Informe de Amenazas de Malware: Mirai Botnet

1. Resumen

Mirai es un malware que convierte dispositivos IoT (como cámaras IP, routers y DVRs) en bots controlados remotamente para llevar a cabo ataques DDoS (Distribuido de Denegación de Servicio). Fue descubierto en 2016 y ha sido responsable de algunos de los mayores ataques DDoS de la historia, como el que afectó a Dyn, un proveedor de DNS.

Este malware tiene dos partes clave. Se usa para recopilar una red de bots de todo el mundo y luego estos bots saturan servidores con datos, dejándolos inoperables.

Mirai es un gusano autorreplicante que infecta dispositivos IoT vulnerables con firmware explotable y se propaga a otros dispositivos. Utiliza una lista de credenciales predeterminadas para forzar el acceso a dispositivos que aún usan configuraciones de fábrica. Además, elimina otros malwares para tomar control exclusivo del dispositivo y borra registros para ocultar su presencia. Originalmente, Mirai solo afectaba dispositivos IoT con el sistema operativo Linux, pero ahora existen variantes que también infectan dispositivos Android.

El Mirai original explotaba vulnerabilidades principalmente en enrutadores y cámaras. En su primer día, logró controlar 67,000 dispositivos, permitiendo ataques con hasta 350,000 bots a la vez. Debido al crecimiento de los dispositivos IoT en áreas urbanas, la mayoría de estos bots estaban en Sudamérica y Asia.

Usando un servidor de comando y control (C2), la red de bots realiza ataques de denegación de servicio distribuido (DDoS). Estos ataques inundan un servidor o recurso de red con paquetes de datos, agotando sus recursos y dejándolo incapaz de responder a usuarios legítimos. En otras palabras, nadie puede conectarse y usar el servidor normalmente.

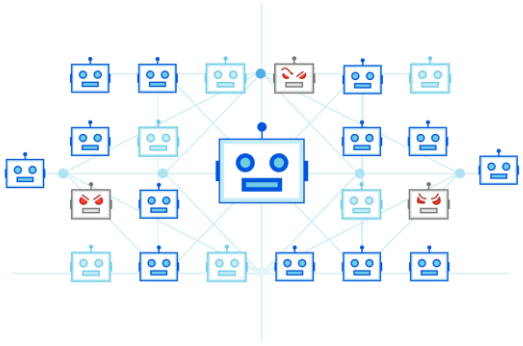

Imagen: red de bots

2. Tipo de Malware

- **Familia:** Botnet
- **Tipo:** Malware IoT
- **Objetivo principal:** Dispositivos IoT con credenciales predeterminadas o vulnerables.

3. Forma de Infección

- **Explotación de credenciales predeterminadas:** Mirai escanea dispositivos en busca de puertos abiertos (como Telnet) y utiliza diccionarios de credenciales predeterminadas para acceder.
- **Vulnerabilidades no parcheadas:** Explota vulnerabilidades conocidas en firmware de dispositivos IoT.

4. Forma de Trabajar en la Máquina

- Una vez infectado, el dispositivo se conecta a un servidor de Comando y Control (C&C).
- El operador de la botnet puede enviar comandos para lanzar ataques DDoS contra objetivos específicos.
- El malware elimina rastros de su presencia y evita la reinfección por otros malware compitiendo por los mismos recursos.

5. Indicadores de Compromiso (IoC)

- **Tráfico de red inusual:** Aumento en el tráfico saliente hacia direcciones IP desconocidas.
- **Puertos abiertos:** Presencia de puertos Telnet (23) o SSH (22) abiertos y accesibles.
- **Procesos sospechosos:** Ejecución de procesos desconocidos en dispositivos IoT.
- **Credenciales modificadas:** Cambios no autorizados en las credenciales de acceso del dispositivo.

6. Prevención y Mitigación

- Cambiar credenciales predeterminadas en dispositivos IoT.
- Mantener el firmware actualizado.
- Monitorear el tráfico de red en busca de actividad inusual.
- Utilizar firewalls para bloquear puertos innecesarios.

Análisis del Código Fuente de Mirai

1. Método de Infección y Propagación

Mirai escanea redes en busca de dispositivos vulnerables con credenciales débiles. El código relevante se encuentra en scanner.c:

```c
int check_credentials(char *username, char *password) {
    return (!strcmp(username, "admin") && !strcmp(password, "1234"));
}
```

Este fragmento ilustra una verificación simplificada de credenciales predeterminadas.

2. Mecanismo de Ataque (DDoS)

Mirai implementa varios métodos de ataque DDoS, como UDP flood y SYN flood. Un ejemplo en attack_udp.c:

```c
void attack_udp_flood(int target_ip, int port) {
    char packet[1024];
    memset(packet, 0, sizeof(packet));
    sendto(socket, packet, sizeof(packet), 0, (struct sockaddr *)&target, size
}
```

Este fragmento envía paquetes UDP vacíos a un objetivo para consumir su ancho de banda.

3. Evasión y Persistencia

Mirai intenta eliminar competidores y ocultar su ejecución. En killer.c, se observan comandos como:

```c
system("killall telnetd");
system("rm -rf /var/log/*");
```

Esto impide conexiones remotas y borra registros que podrían revelar su actividad.

4. Comunicación con el Servidor C&C

Mirai recibe comandos desde un servidor remoto. En bot.c, la conexión se gestiona así:

```c
int sock = socket(AF_INET, SOCK_STREAM, 0);
connect(sock, (struct sockaddr *)&server, sizeof(server));
write(sock, "Hello, Master!", 15);
```

Esto permite al bot recibir instrucciones como ataques DDoS o autoactualización.

Conclusión

Mirai demuestra cómo el malware IoT explota debilidades comunes en seguridad. Este análisis no debe usarse con fines maliciosos, sino como referencia para mejorar la protección de dispositivos conectados.