ЭКЗАМЕНАЦИОННАЯ РАБОТА

«Безопасность компьютерных сетей»

УТИЛИТА ДЛЯ СКАНИРОВАНИЯ БЕЗОПАСНОСТИ СЕТИ NMAP

Содержание

1. Введение
2. КАК РАБОТАЕТ NMAP?
   1. СИНТАКСИС NMAP
   2. КАК ПОЛЬЗОВАТЬСЯ NMAP ДЛЯ СКАНИРОВАНИЯ ПОРТОВ
3. Системы обнаружения вторжения
4. Антируткит
5. Сбор информации до и после взлома
6. ЗАКЛЮЧЕНИЕ

Введение

Две программы, играющие важную роль в анализе сети – это программа nmap, которая сканирует порты удаленного хоста, и программа tcpdump, которая переводит сетевую плату в режим прослушивания, что позволяет принимать все пакеты независимо от того, кому они адресованы.

Nmap – это самый популярный сетевой сканер. Он используется для определения, какие хосты доступны в сети, какие сервисы они выполняют, какие версии операционных систем используют и т.д. Программа Nmap или Network Mapper была разработана Гордоном Луоном и на данный момент используется специалистами по безопасности и системными администраторами по всему миру.

Tcpdump – это утилита, позволяющая перехватывать и анализировать сетевой трафик, проходящий через сетевые интерфейсы. Иногда в целях выявления неисправностей нужно выяснить, какие пакеты передаются по сети. Это несложно сделать с помощью tcpdump.

Я предлагаю, рассмотреть программу Nmap. Эта программа помогает системным администраторам очень быстро понять какие компьютеры подключены к сети, узнать их имена, а также посмотреть какое программное обеспечение на них установлено, какая операционная система и какие типы фильтров применяются. Функциональность программы может быть расширена за счет собственного скриптового языка, который позволяет администраторам автоматизировать много действий.

Например, с помощью скриптов можно автоматически обнаруживать новые уязвимости безопасности в вашей сети. Namp может использоваться с хорошими и плохими намерениями. Мы рассмотрим как пользоваться namp для сканирования портов в операционной системе Linux. Но сначала нужно попытаться понять как работает эта утилита.

Nmap обладает множеством возможностей, о которых можно прочитать в man-странице.

Приведем несколько примеров использования nmap, но сначала давайте разберем возможные состояния портов, которые вы можете видеть при их сканировании.

В Nmap есть 6 возможных состояний для порта.

1. Open - Порт открыт, и служба ожидает запросы на подключение.
2. Closed - Закрытый порт не связан ни с одним приложением и может быть открыт в любой момент.
3. Filtered - Брандмауэр блокирует порт, и nmap не может определить, закрыт порт или открыт.
4. Unfiltered - Порты в этом состоянии отвечают на запросы, но nmap не может определить, закрыты они или открыты.
5. open|filtered - Nmap выдает такой результат, когда не может определить, открыт ли порт или он фильтруется.
6. closed|filtered - Nmap выдает такой результат, когда не может определить, закрыт ли порт или он фильтруется

Nmap можно установить из стандартного репозитория.

**[root@localhost dato]# yum -y install nmap**

КАК РАБОТАЕТ NMAP?

В компьютерных сетях все подключенные устройства имеют свой ip адрес. Каждый компьютер поддерживает протокол ping, с помощью которого можно определить подключен ли он к сети. Мы просто отправляем ping запрос компьютеру, и если он отзывается, то считаем, что он подключен. Nmap использует немного иной подход. Компьютеры также определенным образом реагируют на те или иные сетевые пакеты, утилита просто отправляет нужные пакеты и смотрит какие хосты прислали ответ.

Более интересно то как Nmap узнает какие сервисы запущены на машине. Суть работы всех сетевых программ основана на портах. Чтобы получить сообщение из сети, программа должна открыть порт на вашем компьютере и ждать входящих соединений. А для отправки сообщения по сети нужно подключиться к уже другой программой (адресатом) порту. Затем программе необходимо будет открыть порт, на котором она будет ждать ответа.

Утилита nmap в процессе сканирования сети перебирает доступный диапазон портов и пытается подключиться к каждому из них. Если подключение удалось, в большинстве случаев, передав несколько пакетов программа может даже узнать версию программного обеспечения, которые ожидает подключений к этому порту. Теперь, после того, как мы рассмотрели основы, рассмотрим как пользоваться nmap для сканирования портов и сети.

СИНТАКСИС NMAP

Команда запуска Nmap очень проста для этого достаточно передать ей в параметрах целевой IP адрес или сеть, а также указать опции при необходимости:

**$ nmap опции адрес**

Теперь давайте рассмотрим основные опции:

1. **sL** — просто создать список работающих хостов, но не сканировать порты nmap;
2. **sP** — только проверять доступен ли хост с помощью ping;
3. **PN** — считать все хосты доступными, даже если они не отвечают на ping;
4. **sS/sT/sA/sW/sM** — TCP сканирование;
5. **sU** — UDP сканирование nmap;
6. **sN/sF/sX** — TCP NULL и FIN сканирование;
7. **sC** — запускать скрипт по умолчанию;
8. **sI** — ленивое Indle сканирование;
9. **p** — указать диапазон портов для проверки;
10. **sV** — детальное исследование портов для определения версий служб;
11. **O** — определять операционную систему;
12. **T[0-5]** — скорость сканирования, чем больше, тем быстрее;
13. **D** — маскировать сканирование с помощью фиктивных IP;
14. **S** — изменить свой IP адрес на указанный;
15. **e** — использовать определенный интерфейс;
16. **spoof-mac** — установить свой MAC адрес;
17. **A** — определение операционной системы с помощью скриптов.

