

# 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, გამომიება 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 გამოსაძიებლად. 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
6. 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.

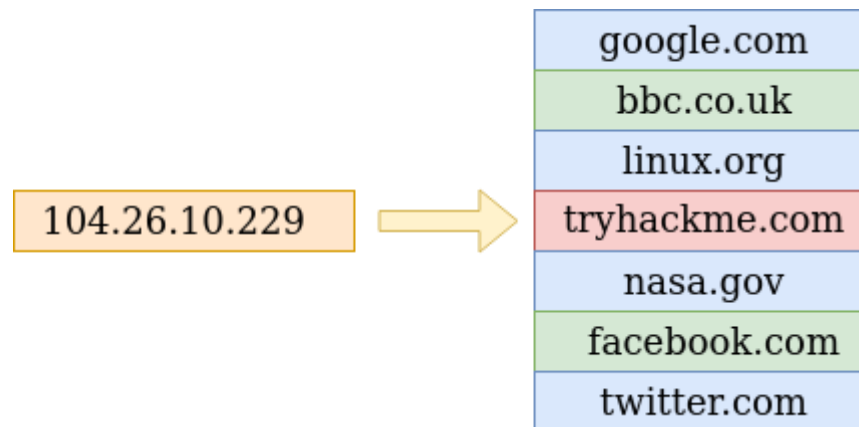
აქტიური FTP კავშირში კლიენტი ხსნის პორტს და უსმენს. სერვერს მოეთხოვება მასთან აქტიური დაკავშირება.

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

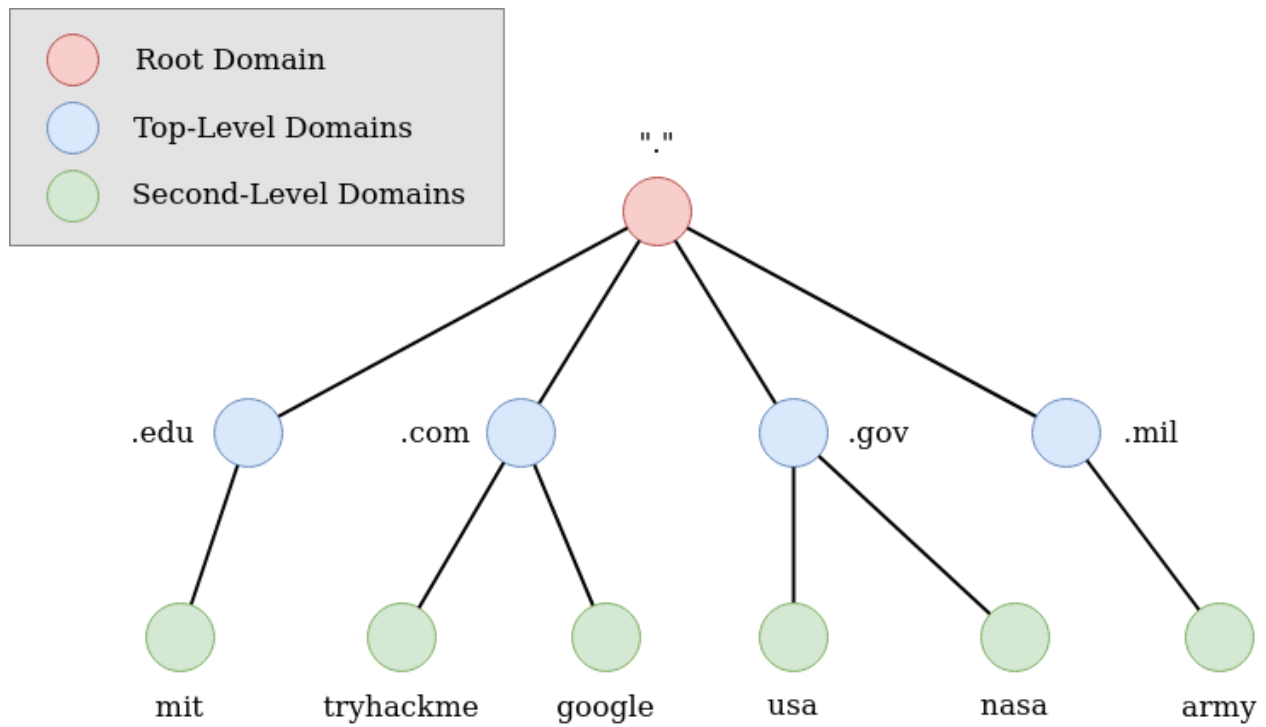
ბრძანების ინფორმაციისა და მონაცემების ცალკეულ არხებად დაყოფა არის გზა სერვერზე ბრძანებების გაგზავნის გარეშე მონაცემთა მიმდინარე გადაცემის დასრულებამდე ლოდინის გარეშე. თუ ორივე არხი ურთიერთდაკავშირებულია, თქვენ შეგეძლოთ შეიყვანოთ ბრძანებები მხოლოდ მონაცემთა გადაცემებს შორის, რაც არ იქნება ეფექტური არც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



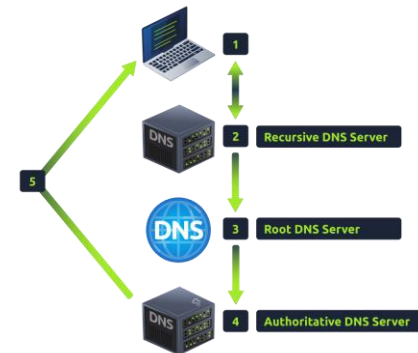
# Domain Hierarchy





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:

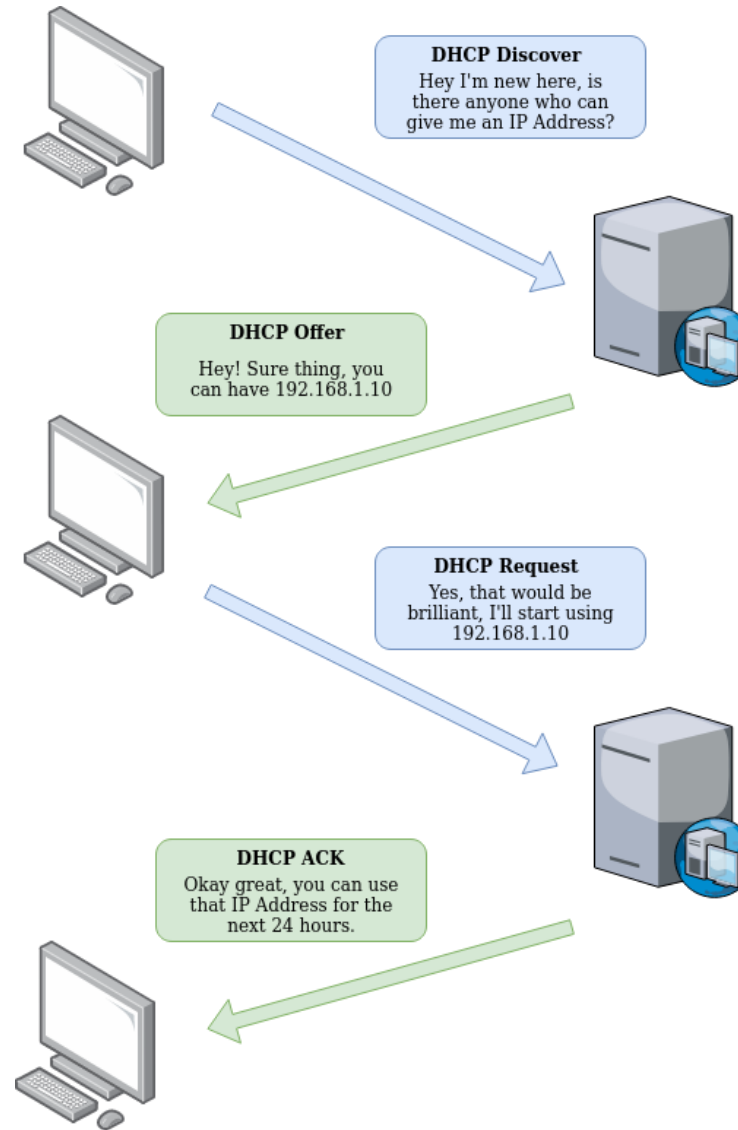
#### ARP მოთხოვნა

#### 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.

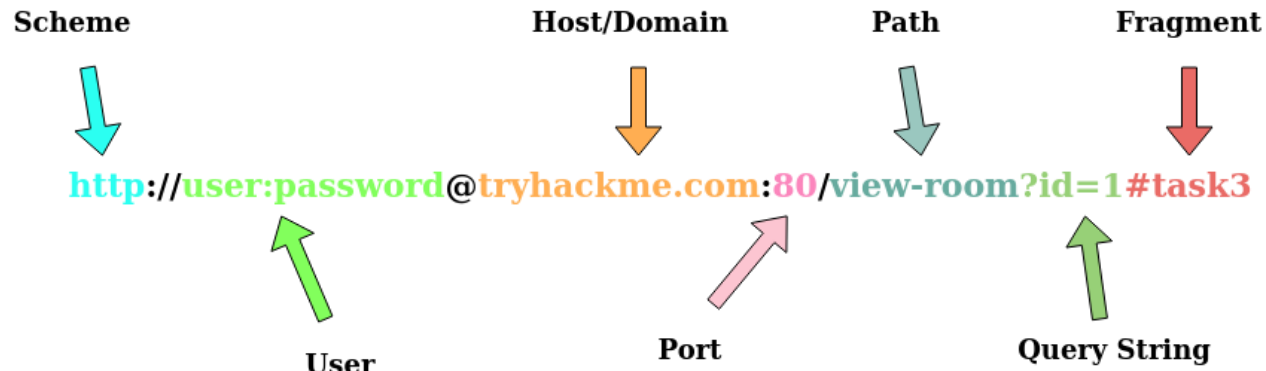
**რა არის HTTPS? (ჰიპერტექსტის გადაცემის პროტოკოლი უსაფრთხო)**

**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, როგორიცაა 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 გახადა @ 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

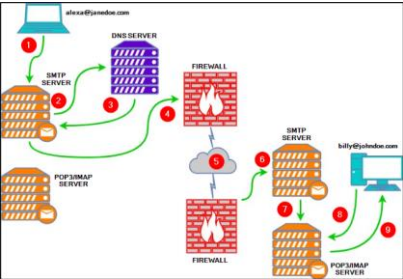
ელ.წერილები ინახება სერვერზე და მათი ჩამოტვირთვა შესაძლებელია მრავალ მოწყობილობაზე.

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.