ივ. ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტის

ზუსტ და საბუნებისმეტყველო ფაკულტეტი

ბერუაშვილი სერგო

ქსელში არსებული საფრთხეები (სნიფერების სახით) და მათგან თავის დაცვის გზები

ინფორმაციული ტექნოლოგიები

ნაშრომი შესრულებულია **ინფორმაციული ტექნოლოგიების მაგისტრის**

აკადემიური ხარისხის მოსაპოვებლად

თბილისი 2014

სამაგისტრო ნაშრომის ხელმძღვანელი

ლელა მირცხულავა

დოქტორი, ასოცირებული პროფესორი

შინაარსი

ანოტაცია

ტექნოლოგიის განვითარებასთან ერთად , მკვეთრად იზრდება ქსელში ჩართული მოწყობილობების რაოდენობა. ქსელისა და კომპიუტერის მნიშვნელობა დღითიდღე იზრდება , მათი მეშვეობით ხდება სხვადასხვა ტიპის ინფორმაციის გადაცემა (პირადი,საიდუმლო,დიპლომატიური და ა.შ.) ამიტომ აუცილებელია სანდო/საიმედო კავშირი მოწყობილობებს შორის , რათა არ მოხდეს ინფორმაციის მოპარვა/მიტაცება.

სამაგისტრო ნაშრომის ამოცანა არის განვიხილოთ ქსელიდან ინფორმაციის “მოპარვის” ერთ-ერთი მეთოდი , რომელსაც ARP Spoofing ეწოდება. მისი არსი მდომარეობს იმაში რომ , ხდება ქსელში ბევრი , ცრუ ARP მოთხოვნების დაგზავნა, “შემტევი” აგზავნის ქსელში ინფორმაციას იმის შესახებ , რომ “მსხვერპლისათვის” განკუთვნილი პაკეტები მივიდეს მასთან და ასეც ხდება. სამაგისტრო ნაშრომში მოყვანილია დეტალური აღწერა ამ პროცესისა და მისგან თავის დაცვის მიდგომები , ასევე პროგრამა რომელიც თავად აღმოაჩენს ქსელში არსებულ საფრთხეს.

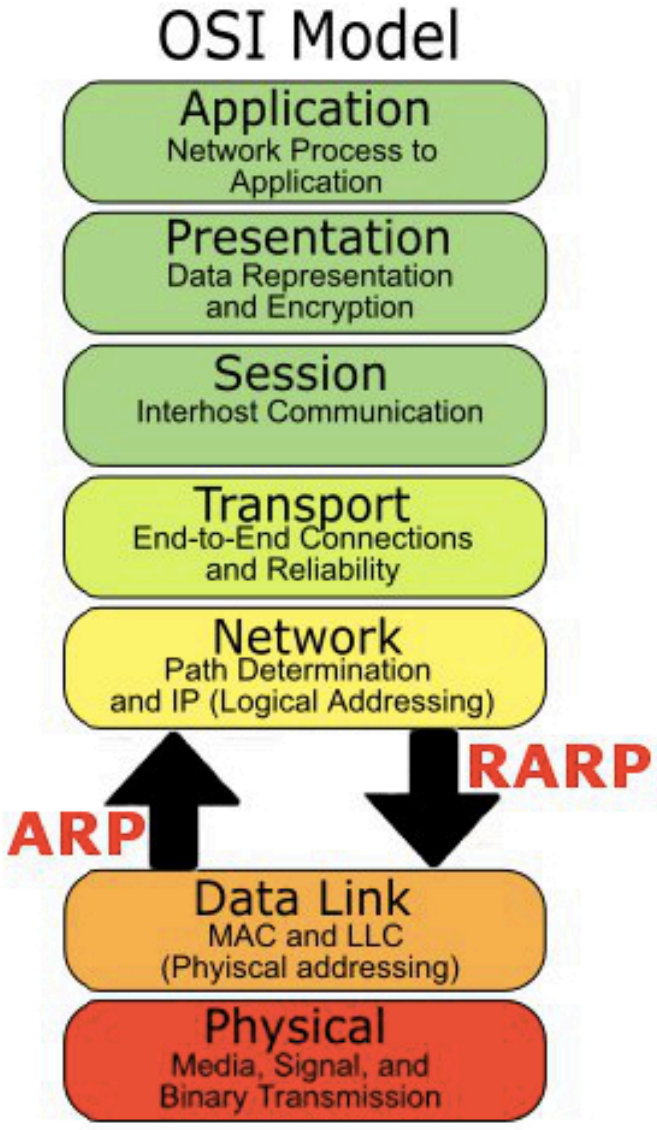
Annotation

The number of devices in network drastically grows as Technology Development. Importance of Computers and Network is getting bigger and bigger. They are used to transfer all kind of information (Personal, Secret, Diplomatic … etc), it is important to have a reliable connection between devices, to protect information and not let them leak.

The goal of this work is to discuss about one of the network-hijacking techniques “ARP Spoofing”. Which produces lots of false “ARP Reply” packet in network, it allows attacker to sniff network traffics and steal them. The whole process of this attack is described in the given work as well as defending techniques. This work also contains software, which will detect ARP Spoofing threats in the network.

რა არის MAC

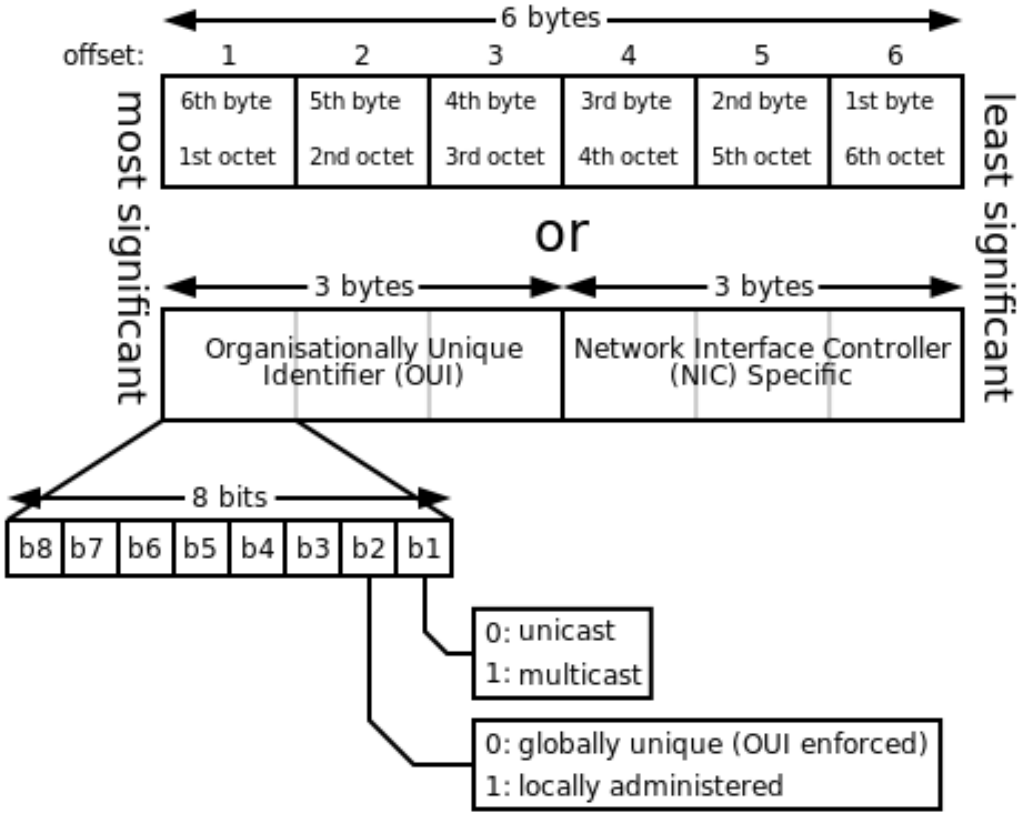
MAC იშიფრება როგორც Media Access Control , იგი არის ქსელში ჩართული მოწყობილობის ფიზიკური მისამართი , ქსელში კომპიუტერის იდენტიფიცირება ხდება სწორედ MAC -ის მეშვეობით , ანუ რომ გავუგზავნოთ რომელიმე მოწყობილობას ქსელში ინფორმაცია , უნდა ვიცოდეთ მისი MAC მისამართი (საუბარია Layer 2 დონეზე) .



MAC არის 48-ბიტიანი მნიშვნელობა , რომლის პირველი 24 ბიტი განსაზღვრავს მწარმოებელს , ხოლო ბოლო 24 ბიტი განსაზღვრავს უშუალო იდენტიფიკატორს ამა თუ იმ ქსელის ბარათისა. MAC არ უნდა მეორდებოდეს მოწყობილობებს შორის.

თუ ერთსა და იმავე ქსელში ორ მოწყობილობას ექნებათ ერთნაირი MAC მისამართი , მაშინ ამ ქსელში იქნება პრობლემები.

სურათზე მოცემულია MAC მისამართის აგებულება



რა არის ARP

ARP პროტოკოლი გამოიყენება ლოკალურ ქსელში ცნობილი IP პროტოკოლის მისამართის მიხედვით, იმ მოწყობილობის მაკ მისამართის გასაგებად, რომელსაც ეს IP გააჩნია. მისი არსებობა გამოწვეულია იმით, რომ IPv4 over Ethernet ქსელებში ხშირად აქვს ადგილი სიტუაციას, როცა ინფორმაციის გადამცემმა იცის IP მისამართი მაგრამ არ იცის მისი ფიზიკური მისამართი,ხოლო ლოკალურ ქსელში ინფორმაციის გადასაცემად აუცილებელია მაკ მისამართის ცოდნა.

ARP პროტოკოლი მოქმედების არეალი შემოისაზღვრება იმ ლოკალური ლინკით, რომელზეც უშუალოდ ჰოსთია დაერთებული. პროტოკოლის მუშაობა ხდება კითხვა - პასუხის (request - answer) რეჟიმში. ერთი ჰოსთი კითხულობს „ვის აქვს X აიპი მისამართი ?“ და ვისაც აქვს პასუხობს „X აიპი მისმართი აქვს Y mac-მისამართს“, ქვემოთ მოყვანილია ლოგი რეალური ქსელიდან, რომელშიც სულ ორი წევრია: 192.168.1.1 (Planet ფირმის მოდემი) და 192.168.1.2 (კომპიუტერი Foxconn-ფირმის დედაპლატით, ან ქსელის კარტით):

No.	Time	Source	Destination	Protocol	Info
1	0.000000	PlanetTe_6e:68:2e	Foxconn_8b:92:0b	ARP	who has 192.168.1.2? Tell 192.168.1.1
2	0.000017	Foxconn_8b:92:0b	PlanetTe_6e:68:2e	ARP	192.168.1.2 is at 00:15:58:8b:92:0b
3	49.710942	PlanetTe_6e:68:2e	Foxconn_8b:92:0b	ARP	who has 192.168.1.2? Tell 192.168.1.1
4	49.710956	Foxconn_8b:92:0b	PlanetTe_6e:68:2e	ARP	192.168.1.2 is at 00:15:58:8b:92:0b
5	109.764182	PlanetTe_6e:68:2e	Foxconn_8b:92:0b	ARP	who has 192.168.1.2? Tell 192.168.1.1
6	109.764199	Foxconn_8b:92:0b	PlanetTe_6e:68:2e	ARP	192.168.1.2 is at 00:15:58:8b:92:0b

ლოგში კარგად ჩანს რომ 109 წამის განმავლობაში, ერთიდაიგივე ჰოსთი 3-ჯერ კითხულობს თუ ვის აქვს ერთიდაიგივე მისამართი.

