infoblox

DECOY DOG NO ES UN PUPY NORMAL:

Separar un malware Sly DNS del paquete

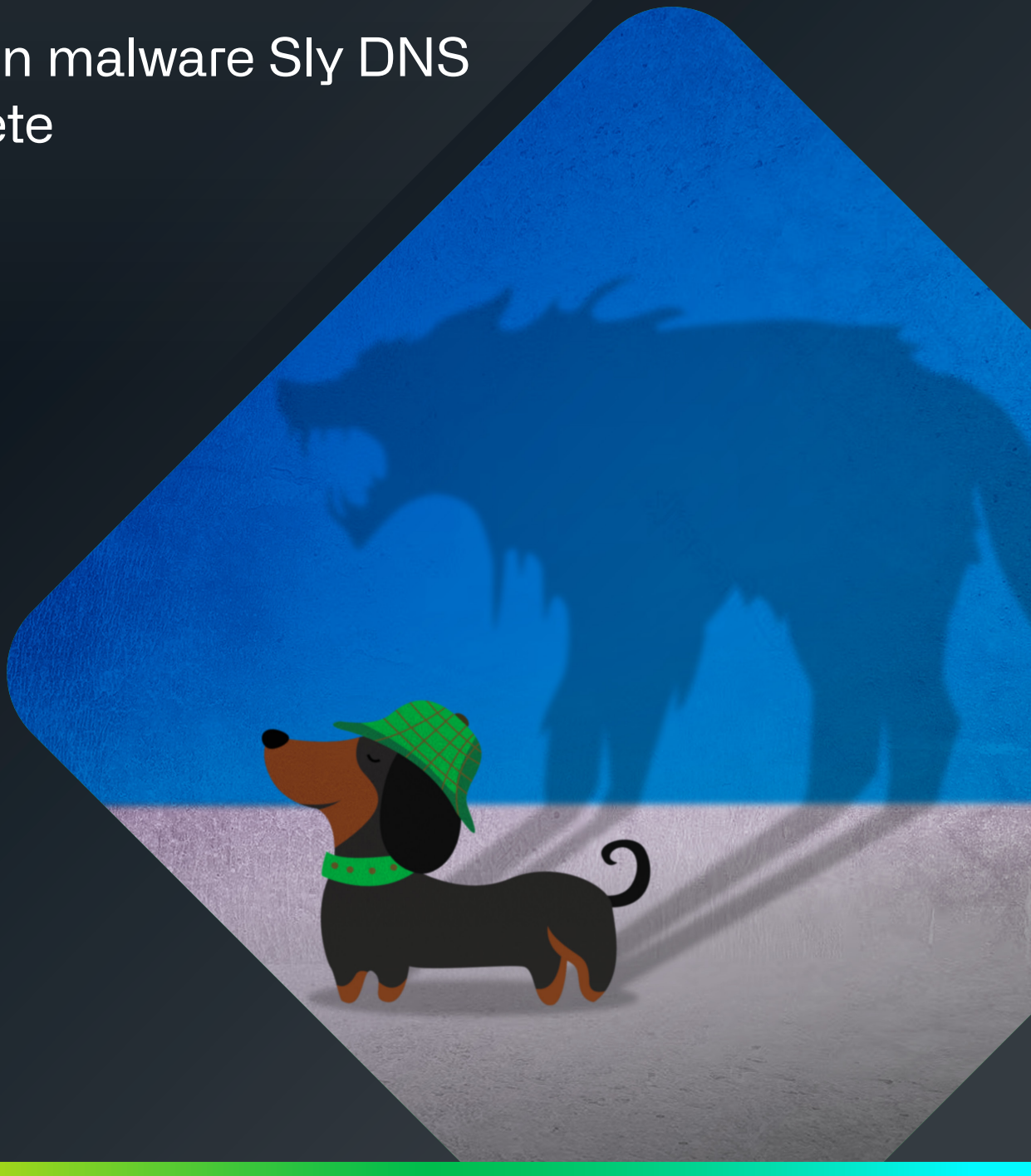

TABLA DE CONTENIDO

RESUMEN EJECUTIVO

Decoy Dog es un kit de herramientas de malware descubierto por Infoblox que utiliza el sistema de nombres de dominio (DNS) para realizar comandos y controles (C2). Un cliente comprometido se comunica con un controlador y recibe instrucciones de un controlador a través de consultas DNS. Ese controlador está integrado en un servidor de nombres DNS al que se transmiten las consultas a través del proceso de resolución normal. Revelamos la existencia de Decoy Dog en abril de 2023 y publicamos un informe detallado de nuestros hallazgos iniciales el 23 de abril. El descubrimiento se basó en la monitorización de los datos del DNS. El análisis en ese momento confirmó que el kit de herramientas se construyó sobre la base de un troyano de acceso remoto (RAT) conocido como Pupy, pero no se sabía qué sistemas estaban siendo explotados, cómo se implementó el kit de herramientas o si Pupy había sido modificado.[1] Esperábamos que, con los detalles que proporcionamos, otros miembros de la comunidad localizaran las máquinas comprometidas y se conociera la historia completa. Sin embargo, el misterio que rodea a Decoy Dog no ha hecho más que crecer.

Desde abril, Infoblox ha llevado a cabo nuevas investigaciones sobre Decoy Dog y Pupy. Este informe es el resultado de esa investigación. Hemos aprendido que Decoy Dog es una actualización importante de Pupy que utiliza comandos y configuraciones que no están en el repositorio público. Desarrollamos algoritmos para separar las comunicaciones de los clientes de Decoy Dog e inferir una serie de otras propiedades sobre cada controlador. Esto nos permite concluir con mucha confianza que el kit de herramientas se ha extendido y está bajo el control de al menos tres actores. Aunque la actividad que hemos observado sigue estando confinada en Rusia y Europa del Este, existen agrupaciones diferenciadas de técnicas, tácticas y procedimientos (TTP) dentro de los controladores que concuerdan con múltiples actores.

Todos los actores de Decoy Dog respondieron a nuestras revelaciones de abril de alguna manera, y las variaciones respaldan nuestra evaluación de múltiples operadores. Inmediatamente después del primer anuncio en las redes sociales, se retiraron algunos de los servidores de nombres. Todos los restantes se modificaron para eliminar el comportamiento que destacamos en nuestro primer artículo, aunque esto se logró de diferentes maneras según el controlador. Un conjunto de controladores comenzó a restringir las respuestas a las consultas en función del país de origen, una técnica llamada geofencing, mientras que otros modificaron su respuesta a las consultas para el subdominio ping.

Un actor respondió tan rápido a nuestra revelación en LinkedIn que al principio pensamos que los nuevos dominios eran registros copiados por investigadores de seguridad. Sin embargo, un análisis adicional mostró que eran dominios de reemplazo. En lugar de cerrar su operación, el actor transfirió los clientes comprometidos existentes a los nuevos controladores. Se trata de una respuesta extraordinaria que demuestra que el actor sentía que era necesario mantener el acceso a sus víctimas existentes. Creó una separación clara entre los TTP de un conjunto de dominios Decoy Dog y todos los demás.

En las semanas siguientes a nuestro anuncio, nos sorprendió que nadie se presentara para identificar el malware y la vulnerabilidad subyacentes que permitieron a Decoy Dog operar. Sin embargo, a medida que avanzaba nuestra investigación, quedó claro por qué las comunicaciones no se habían detectado durante más de un año. Los ataques con Decoy Dog han sido muy selectivos y cada controlador tiene un pequeño número de clientes activos. Algunos servidores han mantenido sistemáticamente de cuatro a ocho clientes activos durante meses. Aunque otros vieron un mayor número de clientes activos simultáneamente a lo largo del tiempo, el número total de dispositivos afectados observados en cualquier momento ha sido inferior a 100. Un conjunto de víctimas pequeño descarta en general a los actores con motivaciones financieras, y la necesidad de persistir en un dispositivo durante un largo periodo de tiempo es coherente con actores muy avanzados.

---

1 https://github.com/n1nj4sec/pupy

Pudimos reconstruir partes de las comunicaciones de Decoy Dog identificando firmas de nuestro propio tráfico de Pupy. Establecimos un servidor Pupy en Internet, que, cuando se combinó con la ingeniería inversa selectiva del código, nos permitió correlacionar las consultas y respuestas de DNS a comandos específicos de Pupy. A partir de esto pudimos a) determinar que Decoy Dog contiene comandos que no se encuentran en Pupy, y b) caracterizar la mayoría de las comunicaciones. Además, los actores de Decoy Dog parecen aprovechar Pupy para utilizar otras capas de transporte distintas del DNS para funciones como el intercambio de claves. Es probable que los actores de amenazas consideren que esta es una de las ventajas de Pupy como troyano de acceso remoto (RAT).

La primera implementación conocida del kit de herramientas Decoy Dog tuvo lugar a finales de marzo o principios de abril de 2022. Se vendió o se robó poco después, como lo indica la aparición de un segundo mando, con diferentes TTP, que estaba activo a mediados de mayo. Se registró un tercer dominio en julio de 2022 y envejeció estratégicamente hasta septiembre. Es posible que estos dos últimos controladores sean propiedad del mismo actor, ya que comparten muchas características, incluido el alojamiento en el espacio IP ruso. Sin embargo, hay algunas diferencias. Unos meses más tarde, se registraron dos dominios más, de nuevo con características distintas a los controladores anteriores. El actor que registró estos dominios migró a sus clientes inmediatamente después de que nos revelaran a nuevos dominios. En total, Infoblox está monitorizando actualmente 21 dominios de Decoy Dog, algunos de los cuales se registraron e implementaron en el último mes.

