջատմամբ: Վնասարարները կարող են նաև տարանջատ թիմերի ներսում գաղտնի պայմանավորվածություններ ունենալ:

Ներքին սպառնալիքների դեմ պայքարի միջոցներից են՝

- նվազագույն արտոնության սկզբունքի ներդրումը, երբ օգտագործողներին տրամադրվում են բացառապես իրենց աշխատանքային պայմանագրով նախատեսված գործառույթներ և ռեսուրսների հասանելիություն,
- աշխատակազմի անդամների վարքագծային վերլուծությունը,
- կանոնավոր ստուգումները,
- կիրեռանվտանգային պարբերական վերապատրաստումները,

- աշխատատեղերում տվյալների հոսքերի և էլեկտրոնային հաղորդակցության վերահսկողությունը և այլն:

Կիրեռասպառնալիքների թվարկված և բացահայտվող վեկտորների ճանաչումը մեծապես նպաստում է դրանց չեզոքացմանը և հետևանքների մեղմացմանը:

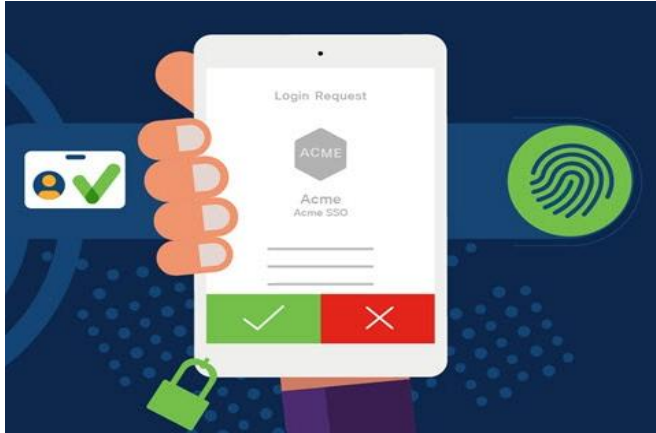
Ստորև բերված են առաջադեմ հետևողական գրոհների դեմ կիրեռանվտանգային լուծումներ և ռազմավարություններ:

Տեղեկատվական համակարգում օգտագործողների նույնականացում և հասանելիության վերահսկողություն (Identity and access management, IAM): Սա կիրեռագրոհներից պաշտպանության առաջին գիծն է, քանի որ այն կանխարգելում է թվային համակարգերի և այնտեղ շրջանառվող տվյալների չարտոնված հասանելիությունը: Վերահսկողություն իրականացնելով՝ կազմակերպությունները կարող են ապահովել, որ միայն լիազորված օգտագործողները և արտոնված հավելվածները կարողանան հասանելիություն ստանալ համակարգի ռեսուրսներին և տվյալներին: Օգտագործողների նույնականացման և ինքնության հաստատման բազմագործոն մեխանիզմների ներդրումն այլևս ժամանակի պահանջ է: Այն նաև կանխարգելում է APT գրոհների տարածումը տեղեկատվական ցանցերի ներսում:

Նույնականացումը (user identification) օգտագործողների ինքնությունը ստուգելու հիմնական գործընթացն է, որը հաճախ իրականացվում է մեկ վկայության, օրինակ՝ օգտագործողի անվան և գաղտնաբառի տրամադրման միջոցով:

Ինքնության բազմագործոն հավաստիացումը (multifactor authentication, MFA) կիրեռանվտանգային խիստ արժանագրություն է, որը պահանջում է օգտագործողի ինքնության հաստատման հետևյալ երեք տարակարգերից երկու կամ ավելի վկայության տրամադրում՝ հետևյալի հավաստիացմամբ.

- **Օգտատերն ի՞նչ գիտե միանձնյա կերպով.** գաղտնաբառ, PIN կոդ, գաղտնի հարցերի պատասխան:
- **Օգտատերն ի՞նչ ֆիզիկական իր է տիրապետում միանձնյա կերպով,** սմարթֆոն, որը գեներացնում է միանգամյա կոդ, նույնականացման քարտ (ID card) և այլն:
- **Օգտատերն ու՞մ է միանձնյա ներկայացնում:** Սա արդեն եզակի կենսաչափական հատկանիշ է՝ մատնահետքեր, դեմքի կամ ձեռքերի երակների պատկեր, դիմախաղ, ձայն, և այլն:



Նկ. 11. Ինքնության բազմագործոն հավաստիացում:

Նշենք, որ բազմագործոն հավաստիացումը պահանջում է տարբեր տարակարգերի համադրույթ: Մեկ տարակարգից մեկից ավելի գործոնների համադրումը չի համարվում բազմագործոն:

Օրինակ՝ երկգործոն հավաստիացումը (2FA) ինքնության բազմագործոն հավաստիացման ամենատարածված ձևն է, որը պահանջում է իրարից տարբերվող ճիշտ երկու գործոն, ինչպիսիք են՝ գաղտնաբառը և բջջային հեռախոսին ուղարկված միանգամյա կոդը (ի՞նչ գիտե օգտատերը և ի՞նչ ֆիզիկական սարք ունի):

Բազմագործոն հավաստիացման որոշ իրականացումներ պահանջում են երեք կամ ավելի գործոններ, ինչպիսիք են՝ գաղտնաբառերը, սարքավորումների գործարկման բանալիները, մատնահետքերի պատկերները և այլն:

Գործողությունների արտոնումը (user authorization) համակարգին, ռեսուրսին կամ ծառայությանը հասանելիություն ձեռք բերելու անվտանգային հավելալ գործընթաց է, որն անպայմանորեն մեկնարկվում է օգտագործողի, ցանցային սարքավորման կամ ծրագրային մոդուլի նույնականացման և ինքնության բազմագործոն հաստատումից հետո՝ խստորեն հետևելով լիազորությունների շրջանակին, որից դուրս այլ գործողությունների իրականացման փորձերն ընկալվում են որպես կիբեռգրոհ: Օրինակ՝ մենք կարողանում ենք օգտվել էլեկտրոնային փոստի ընձեռած հնարավորություններից՝ էլեկտրոնային նամակներ ընդունելու, ընթերցելու, դրանց արձագանքերը և անհատական նախասահմանված կարգաբերումներ իրականացնելու համար,

մինչդեռ արտոնված չենք կատարել գործողություններ՝ ուղարկված նամակների չեղարկման, հերքման կամ բովանդակության փոփոխման համար:

Տեղեկատվական համակարգերի վերջնակետերի մշտադիտարկման գործիքակազմի ներդնումը (endpoint monitoring) թույլ է տալիս կազմակերպություններին հայտնաբերել և արձագանքել վերջնակետերից եկող հավանական կիբեռսպառնալիքներին իրական ժամանակում՝ ապահովելով տեղեկատվական ցանցին միացված անխտիր բոլոր սարքերի տեսանելիություն՝ այդ սարքերում ընթացող գործառնությունների մշտադիտարկման միջոցով: Վերջնակետային սարքերից են՝ աշխատանքային պարտականություններով ամրագրված հեռախոսները, պլանշետները, հեռավար աշխատատեղերը և այլն:

Ներթափանցման թեստերի պարբերական անցկացումը (pentesting) կարևոր ռազմավարություն է APT գրոհներից պաշտպանվելու համար, քանի որ այն թույլ է տալիս մոդելավորել կազմակերպության ցանցի և համակարգերի դեմ կիբեռգրոհների սցենարներ՝ հնարավորություն ընձեռելով կազմակերպություններին հայտնաբերել և չեզոքացնել ցանցերի և համակարգերի խոցելիությունները՝ նախքան դրանք կարող են շահագործվել հաքերների կողմից:

Ցանցային երթևեկի մշտադիտարկումը (network traffic monitoring) մեկ այլ կարևոր ռազմավարություն է կիբեռգրոհներից պաշտպանվելու գործում: Այն թույլ է տալիս կազմակերպություններին հայտնաբերել շեղումներ թվային համակարգերի բնականոն գործունեությունից: Գրոհողներն օգտագործում են հմուտ մեթոդներ՝ հայտնաբերումից խուսափելու և ցանցային նորմալ երթևեկին միանալու համար, ուստի՝ ցանցային երթևեկի մշտադիտարկումը հնարավորություն է՝ բացահայտելու գրոհողի գործողություններն ընթացիկ ժամանակում:

Մեր օրերում հաքերներն օգտագործում են նաև արհեստական բանականության հնարքներ, ճշմարտանման՝ բայց հիմքում կեղծ նամակագրությունների ստեղծման ավտոմատացված գործիքներ, որոնք ի գործ են ոչ միայն վարակել համակարգիչները, այլև խաթարել պետական և ռազմավարական օբյեկտների բնականոն աշխատանքը:

Ուստի՝ արդի կիբեռգրոհների բնույթի և դրսևորումների շարունակական հետազոտումը, վերաբերելի կիբեռանվտանգային մեխանիզմների յուրացումը և ներդնումը համակարգերի տեղեկատվական անվտանգության միակ երաշխիքն է:

ԳԼՈՒԽ 3. Գաղտնագրության ներածություն

Գաղտնագրությունը, որն ընդունված է անվանել նաև ծածկագրություն, կիրեռանվտանգության կարևորագույն բաղադրիչն է, քանի որ հենց գաղտնագրային մեթոդներով է, որ իրականացվում է տվյալների քողարկումը և գաղտնիությունը: Այն նաև հնարավորություն է ընձեռում հավաստիացնել օգտագործողների և թվային սարքերի ինքնությունը և միաժամանակ կանխում է հաղորդագրությունների փոխանցման ժխտումը՝ ապահովելով, որ հաղորդագրություն ուղարկող կողմը հետագայում չկարողանա հերքել տվյալ հաղորդակցությանն իր մասնակցությունը:

Սահմանում. Գաղտնագրությունը ընթեռնելի հաղորդագրությունների վերածումն է անընթեռնելի հաղորդագրությունների, որոնց ընթերցումը վերապահված է բացառապես ապագաղտնագրման բանալուն տիրապետող անձանց: «Գաղտնագրություն» բառը ծագում է հունարեն «kryptos» (թաքնված) և «graphia» (գրություն) բառերի համադրությունից և միջազգայնորեն հայտնի է «կրիպտոգրաֆիա» անվանմամբ:

Գաղտնագրության կիրառության առաջին վկայությունը մեզ տանում է ավելի քան 3000 տարի առաջվա դարաշրջան, երբ ոմն եգիպտացի իր մի ձեռագիր աշխատության մեջ օգտագործեց մեհենագրեր (հիերոգլիֆներ): Մեհենագրերը հասկանալի էին միայն քրմերին և սույնով իրենցից ներկայացնում էին վաղ ժամանակների գաղտնագրեր:

Գաղտնագրությունը գիտություն է և արվեստ: Որպես գիտություն այն հիմնվում է մարդկության ստեղծագործող մտքի վրա, իսկ արվեստ է, քանի որ գաղտնագրերի ձևավորումը գիտելիքի հետ մեկտեղ պահանջում է հարկավոր հմտություններ բանեցնելու կարողություն:

Գաղտնագրության հիմնարար հասկացություններն են՝

- **բաց տեքստ**՝ հաղորդագրության ընթեռնելի բնօրինակ,
- **գաղտնագիր**՝ գաղտնագրված բաց տեքստ,
- **գաղտնագրում**՝ բաց տեքստը գաղտնագրի վերածման գործողություն,
- **ապագաղտնագրում**՝ գաղտնագիրը բաց տեքստի գործողություն,
- **գաղտնագրման բանալի**՝ գաղտնի տեղեկույթ, որն օգտագործվում է բաց տեքստը գաղտնագրելու կամ գաղտնագիրը ապագաղտնագրելու համար,

- **վերծանում՝** գաղտնագիրը բաց տեքստի վերածան գործողություն՝ առանց գաղտնագրային բանալու իմացության՝ ինչպես օրինակ՝ մեհենագրերի (հիերոգլիֆների) վերծանման դեպքում, երբ գիտնականները պարզապես գուշակել են դրանց ձևավորման օրինաչափությունները,
- **գաղտնագրային ալգորիթմ՝** կանոնների և քայլերի ամբողջություն, որով կատարվում է գաղտնագրումը գաղտնագրային բանալու կիրառմամբ,
- **գաղտնավերլուծություն՝** գաղտնագրերը վերլուծելու և վերծանելու գիտություն և արվեստ,
- **գաղտնագրային բանալիների ապահով պահպանում և փոխանակում՝** կողմնակի անձանց կողմից գաղտնագրային բանալիների կողոպուտի կամ հասանելիության կանխարգելման միջոցառումների համախումբ:

Գաղտնագրությունը հին պատմություն ունի: Այն, կարելի է ասել, այնքան հին է, որքան մարդկային հայտնագործությունների մեծագույն հրաշալիքներից մեկը՝ գիրն ու գրությունը: Հին պետականություն և մատենագրություն ունեցող բոլոր ժողովուրդները, գրեթե առանց բացառության, ունեցել են իրենց գաղտնագրությունը: Բացառություն չի կազմել նաև հայ ժողովուրդը՝ բազում դարերի ընթացքում ստեղծելով բազմաբովանդակ և հարուստ գաղտնագրական մատենագրություն, որոնցից զգալի մասն այժմ պահվում է Մատենադարանում: Հայ պատմաբան Ա. Գ. Աբրահամյանը 1978 թվականին հրատարակեց «Հայկական ծածկագրություն» աշխատությունը, որտեղ նա հավաքագրեց պատմական գաղտնագրերի շուրջ 400 նմուշ:

Բազմաթիվ գրքեր ու հոդվածներ են հրապարակվել տարբեր լեզուներով՝ նվիրված հունական, լատինական, արաբական և այլ ժողովուրդների գաղտնագրությանը: Գաղտնագրության վերաբերյալ վաղ ժամանակներից հայտնի են եղել որոշ գրքեր, որոնք անդառնալիորեն կորսված են և որոնց մասին ուղղակի հիշատակություններ կան պատմիչների աշխատություններում:

Գաղտնագրության վաղ ժամանակների գրքերի մեծ մասը գաղտնի էին, քանի որ բովանդակությամբ խորհրդավոր և առեղծվածային էին համարվում և որի համար այդ ժամանակաշրջանը բավական նպաստավոր էր՝ եթե հաշվի առնենք հասարակության խավարամոլության աստիճանը, որը պարտ հող է նման սնոտիապաշտության տարածման համար:

Նման մի աշխատություն՝ Թրիթեմիուսի գիրքը, հետագայում արգելվեց կաթոլիկ եկեղեցու կողմից և պահվեց այսպես կոչված «Արգելված գրապահոցում»՝ որպես սև ուժի և կախարդանքի գործիք:



Նկ. 12. Թրիթեմիուսի գիրքը:

Դիտարկենք վաղ դարաշրջանի եվրոպական գաղտնագրերի օրինակներ:

Կեսարի գաղտնագիրը (Caesar cipher) հռոմեական ժամանակներում ձևավորվում էր սկավառակի օգտագործմամբ:

Սկավառակի արտաքին շրջանի վրա դաջվում էին այբուբենի տառերը ճիշտ հերթականությամբ, իսկ ներքին շրջանի վրա՝ ինչ-որ թվով դեպի աջ տեղաշարժված հերթականությամբ: Այբուբենի վերջին տառը փոխարինվում էր առաջին տառով: **Գաղտնագրման բանալին** հենց այդ թիվն էր: Ստացողը, որն ուներ ճիշտ այդպիսի սկավառակ, հեշտությամբ ապագաղտնագրում էր գաղտնագիրը:

Կեսարի գաղտնագրին և դրա գաղտնավերլուծությանը կարելի է ծանոթանալ այցելելով հետևյալ կայքէջը^{† 3)}:

^{† 3)} <https://www.geeksforgeeks.org/ethical-hacking/caesar-cipher-in-cryptography>



Նկ. 13. Կեսարի սկավառակը:

Սքայթեյլը (Scytale): Ստեղծվել է Սպարտայում: Գաղտնագրումը կատարվում էր գլանի վրա փաթաթված ժապավենի վրա, որտեղ գաղտնի հաղորդագրության տառերը տեղադրվում էին ուղղաձիգ կերպով: Ժապավենը ետ էր փաթաթվում, նրա բաց մասերի վրա անկանոն կերպով գրվում էին կամայական տառեր, որոնց արանքում բուն գաղտնիքը չէր առանձնանում: Ժապավենը փոխանցվում էր ստացողին: Գաղտնագրման բանալին գլանի տրամագիծն էր: Ժապավենը նույն տրամագծով գլանի վրա նույն կերպ փաթաթելով, ստացողը ուղղաձիգ կերպով ընթերցում էր գաղտնագիրը: Սքայթեյլ գաղտնագիրը կարելի է ուսումնասիրել հետևյալ հղմամբ^{‡4)}:



Նկ. 14. Սքայթեյլի գլանը:

‡ 4) https://docs.neu.edu.tr/staff/fahreddin.sadikoglu/Chapter_2%20Crypto_25.pdf

Playfair գաղտնագիրը: Գաղտնագրումը կատարվում էր 5 տողերից և 5 սյուններից բաղկացած աղյուսակի միջոցով, անգլերենի i և j տառերի համար հատկացվում էր մեկ վանդակ՝ օգտվելով դրանց նմանությունից⁴⁾:

Գաղտնագրային բանալին չկրկնվող տառերից բաղկացած բառ էր, որի տառերը տեղադրվում էին աղյուսակի վերին ծախ անկյունից սկսած: Աղյուսակի մնացած վանդակները լրացվում էին այբուբենի մնացած տառերով: Գաղտնագրումը կատարվում էր հետևյալ կանոնների համաձայն:

1. Բաց տեքստը տրոհվում էր 2 տառանոց հատվածների: Եթե տառերի քանակը կենտ էր, ապա ավելացվում էր z տառը:

2. Եթե տրոհումից հետո միևնույն հատվածում երկու տառերը համընկնում էին, ապա այդ տառերի միջև ներդրվում էր x տառը:

3. Եթե երկու տառերը միևնույն սյունակում էին, ապա դրանք գաղտնագրվում էին իրենցից ներքև գտնվող տառերով:

4. Եթե երկու տառերը միևնույն տողում էին, ապա դրանք գաղտնագրվում էին իրենցից աջ գտնվող տառերով:

5. Եթե տառը տողի աջ եզրին էր, ապա գաղտնագրվում էր տվյալ տողի ամենաձախ տառով:

6. Եթե տառը աղյուսակի վերջին տողում էր, ապա գաղտնագրվում էր տվյալ սյունակի ամենավերին տառով:

7. Եթե երկտառանոց հատվածի տառերը միևնույն տողի կամ սյան վրա չէին, ապա կիրառվում էր նրանցով ձևավորվող ուղղանկյան կամ քառակուսու գագաթների փոխատեղման կանոնը՝ ինչպես որ պատկերված է Նկ. 15-ի երրորդ աղյուսակում:

Ստորև բերված նկարում գաղտնագրման բանալին MONARCHY բառն է: Խորհուրդ էր տրվում ընտրել չկրկնվող տառերով, որքան հնարավոր է երկար բառ:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Նկ. 15. Փլեյֆերի գաղտնագիրը:

Վիժենների գաղտնագիրը: Սա աղյուսակ է, որի առաջին տողում տեղադրվում են այբուբենի տառերն ըստ հերթականության, իսկ ամեն հաջորդ տողում՝ նախորդ տողում տեղադրվածից մեկ կարգով դեպի աջ հերթականությամբ: Արդյունքում՝ աղյուսակի վերջին տողում տեղադրվում են այբուբենի տառերը՝ հակառակ հերթականությամբ: Գաղտնագիրը կարելի է ուսումնասիրել այցելով հետևյալ կայքէջը^{§5)}:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Նկ. 16. Վիժենների գաղտնագիրը:

Գաղտնագրային բանալին բառ է, որը գործածվում է հետևյալ կերպ:

ա) Եթե բաց տեքստի երկարությունը հավասար է բանալի-բառի երկարությանը, ապա բաց տեքստի հերթական տառը փնտրվում է աղյուսակի տողերի, իսկ բանալի բառի հերթական տառը՝ աղյուսակի սյունակների վրա: Դրանց հատման կետի տառը գաղտնագիր-տառն է:

բ) Եթե բաց տեքստի երկարությունն ավելին է, քան բանալի-բառի երկարությունը, ապա բանալի-բառը ինքն իրեն կցագրվում է այնքան անգամ, մինչև հավասարվի կամ գերազանցի բաց տեքստի երկարությանը և գաղտնագրումն ընթանա ինչպես նկարագրված էր:

§ 5) https://docs.neu.edu.tr/staff/fahreddin.sadikoglu/Chapter_2%20Crypto_25.pdf

Օրինակ, եթե բանալի բառը «KEY» է, իսկ բաց տեքստը 10 տառանոց է, ապա բանալի-բառը կդառնա «KEYKEYKEYKEY», որի վերջին երկու նիշերը չեն մասնակցի գաղտնագրման գործողությանը: Օրինակ, պարզ տեքստի «H» տառի և բանալու «K» տառի կոորդինատների հատման կետում գտնվում է «R» տառը, որն էլ հենց «H» տառի գաղտնագիրն է: Նույն կերպ, պարզ տեքստի երկրորդ տառի գաղտնագիրը այդ տառը ներկայացնող տողի և բանալի-բառի երկրորդ տառի հատման դիրքում գտնվող տառն է, և այդպես՝ շարունակ:

Պիգպենի գաղտնագիրը: Այստեղ այբուբենի տառերը փոխարինվում էին նիշերով:

a	b	c	d	e	f	g	h	i	j
└	┐	┌	┘	□	├	┐	└	┌	┘
k	l	m	n	o	p	q	r	s	t
┐	┌	┘	└	├	┐	┌	└	┐	>
u	v	w	x	y	z				
<	^	∨	>	◀	▲				

Նկ. 17. Պիգպենի գաղտնագիրը:

Գաղտնագիրը կարելի է ուսումնասիրել այցելելով հետևյալ կայքէջը^{**6)}:

Էնիգմա (թարգմանաբար՝ «Գաղտնիք») մեքենան: Սա Վիժենների գաղտնագրի էլեկտրամեխանիկական իրագործումն էր, որն օգտագործվեց Հիտլերի կողմից՝ Երկրորդ համաշխարհային պատերազմում:



Նկ. 18. Էնիգմա մեքենան:

^{**6)} <https://prof-redden.medium.com/pigpen-cipher-6c66fe7b7524>

Գաղտնագիրը կարելի է ուսումնասիրել այցելելով հետևյալ կայքէջը^{††7)}:

Անգլիացի մաթեմատիկոս Ալան Թյուրինգի ղեկավարած գաղտնավերլուծաբանների թիմը գուշակեց Էնիգմայի կառուցման գաղտնիքը, որի արդյունքում դաշնակցային ուժերը վերահսկողության տակ առան հիտլերյան բանակի առաջխաղացումները և պայմանավորեցին նրա հետագա պարտությունը:

Հայկական պատմական գաղտնագրերի մի քանի հարյուրների հասնող քանակությունը վկայում է այն մասին, որ մեր նախնիները՝ տիրապետելով ռազմարվեստին, տիրապետել են նաև գաղտնի նամակագրության հնարքներին, որը հասկանալիորեն ռազմարվեստի անբաժանելի բաղադրիչ է: Բրիտանական թանգարանում պահպանվում է գաղտնագրեր պարունակող հայկական ձեռագրերի մատյան, որը կարելի է ուսումնասիրել այցելելով հետևյալ կայքէջը^{††8)}:

Դիտարկենք հայկական պատմական գաղտնագրերի որոշ օրինակներ՝ ձևավորված այն ժամանակների այբուբենի 36 տառերի վրա՝ Ա-ից մինչև Ք:

Փոխարինագրեր^{*9)}: Այբուբենը գրանցվում էր 4 տողերից և 9 սյուներից բաղկացած աղյուսակի մեջ: Տառերի փոխարինումը կատարվում էր տողերի միջև՝ երեք տարբերակով՝ հետևյալ սկզբունքով:

Ա	Բ	Գ	Դ	Ե	Չ	Է	Ը	Թ
Ժ	Ի	Լ	Խ	Ծ	Կ	Հ	Ձ	Ղ
Ճ	Մ	Յ	Ն	Շ	Ո	Չ	Պ	Ջ
Ռ	Ս	Վ	Տ	Ր	Յ	<u>Ու</u>	Փ	Ք

Նկ. 19. Փոխարինագրի աղյուսակը:

ա) 1-ին տողի տառերը փոխարինվում էին 2-րդ տողի տառերով, 3-րդ տողինը 4-րդ տողի տառերով և հակառակը:

^{†† 7)} <https://brilliant.org/wiki/enigma-machine>

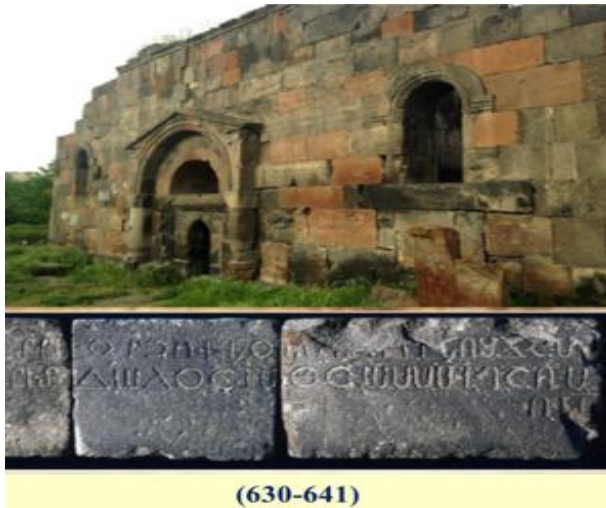
^{†† 8)} <https://archive.org/details/CatalogConybeare1913BritishMuseum/page/n57/mode/2up>

^{*9)} Ա. Գ. Աբրահամյան, Հայկական ծածկագրություն: Երևանի Համլսարանի հրատարակչություն, 1978թ.

բ) 1-ին տողի տառերը փոխարինվում էին 3-րդ տողի տառերով, 2-րդ տողինը 4-րդ տողի տառերով և հակառակը:

գ) 1-ին տողի տառերը փոխարինվում էին 4-րդ տողի տառերով, 2-րդ տողինը 3-րդ տողի տառերով և հակառակը:

Երկլեզվյան վիմագրեր^{*10)}: Սրանց հեղինակը Եզր կաթողիկոսն էր: Գաղտնագրի մի տարբերակ պահպանվում է Հայաստանի պատմության թանգարանում, մեկ այլ տարբերակ հայտնաբերվել է Ավանի Ծիրանավոր եղեկեցու բակում:



Նկ. 20. Երկլեզվյան վիմագիրը:

Կետագիր^{*10)}: Այբուբենի 36 տառերը տրոհվում էին 4 տողերի, տառերը գաղտնագրվում էին միավոր, տասնավոր, հարյուրավոր և հազարավոր տառաշարերի՝ առաջնորդող Ա, Ժ, Ճ, Ռ տառերի օգտագործմամբ, որոնց կցագրվում էին հարկավոր քանակի կետեր՝ տվյալ տառաշարքում տառի դիրքին համապատասխան:

0.	0..	0... 0....	0..... 0.....	0..... 0.....	0.....	0.....	0.....	0.....	Մնացած տառերը
Ա	Բ	Գ	Դ	Ե	Զ	Է	Ը	Թ	կարող եք լրացնել
00.	00..	00...	00....	00.....	00.....	00.....	00.....	00.....	ինքներդ:
Ժ	Ի	Լ	Խ	Ծ	Կ	Հ	Ձ		

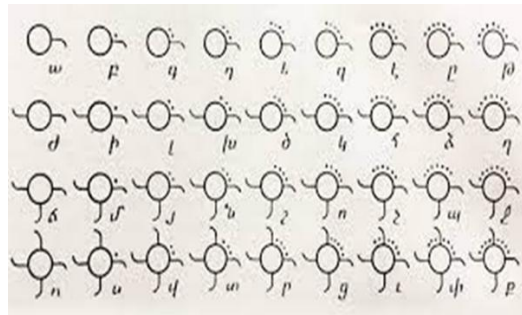
Նկ. 21. Կետագիրը:

*

^{*10)} Ա.Գ. Աբրահամյան, Հայկական ծածկագրություն: Երևանի Համլսարանի հրատարակչություն, 1978թ.