Теперь, когда мы рассмотрели все основные опции, давайте поговорим о том, как выполняется сканирование портов nmap.

КАК ПОЛЬЗОВАТЬСЯ NMAP ДЛЯ СКАНИРОВАНИЯ ПОРТОВ

Сначала давайте рассмотрим как найти все подключенные к сети устройства, для этого достаточно использовать опцию -sL и указать маску нашей сети. в моем случае это 192.168.0.105/24. Маску нашей локальной сети можно найти, выполнив команду:

**[root@localhost dato]# ip addr show**

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
       valid\_lft forever preferred\_lft forever  
    inet6 ::1/128 scope host   
       valid\_lft forever preferred\_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000  
    link/ether 08:00:27:d4:b1:c3 brd ff:ff:ff:ff:ff:ff  
    inet **192.168.0.105/24** brd 192.168.0.255 scope global dynamic enp0s3  
       valid\_lft 2516sec preferred\_lft 2516sec  
    inet6 fe80::ea38:7ca0:47a8:e1c5/64 scope link   
       valid\_lft forever preferred\_lft forever  
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000  
    link/ether 52:54:00:a5:ce:ae brd ff:ff:ff:ff:ff:ff  
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0  
       valid\_lft forever preferred\_lft forever  
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo\_fast master virbr0 state DOWN qlen 1000  
    link/ether 52:54:00:a5:ce:ae brd ff:ff:ff:ff:ff:ff

Команда на сканирование сети nmap будет выглядеть вот так:

**[root@localhost dato]# nmap -sL 192.168.0.105/24**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/) ) at 2018-04-24 22:24 +08  
Nmap scan report for 192.168.0.0  
Nmap scan report for 192.168.0.1  
Nmap scan report for 192.168.0.2  
Nmap scan report for 192.168.0.3  
Nmap scan report for 192.168.0.4  
…  
Nmap scan report for 192.168.0.254  
Nmap scan report for 192.168.0.255  
Nmap done: 256 IP addresses (0 hosts up) scanned in 0.35 seconds

Иногда это сканирование может не дать никаких результатов, потому что некоторые операционные системы имеют защиту от сканирования портов. Но это можно обойти, просто использовав для сканирования ping всех ip адресов сети, для этого есть опция -sn:

**[root@localhost dato]# nmap -sn 192.168.0.105/24**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:25 +08  
Nmap scan report for **192.168.0.1**  
Host is up (0.014s latency).  
MAC Address: C4:6E:1F:E2:B9:E8 (Unknown)  
Nmap scan report for **192.168.0.100**  
Host is up (0.034s latency).  
MAC Address: 84:BE:52:63:EC:AD (Unknown)  
Nmap scan report for **192.168.0.101**  
Host is up (0.010s latency).  
MAC Address: AC:AF:B9:2C:89:32 (Unknown)  
Nmap scan report for **192.168.0.103**  
Host is up (0.00026s latency).  
MAC Address: C4:46:19:23:7E:31 (Hon Hai Precision Ind. Co.)  
Nmap scan report for **192.168.0.105**  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.86 seconds

Как видите, теперь программа обнаружила активные устройства в сети. Дальше мы можем сканировать порты nmap для нужного узла запустив утилиту без опций:

**[root@localhost dato]# nmap 192.168.0.1**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:26 +08  
Nmap scan report for **192.168.0.1**  
Host is up (0.016s latency).  
Not shown: 998 closed ports  
PORT STATE SERVICE  
**80/tcp open http  
1900/tcp open upnp**  
MAC Address: C4:6E:1F:E2:B9:E8 (Unknown)  
Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds

Теперь мы можем видеть, что у нас открыто несколько портов, все они используются каким-либо сервисом на целевой машине. Каждый из них может быть потенциально уязвимым, поэтому иметь много открытых портов на машине небезопасно.

