დოქტორანტის ბიბლიოთეკა

ნუგზარ ამილახვარი, გიორგი ამილახვარი

ინფორმაციული უსაფრთხოება

(პრაქტიკული სამუშაოს მეთოდური მითითებანი)

სტუ-ს „IT კონსალტინგის სამეცნიერო ცენტრი“

საქართველოს ტექნიკური უნივერსიტეტი

ნუგზარ ამილახვარი, გიორგი ამილახვარი

ინფორმაციული უსაფრთხოება

(პრაქტიკული სამუშაოს მეთოდური მითითებანი)

დამტკიცებულია:

სტუ-ს „IT კონსალტინგის
სამეცნიერო ცენტრის“ სარე-
დაქციო კოლეგიის მიერ
ოქმი N7, 15.10.2020

თბილისი - 2020

უაკ 004.5

განხილულია ინფორმაციის უსაფრთხოებისა და საიმედოობის თანამედროვე მოთხოვნების შესაბამისი ძირითადი ასპექტები, მათი რეალიზაციის მეთოდები და ინსტრუმენტული საშუალებები. პრაქტიკული სამუშაოები მაქსიმალუად არის მიახლოებული სახელმწიფო თუ კერძო ორგანიზაციისაში ინდორმაციის დაცვისა და საიმედოობის განსაზღვრისა და პრობლემების აღმოფხვრის რეალურ ამოცანებთან.

პრაქტიკული სამუშაოები წარმოდგენილია ინფორმაციული უსაფრთხოების კონცეფციის ჩამოყალიბება და კომპლექსური დაცვის სისტემის უზრუნველყოფა თანამედროვე ტექნოლოგიებით. შემოთავაზებულია პრაქტიკული სამუშაოს მეთოდური მითითებები ისეთ საკითხებზე, როგორიცაა: პროგრამულ-მათემატიკური კიბერ შეტევები, ინფორმაციის გაჟონვა, Windows და Unix სისტემებში რეაგირების ინსტრუმენტების შექმნა და გამოიყენა, ჰაკერის ინსტრუმენტები და მათთან მუშაობის მეთოდები, პერსონალური მონაცემების დაცვის ორგანიზება და სხვა.

დამხმარე მეთოდური სახელმძღვანელო რეკომენდებულია ინფორმატიკის სპეციალობის დოქტორანტებისათვის, ინფორმაციული და კომუნიკაციური ტექნოლოგიების სფეროში (ICT 0613).

რეცენზენტები:

პროფ. **ო. შონია** (სტუ)
ასოც. პროფ: **კ. ოდიშარია** (სტუ)

ISBN 978-9941-8-2872-0

სასწავლო კურსის მიზანი

მისცეს სტუდენტს ინფორმაციულ უსაფრთხოებაზე ღრმა და საფუძვლიანი ცოდნა. გააცნოს უსაფრთხოების უზრუნველყოფის მხარდამჭერი სისტემები და მათში გამოყენებული მეთოდები, შეასწავლოს კომპიუტერული სისტემების ინფორმაციული უსაფრთხოების უზრუნველყოფის ამოცანები, მონაცემთა კრიპტაციის (შიფრაცია/დეშიფრაცია), ელექტრონული ხელმოწერის, დაიჯესტის გამოთვლის მეთოდები, ალგორითმები, საშუალებები და მათი პრაქტიკული გამოყენება, ღია გასაღებების ინფრასტრუქტურის პროექტირება და რეალიზაცია, უსაფრთხოების მხარდამჭერი სტანდარტები და კანონმდებლობა.

საგნის შესწავლის შედეგად მიღებული ცოდნა და შეძენილი უნარები

1. განსაზღვრავს ინფორმაციული უსაფრთხოების თეორიულ და პრაქტიკულ ამოცანებს.
2. აიდენტიფიცირებს ინფორმაციული უსაფრთხოების მხარდამჭერ სისტემებს და მათში გამოყენებულ მეთოდებს.
3. მიუსადაგებს ინფორმაციულ სისტემებში მონაცემთა კრიპტაციის, ელექტრონული ხელმოწერის, დაიჯესტის გამოთვლის მეთოდებს ალგორითმებსა და საშუალებებს.
4. აანალიზებს ინფორმაციული უსაფრთხოების შექმნისათვის მოთხოვნებს, ინოვაციური მეთოდებით გადაწყვეტებს და აყალიბებს მათზე დამყარებულ დასაბუთებულ დასკვნებს.
5. აფასებს ინფრასტრუქტურის ინფორმაციული უსაფრთხოების პროგრამული, აპარატურული, ინფორმაციის მოძიების დამუშავების და მის საფუძველზე პრობლემის გადაწყვეტის გზებს.
6. განსჯის ინფორმაციული უსაფრთხოების სფეროს პრინციპებს, ფასეულობებსა და ღირებულებებს.
7. აფორმირებს როგორც ზოგადად, ისე ინფორმაციულ უსაფრთხოების გამოყენების ეთიკისა და ღირებულებების მორალის ნორმებს.

საათების განაწილება (სტუდენტის დატვირთვა)

კრედიტების რაოდენობა 8. ლექცია 30 სთ., პრაქტიკული (კომპიუტერულ კლასში) – 30 სთ. დამოუკიდებელი მუშაობა 137 სთ.