მოწყობილობები ქმნიან ARP ჩანაწერების ბაზას, რომელსაც ARP ჰეში ქვია, ანუ აქ წერია რომელ IP-ს რომელი MAC შეესაბამება მოცემულ ლოკალურ ქსელში.

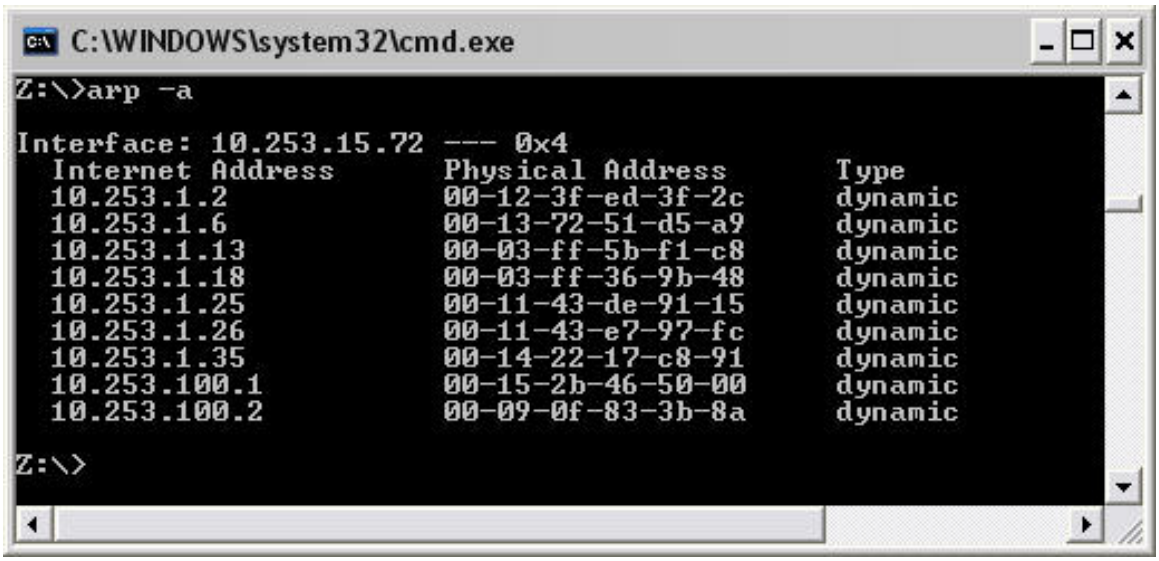
ჩანაწერი შეიძლება იყოს დინამიური (მოპოვებული ARP პროტოკოლის საშუალებებით და სტატიკური: ადმინისტრატორის მიერ ხელით შეყვანილი). სტატიკური ჩანაწერები ARP-ჰეშში მუდმივად რჩება (სანამ ისევ არ წაშლიან ბრძანებით), ხოლო დინამიური 2 წუთის განმავლობაში, თუ ამ 2 წუთის განმავლობაში ამ ჩანაწერზე კიდევ მოვიდა ინფო მისი არსებობა ჰეშში კიდევ 2 წუთით გახანგრძლივდება, თუ ჩანაწერს ჰეშში 10 წუთი „შეუსრულდა“ იგი აუცილებლად წაიშლება და ახალი აფდეითის შემთხვევაში თავიდან გაკეთდება.

როდესაც ჰოსთს სჭირდება გაიგოს ესა თუ ის IP მისამართი ვინმეს ხომ არ აქვს (მაგ IP კონფლიქტის თავიდან ასარიდებლად, ან სკანირების მიზნით ☺), იგი აგზავნის ARP მოთხოვნას სადაც მისი მისმართის მაგივრად 0-ები წერია, ასეთ მოთხოვნას ARP probe ქვია.

როდესაც ჰოსთის MAC მისამართი იცვლება, ან ჰოსთი ირთვება ქსელში იგი აბროადქასტებს ე.წ. ARP announcement, ანუ ყველას აგებინებს რომ „X აიპი მისამართი აქვს Y მაკ მისამართს“.

არსებობს კიდევ Inverse ARP და Reverse ARP პროტოკოლები, რომელთაც აქვთ შებრუნებული ამოცანა: მოცემული მაკ მისამართის მიხედვით დადგინდეს IP ისამართი. Inverse ARP გამოიყენება ATM და Frame Relay ქსელებში. ხოლო Reverse ARP კი ეზერნეტში გამოიყენებოდა, ამ წუთისთვის მისი ფუნქცია სრულად შეითავსა DHCP პროტოკოლმა.

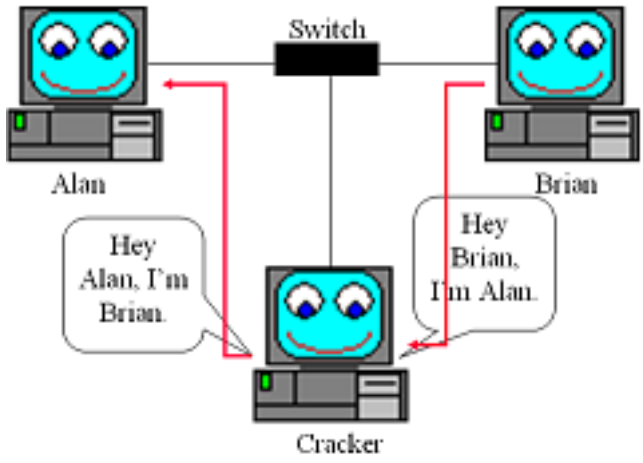
სურათზე მოცემულია ARP ჩანაწერების ცხრილი



Man in the middle

როგორც ვიცით , ქსელში ინფორმაციის გადასაცემად საჭიროა გამგზავნი (source) და მიმღები (destination) , გამგავნი მიმღებს გადასცემს ინფორმაციას ქსელის საშუალებით , თუმცა არის ისეთი შემთხვევებიც , როდესაც ინფორმაციას “გზაში” მესამე პირი ეცნობა და ასე მიდის მიმღებამდე, შესაბამისად ხდება ინფორმაციის მითვისება. განვიხილოთ მაგალითი : დავუშვათ ინფორმაციის გამგზავნი არის *სოფო* ხოლო მიმღები *გურამი*. ასევე გვყავს მესამე პიროვნება *სანდრო* რომელსაც უნდა ამ ინფორმაციის ნახვა. იმისათვის რომ სოფომ გურამს გაუგზავნოს ინფორმაცია , საჭიროა იცოდეს მისი მისამართი , ასევე , სოფოს მისამართიც უნდა იცოდეს გურამმა რათა მოახდინოს მასზე პასუხის გაცემა,ხოლო მათ თუ ერთმანეთის ზუსტი მისამართი ეცოდინებათ , სანდრო ვეღარ მოახდენს ინფორმაციის მითვისებას.ამის საპირისპიროდ , სანდრო უკავშირდება სოფოს როგორც გურამი , ხოლო უკავშირდება გურამს როგორც სოფო , შესაბამისად სოფოს ჰგონია რომ იმისათვის რომ გურამს გაუგზავნოს ინფორმაცია საჭიროა მისთვის ცნობილ მისამართზე გააგზავნოს ეს ინფორმაცია (სწორედ იმ მისამართზე რომელიც სანდრომ მისცა) და პირიქით , როდესაც გურამს ეს ინფორმაცია მიუვა , მას ეგონება რომ ეს სოფომ გაუგზავნა , შესაბამისად უპასუხებს იმ მისამართზე რომელიც მას სანდრომ მისცა. საბოლოოდ სოფოსა და გურამს შორის შედგება ინფორმაციის გაცვლა-გამოცვლა , თუმცა სანდროს გავლით , სოფო და გურამი შესაძლოა სერთოდ ვერ მიხვდნენ იმას რომ მათი ინფორმაციის გაცვლა პირდაპირ არ ხდება .

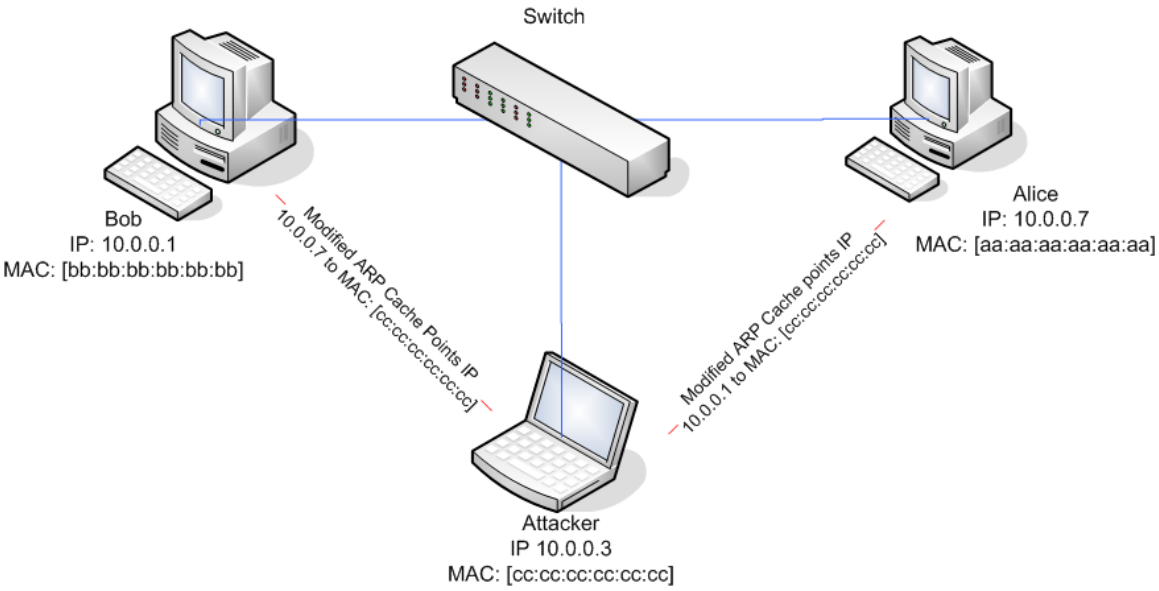
ასეთ მიდგომას ეწოდება Man in the middle , რომლის გამოყენებასაც ფართო ადგილი აქვს ქსელებში ინფორმაციის მოსაპოვებლად.



