|  |
| --- |
| ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  КОМПЬЮТЕРИЙН ТЕХНИК МЕНЕЖМЕНТИЙН СУРГУУЛЬ  ТӨСЛИЙН АЖЛЫН  ТАЙЛАН  Сэдэв: Knocker нээлттэй порт илрүүлэгч програм  Даалгавар: ТСР флаг нэмэх мөн IP хаягийн завсарт хайлт хийх  Гүйцэтгэсэн: А.Насантогтох /HW09D402/    УЛААНБААТАР ХОТ 2012 он |

1. НҮҮР ХУУДАС
2. ГАРЧИГ
3. ТӨСЛИЙН ХЭСЭГ............................................................................................................2
4. ОРШИЛ...............................................................................................................................4
5. ЗОРИЛГО............................................................................................................................4
6. УДИРТГАЛ........................................................................................................................5
7. ОНОЛ..................................................................................................................................5
8. ТУРШИЛТ........................................................................................................................18
9. ТӨСӨЛ..............................................................................................................................26
10. ҮР ДҮН ТЕСТЧИЛЭЛ....................................................................................................31
11. ЕРӨНХИЙ ДҮГНЭЛТ.....................................................................................................34
12. АШИГЛАСАН БҮТЭЭЛИЙН ЖАГСААЛТ.................................................................38
13. ХАВСРАЛТ......................................................................................................................39

**III. ТӨСЛИЙН ХЭСЭГ**

Зургийн жагсаалт

|  |  |  |
| --- | --- | --- |
| д/д | Зургийн дугаар тайлбар | Хуудас |
| 1 | SYN хайлт Порт нээлттэй. | 9 |
| 2 | SYN хайлт Порт хаалттай эсвэл шүүлтүүр хийгдсэн. | 9 |
| 3 | Connect() хайлт порт нээлттэй. | 10 |
| 4 | Connect() хайлт порт хаалттай. | 10 |
| 5 | UDP хайлт шүүгдсэн порт. | 11 |
| 6 | UDP хайлт хаалттай порт. | 11 |
| 7 | UDP хайлт нээлттэй порт. | 11 |
| 8 | UDP хайлт нээлттэй | шүүгдсэн порт. | 11 |
| 9 | SCTP INIT хайлт порт нээлттэй. | 12 |
| 10 | SCTP INIT хайлт порт хаалттай. | 12 |
| 11 | SCTP INIT хайлт порт шүүгдсэн. | 13 |
| 12 | Stealth хайлт порт нээлттэй эсвэл шүүгдсэн. | 14 |
| 13 | Stealth хайлт порт хаалттай. | 14 |
| 14 | Stealth хайлт порт шүүгдсэн. | 14 |
| 15 | ACK хайлт порт шүүгдээгүй. | 15 |
| 16 | ACK хайлт порт шүүгдсэн. | 15 |
| 17 | TCP window хайлт порт шүүгдээгүй. | 15 |
| 18 | ТСР window хайлт порт хаалттай. | 16 |
| 19 | ТСР window хайлт порт нээлттэй. | 16 |
| 20 | ТСР window хайлт порт шүүгдсэн. | 16 |
| 21 | Maimon хайлт порт нээлттэй. | 17 |
| 22 | Maimon хайлт порт хаалттай. | 17 |
| 23 | SCTP COOKIE ECHO хайлт нээлттэй|шүүгдсэн порт. | 17 |
| 24 | SCTP COOKIE ECHO хайлт хаалттай порт. | 18 |
| 25 | Хамгийн эхэнд Ubuntu10.10 дээр компайл хийж ажиллуулахад ажилах үедээ гарсан алдаа. | 19 |
| 26 | Ubuntu 4.10 програм ямар нэгэн алдаа заалгүй ажиллаж байгаа байдал. | 20 |
| 27 | Текстийн өнгө өөрчилж байгаа хэсгийг нь хасаж байгаа процесс. | 22 |
| 28 | Чухал шаардлагатай бус элементүүдийг нь хасаж байгаа процесс. | 23 |
| 29 | Чухал бус элементүүдийг хассаны дараа Ubuntu10.10 дээр ажиллаж байгаа байдал | 24 |
| 30 | Knocker нээлттэй порт илрүүлэгч програмаар хэрхэн google.com-н нээлттэй портуудыг илрүүлж байгаа процесс | 25 |
| 31 | Энэхүү зурган дээр knocker нээлттэй порт илрүүлэгч програмаар loopback интерфэйсийг хайж байгаа байдал | 26 |
| 32 | Програмын блок диаграм | 29 |
| 33 | Энэхүү тестчилэл дээр IP хаягийн завсарт хайлт хийж байгаа процесс | 31 |
| 34 | Хаягийн завсраар хайж байгааг пакетийг нь барьсан байдал | 32 |
| 35 | Флаг өөрчилж (хайлтын арга) хайхад ажиллаж байгаа байдал | 33 |
| 36 | XMAS аргаар хайж байгаа пакетууд. | 34 |

Хүснэгтийн жагсаалт

|  |  |  |
| --- | --- | --- |
| д/д | Хүснэгтийн нэр | Хуудас |
| 1 | Өөрийн кодон дотроос хассан функцүүдийн жагсаалт | 21 |
| 2 | Хийсэн өөрчлөлтүүдийн жагсаалт болон тэдгээрийн хэрэгжүүлэлтүүдийн товч тайлбар | 26 |

1. **ОРШИЛ**

Энэхүү төслийн зорилго нь чөлөөт эхийн програм хангамж болох knocker нээлттэй порт илрүүлэгч програмыг судалж түүний хэрхэн ажиллаж байгааг нь нарийн ойлгох юм. Цаашлаад энэхүү програм дээрээ өөрийн төслийн өөрчлөлт болох IP хаягийг тодорхой завсараар хайдаг болгохоос гадна нээлттэй порт илрүүлэх хайлтын аргыг нэмж оруулах юм. Ингэснээр миний програм маань гараас оруулж өгсөн IP хаягийн тодорхой завсарт ТСР флаг өөрчлөх замаар хайлтын үйл ажиллагааг хийх боломжтой болно. Сүлжээний орчинд нээлттэй порт байгаа эсэхийг ихэвчлэн хакер (дайрагч), эсвэл тухайн сүлжээний администратор нь хайдаг. Хакеруудын хувьд тухайн сүлжээнд тухайн нээлттэй портыг нь ашиглан нэвтэрч орох сонирхлын үүднээс үүнийг шалгаж үздэг бол харин сүлжээний администраторууд өөрийн сүлжээ цаашлаад системийн аюулгүй байдал тасралтгүй үйл ажиллагааг нь хангах зорилгоор нууцлал аюулгүй байдлыг тохиргоог хийхдээ хэрэглэдэг байна. Өөрөөр хэлбэл сүлжээний администраторууд хакеруудын эсрэг системийн хамгааллын төхөөрөмж болон системийн тохиргоог хийж гүйцэтгэхдээ хэрэглэдэг.

1. **ЗОРИЛГО**

Энэхүү төслийн ажлаар хийж гүйцэтгэх зорилго маань чөлөөт эхийн knocker нээлттэй порт илрүүлэгч програм дээр өөрчлөлт хийн хөгжүүлж түүнд гараас IP хаягийн завсар авах мөн түүний хайлтын арга дээр TCP флагуудыг идэвхижүүлсэнээр хайлтын аргуудыг баяжуулах юм.

Дээрх үндсэн зорилгод хүрэхийн тулд дараах дэд зорилтууд гарч ирж байгаа юм.

1. Порт гэж юу болох, түүний үүрэг
2. Knocker нээлттэй порт илрүүлэгч програмд тохирсон linux төрлийн үйлдлийн систем суулгаж knocker програмыг алдаагүй ажиллуулах.
3. Knocker нээлттэй порт илрүүлэгч програмыг судалж, түүний хэрхэн ажиллаж байгааг нь нарийн ойлгож мэдэх.
4. Төгсгөлд нь хаана нь ямар өөрчлөлт оруулахаа тодорхойлоод өөрчлөлтийг оруулна.

1. **УДИРТГАЛ**

Нээлттэй порт илрүүлэгч хэрэгсэл нь сүлжээний хайлт хийдэг програм хангамжийн нэг хэсэг нь бөгөөд энэ нь тухайн сүлжээ цаашлаад системийн аюулгүй байдлыг хангахад чухал үүрэгтэй юм. Дээр хэлсэнчлэн тухайн системийн аюулгүй байдлыг хангах програм болон техник хангамжийн тохируулга хийхэд чухал үүрэгтэй байдаг. Мөн нөгөө талаар халдагчдад ч гэсэн сүлжээний цоорхойтой хэсгийг нь илрүүлэх хэрэгсэл болох нь ч байдаг. Миний хийсэн өөрчлөлтөөр энэхүү програм нь зөвхөн ямар нэгэн хост эсвэл серверийн нээлттэй портыг нь илрүүлэх боломжоор үл хязгаарлагдан цаашлаад тухайн дотоод сүлжээний бүх хостынх нь нээлттэй портыг хурдан хугацаанд, хайлтын боломжит аргуудын хүрээнд илрүүлэх боломжийг олгож байгаа юм. Ингэснээр бид дотоод сүлжээнд байгаа хост бүр дээр хамгаалалтын арга хэмжээг авах боломжоор хангах юм.

