

# ინფორმაციული უსაფრთხოება

ლექცია 1

თამარ ქურდაძე

[tamar.kurdadze@btu.edu.ge](mailto:tamar.kurdadze@btu.edu.ge)

# რა არის ინფორმაცია?

- ინფორმაცია არის ყველა ის მონაცემი, ან მონაცემთა ერთობლიობა, რომელსაც გააჩნია მიზანი და დანიშნულება.
- ინფორმაცია შესაძლოა არსებობდეს:
  1. ქალაქდზე ნაბეჭდი ან ნაწერი, ე.წ „მატერიალური“.
  2. შენახული ელექტრონულად, ციფრულ ან ანალოგურ შემნახველზე, ე.წ „ელექტრონული“.

# რა არის ინფორმაციული უსაფრთხოება?

- ინფორმაციული უსაფრთხოება არის საქმიანობის ერთობლიობა, რომელიც გულისხმობს ორგანიზაციებში არსებული ინფორმაციისა და ინფორმაციული სისტემის წვდომის, ერთიანობის, კონფიდენციალურობისა და მისი განგრძობადი მუშაობის უზრუნველყოფას. იგი ეხება პროცესებსა და ინსტრუმენტებს, რომლებიც შემუშავებულია სენსიტიური ინფორმაციის მოდიფიკაციის, შეფერხების, განადგურებისა და შემოწმებისგან დასაცავად.
- ინფორმაციის უსაფრთხოება მოიცავს პროცესებსა და მეთოდოლოგიებს, რომლებიც შემუშავებულია და დანერგულია ნებისმიერი ინფორმაციის ან მონაცემების უნებართვო წვდომისგან, გამოყენებისგან, გამჟღავნებისგან, განადგურებისგან, მოდიფიკაციის ან შეფერხებისგან დასაცავად.

# რა არის ინფორმაცია და მისი კლასიფიკაცია?

- იმისათვის, რომ ვიცოდეთ თუ რა მნიშვნელობა ენიჭება ინფორმაციულ უსაფრთხოებას, პირველ რიგში, უნდა განვმარტოთ თუ რა არის თავად ინფორმაცია.
- ინფორმაცია, ეს არის შეგროვებული ფაქტები და მონაცემები, რომელიც შეიძლება იყოს შინაარსობრივი ან რიცხობრივი და რომელიც გამოსახული შეიძლება იყოს როგორც ვირტუალურად, ისე მატერიალური სახით. იგი მნიშვნელოვან როლს ასრულებს თითქმის ყველაფერში, რასაც ვაკეთებთ თანამედროვე საზოგადოებაში. ინფორმაცია არის ფაქტები, მონაცემები, ციფრები, სურათები, დოკუმენტები, ხმა ან მოქმედება, რომელიც უნდა “გადაეცეს” მიმღებ პირს, რათა ახსნას, ინფორმირდეს და გადაამოწმოს, რომ მიმღებს შეუძლია გამოიყენოს ასეთი მიწოდებული ინფორმაცია რაიმე კონკრეტული მიზნით.
- პირადი მონაცემები არის პიროვნების იდენტიფიცირების საშუალება, რომელიც შეიძლება იყოს სახელი/გვარი, პირადი ნომერი, ბიომეტრიული მონაცემი, და სხვა უნიკალური მაიდენტიფიცირებელი ნიშანი.
- შესაბამისად, ასეთი მონაცემების ჰაკერისთვის ხელში ჩავარდნით შესაძლებელია ახალი დანაშაულების განხორციელება და პირადი (ყალბი დოკუმენტის დამზადება, საბანკო სესხის აღება) ან/და სამსახურებრივი (მოიპაროს მონაცემები თქვენი სახელით) ზიანის გამოწვევა. ჰაკერს შეუძლია, დაიმალოს თქვენი იდენტობის უკან, რაც ხელს უშლის მის გამომჟღავნებას, გამომიების პროცესში. თუ ვერ დამტკიცდა, რომ დანაშაულებრივი ქმედება განხორციელდა სხვა პირის (ჰაკერის) მიერ, დამნაშავედ კვლავ თქვენ ჩაითვლებით.
- ყოველივე აქედან გამომდინარე, ინფორმაციული უსაფრთხოების მნიშვნელობის მასშტაბები ვრცელდება როგორც ფიზიკურ პირებზე, ისე საჯარო სამართლისა და კერძო სამართლის იურიდიულ პირებზე.

# კლასიფიკაცია

1. კრიტიკული ინფორმაციული სისტემა - ეს არის ინფორმაციული სისტემა, რომლის უწყვეტობაც განსაკუთრებულად მნიშვნელოვანია ქვეყნის თავდაცვის, ეკონომიკური უსაფრთხოების, სახელმწიფოს ან/და საზოგადოების ნორმალური ფუნქციონირებისთვის.
2. კონფიდენციალური ინფორმაცია - ეს არის ინფორმაცია, რომლის ხელყოფაც გამოიწვევს ორგანიზაციის ფუნქციონირების შეფერხებას, ზიანს, საფრთხეს, სახელმწიფო ინტერესების ან/და პირის რეპუტაციის შელახვას. იგი უზრუნველყოფს საიდუმლო ინფორმაციის დაცვას უკანონო გამჟღავნებისაგან. ასეთი ინფორმაცია შეიძლება იყოს დოკუმენტირებულად აღწერილი ან ელექტრონული ფორმით შენახული.
3. შინასამსახურებრივი გამოყენების ინფორმაცია - ეს არის ინფორმაცია, რომელიც გამოიყენება თანამშრომლებსა და სახელშეკრულებო ურთიერთობის მქონე პირებს შორის, სამსახურებრივი მოვალეობის შესრულების მიზნით, რომლის ხელყოფაც გამოიწვევს ორგანიზაციის ფუნქციონირების შეფერხებას, ზიანსა და უსაფრთხოებას.
4. ინფორმაციული აქტივი - ეს არის ორგანიზაციაში არსებული ინფორმაციის ერთობლიობა, რომლითაც შესაძლებელია ინფორმაციის გაგება, გაზიარება, დაცვა და ეფექტურად გამოყენება. ინფორმაციულ აქტივებს აქვთ მნიშვნელოვანი ღირებულება, შეიცავს რისკებს, შინაარსს და სასიცოცხლო ციკლს (შესაძლებელია მისი განადგურება). მაგალითად, ეს შეიძლება იყოს დისკი, "ფლეშკა", კომპიუტერი, სერვერი, ნივთები, თანამშრომლები და ა.შ.
5. ინფორმაციული სისტემა - ეს არის ნებისმიერი მოქმედებების განხორციელება ინფორმაციული ტექნოლოგიების გამოყენებით, რომლის შედეგადაც ხდება ინფორმაციის მართვა ან/და გადაწყვეტილების მიღება.

# ინფორმაციული უსაფრთხოების პოლიტიკა, პრინციპები და რისკები

ინფორმაციული უსაფრთხოების პოლიტიკა არის წესებისა და სახელმძღვანელო პრინციპების ერთობლიობა, რომელიც ადგენს, თუ როგორ უნდა იქნას გამოყენებული, მართული და დაცული საინფორმაციო ტექნოლოგიების (IT) აქტივები და რესურსები. • იგი ვრცელდება ყველა ორგანიზაციაზე, სადაც ციფრულად ინახება ინფორმაცია მის უფლებამოსილებაში. პოლიტიკა ეფუძნება საერთაშორისო სტანდარტებს, შიდა კანონებსა და ნორმატიულ აქტებს, რომლებიც თავის მხრივ, თავსებადი არიან ერთმანეთთან. ორგანიზაცია ასევე პოლიტიკას განსაზღვრავს ყველა არსებული ფაქტორის მიხედვით, საქმიანობის მიზნიდან გამომდინარე (მაგ. რა უფრო მნიშვნელოვანია კომპანიისთვის, გაითვალისწინება ფართობი, მასშტაბი, რისკები, ღირებულებები და ა.შ.).

- გარდა ამისა, არსებობს ინფორმაციული უსაფრთხოების პრინციპები, რომლებსაც ეყრდნობა აღნიშნული პოლიტიკა: კონფიდენციალურობა - ინფორმაციის საიდუმლოების ინდიკატორის, რომელიც უზრუნველყოფს ინფორმაციის დაცვას უკანონო გამჟღავნებისგან. ყველაზე ხშირად, უსაფრთხოების დარღვევა ხდება ინფორმაციის ავტორიზებული წვდომის მქონე პირის მიერ დაშვებული შეცდომის შედეგად. მთლიანობა - ინფორმაციის, ინფორმაციული აქტივის სისწორისა და სი სრულის მახასიათებელი; ხელმისაწვდომობა - უფლებამოსილი პირის/ორგანიზაციის მოთხოვნის შესაბამისად აქტივზე წვდომის და გამოყენების მახასიათებელი. აღნიშნული პრინციპები საერთაშორისო ასპარეზზე მოიხსენიება როგორც CIA Triad



# რისკები და ფიზიკური საფრთხეები

- რამდენადაც ფართო სფეროა ინფორმაციული უსაფრთხოება, იმდენად ბევრი გზა არსებობს ორგანიზაციაში სწორედ ამ უსაფრთხოების დარღვევისთვის, იქნება ეს ადამიანური რესურსებით გამოწვეული რისკები, ტექნიკური საკითხების არცოდნა თუ სხვა დამოუკიდებელი მიზეზები.
- პროცედურული თვალსაზრისით, პირველ რიგში, მნიშვნელოვანია რისკის გამოვლენა, რომელიც გულისხმობს რისკის აღმოჩენასა და გაცნობიერებას. შემდეგი ეტაპი არის რისკის ანალიზი, რომელიც მოიცავს მისი არსის გაცნობიერებას და რისკის დონის დადგენას. ეს უკანასკნელი გულისხმობს გამოვლენილი სისუსტის შედარებას სხვა არსებულ რისკებთან და კრიტერიუმებთან, რომლის მეშვეობითაც განისაზღვრება, რამდენად მნიშვნელოვანია გამოვლენილი რისკი, რა მოპყრობას საჭიროებს იგი, მისი კონტროლის მექანიზმები და რა ოდენობის რესურსის გამოყოფაა საჭირო აღნიშნულის პრევენციისთვის.
- ყოველივე ეს განსაზღვრული უნდა იყოს ინფორმაციული უსაფრთხოების ამოცანებში და მათი შესრულების გეგმებში, რომელიც უნდა შეესაბამებოდეს ინფორმაციული უსაფრთხოების პოლიტიკასა და ინფორმაციული უსაფრთხოების კანონით განსაზღვრულ შესაბამის მოთხოვნებს. ასეთი რისკების გამოვლენა და ანალიზი ორგანიზაციას ხელს უწყობს, სწორად ჩამოაყალიბოს ინფორმაციული უსაფრთხოების პოლიტიკა, პრიორიტეტები და რესურსები.

რამდენადაც მნიშვნელოვანია ქსელური თავდაცვა ორგანიზაციებში, ასევე მნიშვნელოვანია ინფორმაციული აქტივის ფიზიკური უსაფრთხოების უზრუნველყოფაც.

მაგალითად, ერთერთი საფრთხეა არაავტორიზებული წვდომა ტერიტორიაზე, რომელიც გულისხმობს გარეშე პირების ორგანიზაციაში შესვლას, რომელთაც შეუძლიათ პირდაპირი განზრახვით ან გაუფრთხილებლობით გამოიწვიონ მაგალითად ხანძარი, კაბელების დაზიანება, ცრუ განგაში ხანძრის შესახებ, ინფრასტრუქტურის დაზიანება/მოპარვა და ა.შ.

ჯამუშობა ასევე ერთ-ერთი მნიშვნელოვანი საფრთხეა, რომლის განხორციელებასაც არ სჭირდება დიდი ფინანსური რესურსი ან/და მასშტაბური არეალი. მთავარია მხოლოდ სიფრთხილე და ყურადღება, კონფიდენციალურობის დაცვასთან ერთად.



## **Security Operation Center**

უსაფრთხოების ოპერაციების ცენტრის (SOC) ფუნქციაა კიბერ საფრთხეების მონიტორინგი, პრევენცია, გამოვლენა, გამოძიება და რეაგირება 24/7-ზე. SOC

გუნდებს ევალებათ ორგანიზაციის აქტივების მონიტორინგი და დაცვა, მათ შორის ინტელექტუალური საკუთრება, პერსონალის მონაცემები, ბიზნეს სისტემები და ბრენდის მთლიანობა.

## **Security information and event management**

უსაფრთხოების ინფორმაციისა და

ღონისძიებების მენეჯმენტი (SIEM) არის უსაფრთხოების მართვის მიდგომა, რომელიც აერთიანებს უსაფრთხოების ინფორმაციის მართვის (SIM) და უსაფრთხოების ღონისძიებების მართვის (SEM) ფუნქციებს უსაფრთხოების მართვის ერთ სისტემაში.

# ინფორმაციული უსაფრთხოების მენეჯერი

- ინფორმაციული უსაფრთხოების პოლიტიკის მოთხოვნების შესრულების ყოველდღიური მონიტორინგი;
- ინფორმაციული აქტივებისა და მათი წვდომის აღწერა;
- ინფორმაციული უსაფრთხოების პოლიტიკის შინაუწყებრივი დოკუმენტაციის მომზადება;
- ინფორმაციული უსაფრთხოების ინციდენტების შესახებ ინფორმაციის შეგროვება და მათზე რეაგირების მონიტორინგი;
- ინფორმაციული უსაფრთხოების საკითხებზე ანგარიშგება და სხვა სახის ადმინისტრაციული/საორგანიზაციო საქმიანობა;
- ინფორმაციული უსაფრთხოების ზოგადი და დარგობრივი ტრენინგების ორგანიზება და ჩატარება.

# კიბერუსაფრთხოების სპეციალისტი

1. კანონმდებლობით კიბერუსაფრთხოების სპეციალისტს საქართველოში ჰქვია “კომპიუტერული უსაფრთხოების სპეციალისტი”.
2. კომპიუტერული სისტემების ყოველდღიური მონიტორინგი და შეფასება;
3. კომპიუტერული ინციდენტის იდენტიფიცირება, მასზე რეაგირება და კომპიუტერული ინციდენტის შესახებ ინფორმაციის
4. კომპიუტერული ინციდენტებისა და უსაფრთხოების ზომების ანალიზი და ანგარიშგება;
5. დახმარების ჯგუფთან კოორდინაცია.

# კიბერდამნაშავეები

ჰაკერი

**Script Kiddies**

ჰაკერების გუნდები და ჰაქტივისტები  
სახელმწიფოს მიერ დაფინანსებული ჰაკერები

# გავრცელებული საფრთხეები

OWASP Top 10:

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

## The Open Web Application Security Project

- Open Web Application Security Project (OWASP) არის არაკომერციული ფონდი, რომელიც გვაწვდის მითითებებს სანდო და უსაფრთხო პროგრამული აპლიკაციების შემუშავების, შეძენასა და შენარჩუნებაზე. OWASP ცნობილია ვებ აპლიკაციების უსაფრთხოების დაუცველობების პოპულარული ტოპ 10 სიით.

# საერთაშორისო სტანდარტები და ადგილობრივი კანონმდებლობა

- საქართველოს კანონი „ინფორმაციული უსაფრთხოების შესახებ“,
- ISO 27001,
- NIST 800-53.

# ვინდოუს და ლინუქს ოპერაციული სისტემების საფრთხეების მიმოხილვა

ლექცია 2

თამარ ქურდაძე

+995 574 809 721

[Tamar.kurdadze@btu.edu.ge](mailto:Tamar.kurdadze@btu.edu.ge)



Chief Information  
Security Officer (CISO)



Cyber Incident  
Responder



Cyber Legal, Policy and  
Compliance Officer



Cyber Threat  
Intelligence Specialist



Cybersecurity  
Architect



Cybersecurity  
Auditor



Cybersecurity  
Educator



Cybersecurity  
Implementer



Cybersecurity  
Researcher



Cybersecurity Risk  
Manager



Digital Forensics  
Investigator



Penetration  
Tester



## განსახილველი საკითხები:

- ▶ ვინდოუსის პროცესები, ნაკადები და სერვისები;
- ▶ რეესტრი;
- ▶ რესურსებისა და მოვლენების მონიტორინგი;
- ▶ ლინუქსის როლი უსაფრთხოების ოპერაციების ცენტრში;
- ▶ მომხმარებლები და ჯგუფები;
- ▶ ნებართვები და წვდომები;
- ▶ ლინუქსის გამაგრება;
- ▶ მონიტორინგის სერვისული ლოგები;
- ▶ პროცესები.

# Windows

## Processes:

- ▶ Windows პროცესები, ნაკადები და სერვისები Windows ოპერაციული სისტემის განუყოფელი კომპონენტებია, რომლებიც მართავენ რესურსებს, ამუშავებენ შეყვანას(input) და გამოყვანას(output) და უზრუნველყოფენ ფუნქციონირებას სხვა პროგრამებისთვის. მათი მუშაობისა და ურთიერთქმედების გაგება დაგეხმარებათ პრობლემების მოგვარებაში, მუშაობის ოპტიმიზაციაში და თქვენი სისტემის უკეთ მართვაში.
- ▶ პროცესები არის პროგრამები, რომლებიც მუშაობს თქვენს კომპიუტერში. ისინი პასუხისმგებელი არიან სისტემის რესურსების მართვაზე, მათ შორის მეხსიერებაზე, CPU გამოყენებაზე და შეყვანის / გამომავალი ოპერაციებზე. ყველა პროგრამას, რომელიც თქვენს კომპიუტერში მუშაობს, აქვს საკუთარი პროცესი, რაც საშუალებას აძლევს ოპერაციულ სისტემას ცალკე მართოს თითოეული პროგრამის რესურსი. You can view and manage processes using the Task Manager, which allows you to see how much CPU and memory each process is using and terminate any processes that are not responding.

# Windows

## Streams:

- ▶ მეორეს მხრივ, ნაკადები არის პროგრამების კომუნიკაციის საშუალება შეყვანისა და გამომავალი მოწყობილობებისთვის, როგორცაა კლავიშებიანი საკრავები, მაუსები და პრინტერები. ნაკადი არის მონაცემთა ნაკადი პროგრამასა და მოწყობილობას შორის. არსებობს სხვადასხვა ტიპის ნაკადები, მათ შორის შეყვანის ნაკადები და გამომავალი ნაკადები. შეყვანის ნაკადები საშუალებას აძლევს პროგრამებს მიიღონ მონაცემები მოწყობილობებიდან, ხოლო გამომავალი ნაკადები საშუალებას აძლევს პროგრამებს გაუგზავნონ მონაცემები მოწყობილობებს. მაგალითად, როდესაც თქვენს კლავიატურაზე აკრიფებთ, კლავიატურა აგზავნის მონაცემებს კომპიუტერში შეყვანის ნაკადის საშუალებით, ხოლო კომპიუტერი ამუშავებს ამ მონაცემებს პროცესის საშუალებით.

# Windows Services:

- ▶ სერვისები, იმავედროულად, არის პროგრამები, რომლებიც მუშაობს ფონზე და უზრუნველყოფს ფუნქციონირებას სხვა პროგრამებისთვის. ისინი არ უნდა იყოს უშუალოდ ურთიერთქმედებაში მომხმარებელთან, მაგრამ ამის ნაცვლად, ისინი უზრუნველყოფენ ფუნქციონირებას, რომელზეც წვდომა შესაძლებელია სხვა პროგრამებით. სერვისების მაგალითები მოიცავს print spooler service, რომელიც მართავს ბეჭდვის სამუშაოებს და Windows განახლების სერვისს, რომელიც ავტომატურად განახლებს თქვენს სისტემას უსაფრთხოების პატჩებით და სხვა განახლებებით. თქვენ შეგიძლიათ ნახოთ და მართოთ სერვისები Services console-ის გამოყენებით, რაც საშუალებას გაძლევთ დაიწყოთ, შეაჩეროთ ან გადატვირთოთ სერვისები.

▶ პროცესები, ნაკადები და სერვისები ყველა ერთმანეთთან არის დაკავშირებული და ერთად მუშაობს, რათა თქვენი სისტემა შეუფერხებლად მუშაობდეს. მაგალითად, დოკუმენტის დაბეჭდვისას, ბეჭდვის სამუშაოს მართავს the print spooler service, რომელიც სამუშაოს უგზავნის პრინტერს გამომავალი ნაკადის საშუალებით. შემდეგ პრინტერი აგზავნის მონაცემებს კომპიუტერში შეყვანის ნაკადის საშუალებით, რომელიც დამუშავებულია პროცესით. თუ რომელიმე ეს კომპონენტი არ მუშაობს სწორად, თქვენ შეიძლება განიცადოთ პრობლემები ბეჭდვის ან სისტემის სხვა ფუნქციებთან დაკავშირებით.

▶ დასასრულს, იმის გაგება, თუ როგორ მუშაობს Windows პროცესები, ნაკადები და სერვისები და ურთიერთქმედება, დაგეხმარებათ პრობლემების მოგვარებაში, მუშაობის ოპტიმიზაციაში და თქვენი სისტემის უკეთ მართვაში. ამ კომპონენტების როლებისა და ფუნქციების გაცნობიერებით, შეგიძლიათ უფრო ეფექტურად დაადგინოთ და მოაგვაროთ ნებისმიერი პრობლემა, რომელიც წარმოიქმნება და უზრუნველყოთ, რომ თქვენი კომპიუტერი შეუფერხებლად და ეფექტურად მუშაობს.

# Windows Registry:

► Windows რეესტრი არის იერარქიული მონაცემთა ბაზა, რომელიც ინახავს კონფიგურაციის პარამეტრებს და Windows ოპერაციული სისტემის დაინსტალირებული პროგრამების ვარიანტებს. ეს არის Windows ოპერაციული სისტემის კრიტიკული კომპონენტი და შეიცავს ინფორმაციას ტექნიკის, პროგრამული უზრუნველყოფის, მომხმარებლის პროფილების და სისტემის პარამეტრების შესახებ.

► რეესტრი ორგანიზებულია ხის მსგავსი სტრუქტურაში, თითოეული კვანძი წარმოადგენს გასაღებს და თითოეული გასაღები, რომელიც შეიცავს ერთ ან მეტ მნიშვნელობას. რეესტრში გასაღებებისა და მნიშვნელობების წვდომა და შეცვლა შესაძლებელია Windows Registry Editor- ის გამოყენებით ან სკრიპტებისა და პროგრამირების ენების გამოყენებით, რომლებიც ხელს უწყობენ რეესტრის წვდომას.

► Some common types of information stored in the registry include:

1. აპლიკაციის პარამეტრები: დაინსტალირებულმა აპლიკაციებმა შეიძლება შეინახონ კონფიგურაციის პარამეტრები, ლიცენზირების ინფორმაცია და სხვა მონაცემები რეესტრში.
2. მომხმარებლის პროფილები: მომხმარებლის სპეციფიკური პარამეტრები და პრეფერენციები, როგორიცაა დესკტოპის ფონი, ინახება რეესტრში.
3. Hardware settings: რეესტრი ინახავს ინფორმაციას ტექნიკის მოწყობილობების, მათ შორის მათი დრაივერების და კონფიგურაციის პარამეტრების შესახებ.
4. System settings: რეესტრში ინახება სხვადასხვა სისტემის მასშტაბის პარამეტრები, როგორიცაა უსაფრთხოების პოლიტიკა და გაშვების პროგრამები.

► რეესტრის რედაქტირებამ შეიძლება მნიშვნელოვანი გავლენა მოახდინოს Windows ოპერაციული სისტემის მუშაობაზე და დაინსტალირებულ პროგრამებზე, ამიტომ მნიშვნელოვანია ფრთხილად იყოთ რეესტრში ცვლილებების შეტანისას. რეესტრის არასწორად შეცვლამ შეიძლება გამოიწვიოს სისტემის არასტაბილურობა, ავარია და მონაცემთა დაკარგვა. მიზანშეწონილია, რომ მომხმარებლებმა განახორციელონ რეესტრის სარეზერვო ასლი რაიმე ცვლილების შეტანამდე და მხოლოდ რეესტრის შეცვლა, თუ მათ აქვთ მკაფიო გაგება იმის შესახებ, თუ რას აკეთებენ.



# Monitoring resources

► რესურსებისა და მოვლენების მონიტორინგი Windows ოპერაციული სისტემის უსაფრთხოებისა და სტაბილურობის შენარჩუნების მნიშვნელოვანი ასპექტია. აქ მოცემულია რამდენიმე გზა, რომლითაც შესაძლებელია რესურსებისა და მოვლენების მონიტორინგი Windows- ში:

1. Performance Monitor: შესრულების მონიტორი არის ჩაშენებული ინსტრუმენტი, რომელიც შეიძლება გამოყენებულ იქნას სისტემის მუშაობის მეტრიკის მონიტორინგისთვის, როგორცაა CPU გამოყენება, მეხსიერების გამოყენება, დისკის აქტივობა და ქსელის გამოყენება. ის ასევე შეიძლება გამოყენებულ იქნას მორგებული შესრულების მრიცხველებისა და სიგნალების შესაქმნელად.
  2. Event Viewer: Event Viewer არის კიდევ ერთი ჩაშენებული ინსტრუმენტი, რომელიც შეიძლება გამოყენებულ იქნას სისტემის მოვლენებისა და შეცდომების შეტყობინებების მონიტორინგისთვის. ის უზრუნველყოფს სისტემის ჟურნალების ცენტრალიზებულ ხედვას, მათ შორის აპლიკაციის, უსაფრთხოებისა და სისტემის მოვლენებს.
  3. Task Manager: სამუშაო მენეჯერი შეიძლება გამოყენებულ იქნას გაშვებული პროცესებისა და პროგრამების მონიტორინგისთვის, ასევე სისტემის მუშაობის მეტრიკისთვის, როგორცაა CPU გამოყენება, მეხსიერების გამოყენება და დისკის აქტივობა.
  3. Resource Monitor: რესურსების მონიტორი გთავაზობთ დეტალურ ინფორმაციას სისტემის რესურსების გამოყენების შესახებ, მათ შორის CPU, მეხსიერება, დისკი და ქსელის აქტივობა. ის ასევე შეიძლება გამოყენებულ იქნას ინდივიდუალური პროცესებისა და სერვისების მონიტორინგისთვის.
  5. Third-party monitoring tools: Windows- ისთვის ხელმისაწვდომია მესამე მხარის მონიტორინგის მრავალი ინსტრუმენტი, რომელსაც შეუძლია უზრუნველყოს უფრო მოწინავე მონიტორინგისა და გაფრთხილების შესაძლებლობები. მაგალითები მოიცავს Nagios, PRTG ქსელის მონიტორი და SolarWinds სერვერი და აპლიკაციის მონიტორი.
- Windows- ში რესურსებისა და მოვლენების მონიტორინგით, ადმინისტრატორებს შეუძლიათ დაადგინონ და გადაჭრან პოტენციური საკითხები, სანამ ისინი მნიშვნელოვან პრობლემებს გამოიწვევენ. მას ასევე შეუძლია დაეხმაროს შესაძლებლობების დაგეგმვას, პრობლემების მოგვარებას და შესრულების შეფერხებების იდენტიფიცირებას.

# Linux role in SOC:

► Linux გადამწყვეტ როლს ასრულებს უსაფრთხოების ოპერაციებში, რადგან ის არის ფართოდ გამოყენებული ოპერაციული სისტემა სერვერების, ქსელური მოწყობილობებისა და სხვა კრიტიკული ინფრასტრუქტურისთვის. აქ მოცემულია რამდენიმე გზა Linux გამოიყენება უსაფრთხოების ოპერაციებში:

1. Security-focused Linux distributions: არსებობს რამდენიმე Linux განაწილება, რომლებიც სპეციალურად შექმნილია უსაფრთხოების მიზნით. მაგალითები მოიცავს Kali Linux, Parrot Security OS და BlackArch Linux. ეს განაწილება წინასწარ არის დატვირთული უსაფრთხოების ინსტრუმენტებისა და კომუნალური პროგრამების ფართო სპექტრით, რომლებიც შეიძლება გამოყენებულ იქნას დაუცველობის შეფასების, შეღწევადობის ტესტირებისა და სასამართლო ანალიზისთვის.
2. Secure configuration: Linux-ის კონფიგურაცია შესაძლებელია, რომ იყოს ძალიან უსაფრთხო ზედმეტი სერვისების გამორთვით, უსაფრთხო პროტოკოლების გამოყენებით და წვდომის კონტროლის განხორციელებით. ეს ხელს შეუწყობს არავტორიზებული წვდომის თავიდან აცილებას და ექსპლუატაციის რისკის შემცირებას.
3. Open-source security tools: Linux-ისთვის ხელმისაწვდომია მრავალი ღია კოდის უსაფრთხოების ინსტრუმენტი, რომელიც შეიძლება გამოყენებულ იქნას უსაფრთხოების სხვადასხვა ოპერაციებისთვის. მაგალითები მოიცავს Wireshark ქსელის ანალიზისთვის, Snort for intrusion detection და nmap პორტის სკანირებისთვის.
4. Logging and monitoring: Linux უზრუნველყოფს ძლიერი logging და მონიტორინგის შესაძლებლობებს, რომელთა გამოყენება შესაძლებელია უსაფრთხოების ინციდენტების გამოსავლენად და გამოსადიებლად. Logs-ის ცენტრალიზება და ანალიზი შესაძლებელია ისეთი ხელსაწყოების გამოყენებით, როგორიცაა Splunk ან ELK დასტის გამოყენებით.
5. Containerization: Linux-ზე დაფუძნებული კონტეინერიზაციის ტექნოლოგიები, როგორიცაა Docker და Kubernetes, ფართოდ გამოიყენება თანამედროვე აპლიკაციების შემუშავებასა და განლაგებაში. ისინი უზრუნველყოფენ იზოლაციისა და უსაფრთხოების მახასიათებლებს, რაც ხელს შეუწყობს პროგრამების დაცვას და მომსახურება თავდასხმებიდან.

► საერთო ჯამში, Linux-ის მოქნილობა, უსაფრთხოების მახასიათებლები და ღია კოდის ბუნება მას პოპულარულ არჩევანს ხდის უსაფრთხოების ოპერაციებისთვის.



# USERS and GROUPS in Linux

► In Linux, UID (User Identifier) and GID (Group Identifier) are numeric identifiers associated with user accounts and groups, respectively.

► UID გამოიყენება სისტემაში ინდივიდუალური მომხმარებლების იდენტიფიცირებისთვის. Linux სისტემის ყველა მომხმარებლის ანგარიშს ენიჭება უნიკალური UID, რომელიც გამოიყენება ფაილის საკუთრებისა და ნებართვების თვალყურის დევნებისთვის და სისტემის რესურსებზე წვდომის გასაკონტროლებლად.

► GIDs გამოიყენება სისტემაში მომხმარებელთა ჯგუფების იდენტიფიცირებისთვის. Linux სისტემის ყველა ჯგუფს ენიჭება უნიკალური GID, რომელიც გამოიყენება ფაილისა და დირექტორიის ნებართვების სამართავად და სისტემის რესურსებზე წვდომის გასაკონტროლებლად.

► როდესაც ფაილი ან დირექტორია იქმნება, მას ენიჭება მფლობელი და ჯგუფი. ფაილის ან დირექტორიის მფლობელი იდენტიფიცირებულია მათი UID- ის მიერ, ხოლო ჯგუფი იდენტიფიცირებულია მათი GID. შემდეგ ფაილისა და დირექტორიის ნებართვები დადგენილია მფლობელის, ჯგუფისა და სხვა მომხმარებლების ან ჯგუფების საფუძველზე, რომლებსაც შეიძლება მიეცეთ წვდომა.

► You can view a user's UID and associated groups with the `id` command in the terminal. For example, running the command `id username` will display the user's UID, primary group GID, and any additional groups that the user belongs to.

U – user

G – group

O – other A

– all

```
$ ls -l
total 536
drwxrwxr-x 2 carol carol 4096 Dec 10 15:57 Another_Directory
-rw----- 1 carol carol 539663 Dec 10 10:43 picture.jpg
-rw-rw-r-- 1 carol carol 1881 Dec 10 15:57 text.txt
```

- ჩამონათვალის პირველი სვეტი გვიჩვენებს ფაილის ტიპს და ნებართვებს. For example, on `drwxrwxr-x`:
  - პირველი სიმბოლო, D, მიუთითებს ფაილის ტიპზე.
  - The next three characters, `rw`, indicate the permissions for the owner of the file, also referred to as user or u.
  - The next three characters, `rw`, indicate the permissions of the group owning the file, also referred to as g.
  - The last three characters, `r-x`, indicate the permissions for anyone else, also known as others or o.

There are three other files in that directory, but they are hidden. On Linux, files whose name starts with a period (.) are automatically hidden. To see them we need to add the `-a` parameter to `ls`.

## Permissions on Files

- ▶ **r** ნიშნავს წაკითხვის და აქვს octal ღირებულება 4 (არ ინერვიულოთ, ჩვენ განვიხილავთ octals მალე). ეს ნიშნავს ფაილის გახსნისა და მისი შინაარსის წაკითხვის ნებართვას.
- ▶ **w** ნიშნავს წერას და აქვს octal მნიშვნელობა 2. ეს ნიშნავს ფაილის რედაქტირების ან წაშლის ნებართვას.
- ▶ **x** ნიშნავს შესრულებას და აქვს octal მნიშვნელობა 1. ეს ნიშნავს, რომ ფაილი შეიძლება იყოს გაშვებული ან სკრიპტი.
- ▶ ასე რომ, მაგალითად, ფაილი ნებართვებით rw- შეიძლება წაიკითხოს და დაიწეროს, მაგრამ არ შეიძლება შესრულდეს.

## Permissions on Directories

- ▶ **r** ნიშნავს წაკითხვის და აქვს octal ღირებულება 4. ეს ნიშნავს დირექტორიის შინაარსის წაკითხვის ნებართვას, როგორცაა ფაილის სახელები. მაგრამ ეს არ გულისხმობს ფაილების თავად წაკითხვის ნებართვას.
- ▶ **w** ნიშნავს წერას და აქვს octal მნიშვნელობა 2. ეს ნიშნავს ფაილების შექმნის ან წაშლის ნებართვას დირექტორიაში. გაითვალისწინეთ, რომ თქვენ არ შეგიძლიათ შეიტანოთ ეს ცვლილებები მხოლოდ ჩაწერის ნებართვებით, მაგრამ ასევე გჭირდებათ შესრულების ნებართვა (x) დირექტორიაში შესაცვლელად.
- ▶ **x** ნიშნავს შესრულებას და აქვს octal მნიშვნელობა 1. This means permission to enter a directory, but not to list its files (for that r is needed).

- ფაილის ნებართვების შეცვლა ბრძანება `chmod` გამოიყენება ფაილის ნებართვების შესაცვლელად და იღებს მინიმუმ ორ პარამეტრს: პირველი აღწერს რომელი ნებართვების შეცვლა, ხოლო მეორე მიუთითებს ფაილზე ან დირექტორიაზე, სადაც მოხდება ცვლილება. გაითვალისწინეთ, რომ მხოლოდ ფაილის მფლობელის ან სისტემის ადმინისტრატორს (root) შეუძლია შეცვალოს ნებართვები ფაილზე.

```
@debian:~$ chmod u+x filename
@debian:~$ chmod u+r,g-x filename
@debian:~$ chmod u+rw,go=rx filename
@debian:~$ chmod +r filename
```

0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

`rwxr-xr-- 764`

Sticky bit - ფაილის ან საქალაქის მხოლოდ მფლობელის ან root მომხმარებლის დაშვება შესაბამისი დირექტორიის ან ფაილის შეცვლის, გადარქმევის ან წაშლის შესახებ.

## Modifying File Ownership

The command `chown` is used to modify the ownership of a file or directory. The syntax is quite simple:

```
chown USERNAME:GROUPNAME FILENAME
```

For example, let us check a file called `text.txt`:

```
$ ls -l text.txt
-rw-rw---- 1 carol carol 1881 Dec 10 15:57 text.txt
```

The user who owns the file is `carol`, and the group is also `carol`. Now, we will change the group owning the file to some other group, like `students`:


```
$ chown carol:students text.txt
$ ls -l text.txt
-rw-rw---- 1 carol students 1881 Dec 10 15:57 text.txt
```

## Monitoring service logs on Linux:

► Monitoring service logs on Linux is an important part of maintaining the health and security of a system. Here are some common methods for monitoring service logs on Linux:

► Using the system journal: Many modern Linux distributions use the systemd journal as the default logging system. To view logs for a specific service, you can use the `journalctl` command with the `-u` option followed by the name of the service. For example, to view the logs for the SSH service, you would run:

```
journalctl -u sshd
```


 Copy code

► You can also use the `-f` option to follow the logs in real-time.