რა არის ARP Spoofing

ARP Spoofing არის Man in the middle მეთოდის განხორციელება კომპიუტერულ ქსელში. როგორც ვიცით , იმისათვის რომ ქსელში ინფორმაციის გაცვლა მოხდეს საჭიროა მოწყობილობებმა იცოდნენ ერთმანეთის MAC მისამართი , ხოლო MAC მისამართს იგებენ ARP პროტოკოლის დახმარებით , შესაბამისად , ARP Spoofing-ის დროს , ხდება ცრუ შეტყობინებების დაგზავნა მთელს ქსელში (Broadcast) , ამიტომ კომპიუტერებს აქვთ არასწორი მაკ მისამართი და ინფორმაციას გზავნიან “თავდამსხმელთან” , განვიხილოთ უფრო დეტალურად .

სურათზე მოცემულია ARP Spoofing-ის მაგალითი.

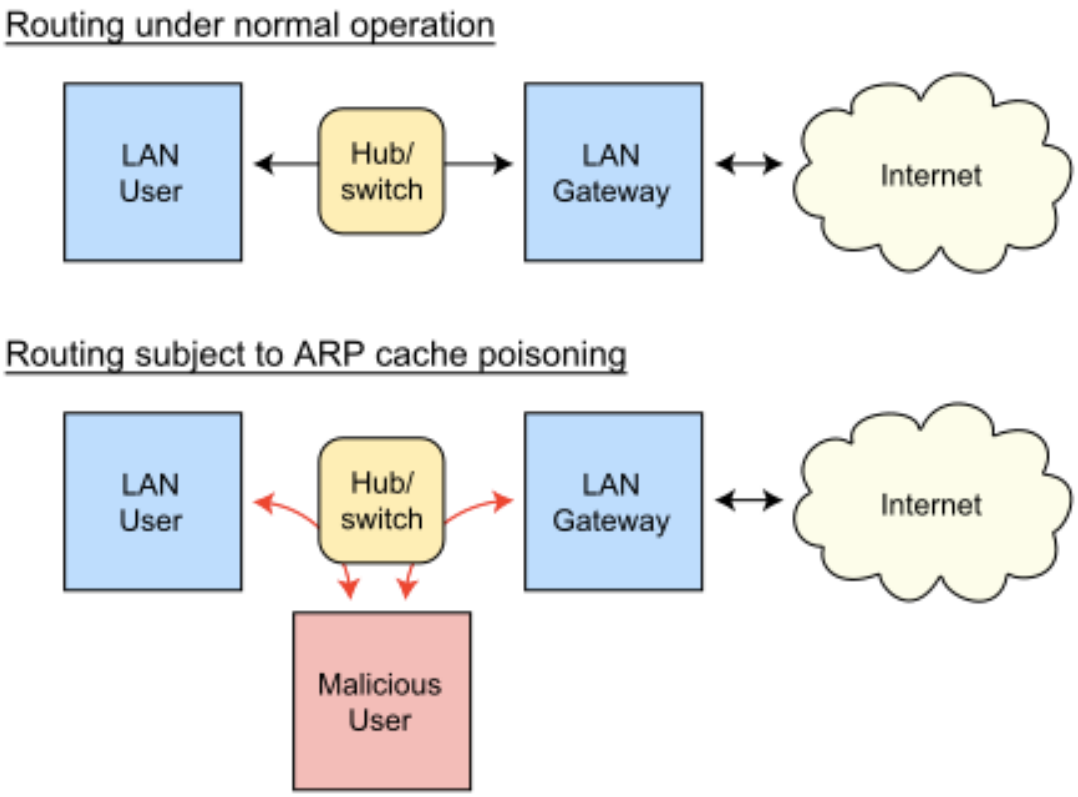


ARP Spoofing-ის დაწყებისას , პირველ რიგში უნდა შეირჩეს “მსხვერპლი” , ეს შეიძლება იყოს როგორც რომელიმე კონკრეტული მოწყობილობა ასევე მთელი ქსელი , მთელი ქსელის შემთხვევაში , როგორც წესი “მსხვერპლი” არის Gateway . მას შემდეგ რაც ამოვირჩევთ მსხვერპლს,უნდა გავიგოთ მისი IP მისამართი , IP მისამართის გაგების შემდეგ იწყება ქსელის “დაბინძურება” , ეს პროცესი მოიცავს ქსელში ინფორმაციის დაგზავნას იმის შესახებ რომ მოცემული IP მისამართი მდებარეობს “თავდამსხმელის” MAC მისამართზე . ხდება ასეულობით ასეთი პაკეტების ქსელში დაგზავნა , შესაბამისად

როდესაც სხვა კომპიუტერი შეეცდება გაიგოს ARP ის საშუალებით თუ რომელიმე IP მისამართი რომელ MAC-ს ეკუთვნის რათა გადასცეს ინფორმაცია , მას მიუვა ცრუ პასუხი , რომელიც დაგენერირებულია “თავდამსხმელის” მიერ. ამის შემდეგ კომპიუტერი დაიწყებს ინფორმაციის გადაცემას თავდამსხმელის მისამართით .

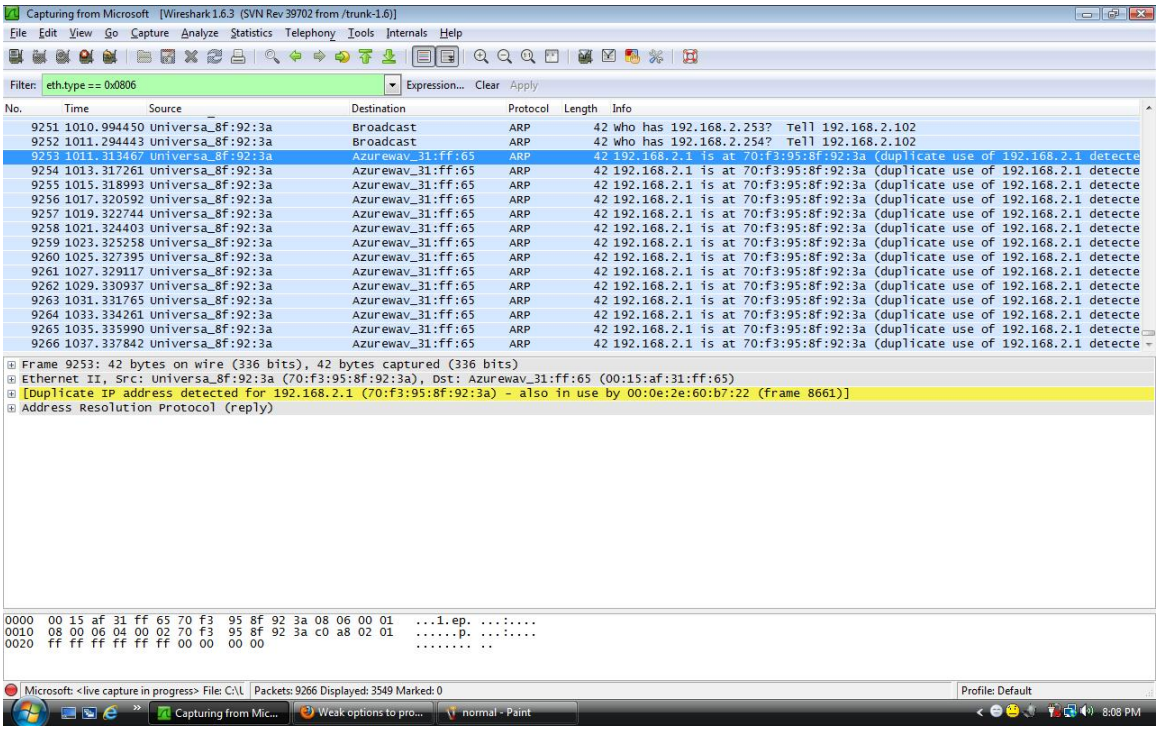
აუცილებელია ასეთ მომენტში თავდამსხმელმა ინფორმაციის მიღების და ჩაწერის შემდგომ , იგივე პაკეტი გადაუგზავნოს ნამდვილ ადრესატს , წინააღმდგ შემთხვევაში ასეთი თავდამსხმელი ადვილი აღმოსაჩენი იქნება.

სურათზე ნაჩვენებია თუ როგორ ხდება ინფორმაცია ჩვეულებრივ რეჟიმში და ARP Spoofing ის დროს



ქსელის ანალიზი ARP Spoofing-ის დასადგენად

იმისათვის რომ დავადგინოთ ხდება თუ არა ქსელში ARP Poisoning/ARP Spoofing საჭიროა მისი ანალიზი. ქსელის ანალიზისათვის ერთ-ერთი ყველაზე პოპულარული პროგრამა არის Wireshark. მისი დახმარებით შესაძლებელია ყველა იმ პაკეტის გახსნა/გაანალიზება რომელიც მიეწოდება ან რომელსაც გადასცემს კონკრეტული ქსელის ბარათი , პროგრამის ჩართვისას მომხმარებელი ირჩევს თუ რომელი ქსელის ბარათიდან აიღოს მან ინფორმაცია , ასევე მას შეუძლია პაკეტების გაფილტვრა მისი ტიპისა სხვადასხვა მახასიათებლების მიხედვით , ჩვენს შემთხვევაში , გვაინტერესებს ARP პროტოკოლის პაკეტები .



სურათზე ნაჩვენებია Wireshark-ით გაფილტრული ARP პაკეტები ARP Spoofing-ის პროცესში , როგორც ვხედავთ იგი გვაძლევს იმის ცნობას რომ 192.168.2.1 IP-ზე ხდება 2 სხვადასხვა MAC მისამართიდან პასუხის მოსვლა , რომ ეს IP მისამართი მას ეკუთვნის.გარდა ამისა , ასეთი მიყოლებით ARP პაკეტები , რომლებშიც მხოლოდ ARP

Response არის მოცემული ძალზედ საეჭვოა , რადგანაც ARP მუშაობს კითხვა-პასუხის რეჟიმში , ხოლო რომელიღაც მოწყობილობა თავად აგზავნის ქსელში მხოლოდ პასუხს , შეგვიძლია თამამად ვთქვათ რომ ამ ქსელში ყველაფერი რიგზე არ არის. თუმცა ეს მიდგომა ARP Spoofing-ის დასადგენად არ არის იდეალური და 100% საიმედო , შესაბამის მიდგომებს უფრო ვრცლად განვიხილავთ.