1. **ОНОЛ**

Порт гэдэг нь компьютерийн сүлжээний орчинд хэрэглэгддэг юм. Үүнийг сүлжээний эцсийн цэгийн төхөөрөмжийн үйлдлийн ситем дээр хэрэглээний програм хангамжуудыг тодорхойлох эсвэл програмын процессын бүтцийг тодорхойлоход хэрэглэнэ. Порт нь IP хаягтай уялдан холбогдож ажиллана. Түүнээс гадна холболт тогтоож байгаа протоколтой хамааралтай байдаг. Энгийн үгээр тайлбарлавал порт нь пакетад суурилсан сүлжээгээр (packet switching network) холбогдсон компьютер тус бүрийн хувьд түүн дээр ачаалалагдаж байгаа процесс эсвэл өөр өөр хэрэглээний програм хангамжийг тодорхойлох дахин давтагдашгүй тодорхойлогч юм. Портыг TCP болон UDP гэсэн дамжуулалтын түвшний протоколууд ашигладаг. Порт нь 16 битийн урттай байх ба портын дугаар гэсэн нэрээр илүү их хэрэглэгдэнэ. Портын дугаар нь компьютерийн IP хаяг дээр нэмэгдсэн байдаг. Портын дугаар болон IP хаягын хослол нь хамт байх ёстой ба энэхүү 2 бүрэлдэхүүн хэсэг хамтдаа байж хүрэх газрын хаягийг бүрэн тодорхойлж чаддаг. Эндээс холболтын үед ялгаатай IP хаяг эсвэл протоколууд нь ижил портын дугаар ашиглаж болно. Нийт 65535 порт байдгаас эхний 1024 портыг олон нийтэд ил байдлаар олонд танигдсан (бараг бүгд IANA-д албан ёсоор бүртгэгдсэн) байдаг ба тэдгээр нь хост дээрх сервисийг тодорхойлдог. Сервер клиент загварын хэрэглээний програм хангамжийн бүтцэд портыг сервер талд нь олон тооны клиентийг хүлээн авахад түүний сервисийг нь ялгах зорилгоор хэрэглэнэ. Ингэж холбогдсоны дараа өөр портоор дахин холболт хийж болдог.

Дамжуулалтын түвшний TCP болон UDP гэх мэт протоколууд нь өөрсдийнхөө толгойн хэсэгт өөрийн гарч байгаа болон очих талын портын дугаарыг нь тодорхойлж өгсөн байдаг. Порт нь 16 битийн урттай байдаг бөгөөд эндээс хязгаар нь 0-ээс 65535 хүртэл хязгаарт утгаа авна. (0 гэсэн портын дугаар нь нөөцөнд байдаг ба хэрэглэгддэггүй). Интернэт соккет дээгүүр дамжиж байгаа процесс нь файлын төрөл, дамжуулалтын протокол, портын дугаар IP хаягуудтай нийлж нэг бүхэл болж байж оролт гаралтаа хаашаа явахаа шийдэж чиглүүлэгддэг. Энэ процессыг хүлээх (bind) гэж нэрлэх нь элбэг байдаг ба энэ нь сүлжээн дээгүүр илгээх болон хүлээн авах хэсгийг нь зөвшөөрдөг. Үйлдлийн системийн сүлжээний програмууд нь бүх хэрэглээний програм хангамжуудаас гарч байгаа дамжуулагдах портоор нь сүлжээ рүү гаргах мөн сүлжээнээс ирж байгаа пакетын процессын тохирч байгаа IP хаяг болон портын дугаараар нь дамжуулан дэмжих үйл ажиллагаа хийдэг. Ижил дамжуулах протокол ашиглаж байгаа ганцхан л IP хаяг болон портын дугаар дээр хүлээх (bind) процесс тодорхойлогддог. Олон програм ижил IP хаяг болон ижил портын дугаар хэрэглэх гэж оролдох үед алдаа үүсдэг энэ нь хэрэглээний програм хангамжуудын дунд нэлээн нийтлэг гардаг алдаа юм. Энэхүү алдааг заримдаа портын зөрчил ч гэж нэрлэх нь тохиолддог.

Хэрэглээний програм хангамжууд нь нийтлэг сервисүүдээ ихэвчлэн тодорхой заагдсан нөөцлөгдсөн байдаг портуудын дугаараас өгдөг бөгөөд олонд танигдсан портын дугаараар клиент талаас хүсэлт илгээдэг. Энэ процессийг чагнах (listening) гэж нэрлэдэг ба энэ маань ижил портын дугаар ашиглан холболт хийж байгаа нэг сервер нэг клиентийн холболт тогтоох процессийн клиент талынх нь хүсэлтийг нь хүлээн авдаг байна. Өөр клиентүүд нь өөр чагнах горимд байгаа (listening) портоор үргэлжлүүлэн холбогдож болдог.

TCP холболтыг тодорхойлох жагсаалт нь түүний холболтыг алдаагүй ажиллуулж байгаа юм. Жагсаалтад {local address, local port, remote address, remote port}-үүд ордог. Олонд танигдсан (well-known) портуудыг IANA [Internet Assigned Numbers Authority](http://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority) (IANA)-аас хурлаар хэлэлцэж байж баталдаг.

Сүлжээний гол чухал гэсэн сервисүүд жишээлбэл вэб сайт гэх мэт нь ихэвчлэн 1024-с бага тооны портын дугаартай байдаг. Олон үйлдлийн системд эдгээр портуудаар хүлээлт (bind) хийх хэрэглээний програм хангамжинд шаардагдах давуу эрх хэрэгтэй болдог яагаад гэвэл IP сүлжээний үйлчилгээ нь дутагдалтай гэж үзэгдсэн. Эсрэгээрээ клиент талын эцсийн хэрэглэгчийн хэрэглэж байгаа портын дугаар хувиарлагдсан цөөн тооны портын дугаараас маш их байдаг. Үүнийг нэг наст (ephemeral) port гэж нэрлэдэг. Портын дугаарууд нь дамжуулах түвшний протоколын толгойд байрладаг ба шууд ойлгож болдог. Зөвхөн илгээж байгаа болон хүлээн авч байгаа талаас нь үл хамааран шууд уншигдахуйц хэлбэртэй байдаг.

Голдуу галт хана нь тохируулгаараа ялгаатай пакетуудын илгээгдсэн талын болон явуулж байгаа портын дугаар дээр нь суурилан тохируулга хийсэн байдаг.

[Internet Assigned Numbers Authority](http://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority) (IANA) нь хариуцлагатай олон улсын DNS root, IP хаяг мөн бусал интернэт протоколын нөөцүүдийг зохицуулдаг байгууллага юм. Энэ байгууллага нь мөн нийтлэг хэрэглэгдэх портуудыг нөөцөлж буюу бүртгэж авдаг.

Портын дугааруудыг 3 ангилж авч үздэг бөгөөд эдгээрт олонд танигдсан (well-known) портууд, бүртгэгдсэн (registered) портуул мөн динамик эсвэл хувийн (private) портууд гэж ангилдаг. Олонд танигдсан порт нь 0-1023 хүртэл байдаг бөгөөд нийтлэг хэрэглэгддэг хэдэн портыг орууллаа.

* *20 & 21*: [File Transfer Protocol](http://en.wikipedia.org/wiki/File_Transfer_Protocol) (FTP)
* *22*: [Secure Shell](http://en.wikipedia.org/wiki/Secure_Shell) (SSH)
* *23*: [Telnet](http://en.wikipedia.org/wiki/Telnet) remote login service
* *25*: [Simple Mail Transfer Protocol](http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) (SMTP)
* *53*: [Domain Name System](http://en.wikipedia.org/wiki/Domain_Name_System) (DNS) service
* *80*: [Hypertext Transfer Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (HTTP) used in the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web)
* *110*: [Post Office Protocol](http://en.wikipedia.org/wiki/Post_Office_Protocol) (POP3)
* *119*: [Network News Transfer Protocol](http://en.wikipedia.org/wiki/Network_News_Transfer_Protocol) (NNTP)
* *143*: [Internet Message Access Protocol](http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol) (IMAP)
* *161*: [Simple Network Management Protocol](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol) (SNMP)
* *443*: [HTTP Secure](http://en.wikipedia.org/wiki/HTTP_Secure) (HTTPS)

Бүртгэгдсэн портуудын хязгаарт 1024-49151 хүртэл ордог. Динамик эсвэл худийн портууд нь 49152-65535. Эдгээрээс нэг наст буюу [ephemeral ports](http://en.wikipedia.org/wiki/Ephemeral_port) портыг сонгож ажилладаг.

Knocker нээлттэй порт илрүүлэгч програм нь TCP холболт тогтоох процессыг ашигладаг (3-way handshake). TCP протоколын нээлттэй портыг илрүүлэх олон арга байдгаас энэ арга нь хамгийн энгийн бөгөөд нийтлэг хэрэглэгддэг барга нь юм. Өөрөөр хэлбэл манай програм нь портыг нь шалгах гэж буй хоструугаа TCP 3 way handshake процессийн эхний процесс болох SYN пакетыг шидэж байгаа юм. Хэрэв түүнээс ACK and SYN пакет ирвэл тухайн порт нээлттэй байна гэсэн хариуг илгээдэг. Харин шалгаж байгаа хостоос нь RST ACK гэсэн хариу ирвэл тухайн портыг хаалттай байна гэж үзнэ. Миний сонгож авсан knocker програмын ажиллах зарчим ийм байна.

Энэхүү програм нь гараас шалгах хостынхоо хаягийг авахдаа string төрлөөр авч байгаа юм. Үүнийхээ дараа түүнийгээ домайн нэр эсэхийг нь шалгаад хэрэв домайн нэр мөн гэж гэсэн хариу ирвэл (серверийн IP хаяг ирвэл) тухайн хаягаар эсрэг тохиолдол тухайн гараас оруулсан string хувьсагчийг шууд өөрийн struct бүтэцтэй хувьсагч болох host\_addr гэсэн заагчаар IP хаягийг нь хөрвүүлэн таньж авч соккетоо үүсгэж ажиллаж байгаа юм. Ингэж аваад портынхоо хязгаарын тоогоор давталт хийж тухайн хост руу энэхүү завсарт байгаа порт бүрийн тоогоор connect() функц үүсгэж дамжуулаад түүнийхээ утгыг нь буцааж шалгаад хэрэв 0 байвал тухайн порт нээлттэй (TCP холболт тогтсон), -1 тохиолдол тухайн порт хаалттай (TCP холболт тогтоож чадаагүй) гэсэн үр дүнг гаргаж ирж байгаа юм.