► Using syslog: Syslog is a traditional logging system that is still commonly used on many Linux systems. Service logs are typically stored in `/var/log/syslog`, and you can view them with a text editor or the `tail` command. For example, to view the last 10 lines of the syslog for the Apache service, you would run:

```
bash
```

```
tail -n 10 /var/log/syslog | grep apache
```

 Copy code

► Using a log management tool: There are many third-party log management tools available for Linux that can provide more advanced logging and analysis capabilities. Examples include Graylog, ELK Stack, and Fluentd.

► By monitoring service logs on Linux, you can identify potential issues and troubleshoot problems before they cause significant problems. It can also help with capacity planning, identifying performance bottlenecks, and detecting security breaches.

## Linux Hardening:

Linux Hardening ეხება Linux სისტემის უზრუნველყოფის პროცესს მისი თავდასხმის ზედაპირის შემცირებით, პოტენციური დაუცველობის შერბილებით და უსაფრთხოებისთვის საუკეთესო პრაქტიკის განხორციელებით.

Linux Hardening-ში ჩართული ზოგიერთი გავრცელებული პრაქტიკა მოიცავს:

- ▶ სისტემის განახლება უსაფრთხოების უახლესი პატჩებითა და განახლებებით.
- ▶ ზედმეტი სერვისებისა და პროგრამების გამორთვა ან წაშლა, რომლებიც არ არის საჭირო სისტემის მუშაობისთვის.
- ▶ სისტემის კონფიგურაცია უსაფრთხო პროტოკოლებისა და დაშიფვრის გამოსაყენებლად, როგორცაა SSH დისტანციური წვდომისთვის და TLS ვებ ტრაფიკისთვის.
- ▶ ძლიერი პაროლებისა და ავთენტიფიკაციის მექანიზმების დანერგვა, როგორცაა ორფაქტორიანი ავთენტიფიკაცია.
- ▶ წვდომის კონტროლისა და ნებართვების დაყენება მგრძნობიარე ფაილებსა და დირექტორიებზე მომხმარებლის წვდომის შეზღუდვის მიზნით.
- ▶ Enabling firewall and intrusion detection systems to monitor and protect the system from external attacks.
- ▶ ისეთი ინსტრუმენტების გამოყენება, როგორცაა SELinux ან AppArmor, რათა განახორციელონ სავალდებულო წვდომის კონტროლი და შეზღუდონ აპლიკაციებისა და პროცესების შესაძლებლობები.
- ▶ უსაფრთხოების აუდიტის განხორციელება და შესვლა სისტემის საქმიანობის მონიტორინგისა და უსაფრთხოების პოტენციური დარღვევების გამოსავლენად.

ამ ზომების განხორციელებით და უსაფრთხოების საუკეთესო პრაქტიკის დაცვით, Linux სისტემა შეიძლება გამკაცრდეს პოტენციური საფრთხეებისგან და უკეთ იყოს დაცული არავტორიზებული წვდომისგან, მონაცემთა დარღვევისგან და უსაფრთხოების სხვა საკითხებისგან.

# PROCESSES:

- ▶ jobs - Display active jobs and their status.
- ▶ sleep - Delay for a specific amount of time.
- ▶ fg - Bring job to the foreground.
- ▶ bg - Move job to the background.
- ▶ kill - Terminate job.
- ▶ exit - Exit current shell.
- ▶ watch - Run a command repeatedly (2 seconds cycle by default).
- ▶ uptime - Display how long the system has been running, the number of current users and system load average.
- ▶ free - Display memory usage.
- ▶ pkill - Send signal to process by name.
- ▶ killall - Kill process(es) by name.
- ▶ top - Display Linux processes.
- ▶ ps - Report a snapshot of the current processes.
- ▶ & - to run a command or process in the background.
- ▶ && - to execute multiple commands
- ▶ •kill



# p s

```
$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root            1  0.0  0.1 204504  6780 ?        Ss   14:04   0:00 /sbin/init
root            2  0.0  0.0      0      0 ?        S    14:04   0:00 [kthreadd]
root            3  0.0  0.0      0      0 ?        S    14:04   0:00 [ksoftirqd/0]
root            5  0.0  0.0      0      0 ?        S<   14:04   0:00 [kworker/0:0H]
root            7  0.0  0.0      0      0 ?        S    14:04   0:00 [rcu_sched]
root            8  0.0  0.0      0      0 ?        S    14:04   0:00 [rcu_bh]
root            9  0.0  0.0      0      0 ?        S    14:04   0:00 [migration/0]
(...)
```

- ▶ USER - Owner of process.
- ▶ PID - Process identifier.
- ▶ %CPU - Percentage of CPU used.
- ▶ %MEM - Percentage of physical memory used.
- ▶ VSZ - Virtual memory of process in KiB.
- ▶ RSS - Non-swapped physical memory used by process in KiB.
- ▶ TT - Terminal (tty) controlling the process.
- ▶ STAT - Code representing the state of process. Apart from S, R and Z (that we saw when describing the output of top), other possible values include: D (uninterruptible sleep—usually waiting for I/O), T (stopped—normally by a control signal). Some extra modifier include: < (high-priority—not nice to other processes), N (low-priority—nice to other processes), or + (in the foreground process group).
- ▶ STARTED - Time at which the process started.
- ▶ TIME - Accumulated CPU time.
- ▶ COMMAND - Command that started the process

p

- ## reporting information about the running processes:

- Lower values have a higher priority than higher ones.

- S - Status of process. Values include: S (interruptible sleep—waiting for an event to finish), R (runnable—either executing or in the queue to be executed) or Z (zombie—terminated child processes whose data structures have not yet been removed from the process table).

