Безпека операційних систем і комп'ютерних мереж

Лекція 14

Безпека протоколів стеку TCP/IP

Питання

- Безпека протоколу IP
- Безпека транспортних протоколів
  - UDP
  - TCP

Функції протоколу IP

- На рівні протоколу IP відпрацьовується передача пакета між мережами, і для цього передбачені ряд засобів:
  - Система глобальної адресації
  - Контроль часу життя пакета
  - Можливість динамічної фрагментації пакетів
  - Підтримка якості обслуговування
- Протокол IP відповідає лише за притаманні йому функції
  - Для передачі пакета всередині кожної з мереж, через які відбувається доставка, протокол IP звертається до засобів нижчого (канального) рівня
  - Питання гарантованої доставки пакета, його повторної передачі тощо — це справа засобів вищого (транспортного) рівня

Формат заголовка IP пакета

<table>
<tr><td>4 біта
Номер версії</td><td>4 біта
Довжина заголовка</td><td colspan="5">8 біт
Тип сервісу
PR | D | T | R</td><td colspan="3">16 біт
Загальна довжина</td></tr>
<tr><td colspan="7">16 біт
Ідентифікатор пакета</td><td colspan="2">3 біта
Прапорці
D | M</td><td>13 біт
Зміщення фрагмента</td></tr>
<tr><td colspan="2">8 біт
Час життя (TTL)</td><td colspan="5">8 біт
Протокол верхнього рівня</td><td colspan="3">16 біт
Контрольна сума</td></tr>
<tr><td colspan="10">32 біта
IP-адреса відправника</td></tr>
<tr><td colspan="10">32 біта
IP-адреса призначення</td></tr>
<tr><td colspan="10">Опції та вирівнювання</td></tr>
</table>

Атаки на протокол IPv4

- Атаки на протокол IPv4, що пов’язані з адресацією
  - Підміна адреси відправника (атака IP spoofing)
  - Атака Land
- Атаки, що ґрунтуються на помилках обробки фрагментованих пакетів
  - Перевищення максимально припустимого розміру IP пакета (атака Ping Death)
  - Використання таймауту очікування фрагмента для DOS-атаки
  - Атаки Teardrop і Bonk
  - Фрагментація IP як засіб проникнення через брандмауер

Підміна адреси відправника (атака IP spoofing)

- IP spoofing — це одна з найпростіших і разом з тим найпоширеніших атак
- Атака полягає в тому, що відправник замість своєї вказує деяку іншу IP-адресу
  - Передумовою можливості цієї атаки є той факт, що маршрутна інформація у пакеті IPv4 відсутня, а тому перевірити, звідки саме надійшов пакет, практично неможливо
- Ця атака компрометує ідентифікацію відправника пакетів за його IP-адресою
- Застосування IP-spoofing може бути корисним для порушника у багатьох випадках:
  - кінцевий вузол, який піддається атаці, надає певні права доступу деяким машинам, які визначаються за їхніми IP-адресами;
  - міжмережний екран, що стоїть на шляху пакетів, фільтрує їх за IP-адресою відправника;
  - у процесі здійснення атаки порушник намагається видати себе за декого іншого, наприклад, для уникнення відповідальності.

Обмеження атаки IP spoofing

- Застосовуючи IP spoofing, порушник має значне обмеження: якщо на його пакети повинні надходити відповіді, то він не може їх отримати — вони будуть надсилатись на ту IP-адресу, яка була вказана як адреса відправника у вихідному пакеті
  - Це не є проблемою при використанні дейтаграмних протоколів, таких як
    - ICMP,
    - OSPF,
    - тих, що використовують як транспорт UDP
      - протокол керування SNMP
      - протокол служби доменних імен DNS
  - В усіх зазначених випадках метою порушника може бути надсилання одного-єдиного пакета, який буде прийнятий за дозволений

