МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

Національний авіаційний університет

Л.В. БУЛАНА, Л.П. ГАЛАТА,

Б.Я. КОРНІЄНКО, В.Г. ПАВЛОВ

БЕЗПЕКА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

Лабораторний практикум

Київ 2012

УДК 004 056.5 (076.5)

ББК 3973.202-082.03

Б

Рецензенти:

Єлізаров А. Б. – к. т. н., доцент кафедри комп'ютерізо-ваних систем захисту інформації Національного авіаційного університету

Душеба В. В. – к. т. н., доцент, старший науковий співробітник Інституту проблем моделювання в енергетиці ім. Г. Є. Пухова НАН України

Протасов А. Г. – к. т. н., доцент, зав. кафедри приладів та систем неруйнівного контролю Національного технічного університету «Київський політехнічний інститут»

Затверджено методично-редакційною радою Національного авіаційного університету (протокол № _/__ від __.__.2011р.)

Булана Л. В.

Б Безпека інформаційно-комунікаційних систем та мереж: лабораторний практикум Л. В.Булана, Л. П.Галата, Б.Я. Кор-нієнко, В.Г. Павлов. – К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2012. – 84 с.

Подано лабораторні роботи з дисципліни «Безпека інформаційно-комунікаційних систем та мереж» короткі теоретичні відомості та порядок їх виконання, а також контрольні запитання, а для деяких робіт - домашні завдання.

Для студентів напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем».

ЗМІСТ

ВСТУП

Теоретичні знання, які здобувають студенти під час опанування матеріалу лекцій з двохсеместрового курсу «Безпека інформаційно-комунікаційних систем та мереж», повинні бути закріплені розв'язанням практичних задач, наведених у лабораторних роботах, що відповідно до навчальної робочої програми мають виконуватися водночас з вивченням кожної теми дисципліни.

Практикум містить лабораторні роботи, які виконуються протягом сьомого та восьмого навчальних семестрів та відповідають змісту першого, другого, четвертого та п'ятого навчальних модулів за навчальною робочою програмою даної дисципліни. Кожна лабораторна робота виконується протягом чотирьох академічних годин та передбачає обов'язкову самостійну підготовку студентів, яка відповідно до навчальної робочої програми повинна займати не менше двох академічних годин.

Самостійна підготовка передбачає: повторення теоретичного матеріалу за темою лабораторної роботи, пошук відповідей на подані контрольні запитання, ознайомлення з порядком виконання лабораторної

роботи, опрацювання отриманих результатів, а в дея-ких лабораторних роботах також виконання домашнього завдання.

Для виконання лабораторних робіт необхідний комп'ютерний клас, який має задовольняти певні вимоги щодо комп'ютерів та їх програмного забезпечення:

1. *ПЕОМ класу IBM PC з такою мінімальною конфігурацією:*

– процесор з робочою тактовою частотою не менше 800 МГц;

– обсяг оперативної пам'яті не менше 256 МБ;

– жорсткий магнітний диск (вінчестер) об'ємом від 10 ГБ;

– пристрої для завантаження зі зовнішнього носія (дискети, CD або FLASH-пам'ять).

2. На комп'ютері має бути встановлено операційну систему (ОС) не нижче *WINDOWS XP* або *WINDOWS 2000* .
Використо-вується також додаткове програмне забезпечення, яке для кожної лабораторної роботи надається окремо.

3. В операційному середовищі системним адміністратором для студентів повинен бути сформований робочий профіль з правами, не меншими ніж «Досвідчений користувач».

Модуль І. ЗАХИСТ КОМП'ЮТЕРНИХ СИСТЕМ

Лабораторна робота 1.1

ДОСЛІДЖЕННЯ СИСТЕМИ ЗАХИСТУ КОМП'ЮТЕРА ЗА ДОПОМОГОЮ BIOS

Мета: розглянути складові частини BIOS, за допомогою яких створюється парольний захист комп'ютера; визначити методи погроз та атак на паролі BIOS.

Завдання:

– розглянути BIOS SETUP як складову підсистеми BIOS;

– прочитати комірки пам'яті CMOS;

– перепрограмувати комірки пам'яті CMOS;

– перевірити ефективність захисту паролів, що зберігаються в CMOS.

Додаткові прикладні програми:

– «*Checkbios.exe*» – програма для зняття у файл дампа з того блоку пам'яті, де зберігається BIOS;

– «*BIOS.exe*» – програма для запису змісту CMOS-пам'яті у файл.

– «*Редактор CMOS 1.0*» – програма для вивчення та редагування інформації, що міститься у CMOS;

– «*CmosPwd 4.3*» – програма для зняття паролю з CMOS;

– «*KiLSMOS*» – програма пошуку та зняття пароля CMOS;

– «*Bdb*» – програма підбору паролів для AMI-BIOS;

– «*BIOS320*» – програма скидання контрольної суми CMOS;

– *«HD Locker», «Cop's CopyLock II»* і *«AMI-BIOS»* – різни сервісні програми, які сприяють вивченню BIOS.

Усі зазначені програми не потребують попередньої інсталяції.

Короткі теоретичні відомості

Складові частини BIOS. BIOS (Basic Input Output System) - базова система вводу-виводу називається так тому, що містить великий набір програм вводу-виводу, які забезпечують взаємодію операційної системи з іншими апаратними складовими комп'ютерної системи: клавіатурою, «вінчестером», пристроями вводу з гнучких магнітних дисків, CD-дисків та FLASH-пам'яті, портами вводу-виводу та ін.

Програма BIOS міститься у спеціальної мікросхемі пам'яті, встановленій на материнській платі комп'ютера. Ця пам'ять відноситься до типу ROM (Read Only Memory), тобто призначена тільки для читання, тому вона зберігає свій вміст при виключенні живлення комп'ютеру на відміну від основної пам'яті.

Для зберігання змінних для програми BIOS використо-вується CMOS-пам'ять, яка відноситься до типу RAM, тобто призначена як для читання, так і для запису. Назва CMOS означає, що ця пам'ять виконана на основі КМОП-структур (CMOS-Complementary Metal Oxide Semiconductor), які характеризуються, як відомо, низьким енергоспоживанням. Тому для живлення CMOS-пам'яті вистачає невеликої батарейки або акумулятора.

Цілісність вмісту CMOS-пам'яті перевіряється за допомогою контрольної суми. Якщо вона не збігається, то вміст CMOS-пам'яті замінюється на стандартні значення – BIOS Setup Default Values, що зберігаються в мікросхемі ROM BIOS. Причиною цього може бути розряд батареї чи акумулятору живлення CMOS, або випадкова чи навмисна зміна вмісту окремих комірок CMOS.

Методи атак на пароль BIOS. Існує багато способів злому паролю BIOS. Адміністратор повинен знати їх, щоб змогти перешкодити використанню цих методів зловмисниками для доступу до комп'ютерної системи.

Якщо є доступ до материнської плати, то можна просто знеструмити CMOS, вийнявши батарейку, або за допомогою спеціального перемикача (джампера) на материнської платі. Більш цікаві методи, що дозволяють скинути пароль BIOS, виконуються виключно програмними засобами.

Насамперед, в деяких версіях BIOS записаний, так званий, «інструментальний» пароль, який (для обізнаних користувачів) відіграє роль такого собі «майстер-ключа» або відмички.

Також ефективним є метод, що ґрунтується на порушенні відповідності контрольної суми CMOS його вмісту. При завантаженні комп'ютера ця відповідність обов'язково перевіряється, і, якщо вона порушена, то виникає помилка «CMOS Checksum Error», і відбувається примусове поновлення вмісту CMOS-пам’яті на деякі стандартні значення, в яких пароля немає. Отже, достатньо порушити цю контрольну суму, записавши в CMOS-пам’ять будь-яку іншу інформацію, щоб зняти пароль на вхід у Setup.

Під’єднання CMOS-пам’яті до процесора виконується через спеціальний контролер, до якого можна звертатися як до пристрою вводу-виводу через порти з шістнадцятковими адресами 70h та 71h. Порт 70h призначений для зв’язку з адресною шиною CMOS, через нього указується адреса потрібної комірки CMOS. Порт 71h дає змогу звертатися до вмісту комірок CMOS, читати або записувати в них інформацію.

Наприклад, у середовище мови ASSEMBLER для читання комірки необхідно виконати такі команди:

out 70h, <адреса потрібної комірки CMOS>

in 71h

Запис у комірку CMOS нового вмісту виконується так:

out 70h, <адреса потрібної комірки CMOS>

out 71h, <нова інформація для зберігання>

У середовищі ОС MS-DOS, потрібні команди можна викона-ти за допомогою DEBUG – відладника, який входить до складу утиліт операційних систем, розроблених фірмою MICROSOFT.

Порядок виконання роботи

1. Перезавантажити комп'ютерну систему, відслідкувати появу повідомлення про доступ до BIOS та увійти до програми за допомогою зазначених клавіш BIOS Setup.

2. Послідовно проглянути зміст пунктів меню BIOS Setup, насамперед звертаючи увагу на такі, де:

- установлюється послідовність застосування пристроїв, з яких можна завантажити комп'ютер;
- установлюються паролі для захисту комп'ютера від його використання сторонніми особами;
- установлюється спосіб застосування цих паролів.

3. Установити будь-який пароль та задати такий спосіб його застосування, щоб обмежити доступ тільки до BIOS Setup.

4. Вийти з BIOS Setup та завантажити систему ОС WINDOWS.

5. Завантажити програму DEBUG та виконати в її середовищі команди читання комірок CMOS-пам'яті шляхом використання команд:

o 70h, <адреса потрібної комірки CMOS>

i 71h

Запис у комірку CMOS нового вмісту виконується так:

o 70h, <адреса потрібної комірки CMOS>

o 71h, <нова інформація для зберігання>

6. Змінити вміст декількох комірок CMOS-пам’яті за вказівкою викладача та перевірити здійснені зміни.

7. Перевірити, чи буде виявлене порушення контрольної суми CMOS при перезавантаженні комп’ютера.

8. Створити завантажувальну дискету або FLASH-пам’ять (див. дод. 1 – 2) та скопіювати на неї потрібні програми для атаки на пароль до BIOS Setup.

9. Перевірити ефективність цих програм шляхом застосування їх по черзі з подальшим перезавантаженням комп’ютера після запуску кожної програми. Спробу атаки на пароль CMOS слід вважати вдалою, якщо під час перезавантаження пароль буде скинутий. (Увага! Програмою «BIOS320» моделювати тільки атаки 1
–
4.
)

1. Занести до протоколу результати перевірки дії усіх запропонованих програм.

Контрольні запитання:

1. Яку роль виконує підсистема BIOS у завантаженні комп’ютера?

2. З яких міркувань для зберігання BIOS використовується пам’ять ROM?

3. Яку роль і чому відіграє підсистема BIOS у захисті комп’ютера?

4. Яке призначення CMOS-пам’яті?

5. В чому сенс використання технології CMOS?

6. Яка мета та які засоби існують для злому пароля BIOS?

7. Як прочитати вміст будь-якої комірки CMOS-пам’яті?

8. Яким чином змінити вміст певної комірки CMOS-пам’яті?

9. Чому спроба зміни контрольної суми CMOS-пам’яті із середовища ОС WINDOWS може бути невдалою?

Лабораторна робота 1.2

ДОСЛІДЖЕННЯ СТРУКТУРИ ТА СКЛАДОВИХ BIOS

Мета: розглянути складові частини підсистеми BIOS, засоби її перепрограмування і визначити види загроз та атак на інструментальні паролі BIOS.

Завдання:

- отримати двійковий файл вмісту BIOS;

- провести дослідження структури цього файлу;

- знайти інструментальний пароль BIOS у файлі і перепрограмувати його на іншій;

- перевірити отриманий результат шляхом шифрування паролю вручну за відповідним алгоритмом.

Додаткові прикладні програми:

- «*Awdflash.exe*» – програма для перепрограмування BIOS.

- «*Modbin.exe*» – програма для розпакування головного модуля BIOS із файлу, створеного з допомогою *Awdflash.exe*, та редагування його вмісту з отриманням нового двійкового файлу для перепрограмування BIOS. Для різних BIOS використовуються різні версії програми: *Modbin60.exe, Modbin66.exe, Modbin80.exe, Modbin6.exe* та *Modbin61.exe* .

- «*Cbrom.exe*» - програма, що надає можливість отримати інформацію про вміст та модулі запакованого файлу BIOS.

- «*Hiew.exe*» – програма для перегляду вмісту файлу з можливістю пошуку за контекстом або адресою.

- Завантажувальний носій операційної системи MS-DOS або Windows 9x, створений згідно з дод. 1 та 2.

Усі програми не потребують попередньої інсталяції, але програма Awdflash.exe коректно працює лише в середовищі операційних систем MS-DOS або Windows 9x.

Короткі теоретичні відомості

Розглянемо дві основні частини BIOS:

1. BootBlock. У його функції входять дії по ініціалізації регістрів процесора та розпакування основної частини BIOS для подальшого виконання, але сам BootBlock не архівований.

2. Основна частина. Складається з декількох модулів, що зберігаються як архіви типу LHA. Як правило, ці модулі мають такі назви:

- original.tmp – головна частина, завжди має розмір 128k. Саме в ній відбувається вся ініціалізація комп'ютера, у ній же розміщується підпрограма BIOS Setup;

- awardext.rom – «розширення» головної частини;

- awardepa.bin – «малюнок» *Energy Star*;

- cpucode.bin – таблиця мікрокодів для Intel-процесорів;

- acpitbl.bin – підпрограма підтримки ACPI,

а також VGA.rom (при інтегрованому відео), logo.bin і ін. Назви можуть дещо відрізнятися, але вміст приблизно однаковий.

Порядок виконання роботи

1. За допомогою програми AWDFLASH.EXE одержати BIOS, запакований у файл. Для цього необхідно:

а) завантажитися із системного носія MS-DOS чи WINDOWS;

б) запустити програму Awdflash.exe, але не вказувати ім'я файлу, в якому міститься новий вміст BIOS: на повідомлення «*File Name to Program»* просто натиснути клавішу «*Enter»*;

в) підтвердити необхідність збереження колишнього вмісту BIOS у файл: на повідомлення «*Do You Want*

To Save Bios (Y/N)»
необхідно натиснути клавішу «
Y»
;

г) указати ім’я файлу, у який запишеться вміст BIOS: після повідомлення «*File Name to Save»* увести ім’я файлу з розширенням «bin» і натиснути клавішу «
Enter»
.

2. Оцінити об'єм отриманого файлу (1 Mbit, 2 Mbit і т.д.).

3. Дослідити структуру файлу BIOS за допомогою програми HIEW.EXE. Для цього:

а) вибрати для перегляду файл, що містить запакований BIOS;

б) за допомогою клавіші «*F4»* (*Mode*) увімкнути режим «*Hex»*;

в) провести пошук (клавіша «*F7»* – *Search*), при якому вказати як контекст для пошуку символи «lh5»;

г) якщо це поєднання символів буде знайдено в тексті, курсор зупиниться на першому зі знайдених символів (символі «l»), який є четвертим від початку відповідного модуля BIOS;

д) перемістити курсор на три символи ліворуч від місця зупинки при пошуку і визначити адресу даного байта;

е) прочитати назву модуля, яка знаходиться на його початку;

є) занести адресу і назву модуля до табл. 2.1;

ж) помістити курсор на назву модуля і натисканням поєднання клавіш «*Ctrl+F7»* продовжити пошук з тим же контекстом, повторивши пункти г) – є);

з) продовжувати пошук, доки не буде виведене повідомлення « *Target not found»*, яке означає, що більше подібних поєднань символів у тексті немає;

и) продовжити перегляд вмісту файлу з BIOS за допомогою клавіші « *PgDn»,* або за допомогою контекстного пошуку і визначити адреси початку блоків з назвами « *Award Decompression Bios»* і « *Award BootBlock Bios»* , також занести дані про адреси даних блоків у табл. 2.1;

і) повторити пункти а) – и) для всіх наданих примірників файлів BIOS та заповнити табл. 2.1:

Таблиця 2.1

Версія BIOS 1,
дата його розробки
Версія BIOS 2,
дата його розробки
. . .
Версія BIOS N,
дата його розробки
адреса — назва модуля

адреса — назва модуля

. . .

адреса — назва модуля

. . .

. . .

. . .

. . .

адреса — назва модуля

адреса — назва модуля

. . .

адреса — назва модуля

4. Вибрати версію програми MODBIN.EXE для розпакування BIOS відповідно до ємності мікросхеми BIOS, визначеної в п. 2:

– для 1Mbit (AWARD 4.5xPG) – MODBIN60.EXE, MODBIN66.EXE чи MODBIN80.EXE;

– для 2Mbit (AWARD 6.xPG) – MODBIN6.EXE чи MODBIN61.EXE;

5. Виконати розпакування основного модуля «*original*» з файлу з BIOS за допомогою відповідної версії програми MODBIN.EXE:

а) завантажити відповідну версію програму MODBIN.EXE;

б) у ній вибрати для завантаження потрібний файл із розширенням «*.bin»;

в) вийти з програми за допомогою клавіші «*ESC*», після чого файл з назвою «*original.tmp*» або «*original.bin*» буде знахо-дитися в тому же каталозі, що й файли з запакованим BIOS;

г) змінити назву цього файлу, наприклад, на «*orig-old.bin*»;

д) за допомогою програми HIEW.EXE переглянути його вміст в області розміщення інструментального пароля BIOS, починаючи з адреси 1EC60, куди можна переміститися за допомогою клавіші «*PgDn*» чи скориставшись клавішею «*F5*» – (Goto) з указівкою потрібної для переміщення адреси;

е) записати вміст байтів інструментального пароля до протоколу;

є) знову повторити дії, як у пункті 5б;

ж) змінити інструментальний пароль шляхом наступних дій:

для AWARD 6xPG

– переміщатися по пунктах меню в такий спосіб:

– *«Change BIOS Option»* <*«ENTER»*> -> «*BIOS Option»* <*«ENTER»*> -> «*Security Default Password»* <*«EN TER»*
>
->
Ввести новий
інструментальний пароль
<
«ENTER»
>;

– за допомогою «ESC» повернутися в головне меню;

– вибрати натисканням клавіші «*ENTER»* пункт меню «*File»*, а потім вибрати «*Save BIOS»* і, нарешті, вказати ім'я файлу, що містить скоректовану програму BIOS;

для AWARD 4.5xPG

– вибрати пункт меню *«Change BIOS Option»* <*«ENTER»*> -> «*Security Default Password»* <*«ENTER»*> -> *B*
вести новий
інструментальний пароль
<
«ENTER»
>;

– ще одним натисканням клавіші «*ENTER»* повернутися в головне меню програми;

– за допомогою пункту «*Update File»* записати оновлений BIOS у файл замість колишнього вмісту.