Habiendo determinado que Decoy Dog difería significativamente de Pupy a través de nuestro análisis de los registros de DNS, examinamos muestras binarias relacionadas disponibles en VirusTotal para ver si las diferencias eran evidentes en los ejecutables. La ingeniería inversa de estas muestras reveló que, aunque se detectaron como Pupy, son mucho más avanzadas que la versión de código abierto. Las muestras incluyen a) la capacidad de ejecutar código Java arbitrario en el cliente, b) varios mecanismos de transporte nuevos y c) nuevos mecanismos DNS para garantizar la persistencia. Un mecanismo es similar a un algoritmo de generación de dominios (DGA) de DNS tradicional y utiliza proveedores de DNS dinámicos gratuitos para conectarse a los llamados controladores de emergencia. Todas las muestras comparten las mismas actualizaciones fundamentales, aunque una de las muestras tiene capacidades únicas que no se ven en las demás, relacionadas con el uso de transportes de streaming.

Por razones que siguen sin estar claras, Decoy Dog viola los principios básicos de las comunicaciones encubiertas, cuyo objetivo general es evitar la detección y recuperación del contenido por parte de un adversario. Aunque los servidores normales de Pupy rechazan las consultas de comunicación repetidas de clientes comprometidos, los servidores Decoy Dog no solo responden a las consultas DNS reproducidas, sino que responderán a cualquier consulta bien diseñada. Este comportamiento es similar a las configuraciones de comodines en DNS y fue un factor importante en la detección de Decoy Dog por parte de Infoblox. Dada la sofisticación de Decoy Dog, especulamos con la posibilidad de que la repetición y el comportamiento de comodín sean intencionados sea cual sea la intención, la repetición generalizada del DNS fue parcialmente responsable de la incapacidad de la industria para ver a Decoy Dog como un nuevo malware.

Un agresivo escaneado de Internet por parte de un proveedor de seguridad condujo a la retransmisión de millones de comunicaciones de Decoy Dog a través de redes mundiales, incluidos varios de nuestros clientes. Esto, a su vez, nos llevó a descubrir el conjunto de herramientas. La incapacidad del vendedor para identificar el tráfico como malware, para evitar reproducir las consultas, activó las conexiones DNS desde redes no infectadas a los controladores Decoy Dog. Estamos seguros de que ningún cliente de Infoblox fue infectado y que las consultas a nuestros solucionadores fueron todas resultado de un escaneo anómalo del proveedor. A pesar de la falta de una amenaza inmediata para nuestras redes de clientes, Decoy Dog sigue siendo un conjunto de herramientas sofisticado con orígenes inciertos y puede que siga extendiéndose.

No solo se ha observado Decoy Dog por primera vez en la naturaleza, sino que, hasta donde sabemos, es el primer uso del componente DNS C2 de Pupy en una operación maliciosa. En parte, es probable que esto se deba a la dificultad de establecer un servidor de nombres Pupy, lo que requiere modificar el software en el repositorio y configurar correctamente el DNS. La falta de exposición hace que sea más difícil para la industria de la seguridad detectar y defenderse tanto de Pupy como de Decoy Dog. Para ayudar a interrumpir las operaciones que utilizan estos sistemas C2, estamos proporcionando a la comunidad un conjunto de datos de investigación que contiene el tráfico DNS de Pupy capturado desde nuestro propio servidor y detalles del funcionamiento interno del software. Esta documentación es la primera de su tipo y permitirá a otros construir algoritmos de detección, así como reproducir nuestros hallazgos.

La historia de Decoy Dog revela el potencial del DNS como fuente de detección y respuesta a amenazas. También revela una debilidad inherente del ecosistema de inteligencia centrado en el malware que domina el sector de la seguridad. El kit de herramientas fue descubierto por los algoritmos de detección de amenazas DNS, y la única defensa contra él en la actualidad es DNS. Además, habíamos marcado varios dominios de control como sospechosos y los hemos bloqueado en nuestros solucionadores anteriores para darnos cuenta de que todos estaban usando un malware común. Este tipo de protección, que genera actividad maliciosa antes de identificarla, y a menudo antes de que se ejecute, es único para los sistemas de detección y respuesta de DNS.

En este documento proporcionamos a los defensores los conocimientos necesarios para identificar Pupy y Decoy Dog. Aunque describiremos el DNS C2 en profundidad, no proporcionaremos información que ayude a los agentes malintencionados a implementar Pupy, ni divulgaremos la firma completa de Decoy Dog DNS. Explicamos algunos comportamientos que identificamos en nuestro artículo original y destacamos en qué se diferencia Decoy Dog de Pupy. Además, describiremos nuestro análisis de grandes volúmenes de tráfico DNS de Decoy Dog que nos permitió estimar el número de clientes y el tráfico de comandos sin poseer el malware en sí o controlar el servidor de nombres. Describimos en qué se diferencian las muestras de Decoy Dog de Pupy. Por último, analizamos cómo reaccionaron los operadores de Decoy Dog a nuestras revelaciones y demostramos rasgos comunes entre los subgrupos de controladores. Los apéndices contienen información técnica adicional de apoyo.

CONTEXTO

Infoblox descubrió Decoy Dog, un conjunto de herramientas de mando y control (C2) que utilizaba el sistema de nombres de dominio (DNS) a principios de abril de 2023. Se basa en un troyano de acceso remoto (RAT) de código abierto llamado Pupy[2] y transporta la comunicación cifrada entre clientes y servidores, o controladores, mediante consultas de nombres de dominio y respuestas de direcciones IP. El descubrimiento surgió de algoritmos que monitorizaban las consultas de DNS pasivas a los solucionadores de Infoblox para detectar un comportamiento anómalo. Se habían realizado consultas sobre dominios de Decoy Dog a partir de dispositivos de seguridad en un pequeño número de redes de clientes. Estas consultas crearon una firma coherente con el comportamiento persistente de malware de perfil bajo. La revisión humana de la actividad fue alarmante porque, aunque el DNS se estaba utilizando claramente como un canal de comunicación confidencial, los dominios no se identificaron como C2 en ningún dato de inteligencia disponible públicamente. De hecho, algunos fueron etiquetados como “acreditados” en los verificadores de reputación en línea. Publicamos un conjunto de dominios el 13 de abril para ayudar a la comunidad a bloquear el tráfico e identificar la naturaleza del compromiso.

2 https://malpedia.caad.fkie.fraunhofer.de/details/win.pupy

Durante nuestra investigación original, Infoblox identificó una firma DNS única que era independiente del software Pupy. Los actores habían implementado y operado su sistema C2 de una manera muy específica; por esta razón, identificamos Decoy Dog como un conjunto de herramientas distinto. Solo un pequeño número de dominios en todo el mundo compartían esta firma, todos los cuales eran servidores de nombres de Decoy Dog.

El 23 de abril, publicamos parte de la firma, el análisis inicial del DNS pasivo y un subconjunto de los dominios del controlador en nuestro informe "Dog Hunt: Finding Decoy Dog Toolkit in Anomalous DNS Traffic."[3] Este artículo destacó un comportamiento específico de Pupy, que devolvió una serie de respuestas localhost a consultas para subdominios específicos que contienen 'ping'. También describió una serie de tendencias dentro de las comunicaciones DNS que no pudimos explicar completamente en ese momento. En particular, identificamos patrones sorprendentes en las direcciones IP que se devolvían en las respuestas y el hecho de que los servidores respondían a consultas reproducidas, lo cual es inesperado para un sistema de comunicación encubierto.

Tras los anuncios, una amplia gama de miembros de la comunidad de seguridad, incluidos proveedores y otras organizaciones, se comunicaron con nosotros. Muchos de ellos habían visto tráfico relacionado en sus propias redes, o en redes de sus clientes, pero nadie había identificado dispositivos comprometidos o reconocido el alcance de la actividad. Algunas de esas organizaciones proporcionaron información que nos llevó a aislar y confirmar cómo se generó el DNS en nuestras propias redes. Otros ayudaron a confirmar la amplitud de la actividad y a probar las hipótesis. Esta colaboración informal fue muy útil y estamos agradecidos.

Para simplificar, usamos el término Pupy en este documento para referirnos específicamente al Pupy DNS C2 y no a Pupy en general.

Pupy

UNA RAZA RARA

Pupy es un troyano de acceso remoto (RAT) de código abierto posterior a la explotación que cuenta con un complejo sistema de transporte modular.[4] Si bien la base de código principal de Pupy estuvo disponible en GitHub en 2015, el mecanismo DNS C2 no se agregó hasta 2019. Este artículo es la primera documentación pública de Pupy C2. Además, estamos proporcionando un conjunto de datos en GitHub para que otros reproduzcan nuestro trabajo y creen defensas para el futuro.