Захист від атаки IP spoofing

- Фільтрація пакетів на міжмережному екрані (брандмауері) на вході у захищену мережу дозволяє значно зменшити можливості найбільш небезпечних підмін адрес
  - Типовим правилом є знищення всіх пакетів, які надходять із зовнішньої мережі, але мають зворотну IP-адресу, що належить внутрішній мережі
    - Такі пакети є спробою зовнішнього порушника видати себе за легального користувача з внутрішньої (захищеної) мережі
- Викорінення атаки IP-spoofing може бути досягнутим лише за умови контролю за проходженням пакетом певного маршруту
  - Тоді можна буде перевіряти, чи дійсно пакет надійшов звідти, звідки він удає

Атака Land

- Атака Land — це окремий випадок атаки IP- spoofing
  - Атака, незважаючи на її простоту, була здатна призводити у багатьох операційних системах до відмови в обслуговуванні:
    - спостерігалось значне навантаження процесора, аж до 100%, а в деяких системах — збої або зависання (UNIX-системи демонстрували “kernel panic”, а Windows — “блакитний екран”)
- Сутність атаки полягала в тому, що
  - в IP-пакеті вказувалась IP-адреса жертви і як адреса призначення, і як адреса джерела
  - як транспортний протокол використовувався TCP (це змушувало систему намагатись відповісти на отриманий пакет з метою встановлення з’єднання)
  - у заголовках TCP портів джерела і призначення вказувався один і той самий порт — будь-який з відкритих портів у системі
    - В результаті система намагалась відповісти собі самій
- Хоча така атака здається досить очевидною, перші відомості про неї датовані лише 1997 роком, і в той час уразливими виявились багато різних систем
- Оскільки атака є фактично атакою на відмову в обслуговуванні, ніщо не заважає підсилити її і замість окремого пакета надіслати значну кількість таких пакетів (спрямований шторм Land-запитів)
- Оскільки атака Land відома вже давно, у сучасних системах нештатні ситуації під впливом цієї атаки виникати не повинні.

Атаки, що ґрунтуються на помилках обробки фрагментованих пакетів

- Алгоритм збирання фрагментованих пакетів не мав завдання виявлення зловживань, і тому в деяких реалізаціях некоректно відпрацьовував спеціально підготовлені порушником фрагменти
  - Результатом було зависання комп'ютерів, "kernel panic" у UNIX та Linux, "блакитний екран" у Windows, або перезавантаження
- Розробники RFC хоча й не передбачали зловмисного приготування спеціальним чином фрагментованих пакетів, все ж передбачали можливість перекриття фрагментів і запропонували шлях уникнення неоднозначностей

Процедура збирання фрагментованих пакетів згідно з RFC 791 (1/2)

- Для того, щоби зібрати фрагменти, модуль IP відбирає дейтаграми, які мають однакові значення чотирьох полів у заголовках:
  - ідентифікатор,
  - адресу джерела,
  - адресу призначення
  - тип протоколу.
- Для збирання пакетів виділяються такі ресурси:
  - буфер даних,
  - буфер заголовка,
  - бітова таблиця блоків фрагментів (один блок = 8 байтів),
  - поле загальної довжини,
  - таймер.

Процедура збирання фрагментованих пакетів згідно з RFC 791 (2/2)

- Дані з фрагмента розміщуються у буфері даних відповідно до зміщення цього фрагмента і його довжини, і при цьому встановлюються відповідні біти у таблиці блоків фрагментів
  - Таким чином під час збирання відстежується заповнення буфера даних
- Вимог до послідовності надходження пакетів не встановлюється.
- Якщо фрагмент є першим (тобто, його зміщення дорівнює нулю), то саме його заголовок копіюється у буфер заголовка
- У випадку, коли два або більше фрагментів містять одні й ті самі дані, або ідентичні, або такі, що частково перекриваються, у підсумкову дейтаграму слід вставити ту копію даних, що надійшла пізніше

Перевищення максимально припустимого розміру IP пакета (атака Ping Death)