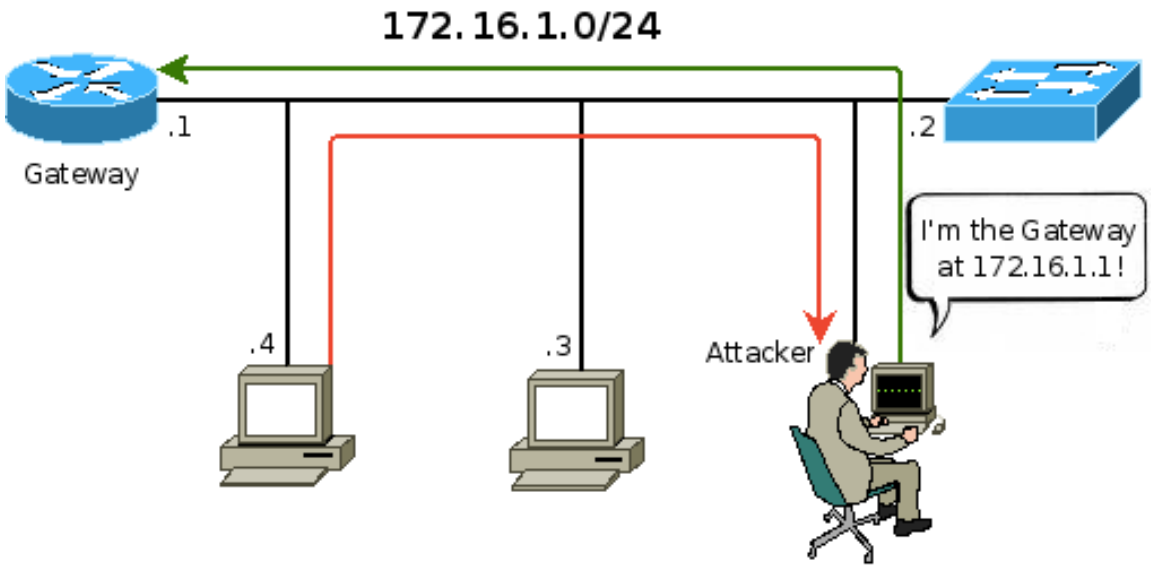
ARP Poisoning -ისას რადგანაც ხდება ბევრი პაკეტის დაგზავნა , ეს იწვევს ქსელის შედარებით ნელა მუშაობას , რადგანაც ქსელში არსებულ მოწყობილობებს აქვთ ზედმეტი ინფორმაცია გადასამუშავებელი , ამან შეიძლება ხანდახან სერიოზული პრობლემები შეუქმნას ქსელის საერთოდ მუშაობასაც კი. (მაგ: ერთ-ერთ მსხვილ ქართულ კომპანიის შიდა ქსელში , სადაც 100-200 კომპიუტერი იყო ჩართული, ერთ-ერთ კომპიუტერს ჰქონდა ვირუსი , რომელიც ახდენდა ARP Poisoning-ს , ამან ისე დატვირთა მთელი ქსელი , რომ ვერანაირი ინფორმაციის გაცვლა გამოცვლა ვეღარ ხდებოდა მისი საშუალებით , მთელი სისტემა გაითიშა ...) .

განსაკუთრებით შესამჩნევია ასეთი მუშაობა Realtime პროგრამებისათვის , რომელთათვისაც რაიმე პაკეტის 0.5 წამით დაგვიანებას დიდი მნიშვნელობა აქვს , ასეთი შეიძლება იყოს ინტერნეტ ტელეფონი , ვიდეო კონფერენცია , რაიმე ონლაინ სტრიმი და სხვა. ხშირად ღია და დიდ ქსელებში მთავარი პრობლემა სწორედ ქსელის დაბინძურებაა , სადაც ასეთ პრობლემასთან გვაქვს საქმე.

ARP Poisoning-ით ასევე შესაძლებელია DoS (Denial of Service) ტიპის თავდასხმა, რაც გულისხმობს იმას , რომ თავდამსხმელი , არ გადაამისამართებს პაკეტებს რეალურ ადრესატთან , რის შემდეგაც სერვისის მიწოდება შეიზღუდება.

http://hakipedia.com/ საიტზე განხილული და დახასიათებულია სხვადასხვა “თავდასხმები” . რაღა თქმა უნდა განხილულ თავდასხმათა შორის არის ARP Poisoning , მას აქვს მინიჭებული შემდეგი მახასიათებლები : სიმწვავე - მაღალი , თავდასხმის შესაძლებლობა - მაღალი.

პროგრამული უზრუნველყოფა ARP Spoofing-ისგან თავდასაცავად



როგორც ვნახეთ , ARP Spoofing ძალზედ საშიში საფრთხეა ქსელში , სამწუხაროდ დღევანდელ ქსელურ მოწყობილობების უმრავლესობას არ გააჩნიათ არანაირი სტანდარტული დაცვის მექანიზმი , რომელიც შეიძლება გამოიყენოს მომხმარებელმა თავდასაცავად.

ეს პრობლემა განსაკუთრებით აქტუალურია ღია WIFI-ების შემთხვევაში, ასეთ მომენტში რიგით გამვლელს შეუძლია ამ ქსელში ჩართული მოწყობილობებისგან მთელი ინფორმაციის წაღება, მაგალითად შეგვიძლია ავიღოთ Tbilisi Loves You-ს უსადენო ქსელი , ან თუნდაც თბილისის სახელმწიფო უნივერსიტეტის უსადენო ქსელი. ნებისმიერ გამვლელს , შეუძლია განახორციელოს ARP შეტევა , მიითვისოს ინფორმაცია , დააბინძუროს ქსელი და ა.შ. სამწუხაროდ აქაც დაცვა მოიკოჭლებს და არანაირი თავდაცვა არ არსებობს ასეთი შემთხვევებისგან.

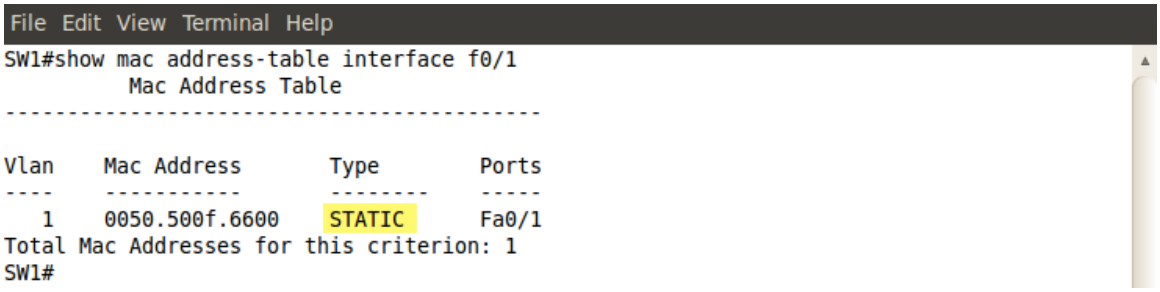
ARP შეტევისგან დასაცავად შეიძლება გამოვყოთ რამდენიმე მიდგომა :

1. S-ARP (Secure ARP)

S-ARP შეიქმნა ARP -ის ჩასანაცვლებლად , იგი იყენებს DSA ალგორითმს რათა გამოიანგარიშოს “signature” თუ რომელი mac მისამართი რომელ ip მისამართს ეკუთვნის , თუმცა მისი “მინუსი” მდგომარეობს იმაში , რომ საჭიროა ქსელში არსებულ ყველა მოწყობილობას ჰქონდეს S-ARP ის მხარდაჭერა , რაც დაკავშირებულია დიდ დროსა და ხარჯებთან , რადგან ქსელური მოწყობილობების მწარმოებლებმა უნდა ჩადონ ოპერაციულ სისტემაში ასეთი პროტოკოლი , რომელიც სამწუხაროდ ჯერ-ჯერობით არ არის იმპლემენტირებული.

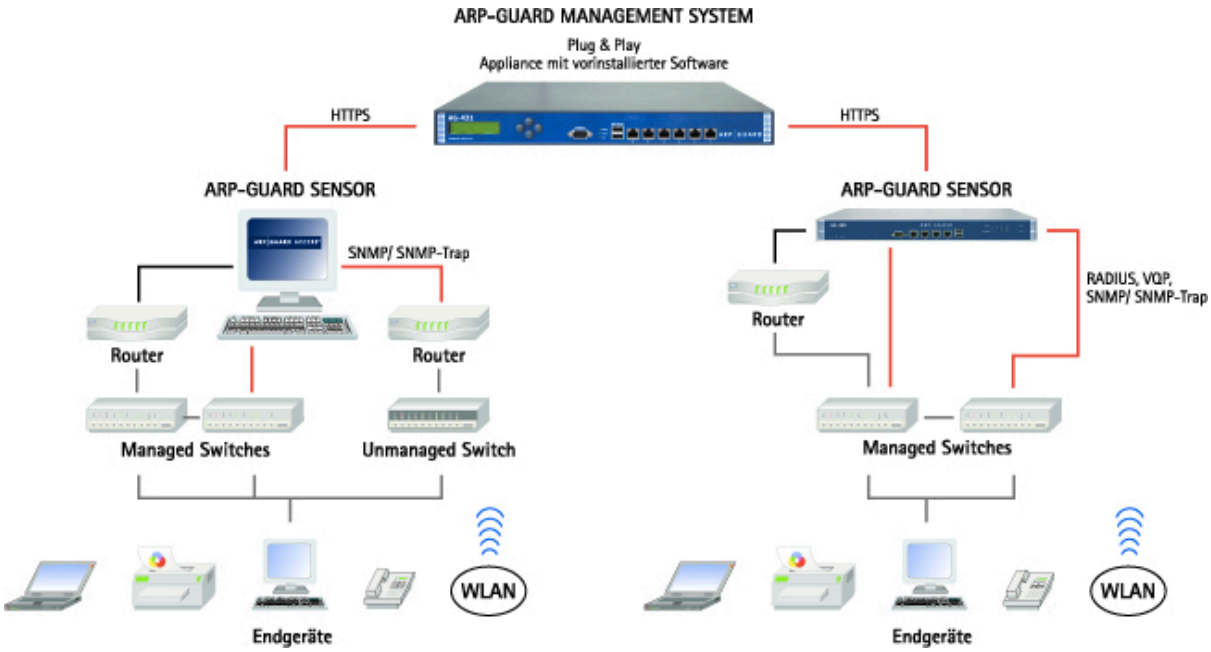
2. სტატიკური MAC ჩანაწერები

თუ ჩვენ წინასწარ გვეცოდინება ჩვენს ქსელში არსებული მოწყობილობების რაოდენობა , IP მისამართები და MAC მისამართები , მაშინ შესაძლებელია წინასწარ შედგეს MAC ჩანაწერების ცხრილი , რომლის მიხედვითაც მოწყობილობა აღარ გააგზავნის ARP მოთხოვნას რათა გაიგოს რომელი MAC მისამართი რომელ IP მისამართს ეკუთვნის , თუმცა დღევანდელ რეალობაში ეს ძალზედ რთულია , ეს ზღუდავს ქსელში ახალი მოწყობილობის დამატებას , ასეთ შემთხვევაში ყველა ჩართულ მოწყობილობაზე უნდა განახლდეს ახალი ჩანაწერი , ასევე ქსელიდან მოწყობილობის გასვლას ან მისი სხვით ჩანაცვლებას .



3. ARPWatch , ARPon და სხვა

ამ საფრთხის თავიდან ასაცილებლად ბევრ ინჟინერსა და პროგრამისტს უფიქრია და შესაბამისად არსებობს რამდენიმე გავრცელებული პროგრამა .



თავდაპირველად ასეთი პროგრამები ითვლიდნენ ARP პაკეტების რაოდენობას , თუ რამდენად ხშირად ხდება ერთი და იმავე მოწყობილობისგან ARP პასუხი , რადგანაც თავდამსხმელი სწორედ ARP პასუხის საშუალებით ანაგვიანებს ქსელს , თუმცა დროთა განმავლობაში , თავდამსხმელები “დაჭკვიანდნენ” , ისინი ცდილობენ აღარ დაანაგვიანონ ქსელი , რადგანაც ასეთ შემთხვევაში უფრო ადვილად ამოსაცნებები არიან , შესაძლოა რომელიმე ქსელს რაიმე შეზღუდვაც კი ჰქონდეს დადებული ქსელის დაბინძურების ასაცილებლად , ხოლო ასეთ შემთხვევაში კი თავდასხმა აღარ განხორციელდებოდა , შესაბამისად ეს მიდგომა თავდამსხმელის ამოცნობისთვის მალევე გახდა უიმედო.