з) повторно розпакувати отримані файли BIOS (див. п. 5а - 5в) і переглядаючи файли *original.** за допомогою програми HIEW.EXE переконатися в тому, що вони містять змінені інструментальні паролі BIOS, для чого розшифрувати парольний хеш за відповідним алгоритмом.

6. Відкрити командне вікно або файл-менеджер «FAR.EXE».

7. Запустити програму CBROM.EXE з опцією «/?» та проглянути інформацію про формат запуску програми та інші опції.

8. Знову запустити програму CBROM.EXE з опцією «/D»:

CBROM.EXE <Ім’я файлу з програмою BIOS> /D

9. У результаті одержати дані про компонування модулів у файлі BIOS з указівкою об'єму кожного модуля до і після упакування.

10. Порівняти отримані результати з тими, що містяться в табл. 2.1, яку побудували раніше.

Домашнє завдання

1. Виконати розпакування вмісту BIOS свого домашнього комп’ютера у файл.

2. Зашифрувати інструментальний пароль BIOS, задавши його у вигляді перших 8 символів свого прізвища.

3. Одержати запакований двійковий файл BIOS без перепрограмування BIOS.

4. Усі отримані файли додати до звіту про виконання роботи.

Контрольні запитання

1. Які назви мають основні модулі програми BIOS?

2. У чому різниця розміщення модуля *original* у версіях BIOS AWARD 4.5xPG і AWARD 6.xPG?

3. Які модулі BIOS розміщаються в ньому в незаархівованому вигляді і чому?

4. Поясніть, чому важко коректно переглянути вміст BIOS із середовища WINDOWS?

5. Які режими перегляду існують в програмі HIEW.EXE?

6. Чому при пошуку початку модуля необхідно шукати послідовність символів «lh5»?

7. Як здійснити пошук за відомою адресою в програмі HIEW.EXE?

8. Які ще параметри BIOS можна відредагувати програмою MODBIN.EXE?

9. Які функції можна виконати за допомогою програми CBROM?

    1. Яку роль виконує контрольна сума BIOS?

Лабораторна робота 1.3
ДОСЛІДЖЕННЯ АТАК ЗА ДОПОМОГОЮ ШТУЧНО ЗАНЕСЕНИХ ПРОГРАМ КЛАСУ SPYWARE

Мета: розглянути деякі види SpyWare-програм та програмні засоби боротьби з ними.

Завдання:

– на прикладі програми-exploit визначити механізм дії шкідливих програм класу SpyWare;

– перевірити ефективність програмних засобів, що знешкоджують комп'ютерні віруси та інші шкідливі програми.

Додаткові прикладні програми:

– програма «*HijackThis.exe»* для пошуку та вилучення програмних закладок. *Не потребує інсталяції;*

– «*Scan Spyware, Spyware Doctor, SpyWare Remover, XoftSpy, SYSCLEAN.EXE»* – програмні засоби для сканування комп'ютера з метою пошуку SpyWare-програм (
усі, крім останньої, *потребують інсталяції*
);

– програма «*Seccheck.exe*» - програма для збору різноманітної інформації про комп'ютерну систему. *Не потребує інсталяції;*

– «*ACTUAL SPY»* – програмний продукт класу SpyWare (*потребує інсталяції*).

– комплект програм для демонстрації дії SpyWare-програми:

- outlooki.exe – SpyWare-програма;
- OutpostPro – програма-firewall, яка фіксує дію SpyWare;
- Iris – програма, яка дозволяє контролювати відправлення мережевих пакетів та їхній вміст.

(усі програми, крім outlooki.exe, *потребують інсталяції*).

Короткі теоретичні відомості

SpyWare визначають як програмне забезпечення, що дозволяє збирати відомості про окремо узятого користувача чи організацію без їх дозволів.

AdWare (від *advertising* – реклама) – програмний код, без відома користувача доданий у програмне забезпечення з метою демонстрації рекламних оголошень.

Відрізняють такі основні групи SpyWare-програм:

Internet-spyware – ці програми збирають інформацію про якість зв'язку, способи підєднання, швидкість модема і т.д;

HDD-spyware – займаються збором інформації про вміст твердого диска і надсилають її своєму розробнику;

Keylogger програми цього типу збирають інформацію про кожну натиснуту клавішу та інші дії користувача;

Логгери додатків – збирають інформації про додатки, з якими працював користувач;

WWW-spyware – програми, що стежать за активністю користувача в Internet: наприклад, інформацією, яка вводиться у формі при здійсненні покупки в електронному магазині;

Cookie-spyware невеликі файли, що використовуються для зберігання різної реєстраційної інформації і т.д;

E-mail spyware – подібні додатки можуть внести зміни в електронне листування користувача, додавши до відправленого чи отриманого листа графічний баннер або рекламний текст.

Методи боротьби з поширенням SpyWare-програм По-перше деяка частина SpyWare виявляється і віддаляється антівірусними програмами.

По-друге, дія SpyWare-програм виявляється за допомогою мережевого екрана – Firewall, наприклад, Conseal PC FireWall, ATGuard, Agnitium Outpost FireWall та ін. Він реєструє та блокує передачу інформації з боку SpyWare.

По-третє, слід встановити програми контролю за змінами в системних файлах і службових областях даних – так звані адвізори (англ. *adviser* – повідомник).

Порядок виконання роботи

1. Вивчить механізм дії програми outlooki.exe, для чого:

- установити програму-firewall OutpostPro;
- створити в адресній книзі поштової програми OUTLOOK-EXPRESS декількох користувачів;
- запустити програму-exploit outlooki.exe і відслідкувати його спробу відкрити TCP-порт для зв'язку за FTP-протоколом з деяким адресатом;
- переглянути вміст двох сформованих для відправлення файлів emdat.tm і emdat.tmp у кореневой папці WINDOWS і переконайтbcz, що в цих файлах знаходиться вміст адресної книги з OUTLOOK-EXPRESS.

2. Отримати за допомогою програми SECCHECK.EXE дані про комп'ютерну систему і встановлене програмне забезпечення. Для цього необхідно виконати зазначену програму у вікні командного процесора, а звіт буде сформований у текстовому файлі «SecCheckLog.txt».

3. Завантажити програму Actual Spy, що належить класу Spyware і дозволяє вести прихований контроль дій, виконуваних на комп'ютері. Для цього слід:

- очистити файл, що зберігає запис дій, за допомогою кнопки «Очистити всі логі»;
- увімкнути запис дій натисканням кнопки «Старт»;
- перевести програму в прихований режим роботи натисканням кнопки «Скрыть».

4. Вивчити основні режими запуску і функціонування програм для пошуку програмних закладок і вразливостей програмного забезпечення комп'ютера:

- Spyware Doctor;
- SpyWare Remover;
- SysClean;
- XoftSpy,

та виконати сканування зазначеного викладачем каталогу.

5. Натисканням поєднання клавіш «Ctrl+Alt+Shift+F8» викликати на екран вікно програми Actual Spy і визначити, які дії користувача комп'ютером були зафіксовані.

6. Зберегти звіт цієї програми в зазначену викладачем папку.

Домашнє завдання

1. На домашньому комп'ютері за допомогою програм:

- HijackThis.exe;
- Scan Spyware;

виконати сканування кореневої папки WINDOWS.

2. Результати сканування зберегти у вигляді текстових *log-файлів*, скопіювати на носій і надати викладачу.

Контрольні питання:

1. Які існують основні групи програмних закладок?

2. Чим відрізняються програми класу AdWare від SpyWare?

3. Які різновиди SpyWare-програм ви знаєте?

4. Як відбувається поширення програмних закладок?

5. Що є об’єктами атак SpyWare-програм?

6. Якої шкоди завдають програмні закладки?

7. Які заходи пропонуються для боротьби з SpyWare?

8. Які існують основні групи програм для боротьби з SpyWare?

9. Яким чином відбувається виявлення SpyWare?

Лабораторна робота 1.4

ДОСЛІДЖЕННЯ СТРУКТУРИ PE-ФАЙЛІВ

Мета: розглянути склад та основні сегменти PE-файлів а також прийоми та засоби їх редагування.

Завдання:

– навчитися аналізувати та використовувати інформацію про структуру PE-файлів;

– з’ясувати принцип, який використовують програми, що змінюють інтерфейс і мову повідомлень.

Додаткові прикладні програми:

– Програма «*PE Tools*», що дозволяє переглядати та редагувати заголовки PE-файлів. *Не потребує інсталяції;*

– «*PE Explorer*» – програма, за допомогою якої можна проаналізувати внутрішню структуру PE-файлів та її вихідний код. *Програма потребує інсталяції*;

– «*Resource Hacker», «Restorator*» – програми, що дають можливість не тільки продивитись, а й відредагувати інтерфейс та ресурси, що використовуються в програмі.
Обидві програми не потребують інсталяції.

Короткі теоретичні відомості

Уперше формат здійснених файлів, називаний Portable Executable (PE) (переносимий формат виконуваних файлів), був представлений в операційній системі WINDOWS NT версії 3.1. Трохи пізніше специфікація цього формату була опублікована і уведена до складу Microsoft Developer Network CD. Новий формат відповідає специфікації COFF (загальний формат об'єктних файлів – Common Object File Format), розповсюдженій у багатьох операційних системах сімейства UNIX. Одночасно для збереження сумісності з попередніми версіями MS-DOS і WINDOWS PE-формат також зберіг старий знайомий MZ-заголовок DOSa.

Структура PE файлу Формат PE-файлів починається з *заголовка MS-DOS*, *програми реального режиму* і *сигнатури PE файлу*
. Далі слідують
заголовок PE-файлу
й
опціональний заголовок
. Після них ідуть
заголовки всіх сегментів
, за яких випливають
тіла цих сегментів
. І ближче до кінця файлу розташовані різні
області даних
, включаючи інформацію про переадресації, таблицю символів, інформацію про номери рядків і дані в таблиці рядків.

Заголовок Реального режиму/MS-DOS. Заголовок MS-DOS потрібен для формування повідомлення на кшталт «This program cannot be run in DOS mode» ("Ця програма не може бути запущена в DOS"), якщо буде спроба запустити програму в середовищі MS-DOS, інакше операційна система могла б просто «зависнути» при спробі запуску такого файлу.

Заголовок MS-DOS займає перші 64 байти PE-файлу та містить 19 полів.

Сегменти PE-файлу. Сегменти, або секції, містять власне вміст файлу, включаючи код, дані, ресурси й іншу інформацію про PE-файл. Кожен сегмент має заголовок і тіло сегмента (безпосередньо дані). Заголовок сегмента (секції) має довжину 40 байт і не вирівнюється по межам поля, кратного двум байтам.

До складу PE-файлів входять дев'ять визначених сегментів: .text, .bss, .rdata, .data, .rsrc, .edata, .idata, .pdata, та .debug, хоча усі вони не є обов'язковими.

Порядок виконання роботи

1. За допомогою програми «PE Explorer» виконати дослідження, запропонованого викладачем, а саме:

- переглянути заголовок файлу і визначити значення таких його стандартних і додаткових полів:

– кількість секцій у файлі;

– адресу точки входу PE-файлу;

– відносний зсув сегмента коду (сегмента .text);

– відносний зсув сегмента даних (сегмента .bss);

– початкову адресу завантаження сегментів у пам'ять;

– обсяг графічної інформації, що міститься у файлі.

Усі отримані дані занести до протоколу лабораторної роботи;

- переглянути список сегментів (секцій), що входять до складу файлу і визначте обсяг кожного з них;
- переглянути список ресурсів, що входять до складу файлу;
- за допомогою пункту «Сканер залежності» меню «Інструменти» визначити перелік бібліотечних файлів, стандартні функції яких використовуються в даному PE-файлі;
- виконати дізасемблювання заданого PE-файлу за допомогою вбудованого дизассемблера.

2. Виконати дослідження аналогічних параметрів того ж файлу за допомогою програми «PE Tools», завантаживши його для перегляду за допомогою пункту «PE Editor» меню «*Tools*». Визначити, які параметри досліджуваного файлу, отримані за допомогою програми «PE Explorer», не можна знайти програмою «PE Tools». Занести до протоколу ті параметри заданого файлу, що аналогічні знайденим у п. 1.

3. Дослідити основні функції «Resource Hacker»:

- переглянути перелік ресурсів, що входять до складу заданого викладачем файлу;
- виконати переклад пунктів меню програми;
- виконати також редагування ресурсу, зазначеного викладачем.

4. Шляхом самостійного дослідження з використанням довідкової системи програми «Restorator» визначити її основні можливості з редагування ресурсів файлів. Виконати редагування зазначеного викладачем ресурсу в заданому файлі.

Контрольні запитання

1. Що визначає специфікація COFF?

2. Як побудовано формат PE-файлів?

3. Яке призначення Stub-програми?

4. Що називається програмою «реального режиму»?

5. Чому в програмах використовується відносна адресація?

6. Що таке точка входу програми?

7. Чому поля опціонального заголовка розділені на групи?

8. Що являє собою сегмент PE-файлу?

9. Назвіть визначені сегменти.

    1. Де розміщаються заголовки сегментів?
    2. Що визначає адреса завантаження сегменту?
    3. Що визначає поле «*Characteristics*» у заголовку сегмента?

Модуль II. ЗАХИСТ ОПЕРАЦІЙНИХ СИСТЕМ (ОС)

Лабораторна робота 2.1

ДОСЛІДЖЕННЯ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS 9X

Мета: розглянути принципи побудови захисту ОС WINDOWS 9x та виявити вразливості файлу парольного кеша.

Завдання:

– визначити складові профілю користувача *WINDOWS 9x*;

– знайти та проглянути вміст файлу парольного кешу;

– відшукати посилання на мережеві ресурсі в цьому файлі.

Додаткові прикладні програми:

– «*Pwltool6_9*» – програма для розшифрування вмісту файла парольного кешу. Ця програма *вимагає інсталяції в середовищі будь-якої*
ОС;

– Програми «*PWLInside.EXE», «IISPass.exe», «REPWL.EXE», «RA-cache.exe», «PWLOSR.EXE»* і «*xIntruder.exe»*
інсталювати не треба, але вони працюють винятково в середовищі ОС WINDOWS 9x.

– «*Hiew.exe*» – програма-viewer для перегляду вмісту файлу з можливістю пошуку за контекстом чи адресою.

Короткі теоретичні відомості

Під парольним кешем ми розуміємо файли для збереження і повторного використання паролів в ОС WINDOWS 9x. Ці файли розміщуються в кореневої папці WINDOWS і мають розширення *.PWL (*PassWord List*). Вони створюються при першій реєстрації користувача на даному комп'ютері і мають ім'я, що за першими восьмома символами збігається з логіном користувача. Після кожної наступної реєстрації на комп'ютері користувач одержує доступ до свого файлу парольного кешу, а відповідно – до усіх своїх паролів.

Для більшого захисту вся інформація у файлі парольного кешу зашифрована із застосуванням досить криптостійких алгоритмів RC4 і MD5. Алгоритм RC4 є алгоритмом потокового шифрування, MD5 – алгоритмом створення дайджестів (згорток) повідомлень, досить швидкий у реалізації.

Процедура шифрування/дешифрування парольного кеша зберігається у файлі динамічної бібліотеки MSPWL32.DLL. Крім того, керування локальними ресурсами машини, наданими в мережу, виконує драйвер VSERVER.vxd, а обмін інформацією з видаленими серверами - VREDIR.vxd. Також у зв'язці працюють бібліотеки MSNET32.dll, MSNP32.dll та інші.

Існують PWL-файли двох різних типів: старі" PWL файли - створені оригінальною ОС WINDOWS 95 (або WINDOWS 3.11) і "нові" - створені WINDOWS 95 OSR2 чи WINDOWS 98. Вони відрізняються сигнатурою, з якої починаються файли: 4D 46 4E ("MFN") для "старого" формату, E3 82 85 96 ("уВЕЦ") для "нового".

Обидва формати файлів до адреси 0208 мають однакову структуру:

0000: 4 байти - "магічне" число чи сигнатура. Дозволяє відразу ж розпізнати, у якому форматі створений даний PWL-файл.

0004: Подвійне слово, що містить лічильник користувача, указує номер PWL-файлу в комп'ютері.

0109: Таблиця Resource Key Entry з 255 байт. Тепер елемент, що не використовується, позначається вже байтом FF. Тут зберігається інформація в кількості парольних записів у якому-небудь ресурсі. Будь який байт у цій таблиці рівний N (крім "FF") означає, що в ресурсі N міститься парольний запис.

Далі розглянемо обидва форматі окремо:

У "старому" форматі:

0208: 20 байт - логін користувача у верхньому регістрі.

21C: зсув у файлі початку ресурсу 0, відносно початку файлу;

21E: зсув у файлі початку ресурсу 1, відносно початку файлу;

...

23A: зсув у файлі початку ресурсу F, відносно початку файлу;

23C: загальна довжині файлу у байтах;

023E: Ресурс 0

Ресурс 1

Ресурс 2

...

Ресурс F

Розглянемо структуру записів ресурсу. Перші два байта кож-ного запису містять довжину запису. Тому навіть порожній ресурс має довжину 2 байта. Якщо всі ресурси в PWL-файлі порожні, то такий файл має довжину 23Eh + 16*2 = 25Eh = 606 байт.

Інформація по адресах 208h-21Bh і 21Ch-23Dh складає єдине поле, що зашифроване гамою, яка накладається зі зсуву 208h.

У "новому" форматі:

0208: 4 байта - зсув CryptoSign.

020C: 64 байта - масив CryptoSeed[16],

024C: 4 байта - CheckSeed.

0250: йдуть два нульових байти.

0252: 16 байт - масив CryptoSign.

0262: 16 байт масив CheckSign - цей масив разом з CryptoSign є "контрольним значенням" для визначення правильності пароля.

0272: масив з 15 двохбайтних слів - це адреси 15 ресурсів, починаючи з другого. Адреса ж першого ресурсу завжди відома і складає 0x290.

0290: безпосередньо ресурси, що записані один за одним.

Якщо ж у файлі ресурси відсутні, то починаючи з адреси 0x290 розміщаються нульові слова для кожного з 16 ресурсів так само, як і в WINDOWS 95. Такий файл буде мати довжину, що розраховується за формулою 290h + 16*2 = 2B0h = 688 байт.