Чтобы узнать более подробную информацию о машине и запущенных на ней сервисах вы можете использовать опцию -sV. Утилита подключится к каждому порту и определит всю доступную информацию:

**[root@localhost dato]# nmap -sV 192.168.0.1**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:27 +08  
Nmap scan report for 192.168.0.1  
Host is up (0.020s latency).  
Not shown: 998 closed ports  
PORT STATE SERVICE VERSION  
**80/tcp open http TP-LINK WR740N WAP http config  
1900/tcp open upnp ipOS 7.0 (TP-LINK TL-WR740N WAP 4.0; UPnP 1.0)**MAC Address: C4:6E:1F:E2:B9:E8 (Unknown)  
Service Info: Device: WAP

Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 7.35 seconds

**[root@localhost dato]# nmap -sV 192.168.0.105**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:28 +08  
Nmap scan report for 192.168.0.105  
Host is up (0.000021s latency).  
Not shown: 998 closed ports  
PORT STATE SERVICE VERSION  
**22/tcp open ssh OpenSSH 7.4 (protocol 2.0)  
111/tcp open rpcbind 2-4 (RPC #100000)**  
Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 6.42 seconds

На нашей машине запущен ssh, а поэтому мы можем попытаться рассмотреть эту службу подробнее с помощью стандартных скриптов nmap. Скрипты позволяют проверить порт более детально, найти возможные уязвимости. Для этого используйте опцию -sC и -p чтобы задать порт:

**[root@localhost dato]# nmap -sC 192.168.0.105 -p 22**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:30 +08  
Nmap scan report for 192.168.0.105  
Host is up (2100s latency).  
PORT STATE SERVICE  
**22/tcp open ssh  
| ssh-hostkey: 2048 3b:83:bd:9d:80:13:19:e0:ad:96:ce:62:a8:66:df:c3 (RSA)  
|\_256 bf:e8:e3:3d:da:7a:ae:62:68:67:01:ca:a7:28:aa:0a (ECDSA)**  
Nmap done: 1 IP address (1 host up) scanned in 0.94 seconds

Мы выполняли скрипт по умолчанию, но есть еще и другие скрипты, например, найти все скрипты для ssh можно командой:

**[root@localhost dato]# find /usr/share/nmap/scripts/ -name '\*.nse' | grep ssh**

/usr/share/nmap/scripts/**ssh-hostkey.nse**  
/usr/share/nmap/scripts/**ssh2-enum-algos.nse**  
/usr/share/nmap/scripts/**sshv1.nse**

Затем попытаемся использовать один из них, для этого достаточно указать его с помощью опции —script. Но сначала вы можете посмотреть информацию о скрипте:

**[root@localhost dato]# nmap --script-help ssh-hostkey.nse**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:34 +08  
ssh-hostkey  
Categories: safe default discovery  
<http://nmap.org/nsedoc/scripts/ssh-hostkey.html>  
  Shows SSH hostkeys.  
  Shows the target SSH server's key fingerprint and (with high enough verbosity level) the public key itself. It records the discovered host keys in <code>nmap.registry</code> for use by other scripts. Output can be controlled with the <code>ssh\_hostkey</code> script argument.  
  The script also includes a postrule that check for duplicate hosts using the gathered keys.

Cкрипт на рисунке 1 будет пытаться определить логин и пароль от FTP на удаленном узле.

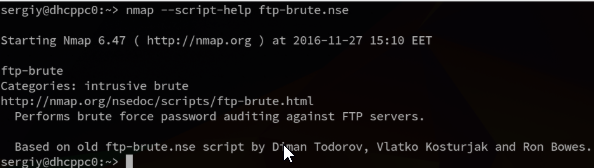
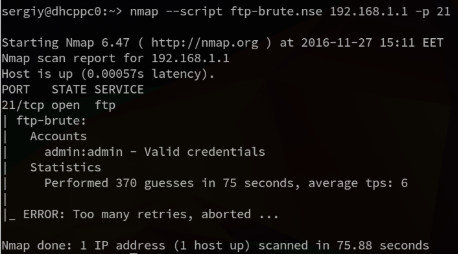


Рис 1. Информация о скрипте FTP

Затем выполняем скрипт:

**[root@localhost dato]# nmap --script ssh-hostkey.nse 192.168.0.105 -p 22**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:36 +08  
Nmap scan report for 192.168.0.105  
Host is up (920s latency).  
PORT STATE SERVICE  
**22/tcp open ssh  
| ssh-hostkey: 2048 3b:83:bd:9d:80:13:19:e0:ad:96:ce:62:a8:66:df:c3 (RSA)  
|\_256 bf:e8:e3:3d:da:7a:ae:62:68:67:01:ca:a7:28:aa:0a (ECDSA)**  
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds



В результате скрипт подобрал логин и пароль, admin/admin. Вот поэтому не нужно использовать параметры входа по умолчанию.

Также можно запустить утилиту с опцией -A, она активирует более агрессивный режим работы утилиты, с помощью которого вы получите большую часть информации одной командой: Опция -A используется для того, чтобы выяснить тип операционной системы и ее версию.

**[root@localhost dato]# nmap -A 192.168.0.105**

Starting Nmap 6.40 **(** [**http://nmap.org**](http://nmap.org/) **)** at 2018-04-24 22:38 +08  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0  
Nmap scan report for 192.168.0.105  
Host is up (0.000085s latency).  
Not shown: 998 closed ports  
PORT STATE SERVICE VERSION  
22/tcp open ssh OpenSSH 7.4 (protocol 2.0)  
| ssh-hostkey: 2048 3b:83:bd:9d:80:13:19:e0:ad:96:ce:62:a8:66:df:c3 (RSA)  
|\_256 bf:e8:e3:3d:da:7a:ae:62:68:67:01:ca:a7:28:aa:0a (ECDSA)  
111/tcp open rpcbind 2-4 (RPC #100000)  
| rpcinfo:   
| program version port/proto service  
| 100000 2,3,4 111/tcp rpcbind  
|\_ 100000 2,3,4 111/udp rpcbind  
Device type: general purpose  
Running: Linux 3.X  
OS CPE: cpe:/o:linux:linux\_kernel:3  
OS details: Linux 3.7 - 3.9  
Network Distance: 0 hops  
OS and Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 9.34 seconds

Здесь есть почти вся информация, которую мы уже видели раньше. Ее можно использовать чтобы увеличить защиту этой машины.