ARP Poisoning-ისგან თავდაცვას სთავაზობს რამდენიმე ანტივირუსიც საკუთარ მომხმარებელს , თუმცა ანტივირუსები ყველა პლატფორმაზე არ არის გათვლილი და არც მათი ARP თავდამსხმელის აღმოჩენის ტექნიკა არის საუკეთესო ,

სამაგისტრო ნაშრომის წერისას ARP Spoofing შეტევა ვერ ამოიცნო ვერცერთმა ნაცადმა ანვიტივურსმა (Kaspersky,AVG).

თუმცა მათი უმეტესობა იყენებს ეგრედ წოდებულ Passive scan-ს , რაც მდგომარეობს იმაში , რომ პროგრამა ჩართვის შემდეგ , აანალიზებს მასთან მისულ პაკეტებს , ნახულობს ARP Reply პაკეტებს , არსებულ ქეშს ადარებს , თუ არ აქვს ეს ჩანაწერი ამატებს , ხოლო თუ შეცვლილია მაშინ პირდაპირ აღიქვამს როგორც საფრთხედ , თუმცა ეს ასე არ არის , შესაძოა ქსელში არაერთი მიზეზის გამო რომელიმე კომპიუტერს IP მისამართი შეეცვალოს , ასეთ შემთხვევაში ეს კომპიუტერი აღიქმება როგორც თავდამსხმელი ასეთი პროგრამების მიერ .
ასეთ პროგრამებს აქვთ კიდევ უფრო დიდი “მინუსი” , ის რომ , თუ თავდამსხმელი არის ქსელში და ახდენს “მოსმენას” მანამ , სანამ პროგრამა ჩაირთვებოდა , მაშინ ეს პროგრამა სწორედ თავდამსხმელის მისამართს ჩაიწერს როგორც “ორიგინალს” ხოლო იმ შემთხვევაში თუ აღმოაჩენს რეალურ მფლობელს , ეგონება თავდამსხმელი. ზემოთ მოყვანილია მაგალითი , თუ როგორ შეიძლება Wireshark-ის მეშვეობით ადმინისტრატორმა გაანალიზოს ქსელი და ნახოს ხომ არ არის რაიმე საფრთხე , თუმცა იქაც იგივე პრობლემასთან მივდივართ , თუ Wireshark (ან რაიმე პროგრამა რომელიც აანალიზებს ქსელს) ჩაირთვება მას შემდეგ რაც თავდამსხმელი უკვე უსმენს ქსელს , ვერ მოხერხდება მისი აღმოჩენა , ამიტომ ასეთი მიდგომა არასაიმედოა. პასიური მოსმენისა და ანალიზის მიდგომა საჭიროებს დახვეწას.

ამ ყველაფრის გათვალისწინებით , საჭიროა ისეთი პროგრამა , რომელიც იქნება საიმედო , რომელიც არ იქნება დამოკიდებული იმაზე თუ როდის იქნება ჩართული (თავდამსხმელის ქსელის მოსმენამდე თუ მას შემდეგ) , მან უნდა გაანალიზოს ქსელი , ასევე მოახდინოს ქსელში საკუთარი პაკეტების გაგზავნა და გადამოწმება ARP ჩანაწერებისა , ARP Spoofing-ის საფრთხის აღმოჩენის შემთხვევაში აცნობებს მომხმარებელს, ქსელის ადმინისტრატორს ან საერთოდაც მოახდენს თავდამსხმელის ქსელიდან დაბლოკვას.

ჩვენი მიდგომა ARP Spoofing-ის აღმოსაჩენად

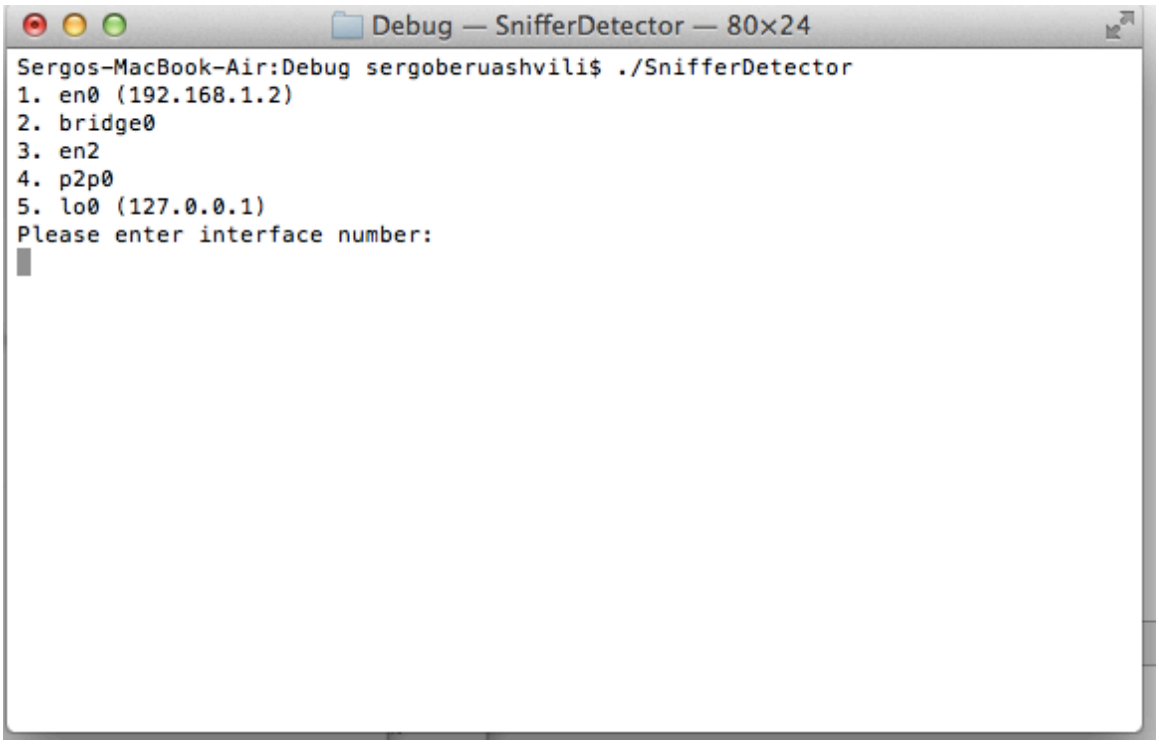
როგორც უკვე ავღნიშნეთ , საჭიროა პროგრამული უზრუნველყოფა რომელიც გააადვილებს ქსელში საფრთხეების აღმოჩენას. სამაგისტრო ნაშრომად წარმოდგენილია სწორედ ასეთი პროგრამა.

იგი მუშაობს შემდეგნაირად :

პროგრამა არის დაწერილი C/C++ ენაში , იგი იყენებს libpcap (http://www.tcpdump.org/) ბიბლიოთეკას ქსელის ბარათიდან პაკეტების წასაკითხად და მათ გასაგზავნად .

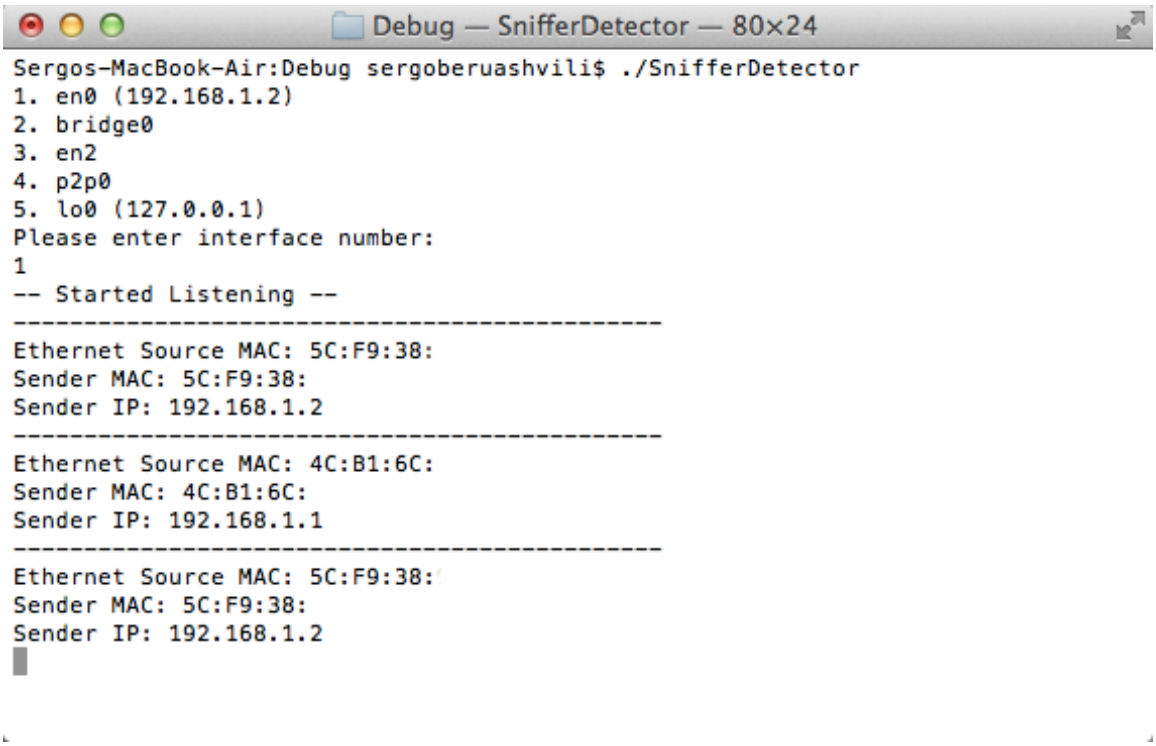
არსებული კოდი არის Cross-platform , რაც იმას ნიშნავს , რომ მისი დაკომპილირება და გამოყენება შესაძლებელია სხვადასხვა პლატფორმაზე (სისტემაზე) , MAC OS , Windows , Linux . ასევე შესაძლებელია არსებული კოდის გამოყენება მობილური მოწყობილობებისთვის.

პროგრამას გააჩნია CLI (command-line interface) , მისი ჩართვისას მომხმარებელი დაინახავს მოცემულ კომპიუტერზე არსებულ ქსელის ბარათების სიას შესაბამისი IP მისამართებით (თუ აქვთ მინიჭებული) , მომხმარებელმა უნდა აირჩიოს იმ ქსელის ბარათის ნომერი , რომელზედაც სურს ARP Spoofing-ისგან თავის დაცვა



ქსელის ბარათის ამორჩევის შემდეგ , პროგრამა შეეცდება დაუკავშირდეს მას და “მოისმინოს” ARP პაკეტები , ამ პროცესში თუ რაიმე ხარვეზი დაფიქსირდა (ვერ მოხერხდა ქსელის ბარათთან დაკავშირება ან “მოსმენის” დაწყება) , პროგრამა გამოიტანს შესაბამის შეცდომას.

