



UNIVERSIDAD DE GRANADA

BOTNETS

Seguridad y Protección de Sistemas Informáticos

Curso 2023/2024

Daniel Alconchel Vázquez
Mario Rodríguez López
Juan Fernández de Cañete

Índice

1. Introducción	3
2. Conceptos y Definiciones. ¿Qué es un Botnet?	3
3. Tipos de Redes Botnets	4
3.1. Basadas en los Protocolos de Comunicación	4
3.2. Basadas en la Topología de la Red	4
3.3. Híbridos	5
4. Creación y Control de Botnets	5
4.1. Creación de software malicioso	5
4.2. Búsqueda e infección	5
4.3. Mantenimiento de la Botnet	6
5. Peligros de la Botnet	6
6. Protección frente a infecciones	7
6.1. Prevenir un ataque	7
6.2. Detección de infecciones	7
6.3. ¿Como hacemos una búsqueda más exhaustiva para verificar si tenemos botnets?	8
6.4. Desinfección de un dispositivo	10
7. Demostración Práctica	10
7.1. Ataque HTTP flood	10
7.2. Slowris	11
8. Conclusiones	13

1. Introducción

En la presente sociedad altamente digitalizada, nos hallamos de manera constante inmersos en la utilización de nuestros dispositivos electrónicos. Esta extendida interconexión entre individuos conlleva numerosos beneficios, sin embargo, también acarrea desventajas notables. Uno de los inconvenientes más significativos radica en la vulnerabilidad de nuestra seguridad, donde los ciberatacantes despliegan diversas tácticas con el propósito de comprometer nuestra información y afectar de múltiples maneras la integridad de nuestras máquinas.

Entre las estrategias empleadas, sobresalen las conocidas como **botnets**, cuya definición será examinada en la sección subsecuente, siendo utilizadas por los atacantes para infiltrar nuestros dispositivos. En este documento, se profundizará en la comprensión de qué constituye una botnet, las distintas variantes existentes, los actores principales involucrados y los riesgos más prominentes inherentes al empleo de botnets por parte de los agresores.

Asimismo, se proporcionará información sobre cómo identificar la presencia de una botnet, estrategias para prevenir ser víctima de ataques perpetrados mediante este medio y medidas para mitigar los daños en caso de que un dispositivo ya haya sido comprometido.

Finalmente, se llevará a cabo una demostración práctica simulando un ataque de denegación de servicio (DDoS) a un servidor web implementado mediante el framework Flask, utilizando una simulación de una botnet compuesta por 4 dispositivos. Cabe destacar que esta exhibición tiene como objetivo ilustrar el funcionamiento, aunque es imperativo señalar que la ejecución de un ataque real requiere de recursos que no están disponibles en este contexto.

2. Conceptos y Definiciones. ¿Qué es un Botnet?

El término **Botnet** proviene de la combinación de las palabras robot y red. Un *robot* es un programa informático que realiza tareas automáticamente, mientras que una *red* es un conjunto de computadoras conectadas que intercambian información.

Se define **Botnet** como un conjunto de dispositivos infectados con malware, llamados bots, que son controlados remotamente por un atacante para llevar a cabo actividades delictivas.

Son herramientas que posibilitan la realización de diversos tipos de ataques, pero no son los ataques en sí mismos. El control sobre las máquinas infectadas varía según el tipo de malware presente, desde simples solicitudes a una dirección IP hasta la capacidad de descargar y ejecutar archivos o iniciar una shell remota.

Lo preocupante es que los ciberdelincuentes pueden controlar estos equipos de forma remota sin que los usuarios lo noten. Cada bot tiene un papel específico dentro de la red. En una red de bots, se distinguen dos grupos principales:

- **Zombie:** Estos bots ejecutan órdenes del Botmaster. Se encuentran infectados y en un estado de letargo hasta recibir comandos para ejecutar.
- **Botmaster:** Controla los bots de tipo 'zombie' y, en algunos casos, a otros 'Botmaster'. Puede estar fuera de la red infectada pero conectado a través de Internet. En este contexto, también se utiliza el término 'Botherder' para referirse al gestor de la Botnet. Los Botherders suelen utilizar sistemas operativos basados en Linux o derivados de BSD para gestionar las Botnets debido a su idoneidad para la administración de sistemas.

La disponibilidad de un dispositivo en la red depende de su conexión y estado de encendido. Por lo tanto, infectar tantos dispositivos como sea posible es conveniente, no solo para aumentar la potencia del ataque, sino también porque rara vez se tiene acceso a la integridad total de la red. Aunque el tamaño de las Botnets sigue siendo objeto de investigación, se estima que Botnets desmanteladas, como Rustock que operó desde 2006 hasta marzo de 2011, podrían haber estado compuestas por más de 1 millón de bots según informes.

3. Tipos de Redes Botnets

La clasificación de las redes Botnet se puede realizar según varios criterios. A continuación, se presentan diversas taxonomías basadas en algunos de estos criterios.