У PWL-файлі WINDOWS 95 для шифрування використовується метод гамування, де гама генерується за алгоритмом RC4, але вона може бути досить легко знайдена.

Справа в тому, що цим алгоритмом зашифровується й ім’я користувача, що розміщується в PWL-файлі в діапазоні адрес з 0208 по 021B. Ім’я ж, у свою чергу, практично завжди відомо або з назви самого PWL-файлу, або з секції [*Passwords Lists*] конфігурючого файлу SYSTEM.INI.

Далі ми використовуємо відому властивість оборотності логічної операції XOR, по якій накладається гама:

(X XOR Y) XOR Y = X

Це означає, що якщо ми маємо в розпорядженні фрагмент файлу з уже накладеною гамою, але знаємо логін користувача, то можемо визначити саму гаму. Правда ми знайдемо лише перші 20 байт гами, але це дає нам можливість розшифрувати по 20 байт кожного ресурсу, оскільки при їхньому шифруванні гама не міняється і залишається такою ж
. Також, при необхідності ми можемо розширити відому частину гами ще на 34 байта

Цей алгоритм і реалізований у відомій програмі Glide, яка розшифровує 54 перших байта гами.

У PWL-файлі WINDOWS 95OSR/OSR2/98 алгоритм розшифровки ресурсів, що містяться в ньому, більш складний, оскільки система захисту була удосконалена із врахуванням недоліків у WINDOWS 95. Тому тут використовується підхід, традиційний для всіх програм парольних зломщиків, а саме перебір можливих комбінацій символів з перевіркою кожної за допомогою 16-байтніх масивів CryptoSign (0x252) і CheckSign (0x262) до тих пір, доки не буде знайдено вірний пароль.

Саме такий підхід реалізований у програмах PWLTool, PWLInside.EXE, REPWL.EXE та інших, ефективність яких досліджується в даній лабораторній роботі.

Порядок виконання роботи

1. Виконати дослідження запропонованого викладачем зразка PWL-файлу "старого" формату в такий спосіб:

- Переписати до протоколу байти, де зашифроване ім’я (логін) користувача;
- Визначити гаму, що накладається на вміст PWL-файлу.
- Розшифрувати за допомогою отриманої гами вміст ресурсів, що записані починаючи зі зсуву 23Eh.
- За допомогою програми Hiew відредагувати зазначені викладачем ресурси в файлі, і зберегти відредагований PWL-файл в іншому каталозі з тим же ім’ям;

1. З використанням програми PWL Tools, дослідити той же PWL-файл "старого" формату та його відредаговану копію і визначити паролі до ресурсів, що зберігаються в ньому.
2. За допомогою програми PWL Tools, визначити пароль і перелік ресурсів, що знаходяться в PWL-файлах "нового" формату, виданих викладачем.

1. Подати отримані результати письмово і на носії викладачу.

Домашнє завдання:

1. Ознайомитися з усіма режимами роботи програми, PWLTool_v 6.9 включаючи настроювання режимів пошуку по словнику, Smart Force і Brute Force.
2. Розробити "Інструкцію користувача", у якій докладно (зі скриншотами) описати роботу програми PWLTool_v 6.9.

Контрольні питання:

1. Яка необхідність створення файлу парольного кеша?

1. Паролі доступу до яких ресурсів WINDOWS записуються у файл парольного кеша?
2. Пояснить суть методу гамування при шифруванні файлу.
3. Пояснить, з якої причини файл парольного кеша WINDOWS 95 має низьку стійкість до розшифровки?
4. Як розраховується довжина порожнього парольного кеша?
5. Пояснить використання словників при пошуку пароля.
6. Пояснить принцип пошуку пароля в режимі “Smart Force”.
7. Пояснить принцип пошуку пароля в режимі “Brute Force”.

Лабораторна робота 6

ДОСЛІДЖЕННЯ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS 2K

Мета: розглянути побудову захисту операційної системи WINDOWS 2k та виявити уразливості файлу облікових записів.

Основні Завдання:

– перевірити надійність зберігання файлу облікових записів;

– дослідити методи атак на файл облікових записів SAM;

– з'ясувати вимоги до надійних паролів користувачів.

Додаткове прикладне програмне забезпечення:

- "*NTFSDOS Professional 4.03*" – програма для створення двох дискет з файлами NTFSPRO.EXE та NTFSCHK.EXE – драйверами для доступу до тих розділів "вінчестера", де використовується файлова система NTFS. Ця програма *потребує інсталяції*, але можна просто скопіювати вміст обох дискет з оригіналу шляхом клонування або за допо-могою команди diskcopy. Якщо в

якості завантажувального носія використовується FLASH-пам’ять, вміст обох дискет треба просто скопіювати на нього в окрему теку;

- Програма L0phtCrack версії 4.0 (LC4+), що дозволяє аналізувати вміст облікових записів у файлі SAM.

- Програма LCP версії 5.04 (rus), яка виконує такі ж самі функції, але не потребує інсталяції та має російськомовний інтерфейс.

- Програма Saminside версії 2.3, яка теж дозволяє аналізувати вміст файлів SAM та не потребує інсталяції.

- Програма для створення завантажувального диску OS Linux з утилітою Offline NT Password & Registry Editor (автор Petter Nordahl-Hagen) та образ цього диску.

- "*Hiew.exe*" – програма - viewer, що дозволяє продивіться вміст окремого файлу з можливістю пошуку за контекстом або адресою.

- "Завантажувальний" носій операційної системи MS-DOS або WINDOWS 9x (виготовлюється згідно додатку 1 або 2).

Короткі теоретичні відомості

До операційних систем (ОС) групи WINDOWS 2k відносяться наступні: WINDOWS NT/2000/2003 і XP Professional.

Основу системи захисту WINDOWS 2k складає система збереження файлів NTFS (New Technology File System). Головна відмінність файлової системи NTFS від інших (FAT, VFAT, HTPS) полягає в тому, що тільки вона задовольняє стандарт безпеки С2, зокрема, NTFS забезпечує захист файлів і каталогів не тільки при віддаленому, але при локальному доступу до них.

Для кожного користувача створюється обліковий запис, у якій у зашифрованому вигляді зберігається вся інформація про його пароль доступу. Всі облікові записи зберігаються в спеціальному файлі SAM (*Security Account Management Database*
).

Структура файлу облікових записів SAM.

На відміну від свого попередника – файлу парольних кешей PWL для ОС WINDOWS 9x, файл SAM має більш складну структуру і більш досконалу систему захисту. Для кожного користувача формується обліковий запис наступної структури:

“ЛогінКористувача:RID:LM-хеш:NT-хеш:ПовнеІм`я,Опис:Каталог”

де:

- RID - (*relative identifier*.) - відносний ідентифікатор, відрізняєть-ся для кожного користувача. Ідентифікатори облікових записів Administrator (адміністратор) і Guest (гість) призначаються самою ОС і дорівнюють відповідно 500 і 501. Інші користувачі при їхньої реєстрації отримають RID від 1000 і вище;

- LM-хеш - (*Lan Manager*) 16-байтній хеш 14-символьного пароля, сумісного з WINDOWS 9x;
- NT-хеш - 16-байтній хеш 128-символьного пароля, використовуваного тільки в ОС WINDOWS 2k;

вміст інших полів – дані про користувача і його робочий каталог.

Алгоритм формування LM-хеша

1. Введений пароль приводиться до верхнього регістру.
2. Кожна з 7-символьних половин пароля шифрується окремо по алгоритму DES (колишній федеральний стандарт США).
3. Отриманий хеш знову шифрується по алгоритму DES, де як ключ використовується RID користувача. Це необхідно, щоб виключити одержання однакових хешей, якщо різні користувачі використовують той самий пароль.

Алгоритм формування NT-хеша

1. Введений пароль перекодується в UNICODE.
2. Для нього будується хеш-функция по алгоритму MD4.
3. Виконуються дії, аналогічні пункту 3 для LM-хеша.

Якщо порівняти обидва алгоритми хешування, то зрозуміло, що

NT-хеш більш захищений, тому LM-хеш завжди відключають крім випадків, коли потрібно забезпечити сумісність в мережі з комп'ютерами, де встановлена ОС WINDOWS 9x.

Додатковим засобом безпеки є шифрування парольних хешей за допомогою утиліти SYSKEY. При цьому унікальний 128-бітовий унікальний ключ для шифрування PEK (
Password Encryption Key
) автоматично зберігається у системному реєстрі.

Порядок виконання роботи

1. Знайти всі екземпляри файлу облікових записів SAM і SYSTEM у папках комп'ютера.

1. Перевірити неможливість перегляду вмісту і копіювання файлів у папці %WindowsRoot%system32config
2. За допомогою утиліти NTFSDOS і завантажувального носія скопіювати екземпляр файлу SAM із зазначеного викладачем комп'ютера.
3. Виконати дослідження запропонованих викладачем зразків файлів SAM у такий спосіб:

- виконати пошук пароля програмою LCP, задаючи по черзі символи кожної з чотирьох груп:
- заголовні і малі літери латинського алфавіту;
- заголовні і малі літери кирилиці;
- цифри;
- знаки;
- для кожної з груп визначити час повного перебору всіх комбінацій символів за умови, що пароль не перевищує 10 символів. Результати занести до протоколу.
- виконати пошук пароля у тому ж файлі утилітою SAMInside і зафіксувати отримані результати в протоколі.

1. Додати отримані результати в протоколі і на носії викладачу.

Домашнє завдання:

1. Створити завантажувальний носій ОС MS-DOS чи WINDOWS 9x.

1. Створити шляхом копіювання папок "NTFSDOS ProfessionalDisk_1" і "NTFSDOS ProfessionalDisk_2" дві дискети з утилітами NTFSDOS або просто додати їх на завантажувальний носій FLASH-пам'яті.

2. Провести дослідження вмісту файлу SAM на домашньому комп'ютері за допомогою програм LCP та SamInside

Контрольні питання:

1. Які нові переваги в безпеці надає файлова система NTFS?
2. Яка роль файлу SAM у системі захисту WINDOWS 2k?
3. Що таке RID? Який діапазон його значень?
4. Чому формується дві хеш-функції паролю в файлі SAM?
5. Поясніть, чому LM-хеш більш легко розшифровується?
6. Яким чином здійснюється захист утилітою SYSKEY?
7. Які види захисту ключем SYSTEM KEY можуть бути реалізовані?
8. Якими прийомами може бути скопійований файл SAM?
9. Назвіть режими роботи програми парольного зломщика.

10. Яким чином можна відключити LM-автентифікацію?

Лабораторна робота 7

АДМІНІСТРУВАННЯ БЕЗПЕКИ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS 2K

Мета: розглянути прийоми та способи адміністрування в операційній системі WINDOWS 2k за допомогою вбудованих та програмних засобів.

Завдання:

– розглянути системні журнали як засоби аудиту роботи;

– з'ясувати можливості віддаленого адміністрування;

– перевірити файли і процеси, які завантажені в WINDOWS.

Додаткове прикладне програмне забезпечення:

- "*AUTORUNS.EXE*" – програма для відображення усіх процесів, що завантажені на даному комп'ютері. Ця програма *не потребує інсталяції.*

- "*FileMon.exe*" – програма - монітор процесів у реальній годині. *Не потребує інсталяції*, але для завантаження в середовищі WINDOWS 2k треба використовувати файл (exeReleaseFileMon.exe).

- Програма "*Hands Off*" для адміністрування комп'ютера (*потребує інсталяції*).

- Програма "*Pwltool6_9*" - програма для розшифровки паролів до мережевих ресурсів, яка була *інстал ьована раніше*
в лабораторної роботі №2.

- Утиліти *USER2SID.EXE* та *SID2USER.EXE* з допомогою яких можна визначити перелік користувачів на віддаленому комп'ютері. *Не потребують інсталяції*.

- Програма *Ideal Administration*, яка надає можливість віддаленого адміністрування (*потребує інсталяції*).

- Утиліта " *DDoSPing.exe* " – програма тестування комп'ютерів на стійкість до DDoS-атак (*не потребує інсталяції*).

Короткі теоретичні відомості

Мистецтво адміністрування полягає в грамотному використанні всіх налаштувань операційної системи, а також у своєчасному поновленні системного програмного забезпечення.

Основною утилітою адміністрування є "Керування комп'ютером" ("Computer Management"), що містить три групи налаштувань:

1. "Службові програми" ("System Tools"), що, у свою чергу, складається із шести програм:

- "Перегляд подій" ("Event Viewer") – перегляд системних журналів;
- "Оповіщення і журнали продуктивності" ("Performance Logs and Alerts") – настроювання виду попереджень і повідомлень:
- "Загальні папки" ("Shared Folders") – перегляд і адміністрування загальних ресурсів комп'ютера;

- "Диспетчер пристроїв" ("Device Manager") – настроювання апаратної частини комп'ютера;
- "Локальні користувачі і групи" ("Local Users and Groups") – адміністрування прав користувачів і груп.

1. "Запам'ятовуючі пристрої" ("Storage") – керування властивостями запам'ятовуючих пристроїв за допомогою розділів: «Зйомні ЗП», «Дефрагментація диска» і «Керування дисками».
2. "Служби і додатки" ("Services and Applications") – інформація про стан різних служб, наприклад, Plug and Play, DNS, DHCP і т.п. .

Другий по важливості в системі безпеки ОС WINDOWS 2k є утиліта "Локальна політика безпеки" ("Local Security Policy"), призначена для адміністрування характеристик паролів, терміну їхньої дії, правил їхнього відновлення і т.д. .

Звичайно, настроювання рівня безпеки виключно індивідуально для кожної комп'ютерної системи і, у кінцевому рахунку, залежить від багатьох чинників. Однак, існують деякі рекомендовані стандарти, яки реалізовані в шаблонах безпеки.

Шаблони безпеки для адміністратора WINDOWS 2k

Шаблон безпеки являє собою звичайний текстовий файл із розширенням *.inf, що знаходиться в папці %WinRoot%=securitytemplates і містить настройки наступних груп параметрів операційної системи.

- Account Policies: політика паролів, правила блокування облікових записів і настроювання протоколу Kerberos. Якщо дані політики застосовуються на рівні організаційного підрозділу (Organization Unit), то вони торкають тільки локальнї бази облікових записів (SAM). Настроювання доменних облікових записів регулюються Default Domain Policy.
- Local Policies: система аудита, права користувачів і основних настроювань безпеки.
- Event Log Settings: настроювання трьох журналів аудита (System, Application, Security).
- Restricted Groups: обмеження членства в групах.
- System Services: режим запуску і контроль доступу до системних служб.
- Registry values: дозвіл на доступ до даних реєстру.
- File System: дозвіл на доступ до файлів і папок.

Стандартне постачання WINDOWS 2k містить кілька шаблонів, яки зберігаються в папці %WinRoot%inf, та використовуються для настроювання системи на різні рівні безпеки,

Порядок виконання роботи

1. Відкрийте групу "Адміністрування" і перегляньте основні утиліти. Визначте, які з них недоступні для перегляду користувачу, що не належить до групи адміністраторів. Запишіть перелік недоступних утиліт до протоколу.
2. Знайдіть усі стандартні шаблони безпеки, що зберігаються в даному комп'ютері. Зафіксуйте їх у протоколі і спробуйте ідентифікувати їх по ступеню безпеки.
3. За допомогою утиліти NET.EXE визначте мережеві імена всіх комп'ютерів у домені кафедри.

Занесіть отримані дані до протоколу.

4. Виконати дослідження локальної мережі за допомогою утиліти NBTSTAT програми PWLTOOLS. Занести до протоколу IP-адреси всіх комп'ютерів домену, а також дані для видаленого комп'ютера, зазначеного викладачем.
5. За допомогою утиліт User2sid і Sid2user визначити перелік логінів користувачів на тому ж видаленому комп'ютері, що й у пункті 4. Отримані дані: SID комп'ютера, логіни і RID усіх знайдених користувачів занести до протоколу.
6. Виконати тестування комп'ютера, зазначеного в пункті 4 за допомогою програми DDoSPing.exe. Результати пред'явити викладачу.

Домашнє завдання:

1. За допомогою утиліти AUTORUNS.EXE визначте перелік ключів системного реєстру, де вказуються програми, що запускаються автоматично при старті ОС WINDOWS. Занесіть даний перелік до протоколу.

1. Виконайте дослідження процесів на Вашому комп'ютері за допомогою утиліти FileMon.exe. Складіть коротке, приблизно на 1 стор. «Керівництво користувача» з описом усіх режимів і можливостей даної програми. Даний опис у текстовому й електронному виді додати до протоколу.

Примітка. Для запуску утиліти FileMon.exe необхідно на даному комп'ютері мати права адміністратора, а, крім того, шляхом запуску SYSINSTALL.BAT повинний бути скопійований відповідний драйвер.

Контрольні питання:

1. Як створюється обліковий запис для нового користувача?
2. Що дає користувачу приналежність до тієї чи іншої групи?
3. Які параметри пароля користувача, на Вашу думку, мають потребу в адмініструванні?
4. Чи може адміністратор довідатися про пароль рядового користувача? Обґрунтуйте Вашу відповідь.
5. Що може зробити адміністратор у випадку, якщо користувач забув пароль?
6. Яким чином може бути встановлене виключне право користувача на доступ до якої-небудь папки?
7. Яка роль системи аудита в безпеці комп'ютера?
8. Для чого потрібні шаблони безпеки?
9. Які погрози безпеці приносить застосування сервісу NetBIOS?

10. Що дає хакеру знання логіна адміністратора на віддаленому комп'ютері?

11. Який збиток наносять комп'ютерній системі DoS-атаки?

Лабораторна робота 8

ДОСЛІДЖЕННЯ СИСТЕМ КРИПТОЗАХИСТУ ІЗ НЕСИМЕТРИЧНИМИ КЛЮЧАМИ

Мета: розглянути програмні засоби захисту інформації з використанням несиметричних ключів.

Завдання:

– з'ясувати переваги використання несиметричних ключів;

– промоделювати процес обміну ключами, які згенеровані за алгоритмом PGP6;

– перевірити дію електронного підпису документів.

Додаткове прикладне програмне забезпечення:

– «WINPGP.6_0» – програмний пакет, за допомогою якого виконується шифрування та/або формування цифрового підпису. Програма потребує інсталяції.

Короткі теоретичні відомості

Шифрування з симетричними ключами має дуже суттєвий недолік, який полягає в тому, що інколи потрібно змінювати ключі, причому для цього треба здійснювати або наочну зустріч для передачі нового ключа, або мати дуже надійний канал зв'язку.