მას შემდეგ რაც პროგრამა დაიწყებს ქსელის მოსმენას , იგი აკეთებს საკუთარ ARP ცხრილს , ინახავს ინფორმაციას თუ რომელ MAC მისამართზე რომელი IP მისამართია და პირიქით - რომელ IP მისამართზე რომელი MAC მისამართია. ასევე გამოაქვს ARP პასუხები ეკრანზე.



ყოველი ARP Reply პაკეტის მოსვლის შემდეგ , პროგრამა ადარებს ინფორმაციას საკუთარ ცხრილს , თუ მოსულ ARP Reply პაკეტში მითითებული IP და MAC მისამართი ემთხვევა ჩვენს ცხრილში არსებულს , მაშინ პროგრამა უცდის შემდეგ პაკეტს . ხოლო თუ რომელიმე შეცვლილია , მაგალითად რომელიმე MAC მისამართი იყო სხვა IP ზე აქამდე მინიჭებული ან/და პირიქით , IP მისამართი იყო სხვა MAC მისამართზე , პროგრამა ჩაინიშნავს ასეთ მოთხოვნას როგორც “საეჭვოს” , შეიტანს სპეციალურ ცხრილში , დააგენერირებს ახალ ARP Request პაკეტს და გააგზავნის ქსელში .

ARP პაკეტი გამოიყურება შემდეგნაირად :

Internet Protocol (IPv4) over Ethernet ARP packet		
octet offset	**0**	**1**
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

მოცემული ARP Request მოითხოვს პასუხს ქსელიდან თუ რომელ MAC მისამართს აქვს ის IP მისამართი მინიჭებული , რომელიც საეჭვოდ იქნა ჩანიშნული . პროგრამა იცდის 5 წამი (დროის ცვლილება შესაძლებელია) და ელოდება რამდენი მოწყობილობა გასცემს მას პასუხს თავის ARP მოთხოვნაზე , თუ მხოლოდ “საეჭვოდ” მიჩნეულმა კომპიუტერმა გასცა პასუხი , ეს იმას ნიშნავს რომ იგი არ ყოფილა საეჭვო და მართლაც შეცვლილა IP და MAC ჩანაწერები ქსელში , პროგრამა განაახლებს საკუთარ ცხრილს

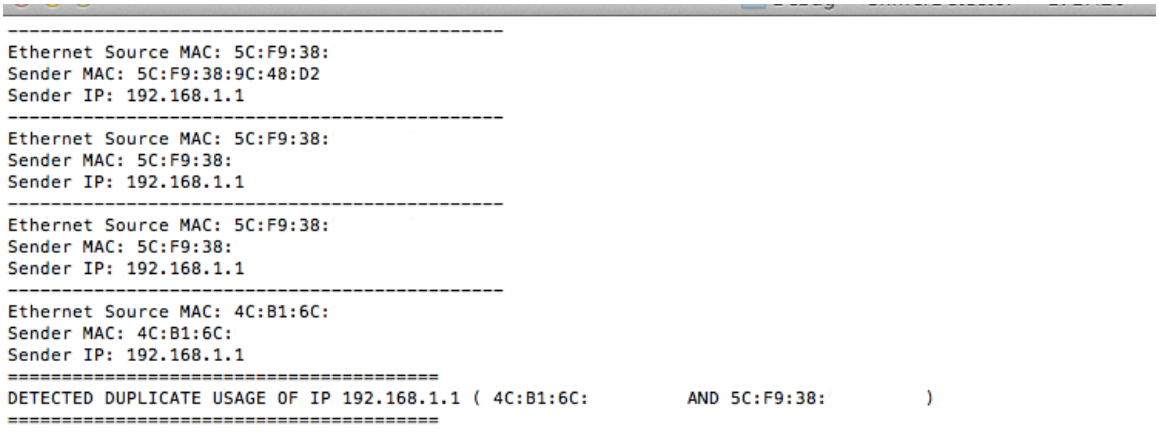
და გააგრძელებს მუშაობას , მაგრამ თუ სხვა კომპიუტერმაც გასცა პასუხი , ეს იმას ნიშნავს რომ 2 კომპიუტერი აცხადებს ერთსა და იმავე მისამართზე საკუთარ თავს , რაც ARP შეტევაზე მიუთითებს.

განვიხილოთ შემთქვევა :

კომპიუტერ A-ს აქვს მინიჭებული IP 10.0.0.1

კომპიუტერ B-ს აქვს მინიჭებული IP 20.0.0.1

B-მ გადაწყვიტა ქსელზე თავდასხმა , A-ს მისამართის მითვისებით , მას რეალურად აქვს ისევ 20.0.0.1 მისამართი , თუმცა , გარდა ამისა იგი ქსელში ავრცელებს ARP Reply მოთხოვნას რომელიც იუწყება რომ B-კომპიუტერზე ასევე 10.0.0.1 მისამართი , თუმცა B კომპიუტერი ვერ აუკრძალავს A კომპიუტერს ARP მოთხოვნაზე პასუხის გაცემას , იმ შემთხვევაში თუ რომელიმე სხვა კომპიუტერი ქსელში მოითხოვს თუ ვის აქვს 10.0.0.1 IP მისამართი , ამაზე A კომპიუტერი აუცილებლად გასცემს პასუხს რომ მას აქვს , თუ იგი კვლავ არის ამ ქსელში ჩართული , გარდა ამისა იგივე პასუხს გაუგზავნის კომპიუტერი B-ც , ასეთ შემთხვევაში , ჩვენი პროგრამა აღმოაჩენს დარღვევას და მომხმარებელს გამოუტანს შეტყობინებას .



ჩვენი პროგრამა მოახდენს ამ ორი მონაცემის შედარებას და თუ აღმოაჩენს რომ , გამომგზავნი MAC მისამართი განსხვავდება პაკეტში მითითებული მისამართისგან , ე.ი. საქმე გვაქვს ARP Spoofing-თან .

არსებობს ARP შეტევები , რომლებიც არ ანაგვიანებენ ქსელს , ანუ Broadcast-ს არ იყენებენ ARP Reply- სთვის , ისინი მხოლოდ კონკრეტულ კომპიუტერს (მსხვერპლს) უგზავნიან საკუთარ არპ პაკეტებს და მხოლოდ მას “ანაგვიანებენ” , ასეთი შეტევის აღმოჩენა მესამე კომპიუტერის მხრიდან შეუძლებელია , რადგანაც თავდამსხმელი მხოლოდ მსხვერპლს უგზავნის პაკეტს , “დამცველს” კი არანაირი ინფორმაცია არ გააჩნია ამ პაკეტის შესახებ. ასეთ შემთხვევაში კარგი იქნებოდა თავად ქსელურ მოწყობილობას , რომელიც ავრცელებს ქსელს (სვიჩი) გაეანალიზებინა ქსელი და თავიდან აეცილებინა საფრთხე , თუმცა როგორც წესი , ასეთი თავდაცვა მათ არ გააჩნიათ , ამიტომ მომხმარებელი დაუცველია , მას უნდა გააჩნდეს საკუთარი თავდაცვა ARP შეტევებისგან (მაგალითად სამაგისტრო ნაშრომში მოყვანილი პროგრამა) , რათა არ მოხდეს მისი ინფორმაციის მიტაცება.

დასკვნა

ARP Spoofing ძალზედ სახიფათო შეტევაა , მისი საშუალებით შესაძლებელია ბევრი მომხმარებლისთვის შეუმჩნევლად ქსელში შეღწევა , მისი მოსმენა და ინფორმაციის მოპარვა . იგი განსაკუთრებით საშიშია დაუშიფრავი ინფორმაციის გადაცემისას , ასეთ შემთხვევაში თავდამსხმელს სრული ინფორმაცია შეუძლია მიინიჭოს , მაგალითად თუ რომელიმე საიტზე შედიხართ და ავტორიაზიაცს გადიხართ , იმ შემთხვევაში თუ ამ საიტს არ გააჩნია SSL კავშირი (HTTPS) და ხდება ქსელის “მოსმენა” , მაშინ თავდამსხმელი პირდაპირ ხედავს მომხმარებლის სახელსა და პაროლს . ხოლო თუ არის დაშიფრული კავშირი (HTTPS) ით , თავდამსხმელი იღებს დაშიფრულ ინფორმაციას , რომლის გაშიფრვრის ალბათობა დამოკიდებულია კრიპტოგრაფიის ალგორითმის სირთულესა და საიმედოობაზე. HTTPS-ის შემთხვევაში მინიმალურია ასეთი ინფორმაციის გაშიფრვის შესაძლებლობა.

ARP Spoofing ძალზედ გამარტივდა მობილური ტექნოლოგიების განვითარებასთან ერთად , რადგანაც დღევანდელ სმარტფონებს აქვთ სრული წვდომა ქსელის ბარათზე , WiFi-ზე და შეუძლიათ განახორციელონ ARP შეტევები . უფრო და უფრო პოპულარული ხდება სხვადასხვა დაწესებულებებში ღია უკაბელო ინტერნეტის სერვისს (კაფეებში , გასართობ ცენტრებში და სხვა) , ასეთ ქსელს უამრავი მომხმარებელი ყავს , თუმცა შესაძლებელია მათი ინფორმაცია იპარებოდეს .

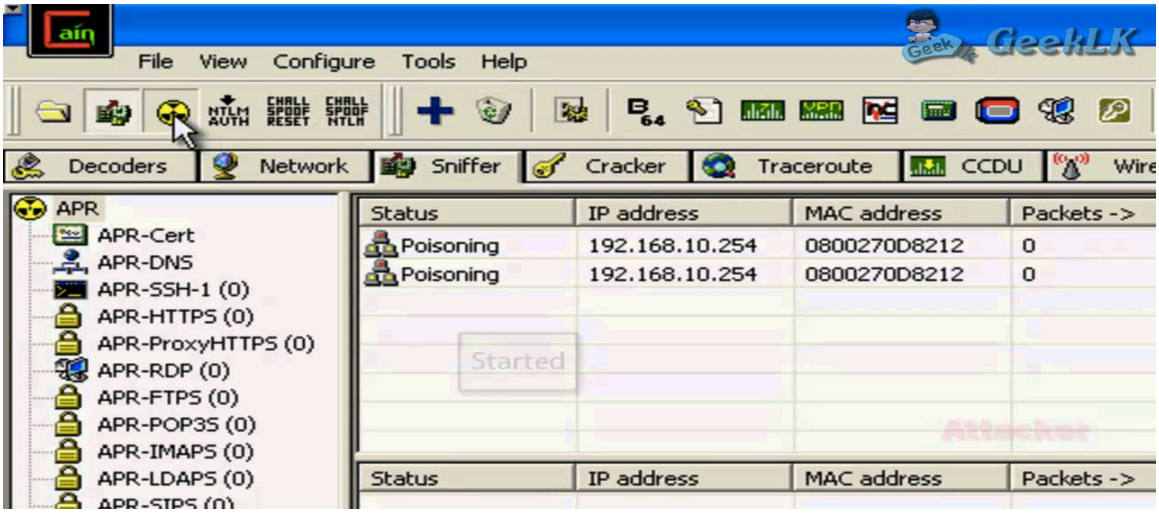
გარდა პოცენციური თავდამსხმელისა , ARP შეტევა შესაძლოა განხორციელდეს “თავდამსხმელის” ცოდნის გარეშე . არსებობს ვირუსები , რომლებიც ქსელში ახდენენ ARP თავდასხმას , შემდეგ აგროვებენ ყველა ისეთ პაკეტს რომლის დამუშავებაც/გაანალიზებაც შეიძლება (დაუშიფრავი პაკეტები , საიტზე ავტორიზაცია , სესიები და სხვა) , ამის შემდეგ კი საკუთარ სერვერზე ტვირთავს , შესაბამისად “თავდამსხმელი” თავად არის “მსხვერპლი” , ხოლო რეალური თავდამსხმელი საერთოდ არ ჩანს ამ პროცესში და მისი აღმოჩენაც ვერ მოხერხდება . ამითომ ძალზედ მნიშვნელოვანია სწორად იქნას შერჩეული ამა თუ იმ სერვისისთვის ინფორმაციის გადაცემის პროტოკოლი და დაცული იყოს ქსელის უსაფრთხოება.