3.1. Basadas en los Protocolos de Comunicación

Para que el Bot Master se comuniquen con los bots zombies, es esencial que haya una conexión entre las computadoras infectadas y la computadora desde la que se envían las órdenes. Las Botnets se clasifican según los protocolos de red de la siguiente manera:

- **IRC-oriented:** Es uno de los primeros y más simples tipos de Botnet. Se controla a través de canales IRC (Internet Relay Chat), un sistema de chat basado en texto diseñado para comunicaciones en grupos de discusión. Cada computadora infectada se conecta al servidor de IRC, que indica el programa Bot y espera las órdenes del Bot Herder en canales específicos.
- **IM-oriented:** Este tipo de Botnet es poco común en comparación con otros. Se diferencia principalmente en que utiliza servicios de mensajería instantánea (IM) como AOL, MSN o ICQ para sus canales de comunicación. El principal desafío radica en que los Bots deben estar constantemente conectados a la red, y cada uno requiere su propia cuenta de IM para la comunicación.

Los administradores de estas redes se enfrentan a la restricción de tener un número limitado y relativamente reducido de cuentas de IM a su disposición. Esto impone una limitación en la cantidad de dispositivos que pueden conectarse simultáneamente en un momento dado. Se pueden adoptar varios enfoques para mitigar este problema, como permitir que múltiples Bots compartan una misma cuenta, establecer horarios específicos para la conexión de cada uno, enviar datos a la cuenta del propietario y esperar un tiempo limitado antes de desconectarse.

En cualquier caso, la ejecución de las órdenes involucradas en cualquier tarea lleva demasiado tiempo en estas redes, lo que resulta en una limitación significativa para su funcionamiento eficiente.

- **WEB-oriented:** Estos son los tipos más recientes y han ganado popularidad debido a su flexibilidad y eficacia. Están diseñados para controlar a los Zombies a través de la World Wide Web. Cada Bot se conecta a un servidor, y la comunicación se realiza mediante la web.
- **Otros:** Hay otros tipos de redes Botnet que se comunican usando únicamente su propio protocolo que está basado en la pila de protocolos TCP/IP, esto es, sólo utilizan los protocolos pertinentes a la capa de transporte como TCP, ICMP y/o UDP.

3.2. Basadas en la Topología de la Red

Un servidor C&C (Command and Control) es una computadora controlada por el atacante o cibercriminal utilizada para enviar comandos a los Bots y recibir datos en respuesta. La estructura de las Botnets en el contexto C&C varía, y la elección entre diferentes posibilidades recae en el Bot Herder, quien evalúa sus ventajas e inconvenientes según la tarea a realizar. Destacan cuatro tipos de redes:

- **Estrella:** En esta topología, un único servidor se comunica con todos los Bots. Cuando un equipo es infectado, se conecta al servidor central y se registra como parte de la red. La ventaja principal es la rapidez de comunicación entre el Bot Herder y la red, facilitando el mantenimiento y corrección de la red, así como ataques más rápidos. Sin embargo, la desventaja es la alta dependencia del servidor C&C; la eliminación o bloqueo del servidor neutralizará toda la red.
- **Multi-Servidor:** Es una extensión lógica de la topología estrella que utiliza varios servidores para comunicarse con los Bots. Los servidores se comunican entre sí para gestionar la red, y si uno falla, los demás continúan. Colocar servidores en ubicaciones estratégicas agiliza la comunicación con los Bots y complica legalmente los ataques. Las ventajas incluyen redundancia y resistencia, pero se incrementan los costos y la complejidad de la gestión.

- **Jerárquica:** Esta topología utiliza algunos Bots como proxys para redirigir órdenes del servidor a otros Bots. Ningún Bot es consciente de la situación completa de la red, facilitando tácticas de propagación y la posibilidad de liberar o vender partes de la Botnet. La desventaja es la mayor latencia en la comunicación debido a intermediarios, dificultando ciertos tipos de ataques.
- **Descentralizada:** Este tipo sigue una arquitectura dinámica maestro-esclavo o P2P. Las órdenes se inyectan a través de cualquier Bot, enviándose mensajes especiales firmados por una autoridad que autoriza al Bot a propagar la orden. La ventaja es su resistencia a bloqueos, pero tiene desventajas como la elevada latencia y la relativa facilidad de identificar miembros de la red mediante la monitorización de un Bot infectado y los dispositivos con los que se comunica.

3.3. Híbridos

Como se discutió anteriormente, cada estrategia de C&C tiene sus ventajas y desventajas en términos de uso, control, dificultad de descubrimiento y abandono. Con el objetivo de aprovechar las ventajas de cada modelo C&C, ya sean de diferentes protocolos o arquitecturas, ha surgido la estrategia híbrida.

Un ejemplo de esta estrategia es la implementación de botnets HTTP2P, que utilizan el protocolo HTTP para eludir firewalls sobre una estructura P2P, eliminando así los inconvenientes asociados con el servidor central C&C tradicional. La estrategia híbrida no se limita al uso de servicios o arquitecturas específicas; de hecho, los botmasters pueden emplear cualquier protocolo aplicable para implementar este modelo.

Un caso concreto de esta flexibilidad es AHP2P, una botnet P2P híbrida que utiliza la tecnología web 2.0 para ocultar sus comunicaciones en sitios web sociales. La estrategia híbrida ofrece a los atacantes la capacidad de adaptarse y evolucionar, aprovechando diversas tecnologías para mejorar la eficacia y la resistencia de las botnets.