Aunque Pupy es de código abierto, el uso del protocolo DNS C2 es raro; no hemos podido identificar su uso fuera de Decoy Dog en la naturaleza.[5] De nuestros propios solucionadores, que sirven a empresas y organizaciones de todo el mundo, no hemos encontrado pruebas del uso histórico del DNS C2 de Pupy. En los DNS mundiales durante los seis primeros meses de 2023, mediante los detectores de DNS que hemos desarrollado para Pupy, no descubrimos ningún uso del software fuera de Decoy Dog. Por último, hemos consultado de forma privada a una amplia gama de vendedores; ninguno de los cuales lo había visto utilizado tampoco. Cuando se informó del uso de Pupy por parte de actores de amenazas

3 https://blogs.infoblox.com/cyber-threat-intelligence/cyber-threat-advisory/dog-hunt-finding-decoy-dog-toolkit-via-anomalous-dns-traffic/

4 https://github.com/n1nj4sec/pupy

5 La expresión "en la naturaleza" se utiliza en la jerga de la ciberseguridad para referirse a su implantación operativa y no como parte de pruebas de penetración o investigaciones aisladas.

persistentes avanzadas (APT), al parecer no se emplearon los componentes C2 del DNS.[6]

El raro uso de Pupy probablemente se deba, al menos en parte, a la dificultad de operar el sistema. Establecer la comunicación de Pupy a través del DNS global no es fácil. Requiere configurar correctamente el servidor de nombres y modificar el código del repositorio de GitHub. Además, hay complejidades en el DNS que varían según los resolutores recursivos y el software Pupy no gestiona correctamente. Es probable que estos desafíos hayan impedido su adopción tanto por parte de los equipos rojos como de los piratas informáticos, a diferencia de herramientas populares como Cobalt Strike, que vemos con bastante frecuencia.[7]

Aunque Pupy DNS C2 es raro hoy en día, el uso de Decoy Dog se está extendiendo y la probabilidad de que los defensores se enfrenten a Pupy de alguna forma está creciendo. Con el fin de ayudar a preparar a la comunidad, Infoblox realizó una investigación significativa tanto sobre Decoy Dog como sobre Pupy. Infoblox implementó un servidor Pupy en Internet para comparar su comportamiento con Decoy Dog. A continuación, capturamos datos de paquetes (pcap) y registros de DNS pasivos de los solucionadores de Infoblox. Utilizamos nuestra implementación de Pupy junto con la ingeniería inversa selectiva del código para comprender mejor la naturaleza única de Decoy Dog. En esta sección, explicamos los componentes de Pupy que son relevantes para nuestra investigación. Para simplificar, limitamos este documento a las comunicaciones que utilizan respuestas IPv4 (registro A), aunque cuando estén disponibles, Pupy utilizará respuestas IPv6 (AAAA). La codificación de consulta descrita en el documento es la predeterminada actual para Pupy, versión 2 (a menos que se especifique lo contrario).[8]

CÓMO FUNCIONA PUPY

En nuestro artículo anterior, dimos una descripción general de Pupy y destacamos algunas características inusuales de Decoy Dog.[9] En este artículo, profundizaremos en el protocolo de comunicación Pupy para demostrar sus conexiones con Decoy Dog y cómo explotar el DNS de Pupy recopilado pasivamente para comprender una operación en curso.

Pupy está diseñado para proporcionar comunicaciones continuas entre los clientes infectados y el servidor, de modo que cuando el actor quiera acceder de forma remota al cliente, la conexión ya se ha establecido. El actor puede supervisar a los clientes conectados y ordenarlos selectivamente para proporcionar una amplia gama de acciones. El DNS solo se utiliza para las comunicaciones C2. Cualquier dato significativo exfiltrado por el cliente se envía a través de una de las muchas otras opciones de transporte que ofrece Pupy. Como resultado, el cliente Pupy DNS se limita a registrarse con el controlador, reconocer comandos, proporcionar información del sistema y un puñado de otras tareas. Entre comando y comando del servidor, el cliente duerme.

Las comunicaciones DNS son iniciadas y mantenidas por el cliente. El cliente envía consultas a través de su ruta de resolución DNS normal o a través de DNS a través de HTTPS

6 https://www.volexity.com/blog/2022/06/15/driftingcloud-zero-day-sophos-firewall-exploitation-and-an-insidious-breach/

7 https://www.esecurityplanet.com/threats/how-cobalt-strike-became-a-favorite-tool-of-hackers/

8 Una versión anterior de Pupy C2 no incluía información del host en cada consulta. Ahora sabemos que Decoy Dog es la versión 3 del cliente, pero la codificación de consulta parece ser la misma que la versión 2.

9 https://blogs.infoblox.com/cyber-threat-intelligence/cyber-threat-advisory/dog-hunt-finding-decoy-dog-toolkit-via-anomalous-dns-traffic/

(DoH) cuando está habilitado y disponible.[10] El controlador envía comandos en respuesta a las solicitudes de los clientes en forma de direcciones IP cifradas. Cada consulta-respuesta es una comunicación completa, lo que significa que ni el cliente ni el servidor pueden dividir los datos de un solo comando en dos consultas DNS. Este protocolo se distingue de los sistemas comunes de tunelización de DNS, por ejemplo, Iodine,[11] donde el cliente establece una sesión sobre DNS que puede incluir la reconstrucción de varios paquetes en cada extremo para procesar la comunicación. El cliente está obligado a reconocer la mayoría de los comandos, y el servidor responde a cada consulta válida del cliente con comandos o acuse de recibo. El vocabulario del cliente es extremadamente limitado. Tiene nueve tipos de consultas mediante las cuales administra sesiones, reconoce comandos, envía información del sistema y establece claves. Se pueden agregar comandos personalizados escribiendo funciones adicionales, pero requieren una comprensión completa del software.

Al despertarse, el cliente consulta al servidor de una de las dos maneras diferentes, dependiendo de si se ha establecido una clave compartida. Esta consulta proporciona al servidor información actual sobre el sistema y el estado del cliente Pupy, o realiza una consulta simple que sirve para iniciar una nueva sesión cifrada. Si bien es posible deshabilitar las sesiones cifradas, este no es el valor predeterminado y no se ha observado en Decoy Dog. En respuesta, el controlador reconoce la solicitud, requiere que el cliente realice un intercambio de claves o envía nuevos comandos. Una vez completado el conjunto de comandos, el cliente dormirá durante el intervalo establecido, por defecto 60 segundos. Este proceso se repite mientras se ejecuta el cliente. En la Figura 1 se muestra una descripción general de alto nivel de las comunicaciones cliente-servidor de Pupy, y en el Apéndice A se puede encontrar una vista más detallada del proceso del cliente.

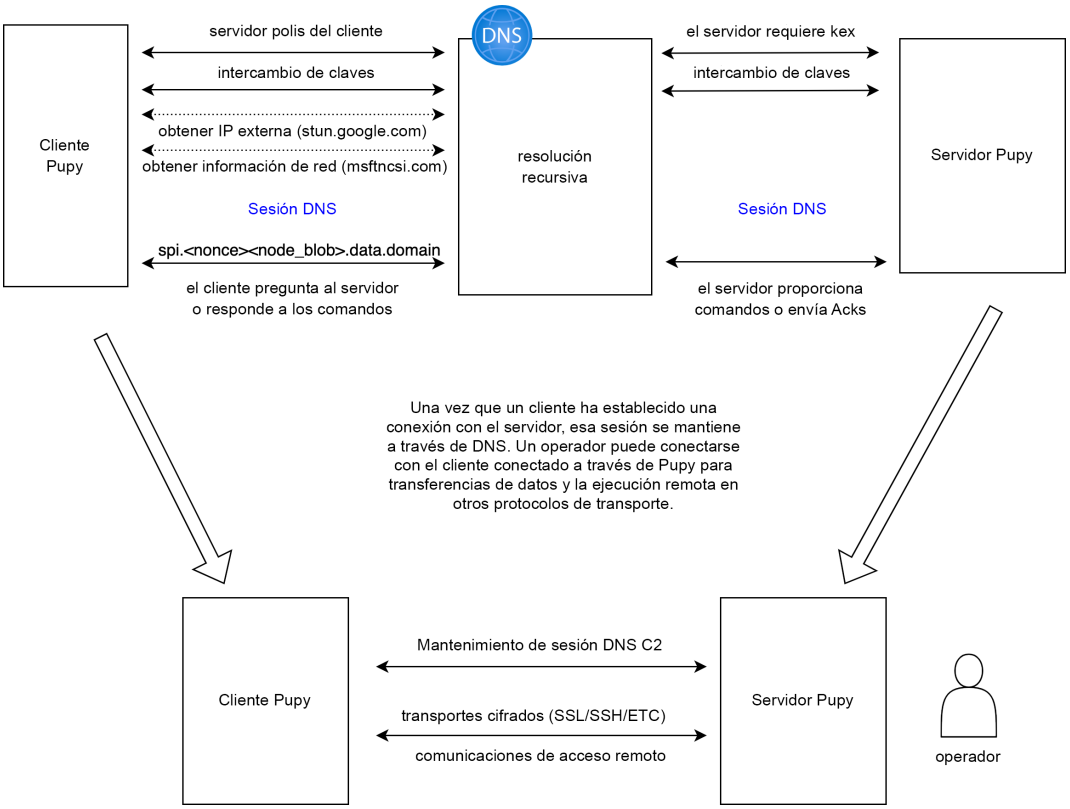

Figura 1. Una descripción general de alto nivel de las comunicaciones de Pupy.

10 Pupy utiliza servidores Quad9 de forma predeterminada para DoH.

11 https://github.com/yarrick/iodine

El actor Pupy interactúa con los clientes desde la utilidad de línea de comandos del controlador. Cuando el cliente se pone en contacto con el controlador, los comandos en cola se codificarán en la respuesta DNS. El operador establece una conexión en un puerto abierto de cliente y especifica la capa de transporte que se utilizará para la exfiltración. Las comunicaciones DNS del servidor siguen siendo bastante restringidas, aunque más extensas que las del cliente. Hay una amplia gama de comandos que se pueden concatenar en una sola respuesta al cliente. Mientras que el cliente inicia el intercambio de comunicaciones, el servidor es responsable de garantizar la seguridad de las comunicaciones. Para ello, impone las llamadas sesiones con cada cliente, que sirven para rotar las claves de cifrado. Esto se describe en la siguiente sección.