Цих недоліків позбавлені системи шифрування з використанням несиметричних ключів, у яких використовуються два ключ. Обидва ключа генеруються користувачем самостійно на основі відомого алгоритму, при чому вони між собою зв’язані, тобто створюють ключову пару. Один ключ має назву

"відкритий" тому що саме його пересилають по відкритим каналам зв’язку. За допомогою цього ключа можна лише шифрувати документи та іншу інформацію.

Інший ключ має назву "таємний", тому що він залишається у користувача та нікуди і нікому не пересилається. Цей ключ дозволяє розшифровувати документи, які були зашифровані парним йому "відкритим" ключем. Якщо зловмиснику навіть вдасться перехопити "відкритий" ключ, то він за його допомогою не зможе розшифрувати документи, що надходять його власнику.

У загальному вигляді це виглядатиме так: кожен, хто бажає приєднатися до групи користувачів, які обмінюються захищеними повідомленнями, повинен створити свою ключову пару і обмінятися своїм "відкритим" ключем з "відкритим" ключем інших користувачів цієї групи.

Але при обміну ключами треба мати впевненість в тому, що він належить певної особі, а не зловмиснику. Виходом з цієї ситуації є незалежне підтвердження приналежності електронного ключа певному кореспондентові за допомогою спеціального сертифікату, який відправлятиметься разом з "відкритим" ключем, утворюючи з ним нерозривне ціле.

Ще однією стороною застосування електронних ключів є впровадження "електронного підпису", який засвідчує цілісність документа. Для деякого документа, який завіряється "електронним підписом" підраховується хеш-функція його вмісту, ключем для якої і є "електронний підпис". Якщо згодом до документа будуть внесені зміни або правки, то хеш-функція не "зійдеться", і буде ясно, що здійснена фальсифікація.

Порядок виконання роботи

1. За допомогою програми PGP виконати обмін зашифрованою інформацією. Для цього необхідно:

а) за допомогою утиліти PGPkeys створити ключову пару дотримуючись такого порядку:

– виконати запуск PGPkeys через меню Start (Пуск) ® Programs (Програми) ® PGP;

– далі вибрати пункт «*New Key*» в меню «*Keys*», в процесі створення ключової пари необхідно вказати власне ім'я (*Full name*) і будь-яку адресу електронної пошти (*Email address*), оскільки саме ці дані будуть асоційовані програмою з вашими ключами;

– вибрати тип і довжину ключа (Key Pair Type) виконуємо за вказівкою викладача;

– для терміну дії ключа обрати варіант «*Key pair never expires*» (безстроковий);

– пароль повинен мати не менше восьми символів;

– після чергового натиснення кнопки «*Next*» (Далі), виконати генерацію випадкових даних шляхом хаотичного переміщення по екрану курсору миші;

– після завершення генерації ключової пари треба відмовитися від її розсилання через Internet, оскільки ключі передбачається розсилати індивідуально.

б) переслати відкритий ключ всім студентам групи, які працюють у комп'ютерному класі:

– у меню «*Keys*» вибрати пункт «*Export*»;

– вказати теку, де зберігати ключ, і дати файлу ім'я;

– дискетою або через локальну мережу переслати файл з відкритим ключем на комп'ютери всіх адресатів;

в) зібрати в одній теці всі файли з відкритими ключами, що надіслані іншими студентами;

г) підключити кожен надісланий файл з відкритим ключем адресата за допомогою пункту «*Import*» у меню « *Key*»;

д) створити текстовий файл та записати в нього своє прізвище, ім'я, по батькові, дату і номер академгрупи;

е) зашифрувати файл за допомогою кожного надісланого відкритого ключа, для цього потрібно:

– у контекстному меню створеного файлу вибрати «*PGP*»® «*Encrypt*»;

– у переліку відкритих ключів вибрати потрібний подвійним клацанням лівої кнопки миші;

– після натиснення кнопки «*Ok*» в тій же теці буде створено зашифрований файл;

є) відправити файл, зашифрований відповідним ключем, кожному адресату, надіславши його за допомогою дискети або через локальну мережу;

ж) отримати через локальну мережу або дискетою файли від кожного студента, що працює в комп'ютерному класі, розшифрувати отримані файли за допомогою пункту «*PGP*»® «*Dencrypt*» в контекстному меню файлу;

з) об'єднати вміст всіх надісланих файлів, скопіювавши їх в один текстовий файл, який подати викладачу.

2. Обмінятися текстовими файлами, завіреними електронним підписом, таким чином:

а) завірити текстовий файл, створений в п. 1-д, електронним підписом, для цього в контекстному меню створеного файлу вибрати «*PGP*» ® «*Sign*» і ввести пароль;

б) цей текстовий файл разом із супроводжувальним файлом з електронним підписом надати кожному адресату, переславши їх дискетою або через локальну мережу;

в) перевірити збереження вмісту кожного надісланого файлу, викликавши за допомогою подвійного клацання лівої кнопки миші утиліту ідентифікації PGPlog;

г) внести до надісланого текстового файлу зміни, зберегти їх і повторити перевірку вмісту, як і в попередньому пункті;

д) переконатися, що в цьому випадку програма зафіксує факт невідповідності вмісту файлу.

Контрольні запитання та завдання:

1. Чим відрізняються блокові і потокові алгоритми шифрування?
2. Опишіть стисло алгоритм шифрування DES.
3. У чому полягає удосконалення алгоритму DES?
4. Охарактеризувати алгоритм шифрування ГОСТ 28147-89.
5. Охарактеризувати рівні протоколу SSL?
6. На якому принципі побудовано алгоритм RSA?
7. Яким вимогам повинні задовольняти множники P і Q?
8. За яких причин алгоритм RSA вважається «повільним» і як його можна прискорити?
9. На якому принципі засновано алгоритм Діффі-Хеллмана?
10. Яка область застосування алгоритму Діффі-Хеллмана?
11. Які варіанти модифікації алгоритму RSA ви знаєте?

12. Що називається цифровим підписом?

1. Опишіть два підходи до побудови цифрового підпису.
2. Для чого потрібна сертифікація ключів?
3. Для заданої викладачем пари простих чисел P і Q отримати за алгоритмом RSA числові ключі N, D, і E. За допомогою чисел D і N (складових відкритого ключа) виконати шифрування за алгоритмом RSA тексту, що включає прізвище студента, записане великими літерами кирилиці у вигляді ASCII-коду. За допомогою чисел E і N (складових секретного ключа) виконати розшифрування закодованого тексту.

Модуль 4. СИСТЕМА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Лабораторна робота 9

СТВОРЕННЯ СЦЕНАРІЮ ВХОДУ ДО МЕРЕЖ. СТВОРЕННЯ ОБ'ЄКТУ КОРИСТУВАЧА

Мета: отримати практичні навички зі створення сценаріїв входу до мережі та створення об’єктів користувачів.

Завдання:

– розглянути сценаріїв входу до мережі;

– розглянути команди створення об’єктів користувачів.

–

Короткі теоретичні відомості

Сценарій входу. Загальні виначення.

Сценарій входу – це набір інструкцій, якими керується операційна система за час процесу входу до мережі. Повідомлення та команди, котрі розташовані в сценарії входу, виконуються кожного разу, коли користувач входить до мережі. Сценарій входу користувача визначає оточення для конкретного користувача, в цей сценарій входу заносяться команди, унікальні для кожного користувача.

За час процесу входу користувача в систему, FreeBSD виконує декілька дій, а саме: ідентифікацію користувача, ініціалізацію оточення користувача та запуск інтерпретатора команд, котрі прийнято називати оболонкою.

Усі варіанти ОС FreeBSD використовують декілька поширених оболонок. Найбільш відомі наступні:

Sh

Оболонка Bourne –

найбільш поширена в порівнянні з іншими

Ksh

Оболонка Korn – розвиток оболонки Bourne. До неї вміщена історія та редагування командного рядка.

Csh

Оболонка C, котра використовує синтаксис, схожий на синтаксис популярної мови C.

Bash

Оболонка Born Again , менш розповсюджена.

Tcsh

Версія оболонки C з інтерактивним редагуванням командного рядка.

Допоміжні команди

man Команда використовується для відображення сторінок оперативного керівництва FreeBSD, які включають команди, файли, підпрограми.

Приклад: Якщо ви хочете знайти дані про команду find, то виконайте наступну команду: man find

echo Відображення на екрані символів, які ви набираєте з клавіатури

Приклад. Якщо ви хочете вивести на екран напис «Привіт , друже»: echo “Привіт , друже “.

more Використовується для виводу вмісту файлу на екран.

date Потрібна для відображення поточної дати і часу в заданому форматі. Для відображення дати і часу потрібно вказати знак + з послідуючим форматом.

Приклад: Якщо Ви хочете відобразити дату без форматування, то використовуйте команду date без будь-якого дескриптору форматування , як у наступному прикладі:

date Sat Dec 7 11:50:59 EST 2000

tail Використовується для відображення файлу в стандартний формі, починаючи з вказаної точки з початку або ж з кінця файлу. По замовчанню tail відображає останні 10 рядків файлу. Деякі із прапорців:

- n number для початку відображення з вказаного номеру рядка.

З усіма прапорцями можна вказати число з попереднім знаком +/-, якщо +, то команда tail починає обробку з початку файлу. Якщо вказати – чи не вказати ніякого знаку, то tail починає обробку з кінця файлу.

last Застосовується для відображення інформації про входи в систему користувачів.

last –u user – інформація про входи в систему користувача user.

crontab Система FreeBSD оснащена демоном, який працює весь час і котрий буде виконувати завдання за вказаний інтервал. В файлі Ви можете вказати завдання, яке буде виконувати команда crontab, та демон Cron буде перевіряти його при ініціалізації або ж в тому випадку, коли файл поповнюється або модифікується.

Записи, які Ви створюєте в файлі crontab, складаються з наступних полів : хвилина,, година,день місяця, рік, день тижня, команда.

Кожне з полів може мати декілька дискретних значень, визначених комами чи діапазоном значень, або *, яка означає, що підходять всі значення.

Далі йдуть деякі із прапорців, котрі можна використовувати з командою crontab :

- l для виводу на екран файлу crontab;

- e для редагування чи створення файлу crontab.

Приклад. .Якщо Ви хочете відобразити рядок Час спати в 10:20 ранку кожного дня, створіть наступний запис :

20 10 * * * echo “ Час спати ”

Команди обслуговування користувачів

Операції генерування, модифікації та знищення записів в файлах паролів та груп значною мірою залежать від версії ОС FreeBSD.

adduser [-batch username [gid, [gid ...]] [uid comment][password]]

Команда adduser без параметрів виконується в інтерактивному режимі . Вона також може приймати інші стандартні опції, які :

1. Викликають копіювання інформації файлів login чи profile у власні каталоги користувачів.
2. Розташовують нових користувачів у стандартній групі.
3. Визначають початкові розділи home для нових користувачів.
4. Виводять повідомлення про запрошення для нових користувачів.
5. Встановлюють стандартну оболонку для нових користувачів.

6. Вибирають нові ідентифікатори користувачів із раніше визначеної групи чисел.

rmuser username знищує користувача

vipw редагує файл /etc/passwd

В кожній із наведених вище команд : username - реєстраційне ім’я користувача. Це єдиний обов’язковий параметр будь-якої команди.

uid comment – те, що буде зберігатися в полі IID comment

dir – початковий каталог користувача

expire – абсолютна дата, коли користувач втрачає право входу в систему

inactive – число неактивних днів підряд, після яких ім’я користувача блокується

gid- ідентифікатор чи ім’я групи, до якої належить користувач

shell- початкова оболонка для користувача

skell_dir –каталог, котрий містить файли для копіювання в знову створений каталог

uid- унікальний ідентифікатор користувача

команда chpass дозволяє редагувати наступну інформацію про користувача або по замовчанню про поточного користувача: login, password, uid, gid, class, change, expire, full name, office location. office phone, home phone, other information, home directory. shell

Користувач має право змінити інформацію full name, office location, office phone, home phone, other information.

Файл паролів

Формат файлу /etc/passwd містить записи, поля котрих діляться двокрапкою:

username:pswd:uid:gid:user class:passwd change:acct expiration:uid comments:directory:shell

pswd - містить пароль. Воно може бути пустим, що вказує на те, що пароль не потрібний. Поле може містити до 13 символів, які позначають зашифрований рядок пароля користувача.

uid - унікальний ідентифікатор користувача, звичайно це додатне число до 65535. Деякі ідентифікатори зарезервовані :

0: суперкористувач

1-10 : демони та псевдо користувачі

11-99: системні, зарезервовані та важливі користувачі 100+: звичайні користувачі

60001: “nobody”

60002: “noaccess”

gid – числовий стандартний груповий ідентифікатор користувача. Це число відповідає запису в файлі /etc/group.

Поле uid comments є джерелом інформації для ОС. Поле повинно містити справжнє ім’я користувача, код компанії або офісу, номер телефону офісу, домашній телефон.

directory - цей каталог дається користувачеві після входу в систему , але перед виконанням його персональних файлів запуску.

shell визначає інтерпретатор команд

class не використовується, але призначене для визначення класу атрибутів користувача.

pswd change визначає час зміни пароля в секундах. котрі пройдуть з 1 січня 1970 року 00:00.

acct expiration число секунд , котрі пройдуть з цього ж моменту до припинення дій бюджету.

Файл груп

Файл /etc/group є частиною загальної схеми FreeBSD. Шаблон запису з поділом полів двокрапкою має наступний вигляд:

group_name:password:group_id: list

group_id містить текстове ім’я групи

password є заповнювачем для зашифрованого пароля груп

group_name містить унікальне числове значення групи

list містить список з поділом користувачів комами, котрі належать до даної групи.

Порядок виконання роботи

1. Запустіть віртуальну машіну FreeBSD за допомогою програми VirtualPC 2007.
2. Ввійдіть в систему під користувачем root та паролем 123456.
3. Додайте нового користувача.
4. Змініть інформацію про користувача: телефон, адресу офісу, строк дії облікового запісу користувача 31.12.поточний рік.
5. Змініть ім'я користувача шляхом редагування файлу passwd.
6. Створіть файл crontab для користувача та редагуйте його.

Контрольні запитання:

1. Яке призначення файлу crontab?
2. За допомогою яких команд можна додати нових користувачів?
3. В якому файлі розміщені паролі користувачів?
4. Назвіть зарезервовані ідентифікатори користувачів.

Лабораторна робота 10

БЕЗПЕКА В МЕРЕЖЕВІЙ ОПЕРАЦІЙНІЙ СИСТЕМІ FREEBSD

Мета: одержати практичні навички по забезпеченню безпеки в мережевій операційній системі FreeBSD.

Завдання: розглянути команди забезпечення безпеки в операційній системі FreeBSD

Короткі теоретичні відомості

Принципи захисту

Оскільки ОС FreeBSD з самого свого зародження задумувалась як багатокористувачева операційна система, у ній завжди була актуальною проблема авторизації доступу різних користувачів до файлів файлової системи.

Ідентифікатори користувача та групи користувачів

З кожним виконуваним процесом в ОС FreeBSD зв’язується реальний ідентифікатор користувача (real user ID), діючий ідентифікатор користувача (effective user ID) і збережений ідентифікатор користувача (saved user ID). Всі ці ідентифікатори встановлюються за допомогою системного виклику setuid, який можна виконувати тільки в режимі суперкористувача. Аналогічно, з кожним процесом зв’язуються три ідентифікатори групи користувачів - real group ID, effective group ID і saved group ID. Ці ідентифікатори встановлюються привілейованим системним викликом setgid. При вході користувача в систему програма login провіряє, чи користувач зареєстрований в системі і знає правильний пароль (якщо він встановлений), створює новий процес і запускає в ньому потрібний для даного користувача shell. Але перед цим login установлює для знов створеного процесу ідентифікатори користувача і групи, використовуючи для цього інформацію, збережену в файлах /etc/passwd і /etc/group. Після того, як з процесом зв’язані ідентифікатори користувача і групи, для цього процесу починають діяти обмеження для доступу до файлів. Процес може отримати доступ до файлу або виконати його (якщо файл містить виконувану програму) тільки в тому разі, якщо збережені обмеження доступу, які відносяться до файлу, дозволяють це зробити. Однак у деяких випадках процес може змінити свої права за допомогою системних викликів setuid і setgid, а іноді система змінює права доступу процесу автоматично.

Розглянемо, наприклад, наступну ситуацію. У файл /etc/passwd заборонений запис усім, крім суперкористувача (суперкористувач може писати в будь-який файл). Цей файл, крім іншого, містить паролі користувачів і кожному користувачу дозволяється змінювати свій пароль. Мається спеціальна програма /bin/passwd, що змінює паролі. Однак користувач не може зробити це навіть за допомогою цієї програми, оскільки запис у файл /etc/passwd заборонений. У системі FreeBSD ця проблема розв’язується в такий спосіб. При виконуваному файлі може бути зазначено, що при його запуску повинні встановлюватися ідентифікатори користувача і/або групи. Якщо користувач запитує виконання такої програми (за допомогою системного виклику exec), то для відповідного процесу встановлюються ідентифікатор користувача, що відповідає ідентифікатору власника виконуваного файлу і/або ідентифікатор групи цього власника. Зокрема, при запуску програми /bin/passwd процес одержить ідентифікатор суперкористувача, і програма зможе зробити запис у файл /etc/passwd. І для ідентифікатора користувача, і для ідентифікатора групи реальний ID є істинним ідентифікатором, а діючий ID - ідентифікатором поточного виконання. Якщо поточний ідентифікатор користувача відповідає суперкористувачу, то цей ідентифікатор і ідентифікатор групи можуть бути перевстановлені в будь-яке значення системними викликами setuid і setgid. Якщо ж поточний ідентифікатор користувача відрізняється від ідентифікатора суперкористувача, то виконання системних викликів setuid і setgid приводить до заміни поточного ідентифікатора істинним ідентифікатором (користувача або групи відповідно).

Захист файлів

Захист файлів від несанкціонованого доступу в ОС FreeBSD ґрунтується на трьох фактах. По-перше, з будь-яким процесом, що створює файл, асоційований деякий унікальний у системі ідентифікатор користувача (UID - User Identifier), що надалі можна трактувати як ідентифікатор власника знов створеного файлу. По-друге, з кожен процесом, що намагається одержати деякий доступ до файлу, зв'язана пара ідентифікаторів - поточні ідентифікатори користувача і його групи. По-третє, кожному файлу однозначно відповідає його описувач - і-вузол.