Опция -T4 для более быстрого выполнения сканирования, так называемый агрессивный режим.

**[root@localhost dato]# nmap -A -T4 192.168.0.105**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:07 +08  
Nmap scan report for 192.168.0.105  
Host is up (0.000061s latency).  
Not shown: 998 closed ports  
PORT STATE SERVICE VERSION  
22/tcp open ssh OpenSSH 7.4 (protocol 2.0)  
| ssh-hostkey: 2048 3b:83:bd:9d:80:13:19:e0:ad:96:ce:62:a8:66:df:c3 (RSA)  
|\_256 bf:e8:e3:3d:da:7a:ae:62:68:67:01:ca:a7:28:aa:0a (ECDSA)  
111/tcp open rpcbind 2-4 (RPC #100000)  
| rpcinfo:   
| program version port/proto service  
| 100000 2,3,4 111/tcp rpcbind  
|\_ 100000 2,3,4 111/udp rpcbind  
Device type: general purpose  
Running: Linux 3.X  
OS CPE: cpe:/o:linux:linux\_kernel:3  
OS details: Linux 3.7 - 3.9  
Network Distance: 0 hops  
OS and Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .  
Nmap done: 1 IP address (1 host up) scanned in 9.35 seconds

Вывод команды будет содержать список открытых портов в удаленной системе, MAC-адрес, тип операционной системы и возможную версию ядра.

Сканируем диапазон портов. Опция -sT – это TCP сканирование с использованием системного вызова connect, опция -p позволяет задать диапазон портов.

**[root@localhost dato]# nmap -sT -p 21-80 192.168.0.105**

Starting Nmap 6.40 ( [http://nmap.org](http://nmap.org/" \t "_blank) ) at 2018-04-24 22:16 +08  
Nmap scan report for 192.168.0.105  
Host is up (0.0020s latency).  
Not shown: 59 closed ports  
PORT STATE SERVICE  
22/tcp open ssh  
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds

## Системы обнаружения вторжения

Системы обнаружения вторжения (Intrusion Detection System, IDS) – это работающие процессы или устройства, анализирующие активность в сети или системе на предмет неавторизованных и злонамеренных действий.

Наиболее распространенными IDS являются так называемые локальные и сетевые системы IDS.

Локальные IDS устанавливаются в каждую систему и занимаются контролем целостности системы изнутри. Такие IDS проверяют различные журналы и сравнивают их с внутренней базой данных стандартных сигнатур известных атак. Популярными локальными IDS являются программные пакеты:

1. Tripwire
2. Swatch
3. Lids

Tripwire является свободным программным обеспечением безопасности и целостности данных инструмент полезен для мониторинга и оповещения о конкретных изменениях файлов в системе. Tripwire с открытым исходным кодом функционирует как хост-система обнаружения вторжений.

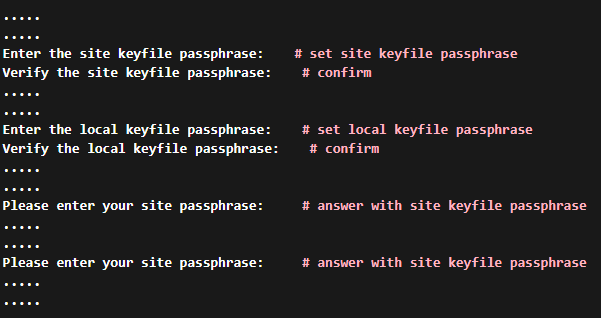
Устанавливаем Tripwire:

**[root@localhost david]# yum -y install epel-release**

**[root@localhost david]# yum -y enablerepo=epel -y install tripwire**

Создание системных файлов криптографических ключей (Рисунок 2):

**[root@localhost david]# tripwire-setup-keyfiles**



**[root@localhost david]# cd /etc/tripwire**

Открываем файл twcfg.txt для редактирования и изменения по мере необходимости. Содержание нашего файла перечислено ниже:

**[root@localhost tripwire]# gedit twcfg.txt**



Создаем конфигурацию:

**[root@localhost tripwire]# twadmin -m F -c tw.cfg -S site.key twcfg.txt**

Please enter your site passphrase:   
Wrote configuration file: /etc/tripwire/tw.cfg

Оптимизируем файл политики с помощью скрипта ниже:

**[root@localhost tripwire]# gedit twpolmake.pl**

#!/usr/bin/perl

# Tripwire Policy File customize tool

# ----------------------------------------------------------------

# Copyright (C) 2003 Hiroaki Izumi

# This program is free software; you can redistribute it and/or

# modify it under the terms of the GNU General Public License

# as published by the Free Software Foundation; either version 2

# of the License, or (at your option) any later version.

# This program is distributed in the hope that it will be useful,

# but WITHOUT ANY WARRANTY; without even the implied warranty of

# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

# GNU General Public License for more details.

# You should have received a copy of the GNU General Public License

# along with this program; if not, write to the Free Software

# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

# ----------------------------------------------------------------

# Usage:

# perl twpolmake.pl {Pol file}

# ----------------------------------------------------------------

#

$POLFILE=$ARGV[0];

open(POL,"$POLFILE") or die "open error: $POLFILE" ;

my($myhost,$thost) ;

my($sharp,$tpath,$cond) ;

my($INRULE) = 0 ;

while (<POL>) {

chomp;

if (($thost) = /^HOSTNAME\s\*=\s\*(.\*)\s\*;/) {

$myhost = `hostname` ; chomp($myhost) ;

if ($thost ne $myhost) {

$\_="HOSTNAME=\"$myhost\";" ;

}

}

elsif ( /^{/ ) {

$INRULE=1 ;

}

elsif ( /^}/ ) {

$INRULE=0 ;

}

elsif ($INRULE == 1 and ($sharp,$tpath,$cond) = /^(\s\*\#?\s\*)(\/\S+)\b(\s+->\s+.+)$/) {

$ret = ($sharp =~ s/\#//g) ;

if ($tpath eq '/sbin/e2fsadm' ) {

$cond =~ s/;\s+(tune2fs.\*)$/; \#$1/ ;

}

if (! -s $tpath) {

$\_ = "$sharp#$tpath$cond" if ($ret == 0) ;

}

else {

$\_ = "$sharp$tpath$cond" ;

}

}

print "$\_\n" ;

}

close(POL) ;

**[root@localhost tripwire]# perl twpolmake.pl twpol.txt > twpol.txt.new   
[root@localhost tripwire]# twadmin -m P -c tw.cfg -p tw.pol -S site.key twpol.txt.new**

Please enter your site passphrase:   
Wrote policy file: /etc/tripwire/tw.pol

Создаем базу данных:

**[root@localhost tripwire]# tripwire -m i -s -c tw.cfg**

Please enter your local passphrase:

**[root@localhost tripwire]# tripwire -m c -s -c /etc/tripwire/tw.cfg**

Open Source Tripwire(R) 2.4.3.5 Integrity Check Report  
Report generated by: root  
Report created on: Чт 26 апр 2018 22:18:14  
Database last updated on: Never  
=====================================================================  
Report Summary:  
=====================================================================Host name: localhost.localdomain  
Host IP address: 127.0.0.1  
Host ID: None  
Policy file used: /etc/tripwire/tw.pol  
Configuration file used: /etc/tripwire/tw.cfg  
Database file used: /var/lib/tripwire/localhost.localdomain.twd  
Command line used: tripwire -m c -s -c /etc/tripwire/tw.cfg   
=====================================================================  
Rule Summary:   
=====================================================================  
  Section: Unix File System  
-------------------------------------------------------------------------------  
  Rule Name Severity Level Added Removed Modified   
  --------- -------------- ----- ------- --------   
  User binaries 66 0 0 0   
  Tripwire Binaries 100 0 0 0   
  Libraries 66 0 0 0   
  Operating System Utilities 100 0 0 0   
  File System and Disk Administraton Programs 100 0 0 0   
  Kernel Administration Programs 100 0 0 0   
  Networking Programs 100 0 0 0   
  System Administration Programs 100 0 0 0   
  Hardware and Device Control Programs 100 0 0 0   
  System Information Programs 100 0 0 0   
  Application Information Programs 100 0 0 0   
  (/sbin/rtmon)  
  Critical Utility Sym-Links 100 0 0 0   
  Shell Binaries 100 0 0 0   
  Critical system boot files 100 0 0 0   
\* **Tripwire Data Files 100 1 0 0**  System boot changes 100 0 0 0   
  OS executables and libraries 100 0 0 0   
  Critical configuration files 100 0 0 0   
  Security Control 100 0 0 0   
  Login Scripts 100 0 0 0   
  Root config files 100 0 0 0   
  Invariant Directories 66 0 0 0   
  Temporary directories 33 0 0 0   
  Critical devices 100 0 0 0   
  (/proc/kcore)  
Total objects scanned: 29427  
Total violations found: 1  
=====================================================================  
Object Summary:   
=====================================================================  
# Section: Unix File System  
-------------------------------------------------------------------------------  
Rule Name: Tripwire Data Files (/var/lib/tripwire)  
Severity Level: 100  
-------------------------------------------------------------------------------  
Added:  
"/var/lib/tripwire/localhost.localdomain.twd"  
=====================================================================  
Error Report:   
=====================================================================  
No Errors  
-------------------------------------------------------------------------------  
\*\*\* End of report \*\*\*

Open Source Tripwire 2.4 Portions copyright 2000-2017 Tripwire, Inc. Tripwire is a registeredtrademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY; for details use --version. This is free software which may be redistributed  
or modified only under certain conditions; see COPYING for details.

All rights reserved.