- \$ top**

```
top - 11:10:29 up 2:21, 1 user, load average: 0,11, 0,20, 0,14
Tasks: 73 total, 1 running, 72 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 0,3 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 1020332 total, 909492 free, 38796 used, 72044 buff/cache
KiB Swap: 1046524 total, 1046524 free, 0 used. 873264 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
436	carol	20	0	42696	3624	3060	R	0,7	0,4	0:00.30	top
4	root	20	0	0	0	0	S	0,3	0,0	0:00.12	kworker/0:0
399	root	20	0	95204	6748	5780	S	0,3	0,7	0:00.22	sshd
1	root	20	0	56872	6596	5208	S	0,0	0,6	0:01.29	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.02	ksoftirqd/0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/u2:0
7	root	20	0	0	0	0	S	0,0	0,0	0:00.08	rcu_sched
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
10	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	lru-add-drain
(...)											

№	სახელი	მნიშვნელობა
1	HUP ან SIGHUP	ტრადიციულად, ეს სიგნალი პროცესს ასრულებს. ის აგრეთვე გამოიყენება ბევრი დემონის მიერ პროცესის რეინიციალიზაციისათვის, ანუ ამ სიგნალის მიღების შემდეგ დემონი გადაიტვირთება (გამოირთვება და ხელახლა ჩაირთვება). შესაბამისად, ხელახლა წაიკითხავს კონფიგურაციის ფაილს.
2	INT ან SIGINT	შეწყვეტა. იგივეა, რაც კლავიატურიდან გადაცემული <span>Ctrl^c</span> კლავიშების კომბინაციით გადაცემული სიგნალი.
15	TERM ან SIGTERM	დამთავრება. ესაა kill ბრძანებაში ნაგულისხმევი მნიშვნელობა, თუ სიგნალი არ არის მითითებული.
9	KILL ან SIGKILLT	მოკვლა. ავარიული გამორთვა. გამოიყენება მაშინ, როდესაც პროცესი დასრულების სხვა სიგნალებზე არ რეაგირებს. ამ სიგნალის გადაცემით პროცესის სწორი გაწმენდა არ ხდება, შესაბამისად, მდგომარეობა არ ინახება.
19	STOP ან SIGSTOP	შეჩერება, დაპაუზება. პროგრამას არ შეუძლია ამ სიგნალის იგნორირება. ის მას გვერდს ვერ აუვლის.
18	CONT ან SIGCONT	STOP-ით შეჩერებულის გაგრძელება.
20	TSTP ან SIGTSTP	შეჩერება, დაპაუზება. იგივეა რაც <span>Ctrl^z</span> . პროგრამას აქვს შესაძლებლობა უგულებელყოს ეს სიგნალი.

- ქსელური პროტოკოლები და სერვისები. განსაზღვრული საკითხები:
- ეზერნეტ და ინტერნეტ პროტოკოლი (IP);
- ARP პროტოკოლი; +
- ტრანსპორტის შრის პროტოკოლები; transport layer protocols; (TCP/UDP) + OSI
- ქსელური სერვისები (DHCP+, DNS+, NAT, FTP+, Email, HTTP+, HTTPS)

რა არის FTP?

ფაილის გადაცემის პროტოკოლი (FTP) არის, როგორც სახელიდან ჩანს, პროტოკოლი, რომელიც გამოიყენება ფაილების დისტანციური გადაცემის საშუალებას ქსელში. ამისათვის ის იყენებს კლიენტ-სერვერის მოდელს და - როგორც მოგვიანებით შევხებით - ბრძანებებს და მონაცემებს ძალიან ეფექტური გზით გადასცემს.

როგორ მუშაობს FTP ?

ტიპური FTP სესია მუშაობს ორი არხის გამოყენებით:

ბრძანების (ზოგჯერ უწოდებენ საკონტროლო) არხს

მონაცემთა არხი.

როგორც მათი სახელები გულისხმობს, ბრძანების არხი გამოიყენება ბრძანებების გადასაცემად, ისევე როგორც ამ ბრძანებებზე პასუხებისთვის, ხოლო მონაცემთა არხი გამოიყენება მონაცემთა გადასაცემად.

FTP ფუნქციონირებს კლიენტ-სერვერის პროტოკოლის გამოყენებით. კლიენტი იწყებს კავშირს სერვერთან, სერვერი ამოწმებს შესვლის ყველა მონაცემს და შემდეგ ხსნის სესიას.

სანამ სესია ღიაა, კლიენტმა შეიძლება შეასრულოს FTP ბრძანებები სერვერზე

## აქტიური vs პასიური

FTP სერვერს შეიძლება ჰქონდეს აქტიური ან პასიური კავშირების მხარდაჭერა, ან ორივე.

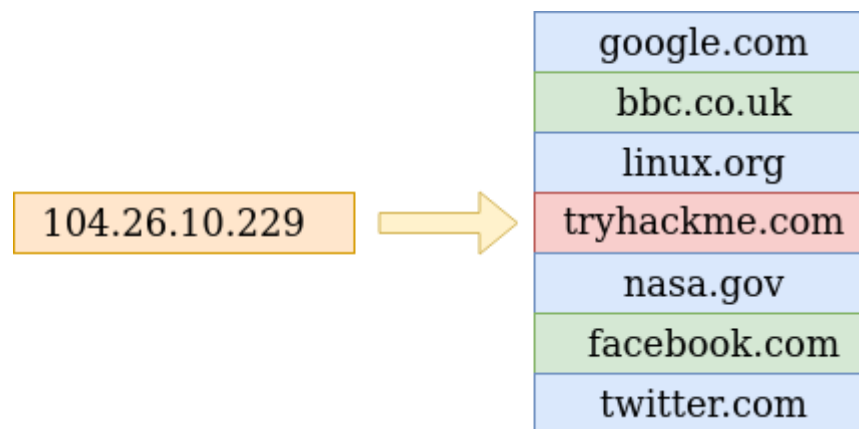
აქტიური FTP კავშირში კლიენტი ხსნის პორტს და უსმენს. სერვერს მოეთხოვება მასთან აქტიური დაკავშირება.

პასიურ FTP კავშირში სერვერი ხსნის პორტს და უსმენს (პასიურად) და კლიენტი უერთდება მას.

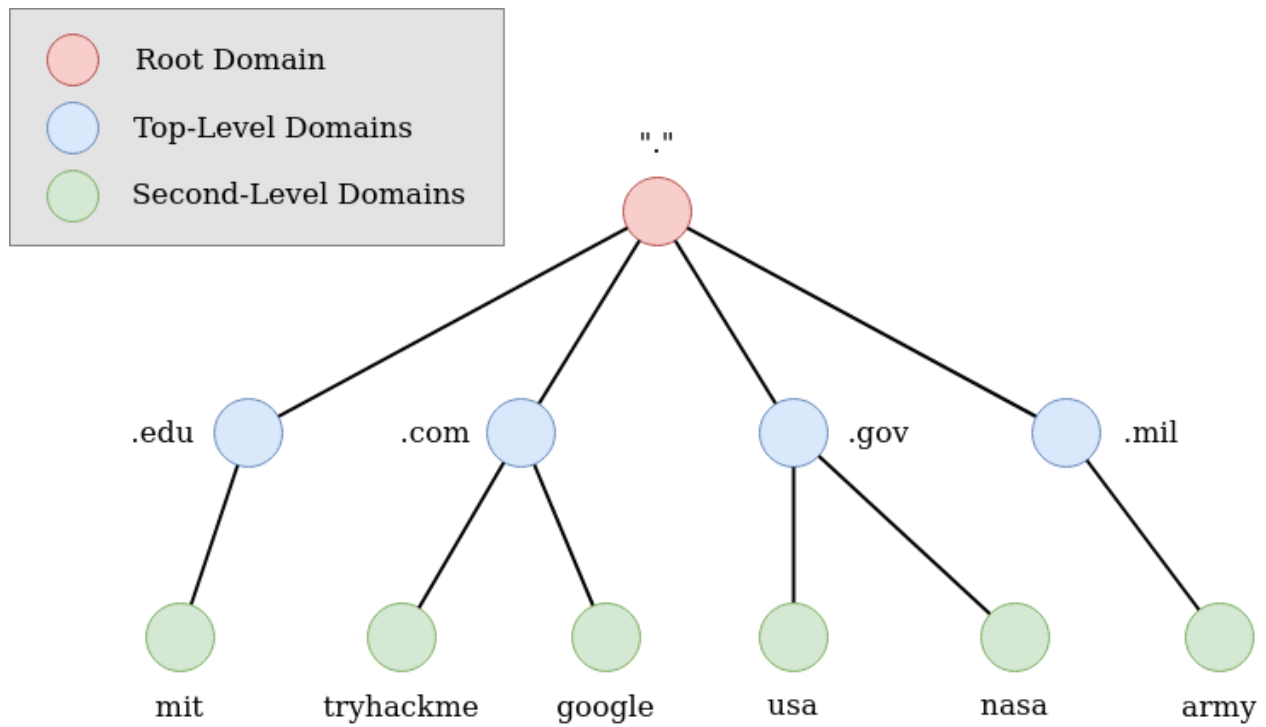
ბრძანების ინფორმაციისა და მონაცემების ცალკეულ არხებად დაყოფა არის გზა სერვერზე ბრძანებების გაგზავნის გარეშე მონაცემთა მიმდინარე გადაცემის დასრულებამდე ლოდინის გარეშე. თუ ორივე არხი ურთიერთდაკავშირებულია, თქვენ შეგეძლოთ შეიყვანოთ ბრძანებები მხოლოდ მონაცემთა გადაცემებს შორის, რაც არ იქნება ეფექტური არც დიდი ფაილების გადაცემისთვის და არც ნელი ინტერნეტ კავშირებისთვის.



რა არის DNS? DNS (დომენის სახელების სისტემა) გვამღევს მარტივ გზას, რომ დაუკავშირდეთ მოწყობილობებს ინტერნეტში რთული ნომრების დამახსოვრების გარეშე. ისევე, როგორც ყველა სახლს აქვს უნიკალური მისამართი ფოსტის გაგზავნისთვის, ინტერნეტში ყველა კომპიუტერს აქვს თავისი უნიკალური მისამართი მასთან კომუნიკაციისთვის, რომელსაც ეწოდება IP მისამართი. IP მისამართი გამოიყურება შემდეგნაირად 104.26.10.229, 4 რიცხვის ნაკრები 0-დან 255-მდე, გამოყოფილი წერტილით. როდესაც გსურთ ვებსაიტის მონახულება, არ არის მოსახერხებელი რიცხვების ამ რთული ნაკრების დამახსოვრება და სწორედ აქ დაგეხმარებათ DNS . ასე რომ, 104.26.10.229-ის დამახსოვრების ნაცვლად, შეგიძლიათ დაიმახსოვროთ tryhackme.com.



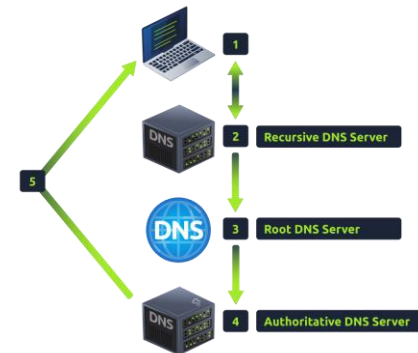
# Domain Hierarchy





რა ხდება DNS მოთხოვნის მიღებისას როდესაც ითხოვთ დომენის სახელს, თქვენი კომპიუტერი ჯერ ამოწმებს მის ლოკალურ ქეშს, რათა ნახოს, ადრე მოძებნეთ თუ არა მისამართი ახლახან; თუ არა, მოთხოვნა განხორციელდება თქვენს რეკურსიულ DNS სერვერზე. რეკურსიული DNS სერვერი ჩვეულებრივ მოწოდებულია თქვენი პროვაიდერის მიერ, მაგრამ თქვენ ასევე შეგიძლიათ აირჩიოთ საკუთარი. ამ სერვერს ასევე აქვს ახლახან მოძიებული დომენური სახელების ადგილობრივი ქეში. თუ შედეგი ადგილობრივად არის ნაპოვნი, ის იგზავნება თქვენს კომპიუტერში და თქვენი მოთხოვნა აქ მთავრდება (ეს ჩვეულებრივია პოპულარული და ძლიერ მოთხოვნადი სერვისებისთვის, როგორიცაა Google, Facebook, Twitter). თუ მოთხოვნა ადგილობრივად ვერ მოიძებნა, მოგზაურობა იწყება სწორი პასუხის მოსაძებნად, დაწყებული ინტერნეტის root DNS სერვერებით.

ძირეული სერვერები მოქმედებენ როგორც ინტერნეტის DNS საყრდენი; მათი ამოცანაა გადამისამართოთ თქვენ სწორ ზედა დონის დომენის სერვერზე, თქვენი მოთხოვნიდან გამომდინარე. თუ, მაგალითად, ითხოვთ [www.tryhackme.com](http://www.tryhackme.com)-ს, root სერვერი ამოიცნობს .com-ის უმაღლესი დონის დომენს და მოგმართავთ სწორ TLD სერვერზე, რომელიც ეხება .com მისამართებს.



TLD სერვერი ინახავს ჩანაწერებს, თუ სად უნდა იპოვოთ ავტორიტეტული სერვერი DNS მოთხოვნაზე პასუხის გასაცემად. ავტორიტეტულ სერვერს ხშირად ასევე უწოდებენ დომენის სახელების სერვერს. მაგალითად, სახელის სერვერი tryhackme.com არის kip.ns.cloudflare.com და uma.ns.cloudflare.com. თქვენ ხშირად იპოვით რამდენიმე სახელების სერვერს დომენის სახელისთვის, რათა იმოქმედოს როგორც სარეზერვო საშუალება იმ შემთხვევაში, თუ ერთი გაქრება.

ავტორიტეტული DNS სერვერი არის სერვერი, რომელიც პასუხისმგებელია DNS ჩანაწერების შესანახად კონკრეტული დომენის სახელისთვის და სადაც განხორციელდება თქვენი დომენის სახელის DNS ჩანაწერების ნებისმიერი განახლება. ჩანაწერის ტიპის მიხედვით, DNS ჩანაწერი იგზავნება უკან რეკურსიულ DNS სერვერზე, სადაც ლოკალური ასლი შეინახება მომავალი მოთხოვნებისთვის და შემდეგ გადაეცემა თავდაპირველ კლიენტს, რომელმაც გააკეთა მოთხოვნა. ყველა DNS ჩანაწერს გააჩნია TTL (Time To Live) მნიშვნელობა. ეს მნიშვნელობა არის რიცხვი, რომელიც წარმოდგენილია წამებში, რომელზეც პასუხი უნდა იყოს შენახული ლოკალურად, სანამ არ მოგიწევთ ხელახლა მოძებნა. ქეშირება დაზოგავს DNS მოთხოვნის გაკეთებას სერვერთან ყოველი კომუნიკაციის დროს.

## ARP პროტოკოლი

**ARP** პროტოკოლი ან **Address Resolution Protocol** მოკლედ არის ტექნოლოგია, რომელიც პასუხისმგებელია მოწყობილობებზე დაუშვას საკუთარი თავის იდენტიფიცირება ქსელში.

უბრალოდ, **ARP** პროტოკოლი საშუალებას აძლევს მოწყობილობას დააკავშიროს თავისი **MAC** მისამართი ქსელის **IP** მისამართთან. ქსელში არსებული თითოეული მოწყობილობა ინახავს სხვა მოწყობილობებთან დაკავშირებული **MAC** მისამართების ჟურნალს.

როდესაც მოწყობილობებს სურთ სხვასთან კომუნიკაცია, ისინი გაგზავნიან მაუწყებლობას მთელ ქსელში, რომელიც ეძებს კონკრეტულ მოწყობილობას. მოწყობილობებს შეუძლიათ გამოიყენონ **ARP** პროტოკოლი კომუნიკაციისთვის მოწყობილობის **MAC** მისამართის (და შესაბამისად ფიზიკური იდენტიფიკატორის) მოსაძებნად.

### როგორ მუშაობს ARP?

თითოეულ მოწყობილობას ქსელში აქვს ჩანაწერი ინფორმაციის შესანახად, რომელსაც ქეში ეწოდება. ARP პროტოკოლის კონტექსტში, ეს ქეში ინახავს ქსელში არსებული სხვა მოწყობილობების იდენტიფიკატორებს.

იმისათვის, რომ ეს ორი იდენტიფიკატორი (IP მისამართი და MAC მისამართი) ერთად ასახოს, ARP პროტოკოლი აგზავნის ორი ტიპის შეტყობინებას:

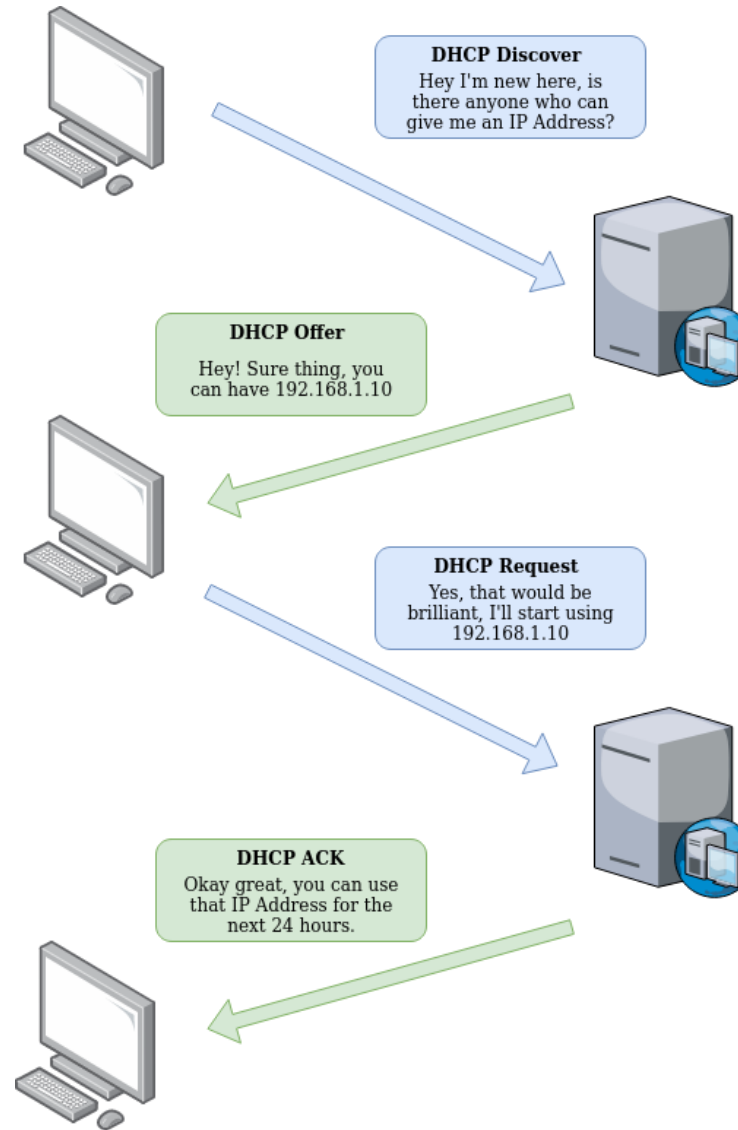
#### ARP მოთხოვნა

#### ARP პასუხი

როდესაც ARP მოთხოვნა გაიგზავნება, შეტყობინება გადაიცემა მოწყობილობის მიერ ქსელში ნაპოვნი ყველა სხვა მოწყობილობაზე, რომელშიც იკითხება ემთხვევა თუ არა მოწყობილობის MAC მისამართი მოთხოვნილ IP მისამართს. თუ მოწყობილობას აქვს მოთხოვნილი IP მისამართი, ARP პასუხი უბრუნდება საწყის მოწყობილობას ამის დასადასტურებლად. საწყისი მოწყობილობა ახლა დაიმახსოვრებს ამას და შეინახავს თავის ქეშში (ARP ჩანაწერი).

## DHCP

**IP მისამართების მინიჭება**  
შესაძლებელია ხელით, მოწყობილობაში  
მათი ფიზიკურად შეყვანით, ან  
ავტომატურად და ყველაზე  
ხშირად **DHCP** (დინამიური ჰოსტის კონ-  
ფიგურაციის პროტოკოლი) სერვერის  
გამოყენებით. როდესაც მოწყობილობა  
უერთდება ქსელს, თუ მას უკვე ხელით  
არ აქვს მინიჭებული IP მისამართი, ის  
აგზავნის მოთხოვნას (**DHCP Discover**),  
რათა ნახოს, არის თუ არა რომელიმე  
**DHCP** სერვერი ქსელში. შემდეგ **DHCP**  
სერვერი პასუხობს IP მისამართით,  
რომლის გამოყენებაც შეიძლება  
მოწყობილობამ (**DHCP Offer**). შემდეგ  
მოწყობილობა აგზავნის პასუხს,  
რომელიც ადასტურებს, რომ მას სურს  
შემოთავაზებული IP მისამართი (**DHCP**  
მოთხოვნა), და ბოლოს, **DHCP** სერვერი  
აგზავნის პასუხს, რომ დაადასტურებს,  
რომ ეს დასრულებულია და  
მოწყობილობას შეუძლია დაიწყოს IP  
მისამართის გამოყენება (**DHCP ACK**).



**რა არის HTTP? (ჰიპერტექსტის გადაცემის პროტოკოლი)**

**HTTP** არის ის, რაც გამოიყენება, როდესაც ხედავთ ვებსაიტს, რომელიც შემუშავებულია ტიმ ბერნერს-ლისა და მისი გუნდის მიერ **1989-1991** წლებში. **HTTP** არის წესების ნაკრები, რომელიც გამოიყენება ვებ სერვერებთან კომუნიკაციისთვის ვებ გვერდის მონაცემების გადასაცემად, იქნება ეს **HTML**, სურათები, ვიდეო და ა.შ.

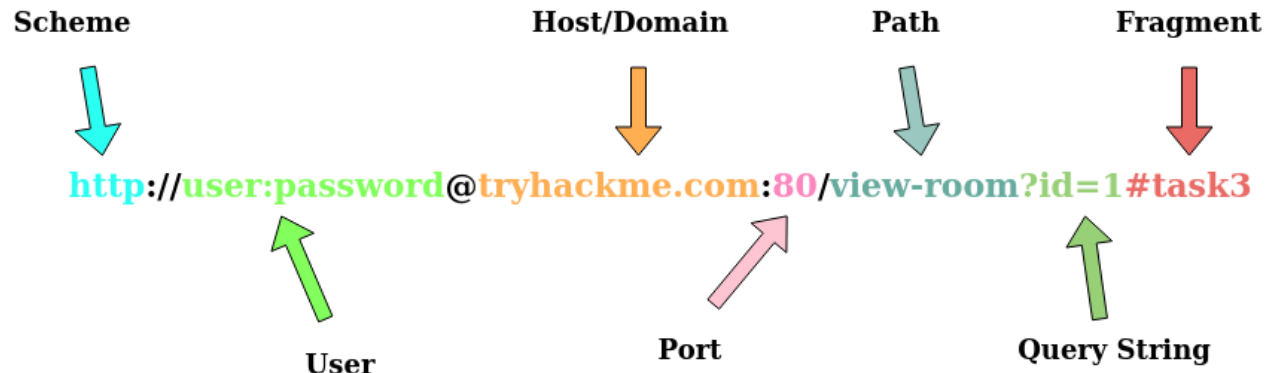
**რა არის HTTPS? (ჰიპერტექსტის გადაცემის პროტოკოლი უსაფრთხო)**

**HTTPS** არის **HTTP**-ის უსაფრთხო ვერსია. **HTTPS** მონაცემები დაშიფრულია, ასე რომ ის არა მხოლოდ აჩერებს ადამიანებს თქვენს მიერ მიღებული და გაგზავნილი მონაცემების დანახვას, არამედ გაძლევთ გარანტიას, რომ ესაუბრებით სწორ ვებ სერვერს და არა რაიმეს იმიტირებული.

**რა არის URL? (რესურსების ერთიანი ლოკატორი)**

თუ იყენებდით ინტერნეტს, ადრე იყენებდით **URL**-ს. **URL** ძირითადად არის ინსტრუქცია, თუ როგორ უნდა შეხვიდეთ რესურსზე ინტერნეტში. ქვემოთ მოყვანილი სურათი გვიჩვენებს, თუ როგორ გამოიყურება **URL** მისი ყველა მახასიათებლით (ის არ იყენებს ყველა ფუნქციას ყველა მოთხოვნაში).





**Scheme:** ეს ინსტრუქციას იძლევა, თუ რა პროტოკოლი გამოიყენოს ისეთ რესურსზე წვდომისთვის, როგორიცაა HTTP, HTTPS, FTP (ფაილის გადაცემის პროტოკოლი).

**User:** ზოგიერთი სერვისი საჭიროებს ავტორიზაციას შესასვლელად, შესასვლელად შეგიძლიათ დააყენოთ მომხმარებლის სახელი და პაროლი URL-ში.

**Host:** სერვერის დომენის სახელი ან IP მისამართი, რომელზეც გასურთ წვდომა.

**Port:** პორტი, რომელსაც აპირებთ დაკავშირებას, ჩვეულებრივ 80 HTTP და 443 HTTPS-ისთვის, მაგრამ ის შეიძლება განთავსდეს ნებისმიერ პორტზე 1-დან 65535-მდე.

**Path:** ფაილის სახელი ან იმ რესურსის მდებარეობა, რომელზეც წვდომას ცდილობთ.

**Query String:** ინფორმაციის დამატებითი ბიტი, რომელიც შეიძლება გაიგზავნოს მოთხოვნილ გზაზე. მაგალითად, `/blog?id=1` ბლოგის გზას ეტყვის, რომ გასურთ ბლოგის სტატიის მიღება 1-ის ID-ით.

**Fragment:** ეს არის მინიშნება მდებარეობის შესახებ მოთხოვნილ რეალურ გვერდზე. ეს ჩვეულებრივ გამოიყენება გრძელი შინაარსის მქონე გვერდებისთვის და შეიძლება ჰქონდეს გვერდის გარკვეული ნაწილი უშუალოდ მასთან დაკავშირებული, ასე რომ, მომხმარებლისთვის ხილვა შესაძლებელია, როგორც კი ისინი შედიან გვერდზე.

**HTTP სტატუსის კოდები:**  
მათი მოთხოვნის შედეგი და ასევე პოტენციურად როგორ გაუმკლავდეს მას. ეს სტატუსის კოდები შეიძლება დაიყოს 5 სხვადასხვა დიაპაზონში:

100-199	Information Response	These are sent to tell the client the first part of their request has been accepted and they should continue sending the rest of their request. These codes are no longer very common.
200-299	Success	This range of status codes is used to tell the client their request was successful.
300-399	Redirection	These are used to redirect the client's request to another resource. This can be either to a different webpage or a different website altogether.
400-499	Client Errors	Used to inform the client that there was an error with their request.
500-599	Server Errors	This is reserved for errors happening on the server-side and usually indicate quite a major problem with the server handling the request.

200 - OK	The request was completed successfully.
401 - Not Authorised	You are not currently allowed to view this resource until you have authorised with the web application, most commonly with a username and password.
404 - Page Not Found	The page/resource you requested does not exist.

**GET** მოთხოვნა - ის გამოიყენება ვებ სერვერიდან ინფორმაციის მისაღებად.  
**POST** მოთხოვნა - გამოიყენება ვებ სერვერზე მონაცემების გასაგზავნად და პოტენციურად ახალი ჩანაწერების შესაქმნელად  
**PUT** მოთხოვნა - გამოიყენება ვებ სერვერზე მონაცემების გასაგზავნად ინფორმაციის განახლებისთვის  
**DELETE** მოთხოვნა - გამოიყენება ვებ სერვერიდან ინფორმაციის/ჩანაწერების წასაშლელად.

საერთო მოთხოვნის სათაურები

ეს არის სათაურები, რომლებიც იგზავნება კლიენტიდან (ჩვეულებრივ თქვენი ბრაუზერიდან) სერვერზე.

**Host:** ზოგიერთი ვებ სერვერი მასპინძლობს მრავალ ვებსაიტს, ასე რომ, ჰოსტის სათაურების მიწოდებით შეგიძლიათ უთხრათ რომელი გჭირდებათ, წინააღმდეგ შემთხვევაში თქვენ უბრალოდ მიიღებთ სერვერის ნაგულისხმევ ვებსაიტს.

**User-Agent:** ეს არის თქვენი ბრაუზერის პროგრამული უზრუნველყოფა და ვერსიის ნომერი, ვებ სერვერს ეუბნება, რომ თქვენი ბრაუზერის პროგრამული უზრუნველყოფა ეხმარება მას ვებსაიტის სწორად ფორმატირება თქვენი ბრაუზერისთვის და ასევე **HTML, JavaScript** და **CSS** ზოგიერთი ელემენტი ხელმისაწვდომია მხოლოდ გარკვეულ ბრაუზერებში.

**Content-Length:** როდესაც მონაცემთა ვებ სერვერზე გაგზავნით, როგორცაა ფორმა, კონტენტის სიგრძე ეუბნება ვებ სერვერს, თუ რამდენ მონაცემს უნდა ელოდოთ ვებ მოთხოვნაში. ამ გზით სერვერს შეუძლია უზრუნველყოს, რომ მას არ აკლია რაიმე მონაცემი.

**Accept-Encoding:** ეუბნება ვებ სერვერს შეკუმშვის რა ტიპებს უჭერს მხარს ბრაუზერს, რათა მონაცემთა დაპატარავება მოხდეს ინტერნეტით გადასაცემად.

ქუქი: სერვერზე გაგზავნილი მონაცემები თქვენი ინფორმაციის დამახსოვრებაში (დამატებითი ინფორმაციისთვის იხილეთ ქუქიების დავალება).



საერთო პასუხის სათაურები

ეს არის სათაურები, რომლებიც კლიენტს უბრუნდება სერვერიდან მოთხოვნის შემდეგ.

**Set-Cookie:** ინფორმაცია შესანახად, რომელიც უბრუნდება ვებ სერვერს ყოველი მოთხოვნით (დამატებითი ინფორმაციისთვის იხილეთ ქუქიების დავალება).

**Cache-Control:** რამდენ ხანს უნდა შეინახოს პასუხის შინაარსი ბრაუზერის ქეშში, სანამ ის კვლავ მოითხოვს მას.

**Content-Type:** ეს ეუბნება კლიენტს, თუ რა ტიპის მონაცემები ბრუნდება, მაგ., **HTML**, **CSS**, **JavaScript**, სურათები, **PDF**, ვიდეო და ა.შ. კონტენტის ტიპის სათაურის გამოყენებით ბრაუზერმა იცის, როგორ დაამუშავოს მონაცემები.

**Content-Encoding:** რა მეთოდი იქნა გამოყენებული მონაცემების შეკუმშვის მიზნით, რათა ის უფრო მცირე იყოს ინტერნეტით გაგზავნისას.

OSI (Open Systems Interconnection) მოდელი არის კონცეპტუალური ჩარჩო, რომელიც გამოიყენება იმის გასაგებად და სტანდარტიზებისთვის, თუ როგორ ურთიერთობენ სხვადასხვა ქსელის პროტოკოლები და ტექნოლოგიები ქსელში. იგი შედგება შვიდი ფენისგან, თითოეულს აქვს კონკრეტული ფუნქცია:

**Physical Layer:** ეს ფენა ეხება მონაცემთა ფიზიკურ გადაცემას ქსელში, მათ შორის კაბელები, კონექტორები და ელექტრო ან ოპტიკური სიგნალები. ის პირველ რიგში ყურადღებას ამახვილებს ქსელური კომუნიკაციის აპარატურულ ასპექტებზე.

**Data Link Layer:** პასუხისმგებელია მონაცემთა ჩარჩოზე, შეცდომების გამოვლენაზე და მედიაზე წვდომის კონტროლზე. ის უზრუნველყოფს მონაცემების სწორად გადაცემას ფიზიკურ ფენაზე.

**Network Layer:** ეს ფენა ეხება მონაცემთა პაკეტების მარშრუტიზაციას წყაროდან დანიშნულების ადგილამდე ქსელის მრავალ კვანძში. ის ადგენს ლოგიკურ ბილიკებს (მარშრუტებს) მონაცემების გასავლელად.

**Transport Layer:** პასუხისმგებელია ბოლოდან ბოლომდე კომუნიკაციაზე, მონაცემთა სეგმენტაციის, ნაკადის კონტროლისა და შეცდომის გამოსწორების ჩათვლით. ის უზრუნველყოფს მონაცემების საიმედოდ მიწოდებას მოწყობილობებს შორის.

**Session Layer:** ეს ფენა მართავს და ადგენს კომუნიკაციის სესიებს ორ მოწყობილობას შორის. ის ამუშავებს სესიის დაყენებას, შენარჩუნებას და შეწყვეტას.

**Presentation Layer:** პასუხისმგებელია მონაცემთა თარგმნაზე, დაშიფვრაზე და შეკუმშვაზე. ის უზრუნველყოფს მონაცემების წარმოდგენის ფორმატში, რომ ორივე გამგზავნმა და მიმღებმა გაიგოს.

**Application Layer:** OSI მოდელის ზედა ფენა, ის ეხება აპლიკაციის სპეციფიკურ პროტოკოლებს და მომხმარებლის ინტერფეისებს. ის აძლევს პროგრამულ აპლიკაციებს ქსელთან და სხვა მოწყობილობებთან ურთიერთობის საშუალებას.

# ინფორმაციული უსაფრთხოება

ლექცია 4

tamar.kurdadze@btu.edu.ge

- სისუსტეები, კიბერსაფრთხეები და თავდასხმები;
- თავდასხმის ინიციატორები, თავდასხმითი ინსტრუმენტები;
- დაზვერვითი თავდასხმები;
- წვდომითი თავდასხმები;
- სოციალური ინჟინერიის თავდასხმები;
- მომსახურებაზე უარის თქმის თავდასხმები dDos;
- IP დაუცველობები და საფრთხეები;
- TCP და UDP დაუცველობები;
- სნიფინგი და სპუფინგი.

კიბერუსაფრთხოების დაუცველობა განისაზღვრება, როგორც სისუსტე ან ხარვეზი სისტემის ან აპლიკაციის დიზაინის, განხორციელების ან ქცევის შესახებ. თავდამსხმელს შეუძლია გამოიყენოს ეს სისუსტეები არაავტორიზებულ ინფორმაციაზე წვდომის ან არაავტორიზებული ქმედებების შესასრულებლად. ტერმინს "დაუცველობა" აქვს კიბერუსაფრთხოების ორგანოების მრავალი განმარტება. თუმცა, მათ შორის მინიმალური ვარიაციაა. მაგალითად, NIST განსაზღვრავს დაუცველობას, როგორც "სისუსტეს საინფორმაციო სისტემაში, სისტემის უსაფრთხოების პროცედურებში, შიდა კონტროლში ან განხორციელებაში, რომელიც შეიძლება გამოყენებულ იქნას ან გამოიწვიოს საფრთხის წყარომ".

დაუცველობა შეიძლება წარმოიშვას მრავალი ფაქტორიდან, მათ შორის აპლიკაციის ცუდი დიზაინიდან ან მომხმარებლისთვის განკუთვნილი მოქმედებების ზედამხედველობისგან.

Vulnerability	Description
Operating System	These types of vulnerabilities are found within Operating Systems (OSs) and often result in privilege escalation.
(Mis)Configuration-based	These types of vulnerability stem from an incorrectly configured application or service. For example, a website exposing customer details.
Weak or Default Credentials	Applications and services that have an element of authentication will come with default credentials when installed. For example, an administrator dashboard may have the username and password of "admin". These are easy to guess by an attacker.
Application Logic	These vulnerabilities are a result of poorly designed applications. For example, poorly implemented authentication mechanisms that may result in an attacker being able to impersonate a user.
Human-Factor	Human-Factor vulnerabilities are vulnerabilities that leverage human behaviour. For example, phishing emails are designed to trick humans into believing they are legitimate.

# თავდასხმის ვექტორები

თავდასხმის აქტორის თვალსაზრისით, თავდასხმის ზედაპირის სხვადასხვა ნაწილი წარმოადგენს პოტენციურ თავდასხმის ვექტორებს. თავდასხმის ვექტორი არის გზა, რომელსაც საფრთხის აქტორი იყენებს უსაფრთხო სისტემაზე წვდომის მოსაპოვებლად. უმეტეს შემთხვევაში, წვდომის მოპოვება ნიშნავს სამიზნეზე მავნე კოდის გაშვებას.

თავდასხმის ვექტორებია:

1. **Direct access**
2. **Removable media**
3. **Email**
4. **Remote and wireless**
5. **Supply chain**
6. **Web and social media**
7. **Cloud**

**წვდომის კონტროლის შეტევები** არის კიბერუსაფრთხოების საფრთხეები, რომლებიც მიზნად ისახავს მექანიზმებსა და პოლიტიკას, რომლებიც გამოიყენება კომპიუტერული სისტემების, ქსელების ან რესურსების ხელმისაწვდომობის მართვისა და შეზღუდვის მიზნით. ეს შეტევები მიზნად ისახავს წვდომის კონტროლის მექანიზმების გვერდის ავლით ან მანიპულირებას მგრძნობიარე მონაცემებზე არასანქცირებული წვდომის მისაღებად ან მავნე ქმედებების შესასრულებლად. აქ მოცემულია რამდენიმე საერთო წვდომის კონტროლის შეტევა:

- 1.Brute Force Attacks: უხეში ძალის შეტევაში, თავდამსხმელი ცდილობს გამოიცნოს მომხმარებლის პაროლი სისტემატურად ცდილობს ყველა შესაძლო კომბინაციას, სანამ სწორი არ მოიძებნება. ეს შეტევა შეიძლება შრომატევადი იყოს, მაგრამ ეფექტურია სუსტი ან ადვილად გამოცნობადი პაროლების წინააღმდეგ.
- 2.Dictionary attacks:უხეში ძალის შეტევების მსგავსად, ლექსიკონის შეტევები იყენებს საერთო სიტყვების ან ფრაზების ჩამონათვალს მომხმარებლის პაროლის გამოსაცნობად. თავდამსხმელები ხშირად იყენებენ სიტყვებისა და ვარიაციების ლექსიკონებს წარმატების შანსების გასაზრდელად.
- 3.Credentials Theft: თავდამსხმელებმა შეიძლება მოიპარონ მომხმარებლის სახელები და პაროლები სხვადასხვა საშუალებით, როგორიცაა ფიშინგი, საკვანძო ნივთები ან მავნე პროგრამები. მას შემდეგ, რაც მათ აქვთ მოქმედი რწმუნებათა სიგელები, მათ შეუძლიათ მიიღონ არავტორიზებული წვდომა სისტემებზე ან ანგარიშებზე.
- 4.Password hacking:პაროლის გატეხვის შეტევები გულისხმობს პროგრამული ინსტრუმენტების გამოყენებას სისტემებში შენახული ჰაშირებული პაროლების გაშიფვრის მცდელობისთვის. თუ სუსტი ჰეშინგის ალგორითმები ან დამარილებული ჰეშები არ გამოიყენება, თავდამსხმელებს შეუძლიათ მიიღონ ორიგინალური პაროლები.
- 5.Escalation of privilege:პრივილეგიების ესკალაციის თავდასხმების დროს, შეზღუდული წვდომის მქონე თავდამსხმელი ცდილობს მოიპოვოს უფრო მაღალი დონის პრივილეგიები ან ადმინისტრაციული წვდომა სისტემაში. ეს შეიძლება მოიცავდეს დაუცველობის ან არასწორი კონფიგურაციების გამოყენებას მათი პრივილეგიების ასამაღლებლად.



Sniffing და spoofing არის ორი საერთო ტექნიკა, რომელიც გამოიყენება კიბერშეტევებში ქსელის ტრაფიკის ჩარევისა და მანიპულირებისთვის. ისინი შეიძლება გამოყენებულ იქნას მავნე აქტორების მიერ, რათა კომპრომეტირება მოახდინონ ქსელში გადაცემული მონაცემების კონფიდენციალურობასა და მთლიანობაზე. აქ მოცემულია თითოეული მათგანის ახსნა:

Sniffing არის პასიური ქსელის მონიტორინგის ტექნიკა, სადაც თავდამსხმელი იღებს და აანალიზებს ქსელის ტრაფიკს, როგორც წესი, არაავტორიზებული გზით. sniffing- ის მიზანია ქსელში გადაცემული მონაცემების მოსმენა საგზაო მოძრაობის ნაკადის შეფერხების გარეშე.

როგორ მუშაობს: Sniffing მოიცავს სპეციალიზებული პროგრამული უზრუნველყოფის ან აპარატურის მოწყობილობების გამოყენებას, სახელწოდებით "sniffers" ან "Package Analyzers". ეს ინსტრუმენტები იპყრობს მონაცემთა პაკეტებს, როდესაც ისინი ქსელში ბრუნავენ. Sniffers ხშირად გამოიყენება ლეგიტიმური მიზნებისათვის, როგორიცაა ქსელის პრობლემების მოგვარება და მონიტორინგი, მაგრამ მათი ბოროტად გამოყენება ასევე შესაძლებელია მავნე საქმიანობისთვის.