4. Creación y Control de Botnets

4.1. Creación de software malicioso

En la etapa inicial, el propietario de la botnet debe desarrollar algún tipo de software que, al ejecutarse inadvertidamente en la máquina de la víctima, establezca el mecanismo C&C, permitiendo que la máquina se una a la botnet de manera imperceptible. Para mantener el sigilo, es común utilizar troyanos, programas informáticos que emplean la estrategia del caballo de Troya para infiltrarse en una máquina y tomar control de esta. Normalmente, se implementan en lenguajes dirigidos y orientados a objetos, permitiendo fragmentar el software en diferentes partes para tomar control de la máquina y posteriormente ser gestionada dentro de la botnet.

4.2. Búsqueda e infección

Una vez se ha desarrollado el software necesario para infectar un ordenador, es crucial lograr que se ejecute en la máquina de la víctima. Para ello, el botmaster suele infectar nuevos dispositivos mediante métodos comunes de inserción de malware:

- **Correos electrónicos de suplantación de identidad (phishing):** El atacante envía correos electrónicos fraudulentos haciéndose pasar por entidades legítimas como empresas, reclutadores, equipos de soporte técnico, empleadores, o colegas. Estos correos electrónicos contienen archivos adjuntos maliciosos, macros o enlaces que, al ser activados por los usuarios, instalan automáticamente el malware de la botnet en sus computadoras.
- **Sitios web maliciosos:** Algunos sitios web albergan malware en diversos elementos como imágenes, videos, canciones, presentaciones de diapositivas, archivos, software y anuncios. Los enlaces y botones también pueden estar infectados. Cuando los usuarios visitan estos sitios y descargan archivos multimedia infectados o hacen clic en enlaces corruptos, el troyano de la botnet puede infectar sus computadoras o dispositivos.

- **Explotaciones de vulnerabilidad:** El botmaster escanea Internet en busca de dispositivos con vulnerabilidades conocidas. Se aprovechan de estas vulnerabilidades para insertar malware en los dispositivos. Una vez que un dispositivo se infecta con un troyano de la botnet, busca otros dispositivos vulnerables para infectar y unirse a la misma red botnet.

4.3. Mantenimiento de la Botnet

Una vez establecida una botnet lo suficientemente grande, es crucial que el botmaster realice tareas para mantener su integridad y evitar posibles cierres o secuestros. Para este propósito, se emplean diversas tecnologías, siendo una de las más destacadas el **DNS Fluxing**, que abarca actividades destinadas a ocultar la ubicación real de recursos dentro de la red.

El **DNS Fluxing** implica cambiar registros DNS con extrema frecuencia y reducir significativamente el tiempo TTL del dominio para evitar que otros servidores DNS almacenen en caché la información, obligando siempre a consultar el DNS raíz del dominio. En este contexto, el recurso oculto sería el mecanismo C&C de la botnet. Existen dos formas principales de implementar DNS Fluxing:

- **Fast Flux o IP Flux (Flujo de IP):** Consiste en el constante cambio de dirección IP asociada a un nombre de dominio con el fin de dificultar la localización del ataque. Las Botnet abusan de esta capacidad de cambiar la dirección IP asociada a un dominio vinculando varias direcciones IP y cambiando rápidamente las direcciones vinculadas. Hay tres tipos de IP Flux:
 - *Single-flux:* Involucra varias direcciones IP asociadas a un nombre de dominio, registrándolas y desregistrándolas rápidamente con valores de TTL muy cortos.
 - *Name Server Flux:* Implica el cambio frecuente de las direcciones IP de los servidores de nombres DNS.
 - *Double-flux:* Combina las técnicas de Single-flux y NS fluxing.
- **Domain Flux (Flujo de Dominios):** Es el inverso del IP flux y consiste en el constante cambio y la asignación de múltiples dominios a una sola dirección IP. Podemos diferenciar entre:
 - *Domain Wildcard:* Utiliza un comodín en un dominio superior para que múltiples dominios apunten a una misma dirección IP. Puede asociarse con Botnets de spam y phishing.
 - *Domain Generation:* Crea una lista dinámica de múltiples nombres de dominio cada día, sondeados por el zombie bot. La rápida rotación dificulta la investigación y bloqueo de nombres de dominio.

5. Peligros de la Botnet

El principal propósito de crear una botnet es aprovecharla para actividades maliciosas que generen beneficios al botmaster. Destacaremos algunos de los usos más importantes:

- **Ataques Distribuidos de Denegación de Servicio (DDoS):** Comprometer un ataque DDoS mediante una botnet es una de las principales amenazas a la seguridad de la información. En este tipo de ataque, los bots visitan un sitio web específico y lo inundan con numerosas solicitudes simultáneas, agotando el ancho de banda y haciendo que el sitio web sea lento o no responda.
- **Ataques de Fuerza Bruta:** Los bots reciben una lista de sitios web o direcciones IP junto con pares de nombres de usuario y contraseñas proporcionados por el botmaster. Intentan autenticar estas credenciales en las direcciones IP designadas, informando al servidor de C&C en caso de éxito.
- **Robos de Datos:** Las botnets permiten a los ciberdelincuentes robar información confidencial de los dispositivos host. Los troyanos instalados en estos dispositivos recopilan datos confidenciales y los envían al botmaster para chantajear o utilizar de alguna manera en su beneficio.