**Нээлттэй порт илрүүлэх аргууд**

* **TCP SYN хайлт**

Энэхүү аргыг ихэвчлэн хагас-нээлттэй хайлтын арга гэж нэрлэх тохиолдол байдаг. Учир нь холболтоо бүрэн дуусгалгүй орхидог. Клиент талаас SYN пакет шидэж хариу хүлээх ба хэрэв SYN/ACK гэсэн хариу ирвэл порт нээлттэй гэсэн үг ба харин RST (reset) хаалттай гэж үздэг. Хэрвээ хэд хэд дахин дамжуулалт хийхэд ямар ч хариу өгөхгүй эсвэл түүний хариуд ICMP (type 3, code 1, 2, 3, 9, 10, or 13) гэсэн хүршгүй алдаануудыг зааж байвал тухайн порт галт ханаар шүүлт хийгдсэн гэсэн үг. Мөн түүнээс гадна энэхүү аргаар хайлт хийхэд SYN шидсэний хариуд буцаагаад зөвхөн SYN (ACK байхгүй) ирэхэд портыг нээлттэй байна гэж ойлгож болно.



**Зураг 1 SYN хайлт** Порт нээлттэй.



**Зураг 2 SYN хайлт** Порт хаалттай эсвэл шүүлтүүр хийгдсэн.

* **TCP connect хайлт**

Хэрвээ танд бүрэн бус (raw) соккет үүсгэх эрх байхгүй (супер хэрэглэгчийн эрх) бол та connect() функцийг дуудаж SYN хайлтын төрлийг хийж болно. Үүнийг connect() хайлт гэдэг. Хэрвээ SYN хайлт хийх боломжтой бол энэ арга нь хэрэгжүүлж болох хамгийн энгийн бөгөөд хялбар арга юм.



**Зураг 3 connect() хайлт** порт нээлттэй.



**Зураг 4 connect() хайлт** порт хаалттай.

* **UDP хайлт**

UDP хайлт нь тухайн хайж буй хостын порт бүр лүү UDP пакет илгээнэ. Нийтлэг үйлчилгээтэй портууд руу өгөгдөл илгээдэг боловч ихэнх порт руу хоосон өгөгдөл дамжуулдаг байна. Өгөгдлийн уртыг нь тодорхойлоод санамсаргүй өгөгдлийг дамжуулж болох ба хэрэв та өгөгдлийн уртыг 0 гэсэн утгаар заасан бол өгөгдлийн хэсгийг тэр чигт нь хааснаас ялгаагүй гэсэн үг. Энэхүү өгөгдлийн хариуд ICMP (type 3, code 3) хүршгүй алдаа ирвэл порт хаалттай гэсэн үг. Харин ICMP (type 3, codes 1, 2, 9, 10, or 13) гэсэн хүршгүй алдаанууд ирвэл порт шүүгдсэн гэсэн үг. Цөөхөн тохиолдолд үүний хариуд UDP пакет ирдэг энэ үед тухайн портыг нээлттэй гэж үзнэ. Хэрэв хэд дахин дамжуулалтын дараа ч гэсэн ямар нэгэн хариу ирэхгүй байвал тухайн портыг нээлттэй|шүүгдсэн гэж үздэг. UDP хайлтын арга нь хугацаа их зарцуулдаг. Яагаад гэвэл UDP хайлтаар хайхад түүнд ихэвчлэн ICMP алдааны хариу ирдэг бөгөөд энэ алдааны хариу их удаан илгээгддэг. Жишээ нь: Linux 2.4.20 цөм нь ICMP хүршгүй алдааны мессежийг 1 секундэд 1-ийг л үүсгэж илгээж чаддаг байна (net/ipv4/icmp.c).



**Зураг 5 UDP хайлт** шүүгдсэн порт.



**Зураг 6 UDP хайлт** хаалттай порт.



**Зураг 7 UDP хайлт** нээлттэй порт.



**Зураг 8 UDP хайлт** нээлттэй | шүүгдсэн порт.

* **SCTP INIT хайлт**

Энэ арга нь SCTP-гийн хувьд TCP SYN хайлтын аргатай ижил арга гэж хэлж болно. Энэ арга нь маш хурдан бөгөөд секундэд мянга мянган портыг хайх боломжтойгоос гадна галт ханын хязгаарлалтанд ордоггүй. Мөн түүнчлэн нээлттэй, хаалттай болон шүүгдсэн портуудыг ялгааг сайн гаргаж өгдөг гэх мэт давуу тал олонтой юм. Энэ аргыг хагас нээлттэй хайлт ч гэж нэрлэх нь байдаг учир нь энэ арга SCTP бүрэн холболт хийдэггүй юм.

Клиент талаас INIT chunk илгээдэг бөгөөд үүний хариуд INIT-ACK ирвэл порт нээлттэй. ABORT chunk ирвэл порт хаалттай харин хэд хэдэн дахин дамжуулалтын дараагаар ямар нэгэн хариу илгээхгүй эсвэл ICMP (type 3, code 1, 2, 3, 9, 10, or 13) төрлийн хүршгүй алдаануудын аль нэг нь ирвэл энэ порт шүүгдсэн гэсэн үг болно.



**Зураг 9 SCTP INIT** хайлт порт нээлттэй.



**Зураг 10 SCTP INIT** хайлт порт хаалттай.



**Зураг 11 SCTP INIT** хайлт порт шүүгдсэн.

* **TCP NULL, FIN болон Xmas хайлт**

Эдгээр аргуудыг мөн хулгайгаар хайх арга ч гэж нэрлэдэг(stealth scan). Хэрэв порт нээлттэй бол ямар нэгэн хариу илгээхгүй. Харин порт хаалттай бол SYN, RST эсвэл ACK гэсэн хариуг илгээнэ.

**Null хайлт**

Ямарч флагийн битийг идэвхижүүлэхгүй (TCP флагийн толгой 0 байна)

**FIN хайлт**

TCP-ийн FIN флагийг идэвхижүүлнэ.

**Xmas хайлт**

ТСР-ийн FIN, PSH болон URG флагийг идэвхижүүлнэ.

Эдгээр аргууд нь шинж чанар нь төстэй бөгөөд хэрэв RST пакет хүлээн авбал порт хаалттай. Ямар нэгэн хариу илгээхгүй байвал порт нээлттэй эсвэл шүүгдсэн гэсэн үг болно. Хэрэв ICMP (type 3, code 1, 2, 3, 9, 10, or 13) төрлийн хүршгүй алдаануудыг аль нэгийг хариу илгээж байвал энэ нь тухайн порт шүүгдсэн гэсэн үг юм.

Энэ аргын давуу тал нь галт хана болон пакет шүүлтүүртэй чиглүүлэгчийг эвтэйхэн хуурч чаддаг боловч галт хана шүүлтүүрийг давах тийм ч сайн арга биш бөгөөд SYN аргаас арай л илүү хуурах чадвартайд тооцогддог. Нээлттэй болон шүүлтүүр хийгдсэн аль нь болохыг нарийн тогтоож чаддаггүй.



**Зураг 12 stealth** хайлт порт нээлттэй эсвэл шүүгдсэн.



**Зураг 13 stealth** хайлт порт хаалттай.



**Зураг 14 stealth** хайлт порт шүүгдсэн.

* **TCP ACK** **хайлт**

Энэ арга маань нээлттэй эсвэл нээлттэй|шүүгдсэн гэсэн тодорхойлолт өгдөггүй бөгөөд галт ханын нөхцөлт дүрэм болон ямар ямар порт шүүгдсэн байгааг тодорхойлдог байна. Энэ арга нь зөвхөн ACK флагийг идэвхижүүлдэг. Түүний хариуд порт нээлттэй ч бай үгүй ч бай RST пакет ирэх бөгөөд үүнийг шүүгдээгүй портууд гэж ойлгоно. Харин хариу өгөхгүй эсвэл ICMP (type 3, code 1, 2, 3, 9, 10, or 13) гэсэн алдааг өгч байвал эдгээр портуудыг шүүгдсэн портууд гэж үздэг.



**Зураг 15 ACK** хайлт шүүгдээгүй порт.



**Зураг 16 ACK** хайлт шүүгдсэн порт.

* **TCP цонхоор** **хайх**

Цонхоор хайх нь үнэн хэрэгтээ АСК хайлтын аргатай бараг ижилхэн ба ихэвчлэн шүүгдээгүй портыг илрүүлдэг. Өөрөөр хэлбэл энэхүү арга нь клиент талаас АСК флаг идэвхижүүлсэн пакет илгээдэг. Хэрэв RST гэсэн хариу ирвэл түүнийг шүүгдээгүй порт гэж үзэх ба үүнийхээ дараа тухайн пакетын ТСР цонхны хэмжээг шалгаж үзэх ба хэрэв цонхны хэмжээ 0 байвал порт хаалттай тэгээс их байвал портыг нээлттэй гэж үздэг.