მიზანი: თავდამსხმელები იყენებენ sniffing- ს მგრძნობიარე ინფორმაციის გადასაწყვეტად, როგორიცაა შესვლის სერტიფიკატები, ფინანსური მონაცემები ან კონფიდენციალური კომუნიკაციები, რადგან ის მოგზაურობს კომპიუტერებსა და მოწყობილობებს შორის ქსელში.

შერბილება: იმისათვის, რომ დაიცვას sniffing შეტევები, ქსელის დაშიფვრის ტექნიკა, როგორიცაა SSL / TLS ან VPN, შეიძლება გამოყენებულ იქნას ტრანზიტში მონაცემების დასაცავად. გარდა ამისა, ქსელის სეგმენტაციის, შეჭრის აღმოჩენის სისტემების (IDS) და შეჭრის პრევენციის სისტემების (IPS) დანერგვა ხელს შეუწყობს არაავტორიზებული sniffing- ის გამოვლენას და თავიდან აცილებას.



**Spoofing** არის აქტიური ქსელის შეტევა, რომლის დროსაც თავდამსხმელი ასახავს სხვა ერთეულს, როგორცაა ლეგიტიმური მომხმარებელი, მოწყობილობა ან სერვერი, რათა მოატყუოს ან მანიპულირებდეს ქსელის ტრაფიკს ან მოიპოვოს არავტორიზებული წვდომა..

•**Types of spoofing:**

- **IP Address Spoofing:** თავდამსხმელები შეცვლიან პაკეტების წყაროს IP მისამართს, რათა ის გამოჩნდეს, თითქოს ტრაფიკი მოდის სანდო წყაროდან, რაც მათ საშუალებას აძლევს გვერდის ავლით წვდომის კონტროლი ან დაიწყონ სხვა შეტევები.
- **MAC Address Spoofing:** თავდამსხმელები თავიანთი ქსელის ინტერფეისის MAC (მედია წვდომის კონტროლის) მისამართს ცვლიან იმავე ლოკალურ ქსელში სხვა მოწყობილობის იმიტაციისთვის.
- **DNS Spoofing:** DNS (დომენის სახელების სისტემა) გაყალბებისას, თავდამსხმელები მანიპულირებენ DNS პასუხებით, რათა გადამისამართონ მომხმარებლები მავნე ვებსაიტებზე ან ჩაერიონ მათი ტრაფიკი.
- **ARP Spoofing:** ARP (Address Resolution Protocol) გაყალბება გულისხმობს ARP ცხრილების მანიპულირებას თავდამსხმელის MAC მისამართის ლეგიტიმურ IP მისამართთან დასაკავშირებლად, რაც იწვევს ქსელის ტრაფიკის გადამისამართებას თავდამსხმელის აპარატის მეშვეობით.
- **Email Spoofing:** ელ.ფოსტის გაყალბებისას, თავდამსხმელები აყალბებენ გამგზავნის ელფოსტის მისამართს, რათა მოატყუონ მიმღებები, რომ სჯეროდეთ, რომ ელფოსტა სანდო წყაროდან არის.

•**დანიშნულება:** Spoofing თავდასხმები შეიძლება გამოყენებულ იქნას სხვადასხვა მავნე მიზნებისთვის, მათ შორის მოსმენა, ადამიანის შუა შეტევები, სესიის გატაცება და სისტემებზე ან ქსელებზე არასანქცირებული წვდომის მოპოვება.

•**შერბილება:** თავდასხმებისგან დასაცავად, ორგანიზაციებს შეუძლიათ განახორციელონ ისეთი ზომები, როგორცაა ძლიერი ავთენტიფიკაცია, ქსელის მონიტორინგი, შეჭრის გამოვლენა, ანტი-სპუფინგის ფილტრები და კრიპტოგრაფიული პროტოკოლების გამოყენება ქსელის ტრაფიკის ნამდვილობის დასამოწმებლად.

DDoS (Distributed Denial of Service) არის კიბერშეტევის სახეობა, რომლის დროსაც ქსელი ან ონლაინ სერვისი გადატვირთულია ტრაფიკის ნაკადით მრავალი წყაროდან, რაც მიუწვდომელია მისი დანიშნულების მომხმარებლებისთვის. DDoS შეტევის მიზანია ვებგვერდის, სერვერის ან ქსელის ნორმალური ფუნქციონირების დარღვევა მისი რესურსების და გამტარუნარიანობის ამოწურვამდე. აქ მოცემულია DDoS შეტევის ძირითადი მახასიათებლები და კომპონენტები:

1. განაწილებული თავდასხმა: განსხვავებით ტრადიციული DoS (მომსახურების უარყოფა) შეტევებისგან, სადაც ერთი წყარო დატბორავს სამიზნეს, DDoS შეტევები მოიცავს მრავალ კომპრომეტირებულ მოწყობილობას, რომლებიც ხშირად ქმნიან ბოტნეტს. ეს მოწყობილობები შეიძლება შეიცავდეს ინფიცირებულ კომპიუტერებს, სერვერებს, IoT მოწყობილობებს და სხვა კომპრომეტირებულ ვებსაიტებსაც კი.
2. აბსოლუტური ტრეფიკი: თავდამსხმელები აგზავნიან მოთხოვნის ან მონაცემთა პაკეტების დიდ რაოდენობას სამიზნეზე. ტრაფიკის ეს ზრდა იწვევს გადატვირთულობას და ამოწურავს სამიზნის რესურსებს, როგორცაა გამტარუნარიანობა, CPU, მეხსიერება და ქსელური კავშირები.
3. მრავალი თავდასხმის ვექტორი: DDoS შეტევებს შეიძლება ჰქონდეს სხვადასხვა ფორმები, სხვადასხვა შეტევის ვექტორების გამოყენებით ქსელის პროტოკოლებში, აპლიკაციის შრეებსა თუ ინფრასტრუქტურაში დაუცველობის მიზნებისთვის. თავდასხმის საერთო ვექტორები მოიცავს SYN/ACK წყალდიდობას, UDP გაძლიერების შეტევებს, HTTP წყალდიდობას და DNS ასახვის შეტევებს.
4. გაძლიერების ტექნიკა: ზოგიერთი DDoS შეტევა იყენებს ამპლიფიკაციის ტექნიკას სამიზნეზე გაგზავნილი ტრაფიკის მოცულობის გასაზრდელად. მაგალითად, თავდამსხმელებმა შეიძლება გამოიყენონ ცუდად კონფიგურირებული სერვერები ან ქსელის პროტოკოლები, რათა გააძლიერონ თავდასხმის ტრაფიკის ზომა.
5. ბოტნეტები: თავდამსხმელები ხშირად აკონტროლებენ კომპრომეტირებული მოწყობილობების ქსელს, რომელიც ცნობილია როგორც ბოტნეტი, DDoS შეტევების განსახორციელებლად. ეს ბოტნეტები შეიძლება შედგებოდეს ათასობით ან თუნდაც მილიონობით მოწყობილობიდან, რაც ართულებს შეტევის შერბილებას.



# Cyber Threat Intelligence (CTI)

Cyber Threat Intelligence (CTI) შეიძლება განისაზღვროს, როგორც მტკიცებულებებზე დაფუძნებული ცოდნა მოწინააღმდეგეების შესახებ, მათ შორის მათი ინდიკატორების, ტაქტიკის, მოტივაციისა და მათ წინააღმდეგ ქმედითი რჩევების ჩათვლით. მათი გამოყენება შესაძლებელია კრიტიკული აქტივების დასაცავად და კიბერუსაფრთხოების გუნდებისა და მენეჯმენტის ბიზნეს გადაწყვეტილებების ინფორმირებისთვის.

ტიპიური იქნებოდა ტერმინების „მონაცემების“, „ინფორმაციის“ და „დაზვერვის“ გამოყენება ურთიერთშენაცვლებით. თუმცა, მოდით განვასხვავოთ ისინი, რომ უკეთ გავიგოთ, როგორ მოქმედებს CTI.

მონაცემები: მოწინააღმდეგესთან დაკავშირებული დისკრეტული ინდიკატორები, როგორიცაა IP მისამართები, URL-ები ან ჰეშები.

ინფორმაცია: რამდენიმე მონაცემთა პუნქტის კომბინაცია, რომელიც პასუხობს კითხვებს, როგორიცაა „რამდენჯერ შედიოდნენ თანამშრომლები tryhackme.com-ზე თვის განმავლობაში?“

ინტელექტი: მონაცემებისა და ინფორმაციის კორელაცია კონტექსტუალურ ანალიზზე დაფუძნებული მოქმედებების ნიმუშების გამოსატანად.

CTI -ის მთავარი მიზანია გაიგოს ურთიერთობა თქვენს ოპერაციულ გარემოსა და თქვენს მოწინააღმდეგეს შორის და როგორ დაიცვათ თქვენი გარემო ნებისმიერი თავდასხმისგან. თქვენ ეძებთ ამ მიზანს თქვენი კიბერ საფრთხის კონტექსტის შემუშავებით, შემდეგ კითხვებზე პასუხის გაცემის მცდელობით:

ვინ გიტევს?

რა არის მათი მოტივაცია?

როგორია მათი შესაძლებლობები?

რა არტეფაქტებს და კომპრომისის (IOC) ინდიკატორებს უნდა მიაქციოთ ყურადღება?



# Social Engineering

- სოციალური ინჟინერია არის ტერმინი, რომელიც გამოიყენება ნებისმიერი კიბერშეტევის აღსაწერად, სადაც სამიზნე ადამიანია (და არა კომპიუტერი); ამ მიზეზით, მას ხანდახან მოიხსენიებენ, როგორც "ხალხის ჰაკერს". მაგალითად, თუ თავდამსხმელს სურს მიიღოს მსხვერპლის პაროლი, მას შეუძლია სცადოს პაროლის გამოცნობა ან უხეში ძალისხმევით — ან შეიძლება უბრალოდ გკითხოთ.
- მიუხედავად იმისა, რომ ზემოთ მოყვანილი მაგალითი შედარებით მარტივია, სოციალური ინჟინერიის თავდასხმები შეიძლება გახდეს ძალიან რთული და ხშირად მოჰყვება თავდამსხმელის მნიშვნელოვან კონტროლს სამიზნის ცხოვრებაზე - როგორც ონლაინ, ასევე ოფლაინზე. სოციალური ინჟინერიის შეტევები ხშირად მრავალფენიანია და ესკალაცია ხდება თოვლის ბურთის ეფექტის გამო. მაგალითად, თავდამსხმელმა შეიძლება დაიწყოს მცირე რაოდენობის საჯაროდ ხელმისაწვდომი ინფორმაციის მოპოვებით მსხვერპლის სოციალურ მედიაში ყოფნიდან, რომელიც შემდეგ მათ შეუძლიათ გამოიყენონ დამატებითი ინფორმაციის მისაღებად, მაგალითად, თქვენი ტელეფონის ან ფართოზოლოვანი პროვაიდერისგან. მეორე ეტაპიდან მიღებული ინფორმაცია შეიძლება გამოყენებულ იქნას უფრო სასარგებლო ინფორმაციის მოსაპოვებლად, შემდეგ კი ეტაპობრივად გადაიზარდოს მსხვერპლის საბანკო ანგარიშზე.
- სოციალური ინჟინერიის გასაგებად საუკეთესო გზა მისი მოქმედებაში დანახვაა! ეს ვიდეოები Defcon23 -დან (მსოფლიოში ერთ-ერთი ყველაზე დიდი ჰაკერული კონფერენცია) და CNN ასახავს უზარმაზარ ძალას სოციალურ ინჟინერიაში. ორივეს ყურება ღირს!



სოციალური ინჟინერიის სხვა ფორმები

ქარიზმატული ჰაკერები, რომლებიც ურეკავენ თქვენს სატელეფონო კომპანიას და ფლობენ თქვენს ანგარიშს, სოციალური ინჟინერიის ერთ-ერთი ფორმაა; თუმცა, არსებობს მრავალი სხვა ტიპი. სოციალური ინჟინერია არის ვრცელი თემა, რომელიც მოიცავს ნებისმიერ შეტევას, რომელიც ეყრდნობა ადამიანების მოტყუებას თავდამსხმელისთვის წვდომის მინიჭების მიზნით, ვიდრე უშუალოდ ტექნოლოგიაზე თავდასხმას. მიუხედავად იმისა, რომ სამიზნეებთან პირდაპირი ურთიერთქმედება სოციალური ინჟინერიის ყველაზე გავრცელებული სტილია, სხვა მაგალითები მოიცავს USB შენახვის მოწყობილობების საჯაროდ ჩამოგდებას (მაგ. კომპანიის ავტოსადგომებში) იმ იმედით, რომ ვინმე (ხშირად კომპანიის თანამშრომელი) აიღებს ერთს და შეაერთებს მას. მგრძნობიარე კომპიუტერი. ანალოგიურად, თავდამსხმელებმა შეიძლება დატოვონ "დამუხტვის კაბელი", რომელიც ჩართულია სოკეტში საჯარო ადგილას. სინამდვილეში, კაბელი შეიცავს მავნე პროგრამულ უზრუნველყოფას, როგორცაა keyloggers ან ინსტრუმენტები მსხვერპლის მოწყობილობაზე კონტროლისთვის.

დაიცავით სოციალური ინჟინერიის თავდასხმებისგან ბევრი თვალსაზრისით, ძალიან სახიფათოა სოციალური ინჟინერიისგან თავის დაცვა, რადგან ყოველთვის არ იქნებით თქვენ, ვისაც ესაუბრება თავდამსხმელი, არამედ ის, ვინც შეძლებს მათ მიაწოდოს ის, რაც მათ სჭირდებათ თქვენი თანხმობის გარეშე (მაგ., დარეკოთ თქვენს ბანკში, ხოლო ვითომ. იყავი თქვენ, რათა შეხვიდეთ თქვენს საბანკო ანგარიშზე). ამის თქმით, ჯერ კიდევ არსებობს ზომები, რომელთა მიღებაც შეგიძლიათ სოციალური ინჟინერიის შეტევებისგან თავის დასაცავად: ყოველთვის დარწმუნდით, რომ დააყენეთ ავთენტიფიკაციის მრავალი ფორმა და დარწმუნდით, რომ პროვაიდერები პატივს სცემენ მათ. მაგალითად, დააყენეთ რთულად გამოსაცნობი - ან სხვაგვარად არასწორი - პასუხები უსაფრთხოების კითხვებზე (დარწმუნდით, რომ პასუხები შეინახეთ სადმე უსაფრთხო ადგილას!) და დარწმუნდით, რომ ეს კითხვები დაისმება, როდესაც ცდილობთ წვდომას ანგარიშებზე ტელეფონით.

არასოდეს შეაერთოთ გარე მედია (მაგ. USB/CD/ა.შ.) კომპიუტერში, რომელიც გაინტერესებთ ან რომელიც დაკავშირებულია სხვა მოწყობილობებთან. იდეალურ შემთხვევაში, საერთოდ არ ჩართოთ მედია და სანაცვლოდ მიეცით იგი თქვენს ადგილობრივ პოლიციას შესანახად.

ყოველთვის დაჟინებით მოითხოვეთ პირადობის დამადასტურებელი საბუთი, როდესაც უცნობი დაგირეკავთ ან გიგზავნით შეტყობინებას, რომ მუშაობთ კომპანიაში, რომლის სერვისებსაც იყენებთ. სადაც შესაძლებელია, დაადასტურეთ ცნობილი ტელეფონის ნომრით ან ელფოსტის მისამართით, რომ მიღებული ზარი ან შეტყობინება იყო ლეგიტიმური (ანუ გამოიყენეთ სანდო მეთოდი კომპანიასთან დასადასტურებლად დასაკავშირებლად). გახსოვდეთ, რომ არცერთი ლეგიტიმური თანამშრომელი არ მოგთხოვთ თქვენს პაროლს ან სხვა ინფორმაციას, რომელიც იცავს თქვენს ანგარიშს.



კიბერ თაღლითობის მეთოდები:

კიბერუსაფრთხოებაში არსებობს თაღლითობის რამდენიმე მეთოდი, რომლებსაც კიბერკრიმინალები იყენებენ მგრძნობიარე ინფორმაციაზე არაავტორიზებული წვდომის, ფულის მოპარვის ან ზიანის მიყენების მიზნით. აქ არის რამდენიმე ყველაზე გავრცელებული მეთოდი:

ფიშინგი: ეს არის თაღლითობის ტიპი, რომელიც მოიცავს ელ.ფოსტის ან შეტყობინებების გაგზავნას, რომლებიც, როგორც ჩანს, არის ლეგიტიმური წყაროებიდან, როგორიცაა ბანკები ან სოციალური მედიის პლატფორმები. მიზანი არის მიმღების მოტყუება, რათა მიაწოდოს პერსონალური ინფორმაცია, როგორიცაა ავტორიზაციის მონაცემები ან საკრედიტო ბარათის დეტალები.

მავნე პროგრამა: მავნე პროგრამა ეხება მავნე პროგრამულ უზრუნველყოფას, რომელიც შექმნილია კომპიუტერული სისტემების დაზიანების ან მგრძნობიარე ინფორმაციის მოსაპარად. ეს შეიძლება შეიცავდეს ვირუსებს, ჭიებს, ტროას და გამოსასყიდ პროგრამას.

სოციალური ინჟინერია: სოციალური ინჟინერია არის ტექნიკა, რომელიც გამოიყენება ადამიანების მანიპულირებისთვის, რათა გააძლავნონ მგრძნობიარე ინფორმაცია ან განახორციელონ ქმედებები, რომლებსაც ისინი ჩვეულებრივ არ გააკეთებდნენ. ეს შეიძლება მოიცავდეს ისეთ ტექნიკას, როგორიცაა პრეტექსტირება, სატყუარა ან quid pro quo.

ბიზნეს ელფოსტის კომპრომისი (BEC): BEC გულისხმობს ელ. ფოსტის გამოყენებას სანდო პარტნიორის ან გამყიდველის, როგორიცაა ბანკი ან მომწოდებელი, პერსონალის მოსატყუებლად, რათა მოატყუოს თანამშრომლები თანხების გადარიცხვაში ან სენსიტიური ინფორმაციის გაზიარებაში.

პირადობის ქურდობა: პირადობის ქურდობა გულისხმობს ვინმეს პირადი ინფორმაციის მოპარვას, როგორიცაა მათი სახელი, მისამართი, სოციალური დაცვის ნომერი ან საკრედიტო ბარათის დეტალები და ამ ინფორმაციის გამოყენება თაღლითობის ჩასადენად.

ბარათის skimming: ბარათის skimming გულისხმობს მოწყობილობების დაყენებას ბანკომატის აპარატებზე ან გადახდის ტერმინალებზე, რომლებსაც შეუძლიათ საკრედიტო ბარათის ინფორმაციის აღება ბარათის გადაფურცვლისას.

პაროლის შეტევები: პაროლის შეტევები გულისხმობს მომხმარებლის პაროლის გამოცნობის ან გატეხვის მცდელობას მის ანგარიშებზე წვდომის მისაღებად. ეს შეიძლება მოიცავდეს უხეში ძალის შეტევებს, ლექსიკონის შეტევებს ან ფიშინგს, რომლებიც შექმნილია მომხმარებლების მოსატყუებლად, რათა გამოავლინონ მათი პაროლები.

სპამი და ფიშინგი:

სპამი და ფიშინგი ჩვეულებრივი სოციალური ინჟინერიის თავდასხმებია. სოციალურ ინჟინერიაში, ფიშინგის შეტევები შეიძლება იყოს სატელეფონო ზარი, ტექსტური შეტყობინება ან ელფოსტა. პირველი ელფოსტა, რომელიც კლასიფიცირებულია როგორც სპამი, თარიღდება 1978 წლით და ის დღესაც აყვავდება. ფიშინგი არის სერიოზული თავდასხმის ვექტორი, რომლისგანაც თქვენ, როგორც დამცველს, მოგიწევთ დაცვა. ბევრი პროდუქტი გვებმარება სპამით ფიშინგთან ბრძოლაში, მაგრამ რეალურად ამ ელ.წერილების მიღება მაინც შესაძლებელია. როდესაც ისინი ამას აკეთებენ როგორც უსაფრთხოების ანალიტიკოსმა, თქვენ უნდა იცოდეთ როგორ გააანალიზოთ ეს ელფოსტა, რათა დაადგინოთ, რა თუ კეთილთვისებიანი. გარდა ამისა, თქვენ მოგიწევთ ინფორმაციის შეგროვება ელფოსტის შესახებ, რათა განაახლოთ იმ უსაფრთხოების პროდუქტები, რათა თავიდან აიცილოთ მავნე ელ.წერილი მომხმარებლის შემოსულებში.

ადამიანი, რომელმაც გამოიგონა ელექტრონული ფოსტის კონცეფცია და გახდა @ სიმბოლო ცნობილი იყო რეიტომლინსონი. ელ.ფოსტის გამოგონება ARPANET-ისთვის 1970-იანი წლებით თარიღდება. დიახ, ალბათ შენს დაბადებამდე.

- ელექტრონული მისამართი:
- მომხმარებლის საფოსტო ყუთი (ან მომხმარებლის სახელი)
- @
- დომენი

ამის კიდევ უფრო გასამარტივებლად, იფიქრეთ ქუჩაზე, რომელზეც ცხოვრობთ.

თქვენ შეგიძლიათ წარმოიდგინოთ თქვენი ქუჩა, როგორც დომენი. მიმღების სახელი/გვარი, ამ სცენარში სახლის ნომერთან ერთად, წარმოადგენს მომხმარებლის საფოსტო ყუთს. ამ ინფორმაციის საშუალებით, ფოსტის მიმწოდებელმა ფოსტის თანამშრომელმა იცის, რომელ საფოსტო ყუთში უნდა ჩადოს წერილი(ები).

გამავალი და შემომავალი ელ.ფოსტის შეტყობინებების გასაადვილებლად ჩართულია 3 კონკრეტული პროტოკოლი:

SMTP (მარტივი ფოსტის გადაცემის პროტოკოლი) - ის გამოიყენება ელ.ფოსტის გაგზავნისთვის:

### POP3

ელ.წერილები ჩამოიტვირთება და ინახება ერთ მოწყობილობაზე.

გაგზავნილი შეტყობინებები ინახება ერთ მოწყობილობაზე, საიდანაც გაიგზავნა წერილი.

ელფოსტაზე წვდომა შესაძლებელია მხოლოდ ერთი მოწყობილობიდან, რომელზედაც წერილები ჩამოიტვირთა.

თუ გსურთ შეტყობინებების სერვერზე შენარჩუნება, დარწმუნდით, რომ ჩართულია პარამეტრი „ელფოსტის სერვერზე შენარჩუნება“, ან ყველა შეტყობინება წაიშლება სერვერიდან ერთი მოწყობილობის აპში ან პროგრამულ უზრუნველყოფაში ჩამოტვირთვის შემდეგ.

### IMAP

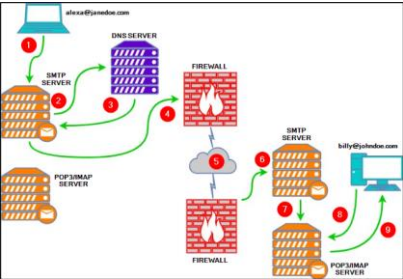
ელ.წერილები ინახება სერვერზე და მათი ჩამოტვირთვა შესაძლებელია მრავალ მოწყობილობაზე.

გაგზავნილი შეტყობინებები ინახება სერვერზე.

შეტყობინებების სინქრონიზაცია და წვდომა შესაძლებელია მრავალ მოწყობილობაში.

ქვემოთ მოცემულია თითოეული დანომრილი წერტილის ახსნა ზემოთ მოცემული დიაგრამიდან:

1. Alexa აგზავნის წერილს ბილისთვის (billy@johndoe.com) მის საყვარელ ელ.ფოსტის კლიენტში. დასრულების შემდეგ ის აჭერს გაგზავნის ღილაკს.
2. SMTP სერვერმა უნდა განსაზღვროს, სად უნდა გაგზავნოს Alexa-ს ელფოსტა. ის ითხოვს DNS johndoe.com-თან დაკავშირებული ინფორმაციისთვის.
3. DNS სერვერი იღებს ინფორმაციას johndoe.com და აგზავნის ამ ინფორმაციას SMTP სერვერზე.
4. SMTP სერვერი აგზავნის Alexa-ს ელფოსტას ინტერნეტით ბილის საფოსტო ყუთში, მისამართზე johndoe.com.
5. ამ ეტაპზე, Alexa-ს ელფოსტა გადის სხვადასხვა SMTP სერვერებზე და საბოლოოდ გადადის დანიშნულების SMTP სერვერზე.
6. Alexa-ს ელფოსტა საბოლოოდ მიაღწია დანიშნულების SMTP სერვერს.
7. Alexa-ს ელფოსტა გადამისამართებულია და ახლა ზის ადგილობრივ POP3/IMAP სერვერზე და ელოდება ბილის.
8. ბილი შედის მის ელფოსტის კლიენტში, რომელიც ითხოვს ლოკალურ POP3/IMAP სერვერს ახალი ელფოსტისთვის მის საფოსტო ყუთში.
9. Alexa-ს ელფოსტა კოპირებულია (IMAP) ან გადმოწერილი (POP3) Billy-ის ელ.ფოსტის კლიენტზე.
10. და ბოლოს, თითოეულ პროტოკოლს აქვს დაკავშირებული ნაგულისხმევი პორტები და რეკომენდებული პორტები. მაგალითად, SMTP ეს არის პორტი 25.



მავნე ელ.ფოსტის სხვადასხვა ტიპები შეიძლება კლასიფიცირდეს, როგორც ერთ-ერთი შემდეგი:

**სპამი** - არასასურველი არასასურველი ელფოსტა იგზავნება დიდი რაოდენობით მიმღებებისთვის. სპამის უფრო მავნე ვარიანტი ცნობილია, როგორც MalSpam.

**ფიშინგი** - სამიზნე(ებ)ისთვის გაგზავნილი ელფოსტა, რომელიც, სავარაუდოდ, სანდო სუბიექტისგანაა, რათა მოატყუოს პირები სენსიტიური ინფორმაციის მიწოდებაში.

**Spear phishing** - გადაიყვანს ფიშინგს კიდევ ერთი ნაბიჯით, მიზნად ისახავს კონკრეტულ ინდივიდ(ებ)ს ან ორგანიზაციას, რომელიც ეძებს მგრძნობიარე ინფორმაციას.

ვეშაპების ნადირობა - მსგავსია შუბის ფიშინგს, მაგრამ ის გამიზნულია სპეციალურად C- დონის მაღალი პოზიციის მქონე პირებისთვის (CEO, CFO და ა.შ.) და მიზანი იგივეა.

**Smishing** - ფიშინგს გადააქვს მობილურ მოწყობილობებზე მობილური მომხმარებლებისთვის სპეციალურად შემუშავებული ტექსტური შეტყობინებებით.

**Vishing** - დარტყმის მსგავსია, მაგრამ სოციალური ინჟინერიის შეტევებისთვის ტექსტური შეტყობინებების გამოყენების ნაცვლად, თავდასხმები ეფუძნება ხმოვან ზარებს.

როდესაც საქმე ფიშინგს ეხება, ოპერაციული რეჟიმი ჩვეულებრივ იგივეა, რაც დამოკიდებულია ელ.ფოსტის მიზნებზე.

მაგალითად, მიზანი შეიძლება იყოს რწმუნებათა სიგელების აღება, მეორე კი კომპიუტერზე წვდომის მოპოვება.

ქვემოთ მოცემულია ფიშინგ ელ.ფოსტის საერთო მახასიათებლები:

გამგზავნის ელფოსტის სახელი/მისამართი გადაიქცევა სანდო სუბიექტად (ელფოსტის გაყალბება)

ელფოსტის სათაურის სტრიქონი და/ან ტექსტი (ტექსტი) იწერება გადაუდებელობის გრძნობით ან იყენებს გარკვეულ საკვანძო სიტყვებს, როგორიცაა ინვოისი, შეჩერებული და ა.შ.

ელფოსტის ტექსტი (HTML) შექმნილია იმისთვის, რომ დაემთხვას სანდო ერთეულს (როგორიცაა Amazon)

ელფოსტის ტექსტი (HTML) ცუდად არის ფორმატირებული ან დაწერილი (წინა პუნქტისგან განსხვავებით)

ელფოსტის ორგანო იყენებს ზოგად შინაარსს, როგორიცაა ძვირფასო სერ/ქალბატონო.

ჰიპერბმულები (ხშირად იყენებს URL-ის შემოკლების სერვისებს მისი ნამდვილი წარმოშობის დასამალად)

მავნე დანართი, რომელიც წარმოადგენს ლეგიტიმურ დოკუმენტს

შეხსენება: ჰიპერბმულებთან და დანართებთან  
მუშაობისას, ფრთხილად უნდა იყოთ, რომ შემთხვევით  
არ დააჭიროთ ჰიპერბმულს ან დანართს.

ჰიპერბმულები და IP მისამართები უნდა იყოს "გაფუჭებული".  
დეფანგირება არის URL/დომენის ან ელფოსტის მისამართის  
დაწკაპუნების გაუქმების გზა, რათა თავიდან იქნას აცილებული  
შემთხვევითი დაწკაპუნებები, რამაც შეიძლება გამოიწვიოს  
უსაფრთხოების სერიოზული დარღვევა. ის ცვლის სპეციალურ  
სიმბოლოებს, როგორიცაა "@" ელფოსტაში ან "." URL-ში,  
სხვადასხვა სიმბოლოებით. მაგალითად, უაღრესად საეჭვო  
დომენი, <http://www.suspiciousdomain.com>, შეიცვლება  
`hxxp[:]//www[.]suspiciousdomain[.]com`-ით, სანამ ის SOC-ის გუნდს  
გადაიგზავნება გამოსავლენად.  
გაანალიზეთ ელ.წერილი სათაურით email3.eml ვირტუალურ  
მანქანაში და უპასუხეთ ქვემოთ მოცემულ კითხვებს.



IP (ინტერნეტ პროტოკოლი), TCP (გადაცემის კონტროლის პროტოკოლი) და UDP (მომხმარებლის მონაცემთა პროგრამის პროტოკოლი) არის ინტერნეტ კომუნიკაციის ფუნდამენტური სამშენებლო ბლოკები. თუმცა, როგორც ნებისმიერ ტექნოლოგიას, მათ აქვთ მოწყვლადობა და მათი გამოყენება შესაძლებელია სხვადასხვა საფრთხეებით. აქ მოცემულია IP, TCP და UDP დაუცველობისა და საფრთხეების მიმოხილვა:

IP (ინტერნეტ პროტოკოლი) დაუცველობა და საფრთხეები:

IP გაყალბება: თავდამსხმელებს შეუძლიათ გააყალბონ პაკეტების წყაროს IP მისამართი სანდო ერთეულის იმიტირებისთვის, რაც გამოიწვევს პოტენციურ არაავტორიზებულ წვდომას ან ტრაფიკის მანიპულირებას.

IP ფრაგმენტაციის თავდასხმები: თავდამსხმელებმა შეიძლება გამოიყენონ IP-ის მიერ პაკეტების ფრაგმენტაცია, რათა გამოიწვიონ რესურსების ამოწურვა ან თავიდან აიცილონ უსაფრთხოების ზომები.

IP მისამართის ამოწურვა: IPv4 მისამართების შეზღუდულმა მიწოდებამ გამოიწვია მისამართების ამოწურვა, რამაც შეიძლება გამოიწვიოს მარშრუტიზაციის არაეფექტურობა და უსაფრთხოების რისკები. IPv6-ის მიღება არის შერბილების სტრატეგია.

IP მარშრუტიზაციის შეტევები: თავდამსხმელებს შეუძლიათ მანიპულირება მარშრუტიზაციის პროტოკოლებით, როგორცაა BGP (საზღვრის კარიბჭის პროტოკოლი), რათა გადამისამართონ ტრაფიკი მავნე კვანძებში, რაც გამოიწვევს მოსმენას ან ტრაფიკის ჩაჭრას.

IPSEC ხარვეზები: IPSEC (ინტერნეტ პროტოკოლის უსაფრთხოება) გამოიყენება უსაფრთხო კომუნიკაციისთვის, მაგრამ დაუცველობამ მის განხორციელებაში ან არასწორ კონფიგურაციაში შეიძლება საფრთხე შეუქმნას უსაფრთხოებას.



TCP (გადაცემის კონტროლის პროტოკოლი) დაუცველობა და საფრთხეები:

TCP/IP სტეკის ექსპლოიტები: ოპერაციული სისტემების TCP/IP დასტაში არსებული დაუცველობის გამოყენება შესაძლებელია კოდის დისტანციური შესრულების ან მომსახურების უარყოფის (DoS) შეტევებისთვის.

SYN Flood Attacks: თავდამსხმელები ადიდებენ სამიზნე სერვერს SYN მოთხოვნების დიდი მოცულობით, აჭარბებენ მის რესურსებს და იწვევს მას უპასუხოდ.

Man-in-the-Middle (MitM) თავდასხმები: თავდამსხმელებს შეუძლიათ შეაჩერონ და მანიპულირონ TCP ტრაფიკი კომუნიკაციის მხარეებს შორის პოზიციონირებით.

სესიის გატაცება: თავდამსხმელები აკონტროლებენ დადგენილ TCP სესიას, ხშირად სესიის ქუქი-ფაილების მოპარვით ან მოწყვლადობის ექსპლუატაციით, მომხმარებლის სახელის მოსაპოვებლად და არავტორიზებული წვდომის მისაღებად.

TCP მიმდევრობითი ნომრის პროგნოზირებადობა: პროგნოზირებადი მიმდევრობის ნომრები შეიძლება გამოიყენონ თავდამსხმელებმა მავნე მონაცემების TCP ნაკადებში შესაყვანად.

UDP (User Datagram Protocol) დაუცველობა და საფრთხეები:

UDP გაძლიერების შეტევები: თავდამსხმელები აგზავნიან მცირე UDP მოთხოვნებს ღია სერვერებზე, როგორცაა DNS ან NTP სერვერები, გაყალბებული წყაროს IP მისამართებით. შემდეგ სერვერი რეაგირებს მსხვერპლის მისამართზე ბევრად უფრო დიდი პასუხით, რაც იწვევს ქსელის გადატვირთულობას და აძლიერებს შეტევას.

UDP ასახვის შეტევები: გამაძლიერებელი შეტევების მსგავსად, ასახვის შეტევები იყენებენ ღია UDP სერვისებს ტრაფიკის სამიზნეზე გადამისამართებლად, რაც იწვევს DoS შეტევას.

პაკეტის დაკარგვა: UDP-ს არ გააჩნია შეცდომების აღმოჩენისა და გამოსწორების მექანიზმები, რაც მას მგრძნობიარეს ხდის პაკეტის დაკარგვისა და მონაცემთა კორუფციის მიმართ.

ავთენტიფიკაციის გარეშე: UDP არ უზრუნველყოფს ჩაშენებულ ავთენტიფიკაციას, რაც კომუნიკაციას დაუცველს ტოვებს ჩარევის ან ხელყოფის მიმართ.

პორტის სკანირება: თავდამსხმელები ხშირად იყენებენ UDP პორტის სკანირებას ღია პორტებისა და სერვისების იდენტიფიცირებისთვის, რომლებიც შეიძლება დაუცველი იყოს ექსპლუატაციისთვის.

HTTP (ჰიპერტექსტის გადაცემის პროტოკოლი): ეს არის მსოფლიო ქსელში მონაცემთა კომუნიკაციის საფუძველი, რომელიც გამოიყენება ვებ გვერდებისა და მათი კომპონენტების გადასაცემად.

HTTPS (ჰიპერტექსტის გადაცემის პროტოკოლი უსაფრთხო): ეს არის HTTP-ის უსაფრთხო ვერსია, რომელიც შიფრავს მონაცემთა გადაცემას, უზრუნველყოფს ვებ კომუნიკაციების კონფიდენციალურობას და მთლიანობას.