- **Difusión de Malware:** Una vez que un dispositivo se infecta, el malware se propaga automáticamente a otros dispositivos para reclutar bots en la misma red. Esto puede incluir el envío de enlaces maliciosos, archivos adjuntos y correos electrónicos de phishing a los contactos del usuario, corrompiendo otros dispositivos conectados.
- **Minería de Criptomonedas:** Las botnets pueden programarse para minar criptomonedas utilizando el poder de cálculo acumulado de miles de computadoras simultáneamente, permitiendo al botmaster robar más criptomonedas de manera más rápida.

Dada la variedad de usos maliciosos de las botnets, es crucial evitar que una máquina pertenezca a una o, si ya está comprometida, detectarlo y abordarlo. Las secciones siguientes se enfocarán en abordar estos problemas.

6. Protección frente a infecciones

Aunque los ordenadores pueden ser vulnerables a ataques debido a fallos en el sistema operativo, en la mayoría de los casos, las infecciones se deben a la falta de medidas de protección por parte del administrador del dispositivo. Aquí se presentan algunas estrategias para prevenir un ataque de botnet:

6.1. Prevenir un ataque

- **Usar Firewall:** El firewall controla y prohíbe el acceso no autorizado a nuestro equipo a través de la red.
- **Limitar Compartir Red:** Aunque compartir una red entre equipos puede ser conveniente, es aconsejable limitar esta función y desactivarla cuando no sea necesario para protegerse contra ataques de botnet.
- **Mantener el Sistema Actualizado:** Las actualizaciones de software incluyen parches de seguridad que corrigen fallos y vulnerabilidades que los atacantes podrían aprovechar. Mantener el sistema actualizado minimiza el riesgo de hackeos.
- **Utilizar Antivirus:** Los antivirus pueden detectar y prevenir la actividad de troyanos, spyware y gusanos, que son comúnmente utilizados en botnets. Los gusanos, en particular, pueden propagarse sin intervención del usuario.
- **Realizar Copias de Seguridad:** Hacer copias de seguridad regularmente es crucial. En caso de ser víctima de ransomware, tener copias de seguridad permite regresar a un punto anterior para recuperar el equipo.
- **Usar contraseñas seguras:** Utilizar contraseñas sólidas es esencial. Algunos ataques de botnets, como el malware Mirai, aprovechan contraseñas predeterminadas en dispositivos como routers y cámaras IP.
- **Cuidado y precaución:** Evitar ejecutar archivos de remitentes desconocidos o descargados de sitios web no confiables. Verificar la autenticidad de los correos electrónicos recibidos y utilizar funciones hash para garantizar la integridad de los archivos descargados.

6.2. Detección de infecciones

Determinar si tu ordenador pertenece a una botnet puede ser complicado, pero algunos síntomas clave incluyen:

- **Tu PC nunca se apaga adecuadamente o tarda más de lo normal:** Indicativo de posibles procesos en segundo plano que impiden un apagado correcto.
- **Tus contactos reciben correos que nunca enviaste:** Algunos agentes utilizan botnets para difundir spam, virus, software espía y fraudes.

- **Aparición de spam en todas partes:** Los dispositivos que pertenecen a una botnet pueden recibir el mismo spam que están enviando.
- **Ventiladores y disco duro trabajan inesperadamente en reposo:** Los bots pueden activarse cuando el equipo está inactivo para evitar ser detectados.
- **Internet funciona inusualmente lento:** Aunque es un indicio, estos síntomas no garantizan la pertenencia a una botnet.

Para disipar sospechas, la Oficina de Seguridad del Internauta ofrece el **Servicio AntiBotnet**, que permite identificar incidentes de seguridad relacionados con botnets desde tu conexión a Internet. La herramienta verifica tu dirección IP pública con la base de datos de INCIBE, que proporciona información en tiempo real sobre IPs públicas españolas asociadas a estas redes.

- **Resultado Positivo:** Indica que algún dispositivo compartiendo tu conexión a Internet, ya sea el que estás usando u otro, puede estar infectado por un malware de botnet u otras amenazas.
- **Resultado Negativo:** Significa que la dirección IP pública de tu red no ha sido registrada en las últimas 3 horas en la base de datos, por lo que no se puede garantizar que el dispositivo no esté infectado, ya que la base de datos puede no haber sido actualizada.

6.3. ¿Como hacemos una búsqueda más exhaustiva para verificar si tenemos botnets?

Ante la incapacidad de asegurar la seguridad de todos los dispositivos conectados a la red se hace necesario detectar Botnet para intentar incapacitarlas. Existen diferentes propuestas para buscar Botnets:

- **Métodos basados en Honeynets:** En general los métodos de Honeynet tienen dos partes, el Honeytrap y el Honeywall. Honeytrap denota un host que es imita sistemas muy vulnerables a ataques maliciosos y por lo tanto es susceptible de verse comprometido en un breve lapso de tiempo. Honeywall es el software utilizado para monitorizar, recolectar, controlar y modificar el tráfico que pasa a través del Honeytrap. Por lo tanto su principal uso es recopilar información relativa a los bots. Con la información recopilada es posible aprender y entender la tecnología utilizada y realizar un análisis completo de la Botnet. A menudo es viable descubrir el servidor C&C, vulnerabilidades de la red y técnicas y herramientas usadas por el atacante y su motivación. Las Honeynets también pueden utilizarse para obtener los binarios de los bots e infiltrarse en sus Botnets. Algunas técnicas de defensa emplean Honeynets para capturar bots. Aunque las Honeynets son esenciales para entender las características y la tecnología utilizada en una Botnet tienen ciertas limitaciones:
 - Capacidad limitada para el seguimiento de las actividades maliciosas
 - No es posible capturar bots que utilicen métodos de propagación distintos de los basados en el escaneo, spam y descargas de la web
 - Solo es posible analizar las máquinas infectadas presentadas como trampas
- **Análisis Host:** Este tipo de técnicas se basan en analizar el comportamiento de la máquina. Cuando un programa bot se ejecuta realiza una serie de llamadas a las librerías del sistema (cambios en los registros, en el sistema de archivos, en la red...) algo diferentes de las que ejecutan los procesos legítimos. Por ejemplo, cuando un antivirus no puede realizar una actualización de su base de datos puede ser indicativo de una infección malware. Una de las ventajas de este tipo de aproximaciones es que son mucho más efectivas frente a ataques de descarga y en general al comienzo de cualquier infección. A pesar de ser una aproximación interesante para minimizar la expansión del malware realizar exámenes individuales a cada uno de los equipos es un proceso complejo, costoso y nada escalable.

- **Monitorización Pasiva:** Otra posible aproximación consiste en pasivamente monitorizar el tráfico que se produce en la red y utilizarlo para extraer los paquetes relacionados con las Botnets. Existen varios tipos de paquetes que se pueden analizar, entre ellos se encuentran los relacionados con los metadatos del tráfico, los datos DNS, los datos Netflow, datos privativos de la organización, complejidad, límites en los tiempos de respuesta etc... A continuación se describen algunas de las principales estrategias basadas en estos principios.

- **Basados en Comportamiento. Basados en firmas:** Para esta técnica son necesarios conocimientos de la Botnet junto con conocimientos de la autenticación utilizada por los bots. Se tiene una base de datos de comandos, patrones y nombres de autenticación comúnmente utilizados en Botnets y se coteja contra la misma. Su mayor desventaja es que sólo se puede aplicar a Botnets cuya existencia es conocida.
- **Basados en Comportamiento. Basado en anomalías:** Esta técnica se basa, como su nombre indica, en la detección de actividad inusual en la red. En este contexto inusual es todo comportamiento que se desvíe de unos criterios establecidos. Binkley and Sigh propusieron un sistema basado en la detección de anomalías de la capa de TCP que utilizaba tokenization IRC y estadísticas de mensajes IRC para detectar bots y rastrear los servidores de la Botnets. El sistema funcionaba por medio de un analizador utilizado para recopilar información acerca de los paquetes TCP y determinar así los canales IRC utilizados. El siguiente paso consistía en cotejar el tráfico a través de los canales con una gran muestra de ejemplos en busca de actividades ilícitas. Por último los canales IRC sospechosos se marcaban como posibles canales Botnet para futuro estudio.
- **Basados en minería de datos:** Para aplicar este tipo de técnicas es necesario un gran volumen de información relacionada con el tráfico de la red de forma que se pueda crear un dataset con suficiente potencia como para extraer una hipótesis suficientemente buena. La elaboración de estos datasets es especialmente complejo y los modelos han de reentrenarse cada cierto tiempo añadiendo nuevas características y ejemplos de forma que se puedan detectar nuevas Botnet. Se tiene capacidad para detectar el tráfico DNS inusual del legítimo pero la detección de las redes y la distinción de los patrones de comunicación C&C sigue siendo una tarea muy compleja para en el diseño de los IDS 2 . Debido a que las Botnets utilizan protocolos usuales para las comunicación con las C&C unido a técnicas de ataques modernas el tráfico generado es con frecuencia muy similar al tráfico normal de la red lo cual dificulta mucho su reconocimiento.
- **Basados en DNS. Peticiones DNS Fallidas(NXDOMAIN):** Una forma de detectar la presencia de Botnets es analizando la distribución de la resolución de las peticiones DNS. Botnets usan dominios de nombres que no están registrados y por lo tanto sus peticiones DNS fallarán. Para evitar esto algunas Botnets como Torpig utilizan dominios de baja entropía para evitar ser detectados. Por lo tanto necesitan un gran cantidad de dominios para funcionar y la mayoría de estos dominios fallarán en su propósito.
- **Basados en DNS. Monitorización de Dominios Maliciosos:** Esta técnica se basa en comprobar las peticiones a todos los dominios DNS y comprobar que ninguno de ellos está en una lista negra como la de DNSBL. Un gran número de organizaciones como SpamRats generan una lista negra de dominios consideradas fiables. La gran desventaja de esta técnica es que no sirve para la detección de Botnet nuevas que no incluyan los dominios DNS antiguos por saber de su pertenencia a cualquiera de estas listas.
- **Basados en DNS. Detección de Tráfico Inusual:** Se basa en la búsqueda de dominios que tengan peticiones DNS inusuales. El objetivo es detectar a los bots por medio de anomalías como un repunte repentino en el tráfico, tráfico en puertos poco comunes, latencia de la red, etcétera. Todos estos sucesos pueden apuntar a la existencia de una Botnet.
- **Basados en DNS. Detección de Tráfico Inusual:** Como ya se ha hablado en el punto 3.3 los hackers utilizan una táctica conocida como flujo IP para modificar constantemente de las IP asociadas con un dominio puesto que cuando la IP cambia detectar fallos se vuelve más complicado. Pero ocurre que estos dominios tienen un TTL muy bajo lo que significa que el sistema DNS refresca la cache de IP relacionada con el dominio reiteradas veces. Esto convierte a los dominios con TTLs bajos en sospechosos si bien es cierto que esta técnica tiene muchos falsos positivos

debido a la gran cantidad de sistemas legítimos conectados a la red que utilizan técnicas similares para otros propósitos.