**Зураг 17 TCP window хайлт** шүүгдээгүй порт.



**Зураг 18 ТСР window хайлт** порт хаалттай.



**Зураг 19 ТСР window хайлт** порт нээлттэй.



**Зураг 20 ТСР window хайлт** шүүгдсэн порт.

* **TCP Maimon** **хайлт**

Энэ нь NULL, FIN болон XMAS аргуудтай төстэй бөгөөд FIN/ACK-ийг оролцуулахгүй. RFC 793(TCP) стандартад порт нээлттэй эсвэл хаалттайгаас үл хамааран RST пакет илгээдэг гэдэг боловч яг амьдрал дээр ихэнх BSD үүсмэл системүүд хэрэв порт нээлттэй байвал тухайн пакетыг шууд хаядаг болох нь ажиглагджээ. Энэ ажиглалтыг хийсэн хүнийг Uriel Maimon гэдэг ба түүний нэрээр энэ хайлтын аргыг нэрлэсэн байна.



**Зураг 21 Maimon хайлт** порт нээлттэй.



**Зураг 22 Maimon хайлт** порт хаалттай.

* **SCTP COOKIE ECHO** **хайлт**

Энэ арга нь ахисан түвшний SCTP хайлтын арга юм. Клиент талаас COOKIE ECHO chunks илгээх ба сервер талд түүнийг чимээгүй хаявал порт нээлттэй. Харин буцаагаад ABORT илгээвэл порт хаалттай гэж үздэг. Энэ арга нь хуурах арга нь сайн боловч сайн IDS төхөөрөмжүүд үүнийг илрүүлдэг. Гэвч энэхүү арга нь танд нээлттэй порт болон шүүгдсэн портын ялгааг хэлж өгч чадахгүй бөгөөд танд нээлттэй шүүгдсэн гэсэн хариуг л үзүүлж чадна.



**Зураг 23 SCTP COOKIE ECHO** хайлт нээлттэй|шүүгдсэн порт.



**Зураг 24 SCTP COOKIE ECHO** хайлт хаалттай порт.

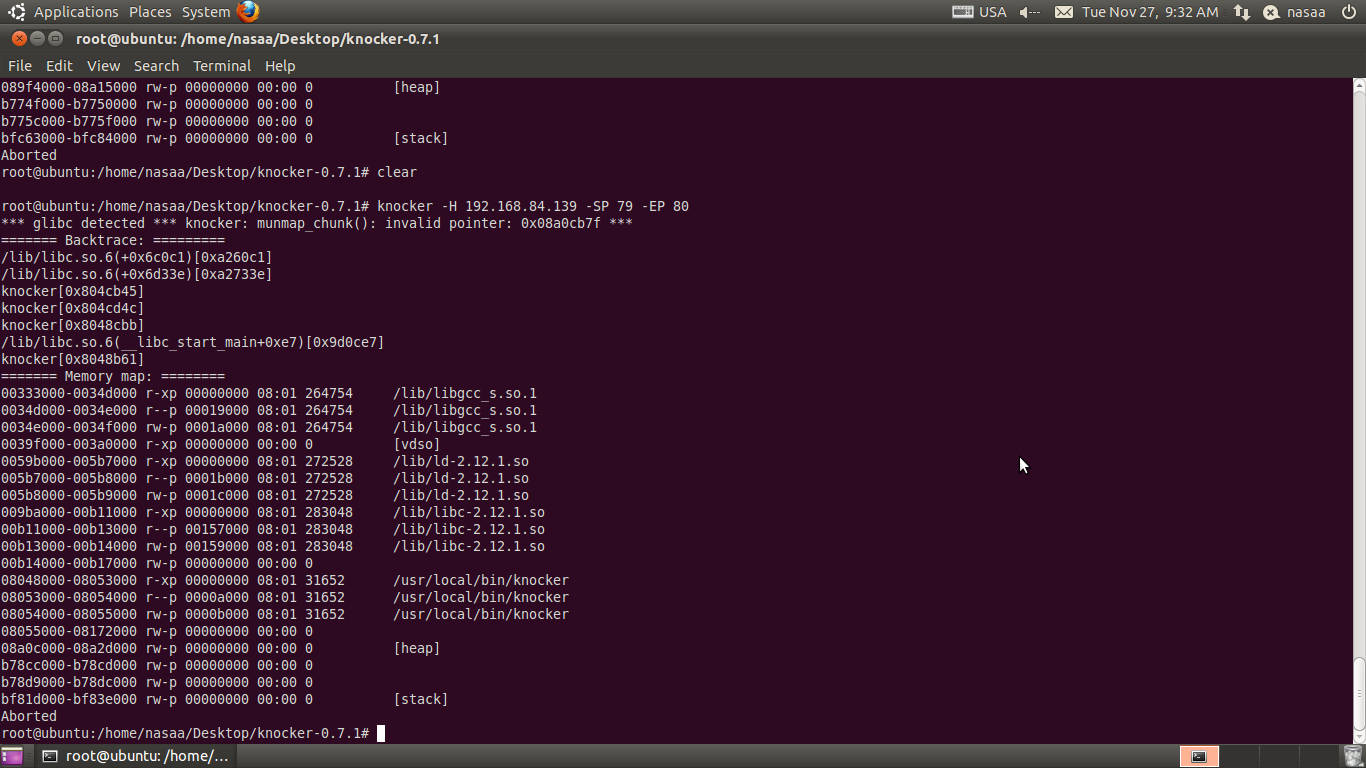
1. **ТУРШИЛТ**

**Туршилт хийсэн орчин:**

Windows 7 үйлдлийн систем дээр VMWare workstation суулгаж түүн дээр суурилсан виртуал орчинд дараах үйлдлийн системүүдийг суулгаж тухайн сүлжээнээс дотоод болон гадаад сүлжээнд хайлт хийсэн.

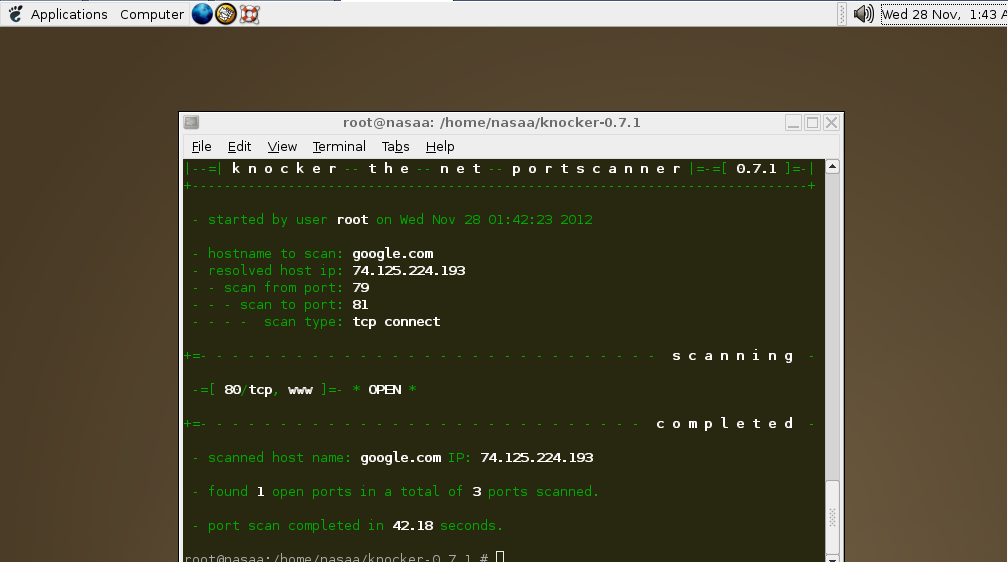
* Ubuntu 10.10
* Ubuntu 4.10

Эхлээд өөрийн төслийн Knocker нээлттэй порт илрүүлэгч програмыг Ubuntu 10.10 үйлдлийн систем дээр суулгаж ажиллуулсан боловч санах ой хувиарлалтын алдаа гарсан юм.

**Зураг 25.** хамгийн эхэнд Ubuntu10.10 дээр компайл хийж ажиллуулахад ажилах үедээ гарсан алдаа.

**Зураг 25:** Эндээс харахад энэхүү прорамд маань санах ой хувиарлалт болон стектэй холбоотой асуудал үүссэн нь түүний тайлбараас харагдаж байгаа бөгөөд энэ алдааг нь хайж үзэхэд тухайн кодонд оруулсан санах ой хувиарлах функцүүд ажиллахдаа үйлдлийн системийн цөмтэй торирохгүй байгаа гэсэн (bug) байсан.

Ийм алдаа заасан учраас би өөрийн кодын тохирох цөмтэй линукс үйлдлийн системийг хайж эхэлсэн. Ингэж хайсны үр дүнд миний код Ubuntu 4.10 гэсэн хувилбар дээр алдаагүй ажиллаж байгааг олж мэдсэн бөгөөд түүн дээр өөрийн кодыг ажиллуулж үзсэн.



**Зураг 26**. Ubuntu 4.10 програм ямар нэгэн алдаа заалгүй ажиллаж байгаа байдал

**Зураг 26:** Энэхүү зургаас knocker нээлттэй порт илрүүлэгч програм нь Ubuntu 4.10 үйлдлийн систем дээр ямар нэгэн файл хувиарлалт болон стекийн алдаа заалгүй зөв ажиллаж байгаа нь харагдаж байна.

Ubuntu 4.10 үйлдлийн систем дээр кодоо зөв ажиллуулснаас хойш би өөрийн кодонд анализ хийж илүү зүйлсийг нь хассан. Ингэснээр код маань хэмжээний хувьд жижигхэн болсон ба ойлгоход болон өөрчлөлт хийхэд хялбар дөхөм болж өгсөн. Ингэж илүү хэсгүүдийг хассаны хамгийн гол давуу тал нь тухайн функцүүдэд орж байсан санах ой хуваарилах функц хагадсан ба үүний үр дүнд санах хуваарилалтын алдаа заахаа болисон. Өөрөөр хэлбэл ихэнх цөм дээр алдаагүй ажиллаж эхэлсэн.