INICIO DE LA SESIÓN

Pupy requiere que se establezca una sesión cifrada entre el cliente y el controlador antes de transmitir los comandos del actor. Esta sesión caduca cuando se agota el tiempo de espera de las comunicaciones con el cliente y puede verse obligada a renovarse por otros motivos, como errores en la descodificación de la consulta DNS o un reinicio del cliente. Las sesiones se identifican por la presencia de una etiqueta de índice de parámetros de seguridad (SPI) en la consulta y se cifran mediante una clave compartida efímera. Dado que los detalles de la comunicación dependen de una variedad de factores, incluido si el cliente se ha conectado previamente al servidor, el protocolo exacto para la inicialización de la sesión puede diferir, creando varianza en los intercambios DNS observados. Sin embargo, el intercambio típico es el siguiente:

- El cliente se registra en el servidor sin una sesión establecida o con una sesión expirada (consulta 1).
- El servidor responde con un comando que requiere un intercambio de claves e información del sistema cliente.[12]
- El cliente reconoce el requisito de información del sistema (consulta 2).
- El cliente genera un par de claves aleatorias, privada y pública, utilizando un algoritmo de curva elíptica y lo envía al servidor; el servidor hace lo mismo y responde con su nueva clave (consulta 3).
- El cliente y el servidor utilizan este intercambio para establecer una nueva clave de sesión compartida, que se utiliza para cifrar paquetes con cifrado AES, y también crear el SPI para identificar la sesión.
- El cliente recopila información sobre su red, incluida su dirección IP externa, utilizando consultas DNS adicionales a otros servicios.
- El cliente transmite esta información utilizando la clave de cifrado compartida y señalando la presencia de una sesión activa con la inclusión del SPI en la consulta (consulta 4).
- El cliente envía información adicional sobre el estado del sistema (consulta 5).

La clave compartida y el SPI se establecen normalmente después de tres consultas, aunque el intercambio de claves es técnicamente una única consulta y respuesta. Durante una sesión, cada consulta y respuesta se cifrará con esta clave compartida. El cifrado también utiliza un nonce de 32 bits generado por el cliente que cambia para cada consulta. Cuando se establece una nueva sesión, las claves se regeneran, pero el valor nonce del cliente continúa mientras el cliente está funcionando. Esto se analiza con más detalle en la sección siguiente.

12 Por lo general, esto se realiza en forma de dos comandos denominados Política y Encuesta en el lado del servidor.

CODIFICACIÓN DE CONSULTAS

El cliente genera consultas que contienen comunicaciones cifradas al servidor. Estos pueden incluir información de intercambio de claves o una respuesta a comandos del servidor. Hay un máximo de 52 bytes de datos transmitidos que se pueden comunicar en cada consulta. Además de los datos transmitidos, cada consulta incluye:

- nonce, un valor incremental de 4 bytes generado por el cliente
- versión, un valor de 1 byte que indica la versión Pupy DNS C2
- cid, un valor de 4 bytes de la configuración del cliente, que se genera aleatoriamente al crear el cliente
- iid, un valor de 2 bytes que contiene los 16 bits inferiores del proceso del cliente Pupy
- id. de nodo, un valor de 6 bytes del cliente, normalmente la dirección MAC del dispositivo
- opcionalmente, SPI, un valor de 4 bytes generado durante el intercambio de claves y presente en consultas que representan una sesión en el servidor para un cliente determinado.

Cada consulta de cliente incluye estos 13 bytes de información de cliente, así como una suma de comprobación de 4 bytes sobre la carga subyacente. La carga subyacente está cifrada y consta de una serie de comandos y datos relacionados.

El cliente cifra y codifica los datos que se transmitirán al servidor como un nombre de dominio completo (FQDN), denominado nombre de consulta (qname) en el protocolo DNS. Todo el proceso, que se muestra en la Figura 2 a continuación, incluye el cifrado, la organización y la codificación tanto de los datos transmitidos como de la información adicional que necesita el servidor. Funciona de la siguiente manera:

- Los datos que se van a transmitir se adjuntan con información específica del host.
- Esta cadena de bytes compuesta se cifra utilizando una clave simétrica compartida y el nonce actual.
- Los primeros bytes cifrados de los datos transmitidos, hasta 35 bytes, se codifican y se utilizan para la primera etiqueta o la más a la derecha del qname.
- El resto de los bytes cifrados, que pueden contener hasta 17 bytes de los datos transmitidos, se antepone con el valor nonce actual y se codifica para crear la segunda etiqueta del qname.
- Si el índice de parámetros de seguridad (SPI) existe en el cliente, se codifica y se utiliza en la tercera etiqueta, o la que está más a la izquierda, del qname; este valor se establece tras un intercambio de claves con el servidor.
- El nonce se incrementa por la longitud de los datos cifrados dentro del cliente que se utilizará para la siguiente consulta.

La codificación de bytes cifrados a una etiqueta de nombre de dominio se describió en nuestro artículo anterior. Utiliza un mapa personalizado en combinación con codificación de 32 bits para garantizar que el resultado final sea un nombre de dominio válido. La estructura de carga útil de datos subyacente se describe en el Apéndice B.

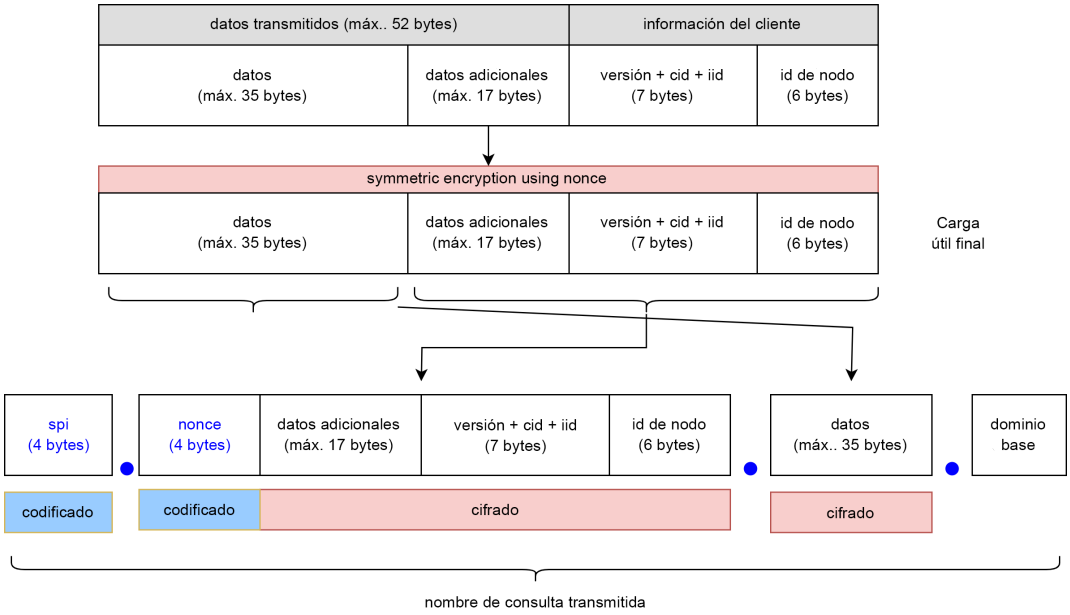

Figura 2. Proceso para convertir datos del cliente en un qname para una consulta DNS. El dominio base es el nombre de dominio del servidor Pupy.

Pupy utiliza AES de forma predeterminada al cifrar consultas de DNS. Si se ha establecido una clave compartida con el cliente, la utiliza para cifrar la cadena de bytes completa de forma simétrica, de lo contrario, utiliza la clave pública establecida. En cualquier caso, el nonce actual también se utiliza en el cifrado para garantizar que la consulta codificada sea única, incluso si los datos transmitidos subyacentes siguen siendo los mismos en varias consultas. Este es un mecanismo estándar para protegerse contra ataques criptográficos. Como resultado, el nombre de dominio consultado se puede decodificar para revelar los datos cifrados, pero los datos cifrados no se pueden descifrar sin la clave. El valor nonce se inicializa con un valor aleatorio de 32 bits y se incrementa según la longitud de la carga útil en cada consulta.

Cuando el servidor de nombre Pupy recibe una consulta, decodifica el nombre de dominio para revelar el valor SPI, el nonce y la carga cifrada. Para asegurarse de que está recibiendo comunicaciones válidas del cliente, el servidor comprueba si el SPI es válido cuando está presente, y que el nonce es mayor que el anterior registrado para el cliente. Realiza varias comprobaciones más de los datos, incluida una comprobación del número de versión, que se cifra en la carga útil. Si falla alguna de estas comprobaciones, devolverá un error al cliente.

En particular, Pupy no responde a la misma consulta dos veces y cualquier servidor Pupy no modificado responderá a una consulta que ya ha recibido en el pasado con un NXDOMAIN (no existe taldominio). Hemos validado este comportamiento con nuestro propio servidor Pupy intentando consultar un nombre de dominio previamente consultado. Esto es importante porque una característica de Decoy Dog es que responde a las consultas de DNS que se reproducen con respuestas coherentes con el protocolo Pupy C2.

Como la consulta DNS contiene una codificación reversible del nonce, y el nonce se incrementa por la longitud de la carga útil en cada consulta, podemos reconstruir hilos de consultas asociados a un único cliente. Como veremos más adelante en este artículo, dada la recopilación pasiva de DNS para un dominio Pupy o Decoy Dog, podemos usar esta reconstrucción para estimar el número de clientes, así como la naturaleza de la comunicación en ciertos casos.