- Максимальний розмір IP-пакета становить 65535 байтів разом із заголовком
- Програмні засоби, що реалізовували стек TCP/IP на кінцевому вузлі, виявились неспроможними коректно обробляти ситуацію, коли надходив пакет більшого розміру
  - При цьому відбувалось переповнення буфера і усі можливі його наслідки
- Такий пакет міг опинитись у буфері завдяки фрагментації пакета
- Атака Ping Death використовувала exo-запити ICMP (ping)
  - У першому варіанті атаки була запропонована проста маніпуляція з параметром довжини ping-пакета
  - Якщо вказати довжину у 65527 байтів
    - команда ping –l 65527 victim, де **victim** — це IP-адреса або доменне ім'я комп'ютера-жертви
  - то загальна довжина IP-пакета буде становити 65555 байтів
    - до заданої довжини додається 8 байтів заголовка ICMP і 20 байтів заголовка IP
- Вже у самому заголовку IPv4 закладена можливість перевищення довжини пакета
  - Повна довжина пакета підраховується разом із заголовком, в той час як зміщення фрагмента відраховується у полі даних
  - Максимальне значення відповідного поля складає $2^{13}-1$, а отже максимальне зміщення — $(2^{13}-1)*8 = 65528$ байтів, що разом із заголовком (щонайменше 20 байтів) вже виходить за межі дозволеного розміру дейтаграми
  - До цього ще треба додати довжину поля даних фрагмента!
- Дивно, але в RFC-791 стосовно збирання дейтаграми не згадується перевірка виходу за дозволені межі її розміру, тобто це питання вирішують розробники ОС

Можливість простої DOS-атаки

- Відсутня можливість перевірити, чи всі фрагменти дейтаграми вже надійшли
  - Єдина можлива перевірка — це контроль заповнення “вікон” від початку поля буфера збирання до кінця фрагмента, який позначений, як останній
  - Якщо ж залишаються незаповнені вікна, то процедура буде чекати на наступні фрагменти впродовж встановленого тайм-ауту
  - Якщо у заголовку фрагмента було встановлено TTL=255, то саме це значення в секундах буде прийнято як тайм-аут
  - Протягом понад чотири хвилини ресурси комп’ютера, включаючи зарезервований буфер, будуть зайнятими!!!
- Достатньо створити два фрагмента:
  - перший будь-якого розміру
  - другий, що помічений як останній, з великим значенням зміщення і TTL=255
- Якщо надіслати значну кількість таких пар фрагментів (міні-шторм), то можна досягти переповнення черги і блокування машини, яку атакують

Атаки Teardrop і Bonk

- Фрагмент коду:

```
end = (offset + total_len) — ihl;
if (prev!=NULL && offset<prev–>end)
{
    i=prev–>end - offset;
    offset += i;
    ptr += i;
}
fp–>offset = offset;
fp–>end = end;
fp–>len = end - offset;
memcpy((ptr+fp–>offset), fp–>ptr, fp–>len);
```

- Якщо другий фрагмент повністю потрапить всередину першого, то
  end<offset, і fp–>len стає від'ємною
- Таку перевірку розробники не передбачили!
  структура **fp** заповнювалась відповідними даними і відбувався виклик функції memcpy()
  функція сприймала третій параметр, що мав від'ємне значення, як велике додатне ціле
  в результаті здійснювалась спроба копіювання занадто великої області пам'яті, що призводило до краху системи

Фрагментація IP як засіб проникнення через брандмауер (1/2)

- Ідея атаки
  - Під час аналізу заголовків брандмауер спочатку визначає, чи не є досліджуваний пакет фрагментом
  - Якщо ні — то за заголовком IP, з якого визначають як мінімум адреси джерела та призначення і тип протоколу, повинен знаходитись заголовок відповідного транспортного протоколу, наприклад, TCP
  - З нього брандмауер визначає порти джерела і призначення (а також, якщо необхідно, іншу інформацію)
  - Ідея полягала у тому, що перший пакет повинен містити у заголовку TCP дозволені значення портів, а наступний пакет, що позначений як фрагмент, повинен мати мінімальне зміщення і містити в собі нові, недозволені параметри (головною метою, звичайно, є підміна порту призначення)
  - Вперше на цю вразливість у 1995 р. вказав Фред Коен (Fred Cohen) у своєму онлайн-журналі “Internet Holes”
- Насправді
  - У такому вигляді загроза є міфічною