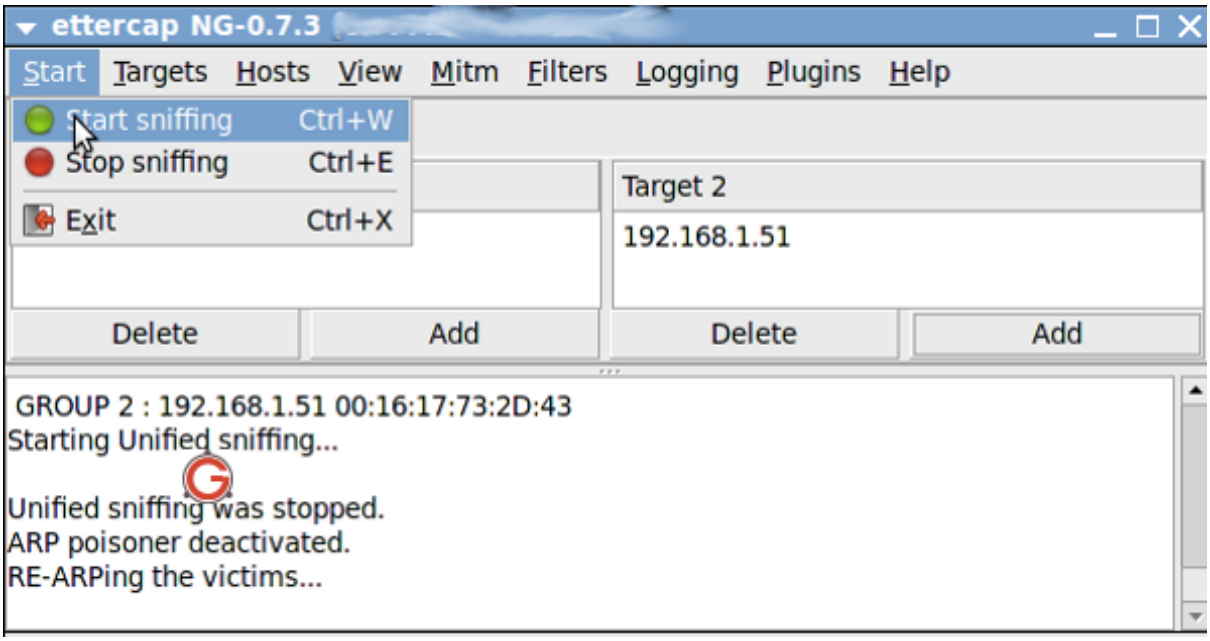
გახშირდა ასევე პროგრამები , რომლითაც შესაძლებელია ARP თავდასხმის განხორციელება , ძველად ასეთი პროგრამები არ იყო ასეთი გავრცელებული , თავდამსხმელს თავად უნდა შეექმნა იგი , რაც საკმაო შრომასა და ცოდნას მოითხოვდა ,

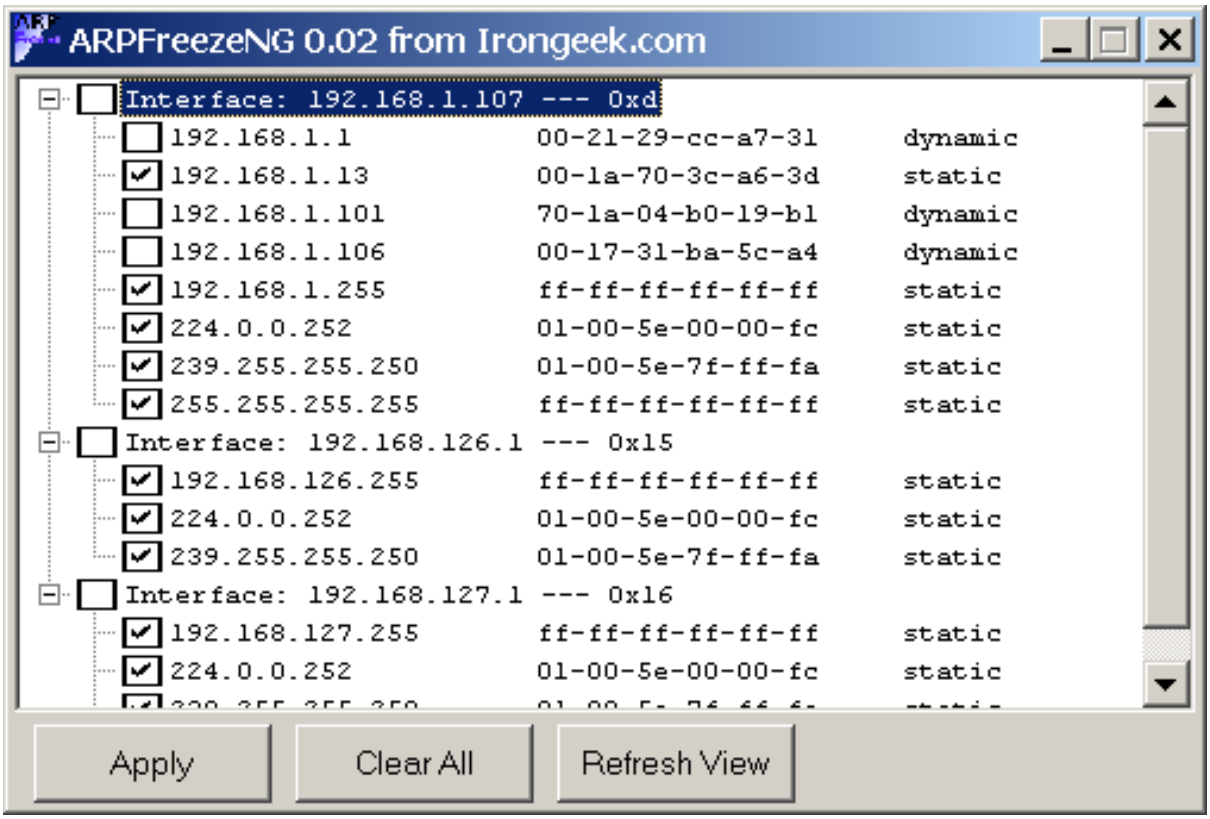
დღესდღეობით ყველას მიუწვდება ხელი ინტერნეტზე , ყველას შეუძლია გადმოიწეროს ასეთი პროგრამა და ღილაკზე ერთი თითის დაჭერით განახორციელოს ARP შეტევა , რომელიც ძალზედ სენსიტიურია .

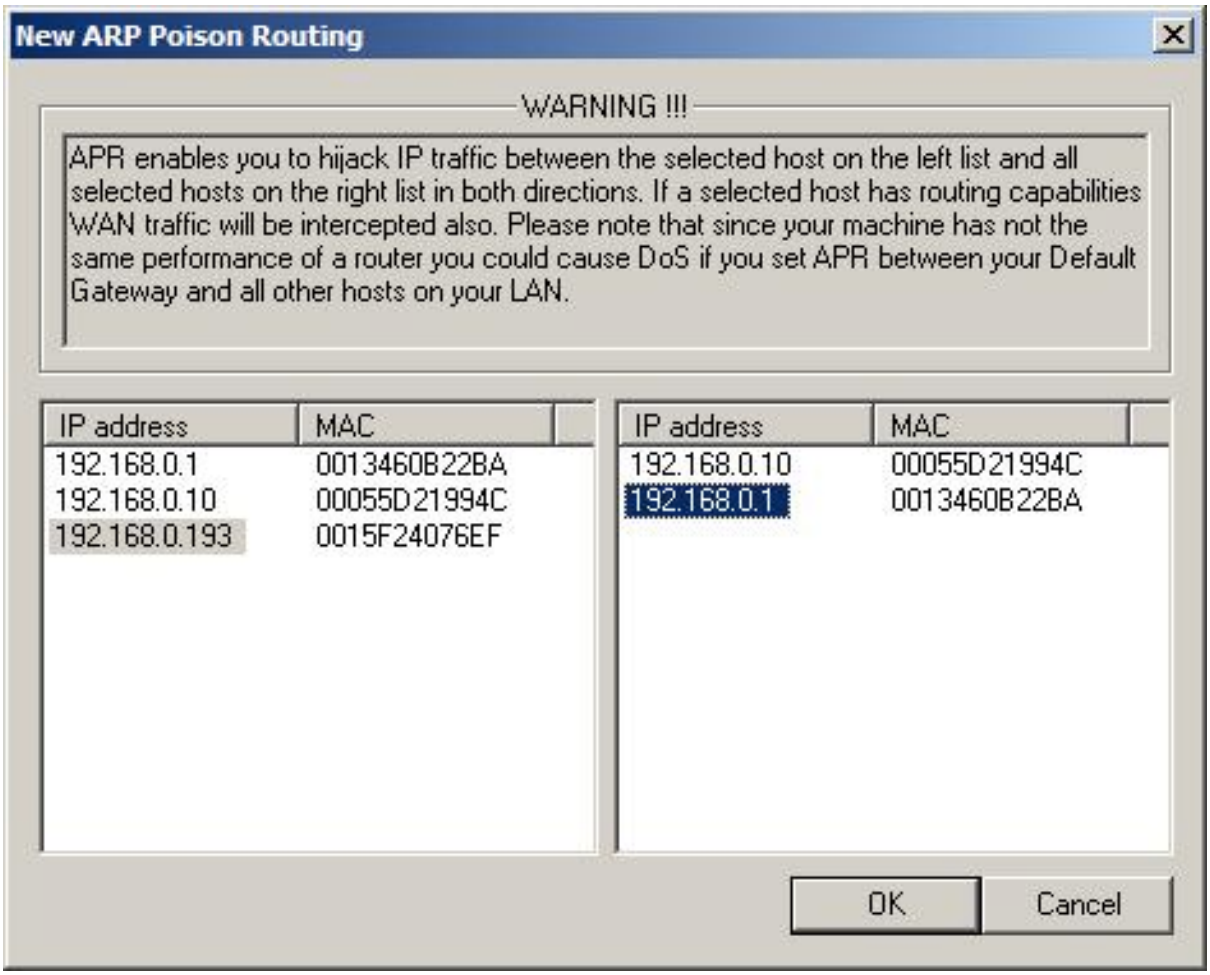
ქვემოთ მოცემულია ARP Poisoning-ის განხორციელებისათვის საჭირო პროგრამათა ჩამონათვალი და რამდენიმე მათგანის სურათიც.