6.4. Desinfección de un dispositivo

Finalmente, si hemos detectado una infección disponemos de diversas medidas para deshacernos de esta:

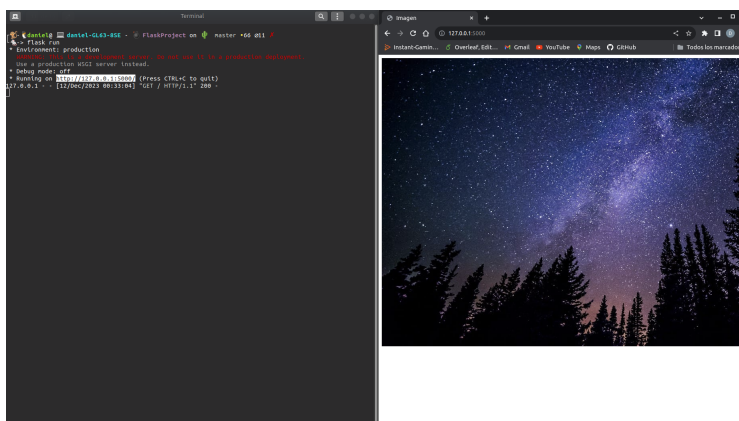
- **Programar un análisis durante el arranque:** Detecta malware antes de que el sistema operativo inicie, eliminando amenazas conocidas.
- **Emplear software especializado en desinfección:** Herramientas como HitmanPro para eliminar una variedad de amenazas, o específicas como Zbotkiller para el troyano Zeus.
- **Cerrar el tráfico a ciertas direcciones IP:** Utiliza listas negras en el router para bloquear el tráfico a direcciones IP asociadas con la botnet.
- **Restaurar el equipo a un punto anterior:** Utiliza copias de seguridad para restaurar el sistema a un estado anterior, eliminando el malware.
- **Infectar a la Botnet para inutilizarla:** Intenta tomar el control de los servidores C&C o desconectar el bot de tu máquina de sus pares. Para conseguir esta infección podemos emplear la técnica denominada peer-poisoning que consiste en introducir equipos capaces de convertirse en nodos de la Botnet. Cada nodo llamado 'contaminado' se encargará de distribuir listas de direcciones IP no válidas a los demás nodos. Cuando se hayan logrado introducir suficientes, la Botnet comenzará a fallar, ya que al tratarse de una red entre pares, puede que gran parte de la red de bots quede aislada del resto.

7. Demostración Práctica

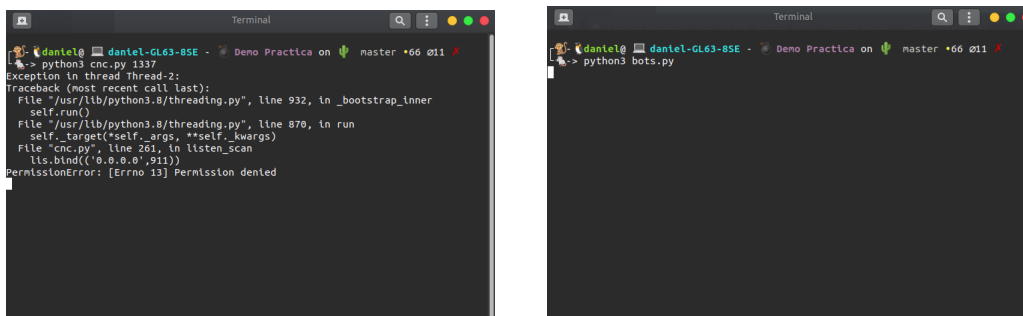
A continuación vamos a realizar dos ataques a un servidor web simulando comportamientos de un botnet centralizado. La simulación se lleva a cabo con botnet escrito en python, consta de un script que lanza un servidor C&C local para controlar los bots y otro script que permite crear dichos bots.