პრაქტიკული სამუშაოს თემები:

1. **ინფორმაციის დაცვის კონცეპტუალური მიდგომა.** ინფორმაციის მოწყვლადობის საფრთხეების კონცეფცია და კლასიფიკაცია კრიტერიუმების მიხედვით. რისკების შეფასების რაოდენობრივი და თვისებრივი მახასიათებლები, მათი უპირატესობებისა და უარყოფითი მხარეების შეფასების შესახებ. ინფორმაციის დაცვის ობიექტების კლასიფიკაცია. შეღწევადობის მოდელები. TCP/IP პროტოკოლის მოწყვლადობა, ქსელური შეტევები და მათთან ბრძოლის მეთოდები
2. **პროგრამულ-მათემატიკური კიბერ შეტევები:** ტროას ცხენები, ვირუსები, ქსელური ჭიები. მათი ფუნქციონირების, შეღწევადობის, ძირითადი თვისებების და კლასიფიკაციის თავისებურებები. ინფორმაციის დაცვის სისტემების ანალიზი: ანტივირუსულ პროგრამები, ქსელთაშორისი ეკრანები და შეღწევადობის გამოვლენის სისტემები.
3. **ინფორმაციის გაჟონვის ტექნიკური არხები, მათი ფიზიკური არსი და კლასიფიკაცია.**
   ელექტრომაგნიტური გამოსხივება და რადიო სიგნალების გადაკეტვის საშუალებები: ეკრანირება, დამიწება, ფილტრაცია და ხმაურის ეფექტები.
4. **მრავალფუნქციური ინსტრუმენტები - Netcat და Cryptcat, მათი ოფციები.** ღია კოდის სისტემები - პროგრამები, რომლებიც თან ერთვიან ოპერაციულ გარემოს. მათი გამოყენებით მაგალითები ამოცანის გადაწყვეტისათვის. X Windows სისტემა - Unix- ის GUI-გარემო. პრობლემების მინიმიზება. პორტ-სკანერები დაჰაკვის პრევენციისათვის და პოტენციური მოწყვლადი მიზნების ამოცნობისათვის. სისტემის მუშაობის მექანიზმების განსაზღვრა და ანალიზი პორტის პასუხების მიხედვით: სერვისები (ვებ სერვერი, FTP სერვერი, ფოსტა სერვერი და ა.შ.), ვერსიის ნომრები, ოპერაციული

სისტემა და სხვ. ინფორმაციის მოპოვება და ანალიზი დისტანციურ Windows-ის სისტემაზე.

5. **Windows და Unix სისტემებში "ცოცხალი რეაგირების" ინსტრუმენტების კომპლექტის შექმნა და გამოყენება** ჰაკერების შეტევების გამოძიებისათვის, როდესაც დაჰაკული კომპიუტერი გაითიშება. Windows და Unix სისტემებში ფაილების დათვალიერების ხელსაწყოები. ინფორმაციის დუბლირების კომერციული ინსტრუმენტალური ხელსაწყოები: EnCase, Safeback, SnapBack და Ghost. ინფორმაციის დუბლირების არაკომერციული ინსტრუმენტალური ხელსაწყოები. ინსტრუმენტალური ხელსაწყოები ინფორმაციის გასაანალიზებლად.
6. **პორტების გადამისამართება - ტრაფიკის გადამისამართებისათვის.** ქსელის ნაკადის ანალიზატორები სხვისი ტრაფიკის „მოსმენისათვის". ავტოდარეკვის პროგრამები, რომლითაც იძებნება მოდემები, რომლებიც უსმენენ. ბრანდმაუზერის, ვებ-სერვერის ან რუტერის TCP/IP დასტის შემოწმების მეთოდები და ინსტრუმენტები წვდომის მართვის სიების სანახავად და წვდომის დონის კორექციისათვის. ინსტრუმენტალური ხელსაწყოები კომპიუტერის ქსელში მუშაობის აღდგენისათვის.
7. **Hacking Tools - ჰაკერების ინსტრუმენტები ვებ-აპლიკაციების გატეხვისათვის და მათთან მუშაობის მეთოდები.** პაროლის გატეხვის/შერჩევის მეთოდები. „უკანა კარის" და დისტანციური წვდომის მეთოდები და მათი ანალიზი. პროგრამული კოდების აუდიტის მარტივი მეთოდები. უკვე კოდის დაწერის ეტაპზე ბუფერის გადავსებასთან ბრძოლა - დაუცველობის მთავარი წყაროსთან. სისტემის აუდიტის კომბინირებული მეთოდები. აპლიკაციებისა და სერვისების შეცდომების ანალიზი და აღმოფხვრა.
8. **ინფორმაციული უსაფრთხოება და პროგრამული უზრუნველყოფა ღია კოდით.** ოპერაციული გარემოს უსაფრთხოების დონის ამაღლების მეთოდები. მართვისა და ანალიზის ხელსაწყოები. მოწყვლადობის სკანერები. შეღწევადობის აღმომჩენი სისტემები.
9. **ავტომატიზებულ და არა ავტომატიზებულ პერსონალურ მონაცემთა სისტემები.** განვიხილოთ პერსონალურ მონაცემთა დაცვის სისტემის

შექმნის ძირითადი პრინციპები. პერსონალური მონაცემების საფრთხეების მოდელი. მოწყვლადობებისა და საფრთხეების ცნებები და კლასიფიკაცია. გავრცელებული შეტევები განხილვა პერსონალურ მონაცემებზე. პერსონალური მონაცემების ინფორმაციული სისტემების კლასიფიკაციის კრიტერიუმები, კლასიფიკაციის მიზნები.