- Arpspoof
- Arpoison
- Ettercap
- Cain&Abel
- Seringe
- ARP-FILLUP
- arp-sk
- ARPOc
- arpalert
- arping
- arpmitm
- arpoison
- ArpSpyX
- ArpToXin
- SwitchSniffer
- APE - ARP Poisoning Engine




ettercap NG-0.7.3
Start
Targets
Hosts
View
Mitm
Filters
Logging
Plugins
Help
Start sniffing Ctrl+W
Stop sniffing Ctrl+E
Exit Ctrl+X
Target 2
192.168.1.51
Delete
Add
Delete
Add
GROUP 2 : 192.168.1.51 00:16:17:73:2D:43
Starting Unified sniffing...
Unified sniffing was stopped.
ARP poisoner deactivated.
RE-ARPing the victims...


ARPFreezeNG 0.02 from Irongeek.com
Interface: 192.168.1.107 --- 0xd
192.168.1.1 00-21-29-cc-a7-31 dynamic
192.168.1.13 00-1a-70-3c-a6-3d static
192.168.1.101 70-1a-04-b0-19-b1 dynamic
192.168.1.106 00-17-31-ba-5c-a4 dynamic
192.168.1.255 ff-ff-ff-ff-ff-ff static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static
255.255.255.255 ff-ff-ff-ff-ff-ff static
Interface: 192.168.126.1 --- 0x15
192.168.126.255 ff-ff-ff-ff-ff-ff static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static
Interface: 192.168.127.1 --- 0x16
192.168.127.255 ff-ff-ff-ff-ff-ff static
224.0.0.252 01-00-5e-00-00-fc static
Apply
Clear All
Refresh View



სერვერებისა და საიტების თავდაცვა თანდათან იზრდება პროვაიდერებისა და ჰოსტინგის მიერ , თავდამსხმელისთვის უფრო მარტივი ხდება მომხმარებლისგან ინფორმაციის მოპარვა ვიდრე საიტისგან ან სერვერისგან , ამიტომაც იქმნება ARP Spoofing-ის ვირუსები , რომლებიც არა მარტო იმ კომპიუტერზე იძლევა ინფორმაციას თუ რომელ კომპიუტერზეც იქნება ეს ვირუსი , არამედ მთელს ქსელზე .

სამაგისტრო ნაშრომის წერის პერიოდში, თბილისში არსებულ რამდენიმე ყველაზე მსხვილ ქსელზე მოხდა ARP შეტევა იმის შესამოწმებლად თუ რამდენად დაცული არიან ეს ქსელები ასეთ შეტევებისგან , თუმცა შედეგი საგანგაშო აღმოჩნდა. არცერთი ქსელი არ იყო ამ შეტევისგან დაცული

ჩვენ განვიხილეთ რამდენიმე ძირითადი მიდგომა და გზა , თუ როგორ შეიძლება აღმოვაჩინოთ ქსელში ასეთი საფრთხე და როგორ დავიცვათ თავი , თუმცა ეს პრობლემა უფრო ფართო მასშტაბებს მოიცავს , საჭიროა მომხმარებელს კარგად ესმოდეს თუ რა

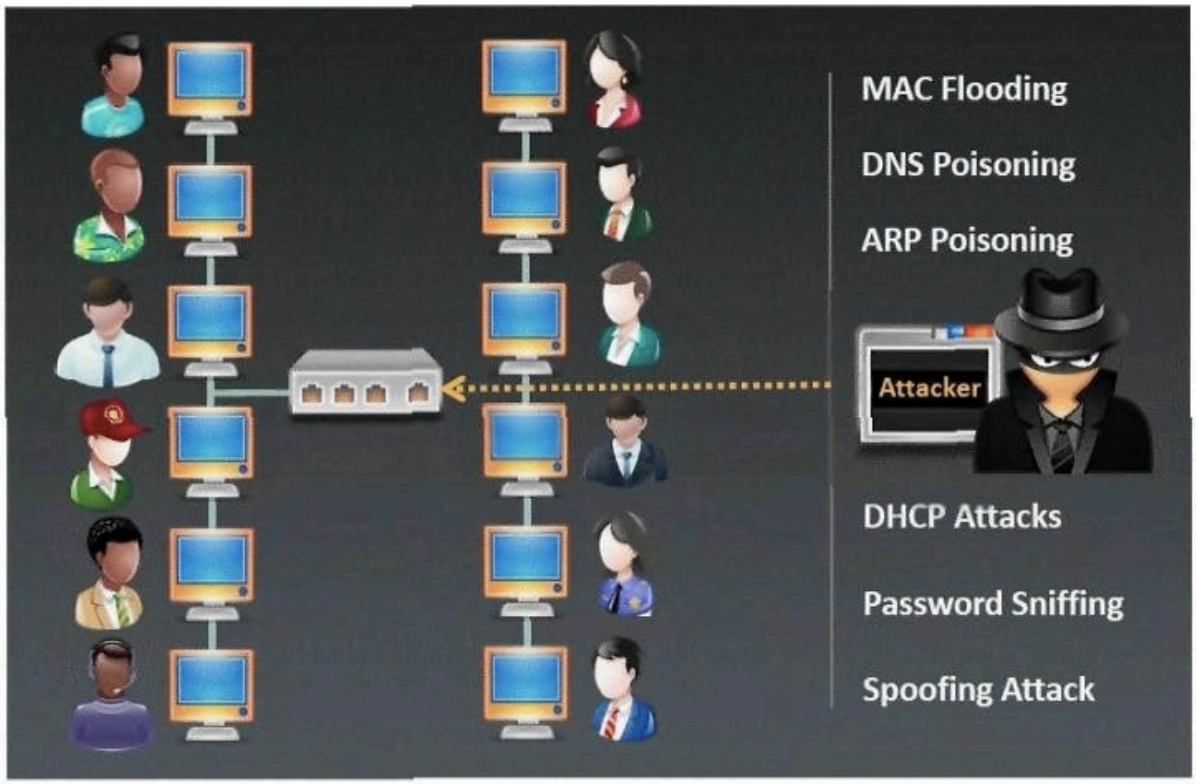
საფრთხესთან არის დაკავშირებული ამა თუ იმ ქსელში შესვლა , რომელი ქსელის ნდობა შეიძლება , სად შეიძლება იყოს დავდამსხმელი და სად არა .

სამაგისტრო ნაშრომში წარმოჩენილი პროგრამა დაეხმარება მომხმარებლებს ამ პრობლემის გადაწყვეტაში , იგი აღმოაჩენს ასეთ საფრთხეს და მომხმარებლებს შეატყობინებს საფრთხის შესახებ , თუმცა ისეთი მომხმარებლების რაოდენობა , რომლებმაც არ იციან ასეთი საფრთხის არსებობის შესახებ საკმაოდ დიდია , ამიტომ ყოველი სერვისი მიმწოდებელი თავად უნდა ზრუნავდეს საკუთარი მომხმარებლის თავდაცვაზე . მაგალითად ინტერნეტ პროვაიდერი (ღია WiFi ქსელებისა) თავად უნდა ზრუნავდეს იმაზე რომ მის ღია ქსელში არ მოხდეს ინფორმაციის მოპარვა , რადგან ასეთ შემთხვევაში მთელი მისი ქსელი დაუცველია , ასევე აღსანიშნავია უნივერსიტეტის ღია WiFi ქსელები , სადაც ყველა სტუდენტს და გამვლელს შეუძლია შესვლა , რომელზეც არიან ასევე უნივერსიტეტის თანამშრომლები დაკავშირებული , ხოლო ამ თანამშრომლებს აქვთ წვდომა სტუდენტების მართვის სისტემასთან , ძალზედ მარტივია სტუდენტისათვის ქსელის “მოსმენა” და ასეთ სისტემაში შესვლა , ამიტომ საჭიროა სერვისის მიმწოდებელს კარგად ქონდეს ეს რისკები გააზრებული .

საუკეთესო მიდგომა ასეთ შემთხვევაში არის ინფორმაციის დაშიფრულად გადაცემა , სადაც გამოყენებული იქნება მძლავრი ალგორითმი ინფორმაციის დასაშიფრად რაც მის გაშიფრვის ალბათობას ნულამდე დაიყვანს . ამიტომ საჭიროა სტუდენტური საიტები და ასეთი სერვისები განთავსებული იყოს HTTPS პროტოკოლის მქონე სერვერზე.

HTTPS-ის გამოყენების დროსაც არის გარკვეული საფრთხეები, რომელიც მომხმარებელმა უნდა იცოდეს. მაგალითად: თუ მომხმარებელი მისი Browser-ის ველში ჩაწერს facebook.com -ს ის შესაძლოა გახდეს თავდასხმის მსხვერპლი , რადგანაც ამ დროს შეიძლება ქსელში იყოს თავდამსხმელი , რომელიც facebook.com-ისკენ მიმართულ მოთხოვნას გადაამისამართებს საკუთარ სერვერზე, შეინახავს მას , ხოლო ამის შემდგომ თავად დაამყარებს კავშირს facebook.com-თან მიუხედავად იმისა , რომ facebook-ს გააჩნია HTTPS პროტოკოლი. ხოლო თუ მომხმარებელი ჩაწერს https://facebook.com-ს ამ შემთხვევაში ასეთ თავდასხმა ვერ მოხერხდება , რადგანაც Browser ცდილობს თავად დაამყაროს პირველ რიგში SSL კავშირი ვებ-გვერდთან , SSL კავშირის დამყარებისას კი ხდება SSL სერტიფიკატის შემოწმება და მონაცემთა დაშიფვრა , ხოლო ამ ყველაფრის მერე მონაცემების გაგზავნა , თავდამსხმელი ვეღარ მოახერხებს ინფორმაციის საკუთარ სერვერზე გაგზავნას, ხოლო თუ უბრალოდ

მოუსმენს ქსელს , დაინახავს დაშიფრულ ინფორმაციას რომლის აღდგენიც შანსიც მინიმალურია.



გამოყენებული ლიტერატურა

- CompTIA Network+ , ავტორი Todd Lammle , გამოცემის წელი 2009 , გამომცემელი Wiley Publishing, Inc
- CompTIA Security+ , ავტორი Mark Ciampa , გამოცემის წელი 2008 , გამომცემელი Course Technology PTR
- http://en.wikipedia.org/wiki/Address_Resolution_Protocol
- http://en.wikipedia.org/wiki/ARP_spoofing
- http://hakipedia.com/
- http://hakipedia.com/index.php/ARP_Poisoning
- Detecting ARP Spoofing: An Active Technique , ავტორები : Vivek Ramachandran (Cisco Systems, Inc., Bangalore India) and Sukumar Nandi (Indian Institute of Technology, Guwahati, Assam, India)