GESTIÓN DE NOMBRES DE DOMINIO ESPECIAL

Al recibir una consulta, el servidor disecciona el nombre de la consulta y determina si coincide con la estructura adecuada para un paquete cifrado de un cliente. Hay algunos casos especiales que tienen un procesamiento único. Salvo en esos casos, rechazará cualquier solicitud que no cumpla con el formato esperado. Uno de esos casos especiales es las solicitudes de ping, que describimos en nuestro documento anterior. Una consulta para un subdominio pingN, donde N es un número entero, devolverá una secuencia de respuestas del host local con una longitud de N. Una consulta de ping en sí devuelve 15 de esas respuestas y una consulta del dominio base devuelve una única respuesta de localhost, es decir, 127.0.0.1.

Fuera de las solicitudes de ping, el servidor se puede configurar para responder a consultas de etiqueta única con una sola dirección IP. Se desconoce el propósito de esta funcionalidad y no parece que se utilice en el cliente; se hace referencia a ella en el código fuente como una solicitud de activación de DNS. Esta capacidad no está documentada y para utilizarla un actor tendría que entender cómo funciona el software del servidor.

El manejo especial de subdominios de etiqueta única se logra configurando entradas de “activación”, que son pares de cadenas de valor clave. El valor se utiliza junto con la clave privada del servidor para crear una dirección IP de respuesta. Esta respuesta se crea utilizando una función hash unidireccional y no se puede invertir. El hash distingue mayúsculas de minúsculas y se define como

MD5(subdomain_label + activation_value + private_key)

CODIFICACIÓN DE RESPUESTA

Cuando el servidor recibe una consulta de un cliente, decodificará, descifrará, comprobará los resultados y procesará los datos del cliente. En particular, una comunicación con el cliente formateada correctamente debe contener dos o tres etiquetas, como se describe anteriormente en la sección sobre codificación de consultas. A continuación, el servidor enviará una respuesta al cliente con uno o varios comandos. Aunque puede devolver consultas IPv4 (A) o IPv6 (AAAA), limitaremos nuestra descripción a las consultas IPv4 (A) para que resulte más sencillo.

La respuesta del servidor es una cadena binaria cifrada que luego se codifica en uno o más registros A.[13] El proceso para esta codificación se muestra en la Figura 3 a continuación. El número máximo de bytes en la respuesta es 64, codificado en 22 segmentos de bytes, lo que da como máximo 4 direcciones.

- En el primer paso, el servidor calcula la longitud de la respuesta y la antepone a los datos de la respuesta. A continuación, anexa bytes aleatorios para crear una cadena compuesta que es un múltiplo de 3 bytes de longitud.[14] A esta cadena compuesta la llamamos carga útil.
- En el segundo paso, las direcciones IPv4 se crean iterativamente a partir de segmentos de 3 bytes de la carga útil. Cada dirección IPv4 está representada por un valor de 32 bits, donde el bit 0 es el bit alto.
- Los 3 bits superiores de cada dirección son aleatorios.
- Cada segmento tiene un índice que permite al cliente pedir los datos al recibirlos; esto se representa con 5 bits. Este índice está en bits 3-7 del resultado.

13 Se reúne una serie de comandos y luego se encripta utilizando una clave compartida y el nonce actual antes de la codificación, si se ha completado un intercambio de claves. De lo contrario, se utiliza la clave privada del servidor, junto con el nonce, para cifrar los datos con un algoritmo de curva elíptica de clave pública.

14 En el código, este proceso es más complicado, pero tiene el mismo resultado.

- El segmento de carga útil se encuentra en los bits 8-30, lo que obliga a que el bit superior del segmento de carga útil sea el bit inferior del primer octeto de la dirección IPv4.
- Por último, el bit menos significativo, el bit 31, es un bit de control generado en el segmento de carga útil. Por la naturaleza de esta suma de control, este bit es 1 en el 75 % de las direcciones IPv4.
- La cadena de 32 bits resultante se interpreta como una dirección IPv4 y se anexa a la respuesta.

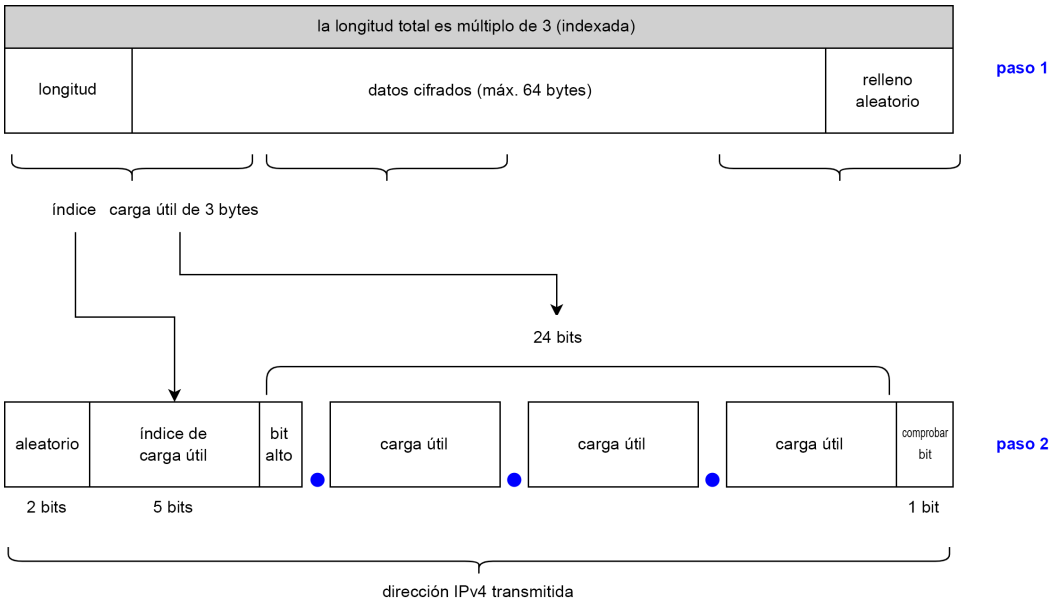

Figura 3. Codificación del servidor Pupy de una respuesta IPv4 a una consulta de cliente. Los datos se codifican en una serie de direcciones IPv4 utilizando 3 bytes de la carga útil en cada dirección.

En nuestro artículo anterior, señalamos que Decoy Dog tiene una distribución sorprendente de las respuestas de IPv4. Ahora sabemos que se trataba de un artefacto de la codificación de respuesta de Pupy. El uso de tres bits aleatorios y un índice incremental como los 7 mejores bits del primer octeto de cada garantiza que las direcciones IPv4 resultantes estén en rangos específicos y que esos rangos se correlacionan directamente con el número de respuestas de la respuesta, que se determina por el tamaño de los datos transmitidos al cliente. En particular, la primera dirección IP siempre estará en el rango 64.0.0.0/8, 128.0.0.0/8 o 192.0.0.0/8.

Cada vez que se aumenta el índice, las opciones para el primer octeto de la dirección IP se desplazan en dos. Específicamente:

- La primera dirección IP comenzará con 64, 128 o 192 porque el índice es 0 y la longitud es un máximo de 64. Como resultado, solo los 3 bits superiores se establecen en la primera dirección IP de la respuesta.
- La segunda dirección IP empezará por 66, 67, 130, 131, 194 o 195 porque el índice es 1, lo que añade 2 a los 3 bits superiores generados aleatoriamente, y el bit superior de la carga de datos puede ser un 0 o un 1.
- La tercera dirección IP comenzará con 68, 69, 132, 133, 196 o 197, etc.

Podemos ver el resultado de este algoritmo para un número creciente de respuestas en la Figura 4 a continuación. En particular, utilizamos un mapa de Hilbert para demostrar cómo el primer octeto de las direcciones IP se correlaciona con el número total de respuestas para 3, 12 y 15 respuestas.

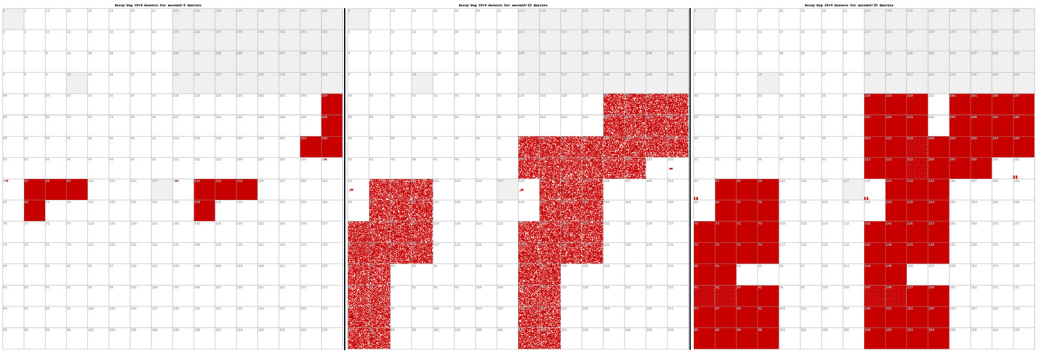

Figura 4. Mapas de Hilbert que muestran la distribución de direcciones IPv4 en respuestas de Pupy que contienen 3, 12 y 15 respuestas, respectivamente.

La estructura de las direcciones IPv4 permite a cualquier persona que observa la respuesta completa reconstruir los datos transmitidos. Aunque estos datos están cifrados, las respuestas se pueden perfilar utilizando el análisis de duración y serie temporal. Este tipo de análisis puede revelar información sobre las comunicaciones, como veremos más adelante en este trabajo.