TCP (გადაცემის კონტროლის პროტოკოლი): ეს არის კავშირზე ორიენტირებული პროტოკოლი, რომელიც უზრუნველყოფს მონაცემთა საიმედო და მოწესრიგებულ მიწოდებას ქსელურ კომუნიკაციაში.

UDP (User Datagram Protocol): ეს არის უკავშირო პროტოკოლი, რომელიც გთავაზობთ მონაცემთა უფრო სწრაფ გადაცემას, მაგრამ საიმედოობისა და შეკვეთის გარანტიების გარეშე.

სოციალური ინჟინერია: ეს არის მანიპულირების ტექნიკა, რომელიც გამოიყენება ინდივიდების ან ორგანიზაციების მოსატყუებლად კონფიდენციალური ინფორმაციის გამოსავლენად, როგორც წესი, მავნე მიზნებისთვის.

სხვა მხარეები: უსაფრთხოების კონტექსტში, „სხვა მხარეები“ ეხება გარე სუბიექტებს ან პირებს, რომლებსაც შეუძლიათ საფრთხე შეუქმნან ორგანიზაციის ინფორმაციულ უსაფრთხოებას, მათ შორის ჰაკერებს, კონკურენტებს ან მავნე აქტორებს.

DHCP (Dynamic Host Configuration Protocol): ეს არის ქსელის პროტოკოლი, რომელიც ავტომატურად ანიჭებს IP მისამართებს და ქსელის კონფიგურაციებს ქსელში არსებულ მოწყობილობებს, რაც ეხმარება ქსელის რესურსების მართვასა და დაცვას.

ARP (Address Resolution Protocol): იგი გამოიყენება IP მისამართის ფიზიკურ MAC მისამართზე ლოკალურ ქსელში გამოსაყენებლად, რაც ხელს უწყობს მონაცემთა პაკეტის მარშრუტიზაციას ქსელში.

Malware - მავნე პროგრამული უზრუნველყოფა, რომელიც სპეციალურად არის შექმნილი კომპიუტერულ სისტემებზე, ქსელებზე ან მომხმარებლის მონაცემებზე არასანქცირებული წვდომის, ექსპლუატაციის ან არაავტორიზებული წვდომის მოსაპოვებლად.

პორტის გამოყენების გაგება და მართვა მნიშვნელოვანია ქსელის უსაფრთხოებისთვის და იმის უზრუნველსაყოფად, რომ სწორ სერვისებსა და აპლიკაციებს შეუძლიათ ეფექტური კომუნიკაცია, არაავტორიზებული წვდომის შემოწმებისას

Malware, მოკლედ "მავნე პროგრამული უზრუნველყოფა", ეხება ნებისმიერ პროგრამულ პროგრამას ან კოდს, რომელიც სპეციალურად შექმნილია კომპიუტერული სისტემების, ქსელების ან მოწყობილობების დაზიანების, ექსპლუატაციის ან კომპრომისისთვის. მავნე პროგრამებს შეიძლება ჰქონდეს სხვადასხვა ფორმები და გამოყენებული იქნას მავნე მიზნების ფართო სპექტრისთვის. მავნე პროგრამების ზოგიერთი გავრცელებული ტიპი მოიცავს:

**1.ვირუსები:** ვირუსები არის თვითგანმეორებადი პროგრამები, რომლებიც თავს უმაგრებენ ლეგიტიმურ ფაილებს ან პროგრამულ უზრუნველყოფას. როდესაც ინფიცირებული ფაილი შესრულებულია, ვირუსი შეიძლება გავრცელდეს სხვა ფაილებზე და დააზიანოს ან დაზიანდეს სისტემა.

**2.Worms:** Worms არის დამოუკიდებელი პროგრამები, რომლებსაც შეუძლიათ თვითრეპლიკაცია და გავრცელება ქსელებში მასპინძელი ფაილის საჭიროების გარეშე. მათ შეუძლიათ მოიხმარონ სისტემის რესურსები და ხშირად ავრცელებენ ქსელის უსაფრთხოების დაუცველობებს.

**3.ტროასები:** ტროას ცხენები, ან ტროასები, არის მავნე პროგრამები, რომლებიც თავს იფარებენ ლეგიტიმურ პროგრამულ უზრუნველყოფას. ისინი ატყუებენ მომხმარებლებს, რომ გადმოწერონ ან შეასრულონ ისინი, რაც თავდამსხმელებს უნებართვო წვდომას აძლევს ინფიცირებულ სისტემაში.

**4.Ransomware:** Ransomware შიფრავს მსხვერპლის ფაილებს და ითხოვს გამოსასყიდს გაშიფვრის გასაღების სანაცვლოდ. გამოსასყიდის გადახდა არ არის რეკომენდირებული, რადგან არ არსებობს გარანტია, რომ თავდამსხმელი მისცემს გასაღებს.

**5.Spyware:** Spyware შექმნილია ინფორმაციის შესაგროვებლად მომხმარებლის ონლაინ აქტივობების შესახებ, როგორიცაა კლავიშების დაჭერა, დათვალიერების ისტორია ან პირადი ინფორმაცია. ეს მონაცემები ხშირად უბრუნდება თავდამსხმელს მომხმარებლის თანხმობის გარეშე.

**6.Adware:** Adware, მოკლედ "რეკლამით მხარდაჭერილი პროგრამული უზრუნველყოფა", აჩვენებს ინტერუზიულ რეკლამებს მომხმარებლის კომპიუტერზე, რაც ხშირად აწარმოებს შემოსავალს თავდამსხმელისთვის. მიუხედავად იმისა, რომ არ არის ისეთი მავნე, როგორც ზოგიერთი სხვა სახის მავნე პროგრამა, ის შეიძლება იყოს შემამფოთებელი და უარყოფითად იმოქმედოს სისტემის მუშაობაზე.

**7.ბოტნეტები:** ბოტნეტები არის კომპრომეტირებული კომპიუტერების (ბოტების) ქსელები, რომლებიც შეიძლება კონტროლდებოდეს ერთი ერთეულის მიერ. ისინი ხშირად გამოიყენება სხვადასხვა მავნე მიზნებისთვის, როგორიცაა განაწილებული სერვისის უარყოფის (DDoS) შეტევების გაშვება ან სპამის გაგზავნა.

**8.Rootkits:** Rootkits არის მავნე პროგრამა, რომელიც მალავს მათ არსებობას ინფიცირებულ სისტემაში. მათ შეუძლიათ თავდამსხმელებს სისტემაში მუდმივი და ამაღლებული წვდომა მისცენ, რაც ართულებს აღმოჩენასა და ამოღებას.

Malware მავნე პროგრამების გავრცელება შესაძლებელია სხვადასხვა გზით, მათ შორის ინფიცირებული ელფოსტის დანართებით, მავნე ვებსაიტებით, პროგრამული უზრუნველყოფის ჩამოტვირთვით და მოსახსნელი მედიით. თქვენი სისტემების მავნე პროგრამებისგან დასაცავად აუცილებელია განახლებული ანტივირუსული და მავნე პროგრამების გამოყენება, თქვენი ოპერაციული სისტემის და პროგრამული აპლიკაციების შენახვა უსაფრთხოების უახლესი განახლებებით, სიფრთხილე გამოიჩინოთ ფაილების ჩამოტვირთვისას ან ბმულებზე დაწკაპუნებისას და შეინარჩუნოთ ძლიერი ძალა. , უნიკალური პაროლები.

გარდა ამისა, გადამწყვეტია საკუთარი თავის და თქვენი ორგანიზაციის მომხმარებლების განათლება კიბერუსაფრთხოების საუკეთესო პრაქტიკის შესახებ, რათა შემცირდეს მავნე პროგრამების თავდასხმების მსხვერპლი გახდომის რისკი.

# ინფორმაციული უსაფრთხოება

ლექცია 10

tamar.kurdadze@btu.edu.ge

- განსახილველი საკითხები:
- ანტივირუსული უსაფრთხოება;
- ჰოსტზე დაფუძნებული შემოჭრების სისტემები;
- აპლიკაციების უსაფრთხოება;
- საბოლოო მოწყობილობის სისუსტეების შეფასება;
- ტოპ ანტივირუსული პროგრამების მიმოხილვა

ანტივირუსული (AV) პროგრამული უზრუნველყოფა არის ჰოსტზე დაფუძნებული უსაფრთხოების ერთ-ერთი აუცილებელი გადაწყვეტა, რომელიც ხელმისაწვდომია საბოლოო მომხმარებლის აპარატში მავნე პროგრამების შეტევების აღმოსაჩენად და თავიდან ასაცილებლად. AV პროგრამული უზრუნველყოფა შედგება სხვადასხვა მოდულისგან, ფუნქციებისა და გამოვლენის ტექნიკისგან. როგორც წითელი გუნდის წევრი ან პენტესტერი, აუცილებელია იცოდეთ და გესმოდეთ, თუ როგორ მუშაობს AV პროგრამული უზრუნველყოფა და მისი აღმოჩენის ტექნიკები . ამ ცოდნის შეძენის შემდეგ, გაგიადვილდებათ მუშაობა AV აცილების ტექნიკაზე.



რა არის AV პროგრამული უზრუნველყოფა?

ანტივირუსული (AV) პროგრამა არის უსაფრთხოების დამატებითი ფენა, რომელიც მიზნად ისახავს აღმოაჩინოს და თავიდან აიცილოს მავნე ფაილების შესრულება და გავრცელება სამიზნე ოპერაციულ სისტემაში.

ეს არის ჰოსტზე დაფუძნებული აპლიკაცია, რომელიც მუშაობს რეალურ დროში (ფონზე) მიმდინარე და ახლად გადმოწერილი ფაილების მონიტორინგისა და შესამოწმებლად. AV პროგრამული უზრუნველყოფა ამოწმებს და წყვეტს, არის თუ არა ფაილები მავნე სხვადასხვა ტექნიკის გამოყენებით.

საინტერესოა, რომ პირველი ანტივირუსული პროგრამა შეიქმნა მხოლოდ კომპიუტერული ვირუსების აღმოსაჩენად და მოსაშორებლად. დღესდღეობით, ეს შეიცვალა; თანამედროვე ანტივირუსულ აპლიკაციებს შეუძლიათ აღმოაჩინონ და წაშალონ კომპიუტერული ვირუსები, ისევე როგორც სხვა მავნე ფაილები და საფრთხეები.

## რას ეძებს **AV** პროგრამული უზრუნველყოფა?

ტრადიციული AV პროგრამული უზრუნველყოფა ეძებს MALWARE-ს წინასწარ განსაზღვრული მავნე შაბლონებით ან ხელმოწერებით. მავნე პროგრამა არის მავნე პროგრამული უზრუნველყოფა, რომლის მთავარი მიზანია ზიანი მიაყენოს სამიზნე მანქანას, მათ შორის, მაგრამ არ შემოიფარგლება მხოლოდ:

- მიიღეთ სრული წვდომა სამიზნე მანქანაზე.
- მოიპარეთ მგრძნობიარე ინფორმაცია, როგორიცაა პაროლები.
- ფაილების დაშიფვრა და ფაილების დაზიანება.
- შეიტანეთ სხვა მავნე პროგრამული უზრუნველყოფა ან არასასურველი რეკლამა.
- გამოიყენა კომპრომეტირებული მანქანა შემდგომი შეტევების შესასრულებლად, როგორიცაა ბოტნეტის შეტევები.

## **AV vs other security products**

გარდა AV პროგრამული უზრუნველყოფის, სხვა ჰოსტზე დაფუძნებული უსაფრთხოების გადაწყვეტილებები უზრუნველყოფს რეალურ დროში დაცვას საბოლოო წერტილის მოწყობილობებს. ბოლო წერტილის გამოვლენა და რეაგირება (EDR) არის უსაფრთხოების გადაწყვეტა, რომელიც უზრუნველყოფს რეალურ დროში დაცვას ქცევითი ანალიტიკის საფუძველზე. ანტივირუსული აპლიკაცია ახორციელებს მავნე ფაილების სკანირებას, აღმოჩენას და წაშლას. მეორეს მხრივ, EDR აკონტროლებს უსაფრთხოების სხვადასხვა შემოწმებას სამიზნე მანქანაში, მათ შორის ფაილის აქტივობებს, მეხსიერებას, ქსელურ კავშირებს, Windows რეესტრს, პროცესებს და ა.შ. თანამედროვე ანტივირუსული პროდუქტები დანერგილია ტრადიციული ანტივირუსული ფუნქციებისა და სხვა მოწინავე ფუნქციების (EDR ფუნქციების მსგავსი) ინტეგრირებისთვის ერთ პროდუქტში, რათა უზრუნველყოს ყოვლისმომცველი დაცვა ციფრული საფრთხეებისგან.

## **AV software in the past and present AV**

### **პროგრამული უზრუნველყოფა წარსულში და აწმყოში**

McAfee Associates, Inc.-მ დაიწყო პირველი AV პროგრამული უზრუნველყოფის დანერგვა 1987 წელს. მას ეწოდა „VirusScan“ და მისი მთავარი მიზანი იმ დროისთვის იყო ვირუსის „ტვინის“ ამოღება, რომელმაც დააინფიცირა ჯონ მაკაფის კომპიუტერი. მოგვიანებით სხვა კომპანიებიც შეუერთდნენ ვირუსების წინააღმდეგ ბრძოლას. AV პროგრამას ეწოდა სკანერები და ისინი წარმოადგენდნენ ბრძანების ხაზის პროგრამას, რომელიც ეძებდა მავნე შაბლონებს ფაილებში

• მას შემდეგ ყველაფერი შეიცვალა. AV პროგრამული უზრუნველყოფა დღესდღეობით იყენებს მომხმარებლის გრაფიკულ ინტერფეისს (GUI) მავნე ფაილების და სხვა ამოცანების სკანირებისთვის. მავნე პროგრამები ასევე გაფართოვდა და ახლა მიზნად ისახავს მსხვერპლს Windows-ზე და სხვა ოპერაციულ სისტემებზე. თანამედროვე AV პროგრამული უზრუნველყოფის მხარდაჭერა

მოწყობილობებისა და პლატფორმების უმეტესობა, მათ შორის Windows, Linux, macOS, Android და iOS. თანამედროვე AV პროგრამული უზრუნველყოფა გაუმჯობესდა და გახდა უფრო ინტელექტუალური და დახვეწილი, რადგან ისინი შეფუთულია მრავალმხრივი ფუნქციების პაკეტს, მათ შორის ანტივირუსს, ანტი-ექსპლოიტს, Firewall-ს, დაშიფვრის ხელსაწყოს და ა.შ..

## Antivirus Engines

AV ძრავა პასუხისმგებელია მავნე კოდებისა და ფაილების პოვნასა და წაშლაზე. კარგი AV პროგრამული უზრუნველყოფა ახორციელებს ეფექტურ და მყარ AV ბირთვს, რომელიც ზუსტად და სწრაფად აანალიზებს მავნე ფაილებს. ასევე, მან უნდა ატაროს და მხარი დაუჭიროს სხვადასხვა ტიპის ფაილებს, მათ შორის საარქივო ფაილებს, სადაც მას შეუძლია თვითმმართველობის ამონაწერი და შეამოწმოს ყველა შეკუმშული ფაილი. AV პროდუქტების უმეტესობა იზიარებს ერთსა და იმავე საერთო ფუნქციებს, მაგრამ განსხვავებულად არის დანერგილი, მათ შორის, მაგრამ არ შემოიფარგლება მხოლოდ:

- Scanner(სკანერი)
- Detection techniques(გამოვლენის ტექნიკა)
- Compressors and Archives(კომპრესორები და არქივები)
- Unpackers(გამაფხვიერებელი)
- Emulators(ემულატორები)

## Scanner

სკანერის ფუნქცია შედის AV პროდუქტების უმეტესობაში: AV პროგრამული უზრუნველყოფა მუშაობს და სკანირებს რეალურ დროში ან მოთხოვნის შესაბამისად. ეს ფუნქცია ხელმისაწვდომია GUI-ში ან ბრძანების სტრიქონში. მომხმარებელს შეუძლია გამოიყენოს ის, როცა საჭიროა ფაილების ან დირექტორიების შესამოწმებლად. სკანირების ფუნქციამ უნდა დაუჭიროს მხარი ყველაზე ცნობილ მავნე ფაილის ტიპებს საფრთხის აღმოსაჩენად და მოსაშორებლად. გარდა ამისა, მას ასევე შეუძლია მხარი დაუჭიროს სკანირების სხვა ტიპებს, AV პროგრამული უზრუნველყოფის მიხედვით, მათ შორის დაუცველობის, ელ. ფოსტის, Windows მეხსიერების და Windows რეესტრის ჩათვლით.

## Detection techniques

AV გამოვლენის ტექნიკა ეძებს და აღმოაჩენს მავნე ფაილებს; გამოვლენის სხვადასხვა ტექნიკა შეიძლება გამოყენებულ იქნას AV ძრავში, მათ შორის :

- ხელმოწერებზე დაფუძნებული ამოცნობა არის ტრადიციული AV ტექნიკა, რომელიც ეძებს წინასწარ განსაზღვრულ მავნე შაბლონებს და ხელმოწერებს ფაილებში.

- ევრისტიკული გამოვლენა არის უფრო მოწინავე ტექნიკა, რომელიც მოიცავს სხვადასხვა ქცევის მეთოდებს საეჭვო ფაილების გასაანალიზებლად

- დინამიური გამოვლენა არის ტექნიკა, რომელიც მოიცავს სისტემის ზარების და API-ების მონიტორინგს და ტესტირებას და ანალიზს იზოლირებულ გარემოში.

ჩვენ განვიხილავთ ამ ტექნიკას შემდეგ ამოცანაში. კარგი AV ძრავა ზუსტია და სწრაფად ამოიცნობს მავნე ფაილებს ნაკლები ცრუ დადებითი შედეგით.

## Compressors and Archives

ფუნქცია „კომპრესორები და არქივები“ უნდა იყოს ჩართული ნებისმიერ AV პროგრამაში.

მას უნდა ჰქონდეს მხარდაჭერილი და შეუძლია გაუმკლავდეს სისტემის ფაილების სხვადასხვა ტიპებს, მათ შორის შეკუმშულ ან დაარქივებულ ფაილებს: ZIP, TGZ, 7z, XAR, RAR და ა.შ. მავნე კოდი ხშირად ცდილობს თავი აარიდოს მასპინძელზე დაფუძნებულ უსაფრთხოების გადაწყვეტილებებს შეკუმშულ ფაილებში დამალვით. ამ მიზეზით, AV პროგრამული უზრუნველყოფა უნდა დეკომპრესირდეს და დაასკანირებდეს ყველა ფაილს სანამ მომხმარებელი გახსნის ფაილს არქივში.

## **PE (Portable Executable) Parsing and Unpackers**

მავენე პროგრამა მალავს და აფუჭებს თავის მავნე კოდს, მისი შეკუმშვით და დაშიფვრით ტვირთის ფარგლებში. ის დეკომპრესირებს და შიფრავს თავს მუშაობის დროს, რათა გაართულოს სტატიკური ანალიზის შესრულება. ამრიგად, AV პროგრამულმა პროგრამამ უნდა შეძლოს ამოიცნოს და განბლოკოს ცნობილი შემფუთვლების უმეტესი ნაწილი (UPX, Armadillo, ASPack და ა.შ.) სტატიკური ანალიზის გაშვებამდე.

Malware დეველოპერები იყენებენ სხვადასხვა ტექნიკას, როგორიცაა შეფუთვა, ზომის შესამცირებლად და მავნე ფაილის სტრუქტურის შესაცვლელად. შეფუთვა შეკუმშავს თავდაპირველ შესრულებად ფაილს, რათა გაძნელდეს ანალიზი. ამრიგად, AV პროგრამას უნდა ჰქონდეს Unpacker ფუნქცია, რომ დაიცვან დაცული ან შეკუმშული შესრულებული ფაილები ორიგინალ კოდში.

კიდევ ერთი ფუნქცია, რომელიც AV პროგრამას უნდა ჰქონდეს არის Windows Portable Executable (PE) header parser. შესრულებადი ფაილების PE გაანალიზება დაგეხმარებათ განასხვავოთ მავნე და ლეგიტიმური პროგრამული უზრუნველყოფა (.exe ფაილები). PE ფაილის ფორმატი Windows-ში (32 და 64 ბიტი) შეიცავს სხვადასხვა ინფორმაციას და რესურსებს, როგორიცაა ობიექტის კოდი, DLL, ხატის ფაილები, შრიფტის ფაილები და ძირითადი ნაგავსაყრელები.

## Emulators

ემულატორი არის ანტივირუსული ფუნქცია, რომელიც შემდგომ ანალიზს აკეთებს საეჭვო ფაილებზე. როგორც კი ემულატორი მიიღებს მოთხოვნას, ემულატორი აწარმოებს საეჭვო (exe, DLL, PDF და ა.შ.) ფაილებს ვირტუალიზებულ და კონტროლირებულ გარემოში. ის აკონტროლებს შესრულებადი ფაილების ქცევას შესრულების დროს, Windows API-ის ზარების, რეესტრის და სხვა Windows ფაილების ჩათვლით. ქვემოთ მოცემულია იმ არტეფაქტების მაგალითები, რომლებიც შეიძლება შეაგროვოს ემულატორმა :

- API calls
- Memory dumps
- Filesystem modifications
- Log events
- Running processes
- Web requests

ემულატორი აჩერებს ფაილის შესრულებას, როდესაც შეგროვდება საკმარისი არტეფაქტები MALWARE-ს გამოსავლენად.

## Other common features

ქვემოთ მოცემულია რამდენიმე საერთო ფუნქცია, რომელიც გვხვდება AV პროდუქტებში :

- თვითდაცვის დრაივერი, რომელიც იცავს მავნე პროგრამებს, რომლებიც თავს დაესხმიან რეალურ AV-ს.
- Firewall და ქსელის შემოწმების ფუნქციონალობა.
- ბრძანების ხაზი და გრაფიკული ინტერფეისის ინსტრუმენტები.
- A daemon or service.
- A management console.



# ინფორმაციული უსაფრთხოება

ლექცია 7

tamar.kurdadze@btu.edu.ge

- ფიზიკური წვდომის კონტროლი;
- ლოგიკური წვდომის კონტროლი;
- ადმინისტრაციული წვდომის კონტროლი;
- სავალდებულო წვდომის კონტროლი, დისკრეციული წვდომის კონტროლი;
- როლზე დაფუძნებული კონტროლი, წესზე დაფუძნებული წვდომის კონტროლი;
- იდენტიფიკაცია, აუთენტიკაცია, ავტორიზაცია, ანგარიშგება;

## რა არის წვდომის კონტროლი?

წვდომის კონტროლი არის უსაფრთხოების მექანიზმი, რომელიც გამოიყენება იმის გასაკონტროლებლად, რომელ მომხმარებლებს ან სისტემებს აქვთ უფლება შევიდნენ კონკრეტულ რესურსზე ან სისტემაზე. წვდომის კონტროლი დანერგილია კომპიუტერულ სისტემებში იმისთვის, რომ მხოლოდ ავტორიზებულ მომხმარებლებს ჰქონდეთ წვდომა რესურსებზე, როგორცაა ფაილები, დირექტორიები, მონაცემთა ბაზები და ვებ გვერდები. წვდომის კონტროლის უპირველესი მიზანია დაცული იყოს სენსიტიური მონაცემები და უზრუნველყოს, რომ ისინი ხელმისაწვდომი იყოს მხოლოდ მათთვის, ვისაც აქვს მასზე წვდომის უფლება.

**დისკრეციული წვდომის კონტროლი (DAC):** ამ ტიპის წვდომის კონტროლის დროს, რესურსის მფლობელი ან ადმინისტრატორი განსაზღვრავს ვის აქვს რესურსზე წვდომის უფლება და რა ქმედებების შეასრულების უფლება გააჩნია. DAC ჩვეულებრივ გამოიყენება ოპერაციულ სისტემებში და ფაილურ სისტემებში. ხალხური სიტყვებით რომ ვთქვათ, წარმოიდგინეთ ციხე, სადაც მეფეს შეუძლია მისცეს გასაღებები თავის მრჩევლებს, რაც მათ საშუალებას მისცემს გააღონ ნებისმიერი კარი, როცა მოესურვებათ. ეს არის თავისუფლება, გააკონტროლო წვდომა საკუთარ რესურსებზე. პასუხისმგებელს, ისევე როგორც ციხის მეფეს, შეუძლია ნებართვა გადასცეს ვისაც სურს, უკარნახოს ვის შეუძლია შესვლა და გამოსვლა.

სავალდებულო წვდომის კონტროლი (MAC): ამ ტიპის წვდომის კონტროლის დროს რესურსებზე წვდომა განისაზღვრება წინასწარ განსაზღვრული წესების ან პოლიტიკის კომპლექტით, რომლებიც აღსრულებულია სისტემის მიერ. MAC ჩვეულებრივ გამოიყენება მაღალ უსაფრთხო გარემოში, როგორიცაა მთავრობა და სამხედრო სისტემები. ხალხური სიტყვებით რომ ვთქვათ, წარმოიდგინეთ ციხე რკინით დაფარული უსაფრთხოების პროტოკოლით. მხოლოდ კონკრეტულ პირებს, რომლებსაც აქვთ სპეციალური უსაფრთხოების ნებართვები, შეუძლიათ წვდომა გარკვეულ ადგილებში, და ეს არ არის შეთანხმებული. უმაღლესი მეთაური ადგენს წესებს და მათ მკაცრად იცავენ. ასე მუშაობს MAC. ეს ჰგავს უსაფრთხოების მკაცრ ოფიცერს, რომელიც არ უშვებს გამონაკლისს წესში.

როლებზე დაფუძნებული წვდომის კონტროლი (**RBAC**): ამ ტიპის წვდომის კონტროლის დროს მომხმარებლებს ენიჭებათ როლები, რომლებიც განსაზღვრავენ რესურსებზე წვდომის დონეს. RBAC ჩვეულებრივ გამოიყენება საწარმოთა სისტემებში, სადაც მომხმარებლებს აქვთ უფლებამოსილების განსხვავებული დონე მათი სამუშაო პასუხისმგებლობის მიხედვით. ხალხური თვალსაზრისით, წარმოიდგინეთ თანამედროვე კორპორაცია. თქვენ გყავთ თქვენი მენეჯერები, თქვენი აღმასრულებლები, თქვენი გაყიდვების პერსონალი და ა.შ. თითოეულ მათგანს განსხვავებული წვდომა აქვს შენობაში. ზოგს შეუძლია შევიდეს გამგეობის დარბაზში, ზოგს შეუძლია გაყიდვების სართულზე შესვლა და ა.შ. ეს არის RBAC-ის არსი - წვდომის მინიჭება ორგანიზაციაში პიროვნების როლზე დაყრდნობით.

ატრიბუტზე დაფუძნებული წვდომის კონტროლი (**ABAC**): ამ ტიპის წვდომის კონტროლის დროს რესურსებზე წვდომა განისაზღვრება ატრიბუტების ნაკრებით, როგორცაა მომხმარებლის როლი, დღის დრო, მდებარეობა და მოწყობილობა. ABAC ჩვეულებრივ გამოიყენება ღრუბლოვან გარემოში და ვებ აპლიკაციებში. ხალხური თვალსაზრისით, იფიქრეთ უაღრესად მოწინავე სამეცნიერო ფანტასტიკის უსაფრთხოების სისტემაზე, რომელიც სკანირებს ინდივიდებს გარკვეული ატრიბუტების დასადგენად. შესაძლოა ის ამოწმებს, არიან თუ არა ისინი კონკრეტული პლანეტიდან, ატარებენ თუ არა კონკრეტულ მოწყობილობას, ან ცდილობენ თუ არა რესურსზე წვდომას კონკრეტულ დროს. ეს არის ABAC. ეს მომავლის ჭკვიან, მოქნილ უსაფრთხოებას ჰგავს.



## 1. ფიზიკური წვდომის კონტროლი:

**Definition:** ფიზიკური წვდომის კონტროლი ეხება ზომებს და მექანიზმებს, რომლებიც გამოიყენება ფიზიკურ სივრცეებზე, ობიექტებსა და აქტივებზე წვდომის შესაზღუდად ან მისაცემად, როგორიცაა შენობები, ოთახები ან მონაცემთა ცენტრები..

### Examples:

- ფიზიკური ბარიერები, როგორიცაა ღობეები, კედლები და კარიბჭე.
- საკეტები, გასაღები ბარათები, ბიომეტრიული სკანერები ან PIN კოდები კარებზე.
- დაცვის თანამშრომლები ან სათვალთვალო სისტემები, რომლებიც აკონტროლებენ შესასვლელებს.
- ვიზიტორთა მართვის სისტემები სტუმრების წვდომის თვალყურის დევნისა და კონტროლისთვის.

**Purpose:** აღკვეთეთ არაუფლებამოსილი პირების ფიზიკურად შეღწევა ან შეზღუდულ ზონებთან ურთიერთობა, აქტივების და პერსონალის ფიზიკური უსაფრთხოების უზრუნველყოფა.

## 2. ლოგიკური წვდომის კონტროლი:

**Definition:** ლოგიკური წვდომის კონტროლი გულისხმობს ტექნოლოგიის გამოყენებას ციფრულ სისტემებზე, ქსელებსა და მონაცემებზე წვდომის დასარეგულირებლად. ის ყურადღებას ამახვილებს მომხმარებლის ავტორიზაციის კონტროლზე და ინფორმაციულ სისტემებზე წვდომის ავტორიზაციაზე..

### Examples:

- Usernames and passwords for system login.
- Multi-factor authentication (MFA) methods.
- Role-based access control (RBAC) assigning permissions based on roles.
- Access control lists (ACLs) specifying permissions for files or network resources.

**Purpose:** დაიცავით ციფრული აქტივები, მონაცემები და საინფორმაციო სისტემები იმით, რომ მხოლოდ უფლებამოსილ პირებს შეუძლიათ მათთან წვდომა, შეცვლა ან ურთიერთქმედება.

### 3. ადმინისტრაციული დაშვების კონტროლი:

**Definition:** ადმინისტრაციული წვდომის კონტროლი მოიცავს ორგანიზაციების მიერ დადგენილ პოლიტიკას, პროცედურებსა და მმართველობით ზომებს, რათა მართონ და გააკონტროლონ წვდომა როგორც ფიზიკურ, ასევე ციფრულ რესურსებზე. იგი მოიცავს წვდომის კონტროლის მექანიზმების ზოგად მართვას.

#### **Examples:**

- წვდომის კონტროლის პოლიტიკა, რომელიც განსაზღვრავს მისაღები გამოყენებისა და წვდომის დონეს.
- Employee onboarding and offboarding procedures.
- რეგულარული უსაფრთხოების სასწავლო და ცნობიერების პროგრამები.
- აუდიტისა და მონიტორინგის წვდომის ჟურნალები შესაბამისობისთვის.

**Purpose:** უზრუნველყოს ჩარჩო ეფექტური წვდომის კონტროლის განსახორციელებლად და შესანარჩუნებლად. ადმინისტრაციული წვდომის კონტროლი უზრუნველყოფს პოლიტიკის განსაზღვრას, კომუნიკაციას და აღსრულებას ორგანიზაციის მასშტაბით.

იდენტიფიკაცია, ავთენტიფიკაცია, ავტორიზაცია და ანგარიშგება კიბერუსაფრთხოებაში წვდომის კონტროლის სისტემების ძირითადი კომპონენტებია. ეს კონცეფციები განუყოფელია ფიზიკურ და ციფრულ რესურსებზე წვდომის მართვისა და უზრუნველსაყოფად. მოდით გამოვიკვლიოთ თითოეული მათგანი:

## 1. იდენტიფიკაცია:

**Definition:** იდენტიფიკაცია არის უნიკალური იდენტიფიკატორის (როგორიცაა მომხმარებლის სახელი, თანამშრომლის ID ან ანგარიშის ნომერი) წარდგენის პროცესი სისტემაში ან ერთეულზე, რათა გამოაცხადოს პირადობა.

**Example:** Entering a username on a login screen.

**Purpose:** თავდაპირველი პრეტენზიის დადგენა იმის შესახებ, თუ ვინ არის მომხმარებელი.

## 2. ავთენტიფიკაცია:

**Definition:** ავთენტიფიკაცია არის მოთხოვნილი პირადობის გადამოწმების პროცესი რწმუნებათა სიგელების (მაგ., პაროლები, ბიომეტრია, უსაფრთხოების ნიშნები) გამოყენებით, რათა დარწმუნდეს, რომ მომხმარებელი არის ის, ვინც ამბობს, რომ არის.

**Example:** პაროლის შეყვანა ან ბიომეტრიული ავთენტიფიკაციისთვის თითის ანაბეჭდის მიწოდება.

**Purpose:** მომხმარებლის მოთხოვნილი პირადობის ლეგიტიმურობის დადასტურება.

## 4. Reporting(მონხენება):

**Definition:** ანგარიშგება მოიცავს წვდომის მოვლენებისა და აქტივობების აღრიცხვას და მონიტორინგს. იგი მოიცავს შესვლის წარმატებული და წარუმატებელი მცდელობების ჩაწერას, სენსიტიურ მონაცემებზე წვდომას და უსაფრთხოების საკითხებთან დაკავშირებულ სხვა მოვლენებს.

**Example:** აუდიტის ჟურნალების გენერირება, რომელიც აღწერს შესვლის მცდელობებს, წვდომის ნებართვებში ცვლილებებს ან უსაფრთხოების ინციდენტს.

**Purpose:** მომხმარებლის საქმიანობის ჩანაწერის უზრუნველყოფა აუდიტის, შესაბამისობისა და უსაფრთხოების ინციდენტებზე რეაგირებისთვის. ანგარიშები დაგეხმარებათ საეჭვო ქცევის იდენტიფიცირებაში, ცვლილებების თვალყურის დევნებაში და ანგარიშვალდებულების შენარჩუნებაში.

## 3. Authorization:

**Definition:** ავტორიზაცია არის მომხმარებლის ან სისტემისთვის წვდომის უფლებებისა და ნებართვების მინიჭების ან უარის თქმის პროცესი მათი დამოწმებული იდენტობისა და წვდომის დონის საფუძველზე.

**Example:** მომხმარებლისთვის კონკრეტული როლების ან ნებართვების მინიჭება სისტემაში შესვლის შემდეგ, იმის დადგენა, თუ რა ქმედებები აქვთ უფლება შეასრულონ.

**Purpose:** რესურსებზე წვდომის კონტროლი და შეზღუდვა ავტორიზებული მომხმარებლის პრივილეგიებზე დაყრდნობით.

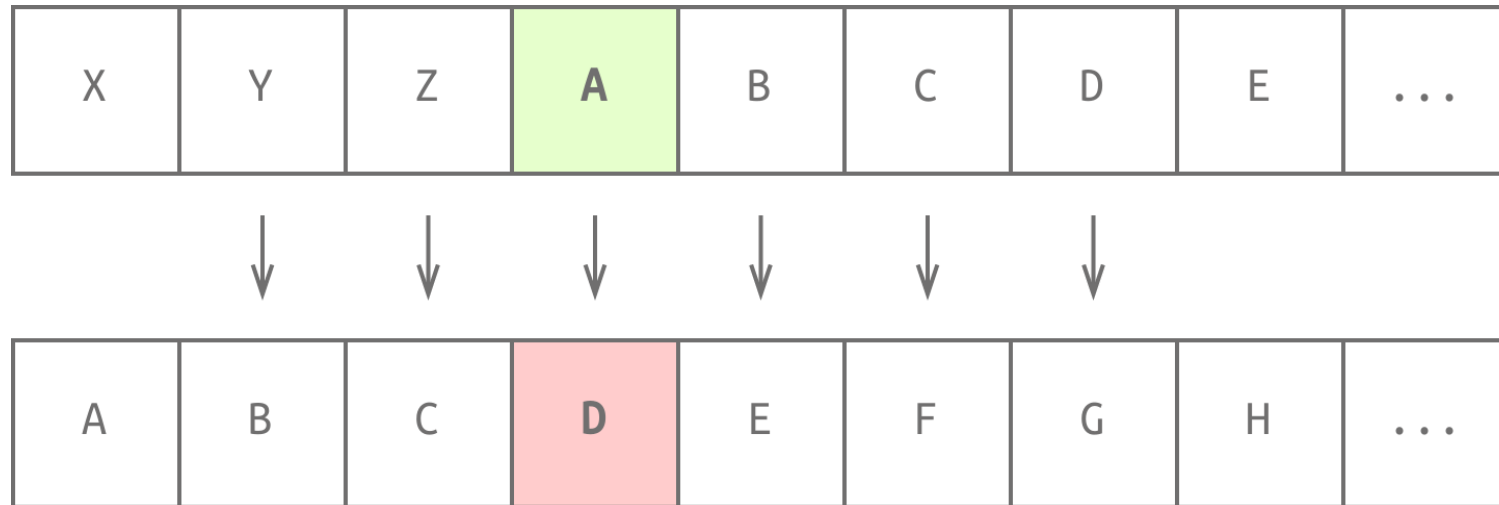