Права доступу до файлу

В операційній системі FreeBSD існують три базових класи доступу до файлу, у кожнім з який установлені відповідні права доступу:

User access (u) Для власника-користувача файлу

Group access (g) Для членів групи, що є власником файлу

Other access (o) Для інших користувачів (крім суперкористувача)

FreeBSD підтримує три типи прав доступу для кожного класу: на читання (read, позначається символом r), на запис (write, позначається символом w) і на виконання (execute, позначається символом x).

За допомогою команди *ls -l* можна одержати список прав доступу до файлу.

Розглянемо, наприклад, права доступу до файлу a.out:

Таблиця 10.1.

Тип файлу

Права власника- користувача

Права власника- групи

Права інших користувачів

-

Rwx

r-x

r--

Звичайний файл

Читання, запис, виконання

Читання і виконання

Тільки читання

Права доступу можуть бути змінені тільки власником файлу або суперкористувачем (superuser) — адміністратором системи. Для цього використовується команда *chmod(l).* Нижче приведений загальний формат цієї команди.

chmod [u g o a] [+ - =] [r w x]   file1 file2 …

Як аргументи команда приймає вказівку класів доступу ('u' — власник-користувач, g' — власник-група, 'o' — інші користувачі, 'a' — усі класи користувачів), права доступу ('r' — читання, 'w' — запис і 'x' — виконання) і

операцію, яку необхідно зробити ('+' — додати, '-' — видалити і '=' — привласнити) для списку файлів *filel, file2* і т.д. Наприклад, команда

$ chmod g-wx ownfile

позбавить членів групи-власника файлу ownfile права на запис і виконання цього файлу.

В одній команді можна задавати різні права для декількох класів доступу, розділивши їх комами. Можна установити відразу всі дев'ять прав доступу, використовуючи числову форму команди *chmod(l):*

$ chmod 754 *

Число визначається в такий спосіб: потрібно представити права доступу в двійковому виді (0 — відсутність відповідного права, 1 — його наявність) і кожну тріаду, що відповідає класу доступу, у свою чергу перетворити в десяткове число.

Таблиця 10.2.

Власник	Група	Інші
r w x	r - x	r - -

111

101

100

7

5

4

Таким чином, наведений приклад еквівалентний наступній символьній формі *chmod(l)'.*

$ chmod u=rwx, g=rx, o=r

Значення прав доступу різне для різних типів файлів. Для файлів операції, які можна робити, випливають із самих назв прав доступу. Наприклад, щоб переглянути вміст файлу командою *cat(l),* користувач повинний мати право на читання (r). Редагування файлу, тобто його зміна, передбачає наявність права на запис (w). Нарешті, для того щоб запустити деяку програму на виконання, потрібно мати відповідне право (x).

Для каталогів ці права мають інший зміст, а для символічних зв'язків вони взагалі не використовується, оскільки контролюються цільовим файлом.Право читання каталогу дозволяє одержати імена (і тільки імена) файлів, що знаходяться в даному каталозі. Щоб одержати додаткову інформацію про файли каталогу (наприклад, докладний лістинг команди *ls* -/), системі прийдеться "заглянути" у метадані файлів, що вимагає права на виконання для каталогу. Права r і x діють незалежно, право x для каталогу не вимагає наявності права м, і навпаки..

Паролі

Наявність пароля дозволяє захистити ваші дані, а можливо (якщо суперкористувач) і всю систему в цілому. Призначити або змінити пароль можна командою *passwd(l).* Звичайний користувач може змінити свій пароль, адміністратор може призначити пароль будь-якому користувачу.

Перед запуском програми *passwd(l)* варто тримати в голові загальне правило вибору пароля: пароль повинний добре запам'ятовуватися і бути важким для підбору.

Порядок виконання роботи

1. Ввійти в мережу під ім'ям root

2. Створити свій особистий об'єкт myname і домашню директорію

2.1 Створимо користувача myname

2.2 Створити домашні директорії

2.3 Переконатися в створенні директорій

2.4 Перевірити їх права доступу

3. Призначити користувачу myname і групі myname права, необхідні для зміни інформації

3.1 Забрати усі права доступу

3.2 Призначити права доступу, для того щоб користувач і група myname одержали право на можливість зміни інформації в об'єкті

4. Призначити користувачу myname і групі myname права, необхідні для того, щоб вони могли запускати файли, що виконуються

5. Призначити користувачу myname і групі myname права, необхідні для того, щоб вони могли проглядати директорію

6. Вийти з мережі

7. Ввійти в мережу під ім'ям myname

8.Перевірити права доступу

8.1 Перевірити права доступу директорій folder і mydir

8.2 Переконатися, що в користувача myname є права тільки на директорію folder

9. Видалити всі створені об'єкти

9.1 Видалити користувача myname

Контрольні питання:

1. Що таке ідентифікатор користувача?
2. Назвіть три базові класи доступу до файлу?
3. За допомогою якої команди можна змінити права доступу до файлу?
4. Яку інформацію зберігає система про користувача?

Лабораторна робота 11

РОЗСИЛАННЯ ПОШТИ В МЕРЕЖЕВІЙ ОПЕРАЦІЙНІЙ СИСТЕМІ FREEBSD

Мета: одержати практичні навички у розсиланні пошти з використанням команди MAIL у мережевій операційній системі FreeBSD.

Завдання: розглянути команди розсиланні пошти у операційній системі FreeBSD

Короткі теоретичні відомості

Команда mail - відправка і отримання пошти

Синтаксис:

mail [-iInv] [-s Коментар] [-c адреса пересилки]

[-b копія адрес пересилки] адресат

[-опції команди SENDMAIL ...]

mail [-iInNv] -f [ім’я користувача]

mail [-iInNv] [-u раніше заданий киристувач]

Mail - це побудована на платформі UNIX система з вбудованим інтерфейсом для обробки поштових повідомлень.

Опції команди MAIL

-v - режим, коли всі деталі пересилки або прийому повідомлень виводяться на екран;

-i - режим ігнорування сигналів збросу, що поступають зі сторони телефонної лінії або мережі;

-N - вивід на екран заголовків повідомлень при роботі з конкретною папкою, в якій вони зберігаються;

-s вказувати мету листа;

-c, -b посилати копії листа списку користувачів, вказаних через кому.

В самому початку mail виконає команди, які прописані в файлах /usr/share/misc/mail.rc, /usr/local/etc/mail.rc і /etc/mail.rc. Потім буде запущений файл ~/.mailrc. Mail перевірить наявність нових повідомлень в поштовій скринці користувача, а також наявність уже отриманих повідомлень в поштовій скриньці.

Якщо в командному рядку була вказана команда mail без параметрів, система увійде в режим роботи з поштовими повідомленнями. При цьому командний рядок буде мати вигляд &

Нижче наведені команди для роботи в цьому режимі

t <список номерів поштових повідомлень через пробіл>

вивести на екран текст повідомлення

n перейти до виводу на екран наступного повідомлення

e <список номерів поштових повідомлень через пробіл>

відредагувати повідомлення

f < список номерів поштових повідомлень через пробіл >

вивести на екран заголовки повідомлень

d < список номерів поштових повідомлень через пробіл >

відмітити повідомлення для подальшого знищення при виході з режиму роботи з поштою

s < список номерів поштових повідомлень через пробіл >

приєднати до повідомлення файл

u < список номерів поштових повідомлень через пробіл>

зняти помітку з повідомлень, які або помічені для видалення при виході з системи

R < список номерів поштових повідомлень через пробіл >

відповісти на повідомлення тому, хто його прислав

r < список номерів поштових повідомлень через пробіл >

відповісти всім на повідомлення

m <список користувачів через пробіл >

надіслати поштове повідомлення всім вказаним в списку

користувачам

! Вийти з системи роботи з поштовими повідомленнями

Порядок виконання роботи

1. Зайти до мережі і переглянути поштове повідомлення.
2. Вивести декілька поштових повідомлень, а також заголовки усіх повідомлень.
3. Написати і відправити лист іншому користувачеві.
4. Вивести на екран та відредагувати текст повідомлення.
5. Знищити написане повідомлення.

Контрольні питання:

1. За допомогою якого протоколу і якої служби відбувається розсилання пошти?
2. Які основні файли необхідно зконфігурувати для розсилання пошти?
3. За допомогою яких команд відбувається розсилання пошти?

Лабораторна робота 12

ВЗАЄМОДІЯ МІЖ КОРИСТУВАЧАМИ

Мета: отримати практичні навички взаємодії між користувачами, які знаходяться в мережі, в рамках операційної системи FreeBSD.

Завдання: розглянути команди для взаємодії корістувачів в операційній системі FreeBSD

Короткі теоретичні відомості

Отримання інформації про користувачів системи

Команда w.

Для отримання інформації про користувачів, що увійшли в систему та зараз в ній знаходяться, використовується команда w.

[sarge@moron][~]

[12:27:16][p0]>w

12:27пп up 23 days, 13:57, 2 users, load averages: 0.00, 0.00, 0.00

USER TTY FROM LOGIN@ IDLE WHAT

sarge p0 tie 12:17пп - w

corvin p1 shamrocky 12:27пп 2 -bash (bash)

При запуску без параметрів команда видає ім´я користувача, спосіб входу в мережу (реальна/віртуальна консоль, FTP), хост, з якого виконується доступ, час, в який було виконано вхід в систему, час простою в хвилинах та поточну програму, яку використовує користувач в даний момент. У заголовку друкується поточний час, час роботи системи, кількість зареєстрованих користувачів. Також у заголовку друкується середня завантаженість системи - кількість процесів, що знаходяться у черзі на виконання від 1.5 до 15 хвилин.

Для отримання інформації про запущені користувачем програми, виконайте команду w –d.

Команда whodo

Команда whodo датує і виводить у форматованому вигляді інформацію, що знаходиться у файлах /etc/utmp і /etc/ps_data.

У заголовку видається поточна дата, час та ім'я системи. Для кожного активного користувача видається ім'я керуючого терміналу, вхідне ім'я, час початку сеансу роботи і далі, список активних процесів, що належать користувачеві. Список містить ім'я керуючого терміналу, ідентифікатор процесу, використаний час ЦП у секундах, а також ім'я процесу.

/etc/whodo

Thu Dec 19 12:00:16 1988

U10I

console hindin   10:56

console   145   0:43 rk.20.01

tty2   galat   10:48

tty2   112   1:51 rk.20.01

tty2   242   0:00 whodo

tty5   dmitriew 10:49

tty5 244 0:01 sh

Команда finger.

Для схожих цілей використовується команда finger без параметрів.

[sarge@moron][~]

[12:27:23][p0]>finger

Login Name TTY Idle Login Time Office Phone

corvin Konstantin Zagorodni p1 6 пт 12:27

sarge Sergey P. KRUKOVSKY p0 пт 12:17

Для отримання детальної інформації про конкретного користувача, виконайте finger з параметром – ім´ям користувача. Ім´я може бути як локальним (arrey), так і включати повне ім´я сервера (arrey@moron.tgh.kiev.ua).

[12:33:45][p0]>finger arrey

Login: arrey Name: Basil Y. ZAULICHNIY

Directory: /home/arrey Shell: /usr/local/bin/bash

Last login чт 20 дек 21:28 (EET) on ttyp2 from shamrocky.tgh.nt

New mail received чт 20 дек 19:43 2001 (EET)

Unread since чт 20 дек 18:40 2001 (EET)

No Plan.

На виході команда видає персональні дані про користувача: повне ім´я, шлях до домашньої директорії, використовуваний шел, час та місце, з якого був проведений останній логін та ситуація з поштовою скринькою.

Команда whois

За допомогою команди whois можна отримати інформацію про користувача Internet. Синтаксис команди:

whois [-h hostname] name ...

Whois передивляється записи у базі даних Network Information Center (NIC) і здійснює пошук вказаного імені у всіх типах записів (name, nicknames, hostname, net address) у базі даних.

Для використання цієї команди має бути доступ до Інтернет.

Команда last.

Дозволяє отримати список користувачів, що заходили в систему за останній проміжок часу. Команда last перевіряє файл wtmp, який включає записи про входження та виходи з системи.

[sarge@moron][/etc]

[12:41:41][p0]>last

corvin ttyp1 tie пт 21 дек 12:27 still logged in

```
sarge          ttyp0    tie    пт  21 дек 12:17   still logged in

sarge          ftp      tie   пт  21 дек 11:25 - 11:32  (00:07)

sarge          ttyp0    tie   пт  21 дек 11:21 - 11:32  (00:10)

taren      ftp      spitfire    пт  21 дек 01:20 - 01:21  (00:00)

taren      ttyp3    spitfire    чт  20 дек 23:14 - 00:35  (01:21)

taren      ttyp2    spitfire    чт  20 дек 23:13 - 00:35  (01:21)

taren      ttyp0    spitfire    чт  20 дек 22:58 - 00:35  (01:37)

arrey      ttyp2    shamrocky   чт  20 дек 21:28 - 21:49  (00:21)

wtmp begins сб   1 дек 19:33:39 2001
```

Якщо команда видає повідомлення про неможливість видачі списку, створіть файл /var/log/wtmp за допомогою команди

```
echo > /var/log/wtmp
```

Команда ac.

Інформацію про час знаходження користувача в системі в годинах можна отримати за допомогою команди ac. При запуску команди ac без параметрів, система видає сумарний час знаходження усіх користувачів в системі:

[temerline@hammer][~]

[14:30:30][p0]>ac

total 103.55

Для отримання інформації про час знаходження в системі окремого користувача, виконайте команду ac з параметром – ім´ям користувача:

[sarge@hammer][~]

[14:31:38][p0]>ac arrey

total 0.69

Інформацію про всіх користувачів, аккаунти на яких заведені на сервері, можна отримати за допомогою команди ac –p

[sarge@hammer][~]

[14:32:24][p0]>ac -p

lu 0.26

greg 0.00

arrey 0.69

sarge 13.41

dialout 0.62

ppp 88.59

ftp 0.01

ap 0.01

total 103.60

Просортований за алфавітом результат виконання цієї ж команди отримується таким чином.

[sarge@hammer][~]

[14:35:41][p0]>ac -p | sort -n

ap 0.01

arrey 0.69

dialout 0.62

ftp 0.01

greg 0.00

lu 0.26

ppp 88.59

sarge 13.44

total 103.62

Взаємодія між користувачами системи

Знаходячись в системі, ви можете надіслати іншому користувачеві текстове повідомлення за допомогою команди write <user>.

Більш широкі можливості надає команда talk. Виконавши talk <user>, ви викликаєте користувача на розмову. Він може підтвердити своє бажання говорити з вами, виконавши команду talk з параметром – вашим ім´ям. Тоді створюється сеанс розмови, екран ділиться навпіл, в одній половині набираєте ви, а в іншій бачите, що набирає ваш співрозмовник.

Для того, щоби працювали дві вищенаведені команди, ваш партнер повинен приймати повідомлення. Змінити стан “приймаю”/”не приймаю” можна за допомогою команди mesg.

[sarge@moron][/etc]

[12:52:46][p0]>mesg y

[sarge@moron][/etc]

[12:57:46][p0]>mesg

is y

[sarge@moron][/etc]

[12:57:48][p0]>mesg n

[sarge@moron][/etc]

[12:57:52][p0]>mesg

is n

[sarge@moron][/etc]

[12:57:54][p0]>

Для настройки мережених інтерфейсів використовується команда ifconfig. При запуску без параметрів вона видає стан наявних в системі мережевих інтерфейсів.

[root@hammer][/home/sarge]

[14:47:01][p0]>ifconfig

rl0: flags=8843<UP,BROADCAST,RUN,SIMPL,MULTST> mtu 1500

inet 194.183.188.200 netmask 0xffffff00 broadcast

194.183.188.255

ether 00:c0:26:2f:1e:7a

media: autoselect (none) status: active

supported media: autoselect 100baseTX <full-duplex> 100baseTX 10baseT/UTP <full-duplex> 10baseT/UTP 100baseTX <hw-loopback>

ed0: flags=8843<UP,BROADCAST,RUN,SIMPL,MULTST> mtu 1500

inet 198.0.0.1 netmask 0xffffff00 broadcast

198.0.0.255

ether 00:80:ad:38:c5:2e

lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500

faith0: flags=8000<MULTICAST> mtu 1500

gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280

gif1: flags=8010<POINTOPOINT,MULTICAST> mtu 1280

gif2: flags=8010<POINTOPOINT,MULTICAST> mtu 1280

gif3: flags=8010<POINTOPOINT,MULTICAST> mtu 1280

lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384

inet 127.0.0.1 netmask 0xff000000

ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500

sl0: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 552

tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500

tun1: flags=8010<POINTOPOINT,MULTICAST> mtu 1500

Порядок виконання роботи

1. Дізнайтеся, хто в даний момент разом з вами знаходиться в системі.
2. Які програми запущені від імен других користувачів?
3. Дізнайтеся про деталі свого аккаунта або аккаунта адміністратора сервера.
4. Отримайте інформацію про останні входи в систему.
5. Визначте ваш час знаходження в системі, час знаходження в системі інших користувачів.
6. Поспілкуйтеся з іншим користувачем за допомогою внутрішніх засобів системи.

Контрольні питання:

1. Як перевірити наявність користувачів у системі?
2. Як довідатися про IP-адресу вашого комп`ютеру?
3. Чим відрізняються IP-адреса та MAC-адреси?

Модуль 5. ЗАХИЩЕНІ ВІРТУАЛЬНІ МЕРЕЖІ ТА ПРОТОКОЛИ БЕЗПЕКИ

Лабораторна робота 13

КОНФІГУРУВАННЯ ПАРАМЕТРІВ TCP/IP ДЛЯ МЕРЕЖЕВИХ ІНТЕРФЕЙСІВ

Мета: отримати практичні навички із структуризації комп`ютерних мереж, навчитися конфігурувати параметри мережевого інтерфейсу, перевірити стан та працездатності комп`ютерної мережі, навчитися відстежувати маршрут проходження пакетів.

Завдання: розглянути команди для конфігурування параметрів TCP/IP для мережевих інтерфейсів

Короткі теоретичні відомості

Процес побудови мережі складається з декількох етапів:

- планування фізичної й логічної структури мережі;
- призначення IP-адрес;
- установка мережевих апаратних засобів;
- настроювання всіх хостів на конфігурування мережевих інтерфейсів під час початкового завантаження;
- настроювання демонів маршрутизації та (або) статичних маршрутів.