ANÁLISIS DE DATOS PASIVOS

Si bien las comunicaciones de Pupy están fuertemente encriptadas, la información necesaria para descifrar y rastrear los paquetes se codifica de manera reversible. Si se recopilan las consultas y respuestas de DNS, se pueden analizar en conjunto para obtener información sobre la implementación y los clientes de Pupy. La colección pasiva de datos DNS, conocida comúnmente como DNS pasivo (también como pDNS), se produce en muchos lugares de Internet, incluidos solucionadores corporativos, solucionadores recursivos públicos, así como servidores raíz y TLD. En las siguientes secciones, mostramos cómo se puede explotar la recopilación pasiva de DNS de consultas de Pupy para obtener información sobre las comunicaciones.

Podemos recuperar una gran cantidad de información sobre un controlador Pupy y sus clientes a partir de DNS pasivo. En particular, podemos recuperar

- el número aproximado de clientes activos en cualquier momento,
- los tipos de intercambios que se producen entre el servidor y los clientes,
- firmas de la implementación, como el intervalo de suspensión del cliente, y
- un cronograma de intercambios de claves de cliente y actividad general.

Utilizamos estas técnicas para analizar el tráfico de nuestro propio servidor, así como de los servidores de Decoy Dog. Esto nos permitió comprender lo similar que es Decoy Dog para Pupy y cómo se parecen los servidores entre sí. En última instancia, estas técnicas nos permitieron perfilar cada implementación de Decoy Dog. Los detalles técnicos de los métodos utilizados se tratan con más detalle en el Apéndice C.

FIRMAS DE CARGA ÚTIL PUPY

La naturaleza de las comunicaciones entre un cliente y un servidor se puede inferir hasta cierto punto utilizando el análisis pasivo de datos. El vocabulario del cliente, es decir, las distintas cargas útiles que puede realizar, es muy restringido: sólo existen nueve tipos de comunicaciones con el cliente. Dos tipos comparten la misma longitud de carga útil, mientras que otro tipo puede tener varias longitudes. Un actor puede crear eventos personalizados en Pupy, lo que podría crear una diversidad adicional en la longitud de la carga útil.

El servidor tiene un vocabulario más flexible y es capaz de transmitir múltiples comandos en una sola respuesta DNS, lo que hace que sea más difícil de perfilar. Sin embargo, la gran mayoría de las comunicaciones en un sistema Pupy se relacionan con la inicialización de la sesión, el intercambio de claves y los latidos del cliente con el servidor. Las comunicaciones del servidor se dominan por el reconocimiento de las solicitudes de los clientes, los mensajes de error, incluida la necesidad de establecer una nueva sesión, e intercambios de claves.

Como resultado, se pueden crear firmas para diferentes tipos de comunicaciones utilizando las longitudes de las cargas útiles subyacentes de las consultas y respuestas de DNS. Estas firmas nos permiten separar la actividad de mantenimiento común de comandos significativos del servidor y aislar el uso de tipos de eventos personalizados. Se pueden utilizar para perfilar el comportamiento general de un cliente y servidor de Pupy observado pasivamente, incluido Decoy Dog.

En la Figura 5, mostramos un mapa térmico de las longitudes de carga útil observadas en las consultas de los clientes y las respuestas del servidor en nuestros propios datos de Pupy. Mientras que las longitudes del servidor presentan más variaciones debido a los argumentos de los comandos y a los comandos concatenados, las comunicaciones del cliente están bien definidas. Para perfilar las comunicaciones, utilizamos la longitud de la carga útil subyacente, incluidas las sumas de comprobación y la información de los nodos. Como resultado, por ejemplo, el acuse de recibo del cliente (Ack) tiene una longitud de 19 bytes y el del servidor de 6 bytes. El Apéndice D contiene tablas para las longitudes comunes de la carga útil del cliente y del servidor y su relación con los comandos.

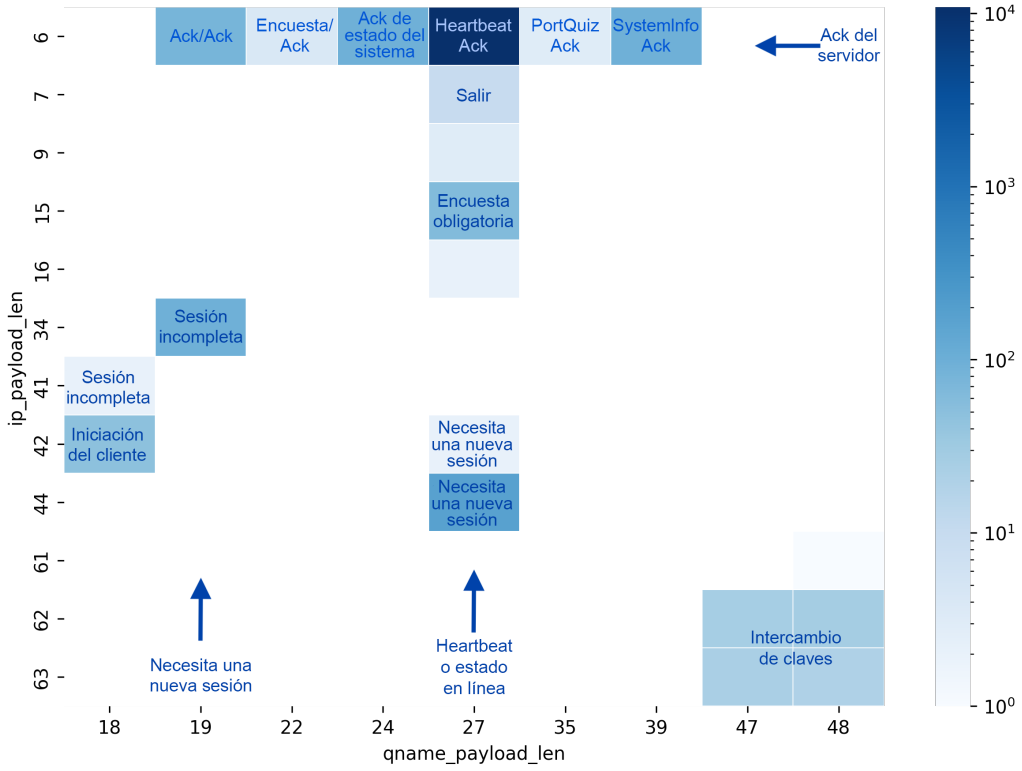

Figura 5. Una distribución anotada de pares de longitudes de carga útil comunes observados en el tráfico de Pupy. La carga útil son los datos cifrados transmitidos en la consulta o en la respuesta. Este gráfico no incluye comandos DNS C2 complejos del servidor y las celdas sin anotación no están completamente identificadas. La longitud se expresa en bytes.

Decoy Dog

Las comunicaciones de Decoy Dog se observaron tanto en los solucionadores de Infoblox como en muchos solucionadores públicos y comerciales. Para comprender mejor las operaciones de Decoy Dog y en qué se diferencia el kit de herramientas de Pupy, utilizamos otras colecciones de DNS pasivas para aumentar las nuestras. En total, nuestro análisis cubre más de 15 millones de eventos DNS durante el período de tiempo del 29 de marzo de 2022 hasta el 16 de junio de 2023. Además, analizamos activamente los servidores de nombres y comparamos el tráfico de DNS recopilado de forma pasiva con el generado por nuestro propio cliente y servidor de Pupy.

Utilizamos una serie de técnicas para comprender mejor a Decoy Dog y sus operaciones. También aplicamos ingeniería inversa a las muestras que se encuentran en el repositorio público VirusTotal, lo que validó nuestros hallazgos de DNS y reveló otras capacidades. En las secciones que siguen, describiremos nuestro análisis en detalle y mostraremos los resultados. Los aspectos más destacados de este trabajo son:

- Decoy Dog no es Pupy, sino un gran refactor que amplía significativamente las capacidades del malware y ayuda a garantizar la persistencia en un dispositivo comprometido.
- Está operado por un puñado de actores, que emplean distintas TTP y han respondido de manera diferente a nuestra revelación de abril de 2023 del conjunto de herramientas.
- El número total de dispositivos afectados es pequeño, con tan sólo cuatro en un único controlador.
- Los nuevos controladores registrados desde abril de 2023 se han adaptado para mitigar las características descritas en nuestro documento original; esto incluye mecanismos de geocerca para limitar las respuestas a las direcciones IP del cliente a determinadas ubicaciones.
- El análisis de DNS demostró ser una herramienta poderosa no solo para detectar Decoy Dog, sino también para comprender su uso y separarlo de Pupy, lo que, combinado con la ingeniería inversa selectiva, brinda una imagen sólida de Decoy Dog y la amenaza que representa.

INTERCAMBIOS DE CLAVES

Como se describió anteriormente, una sesión se inicia cuando se completa el intercambio de claves y se establece el valor SPI. En teoría, una única sesión cifrada puede continuar indefinidamente, pero en la práctica, hay una serie de condiciones bajo las cuales el controlador requerirá que se establezca una nueva sesión. Por lo tanto, una sola instancia en ejecución del cliente suele tener muchas sesiones. Con las firmas de carga útil de Pupy, podemos determinar cuándo se generaron claves compartidas entre un cliente y un servidor, y hacer estimaciones aproximadas del número de inicializaciones del cliente, ya sea desde un nuevo compromiso o un reinicio del cliente, para cada controladora a lo largo del tiempo.

La Figura 6 muestra la cronología de los intercambios de claves de varios controladores Decoy Dog. Hay brechas en los intercambios de claves observados para algunos programadores. El último intercambio de claves para claudfront[.]net se observó en diciembre de 2022, aunque la actividad de los clientes no solo continuó, sino que aumentó en 2023; más del 70 % de todos los valores únicos de SPI se observaron por primera vez en 2023. Del mismo modo, el controlador allowlisted[.]net no tuvo intercambios clave desde diciembre de 2022 hasta después de nuestra divulgación en abril de 2023. Por último, cbox4[.]ignorelist[.]com también muestra un largo periodo de tiempo sin intercambios de claves, con un pequeño número ocurriendo justo antes de que el dominio dejara de funcionar. Sospechamos que los actores reconfiguraron los clientes para realizar el intercambio de claves en un transporte diferente al DNS.