Фрагментація IP як засіб проникнення через брандмауер (2/2)

- Обґрунтування міфічності загрози
  - І кінцеві вузли, і брандмауери визначають перший фрагмент пакета за нульовим зміщенням. Тобто, якщо вказати зміщення “0”, то пакет буде сприйнятий, як перший фрагмент, і брандмауер проаналізує номери портів, що сховані у його заголовках
  - Мінімальне зміщення, яке можна вказати для фрагмента, що не є першим, — це “1”. Це зміщення на вісім байт
  - У заголовку TCP номери портів розміщені таким чином, що перекрити їх шляхом збирання IP-пакета з фрагментів неможливо
  - Оскільки заголовок протоколу UDP має повну довжину у 8 байтів, модифікація жодних його полів подібним шляхом взагалі неможлива
- Але
  - Повна (найменша, без опцій) довжина заголовка TCP становить 20 байтів, і останні 12 з них саме таким чином можна модифікувати
  - З тих полів, які можуть бути модифіковані, потенційно цікавими для порушників є прапорці SYN, ACK, RST, FIN
  - Певні нестандартні комбінації цих прапорців у минулому використовувались для здійснення “прихованого” сканування портів і навіть для деяких атак

Транспортний протокол UDP

16 біт	**Номер порту відправника**	**16 біт**	**Номер порту одержувача**
16 біт	**Довжина повідомлення**	**16 біт**	**Контрольна сума**

- Протокол UDP є дуже простим дейтаграмним протоколом
  - Стандарт його визначений у RFC 768
  - Його основні функції обмежуються мультиплексуванням і демультиплексуванням інформаційних потоків
  - Ідентифікація програм-відправників і програм-одержувачів здійснюється за номерами UDP-портів
- Можна констатувати, що протокол UDP не має вбудованих засобів безпеки
  - Він повністю покладається на ідентифікацію відправника засобами мережного рівня, тобто за IP-адресою
  - 16-розрядна контрольна сума захищає від помилок при передаванні даних, але жодним чином не захищає від цілеспрямованої модифікації даних
  - Протокол UDP не гарантує доставку і не має засобів контролю доставки — це покладається на протокол вищого (прикладного) рівня
- Протокол UDP виконує покладені на нього функції. Слід добре розуміти його обмеження і не покладатись на нього, коли міркування безпеки є суттєвими.
  - Цей протокол дуже часто використовують як транспортний для різних протоколів керування мережею.
  - Розробники протоколу керування повинні чітко розуміти, що цей транспортний протокол є ненадійним і не захищає від підроблення даних і атрибутів відправника, тому засоби захисту необхідно включити до самого протоколу керування.

Безпека протоколу TCP

- Протокол TCP описаний у RFC 793
  - Протокол TCP є протоколом із встановленням логічного з'єднання
  - У межах з'єднання здійснюються
    - реєстрація послідовності пакетів,
    - підтвердження доставки кожного пакета,
    - керування потоком пакетів,
    - повторна передача спотворених пакетів.
  - Деякі із зазначених функцій надають сервіс із захисту від підміни суб'єктів з'єднання
    - деякі протоколи прикладного рівня, в тому числі FTP, TELNET, HTTP, що надають користувачам віддалений доступ, використовують саме TCP як транспортний протокол
    - TCP як транспортний протокол використовує один з основних протоколів маршрутизації в Інтернеті — BGP
    - деякі служби, як, наприклад, DNS, можуть використовувати (як транспорт) і UDP, і TCP
      - як правило, дають рекомендацію щодо обов'язкового використання саме TCP для підвищення стійкості служби проти атак