10. **პერსონალური მონაცემების დაცვის ორგანიზება, მისი პროცედურები,** მათ შორის დაცვის სიტუაციის შეფასება და მეთოდების განსაზღვრა. პერსონალურ მონაცემთა დაცვის სისტემის შექმნა, მისი ორგანიზაციის ეტაპები განსაზღვრა და სხვა აუცილებელი ორგანიზაციულ-ტექნიკური ზომები. პერსონალურ მონაცემთა დაცვის სისტემის სავალდებულო ქვესისტემები, მათი ფუნქციონალი და მუშაობის პრინციპები.
11. **ინფორმაციული უსაფრთხოების სამი მიზანი.** ინფორმაციის უსაფრთხოებაზე თავდასხმის სახეები, რომლებიც საფრთხეს უქმნის კონფიდენციალურობას, უსაფრთხოების სერვისების იდენტიფიცირებას და მათი კავშირები უსაფრთხოების სამ ამოცანასთან. უსაფრთხოების მექანიზმების განსაზღვრა, რომლებიც უზრუნველყოფენ უსაფრთხოების სერვისებს. დაშიფვრის ორი მეთოდის მექანიზმის რეალიზაცია - კრიპტოგრაფია და სტეგანოგრაფია.
12. **კლასიკური შიფრები:** ტექსტის სტრიქონების გადაადგილება, მარტივი ჩანაცვლება, ვიჟნერის შიფრი. კომპოზიციური შიფრები. თანამედროვე რთული შიფრების შედარება კლასიკურ შიფრებთან. ტრადიციული შიფრები სიმეტრიული გასაღებით, მსგავსება და განსხვავებები თანამედროვე სიმეტრიულ შიფრებთან. თანამედროვე ბლოკური შიფრები: P-ბლოკი და S-ბლოკი. ფაისტელისა და არა ფაისტელის შიფრების კლასების განსხვავება. ბლოკური შიფრების გახსნისათვის შეტევების ორი სახეობა: დიფერენციალური და წრფივი კრიპტოანალიზი.
13. **„შიფრი ნაკადისათვის“ და განსხვავებები სინქრონულ და ასინქრონულ შიფრებს შორის.** წრფივი და არაწრფივი რეგისტრის წანაცვლების უკუკავშირი ნაკადური შიფრების რეალიზაციისათვის. მონაცემთა

დაშიფვრის სტანდარტი (DES - DATA ENCRIPTION STANDARD) - თანამედროვე ბლოკ შიფრი სიმეტრიული გასაღებით: ძირითადი სტრუქტურა; ძირითადი ელემენტების დეტალები; გასაღებების რეგისტრაცია რაუნდებისთვის; DES-ის ანალიზი.

14. **მონაცემთა დაშიფვრის დახვეწილი სტანდარტი (AES - ADVANCED ENCRYPTION STANDARD)** - თანამედროვე ბლოკის შიფრი სიმეტრიული გასაღებით, რომელსაც შეუძლია შეცვალოს DES. ასიმეტრიულ-გასაღებითი კრიპტოგრაფიული სისტემები: RSA (RIVERST SHAMIR ADLEMAN); რაბინი, ელ-გამალი, ECC (Elliptic Curve Cryptosystem), რომელიც დაფუძნებულია ელიფსური მრუდეების მეთოდზე.
15. ინფორმაციული უსაფრთხოების სრული ანალიზი და ინოვაციური იდეების გენერირება.

პრაქტიკული სამუშაოს შესრულების ნიმუში

პრაქტიკული სამუშაოების ჩატარების საილუსტრაციოდ განვიხილავთ ორ ამოცანას:

1. პრაქტიკული სამუშაო N 11. პროგრამების დაცვა უნებართვო გამოყენებისაგან ინფორმაციის მატარებელზე მიბმით;
2. პრაქტიკული სამუშაო N 12. მონაცემთა სიმეტრიული დაშიფვრა

პრაქტიკული სამუშაო N 11
პროგრამების დაცვა უნებართვო გამოყენებისაგან ინფორმაციის მატარებელზე მიბმით

სამუშაოს მიზანი: პროგრამებზე დაცვის დაყენების უნარ-ჩვევების მიღება ინფორმაციის მატარებელზე (გარე მეხსიერების მოწყობილობაზე) მიბმით.

1. ძირითადი დებულებები.

ორგანიზაციის ყველა პროდუქტი საჭიროებს უკანონო წვდომისგან დაცვას, განსაკუთრებით ეს აქტუალურია ინფორმაციული ტექნოლოგიის სფეროსათვის. როგორც პრაქტიკა აჩვენებს, ინფორმაციის აბსოლუტური დაცვა არ არსებობს. რაც არ უნდა რთული და ძვირადღირებული იყოს დაცვის მექანიზმები, მათი ეფექტურობა პირობითია. არსებული რეალობის გათვალისწინებით, მარტივი და იაფი დამცავი საშუალებები უფრო მოთხოვნადია, რომელიც დამუშავებულია და დაყენებულია თვით პროდუქტის მწარმოებლის მიერ და მიმართულია კვალიფიციური მომხმარებლების უკანონო ქმედებების წინააღმდეგ.