# ინფორმაციული უსაფრთხოება

ლექცია 6

tamar.kurdadze@btu.edu.ge

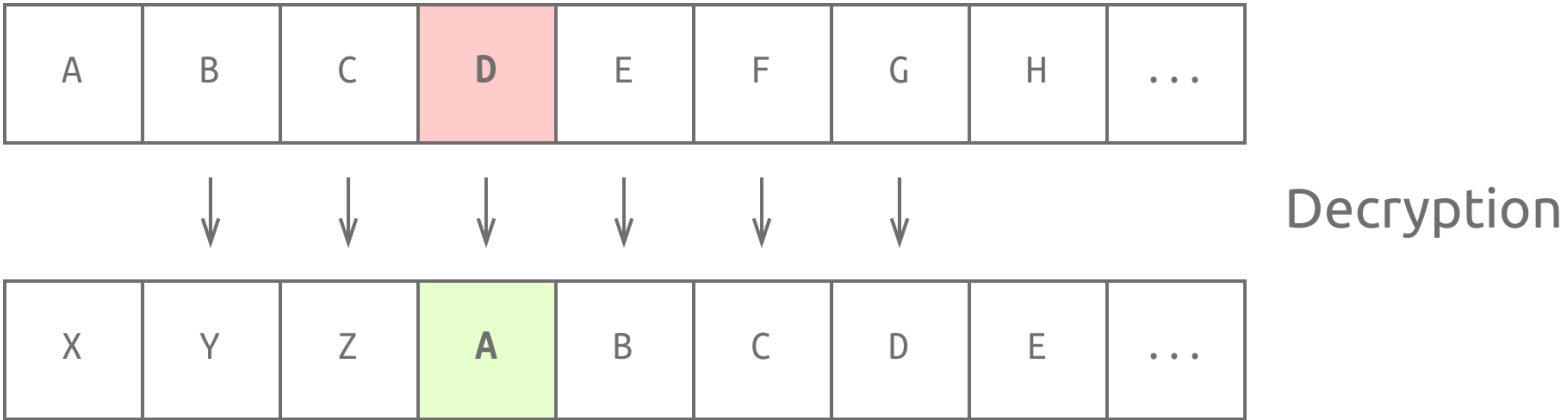
- გადანაცვლებადი შიფრი, ჩანაცვლებადი შიფრი;
- კრიპტოანალიზი, კოდის „გატეხვის“ მეთოდები;
- კრიპტოლოგია;
- კონფიდენციალობა: სიმეტრიული და ასიმეტრიული შიფრაცია, სიმეტრიული და ასიმეტრიული შიფრაციის ალგორითმები;
- ბლოკური და ნაკადური შიფრები;
- DES, 3DES, AES, RSA ალგორითმების განხილვა

ერთ-ერთი უმარტივესი შიფრია კეისრის შიფრი,  
რომელიც გამოიყენებოდა 2000 წელზე მეტი ხნის წინ.  
კეისრის შიფრი ანაცვლებს ასოს ფიქსირებული  
რაოდენობის ადგილებით მარცხნივ ან მარჯვნივ.  
განვიხილოთ დაშიფვრისთვის 3-ით მარჯვნივ  
გადასვლის შემთხვევა, როგორც ეს ნაჩვენებია ქვემოთ  
მოცემულ ფიგურაში.



Encryption

მიმღებმა უნდა იცოდეს, რომ ტექსტი გადავიდა 3-ით მარჯვნივ, ორიგინალური  
შეტყობინების აღსადგენად.

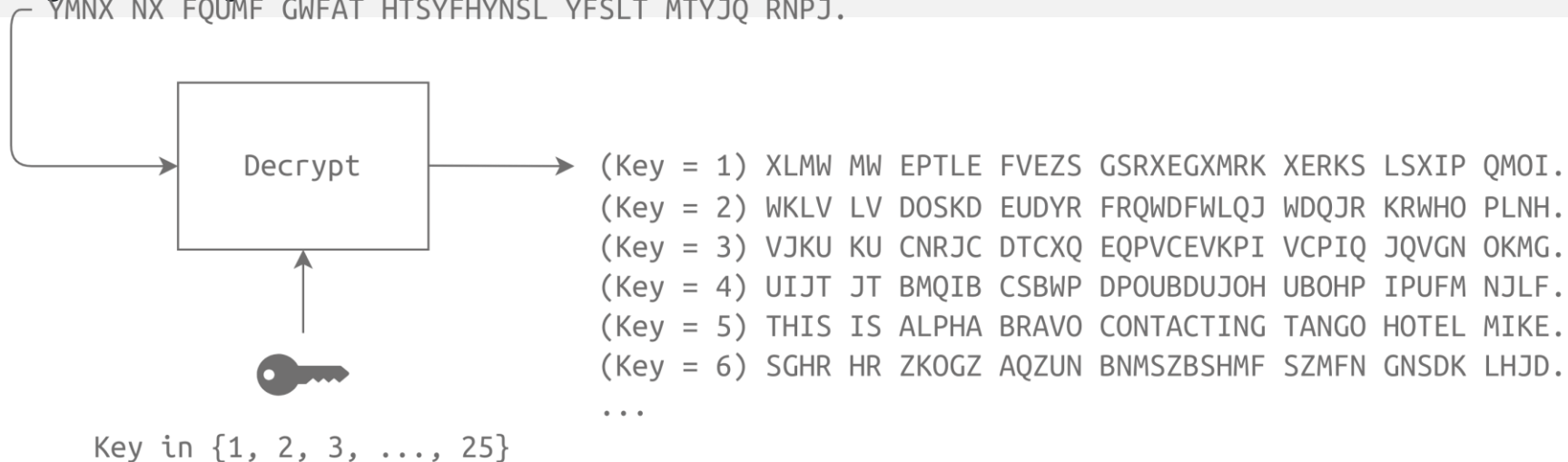




კეისრის შიფრი, რომელიც ზემოთ აღვწერეთ, შეუძლია გამოიყენოს გასაღები 1-დან 25-მდე. 1-ის ღილაკით, თითოეული ასო გადადის ერთი პოზიციით, სადაც A ხდება B, ხოლო Z ხდება A. კლავიშით 25, თითოეული ასო არის გადაინაცვლებს 25 პოზიციით, სადაც A ხდება Z, ხოლო B ხდება A. გასაღები 0 ნიშნავს ცვლილებას; უფრო მეტიც, 26-ის გასაღები ასევე არ გამოიწვევს რაიმე ცვლილებას, რადგან ეს გამოიწვევს სრულ ბრუნვას. შესაბამისად, ჩვენ ვასკვნით, რომ Caesar Cipher-ს აქვს 25-ის საკვანძო სივრცე; არის 25 სხვადასხვა გასაღები, რომელთაგანაც მომხმარებელს შეუძლია აირჩიოს.

განვიხილოთ შემთხვევა, როდესაც თქვენ ჩაწერეთ შეტყობინება, რომელიც დაშიფრულია Caesar Cipher-ის გამოყენებით: „YMNX NX FQUMF GWFAT HTSYFHYNLS YFSLT MTYJQ RNPJ“. ჩვენ გვთხოვენ მისი გაშიფვრა გასაღების ცოდნის გარეშე. ჩვენ შეგვიძლია ვცადოთ ეს უხეში ძალის გამოყენებით, ანუ შეგვიძლია ვცადოთ ყველა შესაძლო გასაღები და ვნახოთ, რომელია ყველაზე გონივრული. შემდეგ ფიგურაში ჩვენ შევამჩნიეთ, რომ გასაღები 5-ს აქვს ყველაზე აზრი, „ეს არის ALPHA BRAVO CONTACTING TANGO HOTEL MIKE“.

YMNX NX FQUMF GWFAT HTSYFHYNLS YFSLT MTYJQ RNPJ.



კეისრის შიფრი ითვლება შემცვლელ შიფრად, რადგან ანბანის თითოეული ასო ჩანაცვლებულია მეორეთი. შიფრის სხვა ტიპს ეწოდება ტრანსპოზიციური შიფრი, რომელიც შიფრავს შეტყობინებას ასოების თანმიმდევრობის შეცვლით. განვიხილოთ მარტივი ტრანსპოზიციური შიფრი ქვემოთ მოცემულ ფიგურაში. ჩვენ ვიწყებთ შეტყობინებით, „THIS IS ALPHA BRAVO CONTACTING TANGO HOTEL MIKE“ და გასაღებით 42351. მას შემდეგ რაც დავწერთ ჩვენი წერილის ასოებს ერთი სვეტის მიყოლებით შევსებით, ჩვენ ვაწყობთ სვეტებს გასაღების მიხედვით და შემდეგ ვკითხულობთ რიგებს. სხვა სიტყვებით რომ ვთქვათ, ჩვენ ვწერთ სვეტების მიხედვით და ვკითხულობთ რიგების მიხედვით. ასევე გაითვალისწინეთ, რომ ამ მაგალითში ჩვენ უგულებელვყავით ღია ტექსტის მთელი სივრცე. შედეგად მიღებული შიფრული ტექსტი „NPCOTGHOTH...“ იკითხება ერთი რიგის მიყოლებით. სხვა სიტყვებით რომ ვთქვათ, ტრანსპოზიციური შიფრი უბრალოდ აწესრიგებს ასოების თანმიმდევრობას, განსხვავებით შემცვლელი შიფრისგან, რომელიც ცვლის ასოებს მათი რიგის შეცვლის გარეშე.

T	P	C	N	O
H	H	O	G	T
I	A	N	T	E
S	B	T	A	L
I	R	A	N	M
S	A	C	G	I
A	V	T	O	K
L	O	I	H	E

Plaintext

THIS IS ALPHA BRAVO  
CONTACTING TANGO HOTEL MIKE

N	P	C	O	T
G	H	O	T	H
T	A	N	E	I
A	B	T	L	S
N	R	A	M	I
G	A	C	I	S
O	V	T	K	A
H	O	I	E	L

Ciphertext

NPCOTGHOTHANEIABTLS  
NRAMIGACISOVTKAHOIEL

ამ დავალებამ შემოიღო მარტივი ჩანაცვლებისა და ტრანსპოზიციის შიფრები და გამოიყენა ისინი ანბანური სიმბოლოებისგან შექმნილ შეტყობინებებზე. იმისათვის, რომ დაშიფვრის ალგორითმი უსაფრთხოდ ჩაითვალოს, შეუძლებელი უნდა იყოს თავდაპირველი შეტყობინების აღდგენა, ანუ ჩვეულებრივი ტექსტი. (მათემატიკური თვალსაზრისით, ჩვენ გვჭირდება მძიმე პრობლემა, ანუ პრობლემა, რომელიც ვერ გადაიჭრება მრავალწევრულ დროში. პრობლემა, რომლის გადაჭრაც შეგვიძლია მრავალწევრულ დროში, არის პრობლემა, რომლის გადაჭრაც შესაძლებელია დიდი შეყვანის შემთხვევაშიც კი, თუმცა ამას შესაძლოა დასჭირდეს კომპიუტერი. თუ დაშიფრული შეტყობინების გატეხვა შესაძლებელია ერთ კვირაში, გამოყენებული დაშიფვრა ჩაითვლება დაუცველად. თუმცა, თუ დაშიფრული შეტყობინების გატეხვა შესაძლებელია 1 მილიონ წელიწადში, დაშიფვრა ჩაითვლება პრაქტიკულად უსაფრთხოდ.

განვიხილოთ მონო-ანბანური შემცვლელი შიფრი, სადაც თითოეული ასო ახალ ასოზეა გადატანილი. მაგალითად, ინგლისურად, თქვენ დააფიქსირებთ „a“-ს 26 ინგლისური ასოდან ერთ-ერთზე, შემდეგ „b“-ს ასახავთ დარჩენილ 25 ინგლისურ ასოს და შემდეგ „c“-ს ასახავთ დარჩენილ 24 ინგლისურ ასოს. , და ასე შემდეგ.

მაგალითად, ჩვენ შეგვიძლია ავირჩიოთ ანბანის ასოები "abcdefghijklmnopqrstuvwxyz", რათა შევიტანოთ შესაბამისად "xpatvrzyjhcsdikbfwunqgmol". სხვა სიტყვებით რომ ვთქვათ, "a" ხდება "x", "b" ხდება "p" და ა.შ. მიმღებმა უნდა იცოდეს გასაღები, „xpatvrzyjhcsdikbfwunqgmol“, დაშიფრული შეტყობინებების წარმატებით გაშიფვრისთვის.

ეს ალგორითმი შეიძლება ძალიან უსაფრთხოდ გამოიყურებოდეს, მით უმეტეს, რომ ყველა შესაძლო გასაღების ცდა შეუძლებელია. თუმცა, ასეთი დაშიფვრის ალგორითმის გამოყენებით შიფრული ტექსტის გასატეხად შეიძლება გამოყენებულ იქნას სხვადასხვა ტექნიკა. ასეთი ალგორითმის ერთ-ერთი სისუსტე ასოების სიხშირეა. ინგლისურ ტექსტებში ყველაზე გავრცელებული ასოებია "e", "t" და "a", რადგან ისინი ჩნდება სიხშირით 13%, 9.1% და 8.2%, შესაბამისად. უფრო მეტიც, ინგლისურ ტექსტებში ყველაზე გავრცელებული პირველი ასოებია "t", "a" და "i", რადგან ისინი 16%, 11.7% და 7.6% შესაბამისად გამოდიან. ამას დაუმატეთ ის ფაქტი, რომ შეტყობინებების სიტყვების უმეტესობა ლექსიკონის სიტყვებია და თქვენ შეძლებთ დაშიფრული ტექსტის გატეხვას ანბანის შემცვლელი შიფრით უმოკლეს დროში.

განვიხილოთ რამდენიმე ტერმინოლოგია:

კრიპტოგრაფიული ალგორითმი ან შიფრა: ეს ალგორითმი განსაზღვრავს დაშიფვრის და გაშიფვრის პროცესებს.

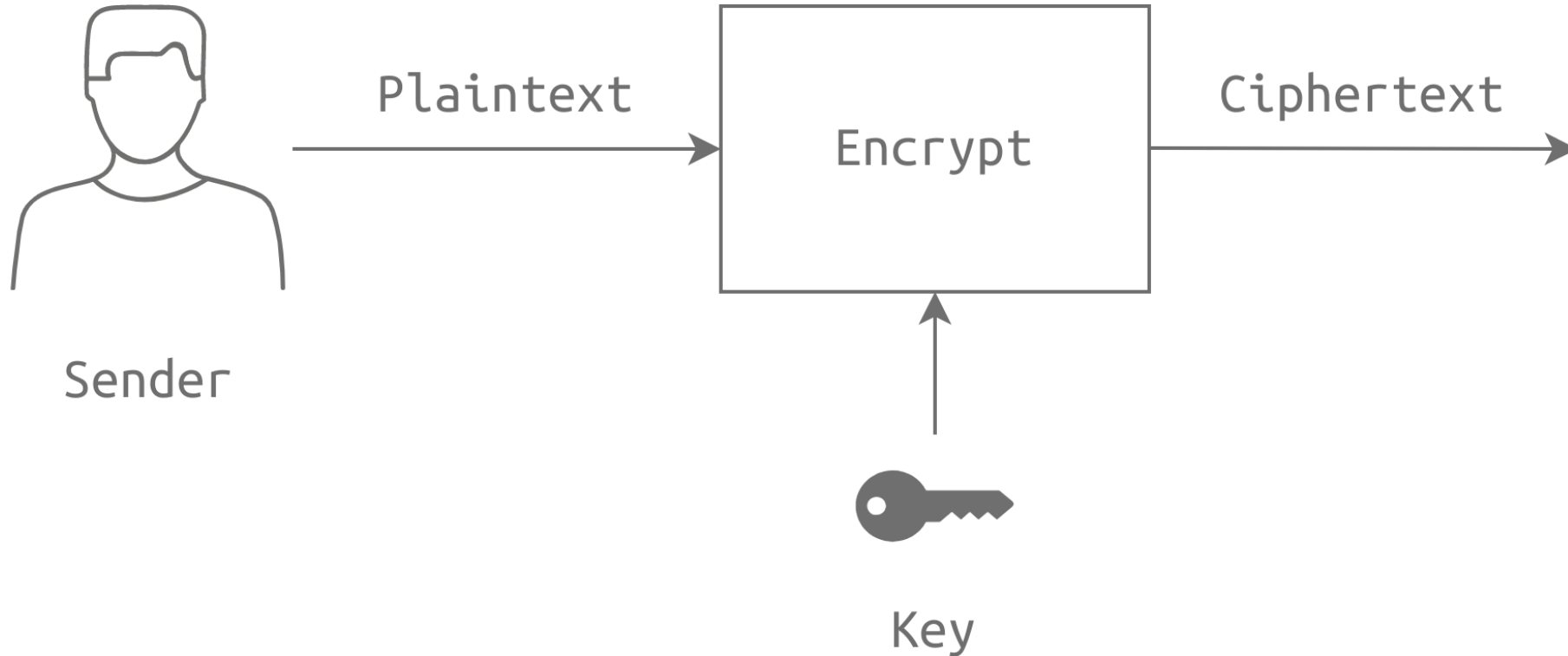
გასაღები: კრიპტოგრაფიულ ალგორითმს სჭირდება გასაღები, რათა გადაიყვანოს ჩვეულებრივი ტექსტი შიფრულ ტექსტად და პირიქით.

ჩვეულებრივი ტექსტი არის ორიგინალური შეტყობინება, რომლის დაშიფვრა გვინდა

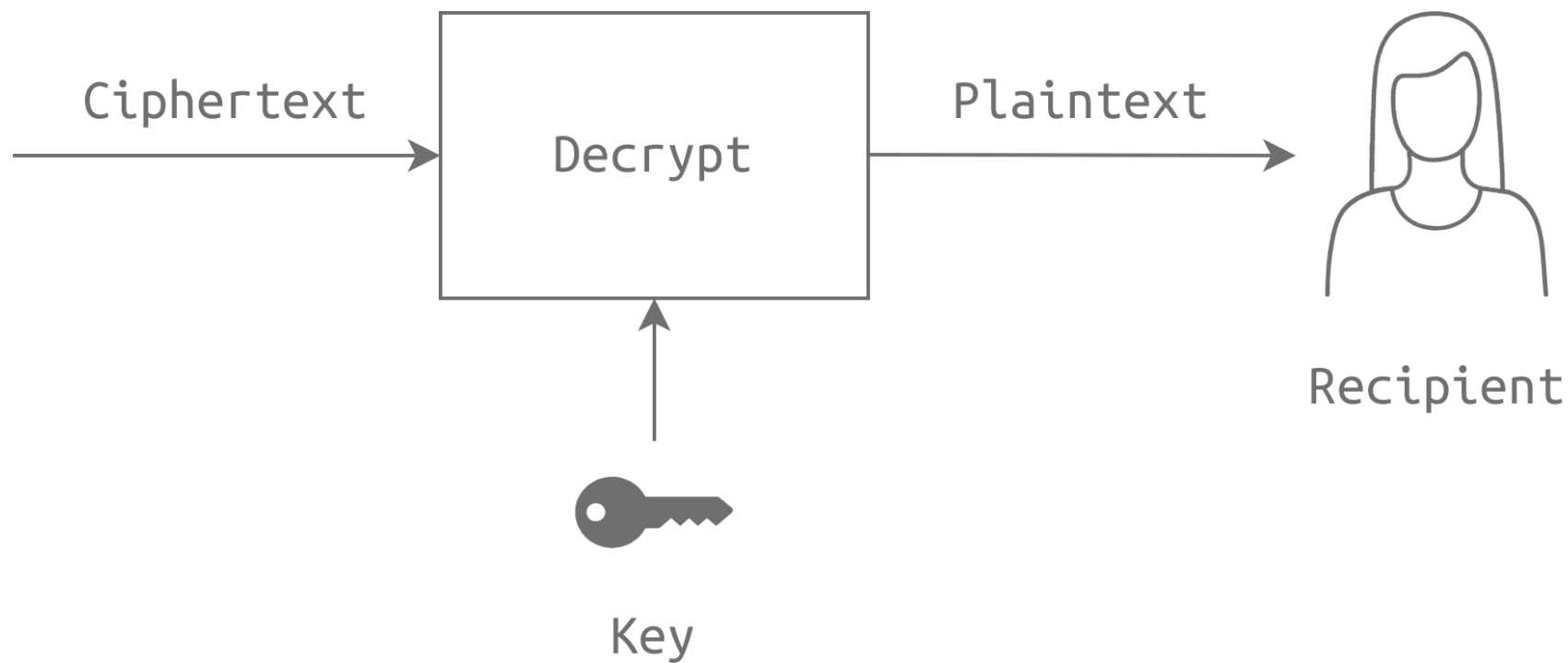
შიფრული ტექსტი ეს არის შეტყობინება მისი დაშიფრული ფორმით

სიმეტრიული დაშიფვრის ალგორითმი იყენებს იმავე გასაღებს დაშიფვრისა და გაშიფვრისთვის. შესაბამისად, კომუნიკაციის მონაწილე მხარეებმა უნდა შეთანხმდნენ საიდუმლო გასაღებზე, სანამ შეძლებენ რაიმე შეტყობინების გაცვლას.

შემდეგ ფიგურაში, გამგზავნი უზრუნველყოფს დაშიფვრის პროცესს უბრალო ტექსტით და გასაღებით შიფრული ტექსტის მისაღებად. შიფრული ტექსტი ჩვეულებრივ იგზავნება რაიმე საკომუნიკაციო არხზე.

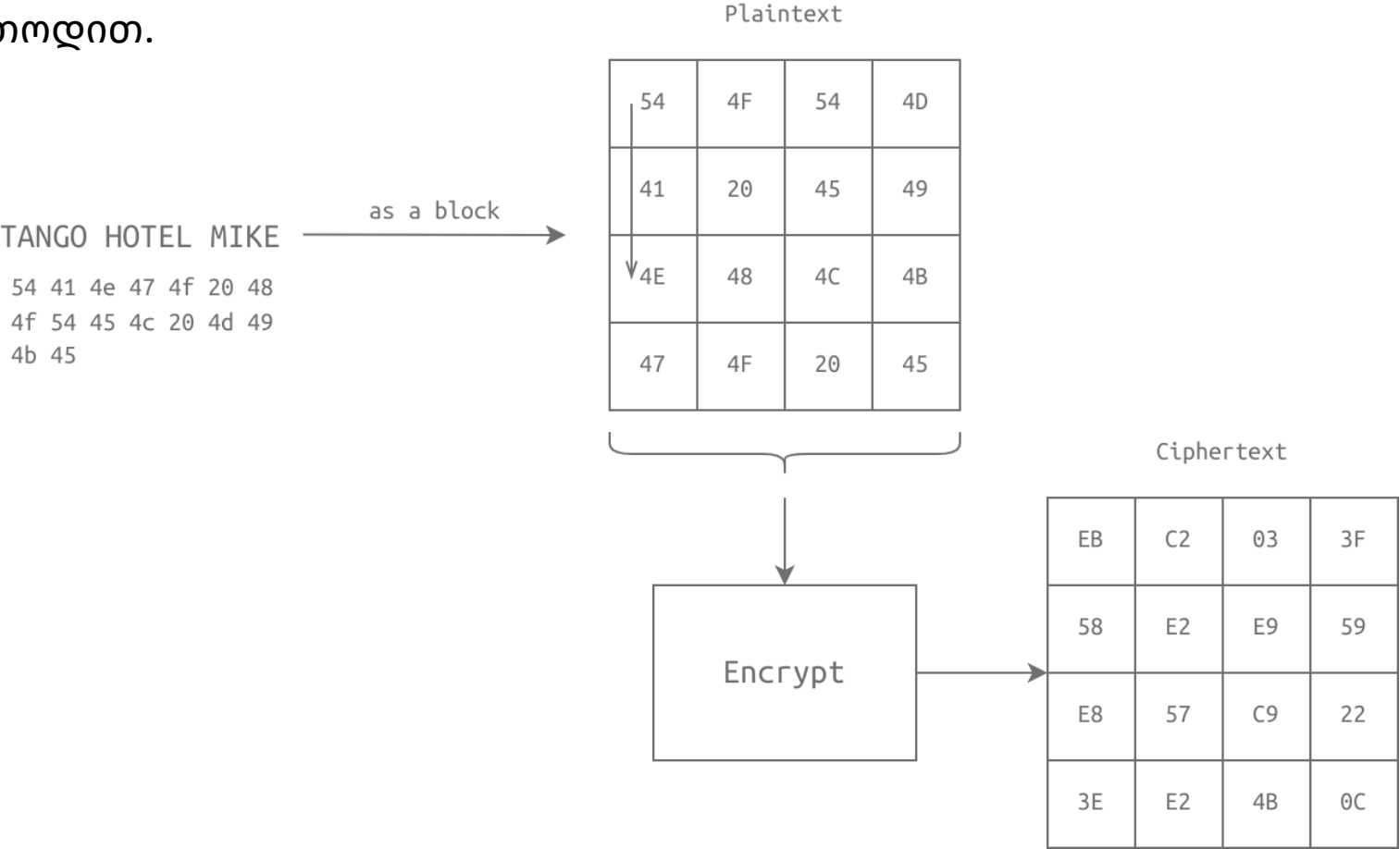


მეორეს მხრივ, მიმღები აწვდის გაშიფვრის პროცესს იმავე გასაღებით, რომელსაც გამოიყენებს გამგზავნი ორიგინალური ჩვეულებრივი ტექსტის აღსადგენად მიღებული შიფრული ტექსტიდან. გასაღების ცოდნის გარეშე, მიმღები ვერ შეძლებს ღია ტექსტის აღდგენას.



Encryption Algorithm	Notes
AES, AES192, and AES256	AES with a key size of 128, 192, and 256 bits
IDEA	International Data Encryption Algorithm (IDEA)
3DES	Triple DES (Data Encryption Standard) and is based on DES. We should note that 3DES will be deprecated in 2023 and disallowed in 2024.
CAST5	Also known as CAST-128. Some sources state that CASE stands for the names of its authors: Carlisle Adams and Stafford Tavares.
BLOWFISH	Designed by Bruce Schneier
TWOFISH	Designed by Bruce Schneier and derived from Blowfish
CAMELLIA128, CAMELLIA192, and CAMELLIA256	Designed by Mitsubishi Electric and NTT in Japan. Its name is derived from the flower camellia japonica.

აქამდე ნახსენები ყველა ალგორითმი არის ბლოკშიფრული სიმეტრიული დაშიფვრის ალგორითმები. ბლოკის შიფრის ალგორითმი გარდაქმნის შენატანს (უბრალო ტექსტს) ბლოკებად და შიფრავს თითოეულ ბლოკს. ბლოკი ჩვეულებრივ 128 ბიტიანია. ქვემოთ მოცემულ ფიგურაში გვინდა დავშიფროთ ჩვეულებრივი ტექსტი „TANGO HOTEL MIKE“, სულ 16 სიმბოლოსგან. პირველი ნაბიჯი არის მისი ორობითი წარმოდგენა. თუ ვიყენებთ ASCII-ს, "T" არის 0x54 თექვსმეტობით ფორმატში, "A" არის 0x41 და ა.შ. ყოველი ორი თექვსმეტობითი ციფრი შეადგენს 8 ბიტს და წარმოადგენს ერთ ბაიტს. 128 ბიტიანი ბლოკი პრაქტიკულად 16 ბაიტია და წარმოდგენილია 4-ზე 4 მასივში. 128-ბიტიანი ბლოკი იკვებება როგორც ერთი ერთეული დაშიფვრის მეთოდით.

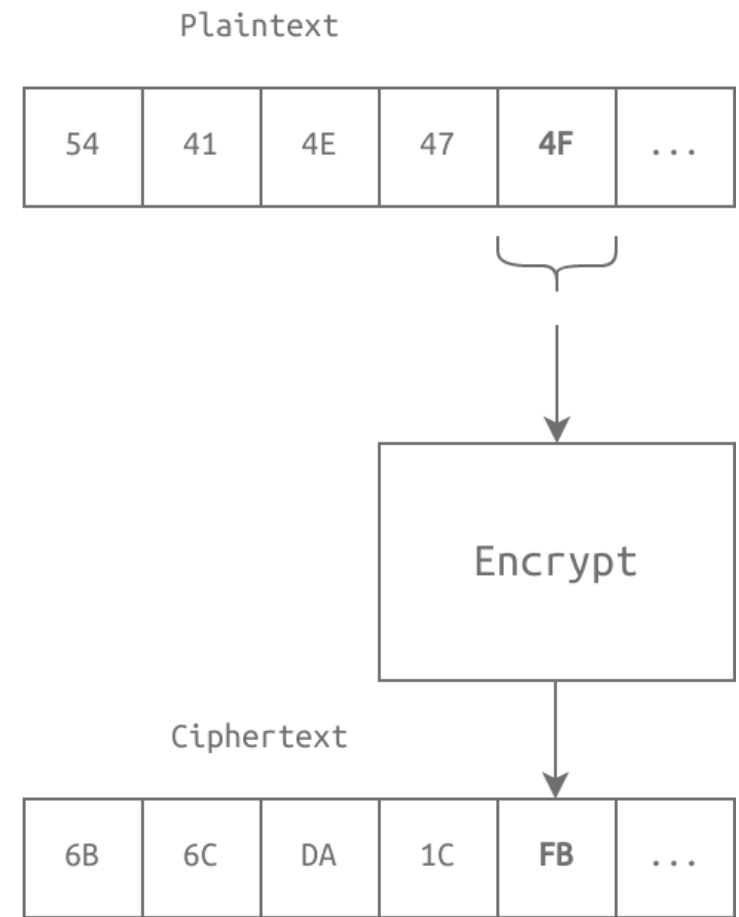




TANGO HOTEL MIKE

as a stream

54 41 4e 47 4f 20 48  
4f 54 45 4c 20 4d 49  
4b 45



სიმეტრიული დაშიფვრის ალგორითმის სხვა ტიპია ნაკადის შიფრები, რომლებიც შიფრავს უბრალო ტექსტს ბაიტ-ბაიტზე. განვიხილოთ შემთხვევა, როდესაც გვინდა დავშიფროთ შეტყობინება „TANGO HOTEL MIKE“;

თითოეული სიმბოლო უნდა გადაკეთდეს მის ორობით წარმომადგენლობაში. თუ ვიყენებთ ASCII-ს, "T" არის 0x54 თექვსმეტობით, ხოლო "A" არის 0x41 და ა.შ. დაშიფვრის მეთოდი ამუშავებს თითო ბაიტს. ეს წარმოდგენილია ქვემოთ მოცემულ ფიგურაში.



# ინფორმაციული უსაფრთხოება

ლექცია 10

tamar.kurdadze@btu.edu.ge

ასიმეტრიული გასაღების ალგორითმები;  
ციფრული ხელმოწერები, ხელმოწერა კოდზე;  
ციფრული სერტიფიკატები, ციფრული სერტიფიკატების  
დამუშავება;  
ღია გასაღების ინფრასტრუქტურა

სიმეტრიული დაშიფვრა მომხმარებლებს მოითხოვს, რომ იპოვონ უსაფრთხო არხი გასაღებების გაცვლისთვის. უსაფრთხო არხით, ჩვენ ძირითადად კონფიდენციალურობაზე და მთლიანობაზე ვზრუნავთ. სხვა სიტყვებით რომ ვთქვათ, ჩვენ გვჭირდება არხი, სადაც ვერც ერთი მესამე მხარე ვერ შეძლებს ტრაფიკის მოსმენას და წაკითხვას; მეტიც, გაგზავნილ შეტყობინებებსა და მონაცემებს ვერავინ შეცვლის.

ასიმეტრიული დაშიფვრა შესაძლებელს ხდის დაშიფრული შეტყობინებების გაცვლას უსაფრთხო არხის გარეშე; ჩვენ უბრალოდ გვჭირდება სანდო არხი. სანდო არხში ვგულისხმობთ იმას, რომ ჩვენ ძირითადად არხის მთლიანობაზე ვზრუნავთ და არა კონფიდენციალურობაზე.

ასიმეტრიული დაშიფვრის ალგორითმის გამოყენებისას, ჩვენ გამოვქმნით გასაღების წყვილს: საჯარო და კერძო გასაღებს. საჯარო გასაღები გაზიარებულია მსოფლიოსთან, უფრო კონკრეტულად, იმ ადამიანებთან, რომლებსაც სურთ ჩვენთან უსაფრთხოდ კომუნიკაცია. პირადი გასაღები უსაფრთხოდ უნდა იყოს შენახული და არავის არ უნდა მივცეთ მასზე წვდომა. უფრო მეტიც, საჯარო გასაღების ცოდნის მიუხედავად, კერძო გასაღების გამოყვანა შეუძლებელია.

როგორ მუშაობს ეს გასაღები წყვილი?

თუ შეტყობინება დაშიფრულია ერთი გასაღებით, მისი გაშიფვრა შესაძლებელია მეორე გასაღებით. სხვა სიტყვებით:

თუ ალისა დაშიფვრავს შეტყობინებას ბობის საჯარო გასაღების გამოყენებით, მისი გაშიფვრა შესაძლებელია მხოლოდ ბობის პირადი გასაღების გამოყენებით.

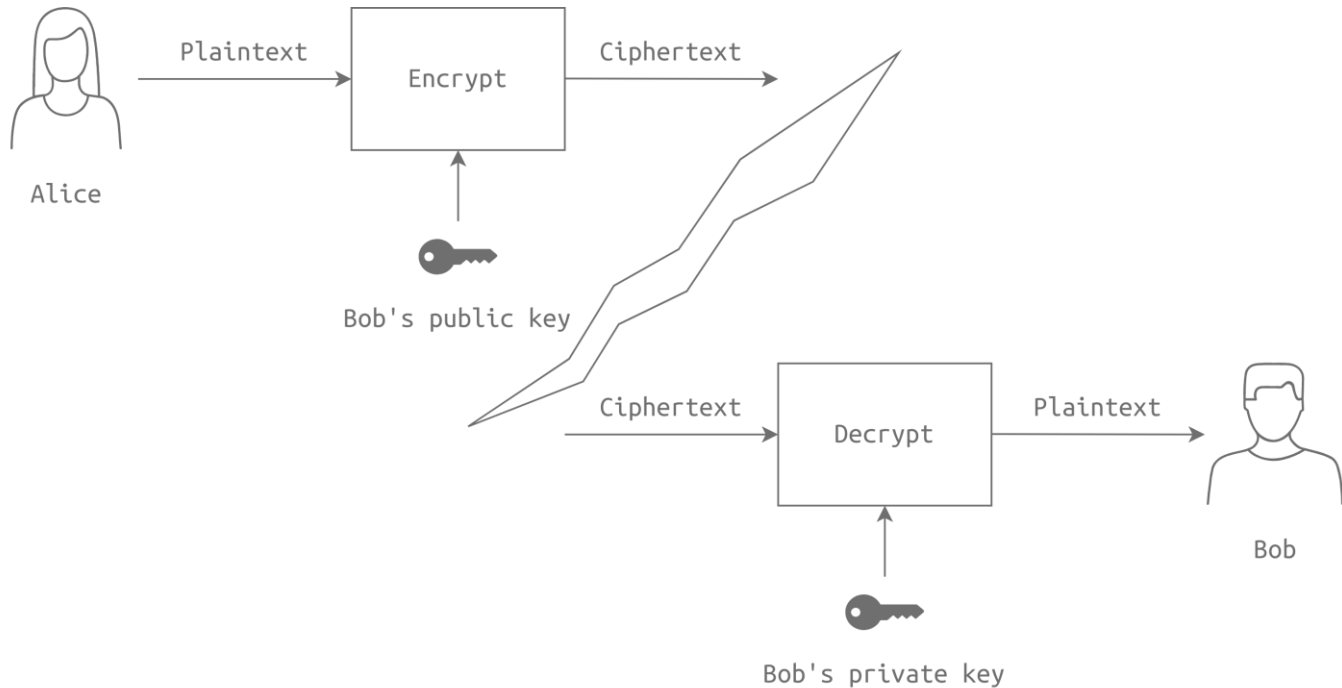
პირიქით, თუ ბობი დაშიფვრავს შეტყობინებას მისი პირადი გასაღების გამოყენებით, მისი გაშიფვრა შესაძლებელია მხოლოდ ბობის საჯარო გასაღების გამოყენებით.

## კონფიდენციალურობა

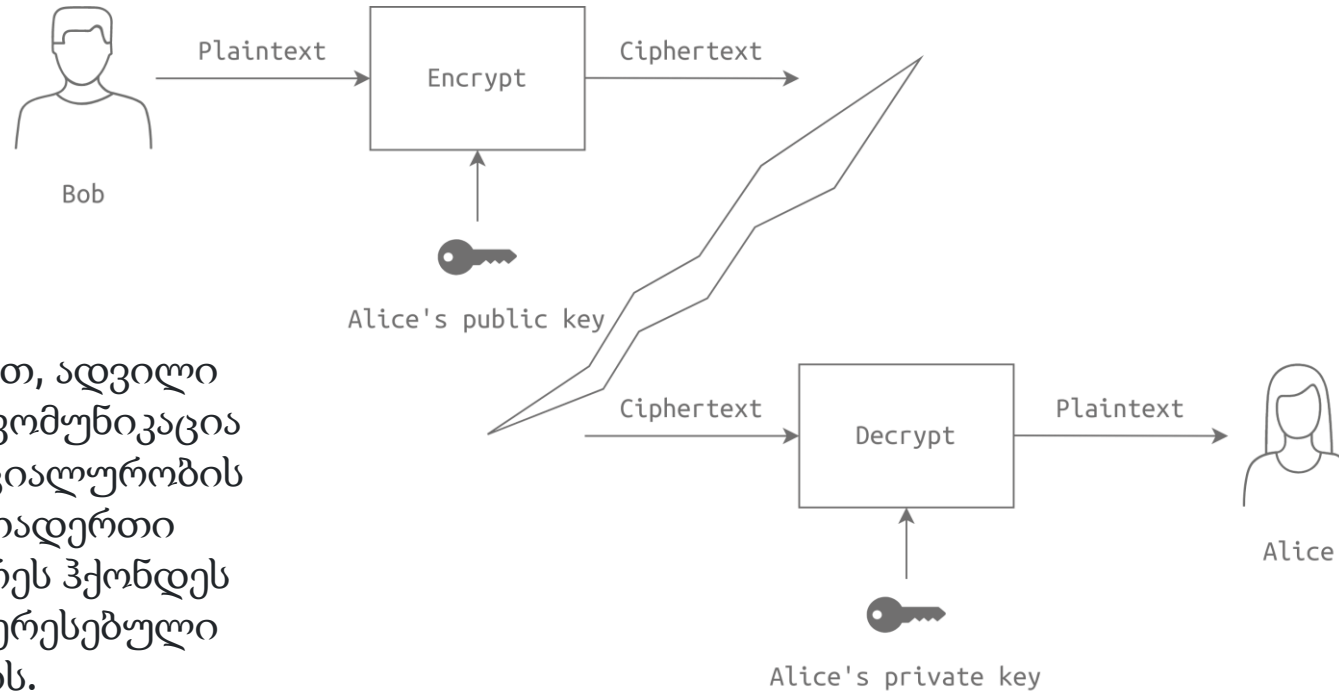
ჩვენ შეგვიძლია გამოვიყენოთ ასიმეტრიული დაშიფვრა კონფიდენციალურობის მისაღწევად შეტყობინებების დაშიფვრით მიმღების საჯარო გასაღების გამოყენებით.

შემდეგ ორ ფიგურაში ჩვენ ვხედავთ, რომ:

ალის სურს უზრუნველყოს კონფიდენციალურობა ბობთან კომუნიკაციას. ის შიფრავს შეტყობინებას ბობის საჯარო გასაღების გამოყენებით, ხოლო ბობი შიფრავს მათ მისი პირადი გასაღების გამოყენებით. მოსალოდნელია, რომ ბობის საჯარო გასაღები გამოქვეყნდება საჯარო მონაცემთა ბაზაში ან მის ვებსაიტზე, მაგალითად.

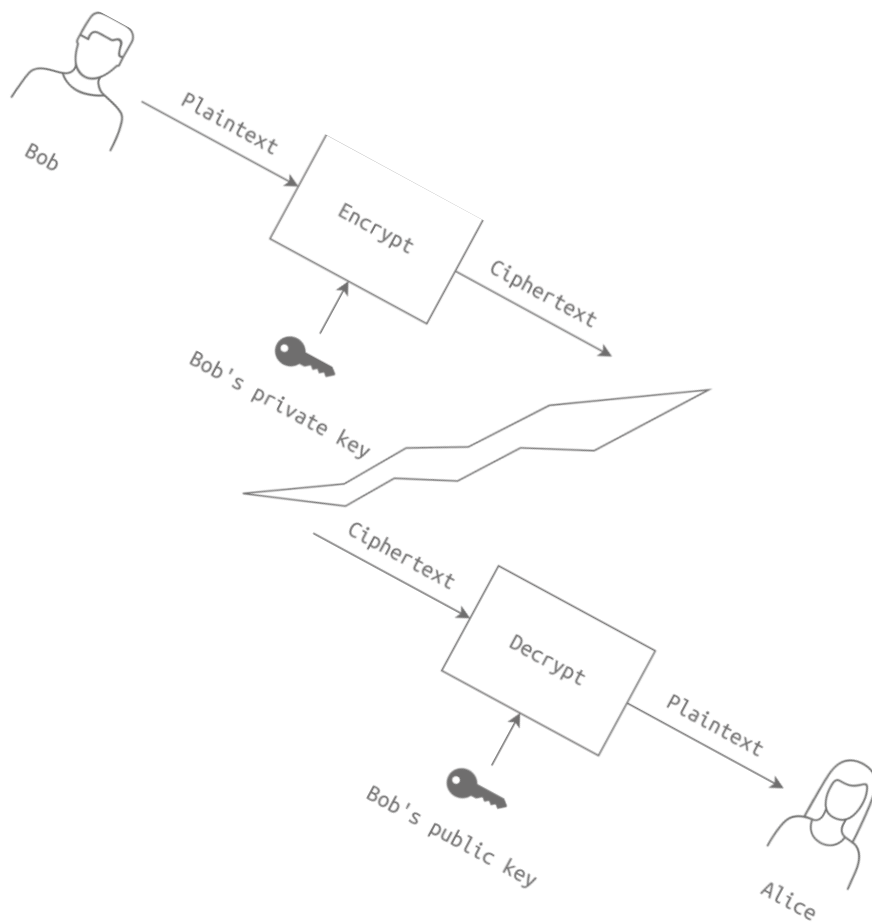


როდესაც ბობს სურს უპასუხოს ალისას, ის შიფრავს მის შეტყობინებებს ალისის საჯარო გასაღების გამოყენებით და ალისს შეუძლია მათი გაშიფვრა მისი პირადი გასაღების გამოყენებით.



სხვა სიტყვებით რომ ვთქვათ, ადვილი ხდება ალისთან და ბობთან კომუნიკაცია შეტყობინებების კონფიდენციალურობის უზრუნველყოფისას. ერთადერთი მოთხოვნაა, რომ ყველა მხარეს ჰქონდეს საჯარო გასაღებები დაინტერესებული გამომგზავნისთვის.

შენიშვნა: პრაქტიკაში, სიმეტრიული დაშიფვრის ალგორითმები იძლევა უფრო სწრაფ ოპერაციებს, ვიდრე ასიმეტრიული დაშიფვრა; ამიტომ, ჩვენ მოგვიანებით განვიხილავთ, თუ როგორ შეგვიძლია გამოვიყენოთ საუკეთესო ორივე სამყაროდან.



### მთლიანობა, ავთენტურობა და არაუარყოფა

კონფიდენციალურობის მიღმა, ასიმეტრიულ დაშიფვრას შეუძლია გადაჭრას მთლიანობა, ავთენტურობა და არაუარყოფა. ვთქვათ, რომ ბობს სურს განცხადების გაკეთება და სურს, რომ ყველამ შეძლოს დაადასტუროს, რომ ეს განცხადება მართლაც მისგან მოვიდა. ბობს სჭირდება შეტყობინების დაშიფვრა მისი პირადი გასაღების გამოყენებით; მიმღებებს შეუძლიათ მისი გაშიფვრა ბობის საჯარო გასაღების გამოყენებით. თუ შეტყობინება წარმატებით გაშიფრულია ბობის საჯარო გასაღებით, ეს ნიშნავს, რომ შეტყობინება დაშიფრულია ბობის პირადი გასაღების გამოყენებით. (პრაქტიკაში, ის დაშიფვრავს თავდაპირველი შეტყობინების ჰეშს. ამას მოგვიანებით განვიხილავთ.)

ბობის საჯარო გასაღების წარმატებით გაშიფვრას რამდენიმე საინტერესო დასკვნამდე მივყავართ.

პირველი, მესიჯი არ შეცვლილა მთელს გზაზე (საკომუნიკაციო არხი); ეს ადასტურებს მესიჯის მთლიანობას.

მეორე, იმის ცოდნა, რომ არავის აქვს წვდომა ბობის პირად გასაღებზე, შეგვიძლია დარწმუნებული ვიყოთ, რომ ეს შეტყობინება მართლაც ბობისგან მოვიდა; ეს ადასტურებს შეტყობინების ნამდვილობას.

დაბოლოს, იმის გამო, რომ ბობის გარდა არავის აქვს წვდომა ბობის პირად გასაღებზე, ბობ არ შეუძლია უარი თქვას ამ შეტყობინების გაგზავნაზე; ეს ადგენს არაუარყოფას.

## RSA

RSA მიიღო თავისი სახელი მისი გამომგონებლების, Rivest-ის, Shamir-ისა და Adleman-ისგან. იგი მუშაობს შემდეგნაირად:

1. აირჩიეთ ორი შემთხვევითი მარტივი რიცხვი,  $p$  და  $q$ . გამოთვალეთ  $N = p \times q$ .
2. აირჩიეთ ორი მთელი რიცხვი  $e$  და  $d$  ისე, რომ  $e \times d = 1 \bmod \phi(N)$ , სადაც  $\phi(N) = N - p - q + 1$ . ეს ნაბიჯი საშუალებას მოგვცემს შევქმნათ საჯარო გასაღები ( $N$ ) გასაღები ( $N, d$ ).
3. გამომგზავნს შეუძლია დაშიფვროს მნიშვნელობა  $x$  გაანგარიშებით  $y = x^e \bmod N$ . (მოდული)
4. მიმღებს შეუძლია გაშიფვროს  $y$   $x = y^d \bmod N$ -ის გამოთვლით. გაითვალისწინეთ, რომ  $y^d = x^{ed} = x^{k\phi(N) + 1} = (x^{\phi(N)})^k \times x = x$ .

ეს ნაბიჯი განმარტავს, თუ რატომ ვაყენებთ შეზღუდვას  $e$  და  $d$ -ის არჩევანზე.

არ ინერვიულოთ, თუ ზემოთ მოყვანილი მათემატიკური განტოლებები ძალიან რთული ჩანდა; თქვენ არ გჭირდებათ მათემატიკა, რომ შეძლოთ RSA-ს გამოყენება, რადგან ის ადვილად ხელმისაწვდომია პროგრამებისა და პროგრამირების ბიბლიოთეკების საშუალებით.

RSA უსაფრთხოება ეყრდნობა ფაქტორიზაციას, რომელიც რთული პრობლემაა. მარტივია  $p$   $q$ -ზე გამრავლება; თუმცა, შრომატევადია  $p$  და  $q$  მოცემული  $N$ -ის პოვნა. უფრო მეტიც, ეს რომ იყოს უსაფრთხო,  $p$  და  $q$  საკმაოდ დიდი რიცხვები უნდა იყოს, მაგალითად, თითოეული 1024 ბიტიანი (ეს არის რიცხვი 300 ციფრზე მეტი). მნიშვნელოვანია აღინიშნოს, რომ RSA ეყრდნობა უსაფრთხო შემთხვევითი რიცხვების გენერირებას, ისევე როგორც სხვა ასიმეტრიული დაშიფვრის ალგორითმებს. თუ მოწინააღმდეგეს შეუძლია გამოიკნოს  $p$  და  $q$ , მთელი სისტემა ჩაითვლება დაუცველად.

განვიხილოთ შემდეგი პრაქტიკული მაგალითი.

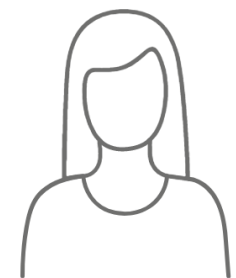
1. ბობი ირჩევს ორ მარტივ რიცხვს:  $p = 157$  და  $q = 199$ . ის ითვლის  $N = 31243$ .
2.  $\phi(N) = N - p - q + 1 = 31243 - 157 - 199 + 1 = 30888$ , Bob selects  $e = 163$ , Bob selects  $d = 379$ .
3. აქ  $e \times d = 163 \times 379 = 61777$  და  $61777 \bmod 30888 = 1$ . საჯარო გასაღები არის (31243, 163) და პირადი გასაღები (31243, 379).
3. ვთქვათ, რომ დაშიფვრის მნიშვნელობა არის  $x = 13$ , შემდეგ ალისა გამოთვლის და გამოგიგზავნის  $y = x^e \bmod N = 13^{163} \bmod 31243 = 16342$ .
4. ბობი გაშიფვრავს მიღებულ მნიშვნელობას  $x = y^d \bmod N = 16342^{379} \bmod 31243 = 13$  გამოთვლით.

გასაღების გაცვლის გამოყენება, როგორც Diffie-Hellman გასაღების გაცვლა, საშუალებას გვაძლევს შევთანხმდეთ საიდუმლო გასაღებზე მომსმენების თვალისა და ყურების ქვეშ. ეს გასაღები შეიძლება გამოყენებულ იქნას სიმეტრიული დაშიფვრის ალგორითმით კონფიდენციალური კომუნიკაციის უზრუნველსაყოფად. თუმცა, გასაღების გაცვლა, რომელიც ადრე აღვწერეთ, არ არის იმუნური ადამიანი-შუაში (MITM) თავდასხმისგან. მიზეზი ის არის, რომ ალისს არ აქვს საშუალება უზრუნველყოს, რომ იგი დაუკავშირდა ბობს, და ბობს არ აქვს საშუალება უზრუნველყოს, რომ ის დაუკავშირდა ალისს საიდუმლო გასაღების გაცვლისას.

განვიხილოთ ქვემოთ მოყვანილი ფიგურა. ეს არის თავდასხმა გასაღების გაცვლის წინააღმდეგ, რომელიც ახსნილია Diffie-Hellman Key Exchange ამოცანაში. ნაბიჯები შემდეგია:

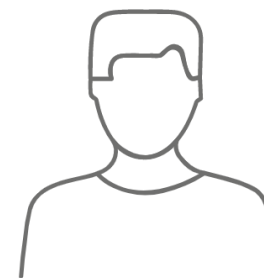
1. ალისა და ბობი თანხმდებიან  $q$  და  $g$ -ზე. ნებისმიერს, ვინც უსმენს საკომუნიკაციო არხს, შეუძლია წაიკითხოს ეს ორი მნიშვნელობა, მათ შორის თავდამსხმელი, მელორი.
  2. როგორც ამას ჩვეულებრივ აკეთებდა, ალისა ირჩევს შემთხვევით ცვლადს  $a$ , ითვლის  $A$  ( $A = (g^a) \bmod q$ ) და აგზავნის  $A$  ბობს. მელორი ელოდა ამ ნაბიჯს და მან აირჩია შემთხვევითი ცვლადი  $m$  და გამოითვალა შესაბამისი  $M$ . როგორც კი მელორი მიიღებს  $A$ -ს, ის აგზავნის  $M$  ბობს, თითქოს ის არის ალისა.
  3. ბობი იღებს  $M$ -ს, როცა ფიქრობს, რომ ის ალისამ გამოგზავნა. ბობმა უკვე შეარჩია შემთხვევითი ცვლადი  $b$  და გამოითვალა შესაბამისი  $B$ ; ის აგზავნის  $B$  ალისას. ანალოგიურად, მელორი წყვეტს შეტყობინებას, კითხულობს  $B$  და აგზავნის  $M$  ალისას.
  4. ალისა იღებს  $M$  და ითვლის გასაღებს  $= Ma \bmod q$ .
  5. ბობი იღებს  $M$  და ითვლის კლავიშს  $= Mb \bmod q$ .
- ალისა და ბობი აგრძელებენ კომუნიკაციას, ფიქრობენ, რომ ისინი უშუალოდ ურთიერთობენ, არ იციან, რომ ურთიერთობენ მელორისთან, რომელსაც შეუძლია წაიკითხოს და შეცვალოს შეტყობინებები, სანამ მათ მიმღებს გაუგზავნის.





Alice

Chooses  $a$   
Sends  $A$



Bob

Agree on  $q$  and  $g$



Mallory

Chooses  $m$   
Sends  $M$  to Bob  
Sends  $M$  to Alice

$M$

$key = M^a \bmod q$

$M$

Chooses  $b$   
Sends  $B$

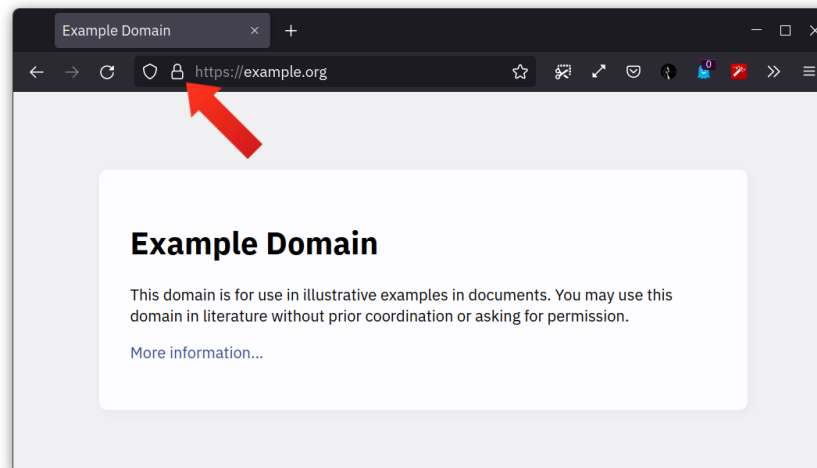
$B$

$key = M^b \bmod q$

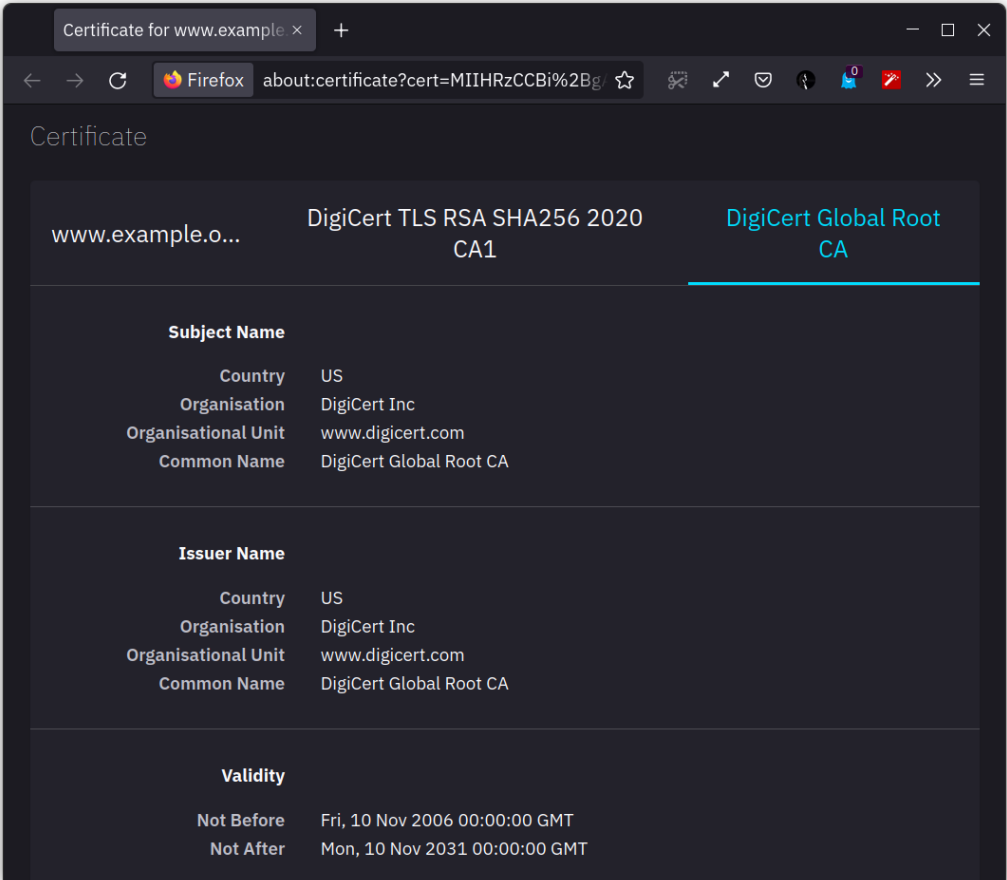
ეს მგრძნობელობა მოითხოვს რაიმე მექანიზმს, რომელიც საშუალებას მოგვცემს დავამტკიცოთ მეორე მხარის ვინაობა. ეს მიგვიყვანს საჯარო გასაღების ინფრასტრუქტურამდე (PKI).

განვიხილოთ შემთხვევა, როდესაც თქვენ ათვალიერებთ ვებსაიტს example.org HTTPS-ით. როგორ შეგიძლიათ იცოთ დარწმუნებული, რომ ნამდვილად გაქვთ კომუნიკაცია example.org სერვერ(ებ)თან? სხვა სიტყვებით რომ ვთქვათ, როგორ შეგიძლიათ დარწმუნებული იყოთ, რომ არცერთმა შუამავალმა არ ჩაჭრა პაკეტები და შეცვალა ისინი თქვენამდე მისვლამდე? პასუხი დევს ვებსაიტის სერთიფიკატში.

ქვემოთ მოყვანილი ფიგურა აჩვენებს გვერდს, რომელსაც ვიღებთ example.org-ის დათვალიერებისას. ბრაუზერების უმეტესობა წარმოადგენს დამიფრულ კავშირს რაიმე სახის საკეტის ხატულასთან. ეს დაბლოკვის ხატულა მიუთითებს, რომ კავშირი დაცულია HTTPS-ზე მოქმედი სერთიფიკატით.



წერის დროს, example.org იყენებს DigiCert Inc.-ის მიერ ხელმოწერილ სერტიფიკატს, როგორც ეს ნაჩვენებია ქვემოთ მოცემულ ფიგურაში. სხვა სიტყვებით რომ ვთქვათ, DigiCert ადასტურებს, რომ ეს სერტიფიკატი მოქმედებს (გარკვეულ თარიღამდე).

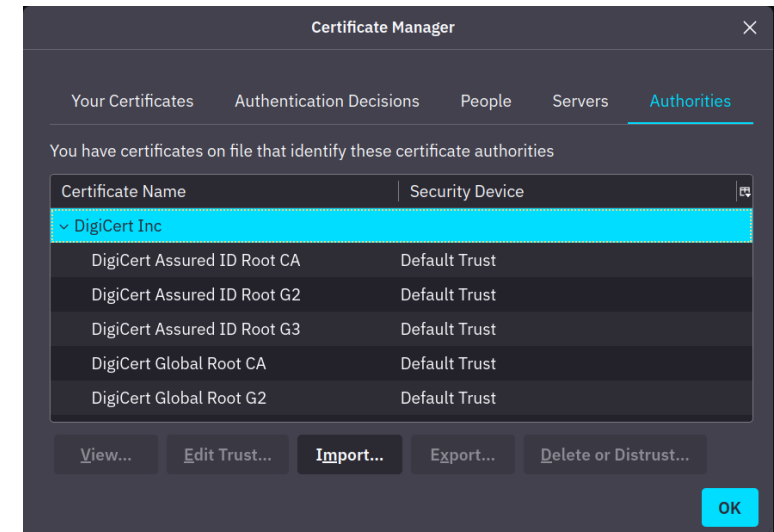


სერტიფიკატის ხელმოწერისთვის სერტიფიკატის ორგანოს მიერ, ჩვენ გვჭირდება:

1) სერტიფიკატის ხელმოწერის მოთხოვნის გენერირება (CSR): თქვენ ქმნით სერტიფიკატს და ავზავნით თქვენს საჯარო გასაღებს მესამე მხარის მიერ ხელმოწერისთვის.

2) გაუგზავნეთ თქვენი CSR სერტიფიკატების ორგანოს (CA): მიზანია, რომ CA მოაწეროს ხელი თქვენს სერტიფიკატს. ალტერნატიული და, როგორც წესი, არასაიმედო გამოსავალი იქნება თქვენი სერტიფიკატის ხელმოწერა.

იმისათვის, რომ ეს იმუშაოს, მიმღებმა უნდა აღიაროს და ენდოს CA, რომელმაც ხელი მოაწერა სერტიფიკატს. და როგორც ჩვენ მოველით, ჩვენი ბრაუზერი ენდობა DigiCert Inc-ს, როგორც ხელმომწერ ორგანოს; წინააღმდეგ შემთხვევაში, ის გამოსცემდა უსაფრთხოების გაფრთხილებას მოთხოვნილ ვებსაიტზე გადასვლის ნაცვლად.



# ინფორმაციული უსაფრთხოება

ლექცია 10

tamar.kurdadze@btu.edu.ge

უსაფრთხოების მონიტორინგი; მონაცემთა ანალიზი შემოჭრების შესახებ.

განსახილველი საკითხები:

გავრცელებული პროტოკოლების მონიტორინგი: Syslog და NTP, DNS, HTTP და HTTPS, ICMP, საფოსტო პროტოკოლები;

ლოგ ფაილები: უსაფრთხოების მონაცემთა ტიპები, საბოლოო მოწყობილობის ლოგები, ქსელური ლოგები;

გამაფრთხილებელი შეტყობინებების შეფასება;

ქსელური უსაფრთხოების მონაცემებთან მუშაობა;

ELSA და SGUIL პლატფორმების განხილვა.

საერთო პროტოკოლების მონიტორინგი აუცილებელია ქსელის უსაფრთხოების, მუშაობისა და საიმედოობის შესანარჩუნებლად. აქ მოცემულია მონიტორინგის მოსაზრებების მოკლე მიმოხილვა ზოგიერთი გავრცელებული პროტოკოლისთვის:

Syslog:

- მონიტორინგის წერტილები: შეაგროვეთ სისტემური შეტყობინებები ქსელის მოწყობილობებიდან, სერვერებიდან და აპლიკაციებიდან.
- ინსტრუმენტები: Syslog სერვერები, როგორიცაა syslog-ng, rsyslog ან Splunk, შეიძლება გამოყენებულ იქნას syslog მონაცემების შესაგროვებლად, შესანახად და ანალიზისთვის.
- მეტრიკა მონიტორინგისთვის: შეტყობინებების მოცულობა, სიმძიმის დონეები, წყაროს IP-ები და შეტყობინებების კონკრეტული ნიმუშები.

NTP (ქსელის დროის პროტოკოლი):

- მონიტორინგის წერტილები: NTP სერვერებსა და კლიენტებს შორის დროის სინქრონიზაციის მონიტორინგი.
- ინსტრუმენტები: NTP მონიტორინგის ხელსაწყოები, როგორიცაა ntpq ან სპეციალური NTP მონიტორინგის გადაწყვეტილებები.
- მეტრიკა მონიტორინგისთვის: დროის ოფსეტური, ჯიტერი, ხელმისაწვდომობა და ფენის დონეები.

DNS (დომენის სახელების სისტემა):

- მონიტორინგის წერტილები: DNS-ის რეზოლუციის დროისა და DNS მოთხოვნის მოცულობის მონიტორინგი.
- ინსტრუმენტები: DNS მონიტორინგის ხელსაწყოები, როგორიცაა dnstop, dnsmasq ან სპეციალური DNS მონიტორინგის გადაწყვეტილებები.
- მეტრიკა მონიტორინგისთვის: შეკითხვის პასუხების დრო, შეკითხვის წარმატების მაჩვენებელი და ავტორიტეტული სერვერის პასუხისმგებლობა.

1.HTTPS და HTTPS:

2.მონიტორინგის წერტილები: ვებ სერვერების მონიტორინგი, ბალანსის ჩატვირთვა და მარიონეტები.

3.ინსტრუმენტები: ვებ სერვერის ჟურნალები, აპლიკაციის შესრულების მონიტორინგის (APM) ინსტრუმენტები, ან HTTP/HTTPS-სპეციფიკური მონიტორინგის ინსტრუმენტები.

4.მეტრიკა მონიტორინგის: რეაგირების დრო, შეცდომის სიჩქარე, HTTP სტატუსის კოდები და სერვერის რესურსების გამოყენება.

5.ICMP (ინტერნეტ კონტროლის შეტყობინების პროტოკოლი):

6.მონიტორინგის წერტილები: ქსელის კავშირისა და შეყოვნების მონიტორინგი.

7.ინსტრუმენტები: Ping, traceroute ან ქსელის მონიტორინგის ინსტრუმენტები, როგორცაა Nagios, Zabbix ან PRTG.

8.მეტრიკა მონიტორინგის: ორმხრივი მოგზაურობის დრო, პაკეტის დაკარგვა და მარშრუტის ცვლილებები.

9.ფოსტის პროტოკოლები (მაგ., SMTP, POP3, IMAP):

10.მონიტორინგის წერტილები: ფოსტის სერვერების და მასთან დაკავშირებული პროტოკოლების მონიტორინგი.

11.ინსტრუმენტები: ფოსტის სერვერის ჟურნალები, ქსელის მონიტორინგის ინსტრუმენტები ან ელ.ფოსტის მონიტორინგის სპეციალიზებული გადაწყვეტილებები.

12.მეტრიკა მონიტორინგის: ელ.ფოსტის მიწოდების დრო, სერვერზე რეაგირების დრო და ფოსტის რიგის სიგრძე.

გადამწყვეტი მნიშვნელობა აქვს გაფრთხილებების დაყენებას ამ მეტრიკების წინასწარ განსაზღვრულ ზღვრებზე დაყრდნობით, პრობლემების პროაქტიულად იდენტიფიცირებისა და გადაწყვეტის მიზნით. გარდა ამისა, განიხილეთ ცენტრალიზებული მონიტორინგის გადაწყვეტილებების გამოყენება, რომლებსაც შეუძლიათ სხვადასხვა წყაროდან მონაცემების გაერთიანება და ქსელის სიჯანსაღის ჰოლისტიკური ხედვა. რეგულარულად გადახედეთ და განაახლეთ მონიტორინგის კონფიგურაციები ქსელის ინფრასტრუქტურისა და აპლიკაციის გარემოში ცვლილებებთან ადაპტაციისთვის.I