Формат заголовка TCP-пакета

<table>
<tr><td colspan="3">16 біт
Номер порту відправника</td><td>16 біт
Номер порту одержувача</td></tr>
<tr><td colspan="4">32 біт
Номер послідовності</td></tr>
<tr><td colspan="4">32 біт
Номер підтвердження</td></tr>
<tr><td>4 біт
Довжина заголовка</td><td>4 біт
Зарезерво-вано</td><td>8 біт
Прапорці</td><td>16 біт
Розмір вікна</td></tr>
<tr><td colspan="3">16 біт
Контрольна сума TCP</td><td>16 біт
Покажчик терміновості</td></tr>
<tr><td colspan="4">Опції та вирівнювання</td></tr>
</table>

Керування з’єднанням у TCP

- У заголовку присутні командні біти — прапорці, за допомогою яких здійснюється керування з’єднанням. Для нашого розгляду суттєві:
  - ACK (Acknowledgment field significant) – квітанція на сегмент, що був прийнятий;;
  - RST (Reset the connection) -– запит на відновлення з’єднання;
  - SYN (Synchronize sequence number) – повідомлення, що використовується для синхронізації лічильників переданих даних при встановленні з’єднання;
  - FIN (No more data from sender) – признак передачі останнього байта переданих даних.
- Найважливішу роль в керуванні потоком інформації через TCP-з’єднання грають два 32-розрядних поля:
  - Номер послідовності
  - Номер підтвердження
- Ці поля відіграють роль ідентифікаторів пакета, з’єднання і суб’єкта з’єднання, а також відіграють роль лічильника пакетів
- Також важливу роль має поле Розмір вікна, яке інформує про розмір вільного буфера на боці приймача

Процедура встановлення TCP-з'єднання (1/2)

- Для встановлення TCP-з'єднання передбачена спеціальна процедура *рукостискання* (*Handshake*) у три кроки
  - Припустімо, що хост A намагається утворити TCP-з'єднання з хостом B
  - Першим кроком є відправлення хосту B на обраний TCP-порт пакета, в якому з прапорців встановлений лише SYN (так званий SYN-пакет)
    - SYN-пакет завжди є найпершим пакетом у будь-якому TCP-з'єднанні
    - При цьому в полі Номер послідовності установлено початкове 32-розрядне значення ISSa, або ISN (*Initial Sequence Number*)
  - Другий крок — відповідь від B до A. У цій відповіді повинні бути
    - встановлені прапорці SYN і ACK
    - у полі Номер послідовності повинно міститися початкове значення ISSb, обране хостом B
    - у полі Номер підтвердження повинно міститися значення, яке підтверджує одержання хостом B першого пакета. Значення підтвердження — це значення, яке хост B очікує в наступному пакеті
  -

Процедура встановлення TCP-з'єднання (2/2)

- На третьому кроці хост A завершує процедуру рукостискання, відсилаючи хосту B пакет, в якому
  - встановлено прапорець ACK
  - поле Номер послідовності містить ISSa+1
  - поле Номер підтвердження містить ISSb+1
- Коротко ці три стадії записують таким чином:
  - A → B: SYN, ISSa
    B → A: SYN, ACK, ISSb, ACK(ISSa+1)
    A → B: ACK, ISSa+1, ACK(ISSb+1)

- Після встановлення з'єднання лічильники Номер послідовності і Номер підтвердження відраховують кількість переданих/прийнятих байтів інформації

Захисні функції TCP (1/2)

- Припустимо, що ми намагаємось встановити з'єднання від імені іншого хосту
- Для цього підставимо як адресу відправника чужу IP-адресу (атака IP spoofing)
  - Припустимо, наш пакет успішно потрапив до хосту-одержувача
- У відповідь хост надсилає SYN-ACK-пакет на ту адресу, яка була вказана як адреса відправника
- Для того, щоби завершити встановлення з'єднання, ми повинні надіслати пакет, в якому міститься правильне значення ISSb+1
  - Для цього ми або повинні мати змогу перехопити пакет SYN-ACK, або повинні вгадати це значення.