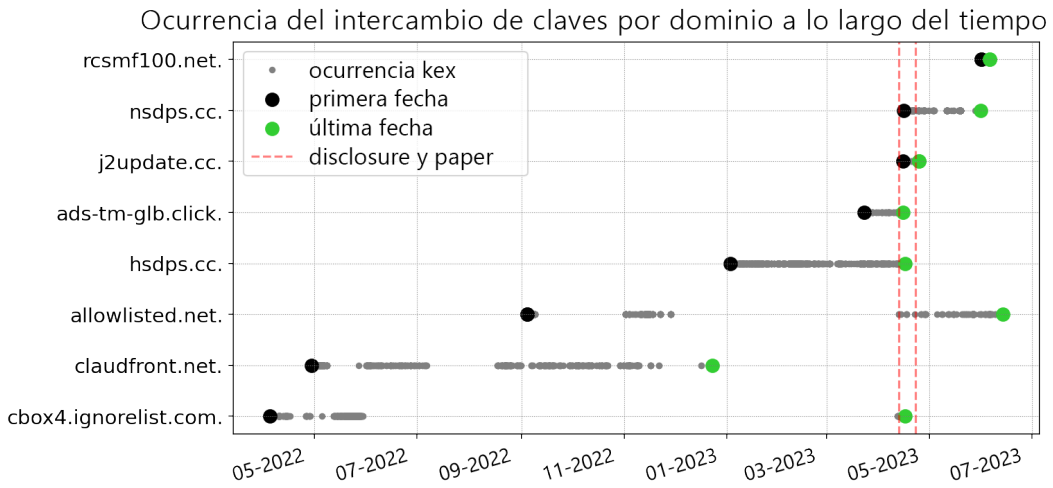

Figura 6. Cronología de los intercambios de claves observados para determinados dominios de Decoy Dog.

PLAZOS DEL CLIENTE

Además del número total de clientes, queríamos determinar cuántos clientes activos mantenía cada controlador a la vez y durante cuánto tiempo los clientes se comunicaban activamente con el servidor. Se utilizó el método de agrupación de valores nonce descrito en el Apéndice C. Este análisis dio lugar a información clave sobre las operaciones de Decoy Dog durante un largo período de tiempo, como se muestra en los gráficos siguientes. En particular:

- Todas las controladoras gestionan un pequeño número de clientes a la vez, y algunas controlan tan solo cuatro y todas probablemente menos de cincuenta.
- El dominio original, cbox4[.]ignorelist[.]com, es uno de los controladores más grandes y exhibe un salto en los clientes en múltiples puntos en el tiempo. También mantiene un número reducido de clientes muy de larga duración.
- El segundo controlador que se observará, claudfront[.]net, tiene un aumento dramático en la actividad en febrero de 2023.
- El tercer controlador que se observará, permitido[.]neto, ha mantenido constantemente un pequeño número de clientes simultáneos.
- Los controladores ads-tm-glb[.]click y hsdps[.]cc transfirieron clientes a nuevos controladores tras nuestra revelación.
- Claudfront[.]net y allowlisted[.]net no modificaron las operaciones en respuesta a nuestra divulgación, cbox4[.]ignorelist[.]com cesó sus operaciones, y ambos hsdps[.]cc y ads-tm-glb[.]click transfirieron clientes a nuevos dominios.

Aunque es difícil estimar el número total de clientes en todo momento, el pequeño número de clientes activos simultáneamente indica que estas operaciones son altamente dirigidas. También explica por qué los proveedores de seguridad no detectaron la actividad y aún no han encontrado dispositivos infectados. Los clientes infectados están presentes en un número muy reducido de redes, aparentemente aquellas que no pueden identificar y bloquear las comunicaciones C2 en DNS.

En los diagramas de líneas siguientes, representamos la actividad de un único cliente como una línea, a la que llamamos hilo de cliente. El eje Y muestra distintos subprocesos de cliente identificados por una cadena nonce. Cuando se reinicia un cliente de Pupy, mediante un reinicio u otros medios, se generará un nuevo nonce y se observará un nuevo subproceso. En algunos diagramas, hay interrupciones claras en la actividad que probablemente indican reinicios del cliente. El eje x indica el tiempo.

En la Figura 7 se muestra la actividad del cliente para el dominio inicial de Decoy Dog cbox4[.]ignorelist[.]com El primer hilo de clientes comienza a finales de marzo de 2022 y el más largo duró casi un año. Podemos ver que este controlador tenía inicialmente sólo unos pocos clientes, pero a mediados de mayo de 2022 se produjo un cambio que dio lugar a casi 40 clientes activos simultáneamente. Periódicamente se produjeron aumentos similares en los hilos de clientes, con el mayor incremento mensual en agosto de 2022; sin embargo, a medida que se iniciaban nuevos hilos de clientes, otros finalizaban. A lo largo de todo el año de actividad, el número de clientes simultáneos parece ser inferior a 50 en todo momento. También podemos ver en la Figura 7 que una cuarta parte de los hilos de clientes persistieron durante seis meses o más, lo que concuerda con una operación sostenida. Todas las comunicaciones cesaron después de la publicación de LinkedIn y no han sido observadas nuevamente.

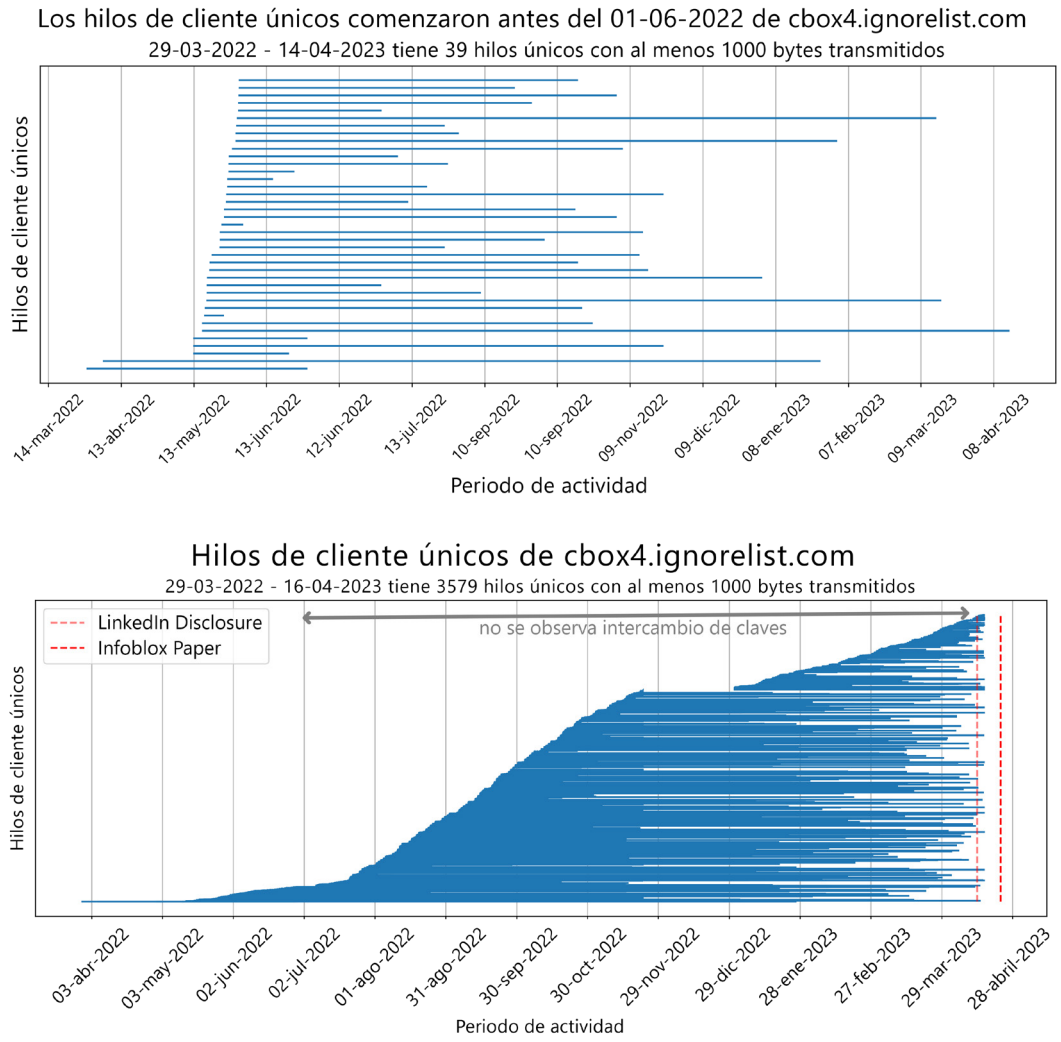

Figura 7. La figura superior muestra los clientes que estaban presentes antes del 1 de junio de 2022 y la figura inferior muestra subprocesos de clientes a lo largo del tiempo.

La actividad de DNS de claudfront[.]net, cronológicamente el segundo dominio de Decoy Dog en aparecer, es muy diferente a la de cbox4. Como se muestra en la Figura 8 de abajo, había menos de diez clientes activos simultáneamente en este controlador hasta principios de febrero de 2023. Después de ese tiempo, el número de clientes aumentó sustancialmente, aunque no en la medida que cabría esperar de una infección generalizada.