Одержання та призначення IP-адрес

Зазвичай кажуть, що IP-адресу призначають конкретній host-машині. Насправді адреси призначають мережевим інтерфейсам, а не машинам. Якщо в машини декілька інтерфейсів, у неї буде кілька адрес, як мінімум по одному в кожній мережі. В кожної адреси буде свій номер мережі, що відбиває той факт, що ці інтерфейси підключені до різних фізичних мереж.

Призначаючи машині IP-адресу, потрібно вказати відповідність між цією адресою й ім'ям машини у файлі /etc/hosts, у доменній системі імен або в мережевій адміністративній базі даних. Ця дозволить звертатися до машин по їх іменам.

Використання файлу /etc/hosts - найпростіший спосіб перетворення імен в IP-адреси. Кожний рядок починається з IP-адреси й містить символьні імена, під якими відома дана адреса.

Конфігурування параметрів TCP/IP

Конфігурування параметрів TCP/IP для локальної мережі можна умовно розділити на 2 частині:

- встановлення мережевих інтерфейсів;
- конфігурування мережевих інтерфейсів.

Для побудови мережі в FreeBSD можуть використовуватися наступні типи інтерфейсів:

- Ethernet/Fast Ethernet
- ATM
- ISDN
- послідовні порти
- паралельні порти

Для правильного конфігурування інтерфейсів Ethernet/Fast Ethernet, необхідно знати ідентифікатор пристрою, використовуваний FreeBSD. Звичайно при першому завантаженні після інсталяції ядро повідомляє про наявність того або іншого Ethernet-інтерфейсу.

Конфігурування мережевих інтерфейсів Ethernet.

Один з головних файлів конфігурації - /etc/rc.conf. У даному файлі вміщується інформація про ім'я комп'ютера, особливості конфігурації, мережевих інтерфейсів, а також які служби запускаються при старті системи.

Початкова ініціалізація файлу починається при установці утилітою /stand/sysinstall. У цьому файлі в секції "Network configuration subsection" перебуває опис мережі.

Спочатку описуються можливі мережеві інтерфейси, а потім команди для настроювання кожного з інтерфейсів.

Наприклад, відомі наступні параметри TCP/IP.

- Адреса мережі 192.168.30.0
- Маска підмережі 255.255.255.0
- Адреса мережевого інтерфейсу для адаптера NE2000 192.168.30.2
- Адреса інтерфейсу ppp для модемного з'єднання 10.18.50.1
- Назва домена company.org
- Ім'я хоста bsdhost.company.org
- Шлюз 192.168.30.1
- Сервер імен DNS 192.168.30.1
- •У файлі конфігурації містяться наступні рядки:

hostname="bsdhost.company.org"

network_interfaces="lo0 ed0 tun0"

ifconfig_lo0="inet lo0 127.0.0.1" # Конфігурується обов'язково

ifconfig_ed0="inet ed0 192.168.30.2 -netmask 255.255.255.0"

ifconfig_tun0="inet tun0 10.18.50.1 -netmask 255.255.255.0"

...defaultrouter="192.168.30.1"

Далі варто визначити перелік серверів імен і доменів. Цей опис можна зробити у файлі /etc/resolv.conf:

search company.org

nameserver 192.168.30.1

Обов'язково звернить увагу на файл /etc/host.conf: рядок bind повинний розташовуватися вище рядка hosts, наприклад так:

% cat /etc/host.conf

 # $Id: ethernet.html,v 1.5 2000/02/24 09:41:11 osa Exp $

 # Default is to use the nameserver first

bind

 # If that doesn't work, then try the /etc/hosts file

hosts

 # If you have YP/NIS configured, uncomment the next line

 # nis

Конфігурування мережевих інтерфейсів утилітою ifconfig

Програма ifconfig використовується для включення й вимикання мережевого інтерфейсу, завдання IP-адреси, широкомовної адреси й пов'язаної з ним маски підмережі, а також для установки інших опцій і параметрів. Вона звичайно виконується під час початкового завантаження, але може застосовуватися й для внесення змін на ходу.

Команда ifconfig зазвичай має наступний формат:

ifconfig *інтерфейс [сімейство] адреса* up *опція ...*

Наприклад:

ifconfig ed0 128.138.240.1 up netmask 255.255.255.0 broadcast

128.138.240.255

Тут *інтерфейс* означає апаратний інтерфейс, до якого застосовується команда. Як правило, це двох-трьох символьне ім'я пристрою, за яким ставиться число. Приклади розповсюджених імен: ed, de, ie, le, ln, en, we, qe, lan. Ім'я інтерфейсу утворюється з ім'я драйвера пристрою, використовуваного для керування їм; звичайно воно відповідає комплекту мікросхем, що використовується в інтерфейсі. Для того, щоб з'ясувати, які інтерфейси є в системі, можна скористатися командою netstat -i.

Завдяки багаторівневій архітектурі мережевого програмного забезпечення з кожним інтерфейсом можна зв'язувати не один, а кілька протоколів. Аргумент *сімейство* показує, протоколи якого рівня потрібно конфігурувати наступними аргументами.

Для випадку використання протоколу IP аргумент *сімейство* повинен мати значення inet. Деякі версії команди ifconfig припускають значення inet, якщо аргумент
сімейство
відсутній; у системі FreeBSD потрібно вказувати його явним чином.

Параметр *адреса* задає IP-адресу інтерфейсу. Як правило, вона дається в традиційному записі із точками, але в більшості систем її можна вказувати як ім'я машини.

Інтерфейс, що закільцьовує звичайно називається lo0. Це - фіктивний елемент апаратури, через який можна маршрутизувати пакети, призначені для самої локальної машини, що дозволяє мережевим протоколам і сервісним програмам функціонувати нормально навіть на автономній машині. Інтерфейс, що закільцьовує потрібно конфігурувати як будь-який інший мережевий інтерфейс; йому варто привласнити IP-адресу 127.0.0.1 (він також відомий під ім'ям localhost).

Ключове слово up включає інтерфейс, а ключове слово down виключає його. Потім ідуть інші опції (їх може бути кілька; значення опцій задаються символічними іменами). Найбільше часто використовувані опції:

netmask задає маску піжмережі для даного інтерфейсу.

broadcast задає широкомовну IP-адресу інтерфейсу в шістнадцятковому записі або записі із точками. Правильна широкомовна адреса - та, у якій всі біти номера машини встановлені в 1.

metric стосується маршрутизації. Звичайно вартість передачі пакета з однієї мережі в іншу становить один

"перехід" (якщо мережі з'єднані безпосередньо, переходів немає). Аргумент опції metric - лічильник (число) переходів, що зв'язується з даним інтерфейсом.

Команда ifconfig *інтерфейс* друкує поточні установки для зазначеного інтерфейсу. У багатьох системах -a розуміються як "всі інтерфейси".

Деякі приклади використання даної утиліти.

 # ifconfig ed0 inet 192.168.30.1 netmask 255.255.255.0

У загальному випадку це повинне працювати, але не завжди. Краще вказати необхідну кількість параметрів. Наприклад:

 # ifconfig ed0 inet 192.168.30.1 netmask 255.255.255.0 media

10base/UTP

Або

 # ifconfig ed0 inet 192.168.30.1 netmask 255.255.255.0 media

10base/UTP broadcast 192.168.30.255

У загальному випадку драйвер вибирає підходящі параметри з'єднання.

Перевірка працездатності мережевих інтерфейсів командою ping

Команда ping служить для примусового виклику відповіді конкретної машини. Для цього використовується дейтаграмма ECHO_REQUEST протоколу ICMP. Це протокол мережевого рівня, що не вимагає наявності серверних процесів на фондованій машині. Більшість версій команди ping працюють у нескінченному циклі, якщо не заданий аргумент "число пакетів". Припинити нескінченне тестування можна за допомогою

комбінації клавіш [Ctrl-C]. Деякі приклади використання команди ping:

% ping tigger

PING tigger.Colorado.EDU (128.138.240.26): 56 data bytes

64 bytes from 128.138.240.26: icmp_seq=0 time=12 ms

64 bytes from 128.138.240.26: icmp_seq=l time=ll ms

64 bytes from 128.138.240.26: icmp_seq=2 time=ll ms

64 bytes from 128.138.240.26: icmp_seq=3 time=ll ms

64 bytes from 128.138.240.26: icmp_seq=4 time=10 ms

^C

------itigger.Colorado.EDU PING Statistics ------

6 packets transmitted, б packets received, 0 % packet loss

round-trip (ms) min/avg/max = 10/11/12

% ping ginkgo

PING ginkgo.Colorado.EDU (128.138.241.3): 56 data bytes

^C

------iginkgo.Colorado.EDU PING Statistics ------

7 packets transmitted, 0 packets received, 100% loss

Інформація про машину tigger містить її IP-адресу, порядковий номер пакета по протоколу ICMP і час повного обходу. Машина ginkgo у другому прикладі швидше за все відключена.

Інформація про стан мережі: команда netstat

Найпоширеніші варіанти використання команди netstat:

• перевірка стану мережевих з'єднань;

• аналіз інформації про конфігурацію інтерфейсів;

• вивчення таблиці маршрутизації;

• одержання статистичних даних про різні мережеві протоколи.

Команда netstat -i показує стан мережевих інтерфейсів. От, приміром, вихідна інформація команди netstat -i, видана на машині host:

netstat -i

Name Mtu Net/Dest Adress Ipkts Ierrs Opkts Oerrs Coll

le0 1500 cu-capp host 51307 452 40114 311 253

iel 1500 cu-cr host 74196 902 79038 103 2271

lo0 1536 127.0.0.0 localhost 1079 0 1079 0 0

Мережі й адреси шлюзів показані - за замовчуванням - у символічній формі: їх числові еквіваленти можна одержати за допомогою опції -n. Два інтерфейси з однією і тією ж адресою host мають різні IP-адреси, які й показує команда netstat -n.

Команда netstat -r видає таблицю маршрутизації ядра.

Пункти призначення й шлюзи можуть показуватися під іменами машин або під IP-адресами. Прапори дають оцінку маршруту: й означає up (активний), g -gateway (шлюз), H - host (машинний). Прапор d (не показаний) позначає маршрут, отриманий у результаті переадресації пакетом ICMP. Прапори G і H разом означають маршрут до машини, що проходить через проміжний шлюз. Інші поля містять статистичні дані про маршрут: поточна кількість TCP-з'єднань із використанням цього маршруту, кількість посланих пакетів і ім'я використаного інтерфейсу.

Відстеження маршрутів проходження IP-пакетів утилітою traceroute

Програма traceroute дозволяє виявляти послідовність шлюзів, через які проходить IP-пакет на шляху до пункту свого призначення. У багатьох системах traceroute відсутній.

Синтаксис команди: traceroute *ім'я машини*

У цієї команди є дуже багато опцій, більшість із яких у повсякденній роботі не застосовуються. Як правило, *ім'я машини* може бути задане в символічній або числовій формі. Вихідна інформація - простий список машин, починаючи з першого шлюзу й закінчуючи пунктом призначення. Наприклад, на деякій машині src команда traceroute dst може видати такий результат:

traceroute to dst (128.138.202.80), 40 byte packets

1 gw1 (128.138.243.120) 3ms 2 ms 2 ms

2 gw2 (128.138.243.41) 3 ms 3 ms 3 ms

3 dst (128.138.202.80) 4 ms 4 ms 4 ms

Ця інформація говорить про те, що для того, щоб потрапити з машини src на машину dst, пакети повинні пройти два наших внутрішніх шлюзи (gw1 і gw2). Крім того, показаний час повного обходу для кожного шлюзу.

Поточний контроль трафіку утилітою tcpdump

Дані програми відносяться до класу інструментів перехоплення пакетів. Вони стежать за трафіком у мережі й реєструють або виводять на екран пакети, які задовольняють певним критеріям, заданим користувачем. Наприклад, можна аналізувати всі пакети, що посилаються на якусь машину або з неї, або TCP-пакети, що відносяться до конкретного мережевого з'єднання.

Запуск програми виконується з командного рядка системи FreeBSD з використанням наступних ключів:

tcpdump [-deflnNOpqSTvx] [-c count] [-F file] [-i interface] [-r file] [-s snaplen] [-T type] [-w file] [expression].

Найбільш важливими опціями є наступні:

c - вихід після перехоплення count пакетів;

d - печатка типу пакета в зручно читаємій формі й вихід;

F - файл file використовується в якості вхідного для вираження фільтрації пакетів;

i - переглядати інтерфейс interface (якщо не вказується, то проглядається

інтерфейс із найменшим номером у системному списку, наприклад, ed0);

n - не перетворювати адреси хостів в імена DNS;

N - не виводити повне DNS-ім'я ;

O - не запускати оптимізатор коду пакетів;

r - читати пакети з файлу file, створеного з використанням опції -w;

w - записувати інформацію у файл file;

Вираз expression дозволяє вказати, які пакети обробляти, а які пропускати.

Найбільш зручним для наступного аналізу є запуск програми у вигляді:

tcpdump -c CNT -i IF

Порядок виконання роботи

1. Перевірити номер IP-адреси своєї машини
2. Включити мережевий інтерфейс за допомогою утиліти ifconfig.
3. Перевірити працездатність мережевих інтерфейсів командою ping.
4. Перевірити інформацію про стан мережі за допомогою команди netstat.
5. Спробувати відстежити маршрут проходження IP-пакетів за допомогою утиліти traceroute.
6. Зробити поточний контроль трафика утилітою tcpdump.

Контрольні питання:

1. З яких етапів складається процес побудови мережі?
2. Яке призначення IP-адреси?
3. Які типи інтерфейсів можуть використовуватися для побудови мережі в FreeBSD?
4. Для чого використовується програма ifconfig?
5. Що таке localhost?

Лабораторна робота 14

КРІПТОГРАФІЧНІ ФУНКЦІЇ В ОПЕРАЦІЙНІЙ СИСТЕМІ FREEBSD

Мета: вивчити принципи захисту інформації в операційній системі FreeBSD, отримати практичні навички із шифрування файлів та паролів, навчитися шифрувати хешуванням, навчитися читати паролі, навчитися генерувати ключі, навчитися кодувати та декодувати файли.

Основні Завдання: розглянути основні команди для шифрування/дешифрування файлів та паролів

Короткі теоретичні відомості

У кожного користувача FreeBSD системи є пароль, пов'язаний з його обліковим записом. Очевидно, що ці паролі повинні бути відомі тільки користувачу і відповідній операційній системі. Для захисту паролів вони шифруються способом, відомим як "однобічний хеш", тобто їх можна легко зашифрувати, але не можна розшифрувати.

Перший спосіб шифрування пароля з появою FreeBSD був заснований на DES, Data Encryption Standard. Це не було проблемою для користувачів, що живуть у США, але оскільки вихідний код DES не можна було експортувати зі США

Рішення було в поділі бібліотек шифрування, щоб користувачі в США могли встановлювати і використовувати бібліотеки DES, а в інших користувачів був метод шифрування, дозволений до експорту. Так FreeBSD прийшла до використання MD5 як метод шифрування за замовчуванням. На даний момент бібліотека підтримує хеши DES, MD5 і Blowfish.

Формат паролів, використовуваних для нових паролів, визначається параметром passwd_format у /etc/login.conf, що може приймати значення des, md5 чи blf.

CRYPT(3X)

Функція CRYPT(3X) призначена для шифрування паролю і файлу.

des_crypt - це функція шифрування паролю. Вона утворена на основі алгоритму шифрування перемішуванням, призначення якого - унеможливити використання апаратних засобів пошуку.

Аргумент key являє собою пароль, введений користувачем. Salt - це двосимвольний рядок, вибирається з множини [a-z A-Z 0-9 ./]. Він налаштовує алгоритм перемішування на один з 4096 варіантів, після чого пароль використовується як ключ для циклічної шифрування деякої рядкової константи. Значення, що повертається, вказує на зашифрований пароль.

Функції des_setkey і des_encrypt забезпечують доступ (на досить примітивному рівні) до алгоритму перемішування. Аргумент функції des_setkey - це символьний масив довжиною 64 символи, що містить тільки символи з числовим значенням 0 і 1 і відображаючий бітовий рядок. У кожній з восьми груп по 8 символів-"бітів" найменш значущий "біт" ігнорується. Отриманий 56-бітовий ключ передається комп'ютеру.

Аргумент функції des_encrypt - символьний масив довжиною 64, що містить тільки символи з числовим значенням 0 і 1. Аргумент на місці перетворюється в масив того ж виду, що відображає біти аргументу після застосування алгоритму перемішування з ключем, встановленим у функції des_setkey. Якщо flag дорівнює 0, аргумент зашифровується, у протилежному випадку розшифровується.

Функції crypt, setkey і encrypt є "фасадними". Вони викликають функції des_crypt, des_setkey і des_encrypt

відповідно.

Run_setkey повертає -1, якщо з'єднання з crypt(1) не встановлено (це можливо в міжнародній версії FreeBSD, де утиліта crypt(1) недоступна). Якщо функції переданий порожній ключ, повертається 0, в інших випадках повертається 1. Run_crypt повертає -1 при неуспішному читанні чи запису по каналу, утвореному run_setkey, у противному випадку повертається 0.

Для шифрування даних використовуються наступні команди:

crypt - використовується для кодування і декодування файлів. Команда зчитує дані зі стандартного введення чи введення з термінала і записує в стандартний чи висновок на термінал;

makekey - команда задає ключ шифрування;

ed -x - команда для виклику редактора ed для редагування файлу, що вже був зашифрований, і створює новий зашифрований файл;

vi -x - команда для виклику редактора vi для редагування файлу, що вже був зашифрований, і створює новий зашифрований файл;

ex -x - команда для виклику редактора ex для редагування файлу, що вже був зашифрований, і створює новий зашифрований файл;

edit -x - команда використовується для виклику редактора edit для редагування файлу, що вже був зашифрований, і створює новий зашифрований файл;

X - команда шифрує файл під час роботи редактора (ed, ex чи edit).

Команда crypt

Команда кодує і декодирует файли для їхнього захисту, використовуючи пароль (ключ). Зашифрований файл неможливо вважати, якщо використовується неправильний пароль для його декодування.

Файл можна зашифрувати в режимі shell, використовуючи команду crypt, чи в режимі редагування, використовуючи параметр -x чи X.

Формат команди для шифрування файлу наступний

crypt < oldfile > newfile

де oldfile - файл, якій шифрується; newfile - зашифрований файл.

Можна також увести команду в такий спосіб:

crypt key < oldfile > newfile

де key - пароль для шифрування; oldfile - файл, якому потрібно зашифрувати; newfile - зашифрований файл.