დაცვის იდეა ემყარება ინფორმაციის მატარებელის მახასიათებლების გამოყენებას: აპლიკაციის ამოქმედებისას, ხორციელდება დაკავშირე-ბული გარე მოწყობილობის არსებობის შემოწმება კონკრეტული სერიული ნომრით და მასზე გაშვებული პროგრამის მოძიება. დაცვის ეს მეთოდი იძლევა საშუალებას დაუბრკოლებლად დაკოპირდეს აპლიკაცია, რომლის დროსაც არ ირღვევა ინფორმაციის წაკითხვა-ჩაწერის არსებული სტანდარტები და მოწყობილობაზე წაკითხვა-ჩაწერისას არ არსებობს სპეციალური მოთხოვნები. პროგრამების კვალიფიციური გამტეხის მუშაობის გართულებისათვის შესაძლებელია პროგრამის სხვადასხვა ადგილიდან მსგავსი პროცედურის შემოწმების ორგანიზება.

პრაქტიკული დავალების შესრულებისას უნდა იქნეს გამოყენებული პროგრამირების გარემო პარალელური კურსების მიხედვით (მაგალითისათვის აქ განხილულია Delphi). აპლიკაციების შემუშავების დაჩქარებისათვის Delphi პირდაპირ მუშაობს API-Windows ფუნქციებთან. მოწყობილობის შესახებ ინფორმაციის მისაღებად გამოიყენება WinAPI-ს

ფუნქცია GetVolumeInformation. Delphi-ს HELP-ში (Windows SDK განყოფილება) მოცემულია ამ ფუნქციის პარამეტრების შემდეგი აღწერა:

```
BOOL GetVolumeInformation(
    LPCTSTR lpRootPathName,     // address of root directory of the file system
    LPTSTR lpVolumeNameBuffer,// address of name of the volume
    DWORD nVolumeNameSize,          // length of lpVolumeNameBuffer
    LPDWORD lpVolumeSerialNumber,     // address of volume serial number
    LPDWORD lpMaximumComponentLength, // address of system's maximum
    //filename length LPDWORD
    lpFileSystemFlags,                // address of file system flags
    LPTSTR lpFileSystemNameBuffer,      // address of name of file system
    DWORD nFileSystemNameSize         // length of lpFileSystemNameBuffer
);
```

ფუნქციის პარამეტრებია:

lpRootPathName – ძირითადი ფოლდერის მისამართი ფაილურ სისტემაში;

lpVolumeNameBuffer – ტომის სახელის მისამართი;

nVolumeNameSize – lpVolumeNameBuffer-ის სიგრძე;

lpVolumeSerialNumber – ტომის სერიული ნომრის მისამართი;

lpMaximumComponentLength – ფაილური სისტემაში მისამართების სიგრძის მაქსიმალური მნიშვნელობა;

lpFileSystemFlags – ფაილური სისტემის დროშების მისამართი;

- **FS_CASE_IS_PRESERVED** – დისკზე ფაილის სახელის შენახვისას ფაილური სისტემა იმახსოვრებს რეგისტრს;
- **FS_CASE_SENSITIVE** – ფაილური სისტემის მგრძნობიარობა ფაილის სახელის რეგისტრთან;

– **FS_UNICODE_STORED_ON_DISK** – ფაილური სისტემის მხარდაჭერა სახელების UNICODE-ში დაწერაზე;

– **FS_PERSISTENT_ACLS** – ფაილური სისტემის მხარდაჭერა წვდომის სიებზე (NTFS);

– **FS_FILE_COMPRESSION** – ფაილური სისტემის ფაილების დონეზე მხარდაჭერა შეკუმშვაზე;

– **FS_VOL_IS_COMPRESSED** – ფაილური სისტემის ტომის დონეზე მხარდაჭერა შეკუმშვაზე.

lpFileSystemNameBuffer – ფაილური სისტემის სახელის მისამართი;

nFileSystemNameSize – lpFileSystemNameBuffer-ის სიგრძე.

პროცედურის კორექტული გამოყენებისათვის განხილული პარამეტრების ტიპები:

lpRootPathName : PChar;

lpVolumeNameBuffer : PChar;

nVolumeNameSize : dWord;

lpVolumeSerialNumber : dWord;

lpMaximumComponentLength : dWord;

lpFileSystemFlags : dWord;

lpFileSystemNameBuffer : PChar;

nFileSystemNameSize : dWord;

პარამეტრების საწყისი მნიშვნელობებია:

lpVolumeNameBuffer := '';

lpVolumeSerialNumber := 0;

lpMaximumComponentLength := 0;

lpFileSystemFlags := 0;

```
lpFileSystemNameBuffer          := '';
GetMem(lpVolumeNameBuffer, Max_Path+1);     //ცვლადისთვის
                                მეხსიერების გამოყოფა
GetMem(lpFileSystemNameBuffer, Max_Path+1); //ცვლადისთვის
                                მეხსიერების გამოყოფა
nVolumeNameSize                 := Max_Path+1;
nFileSystemNameSize             := Max_Path+1;
lpRootPathName                  := PChar(DriveComboBox1.Drive+':\');
                                //TDriveComboBox კომპონენტით
                                განსაზღვრული მოწყობილობის სახელი
```