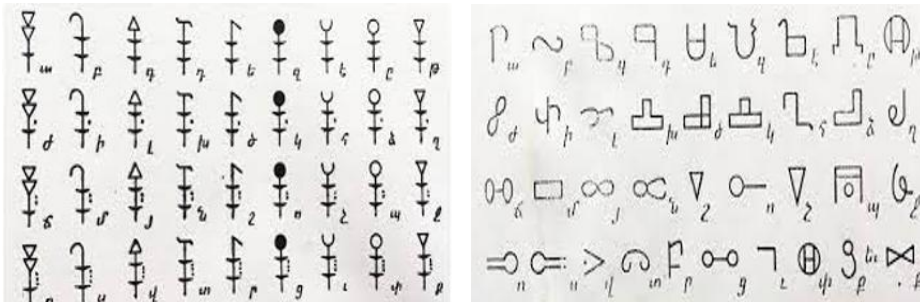
Կետագրի երկրորդ տարբերակում օգտագործվում էին բոլոր տառերը, որոնց կցագրված կետերի միջոցով գաղտնագրում էին հարկավոր տառերը: Այսպիսով՝ օրինակ՝ Ա...-ն (Ա և 3 կետ) Գ տառի գաղտնագիրն էր, իսկ Ճ.....-ն (Ճ և 6 կետ)՝ Ռ-ինը: Այսինքն՝ տվյալ առաջնորդող տառից ունեցած հեռավորությունն էր՝ արտահայտած կետերով, որոնք նաև պարուրածն էին ներկայացվում՝ մագաղաթի վրա ավելի քիչ տեղ զբաղեցնելու համար:

Օղակետագիր^{*1)}: Այբուբենի տառերը փոխարինվում էին օղակների շուրջ պատկերված կետերով և աղեղներով:



Նկ. 22. Օղակետագրի օրինակ:

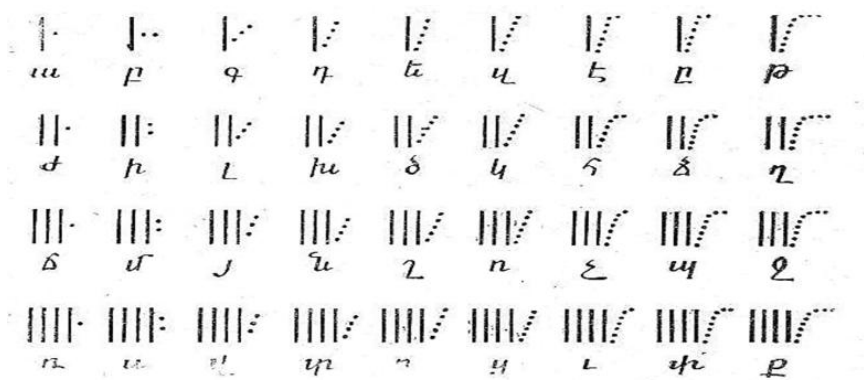
Արիեստագործական գաղտնագրեր^{*}: Այբուբենի տառերը փոխարինվում էին արիեստագործական գործիքներ ներկայացնող պատկերիկներով:



Նկ. 23. Արիեստագործական գաղտնագրեր:

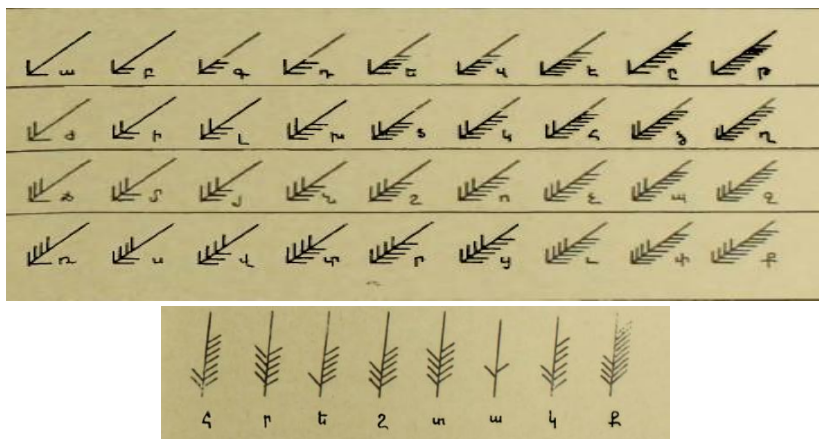
Սյունակետագիր^{**}՝ գծիկների և կետերի տեսքով: Այբուբենի 36 տառերը տրոհվում էին չորս տողերի, գծիկների քանակը որոշում էր տողի համարը, կետերի քանակը՝ համապատասխան սյունակը:

^{*1)} Ա.Գ. Աբրահամյան, Հայկական ծածկագրություն: Երևանի Համլսարանի հրատարակչություն, 1978թ.



Նկ. 24. Սյունակետագիրը:

Ճյուղագիր^{*12)}: Այբուբենի 36 տառերը տրոհվում էին չորս տողերի: Թեք գծերի ձախ կողմում նշված գծիկների քանակը որոշում էր տողի համարը, իսկ թեք գծից աջ պատկերված գծիկների քանակը՝ համապատասխան սյունակը: Ճյուղագիրը տեսքով գրավիչ գաղտնագիր էր. թեք գծերի ձախ և աջ կողմերում պատկերվող գծիկները կորագծորեն պատկերելով՝ գաղտնագիրը կարող էր ծաղկագրի տեսք ստանալ:



Նկ. 25. Ճյուղագիրը:

Գումարգիր^{*12)}: Հեղինակը Զաքարիա Վանեցին էր, 17-րդ դար: Այս գաղտնագիրն արդեն օգտագործում էր թվաբանություն:

^{*12)} Ա.Գ. Աբրահամյան, Հայկական ծածկագրություն: Երևանի Համլսարանի հրատարակչություն, 1978թ.

Այբուբենի տառերին վերագրվում էր թվական արժեք միավորների, տասնյակների, հարյուրյակների և հազարյակների տեսքով: Գաղտնագրվող տառի թվական արժեքը տրոհվում էր այնպիսի մասերի, որոնք մեկ այլ տառերի թվական արժեքների գումարն են: Գումարի տրոհումը գումարելիների և ըստ դրանց գաղտնագրի տառի բացահայտումը չափազանց բարդ էր հակառակորդի համար, քանի որ գումարի գիտենալը գումարելիների քանակի և դրանց արժեքների բացահայտման համար բավարար չէ:

0	ԱԱ	ԱԲ	ԲԲ	ԲԳ	ԳԳ	ԳԴ	ԴԴ	ԴԵ
Ա	Բ	Գ	Դ	Ե	Զ	Է	Ը	Թ
1	2	3	4	5	6	7	8	9
ԵԵ	ԺԺ	ԺԵԺԵ	ԻԻ	ԻԵԻԵ	ԼԼ	ԼԵԼԵ	ԽԽ	ԽԵԽԵ
Ժ	Ի	Լ	Խ	Ճ	Կ	Հ	Ձ	Ղ
10	20	30	40	50	60	70	80	90
ԾԾ	ՃՃ	ՃԾՃԾ	ՄՄ	ՄԾՄԾ	ՅՅ	ՅԾՅԾ	ՆՆ	ՆԾՆԾ
Ճ	Մ	Յ	Ն	Շ	Ո	Չ	Պ	Չ
100	200	300	400	500	600	700	800	900
ՇՇ	ՌՌ	ՌՇՌՇ	ՍՍ	ՍՇՍՇ	ՎՎ	ՎՇՎՇ	ՏՏ	ՏՇՏՇ
Ռ	Ս	Վ	Տ	Ր	Ց	ՌԻ	Փ	Ք
1000	2000	3000	4000	5000	6000	7000	8000	9000

Նկ. 26. Գումարգիրը:

Ակնհայտ է, որ գումարգրի գործածման դեպքում գաղտնագրի երկարությունը զգալիորեն գերազանցում է բաց տեքստի երկարությանը: Նկատենք, որ փոխարինագրերը նման ընդարձակում չունեն. գաղտնագրի տառերի կամ նիշերի քանակը հավասար է բաց տեքստի տառերի քանակին:

Ներկայիս գաղտնագրությունը մաթեմատիկայի վրա հիմնված գիտություն և արվեստ է՝ պայմանավորված համակարգչային դարաշրջանով: Այժմյան համացանցը, բանկային և հեռախոսային բոլոր տեսակի հավելվածները, լարով կամ անլար ցանցերը հեռահաղորդակցության անվտանգությունն ապահովում են գաղտնագրային ժամանակակից մեթոդներով:

ԳԼՈՒԽ 4. Տեղեկատվության գաղտնագրային պաշտպանություն

Այժմյան գաղտնագրային ալգորիթմների երկու հիմնական տեսակներն են՝

ա) **համաչափ գաղտնագրում**^{§13)} մեկ բանալիով (symmetric/secret key cryptography), երբ միևնույն բանալին օգտագործվում է և՛ գաղտնագրման, և՛ ապագաղտնագրման համար՝ ինչպես պատմական բոլոր գաղտնագրերը՝ ներառյալ հայկականները: Նշենք, որ հաղորդակցության երկու կողմերը տվյալ բանալին ձեռք են բերում նախապես՝ մինչև գաղտնի նամակագրության մեկնարկը և բացառապես առերես շփման միջոցով, որը որոշակի անհարմարություններ է ստեղծում, եթե հաշվի առնենք տարածությունը և փոխանցման հնարավորությունն ու ժամանակը:

Համաչափ գաղտնագրման ալգորիթմները երկու տեսակի են՝ **հոսքային** (stream ciphers) և **հատվածային** (block ciphers) գաղտնագրեր:

Հոսքային համաչափ ամենատարածված ալգորիթմներից են՝

- **Միանգամյա նոթատետրը** (one-time pad): Բաց տեքստի երկարության չափով գեներացվում է կամայական բիթային հաջորդականություն, որը գաղտնագրման բանալին է, իսկ գաղտնագիրը բաց տեքստի և այդ բանալու XOR գործողության արդյունքն է: Ապագաղտնագրելու համար գաղտնագիրը XOR-վում է նույն բանալու հետ: Այս ալգորիթմի «միանգամյա» անվանումը հուշում է, որ գեներացված պատահական թվերի հաջորդականությունը կիրառվում է տվյալ բաց տեքստի վրա և ոչ մի պարագայում չպետք է կիրառվի այլ բաց տեքստեր գաղտնագրելու համար՝ բանալու չփոխանցման և գաղտնիության պահպանման նպատակով:

Այս ալգորիթմը լայնորեն կիրառվում է տեղայնացված տվյալների հենքերի գաղտնագրման և անանունացման ընթացակարգերում: Անանունացման արդյունքում տվյալների սեփականատերերի անձին վերաբերող զգալուն տեղեկությունները քողարկվում են և սույնով ուղղակիորեն ընթեռնելի չեն սպասարկող անձնակազմի անդամների և ծրագրային մոդուլների համար:

¹³⁾ Schneier, B., Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley, 1996:

- **Rivest Cipher 4 (RC4)** ալգորիթմը: Գաղտնագրման բանալին այստեղ կամայականորեն գեներացված մեծաքանակ ամբողջ թվերի բազմություն է, որն օգտագործվում է հաղորդագրության բիթ-առ-բիթ գաղտնագրման, համար:

Գաղտնագիրը կարելի է ուսումնասիրել հետևյալ հղմամբ^{***14)}:

Ներկայումս RC4-ը համարվում է խոցելի և ռազմավարական նշանակության համակարգերի տեղեկատվական պաշտպանության համար՝ ոչ պիտանի:

- **Salsa20** ալգորիթմը: Այստեղ գաղտնագրման բանալին նախապես ձևափոխվում է միանգամյա օգտագործման պատահական բաղադրիչների (number used once, nonce) կիրառմամբ, այնուհետև նոր միայն օգտագործվում հաղորդագրության գաղտնագրման համար: Ալգորիթմն ավելի պաշտպանված է, քան RC4-ը և ներկայումս լայնորեն կիրառվում է բջջային հավելվածների գաղտնագրման ընթացակարգերում:

Գաղտնագիրը կարելի է ուսումնասիրել հետևյալ հղմամբ^{†††15)}

Grain-128 ալգորիթմը: Գաղտնի բանալին գաղտնագրում է բաց տեքստը բիթ-առ-բիթ՝ բիթային մեկ կամ մի քանի գործողությունների կիրառմամբ՝ ներառյալ աջ և ձախ շրջանաձև շեղումները: Լայնորեն օգտագործվում է բարձր արագագործության և էներգախնայող ցանցերում, ինչպիսին իրերի համացանցն է (IoT):

Գաղտնագիրը կարելի է ուսումնասիրել հետևյալ հղմամբ^{‡‡‡16)}

Հատվածային համաչափ ամենատարածված ալգորիթմներից են՝

- **Data Encryption Standard (DES)** ալգորիթմը: Բաց տեքստը տրոհվում է հատվածների, որոնք գաղտնագրվում են 16 ռաունդների միջոցով՝ յուրաքանչյուր ռաունդի համար գեներացնելով տվյալ ռաունդում կիրառվող գաղտնագրային բանալի: Գաղտնագրության արդյունքում բաց տեքստը ենթարկվում է փոխարինման և վերադասավորման գործողությունների:

Ալգորիթմն այժմ գործածությունից դուրս է և փոխարինվել է ավելի կատարելագործված Triple DES տարբերակով:

***14) <https://www.geeksforgeeks.org/computer-networks/rc4-encryption-algorithm>

†††15) http://www.crypto-it.net/eng/symmetric/salsa20.html#google_vignette

‡‡‡16) <https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/grain-128aead-spec-final.pdf>

- **Triple DES** ալգորիթմը: Սա նույն DES ալգորիթմն է, որը միևնույն բաց տեքստի հատվածների վրա կիրառվում է երեք անգամ՝ տարբեր բանալիների կիրառմամբ: Սա նույնպես գործածությունից դուրս է, սակայն ծառայեց հիմք՝ AES ալգորիթմի ստեղծման համար:
- **Advanced Encryption Standard (AES)** ալգորիթմը: Հաղորդագրությունը տրոհվում է 128 (10 ռաունդ), 192 (12 ռաունդ) կամ 256 (14 ռաունդ) բիթ երկարության հատվածների, որոնց վրա կիրառվում են փոխատեղման, շեղման, փոխարինման և վերադասավորման գործողություններ՝ գաղտնագրային համապատասխան բանալիների կիրառմամբ: Ալգորիթմի աշխատանքի արդյունքում բաց տեքստի կամ բանալու նույնիսկ աննշան փոփոխությունը հանգեցնում է ելքային գաղտնագրի նշանակալի փոփոխությանը:

Համաչափ բոլոր ալգորիթմները նախագծվում են մեծամասամբ Բուլյան հանրահաշվի դրույթների կիրառմամբ և միայն AES գաղտնագիրն է, որ օգտագործում է նաև մնացորդային հանրահաշվի տարրերը: Այս հանգամանքներով պայմանավորված՝ համաչափ գաղտնագրային ալգորիթմներն առանց բացառության արագագործ են և հաշվարկների տեսանկյունից քիչ ծախսատար: Նպատակահարմար են մեծածավալ տվյալների կամ իրական ժամանակում գործող թվային համակարգերում տվյալների հոսքերի գաղտնագրման համար, ինչպիսիք են, օրինակ՝ մարդկային խոսքի կամ տեսախցիկներից ստացվող տվյալների ընթացիկ գաղտնագրումը: Համաչափ ալգորիթմներում բանալիների գաղտնիության շնորհիվ դրանց երկարությունը համեմատաբար փոքր թիվ է՝ 128-256, մինչդեռ կայունությունը հատարկման դեմ՝ 2^{128} կամ 2^{256} , որը ներկայիս հաշվարկային հզորություններով իրագործելի չէ:

Համաչափ ալգորիթմների գլխավոր թերությունը գաղտնագրային բանալիների ապահով փոխանցման կամ բաշխման խիստ անհարմարությունն է. և՛ ուղարկողը, և՛ ստացողը պետք է ունենան միևնույն բանալին, և բանալին չպետք է հասանելի լինի որևէ երրորդ կողմին: Բանալին փոխանցվում է առձեռն՝ կրիչների միջոցով կամ գաղտնի կապուղու օգտագործմամբ, որը խնդրահարույց կարող է լինել՝ այդ կապուղիների տիրապետման և ապահովության տեսանկյունից: Հաղորդակցության չհերքման ապահովումը մեկ այլ խնդիր է, քանի որ միևնույն բանալու տիրապետման պարագայում անհնար է պարզել, թե կողմերից ո՞րն է ստեղծել տվյալ գաղտնագիրը:



Նկ. 22. Գաղտնագրային բանալիների առձեռն փոխանցում:

Համաչափ ալգորիթմների դեմ տիպային կիբեռգրոհները վերաբերում են գաղտնի բանալու բացահայտմանը, որի սպառնիչ հատարկումը հայտնի է որպես «ուժային» գրոհ: Նշենք, որ անկախ տեսակներից՝ գաղտնագրման ալգորիթմների ապահովությունը չի ենթադրում գաղտնագրի ապահովություն գրոհների դեպ: Ալգորիթմները պետք է նախագծված լինեն այնպես, որ գաղտնագրի կայունությունը հիմնված լինի բացառապես գաղտնագրային բանալիների ապահովության վրա:

Համաչափ ալգորիթմների դեմ գրոհների այլ տեսակների ուսումնասիրությունը թողնում ենք ընթերցողի հայեցողությանը:

բ) **Անհամաչափ գաղտնագրում**^{§§17} երկու՝ իրարից տարբեր բանալիների կիրառմամբ (հանրային բանալիով գաղտնագրում, public key cryptography): Բանալիներից մեկը՝ հանրային բանալին (public key), օգտագործվում է գաղտնագրման, իսկ մյուսը՝ փակ բանալին (private key)՝ ապագաղտնագրման համար: Ակնհայտ է, որ տարբեր են նաև գաղտնագրման և ապագաղտնագրման ալգորիթմները, և, որ հատկանշական է՝ գաղտնագրման ալգորիթմը ևս հանրային է այն իմաստով, որ ցանկացած ոք՝ այդ թվում նաև հաքերը, կարող է օգտագործել բաց բանալին և գաղտնագրման ալգորիթմը՝ գաղտնագիր ստեղծելու համար, սակայն չի կարող իր իսկ կողմից գաղտնագրված տեքստն ապագաղտնագրել, քանի որ չի տիրապետում ապագաղտնագրման բանալուն:

Այս տեսանկյունից՝ անհամաչափ գաղտնագրային ալգորիթմների դեմ կիրառվող տիպային գրոհներից ամենաեականը

§§17) Schneier, B., Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley, 1996:

- բաց բանալուց փակ բանալու հաշվարկն է և
- գաղտնագրման ալգորիթմից ապագաղտնագրման ալգորիթմի բացահայտումը



Նկ. 23. Անհամաչափ գաղտնագրային երկու իրարից տարբեր բանալիները և նրանց տարբեր գործառույթները:

Նշենք, որ չնայած այն հանգամանքին, որ հանրային և փակ բանալիներն իրենց թվային արժեքներով հավասար չեն, հիմքում նրանք մաթեմատիկորեն փոխկապակցված են, որն արտաքուստ հայտնաբերել կամ կռահել հնարավոր չէ: Այսինքն՝ ունենալով ալգորիթմի բաց բանալին, դրանից հաշվարկել փակ բանալին անհնար է: Հենց այս իրողության վրա էլ հիմնված է անհամաչափ ալգորիթմների կայունությունը և գործնական կիրառելիությունը:

Անհամաչափ գաղտնագրային ալգորիթմների դեմ տիպային այլ գրոհների ուսումնասիրությունը թողնում ենք ընթերցողի հայեցողությանը:

Այսպիսով՝ անհամաչափ գաղտնագրային ալգորիթմ գործարկելու համար օգտագործողը՝

- գեներացնում է բանալիների զույգը,
- բաց կապուղով, օրինակ՝ համացանցով, փոխանցում է բաց բանալին հաղորդագրությունն ուղարկող կողմին, որն այդ բանալու և գաղտնագրման ալգորիթմի միջոցով գաղտնագրում է բաց տեքստը և դարձյալ բաց կապուղով փոխանցում ստացողին: Ստացողը ապագաղտնագրում է գաղտնագիրը իր մոտ պահվող և ոչ ոքի չփոխանցվող փակ բանալիով:

Անհամաչափ ալգորիթմներում գաղտնագրման բանալու մի մասնաբաժնի հանրայնորեն հասանելիությունը սկզբնապատճառ դարձավ, որ համաչափ ալգորիթմներում օգտագործվող պարզ գործողությունները փոխարինվեցին բարդ հանրահաշվական գործողություններով՝ փակ բանալու հատարկումն անհնարին դարձնելու նպատակով: Արդյունքում՝ անհամաչափ գաղտնագրերի մի մասի բանալիները 100 և ավելի նիշ ունեցող ամբողջ թվեր են և համաչափ ալգորիթմների հետ համեմատած՝ գաղտնագրման ու ապագաղտնագրման ընթացակարգերը համեմատաբար դանդաղագործ են:

Անհամաչափ գաղտնագրերի ի հայտ գալով ներմուծվեցին այսպես կոչված՝ «**միակողմանի ֆունկցիաների**» և «**գաղտնիքով միակողմանի ֆունկցիաների**» գաղափարները որպես համապատասխանաբար՝ չհակադարձվող և գաղտնիքի իմացությամբ հակադարձվող ֆունկցիաներ: Սա նշանակում է, որ տրված $y = f(x)$ ֆունկցիայի համար տրված x -ի, այսինքն՝ բաց տեքստի համար, համապատասխան y -ի հաշվարկը հնարավոր է, սակայն՝ միակողմանի ֆունկցիայի համար տրված y -ից x -ի ստանալն անհնար է, իսկ գաղտնիքով միակողմանի ֆունկցիայի դեպքում y -ից x -ը կարելի է ստանալ միայն գաղտնի բանալու իմացության դեպքում:

Անհամաչափ գաղտնագրերը, որպես ժամանակի պահանջ առաջ եկան 1975 թվականին, երբ ամերիկացի մաթեմատիկոսներ Վ. Դիֆֆին և Մ. Հելլմանը հայտնագործեցին մի եղանակ, որի միջոցով հաղորդակցության երկու կողմերը համացանցով փոխադարձ հաշվարկների արդյունքում ստացան գաղտնագրային բանալու միևնույն թվական արժեքը՝ սույնով լուծելով բանալու ապահով փոխանցման խնդիրը: Անհամաչափ գաղտնագրերի ի հայտ գալով հնարավորություն ստեղծվեց էլեկտրոնային փաստաթղթերը ստորագրել էլեկտրոնա-թվային եղանակով՝ փոխարինելով ձեռագիր ստորագրությանը: Անձը փաստաթուղթը ստորագրում է (գաղտնագրում է) իր մոտ պահվող և ոչ ոքի չփոխանցվող փակ բանալիով, իսկ ստորագրությունը վավերացվում է (ապագաղտնագրվում է) անձի բաց բանալու միջոցով: «Մարդը արանքում» գրոհից պաշտպանվելու համար հանրային բանալիները հավաստագրվում են տվյալ պետության մեջ գործող այսպես կոչված «Հավաստագրման մարմնի» (certification authority, CA) կողմից: Նշենք, որ համաչափ գաղտնագրերը էլեկտրոնա-թվային ստորագրության ստեղծման համար գործածական չեն, քանի որ համաչափ ալգորիթմներում չկան հան-

րային հասանելիության բանալիներ թվային ստորագրության վավերացման համար: Համաչափ ալգորիթմներում բոլոր բանալիները գաղտնի են:

Անհամաչափ (հանրային բանալիով) գաղտնագրային ալգորիթմները և հաղորդակարգերը հիմնված են 100-ից ավելի թվանշանով պարզ թվերի օգտագործման վրա և հիմքում անպայմանորեն ունեն գաղտնիքով միակողմանի ինչ-որ ֆունկցիա:

Դիտարկենք անհամաչափ գաղտնագրային ամենատարածված ալգորիթմները:

Գաղտնագրային բանալիների փոխանակման ԴիՖֆի-Հելլմանի ալգորիթմը ^{****18)}: Սա արձանագրություն է, որը թույլ է տալիս երկու կողմերին հեռավար կերպով հաշվարկել գաղտնի բանալին անապահով հանրային կապուղու օգտագործմամբ: Հաղորդակցական մի քանի սեանսների ընթացքում կողմերը փոխանակում են որոշակի հանրային պարամետրեր, որոնց իմացությունը, սակայն հնարավորություն չի ընձեռում գաղտնալսողին գուշակել որոշումարար գաղտնի բանալին:

RSA (Rivest-Shamir-Adleman) գաղտնագրային ալգորիթմը հեղինակների կողմից ներկայացվեց 1977 թվականին: Ալգորիթմը հիմնված է առանձնակի խոշոր պարզ թվերի օգտագործման վրա, որտեղ բաց բանալին պարզ թվերի արտադրյալն է, իսկ փակ բանալին՝ այդ արտադրյալի հետ փոխկապակցված հաշվարկի արդյունք: Ալգորիթմի կայունությունը հիմնված է բաց բանալու տրոհման և գաղտնի պարզ արտադրիչների զույգի հայտնաբերման բարդության վրա: Ներկայումս RSA ալգորիթմի գաղտնագրային բանալիների երկարությունը առնվազն 2048 բիթ է, որը մոտ ապագայում կարող է բավարար չլինել՝ պահանջվող գաղտնակայունության ապահովման համար: Ալգորիթմը նաև աշխատատար է՝ առանձնակի խոշոր մեծության պարզ թվերի գեներացման, աստիճան բարձրացման և բաժանման ծախսատար գործողությունների օգտագործման պատճառով:

Գաղտնագրում էլիպտիկ կորերի կիրառմամբ (Elliptic curve cryptography, ECC): Այս ալգորիթմի հեղինակներն են Նիլ Կոբլիցը և Վիկտոր Միլլերը, որոնք 1985 թվականին ներկայացրեցին անհամաչափ գաղտնագրման ալգորիթմ բարձրագույն հանրահաշվի մեթոդների կիրառմամբ: Ալգորիթմի մեկնաբանությունը սույն դասագրքի շրջանակներից դուրս է:

^{****18)} Schneier, B., Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley, 1996:

Նշենք, որ ECC ալգորիթմը՝ ի տարբերություն RSA-ի, էներգախնայող է՝ անհամեմատ նվազ ծախսատար հաշվարկների շնորհիվ, ուստի և լայնորեն կիրառվում է սահմանափակ հաշվարկային հզորություններով օժտված բջջային և ցանցային սարքավորումներում: Համեմատության համար նշենք, որ ECC 512 բիթանոց բանալին նույնքան անվտանգություն է ապահովում, որքան 256 բիթանոց AES և 15360 բիթանոց RSA ալգորիթմները՝ առանձին-առանձին վերցրած:

Էլեկտրոնա-թվային ստորագրության (Digital signature) ալգորիթմը, որը կառուցվում է այսպես կոչված՝ «հեշավորման» (hashing) ալգորիթմների ներգրավմամբ: Հեշ արժեքը խառնատող է, որը հաշվարկվում է կամայական երկարության հաղորդագրության վրա, բայց արտադրում է միևնույն երկարության արժեք: Հեշավորման ալգորիթմների սերունդներից են՝

- md4 (message digest)
- md5
- SHA1 (secure hashing algorithm)
- SHA256 և SHA385:

Ներկայումս SHA 256-ից ցածր հեշավորման ալգորիթմների գործածությունն արգելված է՝ դրանց խոցելիության պատճառով:

Նշենք, որ ի տարբերություն գաղտնագրերի, հեշ արժեքները հակադարձելի չեն, այսինքն՝ տվյալ հաղորդագրությունից կարելի է ստանալ ինչ-որ հեշ, բայց տրված հեշից այդ նույն հաղորդագրությունը ստանալն անհնար է կամ՝ չափազանց բարդ և աշխատատար: Այսինքն՝ հեշավորման ֆունկցիաները միակողմանի ֆունկցիաներ են:

Հաղորդագրության հեշ արժեքը ծառայում է որպես հաղորդագրության ամբողջականության հավաստիացման միջոց: Հաղորդագրության մեջ անգամ չնչին քանակի բիթերի փոփոխումը հանգեցնում է արդյունարար հեշի զգալի՝ առնվազն 50 տոկոսի, փոփոխության:

Օրինակ՝

A տառի հեշ արժեքը SHA256-ի միջոցով հաշվարկելիս ստանում ենք հետևյալ խառնատողը՝

559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd
իսկ C տառի դեպքում՝

12f37a8a84034d3e623d726fe10e5031f4df997ac13f4d5571b5a90c41fb84fe

Տարբերությունն ակնառու է, և այլն էլ այն պարագայում, երբ այդ տառերի երկուական ASCII կոդերը տարբերվում են ընդամենը մեկ՝ նախաձերձին բիթով:

ASCII(A)=01000001

ASCII(C)=01000011

Այցելելով հետևյալ առցանց հարթակ՝ կարելի է փորձարկել տարբեր երկարության և բովանդակության հաղորդագրությունների հեշ արժեքի հաշվարկը՝ SHA256 միջոցով^{††††19)}:

Ինչպե՞ս է ստեղծվում էլեկտրոնա-թվային ստորագրությունը:

Էլեկտրոնային փաստաթղթի սեփականատերը հաշվարկում և գաղտնագրում է փաստաթղթի հեշ արժեքը իր փակ բանալիով, իսկ փաստաթուղթը ստացողը վավերացնում, այսինքն՝ ստուգում է ստորագրությունը համապատասխան հանրային բանալիով: Նշենք, որ փաստաթղթերի թվայնորեն ստորագրումը չի ենթադրում դրանց բովանդակության գաղտնագրում: Փաստաթղթերն ընթեռնելի են, ինչպես օրինակ՝ լիազորագրերը, կտակները, ֆինանսական փաստաթղթերը և այլն: Գաղտնագրվում են այդ փաստաթղթերի հեշ արժեքները՝ փաստաթղթերի չկեղծված լինելը հավաստիացնելու համար:

Էլեկտրոնա-թվային ստորագրությունների կիրառմամբ այժմ հավաստիացվում են հեռահաղորդակցական արձանագրություններ և ծրագրակազմեր՝ վկայելով դրանց ամբողջականությունը և անփոփոխելիությունը ներդրումից հետո:

Գաղտնագրային բանալիների կառավարումը ինչպես համաչափ ալգորիթմների, այնպես էլ անհամաչափ ալգորիթմների օգտագործման պարագայում չափազանց կարևոր է, քանի որ հակառակորդի կողմից բացահայտված բանալիները լիովին կխաթարեն գաղտնագրված տվյալների անվտանգությունը՝ հանգեցնելով տվյալների չարտոնված հասանելիության և արտահոսքի: Բանալիների կառավարման հիմնական գործառույթներն են՝

- **Բանալիների ստեղծումը:** Բանալիները պետք է ստեղծվեն պատահական թվերի հիման վրա և հարկավոր մեծությամբ՝ պահանջվող գաղտնակայունության երաշխավորման համար:
- **Բանալիների պահպանումը,** որը պետք է իրականացվի տեղեկատվական համակարգի մեկուսացված միջավայրում՝ երաշխավորելով

^{††††19)} <https://emn178.github.io/online-tools/sha256.html>

դրանց ուղղակիորեն անհասանելիությունը սպասարկող անձնակազմի, կառավարիչների և նույնիսկ ծրագրավորողների համար:

- **Բանալիների բաշխումը**, որը պետք է անվտանգ կերպով իրականացվի՝ բացառապես լիազորված օգտագործողների միջև:
- **Բանալիների օգտագործումը**, որը պետք է խստորեն վերահսկելի լինի՝ դրանք գաղտնագրման և ապագաղտնագրման նպատակով կիրառելու ողջ ընթացքում՝ չլիազորված օգտագործումների բացառման նպատակով:
- **Բանալիների ռոտացիան**, որը գործող բանալիների պարբերական թարմացումն է՝ բանալիների արտահոսքի արդյունքում տվյալների արտահոսքի չափի նվազեցման նպատակով:
- **Բանալիների ոչնչացումը**, որը դրանց անվտանգ հեռացումն է ակտիվ հաշվողական միջավայրերից, երբ դրանք այլևս անհրաժեշտ չեն: Նպատակը՝ արխիվացված տվյալների չլիազորված ապագաղտնագրման կանխարգելումն է:
- **Հանրային բանալու հրապարակման մեխանիզմը**: Տվյալ երկրի հավաստագրման մարմինը օգտատերերին տրամադրում է հանրային բանալու հավաստագիր (public key certificate), որը շաղկապում է հանրային բանալին որոշակի ինքնության՝ անձի, սերվերի կամ կազմակերպության հետ՝ կանխարգելելու համար համացանցում շրջանառվող հանրային բանալիները այլոց կեղծ բանալիներով փոխարինելու փորձերը: Նման կիբեռգրոհներն անվանվում են նաև «այլոց անձնավորում»:
- **Բանալիների անվտանգ պահպանումը**, որը կարելի է իրականացնել հետևյալ միջավայրերում և եղանակներով.
 - գաղտնագրային բանալիների անվտանգ ստեղծման, պահպանման և կառավարման սարքերում,
 - ամպային բանալիների կառավարման համակարգերում,
 - գաղտնագրված ֆայլային համակարգերում և բանալիների պահոցներում՝ գաղտնագրված տեսքով,
 - գաղտնագրային բանալիների պահպանման համար տեղեկատվական համակարգերում նախատեսված մեկուսացված տիրույթներում՝ դրանց հասանելիության խիստ վերահսկողության և հասանելիությունը բացառապես լիազորված անձնակազմին և համակարգերին վերապահման նպատակով:

- **Գաղտնագրային բանալու փոխանցումը «գրոյական գիտելիքի ապացույցի հիման վրա»:** Սա մեթոդ է (zero-knowledge proof), որտեղ հաղորդակցության մեկ կողմը (ապացուցողը) մաթեմատիկորեն համոզում է մյուս կողմին (ստուգողին), որ տրված պնդումը ճշմարիտ է՝ առանց որևէ այլ լրացուցիչ տեղեկություն բացահայտելու:

Թվարկված բոլոր ալգորիթմները՝ համաչափ և անհամաչափ, ռազմավարական նշանակություն ունեն, քանի որ դրանք դիմակայում են առաջադեմ կիբեռգրոհներին և ապահովում թվային համակարգերի տեղեկատվական անվտանգությունը: Գաղտնագրային այս կամ այն ալգորիթմի և համապատասխանաբար՝ գաղտնագրային բանալիների չափի ընտրությունը կատարվում է դրանց կիրառության միջավայրին հատուկ կիբեռռիսկերի վերլուծության և գնահատման հիման վրա:

Պատահական թվերի գեներատորների կարևորությունը գաղտնագրային ալգորիթմներում: Անկախ գաղտնագրային ալգորիթմի տեսակից՝ համաչափ կամ անհամաչափ, հոսքային կամ հատվածային, գաղտնագրման բանալիներն ընտրվում են պատահական թվերի գեներատորներից արտադրված թվային հաջորդականություններից: Այդ իսկ պատճառով՝ չափազանց կարևոր է, որպեսզի այդ հաջորդականությունները լինեն անկանխատեսելի կառուցվածքի և բանալիների հատարկման համար պահանջեն հաշվողական ռեսուրսների և ժամանակի մեծ ծախս:

Ծրագրավորման բոլոր միջավայրերն ունեն ներկառուցված պատահական թվերի գեներատոր, որը հաջողությամբ օգտագործվում է այսպես կոչված՝ «սինթետիկ»՝ իրական տվյալները նմանակող թվային արժեքներ ներմուծելու համար: Այնուամենայնիվ, որքան էլ դրանք նպատակահարմար են ծրագրերի մշակման փուլում, այնուամենայնիվ, խստիվ արգելվում են ծրագրերի արդյունաբերական շահագործման փուլում, քանի որ արտաքուստ պատահական տեսք ունեցող թվերի հաջորդականությունը դետերմինիստական է, այսինքն՝ որոշակի, որի պատճառով դասվում են այսպես կոչված՝ «կեղծ պատահական թվերի գեներատորների» (Pseudo-random numbers generators, PRNGs) շարքին: Իրապես պատահական թվերի գեներատորները (True random numbers generators, TRNGs) նախագծվում են բնական և անկանխատեսելի աղբյուրներից ստացվող տվյալների հավաքագրման հիման վրա, ինչպիսիք են օրինակ՝ մթնոլորտային երևույթների կամ տեսաշարերի թվայնացված տվյալները, որոնք բնույթով իսկապես անկանխատեսելի են:

Գաղտնագրության մեջ կիրառելիության տեսանկյունից՝

- **Կեզձ պատահական** են այն հաջորդականությունները, որոնք դիմակայում են հասանելի բոլոր վիճակագրական թեստերին, մինչդեռ՝
- **իրապես պատահական** են այն հաջորդականությունները, որոնք անկանխատեսելի են, այսինքն՝ նույնիսկ հաջորդականությունը ստեղծող ալգորիթմի և նախորդող բիթերի իմացության պարագայում հաշվարկային առումով անհնար է կանխատեսել հաջորդող բիթերը: Բացի այդ՝
- **ստացված հաջորդականությունը հնարավոր չպետք է լինի վերարտադրել**, որը նշանակում է, որ գեներատորի երկու անգամ ճիշտ նույն մուտքային տվյալներով, գործարկելու պարագայում պետք է արտադրվեն իրարից խստորեն տարբերվող հաջորդականություններ:

Թվարկված երեք հատկություններին բավարարող գեներատորներն են, որ կարող են կիրառելի լինել գաղտնագրային բանալիների, ցանցային միանգամյա անցաբառերի և գաղտնագրային իրապես պատահական այլ բաղադրիչների ապահով գեներացման համար:

ԳԼՈՒԽ 5. Համակարգչային ցանցերի կիբեռանվտանգություն

Համակարգչային ցանցը փոխկապակցված թվային սարքերի՝ համակարգիչների, սերվերների, տպիչների, սմարթֆոնների և այլ թվային սարքավորումների հավաքածու է՝ տվյալների պահպանման, մշակման և փոխանցման հնարավորությամբ: Համակարգչային ցանցերի միջոցով են իրականացվում էլեկտրոնային փոստի սպասարկումը, նիշքերի փոխանցումը, առցանց խաղերը, հարկավոր տեղեկատվության որոնումը և այլն: Ցանցային սարքավորումները միմյանց հետ կապակցվում են ինչպես լարային (օպտիկական մանրաթելեր կամ Ethernet մալուխներ), այնպես էլ՝ անլար (Wi-Fi, Bluetooth) կապուղիների միջոցով:

Համակարգչային ցանցերին բնորոշ կիբեռսպառնալիքների ճանաչման և դրանց դեմ պաշտպանության մեխանիզմների ըմբռնման նպատակով համառոտ անդրադառնանք համակարգչային ցանցերի ճարտարապետությանը և վերաբերելի ֆիզիկական ու տրամաբանական միավորներին:

- **Հանգույց** (node): Սա ցանկացած սարք է, որը կարող է ստեղծել, ստանալ կամ հաղորդել տվյալներ ցանցի ներսում: Հանգույցի օրինակ են՝ համակարգիչները, տպիչները, մոդեմները և այլն:
- **Արձանագրություններ** (protocols), որոնք ցանցային հանգույցների միջև հաղորդակցության կանոնների և ընթացակարգերի համախումբ են՝ ձևավորված միջազգայնորեն ընդունված ստանդարտների հիման վրա, որոնցով սահմանվում են նաև փոխանակվող հաղորդագրությունների տարատեսակ ձևաչափերը: Արձանագրությունների շնորհիվ ցանցային սարքավորումները «ճանաչում և հասկանում են» միմյանց: Օրինակ, եթե երկու համակարգիչ փոխանակվելու են տվյալներով, ապա դրանցից մեկն ուղարկում է տեխնիկական իմաստով և ձևակերպմամբ՝ «պատրաստ եմ հաղորդել» ազդանշան, իսկ մյուսը՝ տեխնիկական իմաստով և ձևակերպմամբ՝ «պատրաստ եմ ընդունել» պատասխան:
- **IP հասցե** (Internet Protocol address), կամ որ նույնն է՝ ցանցային արձանագրության հասցե: Սա եզակի թվային պիտակ է, որը վերագրվում է համակարգչային ցանցին միացվող յուրաքանչյուր սարքի (հանգույցի), որը հաղորդակցության նպատակով օգտագործում է ցանցային արձանագրություն: Ցանցային սարքերը նույնականացվում են այդ հասցեների միջոցով՝ նման ավանդական փոստային

հասցեներին, որոնց միջոցով նամակներն ուղարկվում են իրենց հասցեատերերին:

Համակարգչային ցանցերը դասակարգվում են ըստ իրենց չափսերի և աշխարհագրական ընդգրկման՝ հետևյալ կերպ:

- **Անձնական տարածքի ցանցեր** (Personal Area Network, PAN): Այս ցանցը սպասարկում է մեկ անհատի: Օրինակներից են՝ սմարթֆոնները, պլանշետները և նոթբուքերը, որոնք նիշքերի փոխանակման նպատակով կապվում են միմյանց հետ Bluetooth-ի միջոցով:
- **Տեղայնացված ցանցեր** (Local Area Network, LAN): Այս ցանցը կապակցում է համակարգիչներն ու սարքերը սահմանափակ ֆիզիկական տարածքում, օրինակ՝ գրասենյակներում, տանը կամ կրթական հաստատության մասնաշենքերում: Այսպիսի ցանցերը, որպես կանոն, արագագործ են, տեղեկատվական անվտանգության տեսանկյունից համեմատաբար ապահով և ավանդականորեն գտնվում են մասնավոր տիրապետության տակ:
- **Լայն տարածքի ցանցեր** (Wide Area Network, WAN): Այս ցանցը կապակցում է համակարգիչները և ավելի փոքրաձավալ ցանցերը լայն աշխարհագրական տիրույթներում՝ փոքրաթիվ քաղաքներից մինչև ամբողջ աշխարհը՝ կապակցելով միլիարդավոր թվային սարքեր: Տեղայնացված ցանցերի հետ համեմատած՝ WAN ցանցերը համեմատաբար դանդաղագործ են և թանկ:

Համակարգչային ցանցերը նախագծվում են երկու հիմնական ճարտարապետությամբ²⁰⁾, այն է՝

- **Հաճախորդ-սերվեր** (Client-Server) ցանցեր: Այս ցանցի հաճախորդ հանգույցները հարցում են ուղարկում սերվերներին՝ վերջիններիս կողմից հիշողության, տվյալների և վերաբերելի այլ ռեսուրսների հայցման համար: Օրինակ՝ երբ մուտք եք գործում ֆիլմերի առցանց հարթակ, ձեր համակարգիչն ընկալվում է որպես հաճախորդ-հան-

²⁰⁾ J. F. Kurose & K. W. Ross, Computer Networking: A Top-Down Approach (8th ed.). Pearson, 2021.

W. Stallings, Network Security Essentials: Applications and Standards (6th ed.). Pearson, 2017.

C. Sanders, Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems (3rd ed.). No Starch Press, 2017.

գույց, իսկ հարթակի տվյալները տրամադրող համակարգիչը հանդես է գալիս որպես սպասարկող կողմ՝ սերվեր:

- **Հավասարազոր (Peer-to-Peer, P2P)** ցանցեր: Այս ցանցում բոլոր թվային սարքերն ունեն միանման արտոնություններ: Ցանցը չունի կենտրոնական սերվեր, յուրաքանչյուր սարք կարող է հանդես գալ ինչպես հաճախորդի, այնպես էլ սերվերի դերում՝ ապահովելով ռեսուրսների համատեղ հասանելիություն (sharing):

Համակարգչային ցանցերի կազմակերպման և ճարտարապետության վերաբերյալ համապարփակ գիտելիք կարելի է ստանալ՝ այցելելով հետևյալ կայքէջերը^{§§§§21)}:

Համակարգչային ցանցերը ևս ենթակա են կիբեռգրոհների՝ կազմակերպության տեղեկատվական ցանցին չարտոնված հասանելիության, դրա արդյունքում տվյալների կոդուպուտի և վնասաբեր այլ գործողությունների իրականացման նպատակով:

Ցանցային գրոհները դասակարգվում են հետևյալ հիմնական տարակարգերի:

- **Պասիվ և ակտիվ գրոհներ:** Պասիվ գրոհի դեպքում հաքերը մուտք է գործում ցանց՝ ցանցային երթևեկի (network traffic) մշտադիտարկման, գաղտնալսման կամ կարևոր տեղեկատվության կոդուպուտման նպատակով: Այս դեպքում որևէ միջամտություն կամ տվյալների նենգափոխում չի կատարվում: Երթևեկի կամ տվյալների միջամտություն կատարվում է ակտիվ գրոհների դեպքում: Ակտիվ գրոհների օրինակներից են՝ տվյալների հեռացումը, նենգափոխումը, փրկագնի պահանջի նպատակով տվյալների չարտոնված գաղտնագրումը և այլն:
- **Արտաքին և ներքին սպառնալիքներ:** Արտաքին սպառնալիքներն այն կիբեռգրոհներն են, որոնք կատարվում են տեղեկատվական ցանցի պարագծից դուրս հաքերների կողմից: Ներքին կիբեռսպառնալիքները կապված են կազմակերպության գործող կամ նախկին աշխատակիցների հետ, ներառյալ՝ մատակարարները և գործընկեր-

§§§§ 21) https://docs.oracle.com/cd/E53394_01/pdf/E54755.pdf

<https://www.geeksforgeeks.org/computer-networks/basics-computer-networking>

<https://www.wikihow.com/Learn-Computer-Networking>

<https://www.juniper.net/us/en/research-topics/what-is-ids-ips.html>

<https://nulab.com/learn/software-development/network-diagrams-guide>

ները, որոնք չվերահսկվող հասանելիություն ունեն ցանցին և չարամտորեն շահագործում են այդ հասանելիությունը:

2023 թվականին Tesla ընկերության երկու նախկին աշխատակիցներ կողոպտեցին գաղտնի տվյալներ՝ պատճառելով ավելի քան 75,000 հաճախորդների անձնական տեղեկատվության արտահոսք: Նման միջադեպերը ցույց են տալիս, որ կազմակերպությունները պետք է արգելափակեն նախկին աշխատակիցների հասանելիությունը տեղեկատվական ցանցին և մշտապես վերահսկեն համակարգում նույնիսկ լիազորված աշխատակիցների կամ օգտագործողների վարքագիծը:

2017 թվականին հաքերները հայտնաբերեցին խոցելիություն Equifax-ի կայքի «Apache Struts» վեբ հավելվածի բաղադրիչում, որի արդյունքում հասանելիություն ստացան մոտ 147 միլիոն սպառողների անձնական տվյալներին՝ ներառյալ նրանց վարկային քարտերի տվյալները: Հատկանշականն այն է, որ այդ խոցելիության մասին ընկերության համապատասխան աշխատակազմն ավելի վաղ տեղեկացվել էր և որի վերացման նպատակով անվտանգային թարմացում էր մշակվել, սակայն ընկերությունը հանցավոր անփութության պատճառով չէր տեղադրել այն: Նման դեպքերը վկայում են այն մասին, որ կիբեռգրոհները հաճախ հաջողվում են ոչ թե «գրոյական օրվա» խոցելիությունների առկայության, այլ անվտանգային խոցելիության անտեսման և ոչ պատշաճ արձագանքման պատճառով: Այս տեսանկյունից՝ համակարգերի և ծրագրային ապահովման մշտադիտարկումը՝ խոցելիությունների ժամանակին հայտնաբերման և կառավարման (vulnerability management) նպատակով, կիբեռանվտանգության ոլորտի կարևորագույն ռազմավարություն է: Ապագային միտված (proactive) շարունակական թարմացումները թույլ են տալիս կանխարգելել կիբեռգրոհները՝ նախքան դրանք տեղի կունենան:

Ցանցային ճարտարապետության յոթ մակարդակների իմացությունը, այն է՝

- կիրառական (Application Layer),
- ներկայացման (Presentation Layer),
- հաղորդակցության սեսիայի (Session Layer),
- տրանսպորտային (Transport Layer),
- ցանցային (Network Layer),
- տվյալների կապակցման (Data Link Layer) և
- ֆիզիկական,

մեծապես նպաստում է կիբեռնապառնալիքների կանխատեսման և չեզոքացման գործընթացների բարեհաջող իրականացմանը:

Օրինակ՝ ցանցային ճարտարապետության կիրառական մակարդակում, որտեղ գործում են օգտատերերի հետ անմիջականորեն փոխազդեցության արձանագրությունները՝ HTTP կամ HTTPS, FTP, DNS, SMTP, տիպային կիբեռգրոհներից են՝

- վնասաբեր ծրագրերի ներբեռնումը,
- ֆիշինգը,
- տվյալների չգաղտնագրված փոխանցմամբ պայմանավորված դրանց չլիազորված հասանելիությունը,
- ավանդական DOS, DDOS,
- SQL injection և
- փրկագնի պահանջի նպատակով (ransomware) գաղտնագրման գրոհները:

Ներկայացման մակարդակում տիպային գրոհներն առավելապես դրսևորվում են ֆիշինգի տեսքով:

Հաղորդակցության սեսիայի մակարդակում տիպային են սեսիայի կոդպուտի գրոհները:

Տրանսպորտային մակարդակում, որն ապահովում է տվյալների փոխանցումը հավելվածների միջև TCP կամ UDP արձանագրությունների հիման վրա, տիպային կիբեռգրոհներից են՝

- ցանցային բաց միացքների (port) սքանավորումը (port scanning) և
- սեսիայի առևանգումը (session hijacking)

Ցանցային մակարդակում, որտեղ տեղի է ունենում տվյալների փաթեթների երթուղավորումը տարբեր ցանցերի միջև IP և CMP արձանագրությունների հիման վրա, տիպային կիբեռգրոհներից են՝

- IP հասցեների կեղծումը (IP Spoofing) և
- «մարդը՝ արանքում» գրոհները:

Ցանցային հասանելիության մակարդակում տեղի է ունենում հաղորդագրությունների բիթ-առ-բիթ փոխանցում Ethernet կամ Wi-Fi միջավայրերում: Այստեղ տիպային կիբեռգրոհներից են՝

- ցանցային երթևեկի (network traffic) զննումը/գաղտնալսումը (Packet Sniffing) և
- MAC հասցեների կեղծումը (MAC spoofing):

Այս գրոհները տիպային են նաև տվյալների կապակցման մակարդակում:

Ֆիզիկական մակարդակի տիպային գրոհներից են ցանցային փաթեթների զննման/գաղտնալսման գրոհները, որոնց միջոցով տեղի է ունենում այդ փաթեթների բովանդակության վերլուծություն՝ զգայուն տեղեկատվության կորզման նպատակով: Կորզված ֆինանսական հաշվետվությունները, չգաղտնագրված տվյալները և այլ զգայուն տեղեկությոը հաքերների կողմից հետագայում կարող են օգտագործվել վնասաբեր գործողություններում: Այս տեսակ գրոհներն առավել տարածված են անապահով հանրային Wi-Fi ցանցերում:

Համակարգչային ցանցերին հատուկ գրոհների տարատեսակ են արդեն իսկ վարկաբեկված ցանցերում հանգույցից-հանգույց, տեղամասից-տեղամաս ներթափանցող այսպես կոչված «պիվոտային» (pivoting) գրոհները: Դրանք շրջանցում են տեղեկատվության պաշտպանության միջոցները և ներխուժում կազմակերպության ներքին ցանց՝ օգտագործելով՝

- **SSH (Secure Shell) թունելավորումը**, որի միջոցով գրոհողը հաջողեցնում է տվյալների արտահոսքը SSH պաշտպանված երթևեկով՝ շրջանցելով ցանցի պաշտպանական մեխանիզմները և թաքնված մնալով ցանցի արտոնված երթևեկի հոսքում,
- **VPN պիվոտավորում**, որի միջոցով գրոհողը հասանելիություն է ստանում ներքին ցանցին՝ ցանցի վարկաբեկված սարքերի վրա VPN կարգավորումների իրականացման կամ սարքը կազմակերպության VPN ցանցին կապակցելու միջոցով, որը հնարավորություն է ընձեռում միջնորդավորված հասանելիություն ստանալ ցանցի այլ սարքակազմերին,
- **միջանկյալ պիվոտավորում (Proxy Pivoting)**, որի միջոցով գրոհողը հասանելիություն է ստանում ներքին ցանցին, կազմաձևում է այնտեղ HTTP/HTTPS կամ SOCKS (Socket Secure) պրոքսի և շահագործում է վարկաբեկված սարքը, որպես միջանկյալ պրոքսի սերվեր,
- **հեռավար RDP պիվոտավորում (Remote Desktop Protocol Pivoting)**, որի միջոցով գրոհողը հասանելիություն է ձեռք բերում ներքին ցանցին՝ վարկաբեկված RDP միջոցով,
- **ցանցի սկանավորման Nmap (Network Mapper)** անվճար հավելվածի գործարկման միջոցով, որի օգնությամբ գրոհողը բացահայտում է

ցանցում տեղակայված օպերացիոն համակարգերի տեսակը, միացքների կարգավիճակը և վերաբերելի այլ զգայուն տեղեկույթ:

Համակարգչային ցանցերում միացքները լինում են՝

- բաց (open port), որը նշանակում է, որ միացքն այդ պահին օգտագործվում է ինչ-որ ծառայության կողմից,
- փակ (closed port), որը նշանակում է, որ միացքը բաց վիճակում է, սակայն դեռևս չի օգտագործվում, ուստի և չի վերահսկվում ցանցի պաշտպանական համակարգերի, օրինակ՝ firewall կողմից և կարող է հասանելի լինել,
- բաց և firewall կողմից վերահսկվող՝ հետևյալ իրականացումներով.
 - **զտվող** (filtered), որի դեպքում Nmap-ը չի կարող բացահայտել միացքի բաց կամ փակ լինելը՝ firewall-ի կողմից միացքի հասանելիության արգելափակման շնորհիվ, երբ արգելափակվում են նաև Nmap-ի հարցումների պատասխանները:
 - **Չզտվող** (unfiltered), որի դեպքում ևս՝ չնայած, որ միացքը հասանելի է, միևնույն է՝ Nmap-ը չի կարող բացահայտել միացքի բաց կամ փակ լինելը, ինչպես օրինակ՝ Acknowledgement scan (ACK scan) դեպքում, երբ TCP փաթեթներն ուղարկվում են հաստատման նշիչով (Acknowledgement flag)՝ բացահայտելու համար՝ արդյո՞ք թիրախային միացքը զտվում է, թե՞ ոչ: Սքանավորումը վերլուծում է firewall-ի պատասխանը, որը կարող է լինել RST (Reset)՝ զտվող միացքի, կամ ICMP (Internet Control Message Protocol)՝ չզտվող, բայց միևնույն ժամանակ անհասանելի միացքի համար:
Այսպիսով՝ միացքի «open | filtered» վիճակը նշանակում է, որ Nmap-ը չի կարող բացահայտել դրա բաց կամ զտվող լինելը, իսկ «closed | filtered» դեպքում՝ միացքի փակ կամ զտվող լինելը:

Համակարգչային ցանցերի սքանավորման համար լայնորեն օգտագործվում է Nmap Host Discovery ծրագրակազմը (ping scanning)՝ զուգակցելով այն ARP (Address Resolution Protocol) արձանագրության հետ՝ հետևյալ հրահանգով.

```
nmap -PR -sn target(IP)
```

Նշենք, որ այս դեպքում սքանավորվում են բացառապես սարքավորումները. միացքների կարգավիճակի մասին տեղեկույթ չի բացահայտվում:

Կարելի է օգտագործել նաև ICMP echo (ping)՝

```
nmap -PE -sn target
```

Հաշվի առնելով այն, որ ICMP echo հարցումները սովորաբար արգելափակվում են, օգտագործվում է նաև ICMP timestamp հարցումը.

```
nmap -PP -sn target
```

Այս դեպքում Nmap-ն ուղարկում է ICMP type13 և սպասում է ICMP type14 պատասխանի:

OS/Service սքանավորման դեպքում Nmap միջոցով կարելի է հայտնաբերել ցանցում տեղակայված օպերացիոն համակարգերի տեսակները, միացքների վիճակը և վերաբերելի այլ տեղեկույթ:

Ստորև բերված են Nmap միջոցով կատարվող սքանավորման առավել տարածված (բայց ոչ ամբողջական) եղանակները:

Ամբողջական ցանցի սքանավորում

```
nmap <network_ip>
```

```
nmap 192.168.0.0/24,
```

Ցանցային սարքավորումների սքանավորում

```
nmap <device_ip>
```

```
nmap 192.168.0.1
```

Միացքների սքանավորում

```
nmap -p <ports> <device_ip>
```

```
nmap -p 22 192.168.0.1
```

```
nmap -p 0-1023 192.168.0.1
```

Օպերացիոն համակարգերի սքանավորում

```
nmap -O <device_ip>
```

```
nmap -O 192.168.0.1
```

Միացքներում գործարկվող ծառայությունների սքանավորում

```
nmap -sV <device_ip>
```

```
nmap -sV 192.168.0.1
```

TCP syn նշիչով (flag) սքանավորում

```
TCP syn ping
```

Այս կերպ հարցում է ուղարկվում TCP syn նշիչով և սպասվում է պատասխան: Բաց TCP միացքը պատասխանում է SYN/ACK հաստատմամբ, իսկ փակ միացքը՝ RST (reset):

Հետևյալ՝ հրահանգով՝

```
nmap -PS80 (ports) -sn target
```

TCP ack ping

հարցում է ուղարկվում TCP ack նշիչով, որի պատասխանը «RST» է:

UDP (User Datagram Protocol) սքանավորում

nmap -PA80(ports) -sn target

UDP ping

Այստեղ հարցումը դեպի բաց միացք ոչ մի պատասխան չի վերադարձնում, իսկ փակ UDP միացքից ստացվում է «ICMP port unreachable» պատասխանը:

Հետևյալ հրահանգն ուղարկելիս՝

nmap -PU80(ports) -sn target

փակ միացքը վերադարձնում է «ICMP type3, code3» պատասխանը, որով իրականում ոչինչ չի վերադարձվում:

Հետևյալ՝

TCP connect scanning

սքանավորումը կատարվում է «3-way handshake» միջոցով, որով պարզվում է ընդամենը TCP միացքի բաց կամ փակ լինելը: Այս ընթացքում TCP արձանագրության մեկնարկ տեղի չի ունենում, հարցմանն ի պատասխան՝ վերադարձվում է «RST/ACK», որից հետո հաղորդակցության սեսիան անմիջապես խզվում է:

Ստորև բերված՝

nmap -ST target

TCP SYN scanning

սքանավորման ընթացքում «3-way handshake» գործողության ավարտին սպասելու փոխարեն հաղորդակցության սեսիան ընդհատվում է սերվերից պատասխան ստանալուց անմիջապես հետո, և, քանի որ TCP հաղորդակցություն չի մեկնարկվել, ապա սքանավորման լոգավորումը դառնում է նվազ հնարավոր:

Եթե սքանավորումը կատարվում է հետևյալ՝

nmap -sS target

UDP scanning

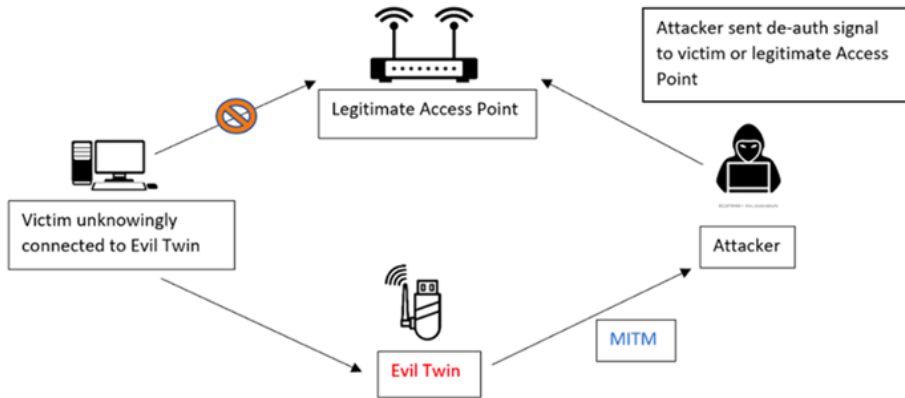
հրահանգների գործարկման միջոցով, ապա բաց միացքներ UDP հարցում ուղարկելիս ոչ մի պատասխան չի վերադարձվում, իսկ փակ միացքների դեպքում վերադարձվում է «ICMP port unreachable» պատասխանը:

Անլար ցանցերին (Wi-Fi) հատուկ կիբեռապառնալիքների ծանոթացման նպատակով ներկայացնենք դրանց զարգացման փուլերը:

- **WEP (Wired Equivalent Privacy):** Սա ստեղծված առաջին արձանագրությունն էր, ուներ գաղտնագրային լուրջ թերակատարումներ, գրոհելի էր թուլությունների ընթացքում և այժմ գործածությունից դուրս է:
- **WPA (Wi-Fi Protected Access):** Ստեղծվել է որպես ժամանակավոր լուծում՝ WEP-ին փոխարինելու համար: Համեմատաբար ավելի պաշտպանված է, սակայն համարվում է հնացած:
- **WPA2 (Wi-Fi Protected Access II):** Երկար ժամանակ գործածվում է որպես ոլորտի ստանդարտ: Օգտագործում է AES գաղտնագրման ալգորիթմը, որի շնորհիվ դեռևս լայնորեն կիրառվում է՝ ապահովելով պահանջվող կայունություն՝ երկար գաղտնաբառերի գործածության դեպքում:
- **WPA3 (Wi-Fi Protected Access III):** Սա արդիական ստանդարտ է, ունի կոշտ դիմակայունություն գաղտնաբառի գույակման գրոհներից և ապահովում է անհատականացված գաղտնագրում նույնիսկ հանրային ցանցերում:

Անլար ցանցերին հատուկ կիբեռգրոհներից են՝

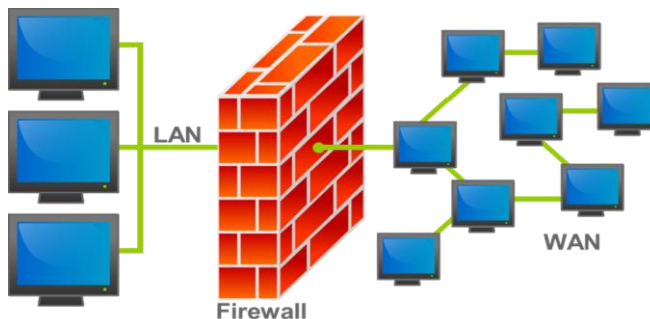
- **Evil Twin Attack:** Այս դեպքում գրոհողը ստեղծում է կեղծ Wi-Fi հասանելիության կետ (access point), որն ունի նույն SSID (Service Set Identifier) անվանումը՝ ինչ-որ օրինական ցանցը (օրինակ՝ "Yerevan_Airport_Free_WiFi"): Երբ օգտատերը միանում է կեղծ ցանցին, նրա ամբողջ երթևեկն անցնում է գրոհողի համակարգչի միջով, ինչպես «մարդը արանքում» գրոհի դեպքում՝ հանգեցնելով մուտքանունների, գաղտնաբառերի և այլ զգայուն տվյալների կողոպուտի:
- **Deauthentication Attack:** Այս դեպքում գրոհողը կարգաբերման հրահանգներ է ուղարկում օգտագործողի թվային սարքին՝ հարկադրաբար անջատելով նրան օրինական Wi-Fi ցանցից: Այս գրոհը հաճախ օգտագործվում է որպես նախապատրաստական քայլ Evil Twin հարձակման համար՝ զոհին ստիպելու վերամիանալ և ընտրել կեղծ ցանց:



Նկ. 24. Անլար ցանցերի կիբեռգորհների կազմակերպումը:

Համակարգչային ցանցերի կիբեռգորհները մշտապես հավանական իրադարձություններ են և այս համատեքստում, համակարգչային ցանցերի և սարքավորումների կիբեռանվտանգությունը միջոցառումների, տեխնոլոգիաների և քաղաքականությունների ամբողջություն է՝ ուղղված շրջանառվող տվյալների գաղտնիության, ամբողջականության և հասանելիության (confidentiality, integrity and availability, CIA) ապահովմանը: Ստորև բերված են ցանցային կիբեռանվտանգության հիմնական գործիքակազմ և տեխնոլոգիական լուծումներ:

- «Կրակե պատնեշ» (Firewall)՝ հայտնի է նաև որպես «հրդեհային պատ»:



Նկ. 25. Ցանցային կրակե պատնեշի սխեման:

Կրակե պատնեշը անվտանգային համակարգ է, որը մշտադիտարկում և վերահսկում է մտից և ելից ցանցային երթևեկը՝ հիմնվելով համակարգ-

չային ցանցի կարգավորումներով նախասահմանված խիստ որոշակի կանոնների վրա: Կիբեռանվտանգային այս գործիքակազմի մեկ այլ անվանում է՝ «միջցանցային էկրան»-ը՝ երբ խոսքը վերաբերում է վերջինիս ապարատային իրականացմանը, որը միջանկյալ պատնեշ է վստահելի տեղայնացված ցանցի և անվտանգության արտաքին համացանցի միջև:

Ավանդական հրդեհային պատնեշները կիրառվել են համացանցի ստեղծումից դեռևս շատ առաջ՝ շինություններում հրդեհի տարածումը կանխելու նպատակով: Ժամանակակից թվային պատնեշները վերահսկում են ցանցային երթևեկը և կանխարգելում կիբեռգրոհների ներթափանցումը:

Կրակե պատնեշների վերաբերյալ համապարփակ ուսումնասիրություն կարելի է կատարել՝ այցելելով հետևյալ կայքէջերը^{****22)}:

Կրակե պատնեշն ապահովում է ցանցային երթևեկի անվտանգ ընդունումը՝ արգելափակելով այն երթևեկը, որը կարող է այդ մակարդակում կիբեռսպառնալիք ներկայացնել: Պատնեշը գործում է որպես զտիչ (filter), տեղակայվում է ներցանցային համակարգիչների, սարքավորումների և արտաքին ցանցի միջև և կանխարգելում է հաքերային ներխուժումները:

Նշենք, որ, չնայած այն հանգամանքին, որ պատնեշը կիբեռանվտանգային գործիքակազմի կարևոր համալրում է, այնուամենայնիվ՝ այն ի գործու չէ ապահովելու ցանցի ամբողջական անխոցելիությունը, իսկ որոշակի դեպքերում կարող է սխալմամբ նույնիսկ խոչընդոտ դառնալ արտոնյալ ծրագրերի ներբեռնման և գործարկման համար:

Ժամանակակից թվային համակարգերի մեծ մասն արդեն իսկ արտադրվում են ներկառուցված ծրագրային կրակե պատնեշներով, ինչպես օրինակ՝ Windows Defender-ն է, և ամենևին հարկավոր չէ կրակե պատնեշ տեղադրել օրինակ՝ գրասենյակային համակարգչի համար: Պարզապես պետք է համոզվել, որ պատնեշը լռելյայն ակտիվացված է, հակառակ դեպքում նոր միայն դիմել հարկավոր միջամտության:

Իսկ ի՞նչ տարբերություն կա կրակե պատնեշի և հակավիրուսային ծրագրի միջև:

****22) <https://www.imperva.com/learn/application-security/what-is-web-application-firewall-waf>
<https://edrawmax.wondershare.com/network-tips/network-diagram-tips.html>



Նկ. 26. Ցանցային կրակե պատնեշների և հակավիրուսային ծրագրերի համեմատությունը:

Հակավիրուսային ծրագիրն ունակ է ստուգել ներբեռնված ֆայլերը և այցելված կայքերը՝ պարզելու համար՝ արդյո՞ք դրանք սպառնալիք են ներկայացնում կամ պարունակում: Պոտենցիալ սպառնալիքի կասկածի դեպքում հակավիրուսային ծրագիրն արգելափակում է սպառնալիքի ներթափանցումը՝ նախքան այն կարող է որևէ վնաս հասցնել համակարգչին կամ սերվերին:

Կրակե պատնեշը վերահսկում և որոշ դեպքերում արգելափակում է մտից և ելից երթևեկը ցանցային ադմինիստրատորի կողմից սահմանված կանոնների (քաղաքականության) հիման վրա, սակայն այն չի ստուգում երթևեկի բովանդակությունը: Սպառնալիքի հայտնաբերման դեպքում պատնեշն արգելափակում է դրա հասանելիությունը ներցանցային սարքավորումներին՝ ցանցային երթևեկի զտման (traffic data filtering) միջոցով, իսկ արձանագրության տվյալները (ip|segment|data|) ստուգվում են հետևյալ փուլերով:

1. արձանագրությունների գլխամասային (header) տվյալների վերլուծություն (Source-Destination, IP, PORT, protocol)
2. արձանագրության համեմատում նախասահմանված կանոնների (սահմանափակումների) հետ, որոնցում, օրինակ՝ կարող է նշված լինել՝ «արգելափակել բոլոր հարցումները 192.168.15.2 հասցեից դեպի 22/tcp միացք»
3. նշված սահմանափակումներին Packet կամ Segment տվյալների համապատասխանության դեպքում երթևեկն արգելափակվում է:

Կրակե պատնեշները համարվում են կիբեռանվտանգության հիմնական և ամենատարածված բաղադրիչներից մեկը: Դրանք սովորաբար տեղակայվում են համակարգչային ցանցերի տարբեր տիրույթների միջև: Օրինակ՝ կազմակերպության ցանցն ու համացանցը տարանջատվում են հատուկ պատնեշի միջոցով, որը վերահսկում է երկկողմանի երթևեկը: Սա նշանակում է, որ այն ոչ միայն արգելափակում է կասկածելի մուտքային ցանցերը, այլև՝ վերահսկում է նաև ելքային հաղորդակցությունը, որպեսզի աշխատակազմի անդամները՝ անվտանգ կերպով օգտվելով արտաքին անհրաժեշտ ծառայություններից՝ web, email և այլն, չկարողանան զգայուն տվյալներ արտահոսեցնել: Նման մոտեցումը ներքին կիբեռսպառնալիքների և ներդրված լրտեսների դեմ պայքարի արդյունավետ միջոց է:

Կրակե պատնեշները տարակարգվում են հետևյալ տեսակների.

- **«Հոսթ» պատնեշ (Host based firewall):** Սա ծրագրային պատնեշ է, որը տեղադրվում է առանձին վերցրած համակարգչի վրա, ինչպես օրինակ՝ ներկառուցված Windows Firewall-ը կամ iptables-ը և սահմանում է, թե ինչ տիպի մուտքային և ելքային երթևեկ է թույլատրվում տվյալ համակարգչի համար:
- **Ապարատային պատնեշ (Hardware firewall):** Սրանք սարքավորումներ են՝ պատնեշներին հատուկ գործառույթներով և որևէ այլ դերակատարում չունեն: Տեղակայվում են ցանցի տարբեր տիրույթների միջև, ինչպես օրինակ՝ Cisco firewall-ը: Նախատեսված են կազմակերպության ցանցի տրոհման համար՝ ապահովելով տարբեր տիրույթների տեղեկատվական անվտանգությունը:
- **Երթուղավորիչների պատնեշ (router firewall):** Չնայած իրենց համեմատաբար պարզ կառուցվածքին, հաջողությամբ կատարում են նույն գործառույթը՝ տարանջատելով համացանցից եկող երթևեկը տեղայնացված ներքին ցանցից:

Չնայած ծրագրային և ապարատային պատնեշների իրականացման տարբերությանը, դրանց հիմնական գործառույթները նույնն են՝ երթևեկի զտում, մոխից/ելից երթևեկի վերահսկում և պոտենցիալ վտանգավոր հարցումների արգելափակում: Պատնեշի նախագծային առանձնահատկություններով պայմանավորված՝ երթևեկի զտումը կատարվում է հետևյալ եղանակներով:

- **Ձտում ըստ փաթեթների և սեգմենտների (Packet/Segment)**, որի դեպքում ստուգվում է միայն IP/Port/Protocol-ի բովանդակությունը: Օրինակ՝ հետևյալը՝ որպես սահմանափակում, կարող է նշված լինել Iptables-ում՝ «արգելափակել բոլոր հարցումները դեպի 23 պորտ (telnet)» և «թույլատրել բոլոր հարցումները դեպի 80 միացք միայն 192.168.0.0/24 ցանցից»:

```
iptables -A INPUT -p tcp --dport 23 -j DROP
iptables -A INPUT -p tcp -s 192.168.0.0/24 --dport 80 -j ACCEPT
```

Մինչդեռ՝ ավելի պարզեցված միջերեսով (interface) պատնեշի (Uncomplicated Firewall, UFW) դեպքում՝ նույն այդ սահմանափակումներն ունեն հետևյալ տեսքը:

```
ufw deny 23/tcp
ufw allow from 192.168.0.0/24 to any port 80 proto tcp
```

- **Ձտում՝ ցանցային կապերի կարգավիճակի աղյուսակի հիման վրա:** Այս դեպքում, եթե օրինակ՝ ներցանցային ինչ-որ հանգույց TCP 443 միջոցով կապ է հաստատում որևէ արտաքին սերվերի հետ, ապա պատնեշը կթույլատրի այդ սերվերից վերադարձվող երթևեկը: Մինչդեռ՝ արտաքին ցանցերից նույն այդ պորտին հասանելիության փորձերն առանց ներսից հարցման՝ պատնեշի կողմից կկանխարգելվեն:
- **Երթևեկի տվյալների զտում:** Այստեղ կատարվում է երթևեկի տվյալների բովանդակության ստուգում: Օրինակ՝ արտաքին ցանցերից վեբ սերվերին ուղղված «GET /etc/passwd» հարցումը կորակվի որպես վտանգավոր հարցում և կարգելափակվի:

Ստորև բերված են կրակե պատնեշների տարատեսակները:

Bastion Host: Սա սարք է, կամ՝ հոսթ, որն ամբողջությամբ ենթակա է գրոհի, քանի որ միացված է արտաքին ցանցին և հանդես է գալիս որպես առաջին պաշտպանական գիծ: Այս տերմինը կիրառվում է ոչ միայն պատնեշների, այլև ցանկացած ներցանցային ռեսուրսների նկատմամբ, որոնք օժտված չեն անվտանգային որևէ այլ լրացուցիչ միջոցով և նրա խնդիրն է՝

ընդունել դեպի ցանց ուղղված բոլոր գրոհները և կրել դրանց հետևանքները: Ավանդականորեն տեղադրվում է համացանցի և ներքին ցանցի միջև:

Իրականացման տարբերակներն են՝

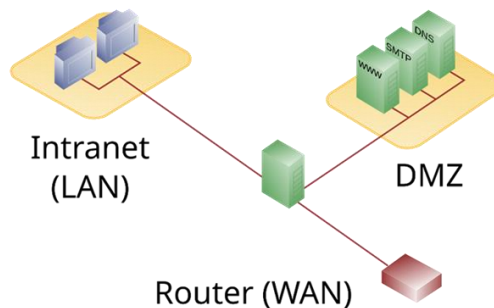
Եզակի ցանցային միջերեսով պատնեշ՝

- մտից/ելից երթևեկն ամբողջությամբ անցնում է այդ միջերեսով,
- երթևեկի զտումն իրականացվում է համապատասխան ծրագրակազմի կիրառմամբ

Կրկնակի ցանցային միջերեսով պատնեշ: Սա առավել տարածված տարբերակն է, ունի երկու ցանցային ինտերֆեյս՝ արտաքին (համացանցի) և ներքին (տեղայնացված) ցանցերի համար: Երթևեկը զտվում է համապատասխան ծրագրակազմի կիրառմամբ, իսկ ցանցերի սեզմենտավորումն իրականացվում է երկու ցանցային քարտերի (Network Interface Card, NIC) միջոցով:

Էկրանավորված ենթացանցային պատնեշ (Screened Subnet Firewall), որը հայտնի է նաև որպես «եռակի պատնեշ»՝ երեք ցանցային միջերեսներով: Այն տրոհում է երթևեկը հետևյալ երեք տրամաբանական սեզմենտների՝

1. հանրային համացանց
2. ներքին ցանց
3. լրացուցիչ գոտի (օր.՝ Demilitarized Zone, DMZ)՝ ինչպես պատկերված է Նկ. 27-ում:



Նկ. 27. Էկրանավորված ենթացանցի պատնեշի սխեմա:

Սեսիայի մակարդակի դարպաս (Circuit-Level Gateway): Սա OSI մոդելում գործում է հաղորդակցության սեսիայի մակարդակում:

Գործողության սկզբունքն է՝

- հարցումների փոխանցում հաճախորդի և սերվերի միջև,

- հաճախորդի և սերվերի միջև ուղղակի կապ չի ստեղծվում,
- պատնեշը ստեղծում է երկու կապ՝ մեկը հաճախորդի, մյուսը՝ սերվերի հետ,
- հետևում է TCP սեսիաներին (Tree-Way Handshake):

Առանձնահատկություններից են՝

- հաստատում է կապի իրավագործությունը (legitimacy),
- բովանդակության վերլուծություն չի իրականացնում, որի հետևանքով վնասաբեր տվյալներ և ծրագրակազմ կարող են անտեսվել,
- ավանդականորեն իրականացվում է NAT (Network Address Translation) հետ միասին,
- ունի պարզեցված պաշտպանություն, որով պայմանավորված՝ վնասաբեր երթևեկը կարող է ընդունել որպես իրավագործ:

Կիրառության մակարդակի պատնեշ (Application-Level Gateway / Proxy Firewall):

Գործողության սկզբունքն է՝

- գործում է որպես proxy server,
- վերլուծում է հարցումը, որից հետո նոր միայն փոխանցում այն սերվերին,
- գործում է տարատեսակ արձանագրություններով՝ HTTP, HTTPS, FTP, SMTP և այլն:

Գործառնություններն են՝

- երթևեկի բովանդակության վերլուծություն,
- կայքերի արգելափակում,
- վնասաբեր ձևանմուշների (SQL Injection, XSS) հայտնաբերում, ինչպես օրինակ՝ Cloudflare WAF-ը:

Առանձնահատկություններից են՝

- առավել մանրամասն վերահսկողություն՝ համեմատած հաղորդակցության սեսիայի մակարդակում տեղակայված պատնեշների հետ,
- ցանցային երթևեկի զտում:

Ցանցի ամբողջական վիճակագրության վերահսկման պատնեշներ (Stateful Inspection Firewalls, SPI): Սա OSI մոդելում անվտանգային մոտեցում է, հետևում է ցանցային կամ ծրագրային իրադարձությունների ամբողջական պատմությանը՝ ցանցի բնականոն վարքագծից շեղումների հայտնաբերման նպատակով՝ ի տարբերություն վիճակազուրկ մեթոդների, որտեղ

իրադարձություններն ուսումնասիրվում են որպես առանձին վերցրած միավորներ:

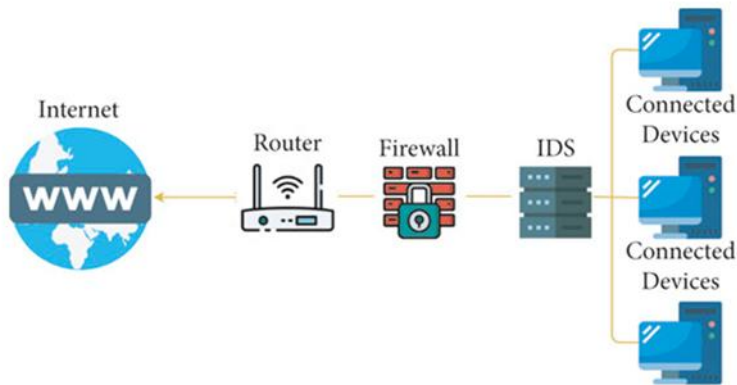
Գործողության սկզբունքն է՝

- պահպանում են հաղորդակցության սեսիայի վերաբերյալ տեղեկությունը և
- փաթեթները դիտարկում են որպես ամբողջական սեսիայի մաս Առանձնահատկություններից են՝.
- վերոհիշյալ երեք մոդելների հնարավորությունների համադրում,
- հաղորդակցական ամբողջական սեսիայի վերլուծության հիման վրա որոշումների կայացում,
- փաթեթների գտման տեսանկյունից՝
 - հնարավորություն են ընձեռում պարզել՝ արդյո՞ք փաթեթը հաղորդակցական սեսիայի մասն է, թե՞ ոչ, և կարող են կանխել կեղծ փաթեթները:

Համակարգչային ցանցերի կիբեռանվտանգային մեկ այլ կարևոր մոտեցում է ցանցային ներխուժումների հայտնաբերման և կանխարգելման համակարգերի ներգրավումը:

Ներխուժումների հայտնաբերման համակարգերն (Intrusion Detection Systems, IDS) ապարատային կամ ծրագրային կիբեռանվտանգային գործիքակազմեր են, որոնք վերահսկում են ցանցը կամ համակարգը և իրազեկում ադմինիստրատորներին հայտնաբերված վնասաբեր գործողությունների, կիբեռանվտանգային քաղաքականության խախտումների, կասկածելի վարքագծերի և պոտենցիալ սպառնալիքների վերաբերյալ: Գործողության սկզբունքը սպառնալիքների նույնականացումն է՝ հիմնվելով երթևեկի վերլուծության վրա: Հետագա գործողությունները թողնվում են ադմինիստրատորին:

Այս համակարգերն ավանդականորեն գործում են՝ համեմատելով ցանցային երթևեկը կամ համակարգի գործունեությունը հայտնի վնասաբեր ծրագրակազմի այսպես կոչված՝ «ստորագրությունների և թվային մատնահետքերի» հետ՝ հայտնաբերելով շեղումներ համակարգերի բնականոն գործունեությունից: Որպես տեղեկատվական պաշտպանության կարևոր միջոց՝ տրամադրում են վաղ ահազանգեր՝ իրականացնելով գրոհի օրինաչափությունների վերլուծություն և կիբեռդիպվածներին արձագանքելու ընթացքի աջակցում:



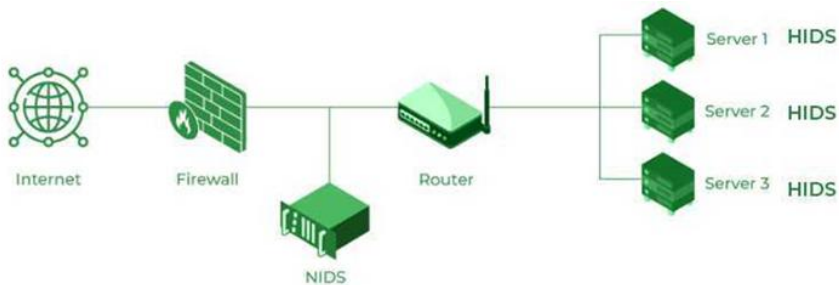
Նկ. 28. Զանցային ներխուժումների հայտնաբերման համակարգերի կառուցվածքային սխեման:

Սույն ձեռնարկում սահմանափակվում ենք ներխուժումների հայտնաբերման առավել տարածված համակարգերի նկարագրությամբ: Այդ համակարգերից են՝

- ցանցային (Network Intrusion Detection Systems, NIDS) ներխուժումների հայտնաբերման համակարգերը,
- այսպես կոչված՝ «հոսթերի» դեմ իրականացվող ներխուժումների հայտնաբերման համակարգերը (Host Intrusion Detection System, HIDS),
- ցանցի պարագծով ներխուժումների հայտնաբերման համակարգերը (Perimeter Intrusion Detection Systems, PIDS) և
- վիրտուալ տեխնոլոգիաների օգտագործմամբ համակարգերը (Virtual Machine-based Intrusion Detection Systems, VMIDS):

Թվարկված և գոյություն ունեցող բոլոր համակարգերը տարբերվում են ըստ տեղադրման վայրի և գործողության սկզբունքի:

Ներխուժումների հայտնաբերման ցանցային համակարգերը տեղադրվում են տեղեկատվական ցանցի ռազմավարական առումով կարևոր տեղամասերում և ապահովում են ցանցային բոլոր սարքավորումների մտից/ելից երթևեկի խորքային վերահսկողություն՝ Data Link մակարդակից մինչև Application մակարդակը:

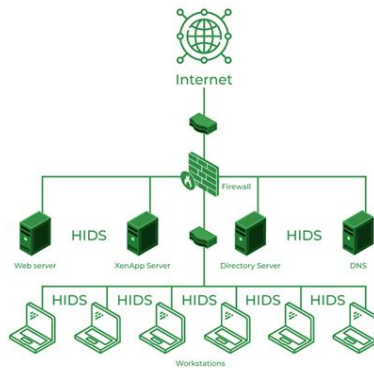


Նկ. 29. Ներխուժումների հայտնաբերման հոսթ համակարգերի կառուցվածքային սխեման:

Այս համակարգերը տարբերվում են կրակե պատնեշներից նրանով, որ վերջիններս հայտնաբերում և գրանցում են միայն արտաքին ցանցերից եկող գրոհները, մինչդեռ՝ NIDS-ը կարող է հայտնաբերել նաև ներքին սպառնալիքները: Սույնով՝ ցանցային ներխուժման հայտնաբերման համակարգերը վերահսկում են ամբողջական ցանցը՝ հնարավորություն ընձեռելով վերանայել անվտանգային լրացուցիչ լուծումների ներդնումը և դրանց առնչվող ծախսերը: Նշենք, որ ցանցային ամբողջական երթևեկի վերահսկման համակարգերը, այդ թվում նաև NIDS-ը, զգալիորեն ռեսուրսատար են: Որքան մեծ է երթևեկի ծավալը, այնքան մեծ են հաշվողական հզորություններին և հիշողությանը ներկայացվող պահանջները: Այս իրողությամբ պայմանավորված, կարող են դիտվել տվյալների փոխանակման զգալի ուշացումներ և ցանցի արագագործության նվազում: Գաղտնիք չէ, որ մեկ համակարգի կողմից տեղեկատվության մեծ քանակության մշակումը կարող է հանգեցնել այդ համակարգի ծանրաբեռնման (single point of failure): Որպես հետևանք՝ կարող են դիտվել փաթեթների վերահսկողության վրիպումներ, որի արդյունքում NIDS համակարգը և, ուստի նաև՝ ցանցը, կարող են դառնալ խոցելի:

Ներխուժումների հայտնաբերման HIDS համակարգերը տեղադրվում են ցանցի ներսում մեկ հոսթի վրա և պաշտպանում են միայն այդ հոսթը՝ իրականացնելով բոլոր մոտից և ելից փաթեթների վերահսկողությունը: Ցանցի «հոսթը» համակարգիչ կամ սարքավորում է, որը միացված է ցանցին և ունի իր սեփական ցանցային հասցեն: Հոսթերի օրինակ են՝ սերվերները, հաճախորդ համակարգիչները և անձնական օգտագործման թվային սարքերը: Հոսթերը շփվում են այլ հոսթերի հետ՝ տեղեկատվություն, ծառայություններ և ծրագրակազմ փոխանակելու համար: Միջնորդ սարքերը, ինչպիսիք են՝

ռաուտերները և փոխանջատիչները, սովորաբար համարվում են ցանցային հանգույցներ, բայց ոչ՝ հոսթ, քանի որ դրանք ընդամենը ուղղորդում են ցանցային երթևեկը՝ առանց սեփական ցանցային հասցեն ունենալու անհրաժեշտության: Հոսթերի վերահսկման HIDS համակարգերում շրջանառվող նիշքերի պատկերները (snapshot) համեմատվում են նախասահմանված պատկերի հետ՝ ստանդարտից դուրս շեղումների և հավանական կիբեռսպառնալիքների բացահայտման նպատակով:



Նկ. 30. Ներխուժումների հայտնաբերման հոսթ համակարգերի կառուցվածքային սխեման:

Խորհուրդ է տրվում HIDS-երը տեղադրել ցանցի առավել կարևոր համակարգիչների վրա, որոնց կարգաբերումները հազվադեպ են փոփոխվում:

Ներխուժումների հայտնաբերման PIDS համակարգերը պաշտպանում են ամբողջական ցանցի պարագիծը՝ արգելափակված վայրերում տեղադրվող փշալարերի և ազդանշանային ցանկապատերի նմանակով:

Վիրտուալ տեխնոլոգիաների հիման վրա ներխուժումների հայտնաբերման VMIDS համակարգերում վերահսկողական մեխանիզմները տեղադրվում են ոչ թե առանձին սարքակազմի վրա, այլ՝ այդ նպատակով տեղակայված վիրտուալ մեքենայի վրա, որը վերահսկում է ցանցի բնականոն գործունեությունը և հայտնաբերում շեղումները:

Ցանցային ներխուժումների հայտնաբերման համակարգերի հետ մեկտեղ գործածվում են նաև **ներխուժումների կանխարգելման համակարգերը** (Intrusion Prevention System IPS): Եթե IDS համակարգերն իրազեկում են ներխուժումների մասին, ապա՝ IPS համակարգերը՝ լինելով ավելի առաջադեմ՝ ինքնաշխատ կերպով կանխում են այդ ներխուժումները: Այս իմաստով, IPS-ը դիտարկվում է նաև որպես IDS-ի զարգացում: Այն իրական ժամանա-

կում զուգակցում է երթևեկի և ցանցային վարքագծի վերլուծության IDS գործիքակազմը չարամիտ գործողությունների արգելափակման գործիքակազմի հետ:

Ո՞րն է IPS համակարգերի գործողության սկզբունքը:

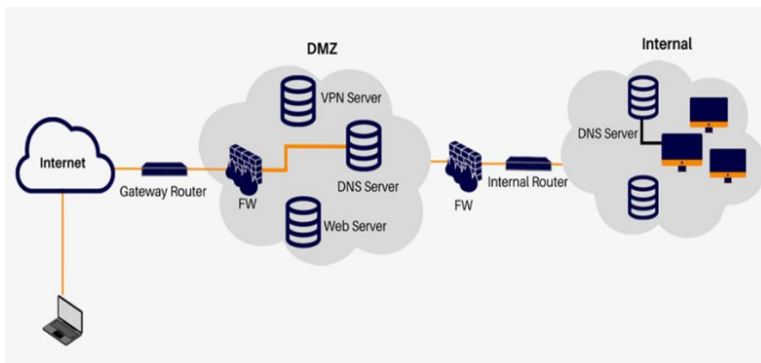
Այս համակարգերը տեղակայվում են անմիջապես երթևեկի հոսքի մեջ և իրականացնում են փաթեթների վերահսկողություն՝ զննելով դրանց բովանդակությունը՝ նախքան դրանց ուղղորդելը հասցեատերերին: Արդյունքում՝ համակարգը որոշում է կայացնում՝

- թույլատրել փաթեթը,
- արգելափակել փաթեթը,
- խմբագրել փաթեթը՝ հեռացնելով վնասաբեր հատվածները,
- արգելափակել կասկածելի հաղորդակցությունը և այլն:

Այսպիսով, IPS համակարգերը ոչ միայն իրազեկում են կիբեռնապահակների մասին, այլև ակտիվորեն արձագանքում են դրանց:

Ըստ տեղակայման վայրի՝ IPS համակարգերը լինում են ցանցային (Network-based IPS, NIPS) և հոսթին միտված (Host-based IPS, HIPS):

Համակարգչային ցանցերում կարող են տեղակայվել նաև այսպես կոչված՝ «ապառազմականացված գոտիներ» (Demilitarized Zone, DMZ), որոնք կիբեռանվտանգության ճարտարապետության կարևոր բաղադրիչներից են: Դրանք առանձին ենթացանցեր են, որոնք մեկուսացված են ինչպես համացանցից, այնպես էլ՝ կազմակերպության ներքին ցանցից:



Նկ. 31. Համակարգչային ցանցերում ապառազմականացված գոտու տեղակայման սխեման:

Ապառազմականացված գոտու հիմնական նպատակը ցանցային ծառայությունների լիազորված հասանելիության ապահովումն է և միաժամանակ՝

չլիազորված հասանելիության կանխարգելումը: Այստեղ ավանդականորեն տեղակայվում են ծառայություններ, որոնք հասանելի են արտաքին ցանցերից, ինչպիսիք են՝

- վեբ սերվերները՝ ներառյալ կազմակերպության կայքերը, առցանց առևտուրը և առցանց հարթակները,
- փոստային սերվերները (SMTP, IMAP, POP3),
- արտաքին հարցումներին պատասխանող DNS սերվերները,
- աշխատակազմի հեռավար հասանելիության VPN (Virtual private network) դարպասները,
- նիշքերի փոխանակման FTP/SFTP (File transfer Protocol, Secure File Transfer Protocol) սերվերները,
- VoIP (Voice Over Internet Protocol) և միջնորդավորված (Proxy) սերվերները, հեռավար հասանելիության RDP (Remote Desktop Protocol) դարպասները:

Թվարկված ծառայությունները պետք է հասանելի լինեն համացանցից, միևնույն ժամանակ՝ կազմակերպության ցանցը պետք է պաշտպանված լինի դրանցից եկող հավանական կիբեռսպառնալիքներից և գրոհներից:

Ապառազմականացված գոտիների ներդնման արդյունքում կատարվում է կազմակերպության ցանցի տրոհում հետևյալ երեք գոտիների.

- Համացանց, որը դիտարկվում է որպես լիովին անվտանգ գոտի:
- DMZ, որը դիտարկվում է որպես ոչ լիարժեք վստահության գոտի:
- Ներքին ցանց, որը դիտարկվում է որպես լիարժեք վստահության գոտի:

Այսպիսով, DMZ տեղակայմամբ՝

- ցանցի մտից/ելից երթևեկը զսկում է կրակե պատնեշի միջոցով,
- համացանցից թույլատրվում են բացառապես անհրաժեշտ ծառայությունները (HTTP, HTTPS),
- ներքին ցանցում գործարկվում են խիստ սահմանափակ հաղորդակցություններ, օրինակ՝ ներքին տվյալների հենքի հասանելիություն վեբ սերվերի կողմից,
- ներքին ցանցն ադմինիստրատորների կողմից ենթարկվում է մշտադիտարկման, կառավարման և թարմացման,
- կիբեռգրոհողների կողմից DMZ սերվերին չլիազորված հասանելիության դեպքում ներքին ցանցի կարևորագույն սերվերները մեկուսացվածության շնորհիվ մնում են պաշտպանված:

ԳԼՈՒԽ 6. Տվյալների հենքերի կիբեռանվտանգություն

Տվյալների հենքը թվային պահոց է, որը նախատեսված է կառուցվածքավորված տվյալների հավաքածուներ պահպանելու և կառավարելու համար: Կազմակերպության ներսում դրանք կարևորագույն ակտիվներ են: Դրանցում պահպանվում են հաճախորդների անձնական տվյալներ (Personally Identifiable Information, PII), ֆինանսական գրառումներ, մտավոր սեփականություն, առևտրային ու ռազմավարական գաղտնիքներ և այլն: Սա նշանակում է, որ տվյալների հենքերի կիբեռանվտանգության ոչ պատշաճ կազմակերպումը չի կարող դիտարկվել որպես պարզապես տեխնիկական ծախսողում: Հաջողված կիբեռգործի արդյունքում կարող է տեղի ունենալ անձնական տվյալների չարտոնված հասանելիություն, նենգափոխում, արտահոսք, որի հետևանքով կազմակերպությունը կարող է նաև ենթարկվել կարգավորող մարմինների կողմից պատժամիջոցների և տուգանքների: Կիբեռանվտանգային դիպվածներն անպայմանորեն հանգեցնում են կազմակերպության վարկանիշի անկման, իսկ եթե տվյալների հենքը նաև գերատեսչական կամ ռազմավարական է, ապա՝ նաև ազգային անվտանգության սպառնալիքի: Այսպիսով, տվյալների հենքերի կիբեռանվտանգության ապահովումը կրիտիկական նշանակություն ունի կազմակերպության կայունության և երկարաժամկետ հաջողության համար:

Տվյալների հենքերի հիմնական բաղադրիչներն են՝

- **Տվյալները՝** որպես պահպանվող հիմնական տեղեկատվություն. տեքստ, թվեր, պատկերներ, տեսանյութեր և այլն:
- **Տվյալների հենքի կառավարման համակարգը:**
- **Ծրագրակազմը,** որը թույլ է տալիս օգտատերերին փոխազդել տվյալների հենքի հետ՝ հարկավոր գործառնությունների իրականացման նպատակով:

Տվյալների հենքերի ճարտարապետությունը

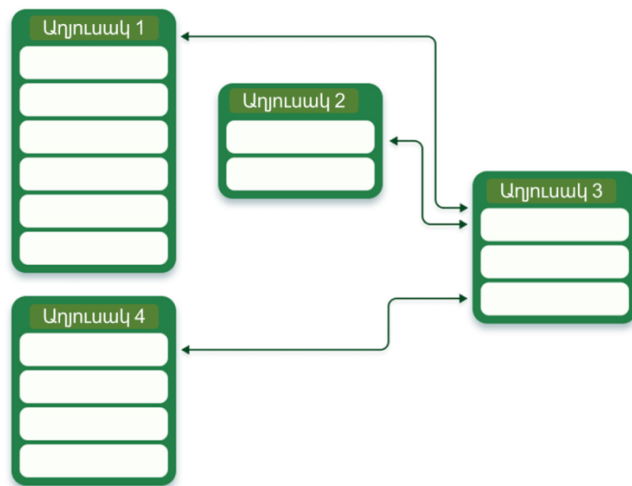
Տվյալների հենքերի կիբեռանվտանգության կազմակերպումն անպայմանորեն ենթադրում է դրանց հիմնական ճարտարապետական մոդելների իմացություն^{††††23)}: Տվյալների հենքերի մոդելները տարբերվում են իրենց

^{††††23)} K.Loney, Oracle Database 19c DBA Handbook. Oracle Press, 2020.

սանդղավորմամբ (ընդլայնման հնարավորություններով), կառուցվածքով և տվյալների կազմակերպմամբ, որով պայմանավորվում է այս կամ այն մոդելի կիրառական տանգային առանձնահատուկ լուծումների շրջանակը:

Կառուցվածքավորված և չկառուցվածքավորված հարցումների օգտագործման տեսանկյունից՝ տվյալների հենքերի կառավարման մոդելները տարանջատվում են ռելացիոն (SQL) և ոչ ռելացիոն (NoSQL) իրականացումների:

Տվյալների ռելացիոն հենքեր (relational data bases): Այստեղ տվյալները կազմակերպվում են աղյուսակների մեջ՝ տողերով և սյուներով: Աղյուսակները նախագծվում են պահպանվող տվյալների, դրանց տիպերի և տվյալների միջև նախասահմանված կապերի հաշվի առնմամբ: Տվյալները կապակցվում են միմյանց հետ բանալիների միջոցով, որոնք եզակի նույնացուցիչներ են և տրամաբանական կապեր են հաստատում տարբեր աղյուսակների միջև: Իրականացումներից են՝ MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database, MariaDB, IBM Db2 և SQLite հենքերը: Ռելացիոն հենքերը հիմնականում սանդղավորվում են ուղղահայաց կերպով (vertical scaling)՝ սերվերների հզորության և հիշողության ավելացման միջոցով:



www.apriorit.com

Նկ. 32. Տվյալների ռելացիոն հենքի մոդելը:

K.Nield., Getting Started with SQL: A Hands-On Approach for Beginners. O'Reilly Media, 2021.

N.Redkar, Troubleshooting SQL Server: A Guide for the Accidental DBA. Apress, 2020.

R.Stephens, SQL Server 2022 Administration Inside Out. Microsoft Press, 2023.

Այս ճարտարապետությունը կառուցվում է տվյալների ատոմարության, հենքի վավերականության, մեկուսացվածության և երկարակեցության սկզբունքների (ACID principle, Atomicity, Consistency, Isolation, and Durability) համաձայն՝ հավաստիացնելով գործառույթների և մշակվող տվյալների ամբողջականությունը: Ատոմարությունը երաշխավորում է գործառույթների «ամբողջական, կամ՝ ոչինչ» լինելը: Սա նշանակում է, որ գործառույթի ընդհատումը հանգեցնում է մինչ այդ պահը կատարված գործողությունների ամբողջական չեղարկմանը, ինչպես օրինակ՝ բանկային գործառույթի ընդհատումը՝ պայմանավորված սխալ հաշվեհամարի, անցաբառի կամ հոսանքի ընդհատման հետ, չի հանգեցնում հաշվեհամարի ինչ-որ փոփոխության: Հենքի վավերականությունը նշանակում է, որ այն մշտապես գործող կարգավիճակում է և հարցումներին արձագանքում է նախասահմանված վարքագծի համաձայն: Հենքի մեկուսացվածությունը նշանակում է, որ մրցակցող գործառույթները, ինչպիսիք են, օրինակ՝ միևնույն հաշվեհամարի մտից/ելից ֆինանսական գործողությունները, չեն վերադրվում և միմյանց վրա որևէ ազդեցություն չունեն: Հենքի երկարակեցությունը պայմանավորում է նրա վթարակայունությունը այն իմաստով, որ առանձին ձախողումները չեն անդադրադադում ամբողջական համակարգի բնականոն գործունեության վրա:

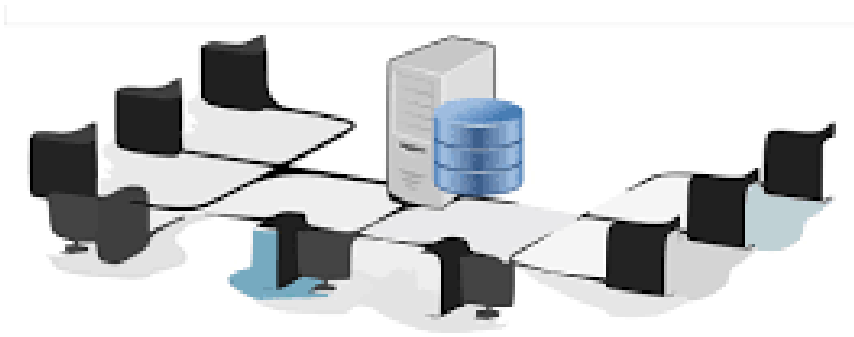
Տվյալների ոչ ռեպացիոն հենքեր: Այս հենքերն ապահովում են տվյալների ճկուն մոդելներ և համատարած կերպով չեն օգտագործում կառուցվածքավորված հարցումների SQL լեզուն: Յուրաքանչյուր NoSQL տվյալների հենք կարող է ունենալ սեփական հարցման լեզու կամ միջերես՝ API: Սանդղավորվում են հորիզոնական ուղղությամբ՝ թույլ տալով տվյալների բաշխում բազմաթիվ սերվերների կամ հանգույցների միջև՝ ապահովելով մեծածավալ և չկառուցվածքավորված տվյալների արդյունավետ մշակում: Այս հենքերի նախագծման հիմքում նույնպես ընկած են սկզբունքներ, որոնք են՝ հասանելիությունը, ճկունությունը և վավերականությունը (Basically Available, Soft state, Eventually consistent, BASE principles): Ճկունությունն այստեղ նշանակում է՝ ետնաբեմային (background) գործողությունների, ինչպիսիք են՝ տվյալների թարմացումները, անվտանգային միջոցառումները և այլն, կատարման ընթացքում ամբողջական հենքի գործառնական վիճակի շարունակականություն: Տվյալների հենքը պետք է նախագծված լինի ետնաբեմային վիճակների արտոնված փոփոխությունները սահունորեն կարգավորելու և կառավարելու համար այնպես, որ դրանք չհանգեցնեն տվյալների

որևէ վնասման կամ կորստի: Ոչ ռեկացիոն հենքերը սանդղավորվում են հորիզոնական ուղղությամբ (horizontal scaling)՝ համակարգին նոր սերվերներ ավելացնելով: Իրականացումներից են՝ MongoDB), Redis, Cassandra և Neo4 հենքերը:

Վերոհիշյալ երկու՝ SQL և NoSQL ճարտարապետությունների տարբերությունները սոսկ նախագծային չեն. դրանցով են պայմանավորվում այդ համակարգերին հատուկ խոցելիությունները: Օրինակ՝ SQL հիմքով նախագծված հենքերի կախվածությունը կառուցվածքավորված հարցումների լեզվից դրանց դարձնում է միատիպ թիրախ՝ հարցումների քերականության (query syntax) վրա հիմնված գրոհների համար, ինչպիսին է՝ SQL Injection գրոհը: Ի հակադրություն այդ հենքերի՝ NoSQL հենքերի ճկունության և հարցումների տարատեսակ API-ներով օժտվածության շնորհիվ, դրանց դեմ ուղղված գրոհներն ուղղվում են արդեն ոչ թե ընդամենը հարցումների միասնական լեզվի, այլ այդ հենքերի տվյալների մոդելի տրամաբանության դեմ, որի օրինակ է JSON օբյեկտների (java Script Object notation) շահարկումը MongoDB հենքերում: Սա իր հերթին նշանակում է, որ տվյալների հենքերին միտված անվտանգային լուծումները մեծապես կախված են ոչ միայն այդ հենքերի ճարտարապետությունից, այլև՝ հարցումների կազմակերպման ընտրված եղանակից: Եթե, օրինակ՝ SQL հիման վրա նախագծված հենքերում կիրառված տրամաբանությունը կառուցվում է միասնական հարցումների օրինաչափություններով պայմանավորված գրոհների հաշվի առնմամբ, ապա NoSQL հենքերում դրանց ավելանում է նաև հենքի կառավարման համակարգի տրամաբանության անխափան ապահովումը և ծառայությունների մատուցման հնարավոր չարաշահումների կանխարգելման երաշխավորումը:

Տվյալների տեղաբաշխման և կառավարման տեսանկյունից՝ տվյալների հենքերը տարանջատվում են հետևյալ հիմնական մոդելների:

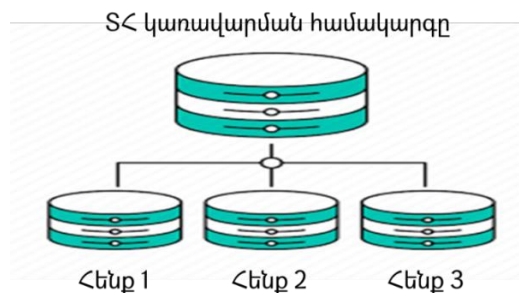
Տվյալների կենտրոնացված հենքեր: Այս դեպքում կազմակերպության բոլոր տվյալները պահվում են մեկ միասնական թվային պահոցում, որի հասանելիությունն իրականացվում է կազմակերպության տեղեկատվական ցանցի միջոցով: Տվյալների հենքի նման կազմակերպումն ապահովում է պարզեցված կառավարում և տեղեկատվության ամբողջական վերահսկում մեկ ադմինիստրատորի կողմից:



Նկ. 33. Տվյալների կենտրոնացված հենքի սխեմա:

Չնայած թվարկված առավելություններին, այս ճարտարապետությունը ստեղծում է ծախսողման մեկ կետի սպառնալիք (single point of failure), ինչը նշանակում է, որ հենքի կառավարման համակարգի խափանումը կարող է հանգեցնել տվյալների մասնակի կամ ամբողջական կորստի և, միաժամանակյա բազմաթիվ հարցումների մշակման ոչ պատշաճ կազմակերպմամբ պայմանավորված՝ ծառայությունների մատուցման այլ ծախսողումների:

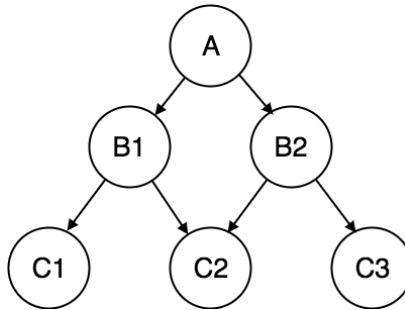
Տվյալների բաշխված հենքեր: Այս ճարտարապետությունը վերացնում է ծախսողման մեկ կետի սպառնալիքը, որը հատուկ է կենտրոնացված ճարտարապետությամբ հենքերին: Տվյալները և դրանց կառավարումը տարածվում են փոխկապակցված թվային պահոցների վրա՝ մեկ միասնական պահոցի փոխարեն: Այս հենքերում տվյալները կարող են աշխարհագրորեն տեղակայվել օգտագործողներին առավել մոտ վայրերում, ինչը նպաստում է հենքի արագագործության մեծացմանը: Բաշխված բնույթը, որի շնորհիվ հնարավոր է տվյալների կրկնօրինակումը, ապահովում է ամբողջական հենքի գործունակությունը՝ նույնիսկ նրա որոշ հանգույցների խափանման դեպքում:



Նկ. 34. Տվյալների բաշխված հենքի սխեմա:

Բաշխված հենքերի մեկ այլ առավելություն է՝ հանգույցների միջև զուգահեռ մշակումը, որն, ընդհանուր առմամբ բարելավում է հարցումների արձագանքման ժամանակը:

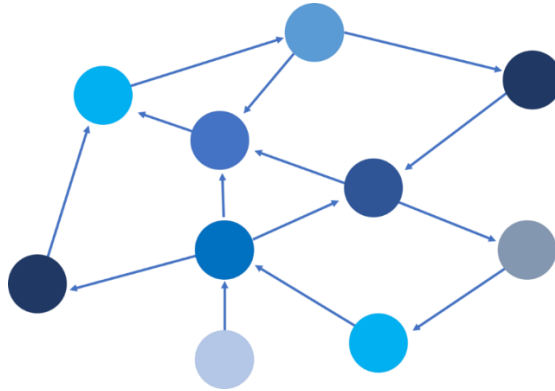
Տվյալների ստորակարգային (hierarchical) հենքեր: Այստեղ տվյալները կազմակերպվում են ծառանման կառուցվածքով, որտեղ յուրաքանչյուր գրառում, բացառությամբ արմատի, ունի մեկ վերին հանգույց՝ ծնող և կարող է ունենալ բազմաթիվ ստորին՝ դուստր-հանգույցներ՝ իրականացնելով «մեկից՝ շատերին» հարաբերությունը ծնող և դուստր հանգույցների միջև (one-to-many relationship):



Նկ. 35. Տվյալների ստորակարգային հենքի սխեմա:

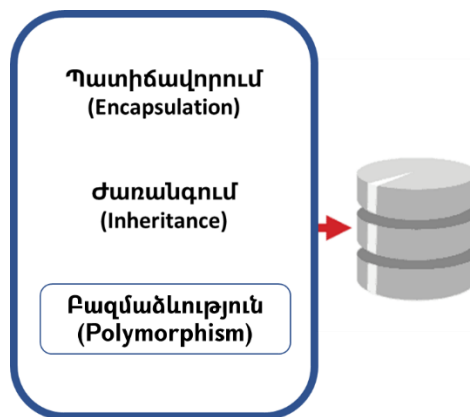
Տվյալների հենքերի ստորակարգային ճարտարապետությունը սովորաբար կոչո է՝ պայմանավորված հանգույցների սևեռված տեղաբաշխմամբ և կապերի նախասահմանվածությամբ: Կազմաձևումները զգալի ազդեցություն են ունենում ամբողջական կառուցվածքի վրա, քանի որ որևէ հանգույցի ավելացումը կամ հեռացումը հանգեցնում է կապակցվածության վերաձևման ծախսատար գործողությունների: Իրականացման լավագույն օրինակներից է՝ ֆայլային համակարգի կազմակերպումը:

Տվյալների ցանցային հենքեր: Ի տարբերություն ստորակարգային հենքերի, ցանցային հենքերն օգտագործում են ճկուն, գրաֆանման տվյալների մոդել, որտեղ հանգույցները կարող են ունենալ բազմաթիվ ծնողական և դուստր հանգույցներ: Սա թույլ է տալիս ստեղծել բարդ հարաբերություններ և արդյունավետ կերպով կառավարել բազմակի կապեր, որոնք համեմատաբար բարդ իրագործելի են կոչո ստորակարգային կամ պարզ ռելացիոն կառուցվածքներում: Ցանցային հենքերն իրականացնում են «շատերից՝ շատերին» հարաբերությունը հանգույցների միջև (many-to-many association):



Նկ. 36. Տվյալների ցանցային հենքի սխեմա:

Տվյալների օբյեկտա-կողմնորոշված հենքեր: Այստեղ տվյալները գրանցվում են իրական աշխարհը և դրա գործառույթները ներկայացնող օբյեկտների տեսքով՝ ներգրավելով օբյեկտա-կողմնորոշված ծրագրավորման լեզուներում սահմանված տվյալների կառուցվածքը և հասկացությունները՝ ինչպիսիք են՝ պատիճավորումը, ժառանգումը և բազմաձևությունը: Օբյեկտների միավորները ներառում են ինչպես տվյալներ (ատրիբուտներ), այնպես էլ դրանց մշակման մեթոդներ:



Նկ. 37. Տվյալների օբյեկտա-կողմնորոշված հենք:

Օբյեկտա-կողմնորոշված հենքերի իրականացումներից են՝ ObjectDB, db4o, Zope Object Database, GemStone/S, Versant Object Database հենքերը:

Տվյալների ամպային հենքեր: Այս հենքերը գտնվում են ամպային հարթակների վրա և ի տարբերություն տեղայնացված ավանդական հենքերի, վերացնում են ֆիզիկական սարքավորումների տեղաբաշխման անհրաժեշ-

տողությունը, առաջարկում են ըստ պահանջի սանդղավորում և գլոբալ հասանելիություն՝ համեմատաբար ցածր գնի դիմաց: Տվյալների ամպային հենքերը տեղակայվում են որպես կառավարվող ծառայություններ, որտեղ մատակարարները կառավարում են ենթակառուցվածքը և սպասարկումը և հավասարապես աջակցում են ինչպես SQL, այնպես էլ NoSQL տվյալների մոդելներին:



Նկ. 38. Տվյալների ամպային հենք:

Տվյալների ամպային հենքերում տվյալների հասանելիությունը և կառավարումն իրականացվում են համացանցի միջոցով՝ անկախ օգտագործողների աշխարհագրական դիրքից: Իրականացումներից են՝ ռեկացիոն Amazon RDS (Relational Database Service), Google Cloud SQL, Microsoft Azure SQL Database և Google Spanner հենքերը և ոչ ռեկացիոն MongoDB Atlas, Amazon DynamoDB, Azure Cosmos DB և Google Cloud Firestore հենքերը:

Տվյալների հենքերի վերոհիշյալ բոլոր իրականացումներն անխտիր ենթակա են կիբեռգրոհների, որոնք մի մասն ընդհանրական է (DoS/DdoS, ներքին սպառնալիքներ, զրոյական օրվա խոցելիություններ, պահուստային պատճենների վնասում և այլն), իսկ մյուս մասը՝ պայմանավորված է կոնկրետ հենքի ճարտարապետությամբ: Ստորև ներկայացնում ենք տվյալների հենքերի դեմ տիպային կիբեռգրոհների դրսևորումները:

Ռեկացիոն հենքերի (SQL) դեմ տիպային գրոհներ: Սրանք դասակարգվում են մի քանի հիմնական տեսակների՝ կախված գրոհի մեթոդից և համապատասխան API արձագանքից:

- **In-band SQL:** Այստեղ գրոհողն օգտագործում է հաղորդակցման միևնույն ուղին՝ թե՛ հարձակումը սկսելու, և թե՛ արդյունքները ստանալու համար

- **Error-based SQL:** Գրոհողը ստիպում է տվյալների հենքին գնեցնել սխալի վերաբերյալ հաղորդագրություններ, որոնց վերլուծությունը կարող է տեղեկություն բացահայտել տվյալների ներդրված կառուցվածքների, աղյուսակների, տողերի կամ սյունակների անվանումների և այլն վերաբերյալ
- **UNION-based SQL:** Գրոհողն օգտագործում է \$UNION\$ օպերատորը՝ սկզբնական հարցման արդյունքների միջոցով այլ աղյուսակներից ստացված տվյալները միավորելու և դրանք կորզելու համար
- **Inferential SQL:** Գրոհի այս տեսակի դեպքում, որպես կանոն, ուղղակիորեն տվյալներ չեն վերադարձվում: Գրոհողը փորձում է տվյալները ստանալ եզրահանգումների միջոցով՝ վերլուծելով սերվերի արձագանքը ճիշտ/սխալ ձևակերպված հարցումներին
- **Boolean-based:** Այս դեպքում գրոհողի ստացված պատասխանները երկու արժեք ունեն՝ կեղծ կամ ճշմարիտ՝ պայմանավորված հարցման պայմանի սխալ կամ ճիշտ լինելով
- **Time-based:** Գրոհողը ներարկում է հրամաններ, որոնց կատարումը որոշակի ժամանակ է ենթադրում, և կախված արձագանքի վրա ծախսված ժամանակից, գրոհողը եզրահանգումների միջոցով փորձում է տեղեկույթ ստանալ տվյալների կամ դրանց բովանդակության մասին
- **Out-of-band SQI:** Գրոհի այս մեթոդը հիմնված է գրոհողի կողմից վերահսկվող սերվերին ցանցային հարցում (օրինակ՝ DNS կամ HTTP) ուղղելու ունակության վրա՝ հարկավոր տեղեկույթի ստացման նպատակով: Սա գրոհը քիչ տարածված, միևնույն ժամանակ՝ համեմատաբար հզոր տեսակ է:

Ռեւաղիոն հենքերի (NoSQL) դեմ տիպային գրոհներ: NoSQL տվյալների հենքերը ևս իրենց կազմաձևումներում կարող են ունենալ անվտանգային թերացումներ. լռելայն գաղտնաբառեր, չվերահսկվող արտոնություններով հասանելիություն, տվյալների չգաղտնագրված փոխանցում և այլն, որոնք թիրախավորվում են գրոհողների կողմից:

- **NoSQL Injection:** Թիրախավորում են ոչ թե SQL հարցման քերականությունը, այլ՝ տվյալների մոդելը և հարցումները սպասարկող հավելվածների (API) տրամաբանությունը: Այս գրոհները հաճախ օգ-

տագործում են հարցումների լեզվի օպերատորներ կամ ներարկում են JavaScript կոդ JSON հարցումների մեջ:

Տվյալների MongoDB հենքի դեմ գրոհներից են՝

- **Operator Injection** (Օպերատորի ներարկման գրոհը): Գրոհողը կարող է շրջանցել նույնականացումը՝ ներարկելով հարցման այնպիսի օպերատոր, ինչպիսին է՝ \$ne (not equal - հավասար չէ), որպեսզի առաջադրի մի պայման, որը մշտապես ճիշտ է:
Օրինակ՝ {"username": {"\$ne": null}, "password": {"\$ne": null}} հարցումը կվերադարձնի առաջին իսկ օգտագործողի տվյալները՝ առանց մուտքագրված գաղտնաբառի իմացության:
- **(Server-side JavaScript code injection, SSJI)** (սերվերի կոդում JavaScript-ի ներարկման գրոհ): Որոշ օպերատորներ, ինչպիսին է \$where-ը, առաջացնում են JavaScript կոդի կատարում սերվերի կոդում, որը կարող է սերվերի աշխատանքի հապաղման (sleep) կամ զգայուն տվյալների նիշ-առ-նիշ արտաբերման պատճառ դառնալ:

Տվյալների հենքերում ոչ խիստ նույնականացման և գործողությունների չվերահսկվող արտոնմանն առնչվող խոցելիությունները ևս թիրախավորվում են գրոհողների կողմից՝ հետևյալ դրսևորումներով:

- **Brute Force / Password Spraying:** Սա ուժային գրոհ է, որի ընթացքում գաղտնաբառի սպառնիչ հատարկում (exhaustive search) է կատարվում մեկ օգտագործողի դեմ (brute force) կամ մեկ գաղտնաբառի փորձարկում՝ բազմաթիվ օգտագործողների դեմ (Password Spraying):
- **Credential Stuffing,** երբ գրոհողն օգտագործում է օգտագործողի վարկաբեկված տվյալների ցուցակները և թվային հավատարմագրերը (credentials)՝ անձնական հաշվեհամար ներխուժելու նպատակով: Գրոհը զուգակցվում է բոտերի տեղադրմամբ՝ գրոհն ինքնաշխատ և սանդղավորված դարձնելու համար: Սույն գրոհի կազմակերպումը հիմնված է այն ենթադրության վրա, որ օգտատերը միննույն գաղտնաբառը կարող է օգտագործել թվային տարբեր ծառայություններից օգտվելու համար:
- **Pass-the-Hash:** Այս դեպքում գրոհողը կողոպտում է օգտագործողի հեշավորված հավատարմագիրը և օգտագործում այն հեռավար սեր-

վերում նույնականացվելու համար՝ առանց գաղտնաբառի իմացության:

Օգտագործողների նույնականացման, ինքնության հաստատման և գործողությունների արտոնման ոչ խիստ մեխանիզմների առկայությունը ևս տվյալների հենքերի կիբեռգրոհների թիրախ են:

Տվյալների հենքերի դեմ ներարկման տիպի գրոհներից պաշտպանության արդյունավետ միջոցներից է պարամետրական հարցումների ներմուծումը, որն առաջարկվել է OWASP ընկերության կողմից:

- **Պարամետրերով օժտված հարցումներ** (Parameterized Queries / Prepared Statements): Սա ամենաարդյունավետ պաշտպանությունն է ներարկման գրոհներից: Այս մոտեցման դեպքում SQL հարցման կառուցվածքը նախապես ուղարկվում է տվյալների հենք, իսկ օգտագործողի կողմից տրամադրված պարամետրերը ներմուծվում են առանձին մուտքագրմամբ: Արդյունքում՝ տվյալների հենքի կառավարման համակարգն այդ պարամետրերը դիտարկում է բացառապես որպես տվյալներ և երբեք ոչ՝ որպես կատարման ենթակա կոդ: Այս մոտեցմամբ արդյունավետորեն չեզոքացվում է ներարկման ցանկացած հրաման:

Դիտարկենք տվյալների հենքերի պաշտպանված և չպաշտպանված ծրագրավորման մի շարք ոճեր:

- **Անապահով կոդի օրինակ:** Հարցման տողը և օգտագործողի տվյալները միավորված են մեկ միասնական հարցման մեջ:

Java code

```
String query = "SELECT account_balance FROM user_data WHERE user_name = " + request.getParameter("customerName");
```

- **Ապահով կոդի օրինակ:** Օգտագործվում է PreparedStatement, որտեղ օգտագործողի տվյալները փոխանցվում են որպես պարամետր:

Java code

```
String query = "SELECT account_balance FROM user_data WHERE user_name =?";
PreparedStatement pstmt = connection.prepareStatement(query);
pstmt.setString(1, custname);
ResultSet results = pstmt.executeQuery();
```

- **Ապահով կողի օրինակ:** Պարամետրով մուտքագրման սկզբունքն օգտագործվում է նաև .NET ընտանիքի ծրագրավորման լեզուներում:

C# (C#.NET):

```
string query = "SELECT account_balance FROM user_data WHERE user_name = @customerName";
OleDbCommand command = new OleDbCommand(query, connection);
command.Parameters.Add(new OleDbParameter("customerName",
CustomerName.Text));
```

Տվյալների հենքերում գործածվում են այսպես կոչված՝ «պահպանվող ընթացակարգեր» (stored procedures)՝ որպես նախապես կոմպիլացված SQL կոդեր: Այդ ընթացակարգերում ներդրված դիսամիկ SQL հարցումները հարմարավետ միջերես են ապահովում հավելվածների համար՝ վերացնելով աղյուսակներին ուղղակի հասանելիության անհրաժեշտությունը: Միևնույն ժամանակ՝ դրանց ոչ պատշաճ նախագծումը կարող է վերոհիշյալ միջերեսների թիրախավորման սպառնալիք առաջացնել:

Ատորև բերված օրինակում հավելվածը կանչում է sp_getAccountBalance անունով պահպանվող ընթացակարգը՝ օգտագործողի տվյալները փոխանցելով որպես պարամետր:

Java code

```
CallableStatement cs = connection.prepareCall("{call sp_getAccountBalance(?)}");
cs.setString(1, custname);
ResultSet results = cs.executeQuery();
```

Տվյալների հենքերի կիբեռպաշտպանական մոտեցումներից են օգտագործողների հասանելիության վավերացումը և տվյալների անաղարտության հավաստիացումները (Input Validation & Sanitization)՝ որպես պաշտպանության երկրորդական, բայց կարևոր մակարդակի վերահսկողական գործառնություններ:

- **Վավերացում (Validation):** Այս կերպ մերժվում են նախասահմանված ձևաչափերից շեղում ունեցող բոլոր կարգի մուտքագրումները՝ արգելելով օրինակ՝ օգտանվանումներում SQL հատուկ նիշերի ներառումը: Մուտքային տվյալները խստիվ վերահսկվում են թույլատրող և արգելող ցուցակների (Allow-list/Deny-list) միջոցով:

- **Տվյալների զտում և հատուկ նիշերի փոխարինում** (Sanitization/Escaping): Զտումը հեռացնում է տվյալներից վտանգավոր կամ արգելված հատվածները, ինչպես օրինակ՝ HTML տողից սկրիպտային թեգերի (tag) հեռացումը կանխարգելում է միջկայքային սկրիպտային (XSS) հարձակումները: Հատուկ նիշերի հակադարձելի փոխարինումը ոչ վնասաբեր համարժեքներով ցուցադրում է դրանք որպես տեքստի մաս, այլ ոչ թե նշագրման լեզվի նիշ (tag):
- **Տվյալների հասանելիության վերահսկողություն՝ դերաբաշխումների հիման վրա** (Role-Based Access Control, RBAC): Այս մոդելում հասանելիության արտոնությունները շնորհվում են ոչ թե ուղղակիորեն օգտագործողներին, այլ կառավարման համակարգում նրանց նախասահմանված դերերին, ինչպիսիք են, օրինակ՝ «Ադմին», «Մենեջեր», «Օգտատեր» և այլն դերակատարումները: Այս մոտեցմամբ նաև զգալիորեն պարզեցվում են ադմինիստրատիվ գործընթացները՝ ներառյալ աուդիտի իրականացումը, բացի այդ՝ նվազում է համակարգի գերբեռնվածության հետևանքով առաջացող սխալների հավանականությունը:
- **Տվյալների տարբերակված հասանելիություն** (Discretionary Access Control, DAC): Այս մոդելում տվյալների սեփականատերն է որոշում, թե ով և ինչպես կարող է հասանելիություն ունենալ դրանց:
- **Նվազագույն արտոնությունների սկզբունքը** (Principle of Least Privilege): Սա կիբեռանվտանգային հիմնարար մոտեցում է, որտեղ յուրաքանչյուր օգտագործողի, հավելվածի, կամ նույնիսկ՝ ադմինիստրատորի, շնորհվում են միայն նվազագույն արտոնություններ՝ որքան որ անհրաժեշտ է տվյալ գործառնության իրականացման համար և ոչ երբեք՝ ավելին: Այս կերպ հնարավոր է դառնում մեղմել այն վնասը, որը գրոհողը կարող է հասցնել դրանց վարկաբեկման արդյունքում:
- **Տվյալների գաղտնագրում:** Տեղեկատվական համակարգերում տվյալները գտնվում են երեք՝ հանգստի (պահուստավորման), փոխանցման (ցանցով) և պրոցեսորի կողմից մշակման վիճակներում: Գաղտնագրումը կրիտիկական է հանգստի և փոխանցման վիճակում գտնվող տվյալները պաշտպանելու համար: Մշակման ընթացքում տվյալները պաշտպանվում են գաղտնի հաշվարկներ իրականացնող տիրույթներում (enclaves), մեկուսացված սերվերներում,

կամ՝ այդ նպատակով առանձնացված այսպես կոչված «ավազարկ-դերում» (sandboxes):

Տվյալների հենքերում հանգստի վիճակում գտնվող տվյալների գաղտնագրման գործածական մեխանիզմներից են՝

- **Տվյալների թափանցիկ գաղտնագրումը** (Transparent Data Encryption, TDE): Հանգստի վիճակում գտնվող տվյալների պաշտպանության նպատակով գաղտնագրվում են հենքի բոլոր նիշքերը՝ ներառյալ պահուստայինները: Մեխանիզմը կոչվում է թափանցիկ, քանի որ գաղտնագրումն ու ապագաղտնագրումն այստեղ իրականացվում են իրական ժամանակում՝ առանց տվյալների հասանելիության հավելվածների ծրագրային կոդերում որևէ փոփոխություն կատարելու անհրաժեշտության: Այս կերպ նաև պաշտպանվում է կորսված կամ գրոհողների կողմից կորզված կրիչներում և կոշտ սկավառակներում հայտնաբերված տեղեկույթը: Գաղտնագրային բանալիների կառավարումը այս մեխանիզմների կարևորագույն բաղադրիչներից է՝ տվյալների չարտոնված ապագաղտնագրումը կանխելու համար:
- **Սյունակների մակարդակի գաղտնագրում** (Column-level encryption): Գաղտնագրում է միայն զգայուն տվյալներ պարունակող սյունակների բովանդակությունը (օրինակ՝ օգտագործելով PostgreSQL-ի pgcrypto մոդուլը):
- **Ֆայլային համակարգի/սկավառակի գաղտնագրում** (Filesystem/Disk Encryption): Տվյալների գաղտնագրումը կատարվում է օպերացիոն համակարգի մակարդակում:

Տվյալների հենքերում տվյալների պաշտպանությունը փոխանցման ընթացքում պաշտպանում է հենքը գաղտնալսումից կամ «մարդը՝ արանքում» գրոհներից: Այդ նպատակով խորհուրդ է տրվում ներգրավել՝

- **ապահով հաղորդակցական արձանագրություններ**, ինչպիսիք են՝ հավելվածի և տվյալների հենքի սերվերի միջև գաղտնագրված հաղորդակցության հաստատման Transport Layer Security (TLS) կամ Secure Sockets Layer (SSL) հաղորդակարգերը:
- **աուդիտ և մշտադիտարկում** (Auditing and Monitoring)՝ տվյալների հենքում տեղի ունեցող իրադարձությունների հետևման և դրանց ընթացքի ու արդյունքի վերաբերյալ տեղեկույթի գրանցման (լոգավոր-

ման) համար: Նպատակը՝ անվտանգության քաղաքականության խախտումների կամ կասկածելի ակտիվությունների բացահայտումն ու արձագանքումն է: Կատարված գործողությունների և իրադարձությունների լոգավորումը ներառում են հաջողված/անհաջող փորձերը, թիրախային համակարգի սխեմաների (օրինակ՝ \$CREATE\$, \$ALTER\$, \$DROP\$) և արտոնությունների փոփոխությունները (\$GRANT\$, \$REVOKE\$), «զգայուն» որակված տվյալների հասանելիության փորձերը: Աուդիտը շրջափուլային (cyclic) գործընթաց է՝ հետևյալ կրկնվող փուլերով.

- խոցելիությունների հայտնաբերում,
- վերահսկողական միջոցների կիրառում,
- լոգավորման գրանցամատյանների մշտադիտարկում,
- կիբեռդիպվածների վերլուծություն,
- վերահսկողական միջոցների թարմացում, կատարելագործում և
- վերադարձ սկզբնական փուլին:

Տվյալների հենքերում խորհուրդ է տրվում ներդնել նաև՝

- **Տվյալների քողարկման մեխանիզմներ** (Data Masking): Սա զգայուն տվյալներն իրատեսական, միևնույն ժամանակ՝ կեղծ տվյալներով փոխարինելու գործընթաց է: Այն կիրառելի է՝ հատկապես երբ տվյալներն օգտագործվում են հենքերից դուրս հաշվողական միջավայրերում, ինչպիսիք են՝ տվյալների վիճակագրական վերլուծությունն այլ կազմակերպությունների կողմից, թեստավորումը կամ ուսուցումը: Տվյալների քողարկման մոտեցումներից են՝
 - **Տվյալների ստատիկ քողարկումը** (Static Data Masking, SDM): Այս դեպքում ստեղծվում է տվյալների հենքի պատճենն առանց լոգավորման գրանցամատյանների: Քողարկումը կիրառվում է այդ պատճենի վրա՝ մշտապես:
 - **Տվյալների դինամիկ քողարկում** (Dynamic Data Masking, DDM): Այս դեպքում քողարկումը կատարվում է իրական ժամանակում՝ տվյալների հարցման ընթացքում և ապահովում է հենքային տվյալների անփոփոխելիությունը: Դինամիկ քողարկման տեխնոլոգիաներից են՝ փոխարինումը (substitution), տվյալների հերթագայության խախտումը (shuffling), գաղտնագրումը, ընդհանրացումը (կոնկրետ արժեքների փոխարեն դրանց միջակայքերի նշու-

մը), հեշավորումը, գրոյացումը (nulling out) և խմբագրումը (Redaction):

Տվյալների հենքերի անվտանգության սկզբունքները, ինչպիսիք են՝ խիստ նույնականացումը, մուտքի վերահսկողությունը, գաղտնագրումը, աուդիտը և մշտադիտարկումը, համընդհանուր մոտեցումներ են, միևնույն ժամանակ՝ իրականացման տեսանկյունից պահանջում են տարբերակված մոտեցումներ: Կիբեռանվտանգության պատասխանատուները պետք է ոչ միայն գիտենան ինչպես գաղտնագրել տվյալների հենքը, այլև տիրապետեն վերոհիշյալ հմտություններին, օրինակ՝ MS SQL Server-ում թափանցիկ գաղտնագրման իրականացման, PostgreSQL-ում pgcrypto, կամ՝ Oracle-ում Virtual Private database կազմաձևման մասով:

Ստորև ներկայացնում ենք անվտանգային կազմաձևումների իրականացման մոտեցումներ առավել հանրահայտ հենքերի օրինակով:

Microsoft SQL Server-ի անվտանգության ամրապնդում: Այստեղ գործում են ինքնության հաստատման Windows Authentication և SQL Server Authentication մեխանիզմները, որոնցից Windows Authentication-ն արդեն իսկ ինտեգրված է Active Directory-ի հետ՝ ապահովելով հենքի կենտրոնացված կառավարում և ամրապնդված կիբեռանվտանգություն: Այս մեխանիզմների հետ մեկտեղ կիրառվում է՝

- **գաղտնագրում,**
- **թափանցիկ գաղտնագրում**
- **զգայուն տվյալների մշտապես գաղտնագրում** (always encrypted): Սա առավել հատիկավորված լուծում է, որը պաշտպանում է տվյալների աղյուսակներում զգայուն տվյալներ ներկայացնող սյունակները: Գաղտնագրման բանալիները պահվում են հաճախորդի կողմում, ինչը նշանակում է, որ անգամ տվյալների հենքի ադմինիստրատորները հասանելիություն չունեն բաց տվյալներին:
- **հենքի կառավարման SQL Server Management Studio (SSMS)** միջավայրի Server Audit և Server Audit Specification գործիքակազմերի միջոցով իրադարձությունների մշտադիտարկում՝ հաջողված և անհաջող հասանելիության կամ չարտոնված գործողությունների վերլուծության նպատակով: Բացի այդ՝ խոցելիությունների գնահատման ներկառուցված SQL Vulnerability Assessment և տվյալների բացահայտման ու դասակարգման SQL Data Discovery & Classification գոր-

ծիքակազմերը հնարավորություն են ընձեռում հայտնաբերել և վերացնել անվտանգային թերացումները:

Oracle Database-ի պաշտպանության կազմակերպումը: Այս հենքի անվտանգության հիմքում ընկած են լոգային գաղտնաբառերի մերժումը, գաղտնաբառի բարդության քաղաքականությունների կիրառումը և ոչ ակտիվ հաշվեհամարների արգելափակումը: Կիրառվում է նաև՝

- **մուտքի վերահսկողություն՝** Virtual Private Database (VPD) միջոցով: Սա հզոր գործիքակազմ է՝ տվյալների աղյուսակների տողային մակարդակի (row-level) անվտանգության ապահովմամբ: VPD-ն գործում է՝ օգտատերերի հարցումներին դինամիկ կերպով ավելացնելով \$WHERE\$ պայմանը՝ հարցումների համատեքստի հիման վրա. դեր, IP հասցե և այլն:
- **աուդիտ՝** Oracle-ի Unified Audit Trail միջոցով, որը թույլ է տալիս կազմաձևել և կառավարել աուդիտի համապարփակ գրառումներ՝ կենտրոնացված կերպով հետևելով տվյալների հենքի ակտիվությանը:
- **լրացուցիչ անվտանգային ծրագրակազմի ներդրում,** ինչպիսիք են՝ Oracle Advanced Security, Database Firewall և Data Masking գործիքակազմերը:

PostgreSQL հենքի անվտանգության կազմակերպումը: Այստեղ իրականացվում է՝

- **հաղորդակցության անվտանգության կարգաբերում** pg_hba.conf նիշքի պատշաճ կազմակերպմամբ, որով սահմանվում են թույլատրված հոսթերը, օգտագործողները և դրանց նույնականացման խիստ մեխանիզմները
- **գաղտնաբառի գաղտնագրում՝** scram-sha-256 օգտագործմամբ
- **աուդիտ** pgAudit միջոցով: Հաշվի առնելով այն իրողությունը, որ PostgreSQL-ն օժտված է ստանդարտ լոգավորման սահմանափակ հնարավորություններով, pgAudit-ը տրամադրում է հաղորդակցության սեսիայի և օբյեկտի մակարդակի առավել մանրամասն աուդիտ:
- **տվյալների գաղտնագրված փոխանցում** SSL/TLS և pgcrypto մոդուլների գործարկման միջոցով

Տվյալների հենքերի կիբեռանվտանգային լուծումների ներդնման և հարկավոր հմտությունների զարգացման նպատակով ստեղծվում են լաբորատոր միջավայրեր: Օրինակ՝ OWASP Juice Shop-ի վրա գործնական միջավայր ստեղծելու համար կատարվում է հետևյալը:

- **Juice Shop-ի տեղակայում** հասանելի եղանակներով, որոնցից ամենադյուրին և մատչելի համարվող տարբերակներն են՝ Heroku կամ Docker իրականացումները:

1. Տեղակայում Heroku-ի միջոցով: Այդ նպատակով՝

- Այցելեք OWASP Juice Shop-ի GitHub պահոց (repository)
- Սեղմեք "Deploy to Heroku" կոճակը
- Հետևեք հրահանգներին՝ ձեր սեփական Juice Shop հավելվածը Heroku-ի ամպային հարթակում ստեղծելու համար

2. Տեղակայում Docker-ի միջոցով

- Համոզվեք, որ ձեր համակարգչում տեղադրված է Docker
- Բացեք տերմինալը և կատարեք հետևյալ հրամանը

Bash

```
docker run --rm -p 3000:3000 bkimminich/juice-shop
```

- Բացեք ձեր զննարկիչը և այցելեք <http://localhost:3000> հղմամբ:

ԳԼՈՒԽ 7. Վեբ սպառնալիքներ և Վեբ կիբեռանվտանգություն

Համաշխարհային սարդոստայնը, որը հայտնի է որպես WWW (World Wide Web) կամ պարզապես՝ վեբ, հիպերհղումներով փոխկապակցված հանրային կայքէջերի (վեբ էջերի) և ռեսուրսների հսկայական հավաքածու է, որը հասանելի է համացանցի միջոցով: Վեբ էջերը ստեղծվում են հիպերտեքստի նշագրման լեզուների (Hypertext Markup Language, HTML) միջոցով, որոնց հասանելիությունն ապահովվում է զննարկիչ (web browser) կոչվող հավելվածների միջոցով: Վեբը նույնը չէ, ինչ համացանցը: Վեբը համացանցի վրա կառուցված բազմաթիվ ծրագրակազմերից մեկն է:

Ինչպես ամեն մի տեղեկատվական համակարգ, վեբը նույնպես ենթակա է կիբեռգրոհների^{####24)}: Հաքերները սքանավորում են համացանցը՝ խոցելի կայքեր հայտնաբերելու և դրանք շահագործելու նպատակով:

Վեբ կիրառությունների դեմ ամենատարածված գրոհներից են՝

- ֆիշինգը,
- ապագադոնագրման դիմաց փրկագնի պահանջը,
- SQL ներարկումը,
- միջկայքային սկրիպտինգը (XSS),
- ծառայության մերժման DoS և DDoS գրոհները,
- վարկաբեկված նույնականացումը,
- գաղտնաբառերի գուշակումը,
- թվային հավատարմագրերի կողոպուտը,
- «մարդը՝ արանքում» գրոհները,
- վնասաբեր կայքեր այցելելու արդյունքում «drive-by-download» գրոհը,
- բանկային քարտերի դեմ գրոհը (carding) և այլն:

Բանկային քարտերի դեմ գրոհը կիբեռհանցագործության տեսակ է, որով կողոպտվում են բանկային քարտերի տվյալները, ինչպիսիք են՝ քարտի համարը, գործողության ժամկետը, CVV/CVC (Card Verification Value) կոդը, որոնք հետագայում օգտագործվում են անօրինական գործարքների կամ հանցագործներին վաճառելու նպատակով:

####24) Stuttard, D., & Pinto, M., The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (2nd ed.). Wiley, 2011.

Hoffman, C., Hacking Exposed Web Applications (3rd ed.). McGraw-Hill Education, 2010.

Owasp Foundation, OWASP Top 10 - 2021. OWASP Foundation, 2021.