Захисні функції TCP (1/2)

- Перехопити пакет можна лише в тому разі, якщо ми
  - або знаходимось на маршруті проходження пакета,
  - або знаходимось в тому ж сегменті мережі, що й хост, від імені якого ми виступаємо, причому в цьому сегменті всі пакети надходять всім хостам (логічна топологія *спільна шина*),
  - або ми вже здійснили відповідну атаку на маршрутизатор, комутатор чи точку доступу бездротового зв'язку
- Вгадати випадкове 32-розрядне значення ISSb за прийнятний час теоретично неможливо
  - якщо алгоритм генерування ISSb не забезпечує випадковість цього числа, у порушника є можливість таку атаку здійснити

- Додатковий захист забезпечується тим, що за стандартом при отриманні неочікуваного пакета хост повинен відправити пакет з прапором RST, і цей пакет розриває з'єднання
  - Зокрема, такий пакет відправляє хост, який не надсилав SYN-пакет і несподівано отримав SYN-ACK-пакет
- Таким чином, прийнята 3-стадійна процедура рукостискання здатна суттєво обмежити можливості порушників у встановленні з'єднань від чужого імені

Атака SYN-Flood

- Кожне окреме з'єднання вимагає певних ресурсів системи
  - На кожний отриманий TCP-запит на створення з'єднання операційна система хосту повинна згенерувати початкове значення ISN та відіслати його у відповідь на хост, що ініціює з'єднання
  - Для того, щоби в подальшому підтримувати TCP-з'єднання, ОС хосту має виділити буфер у пам'яті для тимчасового зберігання отриманих пакетів і підтверджень на надіслані пакети
- Це означає, що існує принципова можливість ці ресурси вичерпати, змусивши систему припинити обслуговування нових з'єднань.
- Якщо від хосту порушника надходить запит на встановлення з'єднання, то атакований хост виділяє на обробку цього з'єднання певні ресурси, відправляє SYN-ACK-пакет і чекає на пакет, що завершує процедуру рукостискання
  - Такий стан називається *напіввідкрите* з'єднання
  - При цьому протягом виділеного тайм-ауту ресурси атакованого хосту продовжують бути зайнятими, в той час як хост порушника жодних ресурсів не резервує
- Для успішної DoS-атаки порушнику достатньо ініціювати значну кількість напіввідкритих з'єднань
- Варіант цієї DoS-атаки спрямований на певний TCP-порт, тобто, на певну мережну службу
  - Він полягає в тому, що на атакований хост передають кілька десятків (можливо, сотень) запитів на підключення до однієї служби (наприклад, до FTP-сервера)
  - Деякі мережні ОС влаштовані так, що виділяють ресурси кожній службі окремо, і не перерозподіляють їх. На практиці це означає, що черга запитів до кожної мережної служби досить чітко обмежена

Захист від атаки SYN-Flood

- За термінологією CERT (Computer Emergency Response Team) ця атака має назву TCP SYN Flooding and IP Spoofing Attack — затоплення TCP-запитами з несправжніх IP-адрес.
- Можливість цієї атаки є прямим наслідком недоліків мережних протоколів IPv4 та TCP. Сервери принципово не можуть бути абсолютно захищені від такої атаки.
- Заходами захисту від цієї атаки є:
  - підвищення продуктивності сервера;
  - обмеження пропускної здатності каналу зв'язку;
  - збільшення обсягів пам'яті сервера, подовження черг запитів до кожного окремого порту;
  - максимальне скорочення тайм-аутів.
- Кожна з цих рекомендацій має свої обмеження:
  - або суто економічні (підвищення продуктивності, збільшення обсягів пам'яті),
  - або з міркувань доступності сервера для звичайних користувачів (обмеження пропускної здатності каналу, скорочення тайм-аутів).
- У минулому більш стійкими до такого типу атак показували себе суто клієнтські системи (наприклад, Windows 9x)
  - Відсутність серверів дозволяла цим системам миттєво поновлювати працездатність після припинення атаки
  - В наш час таких суто клієнтських систем практично не існує
  - Однією з важливих задач адміністрування є відключення всіх служб, які не повинні бути задіяні на конкретній машині