ფუნქციის გამოძახება:

```
If GetVolumeInformation(
        lpRootPathName,
        lpVolumeNameBuffer,
        nVolumeNameSize,
        @lpVolumeSerialNumber,
        lpMaximumComponentLength,
        lpFileSystemFlags,
        lpFileSystemNameBuffer,
        nFileSystemNameSize)
then begin …<მოქმედებები> …end;
```

რეკომენდაციები პროგრამის დაცვის უზრუნველყოფისათვის:

- არ უნდა იქნეს გამოყენებული კომპონენტის სტანდარტული დამმუშავებლები – უნდა შეიქნას შეტყობინებათა ციკლი შემოწმებისათვის;
- არ უნდა იქნეს შენახული კოდები ერთ ადგილას;
- არ უნდა იქნეს გადამოწმებული კოდი მხოლოდ ერთ ადგილას;
- არ უნდა იქნეს გაანალიზებული მახასიათებელი მისი მიღებისთანავე (წაკითხვისთანავე);
- არ უნდა იქნეს შექმნილი ტესტირებისთვის ფუნქცია ან ბიბლიოთეკა;

- არ უნდა იქნეს განხორციელებული შემოწმებასთან დაკავშირებული მოქმედებები დაუყოვნებლივ თვითონ შემოწმების შემდეგ;
- გამოიყენეთ ყურადღების გადატანას შემოწმების ფუნქციები;
- არ უნდა იქნეს შენახული ცვლადებში შემოწმების შედეგები;
- არ უნდა იქნეს შემოწმებული ერთი ალგორითმით კონტროლის მონაცემები;
- არ უნდა იქნეს შენახული რეესტრში შემოწმების შედეგები;
- გამოიყენეთ პროგრამებისა და მონაცემების დაშიფვრა;
- არ უნდა იქნეს პროგრამაში ტექსტური სტრიქონები დაწერილი მათი რეალური ფორმით;

პროგრამების გატეხვისას გამმართველების გამოყენების წინააღმდეგ განსახორციელებელი მოქმედებები:

- გამმართველის განსაზღვრა პროგრამის გაშვებამდე შეცდომის ემულაციისას ან მუშაობის შეწყვეტისას;
- გამართვის სისტემაში შესრულებისას პროგრამის მუშაობის შეცვლა;
- ლისტინგის გართულება;
- რესურსებში დაშიფრული სტრიქონების გამოყენება.

2. სამუშაოს შესრულება.

პრაქტიკული დავალებისათვის მომზადება.

1. შექმენით ახალი საქაღალდე (სახელად გამოიყენეთ თქვენი მონაცემები: „გვარი_სახელი_ჯგუფისN_პრაქტიკულისN“) შემუშავებული აპლიკაციის შესანახად;
2. პროგრამირების გარემოში შექმენით ახალი პროექტი (გარემო შეარჩიეთ პარალელური კურსების მიხედვით. მაგალითისათვის აქ განხილულია Delphi);
3. ჩაწერეთ შექმნილი პროექტი თქვენს საქაღალდეში (File/Save Project as);
4. სისტემურ ბლოკს მიუერთეთ გარე მეხსიერების მოწყობილობა (Flash-მეხსიერება).

პრაქტიკული დავალების მიზანია დაკავშირებული გარე მეხსიერების მოწყობილობის სერიული ნომრის განსაზღვრა და ამ სერიული ნომრით გარე მეხსიერების მოწყობილობის არსებობის შემოწმების აპლიკაციაში ჩასმა.

3. რეკომენდაციები აპლიკაციის შემუშავებისას.

1. გარე მეხსიერების მოწყობილობის სერიული ნომრის განსაზღვრისათვის უნდა შეიქმნას ცალკე აპლიკაცია, რომელშიც უნდა იქნეს გამოყენებული სტანდარტული კომპონენტი TDriveComboBox;
2. აპლიკაციის რეკომენდებული ინტერფეისი უნდა მოიცავდეს:
   2.1. კომბო ბოქს, რომლის გამოყენებითაც მომხმარებელი შეძლებს დისკის შერჩევას;
   2.2. ტომის ნიშნულის გამოსახვას;
   2.3. სერიული ნომრის გამოსახვას;
   2.4. ფაილური სისტემის სახელის გამოსახვას;
   2.5. კლასტერში სექტორების რაოდენობის გამოსახვას;
   2.6. სექტორში ბაიტების რაოდენობის გამოსახვას.
3. პროგრამაში დაცვის ჩართვისას კომპიუტერთან დაკავშირებული გარე მოწყობილობების დასადგენად შესაძლებელია გამოყენებულ იქნეს ფუნქციები GetLogicalDrives: Integer და GetDriveType (Name: PChar): Integer.