Результаты сохраняются в каталоге ниже:

**[root@localhost tripwire]# ll /var/lib/tripwire/report**

итого 8  
-rw-r--r--. 1 root root 7622 апр 26 22:20 localhost.localdomain-20180426-221814.twr

## Антируткит

Программа chkrootkit предназначена для поиска враждебного кода (rootkit) и иных подозрительных событий в системе

Установим gcc

**[root@localhost david]# yum install gcc**

Скачиваем последнюю на текущий момент версию с официального ftp:

[root@localhost david]# wget <ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz>

Распаковываем:

**[root@localhost david]# tar -xzf chkrootkit.tar.gz**

Переходим в директорию:

**[root@localhost david]# cd chkrootkit-0.52**

Собираем модули:

**[root@localhost chkrootkit-0.52]# make sense**

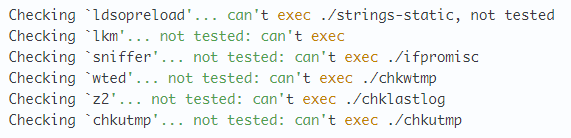
cc -DHAVE\_LASTLOG\_H -o chklastlog chklastlog.c  
cc -DHAVE\_LASTLOG\_H -o chkwtmp chkwtmp.c  
cc -DHAVE\_LASTLOG\_H -D\_FILE\_OFFSET\_BITS=64 -o ifpromisc ifpromisc.c  
cc -o chkproc chkproc.c  
cc -o chkdirs chkdirs.c  
cc -o check\_wtmpx check\_wtmpx.c  
cc -static -o strings-static strings.c  
/usr/bin/ld: cannot find -lc  
collect2: ошибка: выполнение ld завершилось с кодом возврата 1  
make: \*\*\* [strings-static] Ошибка 1

Запускаем шелл-скрипт:

**[root@localhost chkrootkit-0.52]# ./chkrootkit**

ROOTDIR is `/'  
Checking `amd'... not found  
Checking `basename'... not infected  
Checking `biff'... not found  
Checking `chfn'... not infected  
Checking `chsh'... not infected  
Checking `cron'... not infected  
Checking `crontab'... not infected  
Checking `date'... not infected  
Checking `du'... not infected  
Checking `dirname'... not infected  
Checking `echo'... not infected  
Checking `egrep'... not infected  
Checking `env'... not infected  
Checking `find'... not infected  
Checking `fingerd'... not found  
Checking `gpm'... not found  
Checking `grep'... not infected  
Checking `hdparm'... not found  
Checking `su'... not infected  
Checking `ifconfig'... not infected  
Checking `inetd'... not tested  
Checking `inetdconf'... not found  
Checking `identd'... not found  
Checking `init'... not infected  
Checking `killall'... not infected  
Checking `ldsopreload'... can't exec ./strings-static, not tested  
Checking `login'... not infected  
Checking `ls'... not infected  
Checking `lsof'... not infected  
Checking `mail'... not infected  
Checking `mingetty'... not found  
Checking `netstat'... not infected  
Checking `named'... not found  
Checking `passwd'... not infected  
Checking `pidof'... not infected  
Checking `pop2'... not found  
Checking `pop3'... not found  
Checking `ps'... not infected  
Checking `pstree'... not infected  
Checking `rpcinfo'... not infected  
Checking `rlogind'... not found  
Checking `rshd'... not found  
Checking `slogin'... not infected  
Checking `sendmail'... not infected  
Checking `sshd'... not infected  
Checking `syslogd'... not tested  
Checking `tar'... not infected  
Checking `tcpd'... not infected  
Checking `tcpdump'... not infected  
Checking `top'... not infected  
Checking `telnetd'... not found  
Checking `timed'... not found  
Checking `traceroute'... not infected  
Checking `vdir'... not infected  
Checking `w'... not infected  
Checking `write'... not infected  
Checking `aliens'... no suspect files  
Searching for sniffer's logs, it may take a while... nothing found  
Searching for HiDrootkit's default dir... nothing found  
Searching for t0rn's default files and dirs... nothing found  
Searching for t0rn's v8 defaults... nothing found  
Searching for Lion Worm default files and dirs... nothing found  
Searching for RSHA's default files and dir... nothing found  
Searching for RH-Sharpe's default files... nothing found  
Searching for Ambient's rootkit (ark) default files and dirs... nothing found  
Searching for suspicious files and dirs, it may take a while...   
/usr/lib/debug/usr/.dwz  
Searching for LPD Worm files and dirs... nothing found  
Searching for Ramen Worm files and dirs... nothing found  
Searching for Maniac files and dirs... nothing found  
Searching for RK17 files and dirs... nothing found  
Searching for Ducoci rootkit... nothing found  
Searching for Adore Worm... nothing found  
Searching for ShitC Worm... nothing found  
Searching for Omega Worm... nothing found  
Searching for Sadmind/IIS Worm... nothing found  
Searching for MonKit... nothing found  
Searching for Showtee... nothing found  
Searching for OpticKit... nothing found  
Searching for T.R.K... nothing found  
Searching for Mithra... nothing found  
Searching for LOC rootkit... nothing found  
Searching for Romanian rootkit... nothing found  
Searching for HKRK rootkit... nothing found  
Searching for Suckit rootkit... nothing found  
Searching for Volc rootkit... nothing found  
Searching for Gold2 rootkit... nothing found  
Searching for TC2 Worm default files and dirs... nothing found  
Searching for Anonoying rootkit default files and dirs... nothing found  
Searching for ZK rootkit default files and dirs... nothing found  
Searching for ShKit rootkit default files and dirs... nothing found  
Searching for AjaKit rootkit default files and dirs... nothing found  
Searching for zaRwT rootkit default files and dirs... nothing found  
Searching for Madalin rootkit default files... nothing found  
Searching for Fu rootkit default files... nothing found  
Searching for ESRK rootkit default files... nothing found  
Searching for rootedoor... nothing found  
Searching for ENYELKM rootkit default files... nothing found  
Searching for common ssh-scanners default files... nothing found  
Searching for Linux/Ebury - Operation Windigo ssh... not tested  
Searching for 64-bit Linux Rootkit ... nothing found  
Searching for 64-bit Linux Rootkit modules... nothing found  
Searching for Mumblehard Linux ... nothing found  
Searching for Backdoor.Linux.Mokes.a ... nothing found  
Searching for Malicious TinyDNS ... nothing found  
Searching for Linux.Xor.DDoS ... INFECTED: Possible Malicious Linux.Xor.DDoS installed  
/tmp/ks-script-Bg5fkS  
Searching for Linux.Proxy.1.0 ... nothing found  
Searching for suspect PHP files... nothing found  
Searching for anomalies in shell history files... nothing found  
Checking `asp'... not infected  
Checking `bindshell'... not infected  
Checking `lkm'... You have 1 process hidden for readdir command  
You have 1 process hidden for ps command  
chkproc: Warning: Possible LKM Trojan installed  
chkdirs: nothing detected  
Checking `rexedcs'... not found  
Checking `sniffer'... enp0s3: PF\_PACKET(/usr/sbin/dhclient)  
virbr0: not promisc and no PF\_PACKET sockets  
Checking `w55808'... not infected  
Checking `wted'... chkwtmp: nothing deleted  
Checking `scalper'... not infected  
Checking `slapper'... not infected  
Checking `z2'... chklastlog: nothing deleted  
Checking `chkutmp'... not tested: can't exec ./chkutmp  
Checking `OSX\_RSPLUG'... not tested