Для дешифрування файлу використовується команда:

crypt key < crypt_file > new_filename

де key - пароль для шифрування; crypt_file - файл, якому потрібно дешифрувати; new_filename - дешифрований файл.

Шифрування і дешифрування за допомогою редакторів

Редактори (ed, edit, ex чи vi) можна використовувати або для редагування існуючого файлу, що був зашифрований, або для створення нового зашифрованого файлу. Формат команди для виклику редакторів наступний:

ed -x filename

edit -x filename

ex -x filename

vi -x filename

де -x - параметр, що використовується або для редагування існуючого зашифрованого файлу, або для створення нового файлу; filename - ім'я створюваного чи файлу, що редагується.

Для дешифрування файлу використовуйте команду crypt.

Команда X редактора - інший спосіб шифрування файлу в режимі редагування. Команда X працює тільки з редакторами ed, edit чи ex.

Порядок виконання роботи

1. Реалізуйте шифрування паролю і файлу за допомогою функції crypt(3x).
2. Реалізуйте шифрування хешуванням паролю за допомогою функції crypt(3c).
3. Задайте ключ шифрування за допомогою функції makekey.
4. Зашифруйте та дешифруйте файл з допомогою редакторів ed, edit, ex чи vi.

Контрольні питання:

1. Назвіть призначення функції crypt в операційній системі FreeBSD.Вкажіть розмір ключа після шифрування.
2. Вкажіть довжину аргумента функції des_encrypt.
3. Назвіть призначення функції makekey.

4. Назвіть відмінності функцій setkey та encrypt.

Лабораторна робота 15

МІЖМЕРЕЖЕВИЙ ЕКРАН В ОПЕРАЦІЙНІЙ СИСТЕМІ FREEBSD

Мета: отримати практичні навички із захисту комп`ютерної мережі за допомогою брандмауера в операційній системі FreeBSD, навчитися настроювати та задавати правила брандмауера ipfw.

Завдання: розглянути основні команди для конфігурування брандмауера в операційній системі FreeBSD

Короткі теоретичні відомості

Брандмауер (міжмережевий екран) - це система або група систем, що реалізують правила керування доступом між двома мережами. Фактичні засоби, за допомогою яких це досягається, дуже різні, але в принципі брандмауер можна розглядати як пару механізмів: один для блокування передачі інформації, а інший - для пропуску інформації. Деякі брандмауери приділяють більше уваги блокуванню передачі інформації, інші - її пропуску.

Брандмауери також важливі, оскільки дозволяють створити єдину "вразливу точку" ("choke point"), де можна організувати захист і аудит. Брандмауери забезпечують важливі функції журналізації й аудита; часто вони дозволяють адміністраторам одержувати звіти про типи й обсяги переданої через них інформації, про кількість спроб злому і т.п.

Брандмауер не може захистити від атак, що виконуються не через нього. Брандмауери не можуть захистити від передачі по більшості прикладних протоколів команд підставними ("троянськими") чи погано написаними клієнтськими програмами. Брандмауер - не панацея, і його наявність не скасовує необхідності контролювати програмне забезпечення в локальних мережах чи забезпечувати захист хостів і серверів.

Утиліта IPFW

IPFW - є системою обліку та фільтрації пакетів, що постійно знаходиться в ядрі, і має утиліту керування ipfw.

Зміна правил IPFW

ipfw [-N] інструкція [індекс] [директива] [log] протокол [опції]

Значення параметрів:

-N Утіліта ipfw працює з адресами вузлів і номерами портів. Опція –N змушує утіліту перетворювати цю інформацію в імена, використовуючи DNS і файл /etc/services. (ця опція не є обовязковою).

інструкція Це дія, що виконується по відношенню до правила.

Допустимі інструкції:

- add – додає правило до списку;
- delete – видаляє правило зі списку;
- list – виводить весь список правил чи тльки задане правило;
- flush – видаляє всі правила зі списку, за виключенням правила по умолчанию;
- resetlog – скидає лічильник співпадінь для правила.

індекс. Це число від 0 до 65535, що означає положення правила в спску. Якщо пропустити цей параметр, система автоматично згенерує номер, який на 100 більший ніж номер останнього правила за виключенням правила по умолчанию (його номер 65535).

директива. Це дія, що виконується по відношенню до пакета. Можливі директиви:

- allow, accept і pass - директиви, що дозволяють приймати пакет (наведені директиви є синонімами, можна використовувати будь-яку для отримання того самого результату);
- deny і reject – директиви, що блокують пакет, перша змушує систему ігнорувати пакет, щоб відправник думав, що пакет загубився або компютер-адресат недоступний, друга змушує повернути відправникові повідомлення про те, що компютер або порт недоступний;

- count – директива, що змушує систему збільшити лічильники правила, але не виконувати інших дій, обробка пакету буде продовжена наступними правилами.

log. Цей параметр змушує утіліту ipfw виводити інформацію про співпадіння на системну консоль.

Протокол. Це протокол перевіряємих пакетів. Найбільш розповсюджені значення – tcp, udp, icmp. Ключовому слову Ip або all відплвідають пакети будь-якого протоколу.

Адреса. Специфікаця адреси має вигляд:

from адреса/маска [порт] to адреса/маска [порт] [via інтерфейс]

Пояснення до компонентів:

- Пакет TCP/IP має адреси відправника та отримувача. Вони повинні задаватися за допомогою ключових слів from і to. Якщо ардеса не є важливою, то вкажіть ключове слово any.
- *Адреса* може бути задана в традиційному вигляді (наприклад, 172.27.145.31) або у вигляді мережевої адреси з маскою формату CIDR (172.27.145.0/24) або маскою чотирьохбайтового формату (172.27.145.0:255.255.255.0). Ключовому слову any відповідає будь-яка адреса.
- Аргумент *порт* – це номер порту для протоколів, які підтримують це поняття (TCP, UDP). Дозволяється не використовувати номер порту, вказувати декілька портів через кому або задавати діапазон портів через дефіс.
- Якщо необхідно, щоб правило застосовувалося до трафіку конкретного мережевого інтерфейса, скористуйтеся ключовим словом *via*. Аргумент інтерфейс представляє собою IP-адресу або імя інтерфейса.

Опції. Утіліта ipfw підтримує різноманітні опції, що задають тип пакета. Деякі опції:

setup – цьому ключовому слову відповідають пакети, що відсилаються при спробі встановити зєднання;

in – вхідні пакети;

out – вихідні пакети;

established – ця опція дозволяє створювати правила, що дозволяють трафік у відповідь.

На початку виконання завдань введіть команду: #kldload ipfw. Вона дозволить випробовувати можливості утіліти ipfw не переконфігуровуючи ядра системи.

Приклади команд для ipfw

1. Написати команду, що змушуватиме систему приймати пакети від будь-якого компютера і пропускати пакети, адресовані будь-якому компютеру. Ця команда матиме вигляд:

ipfw add 65534 allow all from any to any

Ця команда додала правило з номером 65534.

Після введення команди на екрані ви отримаєте наступне:

65534 allow ip from any to any

Це означатиме, що правило додане.

2. Видалити певне правило зі списку (наприклад 65534). Зробити це можна наступною командою:

#ipfw delete 65534

Після введення команди на екрані ви не отримаєте жодних написів.

3. Написати команду, що дозволятиме вашому компютеру отримувати TCP-пакети з будь-якої адреси.

ipfw add 2207 allow tcp from any to 172.23.145.67

Після введення команди на екрані ви отримаєте наступне:

02207 allow tcp from any to 172.23.145.67

IP-адреса вашого компьютера буде іншою.

4. Написати команду, що дозволятиме вашому компютеру відправляти TCP-пакети на будь-яку адресу.

ipfw add allow tcp from 172.23.145.67 to any

Після введення команди на екрані ви отримаєте наступне:

02307 allow tcp from 172.23.145.67 to any

Оскільки індекс не було задано, то система сама згенерувала номер правила.

5. Написати команду, що дозволятиме вашому компютеру обмінюватися UDP-пакетами через 53-й UDP-порт з сусіднім компютером.

ipfw add 3308 allow udp from 172.23.145.66 53 to 172.23.145.67

ipfw add 3309 allow udp from 172.23.145.67 to 172.23.145.66 53

Після введення команди на екрані ви отримаєте наступне:

03308 allow udp from 172.23.145.66 53 to 172.23.145.67

03309 allow udp from 172.23.145.67 to 172.23.145.66 53

6. Продивитися список існуючих правил фільтрації пакетів.

Це можна зробити наступною командою:

ipfw list

Після введення команди на екрані ви отримаєте наступне:

02207 allow tcp from any to 172.23.145.67

02307 allow tcp from 172.23.145.67 to any

03308 allow udp from 172.23.145.66 53 to 172.23.145.67

03309 allow udp from 172.23.145.67 to 172.23.145.66 53

65535 deny ip from any to any

7. Видалити всі правила зі списку. Зробити це можна наступною командою: #ipfw flush

Порядок виконання роботи

1. Активуйте IPFW на FreeBSD.
2. Сконфігуруйте IPFW.
3. Змініть правила IPFW.
4. Запишіть на диск правила IPFW.

5. Очистіть IPFW пакет лічильників.
6. Настройте міжмережевий екран за допомогою ipfw.
7. Визначіть робочу конфігурацію ipfw.

Контрольні питання:

1. Дайте визначення терміну: “брандмауер”.
2. Назвіть основні типи брандмауерів.
3. Поясніть призначення списку правил IPFW.

Лабораторна робота 16

РЕАЛІЗАЦІЯ ПРОТОКОЛУ IPSEC В ОПЕРАЦІЙНІЙ СИСТЕМІ FREEBSD

Мета: отримати практичні навички реалізації методів захисту інформації за допомогою протоколу IPSec.

Завдання: розглянути основні команди для реалізації методів захисту інформації за допомогою протоколу IPSec.

Короткі теоретичні відомості

Щоб прозоро для користувача і програми захистити мережеві дані, сучасні операційні системи використовують протокол IPSec. Цей протокол забезпечує аутентифікацію, конфіденційність, цілісність даних і фільтрацію для TCP/IP трафіку. IPSec реалізований нижче протоколів прикладного рівня і дозволяє захищати сеанс зв'язку будь-якого додатка без його модифікації.

Скорочення IPSec походить від «IP security». IPSec заснований на двох ключових компонентах: аутентифікуючому заголовку (Authentication Header - AH) і інкапсулюючому протоколі безпеки (Encapsulating Security Protocol - ESP). AH надає аутентифікацію, підтверджуючи, що присланий відправником пакет дійсно прийшов від нього, і що цей пакет дійсно містить ті дані, що були послані. ESP шифрує дані в пакеті і може так само надавати послуги аутентифікації.

ESP і AH можуть бути використані в режимі транспорту і тунелю. Транспортний режим надає захищене з'єднання між двома точками, тоді як тунель дає VPN-подібне з'єднання (VPN-з'єднання дозволяє користувачам, що знаходяться поза локальною мережею мати до неї захищений доступ). У транспортному режимі дані пакета підписуються і шифруються. Потім пакет передається за призначенням, де він перевіряється, дешифрується і далі обробляється як звичайний IP пакет.

У тунельному режимі AH або ESP шифрують весь пакет з даними і поміщають результат усередину нового пакета. Цей пакет відправляється на іншу сторону тунелю, де дані розшифровуються і перевіряються. Після цього відновлений вихідний пакет обробляється як звичайно, і, якщо необхідно, пересилається до необхідної мети.

Основні процедури для впровадження IPsec на ОС FreeBSD:

- Конфігурування ядра
- Конфігурування процесу обміну ключами
- Встановлення політик безпеки .

Архітектура IPSec

IP Security - це комплект протоколів, що стосуються питань шифрування, аутентифікації і забезпечення захисту при транспортуванні IP-пакетів; у його склад зараз входять майже 20 пропозицій по стандартах і 18 RFC.

Специфікація IP Security (відома сьогодні як IPsec) розробляється Рабочою групою IP Security Protocol IETF. Робоча група IP Security Protocol розробляє також і протоколи керування ключовою інформацією. У задачу цієї групи входить розробка Internet Key Management Protocol (IKMP), протоколу керування ключами прикладного рівня, що не залежить від використовуваних протоколів забезпечення безпеки.

Специфікація ISAKMP описує механізми узгодження атрибутів використовуваних протоколів, у той час як протокол Oakley дозволяє встановлювати сесійні ключі на комп'ютери мережі Інтернет. Гарантії цілісності і конфіденційності даних у специфікації IPsec забезпечуються за рахунок використання механізмів аутентифікації і шифрування відповідно.

Перед передачею по Internet ці пакети інкапсулюються в інші IP-пакети. IPSec підтримує кілька типів шифрування, у тому числі Data Encryption Standard (DES) і Message Digest 5 (MD5).

Політика безпеки

Політика безпеки зберігається в SPD (База даних політики безпеки). SPD може вказати для пакета даних одне з трьох дій: відкинути пакет, не обробляти пакет за допомогою IPSec, обробити пакет за допомогою IPSec. В останньому випадку SPD також указує, який SA необхідно використовувати (якщо, звичайно, що підходить SA уже був створений) або вказує, з якими параметрами повинний бути створений новий SA.

ISAKMP/Oakley

Протокол ISAKMP визначає загальну структуру протоколів, що використовуються для встановлення SA і для виконання інших функцій керування ключами. ISAKMP підтримує кілька Областей Інтерпретації (DOI), однієї з яких є IPSec-DOI. ISAKMP не визначає закінчений протокол, а надає "будівельні блоки" для різних DOI і протоколів обміну ключами.

Протокол Oakley – це протокол визначення ключа, що використовує алгоритм заміни ключа Діффі-Хеллмана. Протокол Oakley підтримує ідеальну пряму безпеку (Perfect Forward Secrecy – PFS). Наявність PFS означає неможливість розшифровки усього трафіку при компрометації будь-якого ключа в системі.

IKE

IKE – протокол обміну ключами за замовчуванням для ISAKMP, на даний момент є єдиним. IKE знаходиться на вершині ISAKMP і виконує, власне, установлення як ISAKMP SA, так і IPSec SA. IKE підтримує набір різних примітивних функцій для використання в протоколах. Серед них можна виділити хеш-функцію і псевдовипадкову функцію (PRF).

Конфігурування ядра FreeBSD

Для використання функцій протоколу IPSec рекомендується використовувати версію FreeBSD 4.2-STABLE або більш пізню. Підтримка IPSec ядром системи обов'язкова для функціонування будь-якого режиму цього протоколу.

1. Щоб перевірити чи підтримує поточна конфігурація ядра протокол IPSec, виконаєте наступну команду:

sysctl -a grep ipsec

Якщо Ви не одержали ніякого результату, то ядро буде потрібно переконфігурувати.

2. Скопіюйте файл стандартних настроювань ядра (або файл специфічних настроювань даної системи) у файл IPSEC:

cp /usr/src/sys/i386/conf/GENERIC /usr/src/sys/i386/conf/IPSEC

3. Використовуючи текстовий редактор, відредагуйте цей файл. vi /usr/src/sys/i386/conf/IPSEC

Для підтримки IPSec системою в конфігураційний файл ядра необхідно додати наступні рядки:

Options IPSEC # Підтримка IP Security

Options IPSEC_ESP # Шифрування для IP Security

Options IPSEC_DEBUG # Відладочна інформація

Якщо Ви плануєте використовувати IPSec у тунельному режимі, додайте також псевдо-інтерфейс для тунелю:

Pseudo-device gif 4

1. Після додавання варто перекомпілювати й установити нове ядро:
2. cd /usr/src

6.make buildkernel KERNCONF=IPSEC && make installkernel KERNCONF=IPSEC

Щоб зміни набрали сили, потрібно перезавантажити систему.

Конфігурування процесу обміну ключами

1. Створення асоціацій безпеки за допомогою утиліти setkey. Підтримка ядром FreeBSD протоколу IPSec означає підтримку протоколів AH і ESP, а також баз даних SAD і SPD. Наступним етапом у настроюванні IPSec є створення асоціацій безпеки (SA), що зберігаються в базі SAD. Це можна зробити вручну, використовуючи утиліту setkey. Наприклад:

```
# setkey –c

add 10.1.1.1 20.1.1.1 esp 9876  -E 3des-cbc "hogehogehogehogehogehoge";

add 20.1.1.1 10.1.1.1 esp 10000 -E 3des-cbc

0xdeadbeefdeadbeefdeadbeefdeadbeefdeadbeefdeadbeef;

^D
```

У цьому прикладі встановлюються секретні ключі для протоколу ESP. Параметри у виді десяткових чисел: 9876 і 10000, називаються SPI (індекс параметра безпеки). Ці значення додаються до пакетів ESP, що дозволяє приймаючій стороні вибрати відповідний секретний ключ для даного пакета. Значення SPI повинне бути більше або дорівнювати 256, а також бути унікальним для кожної асоціації безпеки.

Від 10.1.1.1 до 20.1.1.1, ми використовували 3DES-CBC алгоритм, із секретним ключем "hogehogehogehogehogehoge". Цей трафік буде ідентифікований SPI 9876.

Від 20.1.1.1 до 10.1.1.1, ми використовували 3DES-CBC алгоритм, із секретним ключем

0xdeadbeefdeadbeefdeadbeefdeadbeefdeadbeefdeadbeef.

Для видалення всіх асоціацій з SAD можна використовувати команду: setkey –F

За додатковою інформацією про використання команди setkey звернетеся до електронної довідки man: man setkey

Як видно з приклада, асоціація безпеки включає основну інформацію, необхідну для обміну ключами:

1. Напрямок трафіку (вихідний і кінцевий вузол - 10.1.1.1 і 20.1.1.1);

2. Тип протоколу (AH або ESP- аутентифікація або шифрування);

3. Конкретна реалізація протоколу (3des-cbc, blowfish, keyed-md5 …);

4.Секретний ключ.

2. Протокол IKE. Для створення асоціацій безпеки на регулярній основі, краще використовувати спеціальний комунікаційний протокол. IKE, також відомий як ISAKMP, є протоколом обміну ключами, використовуваним IPSec.

В даний час, існує два стандартних способи встановити підтримку IKE на FreeBSD системі: демони racoon і isakmpd.

Демон racoon

1. Установка racoon. У FreeBSD (і NetBSD), IKE підтримується демоном racoon. Встановіть на обох системах racoon з колекції портів (/usr/ports/security/racoon). Для цього виконаєте:

cd /usr/ports/security/racoon; make; make install; make clean

2. Функціональність racoon. Конфігураційні файли racoon за замовчуванням знаходяться в /usr/local/etc/racoon/ і називаються racoon.conf і psk.txt. Змінюючи ці файли можна настроїти racoon для виконання необхідних задач.