Програмаас ямар ямар хэсгүүдийг нь хассаныг дээр ***хүснэгт 1****-*т үзүүлсэн болно.

**Хүснэгт 1**

Өөрийн кодон дотроос хассан функцүүдийн жагсаалт

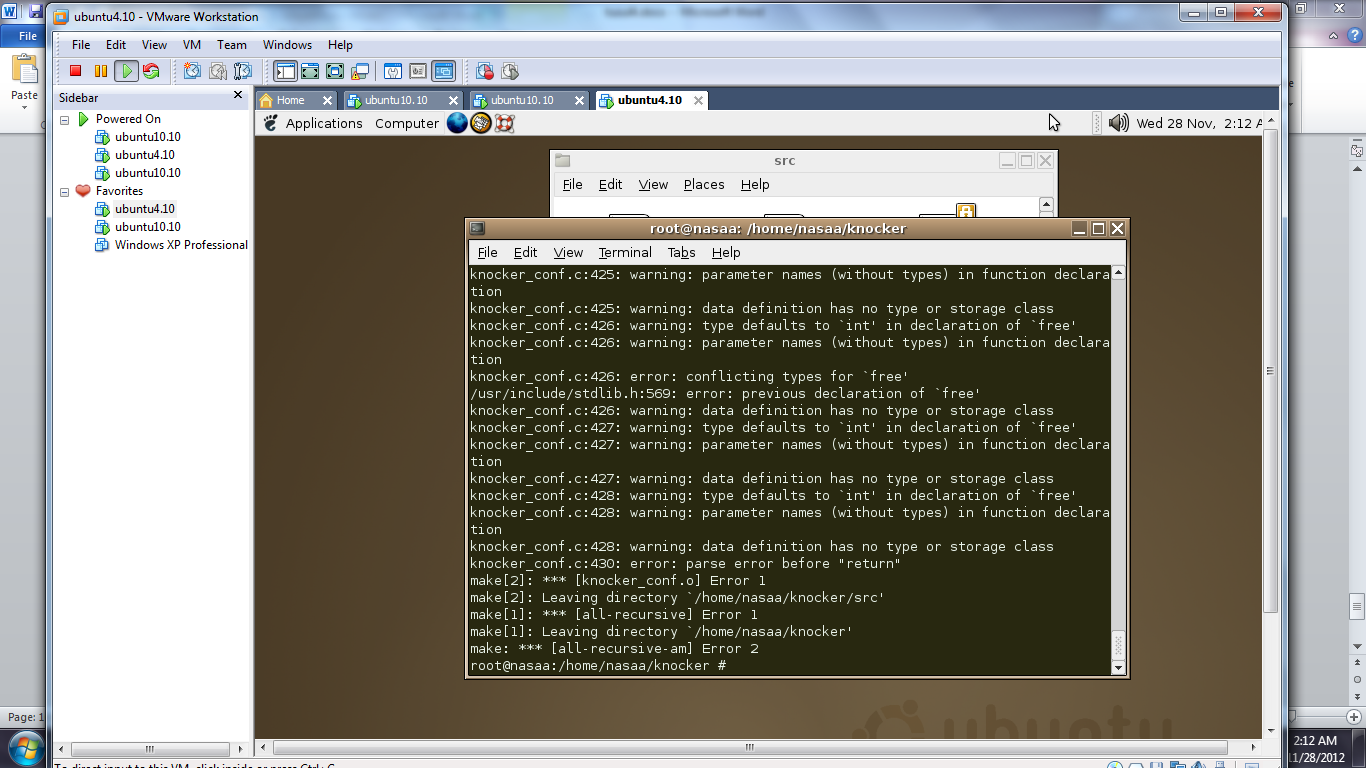
|  |  |
| --- | --- |
|  | Log файл руу бичиж байгаа хэсэг |
|  | Time Нийт хайлт хийсэн хугацааг нь бодох хэсэг |
|  | Service үсгийн фонтын өнгө солих гэх мэт жижиг хэсгүүд |
|  | Windows, freebsd гэх мэт миний ажилд чухал хэрэгтэй бус цөмүүдэд таарах нөхцөл шалгалтын функцүүдийг хаасан (#ifdef \_\_WIN32\_\_ etc…); |

Эдгээр хэсгүүдийг нь хасаж өгснөөр эх кодон дээр байсан malloc() буюу санах ой хувиарлах функцийн Ubuntu 10.10 дээр тохирохгүй байсан үйлдлийн систем алдаа bug-г нь заахаа болисон. Үүний дараагаар миний код Ubuntu 10.10 дээр ажиллах боломжтой болсон ба өөрийн кодын өөрчлөлтийг Ubuntu 10.10 дээрх хийж эхэлсэн.

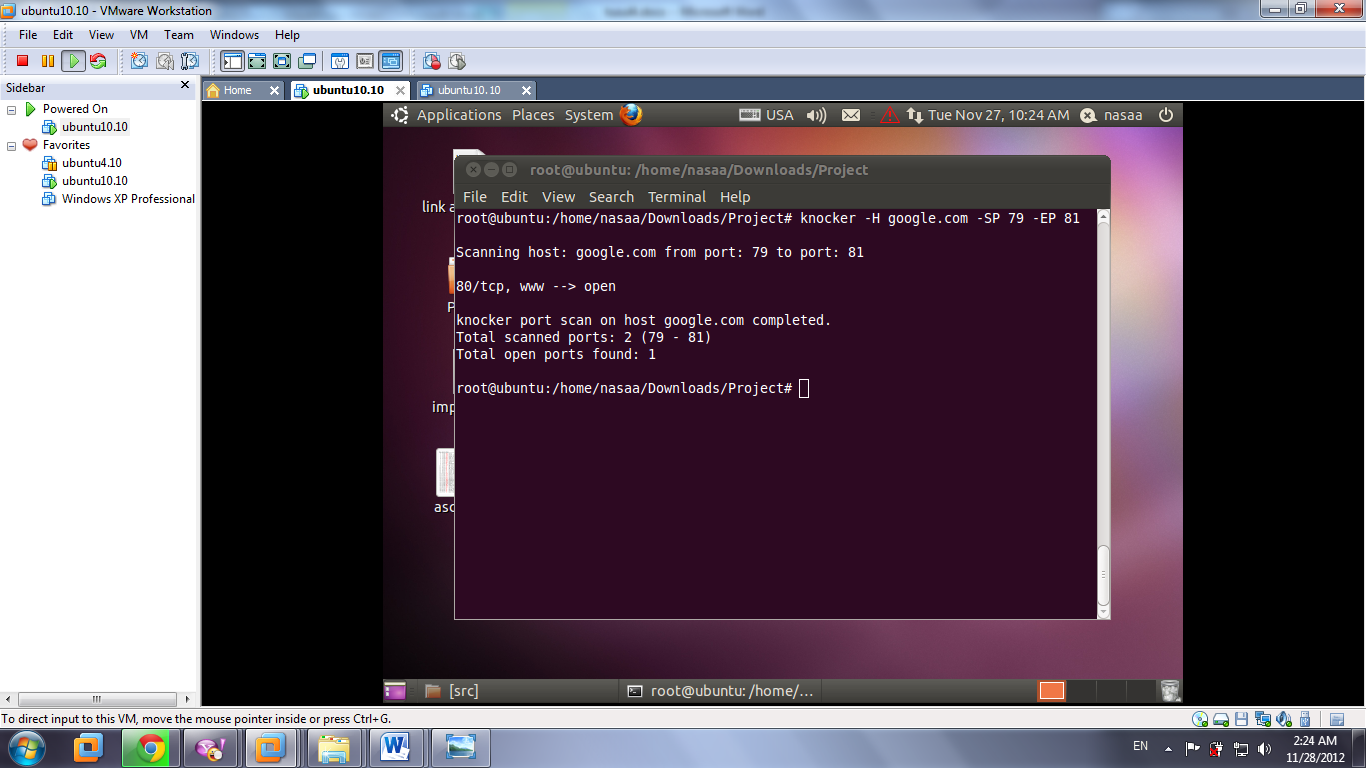
Чухал хэрэгцээт бус функцуудыг хасаж буй үйл явц:

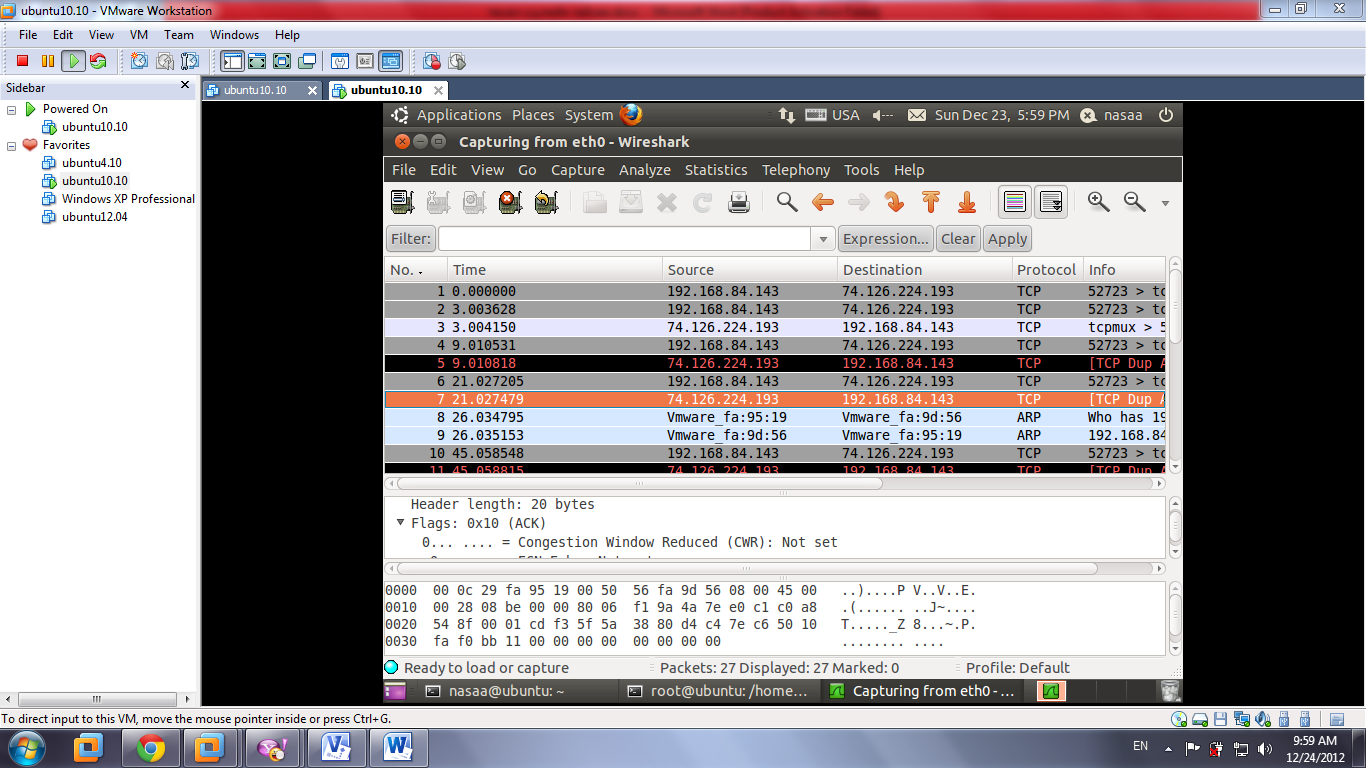


**Зураг 27** текстийн өнгө өөрчилж байгаа хэсгийг нь хасаж байгаа процесс



**Зураг 28** Чухал шаардлагатай бус элементүүдийг нь хасаж байгаа процесс

  
**Зураг 29** Чухал бус элементүүдийг хассаны дараа Ubuntu10.10 дээр ажиллаж байгаа байдал

**Зураг 30** Knocker нээлттэй порт илрүүлэгч програмаар хэрхэн google.com-н нээлттэй портуудыг илрүүлж байгаа процесс



**Зураг 31** Энэхүү зурган дээр knocker нээлттэй порт илрүүлэгч програмаар өөрийнх нь loopback интерфэйсийнх нь нээлттэй портуудыг нь илүүрэх процесс явагдаж байна.

Зураг 30 болон 31-ээс харахад энэхүү програм маань бай руугаа SYN пакет илгээж түүнийхээ хариуд буцааж SYN авсан тохиолдолд тухайн портыг нээлттэй гэж үзэж байгаа бөгөөд эсрэг тохиолдлуудад тухайн портыг хаалттай гэж үзэж байгаа нь харагдаж байна.

1. **ТӨСӨЛ**

**Хүснэгт 2**

Хийсэн өөрчлөлтүүдийн жагсаалт болон тэдгээрийн хэрэгжүүлэлтүүдийн товч тайлбар

|  |  |  |
| --- | --- | --- |
| № | Өөрчлөлтүүд | Тэдгээрийн үүрэг |
| 1. | Удирдлагын команд  -IPR | -IPR гэсэн утыг шалгаж түүнээс хамаарч хаягийн завсарт хайх функц руу шилжинэ. |
| 2. | Гараас оруулсан string утгыг авч ASCII код руу нь хөрвүүлнэ. | Тухайн хаягийн дээд доод заагийг нь бүхэл тоон хэлбэрт шилжүүлж авахын тулд хийж байгаа туслах үйлдэл |
| 3. | ASCII кодоор авсан IP хаягийн завсраас “-” гэсэн тэмдэгтийг хайна. | “-” гэсэн тэмдэгтийн хоёр талын утга нь дээд доод заагийг илтгэнэ. Тиймээс энэ утгыг шалгасанаар IP хаягийн дээд доод заагийг нь авах боломжтой болно. |
| 4. | Хамгийн сүүлийн “.” Гэсэн тэмдэгтийн байрлалыг хадгалж авах | Дээд болон доод заагийн хооронд давтаад үүссэн үр дүнг энэхүү тэмдэгтийн ард нь залгаж өгөх учир энэ тэмдэгтийн байрлалыг нь хадгалж авч байна. |
| 5. | IP хаягийн дээд болон доод заагийг авах (h-дээд зааг, l-доод зааг) | Орон оронгоор нь уншиж авсан IP хаягийн дээд болон доод заагийг бүхэл тоон хэлбэрт шилжүүлж авна. Өөрөөр хэлбэл int хэлбэрт оруулж авж байгаа хэсэг. |
| 6. | IP хаягийн заагийн хооронд давтаж үүссэн утгыг нь ASCII кодоос буцааж string рүү шилжүүлж IP хаяг болгож авна. | IP хаягийн дээд болон доод заагийн хооронд давтаад түүний утга бүрийг нь орон оронгоор нь салгаж буцааж ASCII код болгож string-ийг угсарна. |
| 7. | -ST буюу ямар флаг ивэвхижүүлэхийг заах удирдлагын хэсэг | Үүнд:  FIN  XMAS зэрэг утгуудаас авна. |
| 8. | -SoIP буюу тухайн хайлт хийж байгаа хостын өөрийнх нь хаяг | Бүрэн бус (raw) соккет илгээх функц буюу sendto() функцэд тухайн хостын (source) IP хаягийг тодорхойлох шаардлагатай байдаг. |
| 9. | Соккет үүсгэж илгээж байгаа хэсгийг өөрчилж соккетийг бүрэн бус (raw) соккет болгосон. | Соккетыг бүрэн бус (raw) болгосноор тухан соккетод ТСР толгой зарлаж өгөн түүний флагуудыг өөрчлөх боломжтой болно. |
| 10. | Бүрэн бус (raw) соккетод оруулах ТСР флагуудыг нөхцөл шалгаж зааж өгөх. | Хайлтын төрлүүдийг тодорхойлох хэсэг. |
| 11. | Ирсэн хариуг шалгах | Ирсэн хариунаас шалтгаалж өөр үр дүн буцаана. |

Эхлээд өөрчлөлтийг хийж үзэхдээ өөрийн кодын оролтыг гараас утгаа тус тусад нь авдаг байдлаар хаягийн завсараа оруулахаар авч үзсэн. Өөрөөр хэлбэл энэ нь гараас утгыг дараах байдлаар авна гэсэн үг.

***Knocker –ST XMAS –SoIP 192.168.84.151 –H 192.168.84.150 –SP 1 –EP 1024***жишээний дагуу авахаар хийсэн.



**Зураг 32** Програмын блок диаграм

Төслийн ажлын хүрээнд хийгдсэн зүйлс:

1. IP хаягийн завсарт утга авдаг болох
2. TCP-гийн флаг өөрчилж дамжуулах
3. Энэ даалгаварыг биелүүлэхдээ гараас орж ирэх утгыг ASCII кодонд нь хөрвүүлж түүн дээрээ анализ хийн дээд доод хязгаарыг нь мэдэж авснаар тухайн завсартаа IP хаягуудаа давтаж түүнийхээ ASCII кодыг нь олж буцааж эвлүүлж өрөх замаар тухайн завсар дахь IP хаяг бүрийг string-ээр буцааж авч чадаж байгаа юм. Ингэж IP хаягийг давтаж өгөх бүртээ тус бүр дотор нь портын завсраа оруулж өгөөд шалгаж байгаа хэсгийг багтааж байгаа юм. Ийм зарчмаар ажиллаж байгаа тул миний програм маань IP хаяг тус бүр дээр эхнээс нь IP хаягуудыг нь шалгаж үзээд дараагийн IP хаяг руу орж тэрний портуулыг нь шалгах гэх мэт үргэлжлэн ажиллаж байгаа юм.
4. ТСР дотор флаг нэмэх ажлыг хийж гүйцэтгэхдээ өөрийн програмын хийж байсан connect() функцийг бүхэлд нь raw соккет болон TCP толгой зааж өгөх функцээр сольж ажиллуулсан. Ингэснээр тухайн бай болох хаяг руу TCP-ийн ямар ч флагыг нь идэвхижүүлэн илгээх боломжтой болж байгаа юм.