В принципе, можно не собирать модули, и тогда на проверках, где требуются дополнительные модули, будут ошибки типа:



По умолчанию происходит полная проверка системы по всем доступным тестам, список которых можно получить, выполнив скрипт с ключом ‘-l’. К сожалению, мне не удалось найти в документации описание всех тестов, которые могут быть запущены, но это можно обсудить в комментариях.

Обратите внимание, что chrootkit не делает снимок текущих файлов и не сохраняет в своей базе, как тот же rkunter.

В качестве ключей запуска можно указывать какие из тестов запускать

К примеру, произведет поиск вредоносного кода и запущенных перехватчиков трафика:

**[root@localhost chkrootkit-0.52]# ./chkrootkit aliens sniffer**

ROOTDIR is `/'  
Checking `aliens'... no suspect files  
Searching for sniffer's logs, it may take a while... nothing found  
Searching for HiDrootkit's default dir... nothing found  
Searching for t0rn's default files and dirs... nothing found  
Searching for t0rn's v8 defaults... nothing found  
Searching for Lion Worm default files and dirs... nothing found  
Searching for RSHA's default files and dir... nothing found  
Searching for RH-Sharpe's default files... nothing found  
Searching for Ambient's rootkit (ark) default files and dirs... nothing found  
Searching for suspicious files and dirs, it may take a while...   
/usr/lib/debug/usr/.dwz  
Searching for LPD Worm files and dirs... nothing found  
Searching for Ramen Worm files and dirs... nothing found  
Searching for Maniac files and dirs... nothing found  
Searching for RK17 files and dirs... nothing found  
Searching for Ducoci rootkit... nothing found  
Searching for Adore Worm... nothing found  
Searching for ShitC Worm... nothing found  
Searching for Omega Worm... nothing found  
Searching for Sadmind/IIS Worm... nothing found  
Searching for MonKit... nothing found  
Searching for Showtee... nothing found  
Searching for OpticKit... nothing found  
Searching for T.R.K... nothing found  
Searching for Mithra... nothing found  
Searching for LOC rootkit... nothing found  
Searching for Romanian rootkit... nothing found  
Searching for HKRK rootkit... nothing found  
Searching for Suckit rootkit... nothing found  
Searching for Volc rootkit... nothing found  
Searching for Gold2 rootkit... nothing found  
Searching for TC2 Worm default files and dirs... nothing found  
Searching for Anonoying rootkit default files and dirs... nothing found  
Searching for ZK rootkit default files and dirs... nothing found  
Searching for ShKit rootkit default files and dirs... nothing found  
Searching for AjaKit rootkit default files and dirs... nothing found  
Searching for zaRwT rootkit default files and dirs... nothing found  
Searching for Madalin rootkit default files... nothing found  
Searching for Fu rootkit default files... nothing found  
Searching for ESRK rootkit default files... nothing found  
Searching for rootedoor... nothing found  
Searching for ENYELKM rootkit default files... nothing found  
Searching for common ssh-scanners default files... nothing found  
Searching for Linux/Ebury - Operation Windigo ssh... not tested  
Searching for 64-bit Linux Rootkit ... nothing found  
Searching for 64-bit Linux Rootkit modules... nothing found  
Searching for Mumblehard Linux ... nothing found  
Searching for Backdoor.Linux.Mokes.a ... nothing found  
Searching for Malicious TinyDNS ... nothing found  
Searching for Linux.Xor.DDoS ... INFECTED: Possible Malicious Linux.Xor.DDoS installed  
/tmp/ks-script-Bg5fkS  
Searching for Linux.Proxy.1.0 ... nothing found  
Searching for suspect PHP files... nothing found  
Searching for anomalies in shell history files... nothing found  
Checking `sniffer'... enp0s3: PF\_PACKET(/usr/sbin/dhclient)  
virbr0: not promisc and no PF\_PACKET sockets