Можливість підміни суб’єкта з’єднання

- Пакет буде прийнятий як коректний черговий пакет в певному TCP-з’єднанні, якщо в ньому будуть вірні значення
  - IP-адрес джерела і призначення
  - TCP-портів джерела і призначення
  - Sequence Number та Acknowledgment Number (ISSa та ISSb)
- Порушнику треба знати або вгадати усі зазначені параметри
  - Будемо вважати, що IP-адреси йому відомі
  - Напевно, порушнику відомий номер порту сервера, який тісно пов’язаний із службою, до якої він звертається
  - Номер порту клієнта і два 32-розрядних номера послідовностей треба або знати, або вгадати
- Якщо це вдасться, порушник може надіслати пакет з будь-якого хосту в мережі Інтернету від імені одного з учасників з’єднання (наприклад, від імені клієнта), і цей пакет буде прийнятий як правильний
- Якщо можливе прослуховування трафіка, то протокол TCP не дозволяє захистити з’єднання

Можливість підміни суб'єкта з'єднання без прослуховування (1/2)

- Існує принципова можливість атаки і в тому випадку, коли порушник не має можливості прослуховування
  - Для підміни суб'єкта вже встановленого з'єднання без аналізу трафіка, навіть за умови знання номеру порту клієнта, порушник має вгадати два 32-розрядні числа
  - Для цього необхідні 2^{64} спроби, що за прийнятний час є абсолютно неможливим
- Інша ситуація виникає тоді, коли порушник намагається встановити з'єднання від імені іншого хосту
  - Така атака може бути корисною у випадках, коли порушник намагається здійснити проникнення через брандмауер, або у випадках, коли між хостами існують довірчі відносини
  - Порушнику необхідно вгадати лише одне 32-розрядне значення — початковий номер послідовності ISN, згенерований хостом, до якого здійснюється підключення
  - Якби це значення було випадковим, то вгадати його за прийнятний час було б неможливо. Але насправді і розробники протоколу TCP (автори RFC 793), і розробники мережних функцій ядер різних операційних систем ставились до процедури генерування початкових значень номерів послідовностей без належної уваги
    - В RFC 793 рекомендується збільшувати значення цього 32-розрядного лічильника на 1 кожні 4 мікросекунди
    - В ранніх Berkeley-сумісних ядрах ОС UNIX значення цього лічильника збільшувалося на 128 кожної секунди і на 64 для кожного нового з'єднання
    - В старих ядрах ОС Linux значення ISN обчислювалось в залежності від часу за формулою:

      ISN = μsec + 1000000×sec,

      де μsec — час у мікросекундах; sec — поточний час в секундах (відлік ведеться від 1970-го р.)
    - В ОС Windows NT 4.0 значення ISN збільшувалось на 10 приблизно кожну мілісекунду, тобто

      ISN = 10×msec,

      де msec — поточний час у мілісекундах

Можливість підміни суб’єкта з’єднання без прослуховування (2/2)

- Отже, якщо в ОС використовується часозалежний алгоритм генерування початкового значення ідентифікатора TCP-з’єднання, то порушник має можливість заздалегідь дослідити цей алгоритм і в процесі спроби встановлення з’єднання приблизно визначити значення ISN
- На практиці точне значення ISN визначити не вдається (наприклад, поточне значення лічильника мікросекунд є практично випадковим числом в діапазоні $0...10^6-1$)
- Але діапазон, в якому знаходиться необхідне значення, є достатньо вузьким для того, щоби успішно здійснити підбирання ISN шляхом перебору
  - Порушник спочатку намагається встановити з’єднання з будь-якою дозволеною службою атакованого комп’ютера і, аналізуючи заголовок пакета-відповіді, визначає приблизний час доставки пакета, поточний час на віддаленому комп’ютері і поточне значення ISN
  - Далі, використовуючи заздалегідь визначену ним функцію залежності ISN від часу, порушник здатен з певною точністю математично передбачити значення ISN, що буде згенероване у відповідь на його спробу підключення від імені іншого комп’ютера
- Така атака отримала назву передбачення номера послідовності TCP (TCP Sequence Number Prediction)

Схема атаки Мітника