Ըստ OWASP ընկերության կողմից կատարված վերլուծությունների, արդի վեբ համակարգերին հատուկ կիբեռռիսկերը դասակարգվում են A01-A10 համարակալմամբ տարակարգերի (OWASP Top 10): Ռիսկերի վերաբերյալ համապարփակ ուսումնասիրություն կարելի է կատարել՝ այցելելով հետևյալ կայքէջը^{§§§§§25}:

Ստորև ներկայացնում ենք վերոհիշյալ կիբեռռիսկերը՝ համառոտ շարադրանքով:

A01: Broken Access Control (Մուտքի վերահսկողության վարկաբեկում): Սա վեբ կիրառությունների ամենակրիտիկական ռիսկն է: Նման ռիսկեր առաջանում են, երբ կիրառությունը պատշաճ կերպով չի սահմանում նույնականացված օգտագործողների արտոնությունները, որի հետևանքով համակարգ ներթափանցած գրոհողը հասանելիություն է ձեռք բերում նաև այլ օգտատերերի հաշիվներին՝ դրանից բխող վնասաբեր հետևանքներով:

A02: Cryptographic Failures (Գաղտնագրային խափանումներ): Այս գրոհը նախկինում հայտնի էր որպես «Զգայուն տվյալների բացահայտում»: Այն վերաբերում է տվյալների պաշտպանության հետ կապված խափանումներին՝ տվյալների հանգստի (at-rest) կամ պահուստավորման վիճակներում և փոխանցման ընթացքում (in-transit): Ռիսկերը կարող են պայմանավորված լինել ինչպես գաղտնագրային թույլ ալգորիթմների օգտագործմամբ, այնպես էլ՝ գաղտնաբառերի և գաղտնագրային բանալիների անապահով պահպանմամբ կամ տվյալների փոխանցման չգաղտնագրված HTTP արձանագրության օգտագործմամբ:

A03: Injection (Վնասաբեր կոդի ներարկում): Այս գրոհը համատեղում է ներարկման տարբեր տեխնոլոգիաներ՝ ներառյալ Cross-Site Scripting (XSS) և SQL Injection (SQLi) գրոհները:

A04: Insecure Design (Անապահով նախագծում): Սա կիբեռգրոհի համեմատաբար նոր տարակարգ է, որը սկեռվում է ոչ թե ծրագրավորման ոճերի, այլ վեբ կիրառությունների նախագծման և դրանց ճարտարապետության թերացումների վրա: Անապահով նախագիծը կիբեռռիսկերի մի ամբողջ դաս կարող է գոյացնել՝ անվտանգային պատշաճ վերահսկողության բացակայության կամ անարդյունավետության հետևանքով: Սա ընդգծում է կիբեռապառնալիքների մոդելավորման (threat modeling) և անվտանգ նախագծման սկզբունքների կարևորությունը կիրառությունների և հավելվածների մշակման վաղ փուլերում:

§§§§§25) <https://owasp.org/www-project-top-ten>

A05: Security Misconfiguration (Անվտանգության ոչ պատշաճ կազմաձևում): Այս ռիսկն առաջանում է անապահով, լռելյայն, թերի կամ ad hoc (հարափոփոխ) կազմաձևումների, տվյալների հանրային հասանելիության ամպային պահուստների, սխալ կազմաձևված HTTP խորագրերի (headlines), սխալների վերաբերյալ անհարկի մանրամասնության հաղորդագրությունների տրամադրման և օպերացիոն համակարգերի ոչ պատշաճ կազմաձևման հետևանքով:

A06: Vulnerable and Outdated Components (Խոցելի և հնացած բաղադրիչներ): Ժամանակակից կիրառությունները և հավելվածները հաճախ կառուցվում են բազմակի օգտագործման (reusable) բաղադրիչների (գրադարաններ, շրջանակներ) հիման վրա, որոնք արտադրվում են կազմակերպությունից դուրս երրորդ ընկերությունների կողմից և կարող են չբավարարել կիբեռանվտանգային պահանջվող չափանիշներին: Այդպիսի բաղադրիչները զրոյական օրվա խոցելիությունների առաջացման ռիսկ են, և կազմակերպությունները ժառանգում են այդ ռիսկերը՝ դառնալով կիբեռգրոհի թիրախ:

A07: Identification and Authentication Failures (Նույնականացման և վավերացման խափանումներ): Այս ռիսկը կապված է օգտագործողների ինքնության, նույնականացման և հաղորդակցության սեսիայի կառավարմանն առնչվող գործառույթների ոչ պատշաճ կազմակերպման հետ: Արդյունքում, գրոհողը հնարավորություն է ստանում կողոպտել գաղտնաբառերը, գաղտնագրային բանալիները կամ սեսիայի թղթենները և անձնավորվել որպես օրինական օգտագործող:

A08: Software and Data Integrity Failures (Ծրագրային ապահովման և տվյալների ամբողջականության խափանումներ): Գրոհների այս տարակարգը պայմանավորվում է ծրագրային ապահովման թարմացումների, կրիտիկական տվյալների և այսպես կոչված՝ «Շարունակական ինտեգրում/շարունակական տեղակայում» (Continuous Integration/Continuous Deployment, CI/CD) նախագծային շղթայի հերթական փուլերում տվյալների և ընթացակարգերի ամբողջականության վավերացման բացակայությամբ: Արդյունքում, գրոհողը կարող է ներդնել չարտոնված կոդ կամ վնասաբեր ծրագրակազմ՝ աննկատորեն ներդրվելով համակարգի լիազորված թարմացումների գործընթացներում:

A09: Security Logging and Monitoring Failures (Անվտանգության լոգավորման և մշտադիտարկման խափանումներ): Պայմանավորված են լոգավորման գրանցամատյանների մշտադիտարկման ոչ պատշաճ կազմակերպ-

մամբ, որի արդյունքում անվտանգային դիպվածները կարող են մնալ չհայտնաբերված կամ հայտնաբերվել ուշացումով: Սա գրոհողների համար բավարար ժամանակի հնարավորություն է՝ համակարգում աննկատորեն ներդրվելու և ավելի խորը ներթափանցելու համար:

A10: Server-Side Request Forgery (SSRF) (Հարցումների կեղծում սերվերի կողմում): Այս խոցելիությունները թույլ են տալիս գրոհողին ստիպել սերվերին կատարել ներքին կամ արտաքին ռեսուրսների չարտոնված հարցումներ, օրինակ՝ ներքին ցանցի սքանավորում, զգայուն տվյալներին հասանելիություն, ներքին ծառայությունների հետ փոխազդում, որոնք օրինական օգտագործման պարագայում պաշտպանված են արտաքին աշխարհից և արտաքին միջամտություններից:

OWASP Top 10 ցանկը և բովանդակությունը պարբերաբար թարմացվում են՝ արտացոլելու սպառնալիքների և հարձակման վեկտորների հարափոփոխ բնույթը և տիրույթը: Ծրագրակազմ մշակողների և անվտանգության պատասխանատուների համար OWASP Top 10-ը ծառայում է որպես անփոխարինելի ուղեցույց և ստուգաթերթ՝ մշակված կիրառությունները վնասաբեր գրոհներից պաշտպանելու համար: OWASP Top 10-ի զարգացումն ինքնին պատմում է վեբ կիրառությունների մշակման բնույթի դինամիկայի մասին: Եթե նախկինում ոլորտի փորձագետները սևեռվում էին ծրագրային կոդի մակարդակի թերացումների վրա (XSS կամ SQL ներարկում), ապա ցանկի վերջին թարմացումները ցուցում են ավելի համակարգային, ճարտարապետական և գործընթացային թերությունների արձագանքման անհրաժեշտությանը և ծրագրային ապահովման մշակման ողջ կենսացիկլի ընթացքում անվտանգային համապարփակ մոտեցումների ներդնմանը (secure by design and by default):

Ստորև բերված են Վեբ կիրառությունների որոշ գրոհների կազմակերպման օրինակներ:

WEB SQL Injection (SQLi): Այս ներարկումը տեղի է ունենում, երբ գրոհողը օգտագործողի վեբ ձևաթղթի մուտքագրման պատուհանում տեղադրում է վնասաբեր SQL հարցում, որի դեմ վերահսկողական կամ հակազդման միջոցներ նախատեսված չեն: Արդյունքում, վեբ հավելվածը չի կարողանում տարբերակել օրինական տվյալները հարձակվողի կողմից ներարկված SQL հրամաններից:

Ենթադրենք, հավելվածն օգտագործում է հետևյալ կոդը՝ օգտատիրոջը նույնականացնելու համար.

```
SELECT * FROM users WHERE username = ' + userName + ' AND  
password = ' + password + ';
```

Այս դեպքում, եթե օգտատերը, որպես userName մուտքագրի ' OR '1'='1', իսկ գաղտնաբառի դաշտում՝ որևէ գաղտնաբառ, ապա արդյունարար հարցումը կունենա հետևյալ տեսքը.

```
SELECT * FROM users WHERE username = " OR '1'='1' AND password =  
'...';
```

Քանի որ '1'='1' արտահայտությունը մշտապես ճիշտ է, ապա WHERE պայմանը բավարարվում է, և հարցումը վերադարձնում է users աղյուսակի բոլոր տողերը՝ շրջանցելով օգտագործողի նույնականացման ստուգումը:

2015 թվականին բրիտանական հեռահաղորդակցային իսկա TalkTalk-ը դարձավ կիբեռհարձակման զոհ, որի հիմքում ընկած էր SQL ներարկման խոցելիությունը: Գրոհողները՝ շահագործելով TalkTalk-ի վեբ կայքերից մեկում բացահայտված թերությունը, կարողացան մուտք գործել ընկերության տվյալների հենք: Արդյունքում, կողոպտվեցին մոտ 157,000 հաճախորդների անձնական տվյալները՝ անուններ, հասցեներ, ծննդյան ամսաթվեր, էլեկտրոնային փոստի հասցեներ և հեռախոսահամարներ: Ավելի մտահոգիչ էր այն, որ գրոհողները կարողացան ձեռք բերել նաև 15,656 հաճախորդի բանկային հաշվեհամարները և տեսակավորման կոդերը (sort codes): Միջադեպի հետևանքով TalkTalk-ի ուղղակի և անուղղակի ֆինանսական վնասը գնահատվեց մոտ 77 միլիոն ֆունտ ստերլինգ: Բացի այդ, Մեծ Բրիտանիայի Տեղեկատվության հանձնակատարի գրասենյակը (ICO) ընկերությանը տուգանեց 400,000 ֆունտ ստերլինգով՝ հաճախորդների տվյալների անվտանգությունն ապահովելու հարցում անփութություն դրսևորելու համար:

2019 թվականին տեղի ունեցավ Capital One-ի 106 միլիոն հաճախորդի տվյալների արտահոսք: Գրոհը կարողացել էր համադրել մի քանի խոցելիությունների դեմ գրոհները՝ ներառյալ անվտանգության սխալ կազմաձևումը և սերվերի կողմից հարցումների կեղծման հնարավորությունը: Հարձակման հիմնական թիրախը վեբի կրակե պատնեշի ոչ պատշաճ կազմաձևումն էր, որը հնարավորություն տվեց գրոհողներին ստիպել Capital One-ի սերվերներից մեկին, որը գտնվում էր Amazon Web Services (AWS) ամպային միջավայրում, կատարել AWS մետատվյալների ներքին ծառայությանն (metadata service) ուղղված հարցում: Ի թիվս այլ կարևորագույն տվյալների, այդ ծառայությունը պարունակում է հույժ կարևոր տեղեկույթ թվային հավատար-

մագրերի վերաբերյալ, որոնք պայմանավորում են տվյալ սերվերի դերակատարությունը: Ձեռք բերելով հավատարմագրերը՝ գրոհողը հասանելիություն ստացավ Amazon S3 պահուստներին, որտեղ պահպանվում էին միլիոնավոր հաճախորդների վարկային դիմումների տվյալներ:

Վեբ կիրառությունների կիբեռանվտանգության ապահովման արդի մոտեցումները ներգրավում են տեղեկատվական համակարգերին, ցանցերին, տվյալների հենքերին և ամպային տեխնոլոգիաների պաշտպանության ավանդական մեթոդներ՝ ներառյալ հակավիրուսային ծրագրակազմերը և ներխուժումների հայտնաբերման ու կանխարգելման մեխանիզմները:

Վեբ կիրառությունների կիբեռպաշտպանության ապահովման նպատակով ներմուծվեցին համապատասխան կրակե պատնեշ (Web Applications Firewall, WAF) և URL գտում տեխնոլոգիական լուծումները՝ Վեբ հավելվածի և համացանցի միջև HTTP/HTTPS երթևեկի վերահսկման և կասկածելի կամ վնասաբեր կայքերի զտման համար:

Այսպես կոչված՝ «**Բովանդակության անվտանգության քաղաքականությունը**» (Content Security Policy CSP) կիբեռանվտանգային մեկ այլ հզոր մեխանիզմ է, որը թույլ է տալիս վեբ կայքի ադմինիստրատորին վերահսկել, թե ինչ ռեսուրսներ (սկրիպտներ, ռճեր, պատկերներ և այլն) կարող է ներբեռնել օգտագործողի զննարկիչը տվյալ էջի համար: Այս քաղաքականության հիմնական նպատակը XSS գրոհների կանխարգելումն ու դրանց հետևանքների մեղմացումն է: Ստեղծելով վստահելի բովանդակության աղբյուրների «թույլատրելի ցուցակ» (allow-list), գործիքակազմը հրահանգում է զննարկչին անտեսել կամ արգելափակել ցանկացած ռեսուրս, որը ներառված չէ ցուցակում: Անգամ, եթե գրոհողին հաջողվում է վնասաբեր սկրիպտ ներարկել, եթե այդ սկրիպտի աղբյուրը թույլատրելիների ցուցակում չէ, բրաուզերը պարզապես չի կատարում այն:

Դիտարկենք CSP վերնագրի օրինակ:

HTTP

Content-Security-Policy: default-src 'self'; img-src 'self' https://example.com; script-src 'self' https://apis.google.com;

Սույն վերնագիրը սահմանում է հետևյալ կանոնները.

- default-src 'self': Լռելայն թույլատրվում են բացառապես միևնույն ծագման (միևնույն դոմենին պատկանող) ռեսուրսները
- img-src 'self' https://example.com: Պատկերները կարող են բեռնվել միևնույն դոմենից կամ example.com-ից:

- `script-src 'self' https://apis.google.com`: Սկրիպտները կարող են բեռնվել միևնույն դոմենից կամ `apis.google.com`-ից:

CSP-ն կարող է նաև արգելափակել ներդրված (inline) սկրիպտների և ոճերի օգտագործումը, ինչպես նաև պոտենցիալ վտանգ ներկայացնող JavaScript ֆունկցիաների կիրառումը, ինչպիսին է՝ օրինակ՝ `eval()`-ը:

Ամերիկյան NIST Cybersecurity Framework (CSF) շրջանակը ԱՄՆ Ստանդարտների և տեխնոլոգիաների ազգային ինստիտուտի (NIST) կողմից մշակված ուղեցույց է, որը նախատեսված է ճանաչելու, ըմբռնելու, կառավարելու և նվազեցնելու կիբեռանվտանգային ռիսկերը: Ուղեցույցը չի պարտադրում այս կամ այն գործիքակազմի կամ տեխնոլոգիաների ներդնում: Փոխարենը՝ այն տրամադրում է արդիական շրջանակ, որը կարող է հարմարեցվել ցանկացած կազմակերպության կարիքներին՝ անկախ դրանց չափից, գործարար բնույթից և կիբեռանվտանգության հասունության մակարդակից:

CSF-ը կառուցված է հինգ հիմնական գործառնությունների (Functions) շուրջ.

1. **Identify** (Նույնականացնել): Գնահատել կազմակերպության ակտիվները, գործարար միջավայրը, կառավարման կառուցվածքը և տիպային կիբեռռիսկերը:
2. **Protect** (Պաշտպանել): Ներգրավել համապատասխան անվտանգային միջոցներ՝ կրիտիկական ենթակառուցվածքների և ծառայությունների անխափան մատուցումն ապահովելու համար:
3. **Detect** (Հայտնաբերել): Իրականացնել գործողություններ՝ կիբեռանվտանգային դիպվածների ժամանակին հայտնաբերման համար:
4. **Respond** (Արձագանքել): Կատարել գործողություններ՝ հայտնաբերված կիբեռանվտանգային դիպվածների հետևանքների վերացման կամ մեղմացման համար:
5. **Recover** (Վերականգնել): Իրականացնել գործողություններ՝ կիբեռդիպվածներից հետո գործարար միջավայրի ամբողջական կարողությունները վերականգնելու և կազմակերպության կիբեռանվտանգային հասունությունն ավելի բարձր մակարդակի տեղափոխման ուղղությամբ:

ԳԼՈՒԽ 8. Ամպային ծառայությունների կիբեռնականություն

Ամպային ծառայությունները (Cloud service) տրամադրվում են կազմակերպության պարագծից դուրս տեղակայված և այլ կազմակերպությունների տնօրինության ու կառավարման ներքո գործող հաշվողական միջավայրերի կողմից: Ծառայությունների հասանելիությունն ապահովվում է հեռավար եղանակով՝ համացանցի միջոցով, իսկ հաշվողական ռեսուրսները տրամադրվում են ըստ պահանջի՝ առանց սեփական հաշվողական ենթակառուցվածքը ձեռք բերելու, տեղակայելու և սպասարկելու անհրաժեշտության: Ամպային ծառայությունների առավելություններից են՝ ըստ պահանջի սանդղավորումը, ճկունությունը և ծախսարդյունավետությունը ինչպես կազմակերպությունների, այնպես էլ՝ անհատների համար^{*****26}: Ամպային ծառայությունների տրամադրման օրինակ է Google Cloud-ը, որը ներառում է Google-ի բոլոր ամպային ծառայությունները. Google Workspace, Google Maps, Google docs և այլն, որոնք հասանելի են ամբողջ աշխարհում:

Ամպային ծառայությունների հիմնական տեսակներն են՝

- **Ենթակառուցվածքը որպես ծառայություն** (Infrastructure as a service, IaaS)՝ հաշվողական, ցանցային և պահուստային ռեսուրսների հեռավար տրամադրմամբ
- **Հիմնահարթակը որպես ծառայություն** (Platform as a service, PaaS)՝ կիրառությունների և հավելվածների մշակման հնարավորությամբ՝ առանց դրանց իրագործման համար պահանջվող ենթակառուցվածքների ճարտարապետության յուրացման անհրաժեշտության (ինչպես IaaS դեպքում): Ամպային հիմնահարթակները կարող են ներառել սերվերներ, տվյալների հենքեր, օպերացիոն համակարգեր և ծրագրային ապահովում, ցանցեր և նորարարական տեխնոլոգիաներ՝ արհեստական բանականության, մեծածավալ տվյալների վերլուծության, բլոկչեյնի, իրերի համացանցի և այլն շահագործման հնարավորությամբ:
- **Ծրագրային ապահովումը որպես ծառայություն** (Software as a service, SaaS)՝ վեբ զննարկչի միջոցով հասանելիության տրամադրմամբ, որը հնարավորություն է ընձեռում օգտագործողներին խու-

^{*****26}) Erl, T., Mahmood, Z., & Puttini, R., Cloud Computing: Concepts, Technology & Architecture. Prentice Hall, 2013.

սահել իրենց սեփական համակարգիչների վրա ծրագրակազմ տեղադրելու հետ կապված ռիսկերից, ինչպիսիք են՝ չարտոնված ծրագրերի տեղադրումը, պահանջվող հաշվողական հզորությունների պակասը, օպերատիվ և կոշտ սկավառակների հիշողության կամ պրոցեսորի հաճախության անբավարարությունը և այլն: Ամպային ծրագրակազմի ենթակառուցվածքը՝ ներառյալ սերվերները, օպերացիոն համակարգերը, տվյալների հենքերը, դրանց սպասարկումը և թարմացումները կատարվում են ծառայության մատակարարի կողմից:

Ամպային միջավայրերը ևս ենթակա են կիբեռսպառնալիքների և նույնպես կարող են ունենալ խոցելիություններ: Դրանցից են՝

- **ոչ պատշաճ կազմաձևումները** (misconfigurations), որոնք կարող են պատահել ամպային կիրառությունների և համակարգերի անվտանգության կարգավորումներում, ինչպիսիք են՝
 - սերվերների բաց միացքների առկայությունը,
 - արտոնյալ ինքնությունների գործողությունների վերահսկման թերացումները,
 - մշտադիտարկման թերացումը կամ իսպառ բացակայությունը,
 - տվյալների անապահով պահուստավորումը (Simple Storage Service, S3 bucketu),
 - լռելյայն գաղտնաբառերի և հավաստագրերի (certificates) օգտագործումը,
 - ամպային ծառայությունների սխալ կազմաձևումները և այլն: McGraw Hill ընկերության տեղեկացմամբ՝ 22 TB ծավալով տվյալներ՝ ներառյալ ուսանողների գնահատականներ և անձնական տեղեկություններ, արտահոսեցին 2015 թվականից ի վեր՝ AWS S3 bucket-ի սխալ կազմաձևման հետևանքով:
- **ամպային ենթակառուցվածքների տեսանելիության բացակայությունը:** Ընկերությունները՝ գործարար կարիքներից ելնելով, կարող են համադրել ամպային տեխնոլոգիաներ տարբեր մատակարարներից՝ ստեղծելով բարդ, փոխկապակցված, մշտապես զարգացող և միևնույն ժամանակ ոչ լիարժեքորեն վերահսկվող հաշվողական միջավայրեր: Այդ միջավայրերի տեսանելիության բացակայությունը կարող է խոցելիությունների առաջացման և դրանցից հետևող կի-

բեռագրողների պատճառ դառնալ: Օրինակ՝ հայտնի է դարձել, որ ամպային սխալ կազմաձևումների հետևանքով՝ Toyota Japan-ը գրեթե 10 տարի հասանելիություն է ունեցել իր 2.15 միլիոն հաճախորդների անձնական և տրանսպորտային միջոցների տվյալներին:

- **հասանելիության կառավարման թույլ մեխանիզմների կիրառումը**, որի հետևանքով օգտագործողները կամ ծառայությունները հասանելիություն են ստանում չարտոնված ռեսուրսներին
- **ներքին սպառնալիքները**՝ պայմանավորված ամպային ենթակառուցվածքների և դրանցում հասանելիության մակարդակների ոչ պատշաճ վերահսկողական մեխանիզմներով, որի արդյունքում լիազորված օգտագործողները կարող են չարաշահել իրենց արտոնությունները և մնալ չհայտնաբերված:
- **անապահով կիրառությունները** (APIs) ամպային հավելվածների և տվյալների կենտրոնների միջև հաղորդակցության միջոց են և կարող են խոցելիությունների պատճառ դառնալ, ինչպիսիք են, օրինակ՝ օբյեկտի մակարդակի ինքնության հաստատման խաթարումը (Broken object level authorization, BOLA.), երբ որևէ ծրագրային կիրառություն հասանելիություն է տրամադրում տվյալների օբյեկտներին՝ հիմնվելով օգտատիրոջ դերի, բայց ոչ նրա լիազորությունների կամ արտոնությունների վրա:
- **գրոյական օրվա խոցելիությունները**, որոնք կարող են հայտնաբերվել և շահագործվել կիբեռագրողների կողմից:
- **ստվերային համակարգերը և կիրառությունները** (Shadow IT), որոնք օգտագործվում են կազմակերպության ներսում՝ առանց վերադասի կողմից հստակ հավանության կամ վերահսկողության:
- **գաղտնագրման թերացումները կամ բացակայությունը** տվյալների փոխանցման կամ պահուստավորման ընթացքում, որը կարող է դրանց չլիազորված օգտագործման կիբեռսպառնալիքներ առաջացնել:

Կիբեռագրողների կողմից շահագործվող ամպային ամենատարածված թիրախներից են՝

- **վարկաբեկված վկայագրերը** (compromised credentials), երբ կողոպտված օգտանունները, գաղտնաբառերը և ներցանցային օգտագործման բանալի-պարամետրերն օգտագործվում են զոհի ամպային միջավայր մուտք գործելու և այնտեղ լիազորված գործիքաշար գործարկելու նպատակով:

- **Հաշվեհամարի վերահսկողությունը** (account takeover), որի դեպքում կիբեռգրոհողը չարտոնված հասանելիություն է ստանում զոհի ամպային հաշվեհամարին և փորձում է կողոպտված հավատարմագրերն օգտագործել այլ ծառայություններին հասանելիություն ստանալու համար՝ հույս ունենալով, որ օգտագործողն անխոհեմորեն միևնույն հավատարմագրով է գրանցված այլ ծառայություններում (credential stuffing): Սեսիայի կողոպտումը (session hijacking) զոհի հաշվեհամարի վերահսկողության մեկ այլ հնարավորություն է՝ սեսիայի բանալի-բառերի (session token) զավթման միջոցով:
- **«Մարդը ամպի արանքում»** (Man-in-the-middle in cloud, MitMC) գրոհները, երբ գրոհողը միջամտում է երկու ամպային ծառայությունների օգտատերերի կամ մեկ օգտագործողի և ամպային ծառայության միջև հաղորդակցությանը՝ գաղտնալսելով կամ նենգափոխելով շրջանառվող տվյալները:

Ամպային կիբեռանվտանգության գործիքակազմը ենթադրում է՝

- օգտագործողների ինքնության և համակարգերի հասանելիության կառավարում (IAM),
- բազմագործոն նույնականացում,
- անվտանգային կանոնավոր գնահատումներ և խոցելիությունների հայտնաբերման միջոցառումներ,
- անվտանգային թարմացումների՝ «կարկատանների», կառավարում (Patch Management), որոնք մեծապես նպաստում են կիբեռխոցելիությունների հետևանքների մեղմացմանը,
- տվյալների պաշտպանություն գաղտնագրային մեթոդների համատարած կիրառում,
- կիբեռպառնալիքների հայտնաբերման, ցանցային անվտանգության և մշտադիտարկման պատշաճ գործիքակազմի ներգրավում, համակարգերի տեսանելիության հավաստիացում,
- անձնական տվյալների պաշտպանության (General data protection requirements, GDPR) միջազգայնորեն ընդունված կանոնակարգերի անշեղորեն կատարում:

Ամպային ամենախոշոր մատակարարները՝ AWS-ը, Azure-ը և Google Cloud-ը ներդրել են և շարունակաբար կատարելագործում են իրենց կիբեռանվտանգային գործիքակազմը մարտահրավերներին արձագանքելու և օգտագործողների վստահությանն արժանանալու համար:

ԳԼՈՒԽ 9. Կիբեռանվտանգության ժամանակակից մոտեցումներ

Կիբեռանվտանգության ոլորտի վերջին զարգացումները ներառում են կազմակերպչական և տեխնոլոգիական նորագույն լուծումներ՝ արհեստական բանականության (ԱԲ) և մեքենայական ուսուցման օգտագործմամբ: Խելացի համակարգչային ծրագրերն ունակ են «սովորել» տվյալներից և հայտնաբերել տեղեկատվական համակարգի բնականոն աշխատանքի շեղումները շուրջօրյա ռեժիմում:

ԱԲ ներդրմամբ հնարավորություն է ստեղծվում ինքնաշխատ կերպով իրականացնել հետևյալ գործառույթները.