Сообщения, которые могут быть выданы при сканировании:

* not infected — проверка не обнаружила ничего подозрительного
* INFECTED — программа с большой вероятностью относится к руткиту
* not tested — проверка не была произведена (для этой ОС отсутствует возможность проверки, отсутствие модуля проверки, заданы параметры командной строки, отключающие эту проверку)
* not found — программа не найдена и поэтому не проверялась
* Vulnerable but disabled — вероятность того, что приложение является вредоносным велика, но в момент проверки оно было неактивным.

Для регулярной проверки системы, можно добавить выполнение сканирования chkrootkit в планировщик под пользователем root.

**[root@localhost chkrootkit-0.52]# crontab –e**

no crontab for root - using an empty one  
crontab: installing new crontab

## Сбор информации до и после взлома

Утилита dd создает побитовый образ файлов и разделов. Рассчитав контрольные суммы для каждого образа и проверив их на совпадение, можно сравнить образ раздела или файла перед взломом с образом из взломанной системы.

**[root@localhost david]# dd if=/bin/ls of=ls.dd | md5sum ls.dd > sum.txt**

229+1 записей получено  
229+1 записей отправлено  
 скопировано 117656 байт (118 kB), 0,171573 c, 686 kB/c

Команда file полезна для определения, изменился ли набор статических библиотек, используемых программой, что может указывать на то, что программа модифицирована злоумышленником.

**[root@localhost david]# file /bin/ls**

/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=3d705971a4c4544545cb78fd890d27bf792af6d4, stripped

Команда stat выводит информацию о состоянии файла, включая время последнего доступа, разрешения, значения битов UID и GID и т.д. Это может использоваться для определения последнего использования или модификации во взломанной системе.

**[root@localhost david]# stat /bin/ls**

  Файл: «/bin/ls»  
  Размер: 117656 Блоков: 232 Блок В/В: 4096 обычный файл  
Устройство: fd00h/64768d Inode: 12822786 Ссылки: 1  
Доступ: (0755/-rwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)  
Контекст: system\_u:object\_r:bin\_t:s0  
Доступ: 2018-04-26 22:22:03.188000000 +0800  
Модифицирован: 2016-11-06 04:46:15.000000000 +0800  
Изменён: 2018-04-25 21:25:59.713617724 +0800  
 Создан: -

Команда md5sum используется для вычисления контрольной суммы файла. Вы можете создать список контрольных сумм для популярных файлов, которые обычно могут изменяться при взломе системы. В случае модификации файла контрольная сумма также изменится.

**[root@localhost david]# md5sum /bin/ls >> sums.txt**

## Заключение

Мы рассмотрели как выполняется сканирование портов nmap, а также несколько простых примеров использования этой утилиты. Эти команды nmap могут быть полезными многим системным администраторам, чтобы улучшить безопасность их систем. Рассмотрели системы обнаружения вторжения. анализирующие активность в сети или системе на предмет неавторизованных и злонамеренных действий. И программу chkrootkit, предназначена для поиска враждебного кода (rootkit) и иных подозрительных событий в системе

## Список используемых источников

1. Сетевые сканеры, анализаторы, системы обнаружения вторжения, антируткиты https://drive.google.com/drive/folders/1SUpqELdGy0O2zEDbmfnx35zViexy5xZL
2. КАК ПОЛЬЗОВАТЬСЯ NMAP ДЛЯ СКАНИРОВАНИЯ СЕТИ https://losst.ru/kak-polzovatsya-nmap-dlya-skanirovaniya-seti
3. Tripwire : Install https://www.server-world.info/en/note?os=CentOS\_7&p=tripwire
4. https://www.lisenet.com/2017/configure-tripwire-on-centos-7/
5. Развертывание Snort IDS на CentOS 5.6 https://habr.com/sandbox/29825/
6. Установка и использование Chkrootkit на Centos/Debian <https://kamaok.org.ua/?p=813>
7. Установка, настройка и использование сканера уязвимостей chkrootkit https://habr.com/company/first/blog/243487/
8. Развертывание Snort IDS на CentOS 5.6 https://habr.com/sandbox/29825/