   კომპიუტერზე მიერთებული გარე მოწყობილობების ჩამონათვალის სიის TListBox კომპონენტში გამოყვანის პროცედურის ტექსტი შემდეგია:

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
var i, mask: Integer;
    S: String;
begin
    mask := GetLogicalDrives;
    I := 0;
    while mask <> 0 do
    begin
```

```
            s := chr(ord('a') + i) + ':\';
            if (mask and 1) <>0 then
                  case GetDriveType(PChar(S)) of
                  0: ListBox1.Items.Add(S + 'unknow');
                  1: ListBox1.Items.Add(S + 'not exists');
                  Drive_Removable: ListBox1.Items.Add(S +
                  'removable'); //floppy
                  Drive_Fixed: ListBox1.Items.Add(S + 'fixed');
                        //hard
                  Drive_Remote: ListBox1.Items.Add(S + 'network');
                  Drive_CDROM: ListBox1.Items.Add(S + 'CD_ROM');
                              //cd
                  Drive_RamDisk: ListBox1.Items.Add(S + 'RAM');
            end; //case
            inc(i);
            mask := mask shr 1;
      end;
end;
```

4. პროგრამული უზრუნველყოფის დაცვის პროექტირებისას უნდა იქნეს დაცული ზემოთ მოცემული რეკომენდაციები გარე მოწყობილობის სერიული ნომრის შემოწმებისას.

4. ანგარიშის შინაარსი

ანგარიში ხორციელდება დაარქივებული ფაილის სახით, რომელიც შეიცავს პროექტის ყველა ფაილს (დაგ., Project1.cfg, Project1.dof, Project1.dpr, Project1.exe, Project1.res, Unit1.dcu, Unit1.dfm, Unit1.pas). დაარქივებულ ფაილის სახელად გამოიყენეთ თქვენი მონაცემები: „გვარი_სახელი_ჯგუფისN_პრაქტიკულისN“.

პრაქტიკული სამუშაო N 12
მონაცემთა სიმეტრიული დაშიფვრა

სამუშაოს მიზანი: მონაცემთა დაშიფვრის სიმეტრიული კრიპტოგრაფიული ალგორითმების გამოყენების უნარ-ჩვევების მიღება.

1. ძირითადი დებულებები.

არსებობს კრიპტოგრაფიული ალგორითმების ორი ძირითადი ტიპი:

- სიმეტრიული, რისთვისაც გაშიფვრის გასაღები ემთხვევა დაშიფვრის გასაღებს;
- ასიმეტრიული (საჯარო გასაღების ალგორითმები), ორი განსხვავებული გასაღების დაშიფვრისა და გაშიფვრისთვის.

სიმეტრიული ალგორითმები იყოფა ორ კატეგორიად:

- ნაკადის შიფრები, რომელშიც მონაცემები დამუშავებულია ცოტათი (პერსონაჟი პერსონაჟით);
- ბლოკის შიფრები, რომელშიც ოპერაციები ხორციელდება ბიტების ჯგუფებზე.

კრიპტო-მედეგობა – შიფრის მახასიათებელია, რომელიც განსაზღვრავს მის გამძლეობას გაშიფვრაზე გასაღების ცოდნის გარეშე (ძირითადი მახასიათებლები:

- ყველა შესაძლო გასაღების რაოდენობა;
- კრიპტო-ანალიზისთვის საჭირო საშუალო დრო.

ზოგადი მოთხოვნები კრიპტოგრაფიული ალგორითმების მიმართ:

1. დაშიფრული ტექსტი იკითხება მხოლოდ იმ შემთხვევაში, თუ არსებობს გასაღები;
2. ოპერაციების რაოდენობა გასაღების მოსაძებნად დაშიფრული ტექსტის ფრაგმენტისათვის და მისი შესაბამისი ღია ტექსტის - არანაკლებ შესაძლო გასაღებების საერთო რაოდენობისა;
3. ყველანაირი გასაღების გადამოწმებით გაშიფვრის ოპერაციების რაოდენობას უნდა გააჩნდეს მკაცრი ქვედა ზღვარი და უნდა სცილდებოდეს კომპიუტერების შესაძლებლობებს;

4. დაშიფვრის ალგორითმის ცოდნამ არ უნდა იმოქმედოს დაცვის საიმედოობაზე;
5. გასაღების მცირე ცვლილებამ უნდა გამოიწვიოს დაშიფრული შეტყობინების ტიპის მნიშვნელოვანი ცვლილება, თუნდაც ერთი და იგივე გასაღების გამოყენებისას;
6. დაშიფვრის ალგორითმის სტრუქტურული ელემენტები უნდა იყოს უცვლელი;
7. დაშიფვრის დროს შეტყობინებაში შეტანილი დამატებითი ბიტები მთლიანად და საიმედოდ უნდა იყოს დამალული დაშიფრულ ტექსტში;
8. დაშიფრული ტექსტის სიგრძე უნდა იყოს ორიგინალი ტექსტის სიგრძის ტოლი;
9. გასაღებებს შორის არ უნდა არსებობდეს მარტივი და ადვილად დაყენებადი დამოკიდებულებები, რომლებიც მიმდევრობით გამოიყენება დაშიფვრის დროს;
10. ყველა შესაძლებელიდან ნებისმიერი გასაღები უზრუნველყოფს ინფორმაციის საიმედო დაცვას;
11. ალგორითმმა უნდა დაუშვას როგორც პროგრამული, ასევე აპარატურული რეალიზაცია, ამასთან გასაღების სიგრძის შეცვლამ არ უნდა გამოიწვიოს დაშიფვრის ალგორითმის ხარისხობრივი გაუარესება.

კრიპტოგრაფიული დახურვის მეთოდებს შორის შემდეგი მეთოდებია:

- შეცვლა (ჩანაცვლება);
- გადაადგილება;
- ანალიტიკური გარდაქმნა;
- გამირება;
- კომბინირებული მეთოდები.

გამირება – ტექსტზე გასაღების საფუძველზე გენერირებული ფსევდო-შემთხვევითი მიმდევრობის დადება. ცნობილია გამირების შემდეგი ტიპები:

- მოკლე სასრული დიაპაზონი;
- გრძელი სასრული დიაპაზონი;

- უსასრულო დიაპაზონი.

გამირების მეთოდით დაშიფვრისას ხორციელდება დაშიფრული ტექსტის სიმბოლოების ჩანაცვლება გამა ციფრული ეკვივალენტებით (ან ორობითი კოდის სახით). დაშიფვრის სიმტკიცე განისაზღვრება პერიოდის ხანგრძლივობით და გამა სტატისტიკური მახასიათებლების ერთგვაროვნობით.

2. სამუშაოს შესრულება.

პრაქტიკული დავალებისათვის მომზადება.

1. შექმენით ახალი საქაღალდე (სახელად გამოიყენეთ თქვენი მონაცემები: „გვარი_სახელი_ჯგუფისN_პრაქტიკულისN“) შემუშავებული აპლიკაციის შესანახად;
2. პროგრამირების გარემოში შექმენით ახალი პროექტი (გარემო შეარჩიეთ პარალელური კურსების მიხედვით. მაგალითისათვის აქ განხილულია Delphi);
3. ჩაწერეთ შექმნილი პროექტი თქვენს საქაღალდეში (File/Save Project as).

პრაქტიკული დავალების მიზანია აპლიკაციის შემუშავება, რომელშიც შეყვანილი ტექსტი მითითებული გასაღების გამოყენებით დაიშიფრება სიმეტრიული ალგორითმით.

3. რეკომენდაციები აპლიკაციის შემუშავებისას.

1. აპლიკაციის შემუშავებისას უნდა იქნეს გამოყენებული სტანდარტული ვიზუალური კომპონენტები (მაგ., TMemo, TEdit, TButton, TRadioGroup, TMainMenu, TOpenDialog, TSaveDialog);
2. აპლიკაციის რეკომენდებული ინტერფეისი უნდა მოიცავდეს:
   2.1. სივრცეს საწყისი ტექსტისათვის;
   2.2. სივრცეს გასაღების ტექსტისათვის;
   2.3. სივრცეს დამუშავებული ტექსტისათვის;
   2.4. მოქმედების შესარჩევ მენიუს (დაშიფვრა, გაშიფვრა);
   2.5. ღილაკს „შესრულება“.

3. დაშიფვრის/გაშიფვრის პროცედურების შესაძლო ვარიანტი თითოეული მოქმედების ცალკეულ პროცედურაში გაფორმებით:

```
interface
const
        StartKey    = 981;          {Start default key}
        MultKey     = 12674;{Mult default key}
        AddKey      = 35891;{Add default key}
function Encrypt(const InString:string; StartKey,MultKey,AddKey:Integer):
string;
function Decrypt(const InString:string; StartKey,MultKey,AddKey:Integer):
string;
implementation
{$R-}
{$Q-}

{**Standard Encryption algorithm - Copied from Borland**}

function Encrypt(const InString:string; StartKey,MultKey,AddKey:Integer):
string;
var
        I : Byte;
begin
Result := '';
        for I := 1 to Length(InString) do
        begin
                Result := Result + CHAR(Byte(InString[I]) xor (StartKey shr
        8));
                StartKey := (Byte(Result[I]) + StartKey) * MultKey +
        AddKey;
        end;
end;

{**Standard Decryption algorithm - Copied from Borland**}
```

```
function Decrypt(const InString:string; StartKey,MultKey,AddKey:Integer):
string; var
        I : Byte;
Begin
Result := '';
        for I := 1 to Length(InString) do
        begin
                Result := Result + CHAR(Byte(InString[I]) xor (StartKey shr
                8));
                StartKey := (Byte(InString[I]) + StartKey) * MultKey +
                AddKey;
        end;
end;
{$R+}
{$Q+}
end.
```

4. რეკომენდებულია დაშიფვრა/გაშიფვრის პროცედურების შემუშავება დაშიფრულ ტექსტზე სასრული მოკლე გამის (გასაღების) დადების გამოყენებით:
   - ტექსტისა და გასაღების სიმბოლოები გადაიყვანება ASCII ცხრილის ციფრულ (Byte ტიპის) ანალოგად;
   - იქმნება ტექსტის მასივის ელემენტების გადარჩევის ციკლი შესაბამისი გასაღების მასივის ელემენტების დადებით (შეკრება ან გამოკლება მონაცემთა ტიპების საზღვრებიდან გასვლის კონტროლით).
5. დაშიფვრის გასაღები ფაილში წერილამდე გარდაქმენით XOR დაშიფვრის ალგორითმის გამოყენებით, რომლის მოდულის მაგალითი, მსგავსი ალგორითმის განხორციელებით შემდეგია:

```
program Crypt;
{$APPTYPE CONSOLE}
uses Windows;
var
        key, text, longkey, result : string;
```