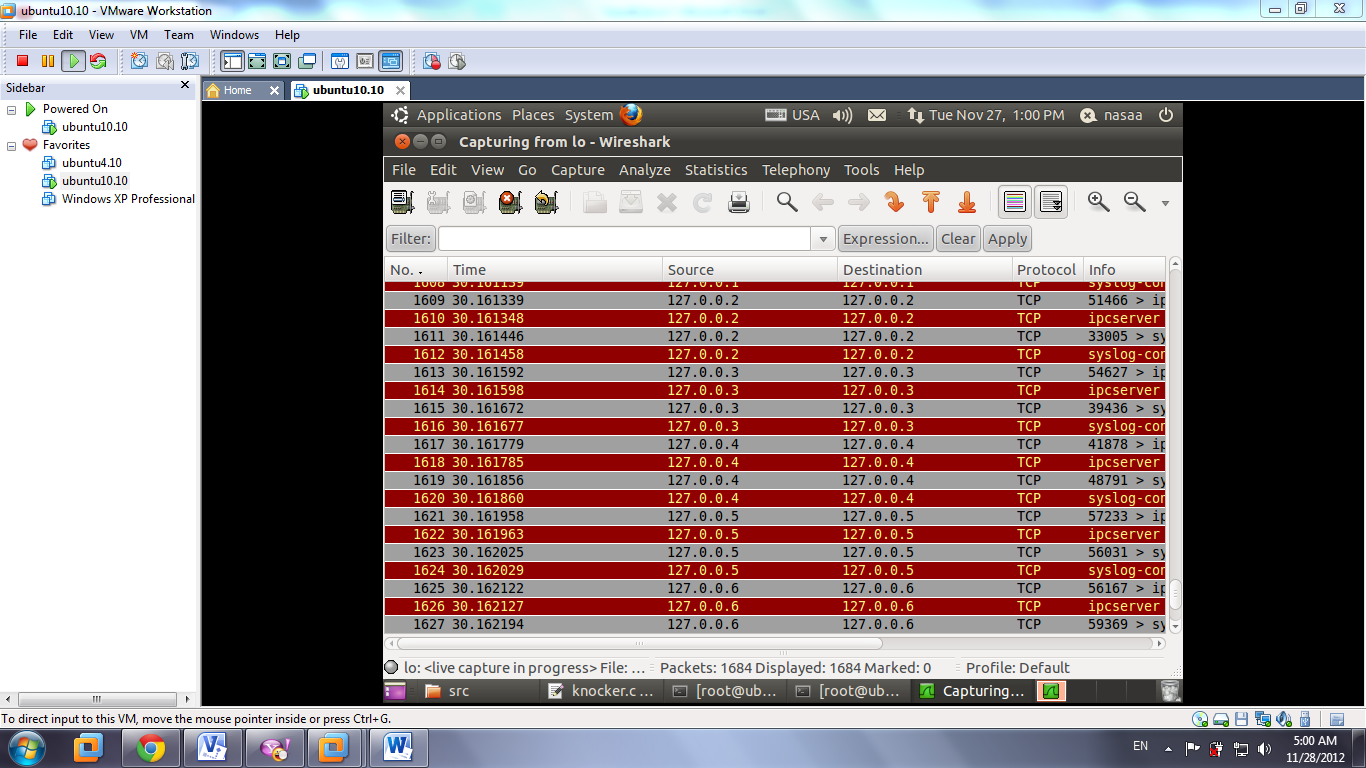
Жич: Бүрэн бус (raw) соккетыг үүсгэхэд супер хэрэглэгчийн эрхээр орсон байх шаардлагатай учир нь үйлдлийн системийн цөмд хийгддэг үйлдлийг хийх эрх олж авсан байх ёстой. Эдгээрийн хэрэгжүүлэлтийг хавсралтан дахь кодноос харна уу.

1. **ҮР ДҮН ТЕСТЧИЛЭЛ**

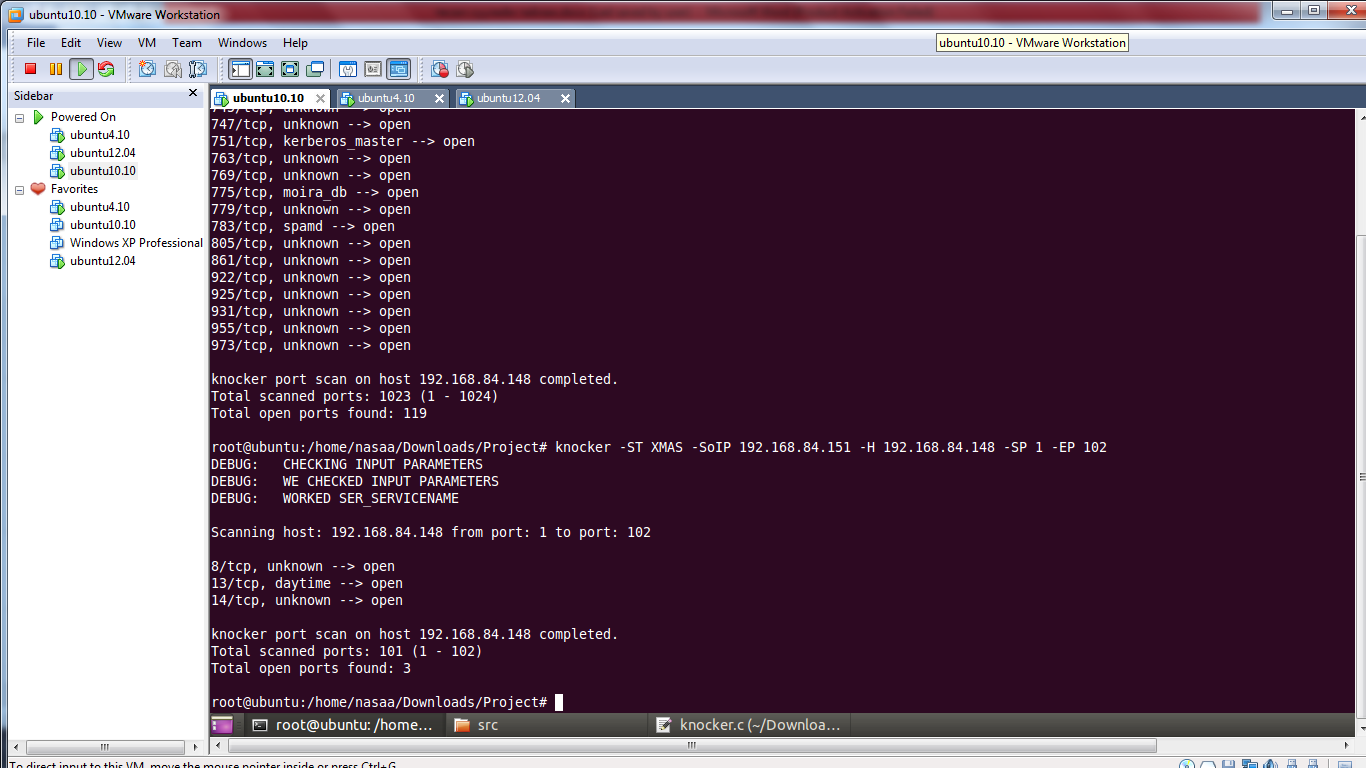


**Зураг 33** Энэхүү тестчилэл дээр 127.0.0.1-20 хүртэлх хаягийн завсарт 1-1000 хүртэлх портуудыг хайж байна.

Энэхүү зургаас харахад IP хаяг тус бүр дээр бүх портоор нь хайж байгаа нь илэрхий харагдаж байна. Харин дараах зурганд ингэж хайж байгаа үйлдлийнхээ пакетыг нь барьж авсан бөгөөд түүнийхээ үр дүнг орууллаа. Эхний даалгаварыг биелүүлсний дараах байдал.

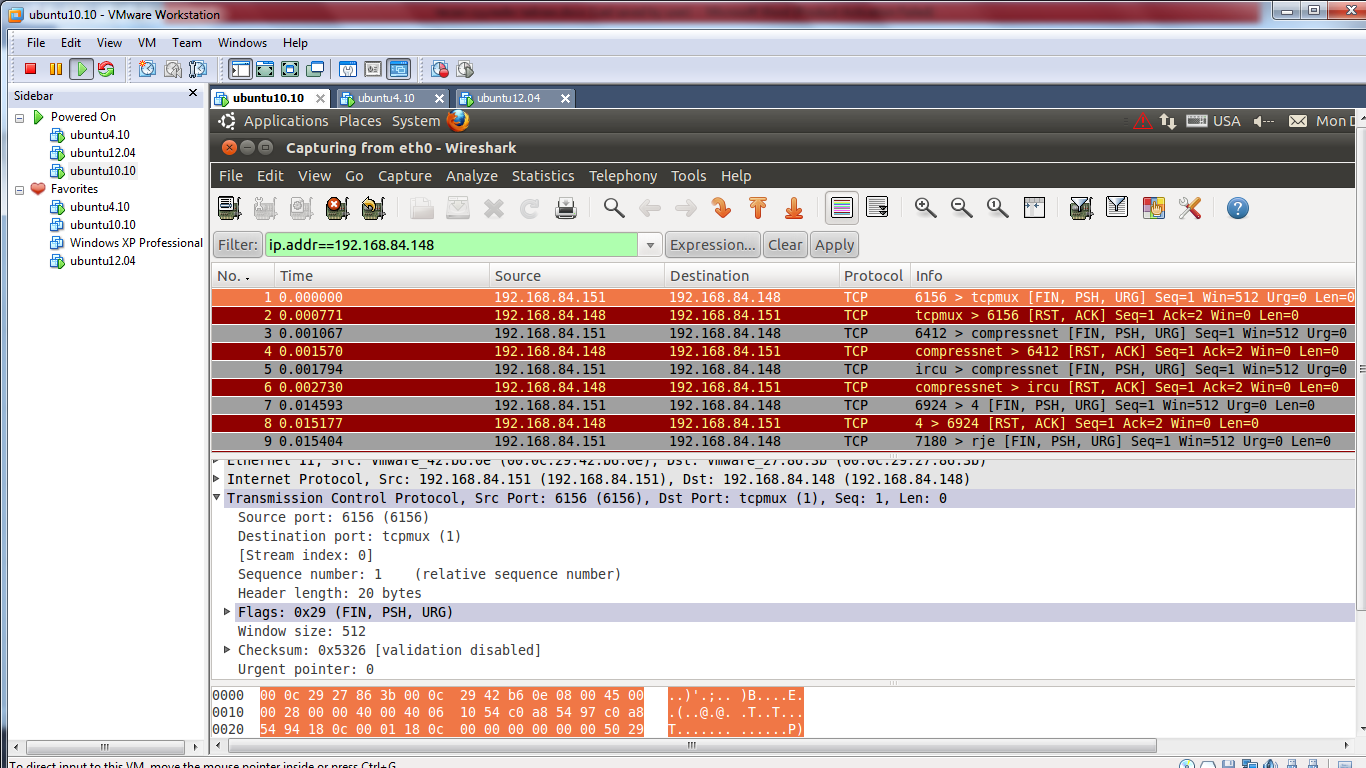


**Зураг 34** хаягийн завсраар хайж байгааг пакетийг нь барьсан байдал. Эхний даалгаварыг биелүүлсний дараах байдал түүний пакетууд



**Зураг 35** флаг өөрчилж (хайлтын арга) хайхад ажиллаж байгаа байдал

Энэхүү зурганд портуудыг нээлттэй|шүүлт хийгдсэн портуудыг ялгаж өгч байна.