ჟურნალის ფაილები(log files) ინფორმაციის გადამწყვეტი წყაროა IT გარემოს უსაფრთხოებისა და მუშაობის მონიტორინგისა და ანალიზისთვის. სხვადასხვა ტიპის ჟურნალები გვაწვდიან ინფორმაციას სისტემისა და ქსელის საქმიანობის სხვადასხვა ასპექტზე. აქ არის უსაფრთხოების მონაცემთა რამდენიმე ტიპი, რომლებიც ნაპოვნია ჟურნალის ფაილებში, ბოლო მოწყობილობის ჟურნალისა და ქსელის ჟურნალის მაგალითებთან ერთად:

**ბოლო მოწყობილობის ჟურნალები:**

**სისტემის ჟურნალები:**

აღწერა: ზოგადი სისტემის მოვლენები, შეცდომები და გაფრთხილებები.

მაგალითები: Windows Event Logs (სისტემა), / var/log/syslog (Linux).

**განაცხადის ჟურნალები:**

აღწერა: ჩანაწერები მოვლენები, რომლებიც სპეციფიკურია მოწყობილობაზე გაშვებული პროგრამებისთვის.

მაგალითები: აპლიკაციის სპეციფიკური ჟურნალის ფაილები.

**მოწყობილობის მართვის ჟურნალები:**

აღწერა: იღებს მოვლენებს, რომლებიც დაკავშირებულია მოწყობილობის კონფიგურაციის ცვლილებებთან და მენეჯმენტის საქმიანობასთან.

მაგალითები: როუტერი ან გადართვის ჟურნალები.

**მომხმარებლის აქტივობის ჟურნალები:**

აღწერა: აღრიცხავს მომხმარებლის მოქმედებებს მოწყობილობაზე.

მაგალითები: Bash history (Linux), PowerShell ჟურნალები (Windows).

უსაფრთხოების მონაცემთა ტიპები:

1)ავთენტიფიკაციის ჟურნალები(log):

აღწერა: აღრიცხავს ინფორმაციას მომხმარებლის შესვლისა და ავტორიზაციის მცდელობების შესახებ.

მაგალითები: Windows Event Logs (უსაფრთხოება), /var/log/auth.log (Linux).

2)ავტორიზაციის ჟურნალები(log):

აღწერა: თვალყურს ადევნებს მომხმარებლის ნებართვებს და წვდომის კონტროლის გადაწყვეტილებებს.

მაგალითები: Windows Event Logs (უსაფრთხოება), /var/log/auth.log (Linux).

3)აუდიტის ჟურნალები:

აღწერა: იჭერს სისტემის და აპლიკაციის აქტივობების დეტალურ ჩანაწერებს უსაფრთხოების ანალიზისთვის.

მაგალითები: Windows უსაფრთხოების ჟურნალი, /var/log/audit/audit.log (Linux).

4)შეჭრის აღმოჩენისა და პრევენციის სისტემის (IDPS) ჟურნალები:

აღწერა: აღრიცხავს სიგნალებს და მოვლენებს, რომლებიც დაკავშირებულია უსაფრთხოების პოტენციურ საფრთხეებთან.

მაგალითები: Snort ჟურნალები, Suricata ჟურნალები.

5)Firewall ჟურნალები:

აღწერა: აღრიცხავს ინფორმაციას Firewall-ის მიერ დაბლოკილი ან დაშვებული ტრაფიკის შესახებ.

მაგალითები: iptables ჟურნალები (Linux), Windows Firewall ჟურნალები.

6)ანტივირუსული და მავნე პროგრამების ჟურნალები:

აღწერა: გვაწვდის ინფორმაციას აღმოჩენილი ან დაბლოკილი მავნე აქტივობების შესახებ.

მაგალითები: ჩანაწერები ანტივირუსული პროგრამული უზრუნველყოფიდან.

7)ბოლო წერტილის უსაფრთხოების ჟურნალები:

აღწერა: იჭერს მოვლენებს, რომლებიც დაკავშირებულია ბოლო წერტილის უსაფრთხოების გადაწყვეტილებებთან.

მაგალითები: ჟურნალები ბოლო წერტილის დაცვის პლატფორმებიდან.

ქსელის ჟურნალები:

1) ქსელის მოწყობილობების ჟურნალები:

აღწერა: იჭერს მოვლენებსა და შეცდომებს ქსელის მოწყობილობებზე (როუტერები, გადამრთველები და ა.შ.).

მაგალითები: როუტერის და გადართვის ჟურნალები.

2) Firewall ჟურნალები:

აღწერა: აღრიცხავს ინფორმაციას firewall-ის მიერ დაშვებული ან უარყოფილი ტრაფიკის შესახებ.

მაგალითები: iptables ჟურნალები (Linux), Windows Firewall ჟურნალები.

3) პროქსის ჟურნალები (Proxy Logs):

აღწერა: აღწერს დეტალებს ვებ ტრაფიკის, მომხმარებლის აქტივობისა და შინაარსის ფილტრაციის შესახებ.

მაგალითები: Squid proxy ჟურნალები.

4) DNS ჟურნალები:

აღწერა: იღებს DNS მოთხოვნისა და პასუხების ინფორმაციას.

მაგალითები: ჟურნალების შეკვრა.

5) VPN ჟურნალები:

აღწერა: ვირტუალური პირადი ქსელის (VPN) კავშირებთან დაკავშირებული ჟურნალები.

მაგალითები: VPN სერვერის ჟურნალი.

ჟურნალის ეფექტური მენეჯმენტი მოიცავს ჟურნალების შეგროვებას, შენახვას და ანალიზს ცენტრალიზებულად ისეთი ინსტრუმენტების გამოყენებით, როგორიცაა SIEM (უსაფრთხოების ინფორმაციისა და მოვლენების მენეჯმენტი) სისტემები. ჟურნალების რეგულარულად გადახედვა გეხმარებათ უსაფრთხოების ინციდენტების იდენტიფიცირებასა და რეაგირებაზე, პრობლემების მოგვარებაში და უსაფრთხოების პოლიტიკასთან შესაბამისობის უზრუნველყოფას.

გამაფრთხილებელი შეტყობინებების შეფასება სისტემის მონიტორინგისა და უსაფრთხოების მართვის მნიშვნელოვანი ასპექტია. გამაფრთხილებელი შეტყობინებები, რომლებიც ხშირად გენერირებულია სხვადასხვა პროგრამული აპლიკაციების, ოპერაციული სისტემებისა და უსაფრთხოების ინსტრუმენტების მიერ, იძლევა მითითებებს პოტენციურ საკითხებზე ან მოვლენებზე, რომლებიც შეიძლება საჭიროებდეს ყურადღებას. აქ არის ძირითადი მოსაზრებები გამაფრთხილებელი შეტყობინებების შესაფასებლად:

კონტექსტური გაგება Contextual Understanding :

შეაფასეთ კონტექსტი Assess the Context : გაიგეთ კონტექსტი, რომელშიც შეიქმნა გამაფრთხილებელი შეტყობინება. განვიხილოთ აპლიკაცია, სისტემის კომპონენტი ან უსაფრთხოების ინსტრუმენტი, რომელიც ქმნის გაფრთხილებას და მის როლს საერთო გარემოში.

სიმძიმე და გავლენა **Severity and Impact** :

სიმძიმის განსაზღვრა: შეაფასეთ გაფრთხილებისთვის მინიჭებული სიმძიმის დონე. უფრო მაღალი სიმძიმის დონე ზოგადად უფრო კრიტიკულ საკითხებზე მიუთითებს.

შეაფასეთ გავლენა: გაითვალისწინეთ გაფრთხილების პოტენციური გავლენა სისტემის მუშაობაზე, უსაფრთხოებაზე ან მომხმარებლის გამოცდილებაზე.

კორელაცია სხვა მოვლენებთან:

მოვლენების კორელაცია: გამაფრთხილებელი შეტყობინების გაანალიზება ჟურნალის სხვა ჩანაწერებთან ან მოვლენებთან ერთად. მოძებნეთ შაბლონები ან კორელაციები, რომლებსაც შეუძლიათ სიტუაციის უფრო ყოვლისმომცველი გაგება.

საბაზისო შედარება:

შედარება საწყისთან: შეადარეთ გამაფრთხილებელი შეტყობინება საბაზისო ან ნორმალური სისტემის ქცევის წინააღმდეგ. საბაზისოდან გადახრები შეიძლება მიუთითებდეს ანომალიებზე ან უსაფრთხოების პოტენციურ ინციდენტებზე.

წყაროს რეპუტაცია:

შეაფასეთ წყაროს რეპუტაცია: გაითვალისწინეთ გამაფრთხილებელი შეტყობინების წარმომქმნელი წყაროს რეპუტაცია და სანდოობა. სანდო უსაფრთხოების ინსტრუმენტები და კარგად დამკვიდრებული აპლიკაციები ზოგადად უფრო საიმედოა.

მომხმარებლის და სისტემის კონტექსტი:

განიხილეთ მომხმარებლის და სისტემის კონტექსტი: გაითვალისწინეთ მომხმარებლის კონტექსტი და კონკრეტული სისტემა, სადაც გაფრთხილება მოხდა. მომხმარებლის სხვადასხვა როლმა და სისტემის კონფიგურაციამ შეიძლება გავლენა მოახდინოს გაფრთხილებების ინტერპრეტაციაზე.

დოკუმენტაცია და ცოდნის ბაზა:

შეამოწმეთ დოკუმენტაცია: მიმართეთ პროდუქტის დოკუმენტაციას, ცოდნის ბაზებს და შესაბამის წყაროებს, რათა გაიგოთ კონკრეტული გამაფრთხილებელი შეტყობინებების მნიშვნელობა. გამყიდველები ხშირად აწვდიან ინფორმაციას საერთო გაფრთხილებებისა და მათი გადაწყვეტილების შესახებ.

ცრუ დადებითი:

შეაფასეთ ცრუ დადებითი: გაითვალისწინეთ ცრუ პოზიტივის პოტენციალი. ზოგიერთი გაფრთხილება შეიძლება გამოწვეული იყოს კეთილთვისებიანი მოვლენებით, არასწორი კონფიგურაციებით ან დროებითი პრობლემებით.

გაფრთხილების ზღურბლები **Alert Thresholds** :

გაფრთხილების ზღვრების გადახედვა: გაიგეთ გამაფრთხილებელი შეტყობინებების გენერირებისთვის დაყენებული ზღვრები. საჭიროების შემთხვევაში დაარეგულირეთ ზღურბლები გარემოს სპეციფიკურ მოთხოვნებსა და მახასიათებლებთან შესაბამისობაში.

რეაგირება და შერბილება **Response and Mitigation** :

განსაზღვრეთ რეაგირების პროცედურები: ჩამოაყალიბეთ მკაფიო პროცედურები გამაფრთხილებელ შეტყობინებებზე რეაგირებისთვის. გამოიკვეთეთ გამოძიების, ესკალაციისა და შერბილების ნაბიჯები გაფრთხილების სიმძიმისა და ხასიათის მიხედვით.

შესვლა და შენახვა:

სათანადო ჟურნალის უზრუნველყოფა: დარწმუნდით, რომ გამაფრთხილებელი შეტყობინებები ადეკვატურად არის დარეგისტრირებული და ჟურნალები ინახება შესაბამისი ხანგრძლივობით. ეს ხელს უწყობს ისტორიულ ანალიზს და სასამართლო გამოძიებას.

უწყვეტი გაუმჯობესება:

გამეორება და გაუმჯობესება: რეგულარულად გადახედეთ და დახვეწეთ შეფასების პროცესი, რომელიც ეფუძნება განვითარებადი სისტემის მოთხოვნებს, საფრთხის ლანდშაფტის ცვლილებებს და ინციდენტებზე რეაგირების აქტივობებს.

ამ მოსაზრებების გამოყენებით გამაფრთხილებელი შეტყობინებების სისტემატური შეფასებით, ორგანიზაციებს შეუძლიათ გააძლიერონ პოტენციური პრობლემების დროულად გამოვლენისა და რეაგირების უნარი, გააუმჯობესონ სისტემის მთლიანი უსაფრთხოება და საიმედოობა.

ქსელის უსაფრთხოების მონაცემებთან მუშაობა მოიცავს მონიტორინგს, ანალიზს და რეაგირებას სხვადასხვა ქსელური მოწყობილობებისა და უსაფრთხოების ინსტრუმენტების მიერ წარმოქმნილ ინფორმაციას კიბერ საფრთხეებისგან დასაცავად. აქ მოცემულია გზამკვლევი, თუ როგორ უნდა ეფექტურად იმუშაოთ ქსელის უსაფრთხოების მონაცემებთან:

1. ქსელის უსაფრთხოების მონაცემების შეგროვება:

გამოიყენეთ ქსელის უსაფრთხოების მოწყობილობები: განათავსეთ ბუხარი, შეჭრის აღმოჩენის/პრევენციის სისტემები და ვებ აპლიკაციების ბუხარი ჟურნალებისა და გაფრთხილებების შესაქმნელად.

ჟურნალების ცენტრალიზება: სხვადასხვა ქსელური მოწყობილობებიდან და უსაფრთხოების ხელსაწყოებიდან ჟურნალების აგრეგაცია ცენტრალიზებულ მდებარეობაში ან SIEM (უსაფრთხოების ინფორმაციისა და ღონისძიებების მენეჯმენტის) სისტემაში.

2. ქსელის ტოპოლოგიის გაგება:

ქსელის დოკუმენტირება: გაიგეთ ორგანიზაციის ქსელის ტოპოლოგია, მათ შორის მარშრუტიზატორები, გადამრთველები, ბუხარი და სხვა მოწყობილობები.

კრიტიკული აქტივების იდენტიფიცირება: განსაზღვრეთ კრიტიკული აქტივების მდებარეობა და მათი კავშირი ქსელში.

3. ქსელის ტრაფიკის მონიტორინგი:

გამოიყენეთ ქსელის მონიტორინგის ხელსაწყოები: გამოიყენეთ ინსტრუმენტები, როგორიცაა Wireshark, tcpdump, ან ქსელის ნაკადის ანალიზატორები ქსელის ტრაფიკის დასაფიქსირებლად და გასაანალიზებლად.

ანომალიების მონიტორინგი: დააყენეთ გაფრთხილებები ქსელის უჩვეულო ქცევისთვის, როგორიცაა ტრაფიკის მოულოდნელი შაბლონები ან მწვერვალები.

4. უსაფრთხოების მოვლენების ანალიზი:

მოვლენების კორელაცია: გააანალიზეთ და დააკავშირეთ უსაფრთხოების მოვლენები სხვადასხვა წყაროდან, რათა დადგინდეს შაბლონები ან უსაფრთხოების პოტენციური ინციდენტები.

შეტყობინებების პრიორიტეტიზაცია: პრიორიტეტული გაფრთხილებები სიმძიმისა და ზემოქმედების მიხედვით, ფოკუსირება მათზე, რომლებიც ყველაზე მნიშვნელოვან რისკს წარმოადგენს.

5. ინციდენტის გამოვლენა და რეაგირება:

ინციდენტზე რეაგირების პროცედურების დადგენა: შეიმუშავეთ დოკუმენტირებული პროცედურები ქსელის უსაფრთხოების ინციდენტებზე რეაგირებისთვის.

გამოიკვლიეთ ინციდენტები: საფუძვლიანად გამოიძიეთ უსაფრთხოების ინციდენტები, რათა დადგინდეს ფარგლები, გავლენა და ძირითადი მიზეზები.



6. საფრთხის ინტელექტის გამოყენება **Utilizing Threat Intelligence** :

საფრთხის დაზვერვის ინტეგრირება: შეიტანეთ საფრთხის დაზვერვის წყაროები თქვენს უსაფრთხოების ინფრასტრუქტურაში საფრთხის აღმოჩენის გასაუმჯობესებლად.

იყავით ინფორმირებული: განაახლეთ უახლესი საფრთხის შესახებ დაზვერვის შესახებ, რათა გაიგოთ განვითარებადი საფრთხეები და დაუცველობა.

7. დაუცველობის მართვა **Vulnerability Management** :

ჩაატარეთ რეგულარული სკანირება: შეასრულეთ დაუცველობის შეფასებები და შეღწევადობის ტესტირება პოტენციური სისუსტეების იდენტიფიცირებისთვის და გამოსასწორებლად.

პაჩების მართვა: განაახლეთ ქსელის მოწყობილობები და პროგრამული უზრუნველყოფა უსაფრთხოების უახლესი პატჩებით.

8. წვდომის კონტროლი და ავთენტიფიკაცია:

ძლიერი წვდომის კონტროლის დანერგვა: განახორციელეთ მინიმალური პრივილეგიების პრინციპი და განახორციელეთ სწორი წვდომის კონტროლი ქსელის მოწყობილობებზე.

მომხმარებლის ავტორიზაციის მონიტორინგი: თვალი ადევნეთ ავტორიზაციის ჟურნალებს, რათა აღმოაჩინოთ არაავტორიზებული წვდომის მცდელობები.

9. შესვლა და აუდიტი:

ჟურნალის ჩართვა: დარწმუნდით, რომ ჟურნალი ჩართულია ქსელის მოწყობილობებზე და ჟურნალები რეგულარულად განიხილება.

აუდიტის კონფიგურაციები: პერიოდულად შეამოწმეთ ქსელის მოწყობილობების კონფიგურაციები უსაფრთხოების არასწორი კონფიგურაციის იდენტიფიცირებისთვის და გამოსასწორებლად.

10. დაშიფვრა და ქსელის სეგმენტაცია:

გამოიყენეთ დაშიფვრა: დაშიფრეთ მგრძნობიარე მონაცემები ტრანზიტში პროტოკოლების გამოყენებით, როგორცაა HTTPS, SSL/TLS.

ქსელის სეგმენტაციის განხორციელება: ქსელის სხვადასხვა სეგმენტის იზოლირება თავდამსხმელთა გვერდითი მოძრაობის შესაზღუდად.

11. თანამშრომელთა ტრენინგი და ინფორმირებულობა:

თანამშრომლების განათლება: ჩაატარეთ ტრენინგი უსაფრთხოების საუკეთესო პრაქტიკის შესახებ, რათა შემცირდეს სოციალური ინჟინერიის თავდასხმების და ინსაიდერული საფრთხეების ალბათობა.

12. უწყვეტი გაუმჯობესება:

გადახედეთ და ადაპტირება: რეგულარულად გადახედეთ უსაფრთხოების პროცესებს, ინციდენტებზე რეაგირებას და მიღებული გაკვეთილები ქსელის უსაფრთხოების ზოგადი მდგომარეობის გასაუმჯობესებლად.

13. შესაბამისობა და ანგარიშგება:

შესაბამისობის უზრუნველყოფა: გაიგეთ და დაიცავით შესაბამისი შესაბამისობის სტანდარტები და რეგულაციები.

ანგარიშების გენერირება: შეადგინეთ რეგულარული ანგარიშები ქსელის უსაფრთხოების მეტრიკისა და ინციდენტების შესახებ მენეჯმენტისა და შესაბამისობის მიზნებისთვის.

ქსელის უსაფრთხოების მონაცემებთან მუშაობა მოითხოვს ყოვლისმომცველ მიდგომას, აერთიანებს ტექნოლოგიას, პროცესებს და ადამიანურ გამოცდილებას, რათა ეფექტურად აღმოაჩინოს, უპასუხოს და შეამსუბუქოს უსაფრთხოების საფრთხეები დინამიურ და განვითარებად ლანდშაფტში.



ELSA (Elasticsearch, Logstash და Kibana) არის ღია კოდის ცენტრალიზებული ჟურნალის მართვის პლატფორმა, რომელიც შექმნილია ჟურნალის ეფექტური ანალიზისთვის. ის იყენებს Elasticsearch-ს მონაცემთა შესანახად და მოსაპოვებლად, Logstash-ს ჟურნალის დამუშავებისა და გამდიდრებისთვის და Kibana-ს ვიზუალიზაციისთვის. ELSA განსაკუთრებით ცნობილია თავისი მასშტაბურობით, რაც საშუალებას აძლევს ორგანიზაციებს ეფექტურად გაუმკლავდნენ ჟურნალის დიდი მოცულობის მონაცემებს. ის მხარს უჭერს ჟურნალის მონაცემების რეალურ დროში ძიებას, ანალიზს და ვიზუალიზაციას, რაც ანომალიებისა და უსაფრთხოების ინციდენტების სწრაფი გამოვლენის საშუალებას იძლევა.

მეორეს მხრივ, SGUIL (უსაფრთხოების ინფორმაციის და მოვლენების მართვის (SIEM) გრაფიკული მომხმარებლის ინტერფეისი ჟურნალის მონაცემებისთვის) არის უსაფრთხოებაზე ორიენტირებული პლატფორმა, რომელიც უზრუნველყოფს მომხმარებლის გრაფიკულ ინტერფეისს Suricata IDS/IPS-ის მიერ შეგროვებული მონაცემებისთვის (შექრის გამოვლენის სისტემა/შექრის პრევენცია). სისტემა). SGUIL ინტეგრირდება უსაფრთხოების სხვა ინსტრუმენტებთან და უზრუნველყოფს ქსელის უსაფრთხოების მოვლენების კონსოლიდირებულ ხედვას. ის გთავაზობთ ისეთ ფუნქციებს, როგორიცაა რეალურ დროში მოვლენის კორელაცია, პაკეტის დაჭერის ანალიზი და ინციდენტზე რეაგირების შესაძლებლობები. SGUIL აღიარებულია თავისი აქცენტით უსაფრთხოების ანალიტიკოსების დახმარებაზე უსაფრთხოების ინციდენტების იდენტიფიცირებაში და მათზე ეფექტური რეაგირებისთვის.

ორივე ELSA და SGUIL წვლილი შეაქვს ორგანიზაციის უსაფრთხოების პოზის გაძლიერებაში, შესთავაზა ლოგის მართვისა და ანალიზის ძლიერი შესაძლებლობები. ELSA აჯობებს ჟურნალების აგრეგაციასა და ანალიზს მონაცემთა მრავალფეროვან წყაროებში, რაც მას შესაფერისს ხდის უფრო ფართო გამოყენების შემთხვევებისთვის უსაფრთხოების მიღმა. SGUIL, სპეციალურად მორგებული უსაფრთხოების ოპერაციებისთვის, ფოკუსირებულია უსაფრთხოების მოვლენების ყოვლისმომცველი ხედვის უზრუნველყოფაზე, ეხმარება ანალიტიკოსებს ინფორმირებული გადაწყვეტილებების მიღებაში ინციდენტზე რეაგირების დროს.

ELSA-სა და SGUIL-ს შორის არჩევანი დამოკიდებულია ორგანიზაციის კონკრეტულ საჭიროებებზე და პრიორიტეტებზე. ELSA შეიძლება უპირატესობა მიანიჭოს მისი მრავალფეროვნებისა და ჟურნალის მართვის უფრო ფართო შესაძლებლობების გამო, ხოლო SGUIL შეიძლება არჩეული იყოს მისი სპეციალიზაციისთვის უსაფრთხოებასთან დაკავშირებულ გამოყენების შემთხვევებში, განსაკუთრებით მაშინ, როდესაც ინტეგრირებულია Suricata-სთან. საბოლოო ჯამში, არჩევანი უნდა შეესაბამებოდეს ორგანიზაციის მიზნებს, ჟურნალის მონაცემების მასშტაბს და აქცენტს ზოგადი ჟურნალის მენეჯმენტზე უსაფრთხოების სპეციფიკურ მოთხოვნებთან მიმართებაში.

# ინფორმაციული უსაფრთხოება

ლექცია 8

tamar.kurdadze@btu.edu.ge

- ფაიერვოლები;
- შემოჭრების გამოვლენისა და პრევენციის სისტემები;
- წვდომის კონტროლის სიები;
- SNMP, NetFlow, Syslog
- NTP; AAA სერვერები;
- VPN

Intrusion detection systems(IDS) არის ინსტრუმენტი, რომელიც ჩვეულებრივ გამოიყენება ქსელების დასაცავად საექვო აქტივობის გამოვლენის ავტომატიზაციის გზით. როდესაც firewall-მა, ანტივირუსმა ან ავტორიზაციის სისტემამ შეიძლება ხელი შეუშალოს გარკვეული აქტივობის წარმოქმნას IT აქტივებზე ან მის წინააღმდეგ, IDS სანაცვლოდ მონიტორინგს გაუწევს აქტივობას, რომელიც არ არის შეზღუდული და დააღაგებს მავნე ფაქტორებს. IDS ჩვეულებრივ იყენებს გამოვლენის ორი განსხვავებული მეთოდოლოგიიდან ერთს; ხელმოწერებზე (ან წესებზე) დაფუძნებული IDS გამოიყენებს დიდი წესების კომპლექტს მონაცემთა ერთი ან მეტი წყაროს საექვო აქტივობის მოსაძიებლად, ხოლო ანომალიებზე დაფუძნებული IDS ადგენს, თუ რა ითვლება ნორმალურ აქტივობაზე და შემდეგ ამაღლებს გაფრთხილებებს, როდესაც აღმოჩენილია აქტივობა, რომელიც არ შეესაბამება საბაზისო ხაზს. .

ნებისმიერ შემთხვევაში, ინციდენტის აღმოჩენის შემდეგ, IDS გამოიმუშავებს გაფრთხილებას და შემდეგ გააგზავნის მას უსაფრთხოების ჯაჭვში ჟურნალის აგრეგაციის ან მონაცემთა ვიზუალიზაციის პლატფორმებისთვის, როგორცაა Graylog ან ELK Stack.

ზოგიერთ IDS-ს შეიძლება ასევე ჰქონდეს შეჭრის პრევენციის გარკვეული ფორმა და შეიძლება ავტომატურად რეაგირებდეს ინციდენტზე.

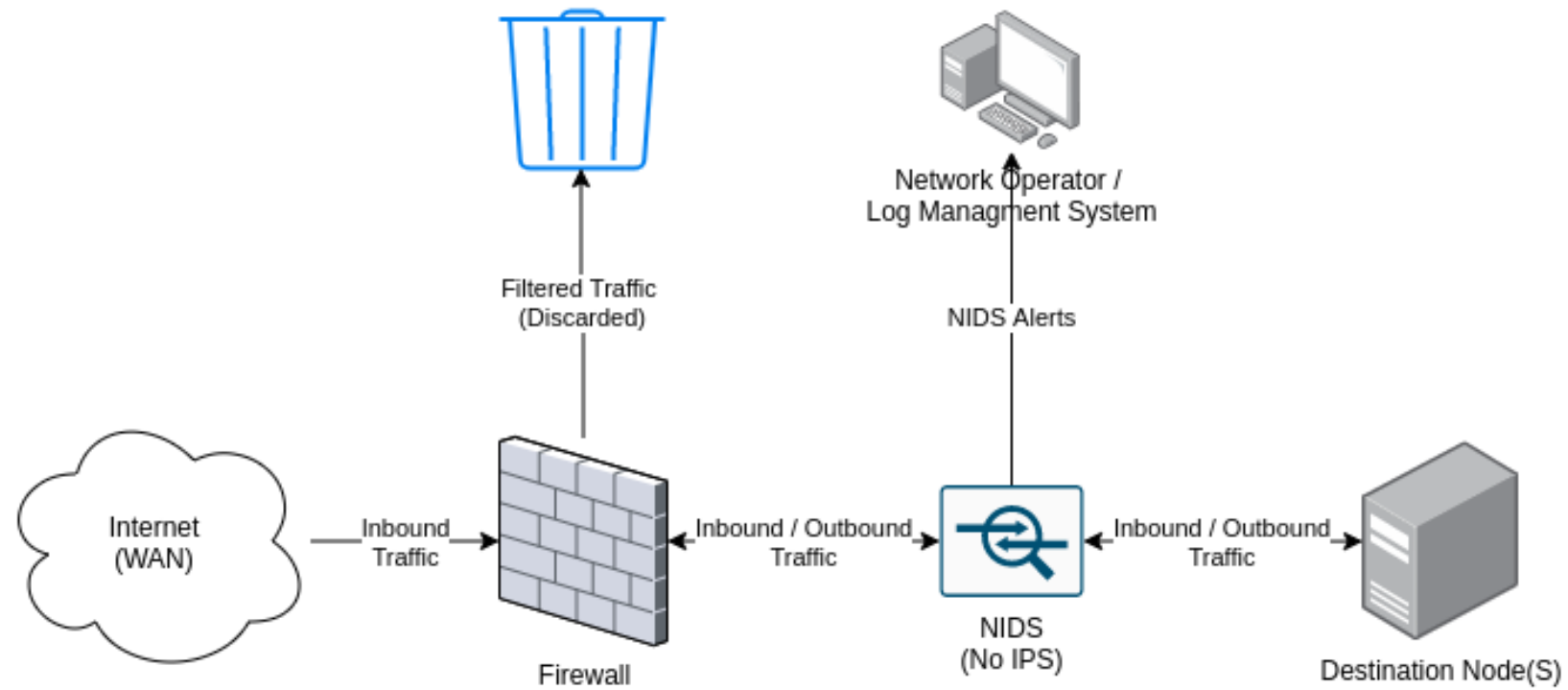
ამ დემო ვერსიას მიმაგრებულია ორი ხელმოწერებზე დაფუძნებული IDS; Suricata, ქსელზე დაფუძნებული IDS (NIDS) და Wazuh, ჰოსტზე დაფუძნებული IDS (HIDS). ორივე IDS ახორციელებს ხელმოწერის აღმოჩენის იგივე ყოვლისმომცველ მეთოდოლოგიას; თუმცა, მათი საერთო ქცევა და თავდასხმების ტიპები, რომელთა აღმოჩენაც მათ შეუძლიათ, მნიშვნელოვნად განსხვავდება. ზუსტ განსხვავებებს უფრო დეტალურად განვიხილავთ შემდეგ ამოცანებში.

როგორც სახელი გულისხმობს, ქსელში შეჭრის აღმოჩენის სისტემები ან NIDS მონიტორინგს უწევს ქსელებს მავნე აქტივობისთვის, ამოწმებს პაკეტებს აქტივობის კვალზე, რომელიც დაკავშირებულია მრავალფეროვან მტრულ ან არასასურველ აქტივობასთან, მათ შორის:

- Malware command and control
- Exploitation tools
- Scanning
- Data exfiltration
- Contact with phishing sites
- Corporate policy violations

ქსელზე დაფუძნებული ამოცნობა საშუალებას აძლევს ერთ ინსტალაციას დააკვირდეს მთლიან ქსელს, რაც NIDS-ის განლაგებას სხვა ტიპებთან შედარებით უფრო მარტივს ხდის. თუმცა, NIDS უფრო მიდრეკილია ცრუ პოზიტივის წარმოქმნისკენ, ვიდრე სხვა IDS, ეს ნაწილობრივ განპირობებულია ტრაფიკის დიდი მოცულობით, რომელიც გადის თუნდაც მცირე ქსელში და, საკმარისად მოქნილი წესების შექმნის სირთულით, რათა საიმედოდ აღმოაჩინოს მავნე ტრაფიკი გარეშე. უსაფრთხო აპლიკაციების აღმოჩენა, რომლებმაც შეიძლება დატოვონ მსგავსი კვალი. ეს შეიძლება გარკვეულწილად შემსუბუქდეს IDS-ის დარეგულირებით, რათა მხოლოდ იმ წესების აღსრულება მოხდეს, რომლებიც ჩაითვლება არანორმალურ ტრაფიკად რომელიმე კონკრეტული ქსელისთვის, თუმცა ამას გარკვეული დრო სჭირდება, რადგან IDS უნდა განთავსდეს ქსელში გარკვეული დროით, რათა დადგინდეს რა არის ტრაფიკი. ნორმალური. NIDS შეიძლება განლაგდეს firewall-ის ორივე მხარეს, თუმცა ისინი განლაგებულია LAN-ის მხარეს, რადგან შეზღუდული მნიშვნელობა აქვს გარე კვანძების წინააღმდეგ განხორციელებულ შეტევებს, რადგან ისინი მუდმივად იქნებიან თავდასხმის ქვეშ.

### Example NIDS Deployment



მავენე აქტივობის ყველა ფორმა არ მოიცავს ქსელურ ტრაფიკს, რომელიც შეიძლება გამოვლინდეს NIDS-ის მიერ, გამოსასყიდი პროგრამა, მაგალითად, შეიძლება შეწუხდეს ელ.ფოსტის სერვისის გარე პროვაიდერის მეშვეობით, რომელიც დაინსტალირებული და შესრულებულია სამიზნე მანქანაზე და NIDS-ის მიერ მხოლოდ ერთხელ აღმოჩენილია, ის რეკავს. სახლში მისი წარმატების შეტყობინებებით, რაც, რა თქმა უნდა, ძალიან გვიანია. ამ მიზეზით, ხშირად მიზანშეწონილია განათავსოთ ჰოსტზე დაფუძნებული IDS NIDS-თან ერთად, რათა შეამოწმოთ საექვო აქტივობა, რომელიც ხდება მოწყობილობებზე და არა მხოლოდ ქსელში, მათ შორის:

- Malware execution
- System configuration changes
- Software errors
- File integrity changes
- Privilege escalation

HIDS განლაგება შეიძლება იყოს ბევრად უფრო რთული, ვიდრე NIDS, რადგან ისინი ხშირად საჭიროებენ აგენტის ინსტალაციას და მართვას თითოეულ ჰოსტზე, რომელიც უნდა იყოს დაფარული HIDS-ით. ეს აგენტი, როგორც წესი, აგზავნის აქტივობას სისტემის მონაცემთა წყაროებიდან ცენტრალურ მართვისა და დამუშავების კვანძში, რომელიც შემდეგ მიმართავს წესებს გადაგზავნილ მონაცემებზე ნებისმიერი სხვა IDS-ის მსგავსად. ეს მონაცემთა წყაროები, როგორც წესი, მოიცავს:

- Application and system log files
- The Windows registry
- System performance metrics
- The state of the file system itself



შექრის პრევენციის სისტემა (IPS) არის ქსელის უსაფრთხოების გადაწყვეტა, რომელიც შექმნილია ქსელის ან სისტემის აქტივობების მონიტორინგისა და ანალიზისთვის მავნე ან არასასურველი ქცევისთვის და რეაგირება ამ აქტივობებზე რეალურ დროში. ეს არის პროაქტიული უსაფრთხოების ღონისძიება, რომელიც სცილდება ტრადიციული ბუხრისა და შექრის აღმოჩენის სისტემების (IDS) შესაძლებლობებს.

აქ მოცემულია IPS-ის რამდენიმე ძირითადი მახასიათებელი და ფუნქცია:

**1.Monitoring and Analysis:** IPS მუდმივად აკონტროლებს ქსელის ტრაფიკს და აანალიზებს მას შაბლონებისთვის, რომლებიც შეიძლება მიუთითებდეს მავნე აქტივობაზე ან უსაფრთხოების პოლიტიკის დარღვევაზე.

**2.Signature-based Detection:** IPS სისტემები ხშირად იყენებენ ცნობილი თავდასხმის ხელმოწერების მონაცემთა ბაზას, რათა ამოიცნონ და დაბლოკონ ცნობილი საფრთხეები, როგორცაა მავნე პროგრამები ან თავდასხმის შაბლონები.

**3.Anomaly-based Detection:** ხელმოწერებზე დაფუძნებული გამოვლენის გარდა, IPS-ს ასევე შეუძლია გამოიყენოს ანომალიებზე დაფუძნებული გამოვლენა ქცევის არანორმალური ნიმუშების დასადგენად, რაც შეიძლება მიუთითებდეს ახალ ან უცნობ საფრთხეზე.

**4.Real-time Response:** როდესაც აღმოჩენილია საეჭვო ან მავნე აქტივობა, IPS-ს შეუძლია დაუყოვნებლივ მიიღოს ზომები საფრთხის დაბლოკვის ან თავიდან ასაცილებლად. ეს შეიძლება მოიცავდეს კონკრეტული ქსელური ტრაფიკის დაბლოკვას, ქსელური კავშირების დახურვას ან სხვა მოქმედებებს თავდასხმის პოტენციური ზემოქმედების შესამცირებლად.

**5.Integration with Firewalls:** IPS ხშირად ინტეგრირებულია Firewall-ებთან, რათა უზრუნველყოს ყოვლისმომცველი უსაფრთხოების გადაწყვეტა. მიუხედავად იმისა, რომ firewalls ფოკუსირებულია შემომავალი და გამავალი ქსელის ტრაფიკის კონტროლზე, წინასწარ განსაზღვრული უსაფრთხოების წესების საფუძველზე, IPS ამატებს დაცვის დამატებით ფენას პოტენციური საფრთხეების აქტიური იდენტიფიკაციისა და დაბლოკვით.

**6.Tuning and Customization:** IPS გადაწყვეტილებები ჩვეულებრივ ადმინისტრატორებს საშუალებას აძლევს დააკონფიგურირონ და დააკონფიგურირონ სისტემა თავიანთი ქსელის სპეციფიკურ საჭიროებებზე. ეს შეიძლება მოიცავდეს გამოვლენის მგრძნობელობის რეგულირებას, მორგებული წესების შექმნას ან რეაგირების მოქმედებების დაზუსტებას.

Firewall არის ქსელის უსაფრთხოების მოწყობილობა ან პროგრამული უზრუნველყოფა, რომელიც აკონტროლებს და აკონტროლებს შემომავალი და გამავალი ქსელის ტრაფიკს უსაფრთხოების წინასწარ განსაზღვრული წესების საფუძველზე. Firewall-ის მთავარი მიზანია დაამყაროს ბარიერი უსაფრთხო შიდა ქსელსა და არასანდო გარე ქსელებს შორის, როგორიცაა ინტერნეტი. Firewalls არის ქსელის უსაფრთხოების აუცილებელი კომპონენტები და ხელს უწყობენ არაავტორიზებული წვდომის, მონაცემთა ექსფილტრაციისა და უსაფრთხოების სხვა საფრთხეების თავიდან აცილებას. აქ არის Firewall-ის რამდენიმე ძირითადი ასპექტი:

**Packet Filtering:** Firewall-ებს შეუძლიათ ქსელის პაკეტების გაფილტვრა წინასწარ განსაზღვრული წესების საფუძველზე. ეს წესები განსაზღვრავს თუ რომელი პაკეტებია დაშვებული ან უარყოფილი კრიტერიუმების საფუძველზე, როგორიცაა წყარო და დანიშნულების IP მისამართები, პორტები და გამოყენებული პროტოკოლის ტიპი.

**Stateful Inspection:** სახელმწიფო ინსპექტირება, რომელიც ასევე ცნობილია როგორც დინამიური პაკეტის ფილტრაცია, მოიცავს აქტიური კავშირების მდგომარეობის თვალყურს და გადაწყვეტილებების მიღებას ტრაფიკის კონტექსტზე დაყრდნობით. ეს საშუალებას აძლევს Firewall-ებს მიიღონ უფრო ინფორმირებული გადაწყვეტილებები პაკეტების დაშვების ან დაბლოკვის შესახებ.

**Proxy Services:** ზოგიერთი firewalls მოქმედებს როგორც შუამავალი შიდა მომხმარებლებსა და გარე სერვერებს შორის. მათ შეუძლიათ პროქსის მოთხოვნები, რაც უფრო რთულს გახდის თავდამსხმელებისთვის უშუალოდ შიდა სისტემებთან ინტერაქციას. პროქსი სერვისებს ასევე შეუძლიათ უზრუნველყონ შინაარსის გაფილტვრა და ჟურნალის შესაძლებლობები.

**Network Address Translation (NAT):** Firewalls ხშირად იყენებენ NAT-ს, რათა შეცვალონ ქსელის მისამართის ინფორმაცია პაკეტის სათაურებში, როდესაც ტრაფიკი გადის. ეს ხელს შეუწყობს შიდა IP მისამართების დამალვას გარე ქსელებიდან.

**Application-layer Filtering:** ზოგიერთი თანამედროვე ): Firewall უზრუნველყოფს აპლიკაციის ფენის ფილტრაციის შესაძლებლობებს, ამოწმებს მონაცემთა პაკეტების შიგთავსს კონკრეტული აპლიკაციების ან სერვისების იდენტიფიცირებისა და დაბლოკვის მიზნით. ამას ხშირად უწოდებენ ღრმა პაკეტის შემოწმებას.

**Logging and Reporting:** Firewalls ჩვეულებრივ ინახავს ქსელის აქტივობის ჟურნალებს, რაც შეიძლება სასარგებლო იყოს უსაფრთხოების მოვლენების მონიტორინგისა და ანალიზისთვის. ბევრი Firewall ასევე გთავაზობთ მოხსენების ფუნქციებს, რათა დაეხმაროს ადმინისტრატორებს გააცნობიერონ ქსელის ტრაფიკის შაბლონები და უსაფრთხოების პოტენციური საფრთხეები.

WAF" ნიშნავს Web Application Firewall-ს. Web Application Firewall არის უსაფრთხოების გადაწყვეტა, რომელიც შექმნილია ვებ აპლიკაციების დასაცავად სხვადასხვა კიბერ საფრთხეებისგან, მათ შორის საერთო ვებ აპლიკაციების დაუცველობისა და თავდასხმებისგან. ის მუშაობს OSI (ღია სისტემების ურთიერთდაკავშირების) მოდელის აპლიკაციის ფენაზე. HTTP ტრაფიკის შემოწმება და კონტროლი ვებ აპლიკაციიდან.

ვებ აპლიკაციის Firewall-ის ძირითადი მახასიათებლები და ფუნქციები მოიცავს:

**1.Application-Layer Protection:** WAF-ები ფოკუსირებულია ვებ აპლიკაციების დაცვაზე განაცხადის ფენაზე, სადაც ხდება მრავალი დაუცველობა და შეტევა, რომელიც მიზნად ისახავს ვებ აპლიკაციებს.

**2.HTTP Traffic Monitoring:** WAF-ები აკონტროლებენ და აანალიზებენ HTTP ტრაფიკს ვებ კლიენტებსა და ვებ სერვერებს შორის. ისინი ამოწმებენ HTTP მოთხოვნებისა და პასუხების შინაარსს, რათა ამოიცნონ და დაბლოკონ მავნე აქტივობა.

**3.Attack Signature Detection:** შეჭრის პრევენციის სისტემების მსგავსად, WAF-ები ხშირად იყენებენ წინასწარ განსაზღვრულ თავდასხმის ხელმოწერებს, რათა აღმოაჩინონ და დაბლოკონ ცნობილი ვებ აპლიკაციების შეტევები, როგორიცაა SQL ინექცია, საიტის სკრიპტირება (XSS) და საიტის მოთხოვნის გაყალბება (CSRF).

კიბერუსაფრთხოებაში წვდომის კონტროლის სია (ACL) არის წესების ან ნებართვების ერთობლიობა, რომლებიც გამოიყენება რესურსებზე ან ქსელურ სერვისებზე წვდომის გასაკონტროლებლად. ACL ჩვეულებრივ გამოიყენება კომპიუტერულ სისტემებზე, ქსელებზე ან კონკრეტულ ფაილებსა და კატალოგებზე წვდომის დასარეგულირებლად. ACL-ის მიზანია განსაზღვროს რომელ მომხმარებლებს ან სისტემის პროცესებს ენიჭებათ ან უარს ეუბნებიან წვდომას გარკვეულ რესურსებზე წინასწარ განსაზღვრული კრიტერიუმების საფუძველზე.

კიბერუსაფრთხოებაში წვდომის კონტროლის სიების ძირითადი მახასიათებლები მოიცავს:

**1. Identification of Resources:** ACL, როგორც წესი, განსაზღვრავს რესურსებს ან ობიექტებს, რომლებზეც ვრცელდება წვდომის კონტროლის წესები. ეს რესურსები შეიძლება შეიცავდეს ფაილებს, დირექტორიებს, ქსელურ მოწყობილობებს ან კონკრეტულ სერვისებს.

**2. Permissions:** ACL განსაზღვრავს ნებართვებს ან ქმედებებს, რომლებიც დაშვებულია ან უარყოფილია კონკრეტული რესურსისთვის. ნებართვები შეიძლება შეიცავდეს წაკითხვას, ჩაწერას, შესრულებას, შეცვლას, წაშლას ან სხვა სპეციფიკურ მოქმედებებს რესურსის ტიპის მიხედვით.

**3. Subjects (Users or Groups):** ACL-ები განსაზღვრავს სუბიექტებს ან ერთეულებს, რომლებიც ექვემდებარებიან წვდომის კონტროლის წესებს. ეს ერთეულები შეიძლება იყოს ინდივიდუალური მომხმარებლები, მომხმარებელთა ჯგუფები ან სისტემის პროცესები.

ვირტუალური პირადი ქსელი (VPN) არის ტექნოლოგია, რომელიც უზრუნველყოფს უსაფრთხო და დაშიფრულ კავშირს ინტერნეტის საშუალებით, რაც მომხმარებლებს საშუალებას აძლევს წვდომას იქონიონ კერძო ქსელში, როგორცაა კორპორატიული ინტრანეტი, დისტანციური ადგილიდან. VPN-ები ჩვეულებრივ გამოიყენება კონფიდენციალურობისა და უსაფრთხოების გასაუმჯობესებლად ინტერნეტში წვდომისას, განსაკუთრებით საჯარო ქსელების გამოყენებისას, როგორცაა Wi-Fi ცხელ წერტილები.

აქ მოცემულია VPN-ის ძირითადი ასპექტები:

**Secure Data Transmission:** VPN-ის ერთ-ერთი მთავარი მიზანია შექმნას უსაფრთხო და დაშიფრული გვირაბი ინტერნეტით მონაცემთა გადაცემისთვის. ეს დაშიფვრა ხელს უწყობს მომხმარებლის მოწყობილობასა და VPN სერვერს შორის გადაცემული მონაცემების კონფიდენციალურობისა და მთლიანობის დაცვას.

**Remote Access:** VPN-ები საშუალებას აძლევს დისტანციურ მომხმარებლებს დაუკავშირდნენ კერძო ქსელს, თითქოს ისინი ფიზიკურად იმყოფებოდნენ იმავე ადგილას. ეს განსაკუთრებით სასარგებლოა იმ თანამშრომლებისთვის, რომლებიც მუშაობენ სახლიდან ან კომპანიის რესურსებზე წვდომას ახდენენ მოძრაობისას.

**Anonymity and Privacy:** ინტერნეტ ტრაფიკის დაშიფვრით, VPN-ები ხელს უწყობენ მომხმარებლის კონფიდენციალურობის გაძლიერებას მესამე მხარის, როგორცაა ინტერნეტ სერვისის პროვაიდერების ან ჰაკერების, მონაცემების მონიტორინგის ან ჩარევის თავიდან აცილების გზით. გარდა ამისა, VPN-ებს შეუძლიათ შენიღბოს მომხმარებლის IP მისამართი, რაც ართულებს მათი ონლაინ აქტივობების მიკვლევას.

**Bypassing Geo-restrictions:** მომხმარებლებს შეუძლიათ გამოიყენონ VPN, რათა გამოჩნდნენ ისე, თითქოს ისინი წვდებიან ინტერნეტს სხვა მდებარეობიდან, რაც მათ საშუალებას აძლევს გვერდის ავლით გადალახონ გეო-შეზღუდვები და მიიღონ წვდომა კონტენტზე, რომელიც შეიძლება შეიზღუდოს მათ რეალურ ფიზიკურ მდებარეობაში.

**Types of VPNs:**

1. **Remote Access VPN:** ცალკეულ მომხმარებლებს საშუალებას აძლევს უსაფრთხოდ დაუკავშირდნენ კერძო ქსელს დისტანციური მდებარეობიდან.
2. **Site-to-Site VPN:** აკავშირებს მთელ ქსელებს ერთმანეთთან, როგორცაა ფილიალების ცენტრალურ კორპორატიულ ქსელთან დაკავშირება.
3. **Client-to-Site VPN :** VPN დისტანციური წვდომის მსგავსი, მაგრამ, როგორც წესი, გულისხმობს ცალკეული კლიენტების (მოწყობილობების) კონკრეტულ საიტთან დაკავშირებას.

SNMP (Simple Network Management Protocol), NetFlow და Syslog არის ყველა ქსელის მართვის პროტოკოლი და ხელსაწყო, რომელიც გამოიყენება ქსელის სფეროში, ქსელის მოწყობილობების მონიტორინგისა და მართვისთვის, ქსელის აქტივობების შესახებ ინფორმაციის შეგროვებისა და პრობლემების მოსაგვარებლად. აქ მოცემულია თითოეულის მიმოხილვა:

#### Syslog:

**მიზანი:** Syslog არის სტანდარტი ქსელის მოწყობილობებიდან, აპლიკაციებიდან და ოპერაციული სისტემებიდან სისტემის შეტყობინებების შესვლისა და შეგროვებისთვის.

**შეტყობინებების ფორმატი:** Syslog შეტყობინებები, როგორც წესი, შეიცავს ინფორმაციას მოვლენების, შეცდომების, გაფრთხილებებისა და სისტემასთან დაკავშირებული სხვა აქტივობების შესახებ. თითოეული შეტყობინება შეიცავს დროის ნიშანს, შეტყობინების წყაროს და დეტალებს მოვლენის შესახებ.

**ცენტრალიზებული შესვლა:** Syslog შეტყობინებები შეიძლება გაიგზავნოს ცენტრალიზებულ syslog სერვერზე შენახვის, ანალიზისა და მონიტორინგისთვის. ეს ცენტრალიზებული ჟურნალი ხელს უწყობს პრობლემების აღმოფხვრას და აუდიტის ბილიკის შენარჩუნებას.

**სიმძიმის დონეები:** Syslog შეტყობინებები დაყოფილია სიმძიმის დონეებად, დაწყებული ინფორმაციულიდან კრიტიკულამდე, რაც ადმინისტრატორებს აძლევს პრიორიტეტების მინიჭების საშუალებას და შესაბამისი რეაგირების საკითხებს.

#### NetFlow:

**მიზანი:** NetFlow არის ქსელის პროტოკოლი, რომელიც შემუშავებულია Cisco-ს მიერ IP ტრაფიკის ინფორმაციის შეგროვებისა და ქსელის ნაკადის მონაცემების მონიტორინგისთვის.

**მონაცემთა შეგროვება:** NetFlow ჩართული მოწყობილობები (როგორიცაა მარშრუტიზატორები და გადამრთველები) აგროვებენ ინფორმაციას ქსელის ტრაფიკის შესახებ, მათ შორის წყაროსა და დანიშნულების IP მისამართებს, პორტებს, პროტოკოლებს და გადაცემული მონაცემების რაოდენობას.

**ანალიზი:** NetFlow მონაცემები გამოიყენება ქსელის ტრაფიკის ანალიზისთვის, გამტარუნარიანობის მონიტორინგისთვის და ქსელის ქცევაში ტენდენციების ან ანომალიების დასადგენად.

**ვერსიები:** არსებობს NetFlow-ის სხვადასხვა ვერსიები, NetFlow v5 და NetFlow v9 ჩვეულებრივ გამოყენებულ ვერსიებს შორისაა.

- მარტივი ქსელის მართვის პროტოკოლი (SNMP):
- მიზანი: SNMP არის პროტოკოლი, რომელიც გამოიყენება ქსელური მოწყობილობების მართვისა და მონიტორინგისთვის, როგორიცაა მარშრუტიზატორები, გადამრთველები, სერვერები და პრინტერები.
- კომპონენტები: SNMP შედგება სამი ძირითადი კომპონენტისგან: SNMP მენეჯერი, SNMP აგენტი და მართვის საინფორმაციო ბაზა (MIB). მენეჯერი ურთიერთობს აგენტებთან ქსელურ მოწყობილობებზე და MIB განსაზღვრავს მონაცემებს, რომელთა გაცვლაც შესაძლებელია მათ შორის.
- ოპერაციები: SNMP საშუალებას აძლევს მენეჯერს მიიღოს ინფორმაცია აგენტებისგან, დააყენოს კონფიგურაციის პარამეტრები და მიიღოს შეტყობინებები (ხაფანგები) კონკრეტული მოვლენების შესახებ ქსელის მოწყობილობებზე.
- ვერსია: არსებობს SNMP-ის სხვადასხვა ვერსია, SNMPv3 არის ყველაზე უსაფრთხო და ფართოდ გამოყენებული ვერსია.

NTP (ქსელის დროის პროტოკოლი) და AAA (ავთენტიფიკაცია, ავტორიზაცია და აღრიცხვა) ორი განსხვავებული კონცეფციაა ქსელის სფეროში.

**Network Time Protocol (NTP):**

მიზანი: NTP არის პროტოკოლი, რომელიც გამოიყენება ქსელში კომპიუტერული სისტემების საათების სინქრონიზაციისთვის. ის უზრუნველყოფს, რომ მოწყობილობებს ქსელის მასშტაბით აქვთ თანმიმდევრული და ზუსტი დრო.

მნიშვნელობა: დროის ზუსტი აღრიცხვა გადამწყვეტია სხვადასხვა ქსელის ოპერაციებისთვის, უსაფრთხოების პროტოკოლებისთვის და ჟურნალისთვის. NTP ეხმარება განაწილებულ სისტემებს შორის სინქრონიზაციის შენარჩუნებაში.

ოპერაცია: NTP მუშაობს კლიენტ-სერვერის მოდელზე, სადაც კლიენტის მოწყობილობები სინქრონიზებენ საათებს დანიშნულ დროის სერვერთან. თავად დროის სერვერი შეიძლება სინქრონიზდეს სხვა სერვერებთან იერარქიული გზით.

**AAA (Authentication, Authorization, and Accounting):**

მიზანი: AAA არის ჩარჩო ან პროტოკოლების ნაკრები, რომელიც ერთობლივად უზრუნველყოფს წვდომის კონტროლის, მომხმარებლის ავთენტიფიკაციის, ავტორიზაციისა და ქსელის რესურსების აღრიცხვის საშუალებას.

ავთენტიფიკაცია: მომხმარებლების ან მოწყობილობების იდენტიფიკაციის შემოწმება, რომლებიც ცდილობენ ქსელში წვდომას.

ავტორიზაცია: წვდომის დონის ან პრივილეგიების განსაზღვრა ავტორიზებული მომხმარებლებისთვის ან მოწყობილობებისთვის.

ბუღალტერია: მომხმარებლების ან მოწყობილობების საქმიანობის აღრიცხვა და თვალყურის დევნება აუდიტისა და ბილინგის მიზნებისთვის.

პროტოკოლები: საერთო AAA პროტოკოლებს შორისაა RADIUS (დისტანციური ავთენტიფიკაციის Dial-In მომხმარებლის სერვისი) და TACACS+ (ტერმინალის წვდომის კონტროლური Access-Control System Plus).

გამოყენების შემთხვევები: AAA ჩვეულებრივ გამოიყენება ქსელის უსაფრთხოებაში, განსაკუთრებით ისეთ სცენარებში, როგორიცაა დისტანციური წვდომის ავტორიზაცია, VPN (ვირტუალური პირადი ქსელი) წვდომის კონტროლი და მოწყობილობის ადმინისტრირება.



დავალება:  
Elearning.gov.ge

# ინფორმაციული უსაფრთხოება

ლექცია 6

tamar.kurdadze@btu.edu.ge

- კრიპტოგრაფიული ჰეშები;
- ჰეშირების ალგორითმები;
- მთლიანობა MD5, SHA-1, SHA-2 ალგორითმებით;
- პაროლების ჰეშირება;
- დანამატი (salting);
- აუთენტიკაცია HMAC-ით;
- მონაცემთა ბაზების მთლიანობის უზრუნველყოფა

კრიპტოგრაფიული ჰეშები და ჰეშირების ალგორითმები ფუნდამენტური ცნებებია კრიპტოგრაფიისა და ინფორმაციის უსაფრთხოებაში. კრიპტოგრაფიული ჰეშის ფუნქცია არის მათემატიკური ალგორითმი, რომელიც იღებს შეყვანის მონაცემებს (ან „შეტყობინებებს“) და აწარმოებს სიმბოლოების ფიქსირებული ზომის სტრიქონს, რომელიც, როგორც წესი, თექვსმეტობით რიცხვს წარმოადგენს. აქ მოცემულია კრიპტოგრაფიული ჰეშებისა და ჰეშირების ალგორითმების ძირითადი ასპექტები:

## Cryptographic Hashes:

### 1.Purpose:

1. Data Integrity: კრიპტოგრაფიული ჰეშები ჩვეულებრივ გამოიყენება მონაცემთა მთლიანობის უზრუნველსაყოფად. შეყვანის მონაცემების მცირე ცვლილებამაც კი უნდა გამოიწვიოს მნიშვნელოვნად განსხვავებული ჰეში.
2. Digital Signatures: ჰეშები გამოიყენება ციფრულ ხელმოწერებში დოკუმენტის ან შეტყობინების უნიკალური წარმოდგენის შესაქმნელად, რაც უზრუნველყოფს გამგზავნის ავთენტურობის გადამოწმების საშუალებას.
3. Password Storage: ჰეშის ფუნქციები გამოიყენება პაროლების უსაფრთხოდ შესანახად. რეალური პაროლების შენახვის ნაცვლად, სისტემები ინახავს პაროლის ჰეშს და ამატებენ დაცვის ფენას.
4. Properties of Cryptographic Hash Functions:
5. Deterministic: ერთი და იგივე შეყვანა ყოველთვის გამოიმუშავებს იგივე ჰეშის გამომავალს.
6. Fast Computation: ჰეშის ფუნქცია უნდა იყოს ეფექტური გამოსათვლელად.
7. Irreversibility: გამოთვლებით შეუძლებელი უნდა იყოს ჰეშის შებრუნება და ორიგინალური შეყვანის მიღება.
8. Fixed Output Size: ჰეშის გამომავალს აქვს ფიქსირებული ზომა, შეყვანის ზომის მიუხედავად.
9. Avalanche Effect: შეყვანის მცირე ცვლილებამ უნდა გამოიწვიოს სრულიად განსხვავებული ჰეში.

# Hashing Algorithms:

## 1. Common Cryptographic Hashing Algorithms:

- **MD5 (Message Digest Algorithm 5):** მიუხედავად იმისა, რომ წარსულში ფართოდ გამოიყენებოდა, MD5 ახლა სუსტად ითვლება დაუცველობის გამო. არ არის რეკომენდებული უსაფრთხოების მიმართ მგრძნობიარე აპლიკაციებისთვის.
- **SHA-1 (Secure Hash Algorithm 1):** MD5-ის მსგავსად, SHA-1-ს აქვს მოწყვლადობა და მისი გამოყენება უსაფრთხოებისადმი მგრძნობიარე აპლიკაციებში დაუშვებელია.
- **SHA-256, SHA-384, SHA-512 (Secure Hash Algorithm 2):** ეს SHA-2 ვარიანტები ამჟამად ითვლება უსაფრთხოდ და ფართოდ გამოიყენება უსაფრთხოების სხვადასხვა აპლიკაციებისთვის.
- **SHA-3 (Secure Hash Algorithm 3):** ოჯახის უახლესი წევრი, შექმნილია უზრუნველყოს დამატებითი ვარიანტი უსაფრთხო ჰეშინგისთვის.
- **BLAKE2:** კრიპტოგრაფიული ჰეშის ფუნქცია, რომელიც უფრო სწრაფია ვიდრე MD5, SHA-1, SHA-2 და SHA-3. ის შექმნილია სწრაფი და უსაფრთხო აპლიკაციების ფართო სპექტრისთვის.
- **Choosing a Hashing Algorithm:**
- **Security Requirements:** ჰეშირების ალგორითმის არჩევანი დამოკიდებულია კონკრეტული აპლიკაციის უსაფრთხოების მოთხოვნებზე. უმეტეს აპლიკაციებისთვის რეკომენდებულია SHA-256 ან უფრო მაღალი.
- **Performance:** განვიხილოთ ჰეშირების ალგორითმის გამოთვლითი ეფექტურობა, განსაკუთრებით აპლიკაციებისთვის, რომლებსაც აქვთ მაღალი გამტარუნარიანობა.
- **Compatibility:** უზრუნველყოს არსებულ სისტემებთან და სტანდარტებთან თავსებადობა.
- **Cryptographic Hash Example (SHA-256):**

plaintext Copy code

Input: "Hello, World!"  
SHA-256 Hash: a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9

## 1. MD5 (Message Digest Algorithm 5):

- Original Purpose:** MD5 შეიქმნა 128-ბიტისანი ჰეშის მნიშვნელობის (32 ბიტის სიგრძით სიმბოლო) შესაქმნელად და ფართოდ გამოიყენებოდა შემოწმების ჯამებისა და მთლიანობის გადამოწმებისთვის

- Security Concerns:** MD5 ახლა კრიპტოგრაფიულად გატეხილია და უვარგისია შემდგომი გამოყენებისთვის დაუცველობის გამო, რომელიც საშუალებას აძლევს შეჯახების შეტევებს (სხვადასხვა შენატანები წარმოქმნიან იმავე ჰეშს).

- Recommendation:** MD5 არ უნდა იქნას გამოყენებული უსაფრთხოების მიმართ მგრძნობიარე აპლიკაციებისთვის, როგორცაა ციფრული ხელმოწერები ან სერტიფიკატების გენერირება.

## 2. SHA-1 (Secure Hash Algorithm 1):

- Original Purpose:** SHA-1, ისევე როგორც MD5, შეიქმნა მთლიანობის გადამოწმებისთვის და აწარმოებდა 160-ბიტის ჰეშის მნიშვნელობას (40 ბიტის სიმბოლო).
- Security Concerns:** SHA-1 მგრძნობიარეა შეჯახების შეტევების მიმართ, სადაც ორი განსხვავებული შეყვანა წარმოქმნის ერთსა და იმავე ჰეშს. ამ დაუცველობამ გამოიწვია მისი გაუქმება უსაფრთხოების მიმართ მგრძნობიარე მიზნებისთვის.
- Recommendation:** SHA-1 თავიდან უნდა იქნას აცილებული კრიპტოგრაფიული მიზნებისთვის, ხოლო SHA-2 ან უფრო ძლიერი ალგორითმები უნდა იქნას გამოყენებული.

## 3. SHA-2 (Secure Hash Algorithm 2):

- Original Purpose:** SHA-2 არის კრიპტოგრაფიული ჰეშის ფუნქციების ოჯახი, მათ შორის SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 და SHA-512/256. SHA-256 და SHA-512 ყველაზე ხშირად გამოყენებული ვარიანტებია.
- Security Features:** SHA-2 ითვლება უსაფრთხოდ და ფართოდ გამოიყენება მონაცემთა მთლიანობის, ციფრული ხელმოწერებისა და სხვა კრიპტოგრაფიული აპლიკაციებისთვის. ის უზრუნველყოფს უსაფრთხოების უფრო მაღალ დონეს MD5 და SHA-1-თან შედარებით.
- Recommendation:** უსაფრთხოების აპლიკაციების უმეტესობისთვის, განსაკუთრებით მონაცემთა მთლიანობისთვის, SHA-256 ან SHA-512 რეკომენდებულია MD5-ზე და SHA-1-ზე.

რაც შეეხება პაროლების შენახვას, გადამწყვეტი მნიშვნელობა აქვს უსაფრთხო და ადაპტირებული ჰეშირების ტექნიკის გამოყენებას მომხმარებლის რწმუნებათა სიგელების არაავტორიზებული წვდომისგან დასაცავად. აქ არის საუკეთესო პრაქტიკა პაროლების უსაფრთხო ჰეშირებისთვის:

## **2. Salt the Passwords:**

გამოიყენეთ უნიკალური შემთხვევითი salt თითოეული პაროლისთვის ჰეშირებამდე. salt არის შემთხვევითი მნიშვნელობა, რომელიც შერწყმულია პაროლთან ჰეშირებამდე, რაც უზრუნველყოფს იმას, რომ მაშინაც კი, თუ ორ მომხმარებელს აქვს იგივე პაროლი, მათი ჰეშირებული მნიშვნელობები განსხვავებული იქნება. ეს ხელს უშლის თავდამსხმელებს, გამოიყენონ წინასწარ გამოთვლილი ცხრილები (ცისარტყელას ცხრილები) პაროლის შეტევებისთვის.

## **3. Use Key Stretching:**

გასაღების გაჭიმვა გულისხმობს ჰეშის ფუნქციის მრავალჯერ გამოყენებას (გამეორებას) ჰეშირების პროცესის შესანელებლად. ეს ამატებს უსაფრთხოების დამატებით ფენას, რაც თავდამსხმელებს უფრო შრომატევადს და რესურსებს ხდის უხეში ძალის ან ლექსიკონის შეტევების განხორციელებას.

## **4. Implement Adaptive Hashing:**

აირჩიეთ ჰეშირების ალგორითმი, რომელიც მხარს უჭერს ადაპტირებულ ტექნიკას, არეგულირებს ჰეშირების პროცესის სირთულეს დროთა განმავლობაში. ეს ხელს უწყობს გამოთვლითი სიმძლავრისა და ტექნოლოგიების მიღწევების გავლენის შემცირებას.



## 1. Use a Cryptographically Secure Hashing Algorithm:

აირჩიეთ ძლიერი და უსაფრთხო ჰეშირების ალგორითმი, რომელიც შექმნილია პაროლის ჰეშირებისთვის. საერთო არჩევანი მოიცავს:

Argon2: რეკომენდებულია ექსპერტების მიერ პაროლის ჰეშირებისთვის, როგორც GPU, ასევე ASIC შეტევებისადმი მისი წინააღმდეგობის გამო.

bcrypt: კარგად ჩამოყალიბებული ალგორითმი კონფიგურირებადი სამუშაო ფაქტორით, რაც მას შესაფერისს გახდის პაროლის ჰეშირებისთვის.

scrypt: შექმნილია მეხსიერების ინტენსიურად, რომელიც უზრუნველყოფს დაცვას გარკვეული ტიპის შეტევებისგან.

## **5. Regularly Update Hashing Algorithms:**

იყავით ინფორმირებული კრიპტოგრაფიისა და უსაფრთხოების უახლესი მოვლენების შესახებ. პერიოდულად შეაფასეთ და განაახლეთ თქვენი პაროლის ჰეშირების ალგორითმები, რათა დარწმუნდეთ, რომ ისინი შეესაბამება მიმდინარე საუკეთესო პრაქტიკას.

## **6. Store Hashed Passwords Securely:**

შეინახეთ მხოლოდ ჰეშირებული პაროლები და დაკავშირებული მარილის მნიშვნელობები. არასოდეს შეინახოთ უბრალო ტექსტის პაროლები ან შექცევადი დაშიფვრის გასაღებები. დარწმუნდით, რომ ჰეშირებული პაროლების შენახვის მექანიზმი თავისთავად დაცულია არაავტორიზებული წვდომის თავიდან ასაცილებლად.

## **7. Keep Pace with Security Standards:**

იყავით განახლებული ინდუსტრიის სტანდარტებისა და პაროლის ჰეშირების რეკომენდაციების შესახებ. იფიქრეთ ისეთი ორგანიზაციებისგან, როგორიცაა NIST (სტანდარტებისა და ტექნოლოგიების ეროვნული ინსტიტუტი) ან OWASP (Open Web Application Security Project) მიმართულებები.

HMAC (Hash-based Message Authentication Code) არის მექანიზმი, რომელიც გამოიყენება შეტყობინების ან მონაცემთა ნაწილის მთლიანობისა და ავთენტურობის დასადასტურებლად. იგი მოიცავს კრიპტოგრაფიულ ჰეშის ფუნქციას და საიდუმლო გასაღებს ჰეშის მნიშვნელობის შესაქმნელად, რომელიც შეიძლება გამოყენებულ იქნას მონაცემთა მთლიანობის შესამოწმებლად. HMAC ჩვეულებრივ გამოიყენება მონაცემთა ბაზებისა და სხვა მგრძნობიარე ინფორმაციის მთლიანობის უზრუნველსაყოფად. აი, როგორ შეგიძლიათ გამოიყენოთ HMAC ავთენტიფიკაციისთვის და უზრუნველყოთ მონაცემთა ბაზების მთლიანობა:

1. მიამაგრეთ HMAC მონაცემებს:

2. დაამატეთ გამოთვლილი HMAC თავდაპირველ მონაცემებს. ეს ქმნის კომბინირებულ შეტყობინებას, რომელიც მოიცავს მონაცემებს და მის HMAC-ს.

3. გაგზავნეთ კომბინირებული შეტყობინება:

4. გაგზავნეთ კომბინირებული შეტყობინება (მონაცემები + HMAC) მიმღებს.

5. გადაამოწმეთ მიღების ბოლოს:

6. მიმღების ბოლოს, ხელახლა გამოთვალეთ HMAC მიღებული მონაცემებისა და საერთო საიდუმლო გასაღების გამოყენებით. შეადარეთ ეს ხელახლა გამოთვლილი HMAC მიღებულ HMAC-თან. თუ ისინი ემთხვევა, მონაცემები ითვლება ავთენტურად და არ არის გაყალბებული.

**HMAC-ის გამოყენება ავთენტიფიკაციისთვის:**

**აირჩიეთ ჰეშის ფუნქცია:**

აირჩიეთ უსაფრთხო კრიპტოგრაფიული ჰეშის ფუნქცია, როგორიცაა **SHA-256** ან **SHA-3**. ჰეშის ფუნქციის არჩევანი დამოკიდებულია თქვენს უსაფრთხოების მოთხოვნებზე.

**შექმენით საიდუმლო გასაღები:**

შექმენით საიდუმლო გასაღები, რომელიც გაზიარებული იქნება გამგზავნს (მონაცემთა ბაზას) და მიმღებს (აპლიკაციას ან მომხმარებელს) შორის. ეს გასაღები კონფიდენციალური უნდა იყოს.

**შექმენით HMAC:**

მონაცემთა თითოეული ნაწილისთვის (მაგ. მონაცემთა ბაზის ჩანაწერი), გამოთვალეთ HMAC შერჩეული ჰეშის ფუნქციისა და საიდუმლო გასაღების გამოყენებით. HMAC-ის ფორმულა არის:

**$$\text{HMAC}(\text{გასაღები}, \text{შეტყობინება}) = \text{ჰეშ}((\text{გასაღები XOR opad}) \parallel \text{ჰეშ}((\text{გასაღები XOR ipad}) \parallel \text{შეტყობინება}))$$**

სად  $\parallel$  აღნიშნავს შეერთებას, opad არის გარე padding, ipad არის შიდა padding და hash არის არჩეული ჰეშის ფუნქცია.

# Ensuring Integrity of Databases:

## 1. Apply HMAC to Database Records:

- 1. მონაცემთა ბაზაში თითოეული ჩანაწერისთვის გამოიყენეთ HMAC პროცესი, როგორც ზემოთ აღწერილი. გამოიყენეთ უნიკალური გასაღები თითოეული ჩანაწერისთვის ან გაზიარებული გასაღები მთელი მონაცემთა ბაზისთვის, თქვენი უსაფრთხოების მოდელიდან გამომდინარე.

## 2. Store HMAC Values:

- 3. შეინახეთ გამოთვლილი HMAC მნიშვნელობები მონაცემთა ბაზის ჩანაწერებთან ერთად. **Verification During Retrieval:**

- 4. მონაცემთა ბაზიდან ჩანაწერების მიღებისას, ხელახლა გამოთვალეთ HMAC თითოეული ჩანაწერისთვის შენახული გასაღების გამოყენებით. შეადარეთ ხელახლა გამოთვლილი HMAC შენახულ HMAC-თან. თუ ისინი ემთხვევა, ჩანაწერის მთლიანობა ხელუხლებელია.

## 5. Protecting Against Tampering:

- 6. თუ თავდამსხმელი შეეცდება მონაცემთა ბაზის ხელყოფას ჩანაწერების შეცვლით, HMAC ვერიფიკაცია ვერ მოხერხდება, რაც მიუთითებს, რომ მონაცემები გატეხილია.
- 7. HMAC-ის გამოყენებით, თქვენ შეგიძლიათ უზრუნველყოთ მონაცემთა მთლიანობის ძლიერი გარანტია, იმის უზრუნველსაყოფად, რომ მონაცემთა ბაზის ჩანაწერები არ არის შეცვლილი ან გაყალბებული. ეს განსაკუთრებით მნიშვნელოვანია აპლიკაციებისთვის, რომლებიც საჭიროებენ სენსიტიური ინფორმაციის უსაფრთხო და სანდო შენახვას.