7.1. Ataque HTTP flood

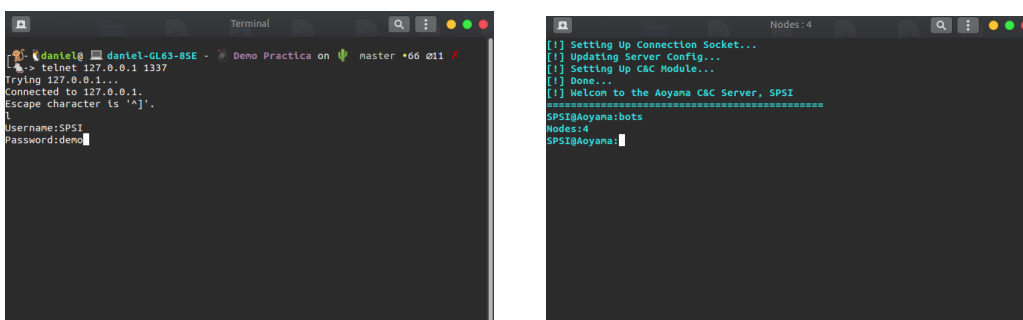
Los ataques HTTP flood son un tipo de ataque DDoS en el que consiste en el envío de múltiples solicitudes GET o POST a partir de numerosos dispositivos con el fin de inundar el servidor con solicitudes o respuestas entrantes. En consecuencia, se producirá una denegación de servicio a las solicitudes adicionales que provengan de fuentes de tráficos legítimas. Comenzamos lanzando un servidor web Flask en nuestro equipo, que será el objetivo de nuestros ataques, este servidor nos muestra simplemente una página con un imagen de cielo estrellado:



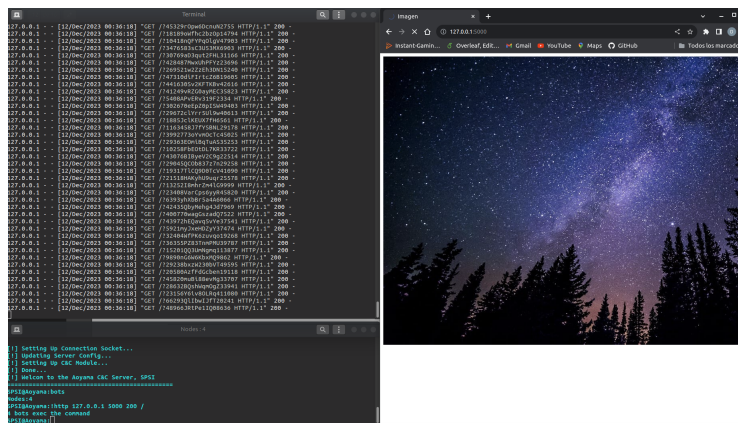
Acto seguido, lanzamos nuestro servidor C&C y nuestros bots. En este caso vamos a lanzar 4 bots:



Una vez lanzado el servidor C&C podemos conectarnos a través de telnet introduciendo el usuario y podemos comprobar los números de bots que están en línea:



Por último, lanzamos el ataque, para lo que usamos el comando 'http' para indicar que el ataque que lanzamos es HTTP flood. Indicamos también la url y el puerto, junto al número de hebras por bot que realizarán peticiones http. En este caso, cada bot solicitará 200 peticiones GET al servidor web:



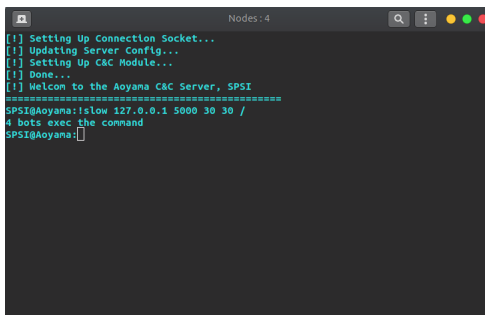
Tras lanzar el ataque, podemos observar que el servidor Flask se ha llenado de peticiones GET solicitados por los bots. Si intentamos recargar la página, podemos ver que tarda mucho más por la cantidad de peticiones, que están provocando la saturación del servidor.

7.2. Slowris

Un ataque Slowris es también un tipo de ataque DDoS. En esta ocasión, usamos envíos de encabezados de solicitudes HTTP parciales. Manteniendo esas solicitudes abiertas con envíos periódicos de encabezados de solicitudes parciales, hace que el servidor objetivo no sea capaz de liberar ninguna de estas conexiones,

agotando así su máximo de conexiones posibles. Una vez que esto ocurra, el servidor será incapaz de responder solicitudes adicionales realizadas desde el tráfico regular.

Todo el despliegue es igual que en el punto anterior, solo cambia el ataque que indicamos. Esta vez, indicamos 'slow', la url a atacar, el puerto, el número de hebras por bot y el número de conexiones al servidor por hebra.

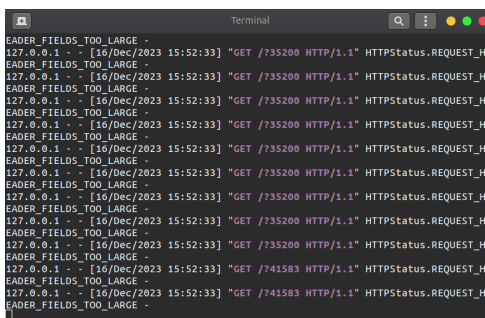


```

Nodes: 4
[1] Setting Up Connection Socket...
[1] Updating Server Config...
[1] Setting Up C&C Module...
[1] Done...
[1] Welcom to the Aoyana C&C Server, SPSI
=====
SPSI@aoyana:slow 127.0.0.1 5000 30 30 /
4 bots exec the command
SPSI@aoyana:

```

Tras el lanzamiento, podremos ver como llegan las múltiples solicitudes HTTP al servidor flask de gran tamaño:

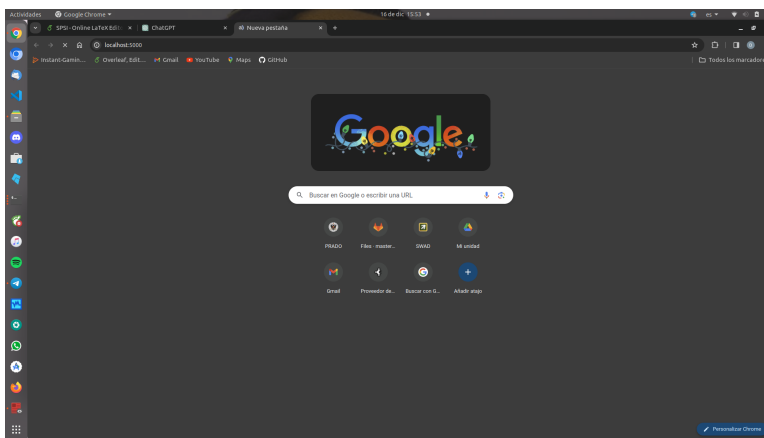


```

EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /735200 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /741583 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -
127.0.0.1 - [16/Dec/2023 15:52:33] "GET /741583 HTTP/1.1" HTTPStatus.REQUEST_H
EADER_FIELDS_TOO_LARGE -

```

Los bots mantendrán las conexiones de dichas solicitudes agotando así el número de conexiones posibles del servidor, hasta que, finalmente, agota toda la capacidad del servidor. Si en ese momento intentamos acceder al servidor, podemos comprobar que esto es imposible, ya que no tiene capacidad suficiente para aceptar nuevas solicitudes:



8. Conclusiones

Después de llevar a cabo este trabajo, podemos deducir que las redes botnet representan una amenaza significativa. Por un lado, un usuario sin conocimientos en ciberseguridad es susceptible de integrarse en una botnet. Por otro lado, la potencia de estas redes no radica en su complejidad computacional, sino en la cantidad de ordenadores que las componen. Por lo tanto, cuanto más extensa sea la infección de ordenadores, más difícil será dismantelarlas y mayor será su capacidad de ataque.

No obstante, es posible prevenir este tipo de ataques como usuario tomando las precauciones necesarias. Los sistemas informáticos son susceptibles, y un ataque a estos puede tener consecuencias catastróficas. Por ende, es crucial que la sociedad no confíe totalmente en dispositivos electrónicos para almacenar documentos esenciales.

En resumen, nos encontramos en una era altamente digitalizada, donde cualquier individuo puede caer víctima de un ataque de este tipo. Por lo tanto, es responsabilidad de todos los usuarios con dispositivos electrónicos adquirir un conocimiento mínimo en seguridad, para evitar que personas con intenciones maliciosas puedan tomar el control de nuestros dispositivos.

Referencias

- [1] AYUDALEY. *Así funciona una botnet o red de ordenadores zombie*. 2022. URL: <https://ayudaleyprotecciondatos.es/2021/04/21/botnet/>.
- [2] CLOUDFLARE. *¿Qué es una red de robots (botnet) de DDoS?* URL: <https://www.cloudflare.com/es-es/learning/ddos/what-is-a-ddos-botnet/>.
- [3] CLOUDFLARE. *Ataque DDoS con Slowloris*. URL: <https://www.cloudflare.com/es-es/learning/ddos/ddos-attack-tools/slowloris/>.
- [4] CLOUDFLARE. *Ataque de inundación HTTP*. URL: <https://www.cloudflare.com/es-es/learning/ddos/http-flood-ddos-attack/>.
- [5] MEISAM ESLAHI, ROSLI SALLEH, Y NOR BADRUL ANUAR. *Bots and botnets: An overview of characteristics, detection and challenges*. En: 2012 IEEE International Conference on Control System, Computing and Engineering. 2012, págs. 349-354. DOI: 10.1109/ICCSCE.2012.6487169.
- [6] INCIBE. *Botnet*. URL: <https://www.incibe.es/aprendeciberseguridad/botnet>.
- [7] LEEON123. *Aoyama*. URL: <https://github.com/Leeon123/Aoyama>.
- [8] NETACEA. *Types of botnets*. 2021. URL: <https://netacea.com/glossary/types-of-botnets/>.
- [9] ONESPAN. *¿Qué son los bots, las redes de bots y los zombis?* 2020. URL: <https://www.onespan.com/es/blog/que-son-los-bots-las-redes-de-bots-y-los-zombis>.
- [10] OSI. *Desinfecta tus dispositivos*. URL: <https://www.osi.es/es/desinfecta-tu-ordenador>.
- [11] REDESZONE. *Detecta si tu equipo ha sido infectado y es una botnet*. URL: <https://www.redeszone.net/tutoriales/seguridad/detectar-equipo-botnet-seguridad/>.
- [12] REDESZONE. *Evita ataques de botnet en tu equipo con estos consejos*. URL: <https://www.redeszone.net/tutoriales/seguridad/evitar-ataques-botnet-dispositivos/>.
- [13] WIKIPEDIA. *DNS Fluxing*. 2021. URL: https://es.wikipedia.org/wiki/DNS_Fluxing.
- [14] WIKIPEDIA. *Mirai (malware)*. URL: [https://es.wikipedia.org/wiki/Mirai_\(malware\)](https://es.wikipedia.org/wiki/Mirai_(malware)).