**Зураг 36** XMAS аргаар хайж байгаа пакетууд.

**IV. ЕРӨНХИЙ ДҮГНЭЛТ**

Нээлттэй порт илрүүлдэг хэрэглээний програм хангамж нь хост эсвэл серверийн нээлттэй портыг нь судлах анализ хийх үйлдэл хийдэг сүлжээний анализ хийдэг хэрэгслүүдийн нэг хэсэг нь юм. Ийм хэрэгслийг ихэвчлэн сүлжээний админисртаторууд өөрийн сүлжээний аюулгүй байдлын бодлого хянах шалгах зорилгоор хэрэглэдэг. Харин халдагч этгээд нь одоо хийгдэж байгаа сервис үйлчилгээний тодорхойлолт цаашлаад хост дээр хэрхэн тохирч үйл ажиллгагаа явагдаж байгааг харах зорилгоор хэрэглэнэ.

Порт хайлтыг өөрөөр тодорхойлбол идэвхитэй байгаа портыг нээж олон түүнийгээ ашиглан тухайн портоор дамжигдан явах үйлчилгээний эмзэг байдлыг нь мэдэж авах зорилготой хостын тал дээр серверийн портыг тодорхой завсраар авч тэдгээр лүү хүсэлт илгээж байгаа дайралтын нэг хэлбэр юм. Хэдий тийм боловч үүнийг үндсэн хэрэглээ нь дайралт хийх бус алслагдсан машин дээр ямар ямар сервис хийж болох эсэхийг тодорхойлох илрүүлэхэд хэрэглэдэг.

Portsweep гэдэг нь олон хостууд тодорхой портыг чагнаж байх юм. Сүүлийн үед ерөнхийдөө тодорхой сервисүүдийг хайхад хэрэглэдэг болсон жишээлбэл SQL дээр суурилсан компьютерийн өт portsweep-ээр хостуудаас TCP 1433 гэсэн порт нь чагнах төлөвт байгааг нь хайж болно.

Интернэтийн загвар болон үйл ажиллагаа нь IP-internet protocol дээр суурилдаг. Ихэвчлэн TCP/IP гэж нэрлэдэг. Ингэж дамжигдаж байгаа хост болон хостын ямар нэгэн сервис нь хаяг болон портын дугаар гэсэн 2 хэмжигдэхүүнээр зохицуулагддаг. Ихэнх сервис-үүд хязгаарлагдмал завсарт дугаарлалт авдаг. 65535 хэрэглэгдэж болохуйц портын дугаарлалт байдаг. TCP порт нь дараах 3-н үр дүнгээс хамаарч хайж байгаа порт нээлттэй эсвэл хаалттай гэсэн үр дүнг илгээдэг.

1. *Open* or *Accepted*: Хост тал нь тухайн сервис авахыг хүссэн порт нь чагнах горим дээр байгаа порт гэдгийг илтгэх хариу илгээдэг.
2. *Closed* or *Denied* or *Not Listening*: Харин энэ нь хост тал холболтыг хийж болохгүй буюу хандалтыг авахыг татгалзсан байна гэсэн хариу илгээх
3. *Filtered*, *Dropped* or *Blocked*: Энэ тохиолдолд хариу ирэхгүй. Гэхдээ эдгээр нь тухайн ТСР толгойд ямар ямар флаг зааж өгснөөс хамааран ямар хариуг яаж ойлгох вэ гэдэг нь өөр өөр болж эхэлдэг. Энэхүү арга нь энгийн буюу зөвхөн холболт тогтоох процессд суурилсан жишээ юм.

Нээлттэй портууд нь эмзэг байдлын эх үүсвэр болдог ба администратор нь хянуур байх ёстой.

1. Аюулгүй байдал болон тогтвортой байдлын нөлөөлөл нь хүргэгдэж байгаа сервисүүд үр дүнтэй нийлсэн байна. – нээлттэй портууд
2. Аюулгүй байдал болон тогтвортой байдлын нөлөөлөл нь хост дээр ачаалагдаж байгаа үйлдлийн системээс хамааралтай байдаг. – нээлттэй эсвэл хаалттай портууд

Шүүлт хийсэн портууд нь эмзэг байдлын хандлагад нөлөөлөхгүй.

Ихэнх порт нээлттэй эсэхийг топорхойлж байгаа процесс нь RFC 793- TCP байна гэж үзэж хайлт хийдэг юм. Харин хостоос үл танигдах өөр пакет эсвэл буруу мэдээлэл өгөх боломж байдаг бөгөөд энэ нь хост талын стек нь TCP/IP биш эсвэл түүний TCP/IP-г өөрчилсөн байдлаар хамаарна.

Сүлжээний хайлт хийдэг цөөн хэдэн хүчирхэг хэрэглүүрүүд л IP хаягийн завсарт утга авч түүнийхээ утгыг нь шалгаж хариу өгч байсан юм. (жишээ нь nmap). Ингэж IP хаягийн завсарт авдаг болсноор сүлжээний амдинистратор маань богино хугацаанд дотоод сүлжээнийхээ бүх компьютерийн аюулгүй байдлын бодлого тохиргоог хянах боломж олгож байгаа юм. Яагаад гэвэл систем гэдэг ойлголтод маань дотоод сүлжээ түүний гарц, тухайн сүлжээнд байгаа компьютерууд гээд бүх л зүйл орох бөгөөд хэрэв эдгээр компьютеруудаас аль нэг нь л аюулгүй байдлын бодлогын доод шаардлагыг хангаж чадаагүй байвал энэ систем тэр чигтээ тийм ч найдвартай систем биш болж хувирна. Тиймээс тухайн системд байгаа комьютер бүр системд чухал нөлөө үзүүлж байгаа учир аюулгүй байдлын бодлогыг хангасны дараагаар сүлжээний анализ хийж үзэх хэрэгтэй болно. Энэ үед хэрэг сүлжээ маань олон компьютертэй тэдгээр нь физик байрлалаараа нэг дор байрлаагүй бол компьютер тус бүр дээр нь нэг бүрчлэн анализ хийж үзэхэд цаг хугацааны болон зардлын хувьд алдагдалтай юм. Тиймээс энэхүү асуудлыг шийдэхэд IP хаягийн завсарт авч сүлжээний анализ хийж байгаа програм нь чухал ач холбогдолтой програм хангамж болж байгаа бөгөөд цаг хугацаа болон зардлыг асар ихээр хөнгөвчлөх юм. Мөн төрөл бүрийн аргаар хайж үзсэнээр бүхий л эмзэг байдлын нүх сүвийг илрүүлэхэд дөхөм болох юм. Ийм байдлаар сүлжээний администраторын ажлыг асар их хөнгөвчилж өөрийн сүлжээг хянах боломж олгох хэдий ч гэсэн мөн халдагч этгээдэд давуу тал олгох зүйл энэ програмд маань байгаа бөгөөд яагаад гэвэл халдагч этгээд ч гэсэн ийм сүлжээний хайлт ашиглан хэрэглүүр ашиглан тухайн сүлжээний бүхий л компьютерт анализ хийх боломж бүрдэж байгаа юм.. Өөрөөр хэлбэл хэрэв сүлжээний олон зуун компьюьтерийн дунд ганц ширхэг аюулгүй байдлын хамгийн бага шаардлагыг хангаагүй компьютер байгаа гэж үзвэл халдагч тухайн компьютерийн хаяг болон мэдээлэл мөн түүнийг аюулгүй байдлын шаардлага хангаагүй байна гэдгийг цаг хугацааны хувьд асар хурдан мэдэх боломж олгох юм. Нэгэнт халдагч аюулгүй байдлын шаардлага хангаагүй хост олоод түүн рүүгээ нэвтэрсэн бол сүлжээнд байгаа бусад компьютерүүдтэй нэг ижил дотоод сүлжээнд байгаа мэтээр хандах боломжтой болно гэсэн үг. Тиймээс сүлжээний болон системийн админ хүн маш хянуур байж ялангуяа том сүлжээ хариуцаж байгаа тохиолдолд системийн бүхий л хэсэгт өөрийн хяналт зохицуулалтыг хийж байх хэрэгтэй. Ингэхийн тулд сүлжээний хайлт хийдэг эдгээр хэрэглүүрүүд нь маш чухал хэрэгтэй болж байгаа юм. Дээр дурдагдсан portsweep нь IP хаягийн завсарт хайж байгаа процессын тухайн тохиолдол юм.

Миний төслийн ажил хийж гүйцэтгэсэн энэхүү програмын хөгжүүлэлтийн хувьд өөр боломжууд байгаа бөгөөд жишээлбэл үйлдлийн системийг тодорхойлох сүлжээний энгийн топологи гаргах гэх мэт.

**V. АШИГЛАСАН БҮТЭЭЛИЙН ЖАГСААЛТ**

1. <http://en.wikipedia.org/wiki/TCP_and_UDP_port>
2. <http://en.wikipedia.org/wiki/Port_scanner>
3. W.Richard Stevens, Bill Fenner, Andrew M.Rudoff “UNIX Network Programming-The Sockets Networking API”, 2003 он.
4. Д. Цэдэвсүрэн **“**С**++** програмчлал”, Улаанбаатар 2002 он.
5. David Haskins “C Programming in linux”, 2009 он.
6. Nmap.org