- **Կիբեռպառնալիքների հայտնաբերում:** ԱԲ համակարգերը վերլուծում են միլիոնավոր ազդանշաններ և կարող են հայտնաբերել տեղեկատվական համակարգերի չլիազորված հասանելիության փորձերը կամ նիշքերի նենգափոխումները:
- **Կանխատեսողական վերլուծություն:** ԱԲ համապատասխան գործիքաշարը կարող է հայտնաբերել կիբեռ-խոցելիություններ տեղեկատվական համակարգում:
- **ԱԲ հիման վրա մշակված կիբեռգործիչների կանխում:** Հաքերները նույնպես օգտագործում են ԱԲ մեթոդներ՝ կեղծ տվյալների գեներացման, դրանց տարածման, այլոց անձնավորման և այլ չարամիտ նպատակներով: ԱԲ ներդրված գործիքակազմը՝ սքանավորելով վայրկյանում հազարավոր էլեկտրոնային նամակներ, կարող է հայտնաբերել, թե որոնք են օրինակ՝ ֆիշինգային փորձերը, և զտել դրանք, ինչը վեր է մարդկային հնարավորություններից:

Այսպես կոչված՝ «Զրոյական վստահությունը» ժամանակակից կիբեռանվտանգային մեկ այլ ժամանակակից մոտեցում է, որը կառուցված է պարզ գաղափարի վրա. «Երբեք մի վստահիր, այլ՝ մշտապես ստուգիր»: Կիբեռանվտանգային ավանդական մոտեցումը կարող է ենթադրել տեղեկատվական համակարգի ադմինիստրատորների անսահմանափակ հասանելիություն բոլոր ռեսուրսներին: Զրոյական վստահության մեխանիզմների ներդրմամբ սպասարկող անձնակազմի լիազորությունները տրամադրվում են բացառապես իրենց վերագրված գործառույթների շրջանակներում՝ բացառելով այդ լիազորությունների ցանկացած չարաշահում:

Զրոյական վստահության հիմնարար սկզբունքներից են՝

- օգտագործողների՝ առանց բացառության, ինքնության հավաստիացումը
- նվազագույն արտոնությունների սահմանումը՝ անհրաժեշտ նվազագույն հասանելիության շնորհմամբ:
- հասանելիության մշտադիտարկման կազմակերպումը և կատարված գործողությունների լրգավորումը՝ համակարգի բնականոն աշխատանքի շեղումների հայտնաբերման նպատակով:

ԳԼՈՒԽ 10. Համակարգչային հետախուզություն և հետաքննություն

Համակարգչային հետախուզությունն ու հետաքննությունը, որը հայտնի է նաև որպես թվային հետաքննություն, համակարգչային դատաբժշկական գիտություն կամ կիբեռդատաքննություն, համատեղում է համակարգչային գիտությունը և իրավական դատաբժշկական փորձագիտությունը՝ թվային ապացույցներ հավաքելու այնպիսի եղանակով, որն ընդունելի է դատարանում:

Թվային ապացույցները հավաքագրվում են տեղեկատվական համակարգերում ներդրված թվային և էլեկտրոնային սարքերից, համակարգիչներից, սմարթֆոններից և սերվերներից և վերջնակետային սարքերից:

Համակարգչային հետաքննության հիմնական գործառույթներն են՝

- **Թվային սարքերի կամ կրիչների նույնականացումը**, որոնք կարող են հետաքննությանը նպաստող տվյալներ պարունակել: Սարքերը մեկուսացվում են տեղեկատվական համակարգից և տեղադրվում հետաքննության համար առանձնացված տեղամասում՝ հետազոտության անցկացման և տվյալների հավաքագրման համար:
- **Թվային ապացույցների հավաքագրումը**: Փորձագետները թվային տվյալներ են հավաքագրում հետաքննության առարկա աղբյուրներից, այդ թվում՝ համակարգիչների կոշտ սկավառակներից, ցանցային սերվերներից, տվյալների հենքերից, բջջային և տեղեկատվական համակարգի վերջնակետային սարքերից: Արդյունքում ստեղծվում է պահպանվող տվյալների պատկերը՝ բիրթ առ բիրթ պատճենն այնպես, որ անվտանգ կերպով պահպանվեն և՛ պատկերը, և՛ բնօրինակը՝ դրանք փոփոխվելուց կամ ոչնչացումից պաշտպանելու համար: Հավաքագրվում են երկու տեսակի տվյալներ՝ մշտական տվյալներ, որոնք պահվում են սարքի տեղայնացված կոշտ սկավառակի վրա, և անկայուն տվյալներ, որոնք գտնվում են գրանցամատյաններում, քեշ և օպերատիվ հիշողության մեջ: Անկայուն տվյալները մշակվում են առանձնահատուկ զգուշությամբ, քանի որ դրանք ժամանակավոր են և կարող են ընդմիջտ անհետանալ՝ հավաքագրման ընթացքում միջամտությունների կամ սնուցման ընդհատման հետևանքով:
- **Հավաքագրված տվյալների պահպանումը**: Հավաքագրված ապացույցների պահպանումը խիստ կարևոր գործընթաց է: Նպատակը

այդ ապացույցների ամբողջականության և անաղարտության պահպանումն է, որի շնորհիվ կանխարգելվում է հավաքագրված տվյալների ցանկացած նենգափոխում՝ հնարավորություն ընձեռելով ստեղծել պահպանման շղթա և հստակ պատմություն:

- **Հավաքագրված տվյալների և ապացույցների դասակարգումը:** Տվյալները և համապատասխան ապացույցները դասակարգվում են ըստ կարևորության:
- **Հավաքագրված տվյալների վերլուծությունը:** Այս փուլում հետաքննողները վերլուծում են առկա տվյալները՝ համապատասխան թվային ապացույցներ հայտնաբերելու նպատակով: Վերլուծության արդյունքում կարող են հայտնաբերվել միտումնավոր կամ ոչ միտումնավոր հեռացված ֆայլեր, ցանցային զննարկման պատմություն, էլեկտրոնային նամակներ և թաքցված ֆայլեր՝ հնարավոր կիրքեռգրոհների մեկնարկման մտադրությամբ: Թաքնված տվյալներ կամ մետատվյալներ բացահայտելու նպատակով կիրառվում են մասնագիտացված այնպիսի մեթոդներ, ինչպիսիք են՝ իրական ժամանակի վերլուծությունը, որը գնահատում է դեռևս գործող համակարգերի անկայուն տվյալները և թաքնավերլուծական գործիքաշարը, որի օգնությամբ հայտնաբերվում են տեղեկատվական համակարգի օրինական ռեսուրսներում անօրինական կերպով քողարկված բովանդակությունները: Հետաքննիչներն օգտագործում են մասնագիտացված գործիքակազմ՝ կիրքեռանվտանգային միջադեպի պատմությունը վերլուծելու, պատճառված վնասի չափը գնահատելու և թվային սարքի տեղեկատվական ամբողջականությունը վերականգնելու համար:
- **Հետաքննողական հաշվետվության ներկայացումը:** Փորձագետները կազմում են պաշտոնական զեկույց, որտեղ ուրվագծվում են իրենց վերլուծությունը, վերլուծության արդյունքները, եզրակացությունը և հավանական խորհրդատվությունը:
- **Հետաքննության իրավական ընդունելիությունը:** Հետաքննության ողջ գործընթացն իրականացվում է ապացույցների հավաստիության և պատշաճ փաստաթղթավորման ապահովմամբ:

Համակարգչային հետաքննության հավելյալ ուսումնասիրության համար կարելի է այցելել հետևյալ կայքէջը ^{††††††27)}

^{††††††27)} <https://www.eccouncil.org/cybersecurity-exchange/computer-forensics/what-is-digital-forensics>

ԳԼՈՒԽ 11. Կիրեռանվտանգության օրենսդրություն և էթիկա

Կիրեռանվտանգության մասին Հայաստանի Հանրապետության օրենքի ամբողջական բովանդակությանը կարելի է ծանոթանալ հետևյալ հղմամբ^{#####28)}:

Օրենքում բերված են՝

- օգտագործվող հիմնական հասկացությունները,
- կիրեռանվտանգության ապահովման սկզբունքները,
- կիրեռանվտանգության ոլորտի պետական կառավարումը և կարգավորումը,
- կիրեռանվտանգության ապահովման կանոնները,
- կիրեռանվտանգության ապահովումը՝ ներառյալ.
 - կիրեռամիջադեպերի հայտնաբերումը, կանխարգելումը և լուծումը,
 - ինքնավար մարմնի կողմից տեղեկատվական համակարգի կամ կրիտիկական տեղեկատվական ենթակառուցվածքի օգտագործումը կամ դրանց հասանելիությունը սահմանափակելը,
 - կիրեռամիջադեպերի գրանցամատյանը,
- կենսական նշանակության ոլորտների կիրեռանվտանգության ապահովման առանձնահատկությունները,
- կիրեռանվտանգության ապահովման ծառայություն մատուցողներին ներկայացվող պահանջները,
- կիրեռանվտանգության աուդիտը,
- օրենքի և դրա հիման վրա ընդունված իրավական ակտերի պահանջների կատարման նկատմամբ վերահսկողությունը, մշտադիտարկումը և պատասխանատվությունը:

Կիրեռանվտանգության մասին պետականորեն սահմանված դրույթների ուսումնասիրությունը չափազանց կարևոր է, քանի որ այն երաշխավորում է բոլոր տեսակի կիրեռանվտանգային միջոցառումների իրականացման իրավական հիմնավորումը և այդ միջոցառումներն իրականացնող անձանց օրենքով պաշտպանվածությունը:

Կիրեռանվտանգության էթիկան բարոյական սկզբունքների համադրությամբ է, որով կարգավորվում են ոլորտի մասնագետների վարքագիծը և գոր-

#####28) <https://www.e-draft.am/projects/8642/about>

ծելակերպը՝ տվյալները, համակարգերը և տեղեկատվական ցանցերը վնասից և չարտոնված հասանելիությունից պաշտպանելու համար:

Կիբեռանվտանգության ոլորտում աշխատանքը ենթադրում է էթիկական կանոնների իմացություն և պահպանում՝ ներառյալ.

- ազնվությունը և նվիրվածությունը սեփական պարտականություններին,
- անձնական և այլոց կիբեռպաշտպանված կենսակերպի նկատմամբ հարգանքը,
- օգտագործողների անձնական տեղեկատվության պատշաճ կազմակերպումը,
- վստահված զգայուն տեղեկատվության գաղտնիության պահպանումը (data confidentiality),
- օգտագործողների համաձայնության ստացումը նրանց տվյալների որևէ կերպ հասանելիության և մշակման համար (principle of user data privacy),
- բոլոր օգտագործողների համար հավասար վերաբերմունքի դրսևորումը և կողմնակալությունից խուսափումը՝ կիբեռանվտանգային միջոցառումների իրականացման ողջ ընթացքում,
- հաշվետունակությունը՝ ընդունված որոշումների և կատարված գործողությունների վերաբերյալ,
- անվնաս գործելակերպը,
- թվային համակարգերին չարտոնված հասանելիությունից խուսափումը,
- էլեկտրոնային հարթակների և ցանցային ենթակառուցվածքների բնականոն վարքագծից շեղման իրազեկումը վերադասին,
- անփութության կամ չիմացության հիմքով կիբեռանվտանգային խնդիրների մասին իրազեկումը վերադասին՝ առանց թաքցնելու սեփական սխալները,
- ձեռնարկության ներքին գործառույթների գաղտնիության խստիվ պահպանումը և տվյալների կողոպուտի բացառումը այն գիտակցությամբ, որ ներքին մատնիչները և հաքերները նույնպիսի հանցագործներ են և ենթակա են քրեական համարժեք պատասխանատվության:

Կիբեռանվտանգության օրենսդրությունն ու էթիկան լավագույն ուղեցույցն են կիբեռանվտանգության մասնագետների համար: Մինչ մասնագի-

տական կրթվածությունը և ձեռք բերված կարողունակությունները նպաստում են աշխատաշուկայի պահանջները բավարարելուն, կիբեռանվտանգության օրենսդրության և էթիկայի նորմերի տիրապետումը օգնում է անհատին ինքնահաստատվել իր դիրքում՝ աշխատանքային կարիերայի պահպանման և շարունակական վերելքի երկարաժամկետ հնարավորությամբ:

ՀԱՎԵԼՎԱԾ

Ինքնաստուգման առաջադրանքների տիպեր՝ ԳԼՈՒԽ 4-ի վերաբերյալ

1. Նախընտրած տեսաշարի կիրառմամբ ծրագրավորեք իրապես պատահական թվային գեներատոր և գեներացրեք ինչպես դրական, այնպես էլ բացասական, կոտորակային և բիթային հաջորդականություններ:
2. Ծրագրավորեք «միանգամյա նոթատետր» գաղտնագիրը:
3. Ծրագրավորեք «RC4» գաղտնագիրը:
4. Ծրագրավորեք «Salsa20» գաղտնագիրը:
5. Ծրագրավորեք DES ալգորիթմի բոլոր ռաունդների գաղտնագրային բանալիները:
6. Կիրառեք SHA256 ալգորիթմը՝ տրված նիշքի հեշ արժեքը հաշվելու համար: Այնուհետև կատարելով չնչին բովանդակային փոփոխություններ, հաշվեք փոփոխված նիշքի հեշ արժեքը: Երկու հեշ արժեքների համեմատության և արդյունքի ցուցադրման համար ստեղծեք լրացուցիչ ծրագրային մոդուլ:
7. Օգտվելով առցանց հարթակներից, գաղտնագրեք նիշք AES256 գաղտնագրի միջոցով:
8. Ծրագրավորեք RSA ալգորիթմը՝ նախընտրած երկարության գաղտնագրային բանալիներով:
9. Ծրագրավորեք Դիֆֆի-Հելլմանի բանալիների փոխանցման ալգորիթմը:
10. Ծրագրավորեք «գրոյական գիտելիքի ապացույցի» նախընտրած որևէ ալգորիթմ:
11. Ծրագրավորեք թվային ստորագրության ալգորիթմ:

Ինքնաստուգման առաջադրանքների տիպեր՝ ԳԼՈՒԽ 5-ի վերաբերյալ

WPA2 գաղտնաբառի կայունության ստուգում

Կիրառվող գործիքակազմը: Kali Linux, Wi-Fi ադապտեր, որն աջակցում է «monitor mode» ռեժիմին:

Էթերկական նախազգուշացում: գործիքակազմը կարելի է օգտագործել բացառապես սեփական համակարգչային ցանցի վրա և ուսումնական նպա-

տակներով: Այլոց ցանցերի դեմ գրոհ կազմակերպելն անօրինական է և քրեորեն պատժելի:

1. Ստեղծել Wi-Fi թեժ կետ (hotspot) իր սեփական հեռախոսով՝ պարզ գաղտնաբառով (օրինակ՝ 12345678)
2. Kali Linux-ում, օգտագործելով Airodump-ng, ֆիքսել WPA2 4-way handshake-ը (կապի հաստատման գործընթացը), երբ մեկ այլ սարք միանում է այդ ցանցին:
3. Օգտագործելով Aircrack-ng և բառարան-նիշքը (dictionary file), իրականացնել արտացանցային բառարանային հարձակում (dictionary attack)՝ գաղտնաբառի գուշակման նպատակով:
4. Կրկնել 4-րդ կետի կատարումը՝ օգտագործելով 12 և ավելի նիշանոց տառա-թվային-կետադրական գաղտնաբառով՝ համոզվելու համար, որ գրոհն այս դեպքում անհամեմատ ավելի երկար է տևում կամ դառնում է անհնար:

Ինքնաստուգման հարցերի տիպեր

- Ինչպե՞ս կարող է սովորական օգտատերը սրճարանում պաշտպանվել Evil Twin հարձակումից.
 - անջատելով ինքնաշխատ միացումն անլար ցանցին,
 - օգտագործելով VPN,
 - ուշադրություն դարձնելով զննարկչի (browser) անվտանգության նախազգուշացումներին:
- Արդյո՞ք MAC հասցեների զտումը (MAC filtering) անվտանգության արդյունավետ միջոց է:

Ինքնաստուգման լաբորատոր առաջադրանքների տիպեր ԳԼՈՒԽ 6-ի վերաբերյալ

Լաբորատոր աշխատանք #1: SQL Injection միջոցով մուտքի շրջանցում (login bypass):

- Նպատակն է՝ ստանալ ադմինիստրատիվ մակարդակի հասանելիություն՝ շրջանցելով գաղտնաբառը
- Կիրառվող գործիքակազմը՝ վեբ զննարկիչ
- Քայլ առ քայլ հրահանգները

1. Բացեք Juice Shop հավելվածը և անցեք մուտքագրման էջ (/#/login)
2. Էլեկտրոնային փոստի դաշտում մուտքագրեք՝ admin@juice-sh.op' - կամ ' or 1=1-- 22
3. Գաղտնաբառի դաշտում մուտքագրեք ցանկացած տող
4. Սեղմեք "Log in" կոճակը

Դուք կնկատեք, որ անարգել մուտք եք գործել որպես ադմինիստրատոր:

- Վերլուծություն: Ինչպե՞ս և ինչու՞ է սա հաջողվում: Սա նշանակում է, որ հավանաբար, սերվերի կողմում կատարվող հարցումն ունի հետևյալ տեսքը. \$SELECT * FROM Users WHERE email = '[email_input]' AND password = '[password_hash]': Երբ օգտագործողը ներարկում է նշված payload-ը, հարցումը վերածվում է հետևյալի. \$SELECT * FROM Users WHERE email = " or 1=1--' AND...\$, և, քանի որ \$1=1\$ պայմանը միշտ ճիշտ է, իսկ -- նշանը մեկնաբանության է վերածում հարցման մնացած մասը՝ ներառյալ գաղտնաբառի ստուգումը: Արդյունքում, տվյալների հենքը վերադարձնում է աղյուսակի առաջին օգտագործողի տվյալները, որն ավանդականորեն հենքի ադմինիստրատորն է:

Լաբորատոր աշխատանք #2: Իրականացնել տվյալների ստացում UNION օպերատորի օգտագործման միջոցով:

- Նպատակն է՝ օգտագործելով UNION օպերատորը, հասանելիություն ստանալ SQL տվյալների հենքի մեկ այլ աղյուսակին՝ զգայուն տվյալների՝ օգտանունների, գաղտնաբառերի և այլն դուրս բերման համար
- Կիրառվող գործիքակազմը՝ Վեբ զննարկիչ

Քայլ առ քայլ հրահանգները:

1. Գտեք խոցելի մուտքային դաշտ, օրինակ՝ որոնման դաշտը հիմնական էջում
2. Նախասահմանեք սկզբնական հարցման կողմից վերադարձվող սյունակների քանակը: Դրա համար ներարկեք՝ UNION SELECT NULL- - և ավելացրեք NULL-երի քանակը, մինչև սխալն անհետանա:
3. Այժմ ներարկեք հետևյալ payload-ը՝ Users աղյուսակից տվյալներ ստանալու համար. ' UNION SELECT email, password FROM Users-- 24:

4. Կտեսնեք, որ էջում հայտնվել են 2-րդ կետով ձեր կողմից նախատեսված սյունակների քանակով օգտատերերի էլեկտրոնային փոստերի հասցեները և գաղտնաբառերի հեշերը:

Վերլուծություն: \$UNION\$ օպերատորը թույլ է տալիս գրոհողին միավորել սկզբնական օրինական հարցման արդյունքները երկրորդ՝ վնասաբեր հարցման արդյունքների հետ: Սա հնարավորություն է տալիս դուրս բերել տվյալներ հենքի ցանկացած աղյուսակից, որին հավելվածի օգտատերն ունի հասանելիություն: Ուստի՝ տվյալների հենքի ծրագրակազմը չպետք է ունենա \$SELECT\$ հարցման թույլտվություն Users աղյուսակի նկատմամբ, առավել ևս, եթե նրա միակ գործառույթը տվյալների որոնումն է:

Լաբորատոր աշխատանք #3: Իրականացնել հենքի աուդիտ և վերլուծություն:

- Նպատակն է՝ կազմաձևել աուդիտ՝ տվյալների հենքի սերվերի վրա և վերլուծել գրանցամատյանները կասկածելի ակտիվություն հայտնաբերելու համար: Գործնական օրինակը դիտարկեք PostgreSQL pgAudit ընդլայնման ներգրավմամբ:
- Կիրառվող գործիքակազմը՝ PostgreSQL-ի տեղադրված օրինակ

Քայլ առ քայլ հրահանգները:

1. Միացեք ձեր PostgreSQL տվյալների հենքին և կատարեք. \$CREATE EXTENSION pgaudit;\$ 67՝ տեղադրման նպատակով
2. Կազմաձևման նպատակով postgresql.conf նիշքում ավելացրեք pgaudit-ը shared_preload_libraries պարամետրին: Այնուհետև՝ սահմանեք լոգավորման մակարդակը՝ կատարելով հետևյալ SQL հրամանները. \$ALTER SYSTEM SET pgaudit.log = 'all';\$ և \$SELECT pg_reload_conf();\$ 68: Արդյունքում բոլոր կատարված հրամանները կցուցադրվեն:
3. Ակտիվացման և սիմուլյացիայի նպատակով կատարեք մի քանի գործողություն՝ հաջողված մուտք, անհաջող մուտքի փորձ, \$SELECT\$ հարցում զգայուն աղյուսակից և \$GRANT\$ հրաման:
4. Գրանցամատյանների բովանդակության վերլուծության նպատակով գտեք PostgreSQL-ի լոգ ֆայլը և ուսումնասիրեք AUDIT:-ով սկսվող գրառումները, որոնցից կբացահայտեք և՛ օգտատերերին, և՛ կատարված հրամանները, օբյեկտները, որոնց մուտք է գործվել, և գործողության դասը, օրինակ՝ READ, ROLE: Գրանցամատյանները տրա-

մադրում են օգտագործողների ակտիվության անհերքելի արձանագրում, որը կրիտիկական է կիբեռ-դիպլաժներից հետո հետաքննության անցկացման և իրական ժամանակում համակարգի բնականոն վարքագծից շեղումների հայտնաբերման համար:

Ինքնաստուգման լաբորատոր առաջադրանքների տիպեր ԳԼՈՒԽ 7-ի վերաբերյալ

Լաբորատոր աշխատանք #1. Խոցելիությունների շահագործում: Միջավայրի նախապատրաստում՝ Damn Vulnerable Web Application (DVWA) և Juice Shop տեղադրմամբ:

Juice Shop-ը ժամանակակից, բարդ և դիտավորյալ անապահով վեբ հավելված է, որը գրված է Node.js, Express և Angular տեխնոլոգիաներով: Այն ընդգրկում է OWASP Top 10-ի բոլոր խոցելիությունները և այլ անվտանգային թերություններ: Juice Shop-ը նախագծված է խաղի տեսքով և ուսուցողական հարմարավետ միջավայր է՝ խոցելիություններ հայտնաբերելու և կիբեռ մարտահրավերներ լուծելու համար:

Damn Vulnerable Web Application (DVWA) տեղադրում Docker-ի միջոցով: Docker-ն ամենահարմարավետ և ամենաանվտանգ միջոցն է DVWA-ն գործարկելու համար, քանի որ այն մեկուսացնում է հավելվածը օպերացիոն համակարգից:

1. Համոզվեք, որ ձեր համակարգչում տեղադրված է Docker:

2. Բացեք տերմինալը (Command Prompt, PowerShell կամ Terminal) և կատարեք հետևյալ հրամանը.

Bash

```
docker run --rm -it -p 80:80 vulnerables/web-dvwa
```

Այս հրամանի արդյունքում ներբեռնվում է DVWA Docker image-ը և գործարկվում այն կոնտեյների մեջ, որի 80-րդ միացքը հասանելի է ձեր համակարգչի 80-րդ միացքից:

3. Բացեք ձեր վեբ զննարկիչը և այցելեք <http://localhost> կամ <http://127.0.0.1>:

OWASP Juice Shop տեղադրում Docker-ի միջոցով:

1. Համոզվեք, որ ձեր համակարգչում տեղադրված է Docker:

2. Բացեք տերմինալը և կատարեք հետևյալ հրամանը.

Bash

```
docker run --rm -p 3000:3000 bkimminich/juice-shop
```

Այս հրամանը գործարկում է Juice Shop-ը կոնտեյնների մեջ՝ այն հասանելի դարձնելով ձեր համակարգչի 3000-րդ միացքից:

3. Բացեք ձեր վեբ զննարկիչը և այցելեք <http://localhost:3000>:

Լաբորատոր աշխատանք 1A. Command Injection DVWA-ում:

Քայլ առ քայլ ուղեցույց:

1. Գործարկեք DVWA կոնտեյնները և զննարկչով այցելեք <http://localhost>:
2. Մուտք գործեք՝ օգտագործելով լռելյայն հավատարմագրերը՝ օգտանուն՝ admin, գաղտնաբառ՝ password 29:

3. Ձախ կողմի գործիքացանցից (menu) ընտրեք "DVWA Security" էջը: Անվտանգության մակարդակն (Security Level) ընտրեք "Low" և սեղմեք "Submit" կոճակը:

4. Ձախ գործիքացանկից ընտրեք "Command Injection" էջը:

5. Բացվող պատուհանը նախատեսված է IP հասցե մուտքագրելու և ping հրամանը կատարելու համար: Սկզբում մուտքագրեք 127.0.0.1 և սեղմեք "Submit" կոճակը:

6. Բացվող պատուհանում մուտքագրեք հետևյալը.

127.0.0.1; cat /etc/passwd 30

7. Սեղմեք "Submit" կոճակը:

8. Բացվող էջում՝ բացի ping հրամանի արդյունքից, կտեսնեք նաև /etc/passwd նիշքի բովանդակությունը, որը գտնվում է սերվերի օպերացիոն համակարգում:

Բացատրություն: Անվտանգության "Low" մակարդակում DVWA կողը օգտատիրոջ կողմից մուտքագրված տողն ուղղակիորեն տեղադրում է shell հրամանի մեջ՝ առանց որևէ ստուգման կամ զտման: Նկատենք, որ «կետ-ս-տորակետ» նշանը հավելվածներում հաճախ օգտագործվում է որպես հրամանների բաժանարար: Հետևաբար, համակարգը նախ կատարում է ping 127.0.0.1 հրամանը, այնուհետև՝ ներարկված cat /etc/passwd հրամանը:

Լաբորատոր աշխատանք 1B. SQL Injection Juice Shop-ում՝ SQL ներարկման միջոցով նույնականացման մեխանիզմի շրջանցում:

Մուտք գործելու համար որպես ադմինիստրատոր՝ առանց վերջինիս գաղտնաբառը գիտենալու (Juice Shop-ի "Login Admin" մարտահրավեր):

Քայլ առ քայլ ուղեցույց:

1. Գործարկեք OWASP Juice Shop կոնտեյնները և զննարկչով այցելեք <http://localhost:3000>

2. Կայքէջի վերևի աջ անկյունում սեղմեք "Account", այնուհետև՝ "Login" կոճակները

3. Կցուցադրվի մուտքի ձևաթուղթը՝ "Email" և "Password" դաշտերով:

4. "Email" դաշտում մուտքագրեք հետևյալ SQL ներարկման հրամանը.
' OR 1=1--

5. "Password" դաշտում մուտքագրեք ցանկացած տեքստ (օրինակ՝ "password" կամ "123"):

6. Սեղմեք "Log in" կոճակը:

Արդյունքում, դուք հաջողությամբ մուտք կգործեք համակարգ որպես ադմինիստրատոր (admin@juice-sh.op): Էկրանի վերևում կհայտնվի կանաչ ծանուցում, որը կհաստատի, որ դուք լուծել եք "Login Admin" մարտահրավերը:

Բացատրություն: Juice Shop-ի backend կողմից կառուցվել է SQL հարցում՝ հետևյալ կերպ.

```
SELECT * FROM Users WHERE email = " AND password = "
```

Երբ մենք որպես email մուտքագրում ենք ' OR 1=1--, հարցումը դառնում է.

```
SELECT * FROM Users WHERE email = " OR 1=1--" AND password = '...'
```

Այստեղ.

- առաջին ' մեջբերում նշանը փակում է email դաշտի համար նախատեսված մեջբերումը:
- OR 1=1-ը ավելացնում է պայման, որը մշտապես ճշմարիտ է:
- -ը SQL-ում մեկնաբանության նշան է, որը ստիպում է տվյալների հենքին անտեսել հարցման մնացած մասը՝ ներառյալ գաղտնաբանի ստուգումը:

Քանի որ WHERE պայմանը միշտ ճշմարիտ է, հարցումը վերադարձնում է Users աղյուսակի առաջին տողը, որը, որպես կանոն, ադմինիստրատորի հաշվեհամարն է:

Գրականության ցանկը

1. Ա.Գ. Աբրահամյան, *Հայկական ծածկագրություն*, Երևանի Պետական համալսարանի հրատարակչություն, 1978 թ.
2. B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.)*. John Wiley, 1996:
3. Erl, T., Mahmood, Z., & Puttini, R., *Cloud Computing: Concepts, Technology & Architecture*. Prentice Hall, 2013.
4. R.L. Krutz & R.D. Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Wiley, 2010.
5. J. Erickson, *Hacking: The Art of Exploitation (2nd ed.)*. No Starch Press, 2008.
6. Ali, C., & McAndrew, H., *Generative AI in Cybersecurity: A CISO's Guide to the Future of Threat Detection*. Wiley, 2024.
7. B. Creane & A. Rawal, *Kubernetes Security and Observability: A Holistic Approach to Securing Containers and Cloud-Native Applications*. O'Reilly Media, 2023.
8. A. Gupta & J. Cole, *Practical Cloud-Native Security with Falco: Secure your cloud-native systems with Falco, the open-source cloud-native runtime security tool*. Packt Publishing, 2023.
9. J. F. Kurose & K. W. Ross, *Computer Networking: A Top-Down Approach (8th ed.)*. Pearson, 2021.
10. W. Stallings, *Network Security Essentials: Applications and Standards (6th ed.)*. Pearson, 2017.
11. C. Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems (3rd ed.)*. No Starch Press, 2017.
12. K.Loney, *Oracle Database 19c DBA Handbook*. Oracle Press, 2020.
13. K.Nield., *Getting Started with SQL: A Hands-On Approach for Beginners*. O'Reilly Media, 2021.
14. N.Redkar, *Troubleshooting SQL Server: A Guide for the Accidental DBA*. Apress, 2020.
15. R.Stephens, *SQL Server 2022 Administration Inside Out*. Microsoft Press, 2023.
16. D.Stuttard & M. Pinto, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws (2nd ed.)*. Wiley, 2011.
17. C. Hoffman, *Hacking Exposed Web Applications (3rd ed.)*. McGraw-Hill Education, 2010.
18. Owasp Foundation, OWASP Top 10 - 2021. OWASP Foundation, 2021.

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

Հեղինակների կողմից	3
Ներածություն	6
ԳԼՈՒԽ 1. Կիրեռանվտանգության ներածություն	10
ԳԼՈՒԽ 2. Կիրեռոչիսկեր և կիրեռսպառնալիքի վեկտորներ	17
ԳԼՈՒԽ 3. Գաղտնագրության ներածություն.....	30
ԳԼՈՒԽ 4. Տեղեկատվության գաղտնագրային պաշտպանություն.....	42
ԳԼՈՒԽ 5. Համակարգչային ցանցերի կիրեռանվտանգություն.....	54
ԳԼՈՒԽ 6. Տվյալների հենքերի կիրեռանվտանգություն.....	77
ԳԼՈՒԽ 7. Վեբ սպառնալիքներ և Վեբ կիրեռանվտանգություն	95
ԳԼՈՒԽ 8. Ամպային ծառայությունների կիրեռանվտանգություն	102
ԳԼՈՒԽ 9. Կիրեռանվտանգության ժամանակակից մոտեցումներ	106
ԳԼՈՒԽ 10. Համակարգչային հետախուզություն և հետաքննություն	108
ԳԼՈՒԽ 11. Կիրեռանվտանգության օրենսդրություն և էթիկա.....	110
ՀԱՎԵԼՎԱԾ	113
Գրականության ցանկը.....	120

Չափս՝ 70x100 1/16, թուղթ՝ օֆսեթ, տպ. մամուլ՝ 7.625
Տպաքանակ՝ 85 օրինակ