3. Файл racoon.conf. Порт racoon встановлює в директорію /usr/local/etc/racoon приклад конфігураційних файл

ls /usr/local/etc/raccoon psk.txt.dist racoon.conf.dist

Для створення конфігураційного файлу racoon.conf рекомендується скопіювати його з прикладу-файлу racoon.conf.dist, а потім внести в нього необхідні зміни.

cd /usr/local/etc/raccoon cp racoon.conf.dist racoon.conf

vi racoon.conf

4. Поділювані ключі. Файл psk.txt. Для процесу обміну ключами потрібно, щоб обоє учасника заздалегідь домовилися про деякий визначений секрет. Це може бути виконане за допомогою сертифікатів X.509 або так званих секретних pre-shared (поділюваних) ключів. Файл psk.txt містить адреси вузлів і, асоційовані з ними, pre-shared ключі. Наприклад: Ваш вузол 192.168.1.254. Ви настроюєте IPSec, для зв'язку з вузлом 192.168.2.254 (адреси повинні вузлів належати однієї мережі), при цьому, домовивши використовувати pre-shared ключ «thisisatest». Тоді файл psk.txt на вашому вузлі повинний включати наступний рядок

192.168.2.254 thisisatest

а на вузлі 192.168.2.254

192.168.1.254 thisisatest

Створити і відредагувати файл psk.txt можна за допомогою редактора vi:

vi /usr/local/etc/racoon/psk.txt

Варто переконатися, що файл psk.txt має права доступу 600, а власником файлу є root. У противному випадку демон racoon не буде його використовувати. Для цього виконаєте:

chown root.wheel /usr/local/etc/racoon/psk.txt

chmod 0600 /usr/local/etc/racoon/psk.txt

ls –la /usr/local/etc/racoon/psk.txt

5. Запуск racoon. Файл, що виконується, racoon знаходиться в директорії /usr/local/sbin. Його варто запускати після виконання всіх конфігураційних процедур.

Запуск racoon у звичайному (фоновому) режимі:

/usr/local/sbin/racoon -f /usr/local/etc/racoon/racoon.conf

При запуску racoon у звичайному режимі не буде виводитися ніякої службової інформації.

Запуск racoon у відладочному режимі – виводить всю інформацію на консоль:

/usr/local/sbin/racoon -f /usr/local/etc/racoon/racoon.conf -F –d

При запуску racoon у режимі протоколювання службова інформація про роботу демона буде вноситися у файл журналу (/var/log/racoon.log)

/usr/local/sbin/racoon -f /usr/local/etc/racoon/racoon.conf -l

/var/log/racoon.log && echo -n ' racoon'

Встановлення політик безпеки

1. Ми повинні повідомити ядру, для яких пакетів використовувати IPsec і як саме його застосовувати. Для цієї мети використовуються так називані політики безпеки (SP). Політика безпеки IPsec визначають, які протоколи (AH, ESP, IPCOMP) будуть використовуватися для конкретного типу пакетів. Ядро дозволяє використовувати будь-як комбінацію протоколів AH, ESP і IPCOMP. Інформація про політиків зберігається в базі даних SPD на рівні ядра.

Для настроювання політик використовується програма setkey. Наприклад:

```
# setkey –c

spdadd 1.2.3.4 5.6.7.8 any -P out ipsec

esp/transport/1.2.3.4-5.6.7.8/require;

spdadd 5.6.7.8 1.2.3.4 any -P in ipsec

esp/transport/5.6.7.8-1.2.3.4/require;

^D
```

Цей приклад демонструє установку політик безпеки на вузлі 1.2.3.4 для установки зв'язку по протоколі IPSec з вузлом 5.6.7.8. При цьому на вузлі 5.6.7.8 повинні бути встановлені такі ж політики з урахуванням напрямку пакетів (тобто параметри out і in варто поміняти місцями).

2. Для виконання різних дій за допомогою програми setkey, її можна запускати зі спеціальними опціями або виконувати серію вбудованих команд. Програма setkey зчитує серію команд із командного рядка (якщо викликано з опцією -c) або з файлу (якщо викликано з опцією -f <ім'я файлу>).

Для перегляду встановлених у SPD політик запустіть setkey з опцією –DP: setkey –DP

або виконайте вбудовану команду spddump

setkey –cspddump;^D

Для видалення встановлених у SPD політик запустіть setkey з опцією –FP: setkey –FP

або виконайте вбудовану команду spdflush

setkey –c spdflush; ^D

3. Після установки політик безпеки запису про них зберігаються в SPD (як і асоціації в SAD) тільки на час поточного сеансу системи (до перезавантаження). Тобто після перезавантаження їх доведеться встановити знову. Для автоматизації цього процесу конфігураційний файл початкового завантаження /etc/rc.conf має параметр "ipsec_enable".

ipsec_enable="YES"

Значення "YES" для цього параметра, означає, що під час завантаження системи (перед початком якої або мережевої активності) буде виконана наступна команда:

/sbin/setkey -f /etc/ipsec.conf

Тобто програма setkey виконає команди, записані у файлі /etc/ipsec.conf.

Файл /etc/ipsec.conf повинний містити команди підтримувані програмою setkey і може, приміром, виглядати в такий спосіб

Flush;

Spdflush;

Add 1.2.3.4 5.6.7.8 esp 9991 -E blowfish-cbc

"AAAABBBBZZZZZZZZZZZZZZZZZZGGGGGGGGGGGGG111111111111";

add 5.6.7.8 1.2.3.4 esp 9992 -E blowfish-cbc

"11111199999999900000000000000000ERBBBBAAAA";

spdadd 10.10.1.0/24 10.0.1.0/24 any -P out ipsec

esp/tunnel/1.2.3.4-5.6.7.8/require;

spdadd 10.0.1.0/24 10.10.1.0/24 any -P in ipsec

esp/tunnel/5.6.7.8-1.2.3.4/require;

Перші два рядки (flush; і spdflush;) очищають бази SAD і SPD. Два рядки, що починаються з “add” додають асоціації безпеки в базу SAD. Два рядки, що починаються з “spdadd” додають політики безпеки в базу SPD.

Транспортний режим IPSec. Практичні приклади.

Приклад 1. Керування ключами вручну.

Як приклад покажемо установку транспортного режиму зв'язку по протоколі IPSec між двома вузлами

локальної мережі: HOST_A (192.168.0.1) і HOST_B (192.168.0.2). Передбачається, що ядро системи підтримує протокол IPSec.

Нам буде потрібно установити асоціації безпеки (security association – SA). У даному прикладі це буде зроблено вручну за допомогою утиліти setkey .

Режими. Продемонструємо трохи ускладнений приклад: для передачі даних від HOST_A до HOST_B, буде використовуватися тільки аутентифікуючий режим old-ah, а від HOST_B до HOST_A – комбінація ah і esp.

Алгоритми. Ми повинні вибрати алгоритми, що будуть використовуватися для old-ah, ah і esp . Виберемо: keyed-md5 для old-ah, hmac-sha1 для ah, і 8 байтний des-cbc для esp.

Ключі. Вибір довжини ключа залежить від конкретного алгоритму. Наприклад, для keyed-md5 довжина ключа повинна складати 16 байт (128 біт), для hmac-sha1 – 20 байт (160 біт), і 8 байт (64 біт) для des-cbc. Виберемо відповідно "MYSECRETMYSECRET",

"KAMEKAMEKAMEKAMEKAME", "PASSWORD".

SPI. Тепер потрібно призначати SPI (Індекс Параметра Безпеки) для кожного протоколу. Зверніть увагу, що нам буде потрібно 3 SPI для цього безпечного каналу, тому що обрані три безпечних напрямки (один від HOST_A до HOST_B, і два від HOST_B до HOST_A). Значення SPI повинне бути більше або дорівнювати 256. Ми виберемо: 1000, 2000, і 3000, відповідно. Асоціації. Установимо відповідні асоціації безпеки. Виконаєте setkey на HOST_A і HOST_B:

```
# setkey –c

add 192.168.0.1 192.168.0.2 ah-old  1000 -m transport

-A keyed-md5 "MYSECRETMYSECRET";

add 192.168.0.2 192.168.0.1 ah  2000 -m transport

-A hmac-sha1 "KAMEKAMEKAMEKAMEKAME";
```

add 192.168.0.2 192.168.0.1 esp 3000 -m transport

-E des-cbc "PASSWORD"; ^D

Політики. Установимо політики безпеки на кожній з машин:

На HOST_A:

setkey –c

spdadd 192.168.0.1 192.168.0.2 any -P out ipsec

ah/transport/192.168.0.1-192.168.0.2/require ; ^D

На HOST_B:

setkey –c

spdadd 192.168.0.2 192.168.0.1 any -P out ipsec

esp/transport/192.168.0.2-192.168.0.1/require ;

spdadd 192.168.0.2 192.168.0.1 any -P out ipsec

ah/transport/192.168.0.2-192.168.0.1/require ; ^D

Тестування. Перевіримо, чи правильно встановлені асоціації і політики. Для цього на кожній з машин

виконаєте наступні команди:

- для перегляду встановлених асоціацій Setkey –D

- для перегляду встановлених політик Setkey –DP

Тепер перевіримо функціонування протоколу IPSec. У першій консолі на HOST_A запустимо програму ping, указавши як параметр IP адреса HOST_B:

Ping 192.168.0.2

Ми повинні побачити успішні ICMP-відповіді. Це означає, що зв'язок з HOST_B установлена. У противному випадку буде потрібно перевірити настроювання мережі.

Приклад 2. Автоматичне керування ключами за допомогою racoon.

У цьому прикладі, як і в прикладі №1, покажемо установку транспортного режиму зв'язку по протоколі IPSec між двома вузлами локальної мережі: HOST_A (192.168.0.1) і HOST_B (192.168.0.2). Передбачається, що ядро системи підтримує протокол IPSec, а також заздалегідь установлений демон racoon.

Файл psk.txt. Створимо і відредагуємо файл поділюваних ключів /usr/local/etc/racoon/psk.txt на кожній з машин.

На HOST_A файл psk.txt повинний містити: 192.168.0.2 thisisatest

На HOST_B: 192.168.0.1 thisisatest

Поділюваний ключ (thisisatest) може бути обраний довільно, однак, він повинний бути однаковим на обох вузлах.

Файл racoon.conf. Створіть і відредагуйте конфігураційний файл /usr/local/etc/racoon/racoon.conf на кожній

з машин.

На HOST_A файл racoon.conf повинний виглядати в такий спосіб:

```
Path include "/usr/local/etc/racoon";

# файл, де зберігаються  pre-shared ключі, необхідні для установки з'єднання:

path pre_shared_key "/usr/local/etc/racoon/psk.txt";

log debug;  # висновок відладочний інформації

padding # цю секцію залишити без змін

{

maximum_length 20;

randomize off;

strict_check off;

exclusive_tail off;

}

listen
```

```
{

#свої адреса і порт, що «слухає» демон IKE

isakmp 192.168.0.1 [500]; #для HOST_A

}

timer #цю секцію залишити без змін

{

counter 5;

interval 20 sec;

persend 1;

phase1 30 sec;

phase2 15 sec;

}

remote 192.168.0.2

{
```

```
exchange_mode aggressive,main;

doi ipsec_doi;

situation identity_only;

nonce_size 16;

lifetime time 60 min;

lifetime byte 5 MB;

initial_contact on;

support_mip6 on;

proposal_check obey;

proposal

{

encryption_algorithm blowfish;

hash_algorithm sha1;

authentication_method pre_shared_key;
```

```
dh_group 5;

}

}

sainfo 192.168.0.2

{

pfs_group 5;

lifetime time 360 min;                          #час поновлення ключів

lifetime byte 5000 KB;

encryption_algorithm blowfish;    #алгоритм шифрування

authentication_algorithm hmac_sha1;  #алгоритм аутентифікації

compression_algorithm deflate;   #алгоритм компресії

}
```

На HOST_B файл racoon.conf повинний виглядати аналогічно, за винятком рядків:

```
isakmp 192.168.0.1 [500]; #для HOST_A
```

…

remote 192.168.0.2

…

sainfo 192.168.0.2

Які, відповідно, повинні бути замінені на наступні:

isakmp 192.168.0.2 [500]; #для HOST_B

…

remote 192.168.0.1

…

sainfo 192.168.0.1

У даному прикладі як алгоритм шифрування використовується blowfish, а як алгоритм аутентифікації - hmac_sha1.

Встановлнення політик безпеки і запуск racoon. На кожній з машин створимо скрипт /tmp/ipsec.sh, у якому будемо наповняти базу даних SPD і стартувати racoon для встановлення нашого захищеного з'єднання. Для цього виконаємо команду:

vi /tmp/ipsec.sh

На HOST_A файл ipsec.sh повинний виглядати так:

```
#!/bin/sh
 /usr/sbin/setkey -FP
 /usr/sbin/setkey -F
 /usr/sbin/setkey -c << EOF
 spdadd 192.168.0.1 192.168.0.2 any -P in ipsec


esp/transport/192.168.0.1-192.168.0.2/require;
 spdadd 192.168.0.2.168.0.1 any -P out ipsec


esp/transport/10.10.10.254-192.168.0.1/require;
 EOF
 /usr/local/sbin/racoon -F -v -f /usr/local/etc/racoon/racoon.conf
```

На HOST_B файл ipsec.sh повинний виглядати аналогічно, однак, параметри «in» і «out» варто поміняти місцями.

Запустимо створений скрипт ipsec.sh на кожній з машин:

```
/tmp/ipsec.sh
```

Демон racoon повинний вивести інформацію про успішний запуск.

Тестування. Тестування захищеного каналу проведемо як у прикладі №1.

Порядок виконання роботи

1. У комп'ютерному класі реалізувати і домогтися працездатності практичних прикладів транспортного режиму IPSec.
2. На практиці, реальні IP адреси машин будуть відрізнятися від приведених у прикладах. Їх слід уточнити у викладача або адміністратора.

3. Змінити роботу протоколу IPSec відповідно до індивідуального завдання викладача. У звіті відзначити про зміни внесених у конфігураційні файли або виконувані команди.
4. Знищити усі внесені в систему зміни: видалити конфігураційні файли і скрипти, очистити бази SAD і SPD.

Контрольні запитання та завдання:

1. Для чого використовується протокол IPSec?
2. Які функції безпеки надає протокол IPSec. Які його компоненти за це відповідають?
3. У чому відмінність між транспортним і тунельним режимом IPSec? Для чого вони використовуються?
4. Що таке Асоціації безпеки і Політики безпеки в контексті IPSec?
5. Які існують способи установки Асоціацій безпеки й обміну ключами?

Список літератури.

1. *Павлов В.Г., Михальчук І.І.* Структурна організація та архітектура комп'ютерних систем: Конспект лекцій. – К.: НАУ, 2010, 64 с.
2. *Корнієнко Б.Я., Фомін М.М., Щербак Л.М.* Захист інформації в комп`ютерних системах та мережах (модульні технології навчання). Навчально-методичне видання, Київ: НАУ. – 2004, 107 с.
3. *Корнієнко Б.Я., Щербак Л.М.* Захист інформації в комп'ютерних системах та мережах, частина 2 (модульні технології навчання). Навчально-методичне видання, Київ: НАУ. – 2005, 139 с.
4. *Анин Б.Ю.* Защита компьютерной информации. / Анин Б.Ю.– СПб.: BHV, 2000. – 384 с.
1. *Алферов А.П.* Основы криптографии / Алферов А.П., Зубов А.Ю., Кузьмин А.С. и др. – М.: Гелиос АРВ , 2003. – 480 с
2. *Касперски К.* Техника и философия хакерских атак. / Касперски К. – М.: Солон, 2001. - 272 с.
3. *Макнамара Дж.* Секреты компьютерного шпионажа. Тактика и контрмеры. / Макнамара Дж. – М.: Бином, 2006, – 536 с.
4. *Стенг Д.* Секреты безопасности сетей. / Стенг Д., Мун С – К.: Диалектика, 1995, – 544 с.

Додаток 1

Створення завантажувальної дискети 3,5"

1. Для створення дискети для завантаження операційної системи MS-DOS необхідний комп'ютер, де ця операційна система вже завантажена.
2. Далі необхідно виконати утиліту FORMAT з наступними опціями:

FORMAT A: /S /U

- опція S указує на створення завантажувального носія;

- опція U відключає попередню перевірку формату дискети.

1. Якщо дискета створюється з середовища WINDOWS, то можна використовувати командний рядок, куди записати вказану вище команду, або вибрати пункт "Форматувати" в контекстному меню відповідного логічного диска (A). У останньому випадку потрібно у вікні форматування поставити позначку у пункті "*Створення завантажувального диска MS-DOS*".

2. Нарешті, якщо є в наявності зразок завантажувальної дискети, то його можна "клонувати" за допомогою образу або команди "diskcopy".

Додаток 2

Створення завантажувального пристрою FLASH-памяти.

Використовується безкоштовна програма "USB Disk Storage Format Tool", розроблена фірмою *Hewlett-Packard*. Вона дозволяє в середовищі XP/Vista/Windows 7 створити на будь-якому Flash-накопичувачі завантажувальний розділ з підтримкою FAT32, що дає можливість працювати з довгими іменами файлів. Єдино, що потрібно – мати системні файли тієї операційної системи, яка завантажуватиметься за допомогою даного Flash-накопичувача.

Дана програма може бути вільно завантажена з сайту http://www8.hp.com/ru/ru/home.html і проинсталлирована в теку DriveKey. У її складі всього дві утиліти:

- HPUSBF.EXE – для роботи в консольному режимі;

- HPUSBFW.EXE – для створення завантажувального Flash-накопичувача із середовища WINDOWS.

Все інше просто, логічно і доступно для розуміння, тому не потребує детального пояснення.

Навчальне видання

Булана Людмила Вікторівна

Галата Лілія Павлівна

Корнієнко Богдан Ярославович

Павлов Валерій Георгійович

БЕЗПЕКА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

Лабораторний практикум

Технічний редактор

Коректор

Підп. до друку __.__.11 Формат Папір офс.

Офс. Друк. Ум. друк. арк. 5,0 Обл. – вид. арк.

Тираж 100 пр. Замовлення № Вид. №

Видавництво НАУ

03058, Київ-58, проспект Космонавта Комарова, 1

Свідоцтво про внесення до Державного реєстру ДК

№ від __.__.____