```
        i : integer;
        toto, c : char;
        F : TextFile;
begin
    writeln('Enter the key:');
    readln(key);
    writeln('Enter the text:');
    readln(text);
    for i := 0 to (length(text) div length(key)) do
            longkey := longkey + key;
    for i := 1 to length(text) do
    begin
            // XOR ალგორითმი
            toto := chr((ord(text[i]) xor ord(longkey[i])));
            result := result + toto;
    end;
    writeln('The crypted text is:');
    writeln(result);
    write('Should i save it to result.txt ?');
    read(c);
    if c in ['Y','y'] then
    begin
            AssignFile(F,'result.txt');
            Rewrite(F);
            Writeln(F,result);
            CloseFile(F);
    end;
end.
```

6. შექმენით აპლიკაცია, რომელშიც დაშიფრული ტექსტი ინახება ტექსტური ფორმატის ფაილში გასაღებთან ერთად.

   დაშიფრული ტექსტისა და გასაღების ფაილში ჩაწერამდე, წინასწარ აურიეთ ჩასაწერი მასივები (მიღებულ მასივში გასაღების სიმბოლოების შესანახად უნდა გამოიყოს ფიქსირებული პოზიციები,

გასაღების სწორი შეკრებისათვის უნდა მიეთითოს გასაღების სიგრძე სიმბოლოებში და ფიქსირებული პოზიციების ბიჯი, ანუ დაშიფრულ ტექსტში რამდენ სიმბოლოს შემდეგ იმყოფება გასაღების სიმბოლოები. ეს ნიშნები შეიძლება დაიწეროს როგორც ფაილის პირველ ორი ბაიტში, ასევე ფაილის სხვა პოზიციებში).

ფაილის შინაარსის გაშიფვრა შემდეგი ნაბიჯებისგან შედგება:

- გასაღების მახასიათებლების წაკითხვა;
- გასაღების შეკრება;
- ტექსტის გაშიფრა.

4. ანგარიშის შინაარსი

ანგარიში ხორციელდება დაარქივებული ფაილის სახით, რომელიც შეიცავს პროექტის ყველა ფაილს (მაგ., Project1.cfg, Project1.dof, Project1.dpr, Project1.exe, Project1.res, Unit1.dcu, Unit1.dfm, Unit1.pas). დაარქივებულ ფაილის სახელად გამოიყენეთ თქვენი მონაცემები: „გვარი_სახელი_ჯგუფისN_პრაქტიკულისN“.

რეკომენდებული ლიტერატურა:

1. ო. შონია, თ. შეროზია. ინფორმაციული ტექნოლოგიები და უსაფრთხოება თბილისი სტუ. 2008, ბიბლიოთეკის ინდ. 681.142/10.14

2. ო. შონია, გ. ჯანელიძე, ბ. მეფარიშვილი, ინფორმაციული და ქსელური რესურსების უსაფრთხოების უზრუნველყოფა. ,,ტექნიკური უნივერსიტეტი'', 2009, 204 გვ. ISBN 978-9941-14-588-9, სტუ-ს ბიბლიოთეკა 681.142(02)/137

3. Walker M., Certified Ethnical Hacker – Exam Guide, Second Edition, ©2014, -465 p. ISBN: 978-0-07-183647-0

4. ო. შონია, ნ. თოფურია, გ. მაისურაძე, ინფორმაციული უსაფრთხოების სისტემების აგება კორპორაცია Microsoft-ის ტექნოლოგიების გამოყენებით (სახელმძღვანელო), „ტექნიკური უნივერსიტეტი“, 2009, 120 გვ. ISBN 978-9941-14-366-3, სტუ-ს ბიბლიოთეკა https://gtu.ge/book/nino-usafrtx.pdf

5. Computer Crime, Investigation and the Law ,Chuck Easttom and Det. Jeff Taylor, ISBN-10: 1-4354-5532-0, Course Technology PTR A part of Cengage Learning, 2010

6. Secure Java For Web Application Development, Abhay Bhargav and B.V. Kumar, ISBN-13: 978-1-4398-2356-9CRC Press,2010,

7. Foundations of Security ,What Every Programmer Needs to Know, Neil Daswani, Christoph Kern,and Anita Kesavan, ISBN-13 (pbk): 978-1-59059-784-2,Press,2007

8. The J2EE Architect's Handbook: How to be a Successful Technical Architect for J2EE Applications. by Derek Ashmore, ISBN:0972954899, DVT Press © 2004 (284 pages) CD-96 138

9. Web 2.0 Architectures, James Governor, Dion Hinchcliffe, and Duane Nickull,O'REILLY,2000, CD-96 138

10. Security Informatics, Christopher C. Yang, Michael Chiu-Lung Chau ,Jau-Hwang Wang , Hsinchun Chen, ISSN 1934-3221,Springer,2010

გადაეცა წარმოებას 12..11.2020. ხელმოწერილია დასაბეჭდად 20.11.2020. ოფსეტური ქაღალდის ზომა 60X84 1/16. პირობითი ნაბეჭდი თაბახი 1. ტირაჟი 50 ეგზ.

Verba volant, scripta manent

სტუ-ს „IT კონსალტინგის ცენტრი“,
თბილისი, მ. კოსტავას 77