El momento de este aumento es poco antes de que se envíe una muestra binaria con el dominio del controlador a VirusTotal el 13 de febrero.[15] A diferencia de cbox4 [.] ignorelist[.] com, no hubo cambios notables en las consultas de claudfront[.]net tras nuestra revelación.

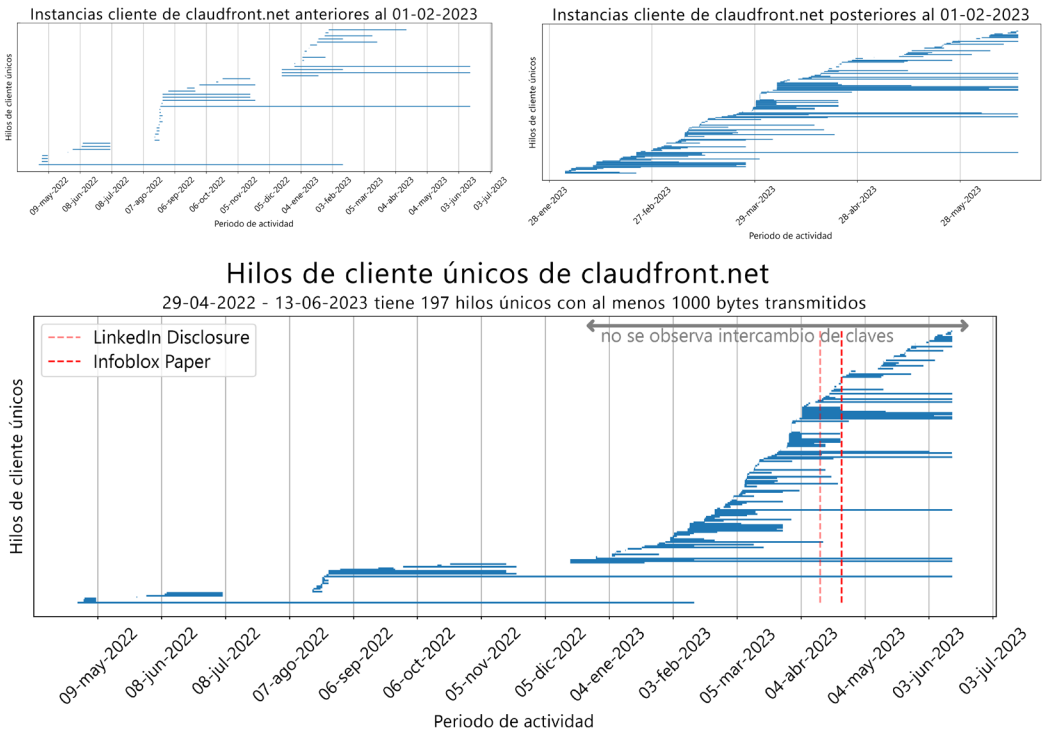

Figura 8. Hilos de clientes para claudfront[.]net basado en hilos de Nonce a lo largo del tiempo. Hay un cambio significativo a principios de febrero de 2023, que se amplía con imágenes separadas que muestran distintos períodos de tiempo.

El tercer dominio, allowlisted[.]net, muestra otra variación en el comportamiento. En este caso, el número de clientes es sistemáticamente pequeño: menos de diez en cualquier momento. A diferencia de claudfront[.]net no hay cambios en febrero de 2023 y no hay ninguna muestra binaria conocida que contenga allowlisted[.]net disponible. No se observan intercambios de claves desde mediados de noviembre de 2022 hasta poco después de nuestra revelación, lo que coincide con el brusco final de la actividad de los clientes y el reinicio de varios hilos en abril de 2023, como se muestra en la figura 9.

Por último, observamos actividad relacionada de hsdps[.]cc, nsdps[.]cc, ads-tm-glb[.]click y j2update[.]cc. Los dominios hsdps[.]cc y ads-tm-glb[.]click dejaron de funcionar tras nuestra publicación en las redes sociales, pero varios de sus clientes fueron transferidos a nsdps[.] cc y j2update[.]cc, respectivamente. Lo descubrimos creando cadenas de nonce en todos los dominios a lo largo del tiempo e identificando los subprocesos que comenzaban a comunicarse con un controlador y terminaban con otro.[16]

15 0375f4b3fe011b35e6575133539441009d015ebecbee78b578c3ed04e0f22568, presentado por primera vez 2023

16 La probabilidad de que esto ocurra al azar con un nonce aleatorio de 32 bits es extremadamente baja, y el número de “transferencias” nonce de un controlador a otro para estos dominios era alto.

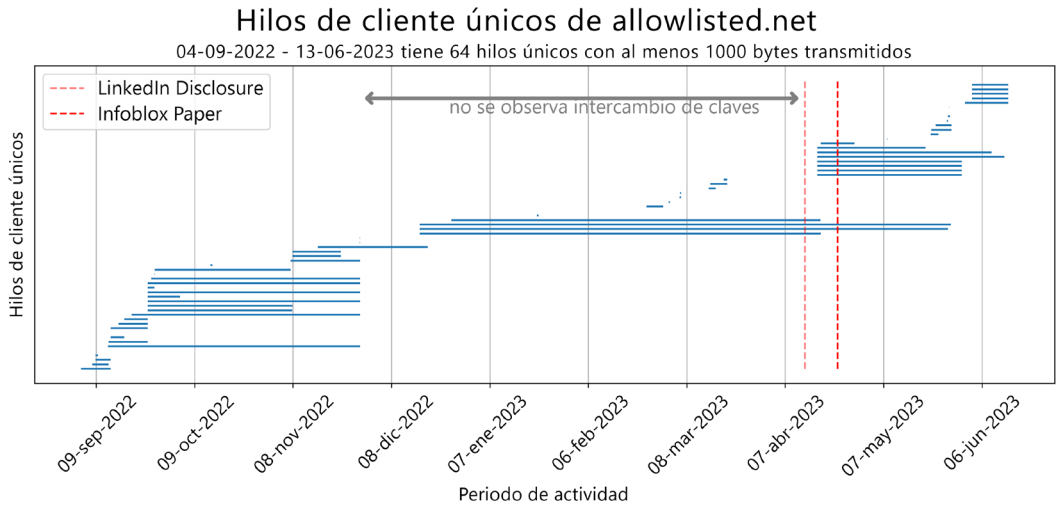

Figura 9. Hilos de cliente para allowlisted[.]net. Hay un número muy reducido de clientes en allowlisted[.]net históricamente y esto no ha cambiado desde la divulgación.

Los nuevos dominios, nsdps[.]cc y j2update[.]cc, se registraron menos de 48 horas después de nuestros anuncios en las redes sociales. Podemos ver en los diagramas de subprocesos de clientes que un conjunto de dominios deja de funcionar mientras otros comienzan. Los programadores comenzaron a comunicarse activamente con los clientes casi inmediatamente después. Tras el descubrimiento de la transferencia de clientes a través del análisis de DNS, encontramos pruebas en muestras binarias de un comando para realizar este cambio, como lo describiremos más adelante.

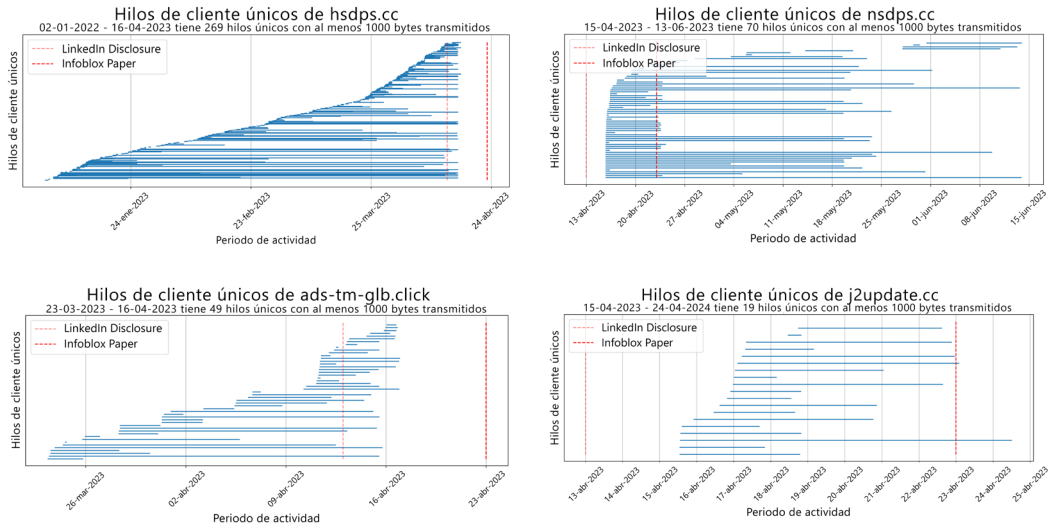

Figura 10. Una comparación cronológica de cuatro dominios de control Decoy Dog. Los controladores hspds[.]cc y ads-tm-glb[.]click cesan las comunicaciones tras la divulgación de Infoblox y los dominios nsdps[.]cc y j2update[.]cc empiezan las comunicaciones. También hemos observado las transferencias de clientes entre estos dominios.

Desde nuestro documento original, hemos visto cómo se activaban otros controladores, cada uno con un número muy reducido de clientes. El comportamiento del cliente que se muestra aquí, junto con la respuesta a nuestro anuncio, indica que varios actores están utilizando el kit de herramientas de Decoy Dog.

02-13 07:39:55 UTC

FIRMAS DE CARGA ÚTIL DE DECOY DOG

Decodificamos las longitudes de carga útil del cliente y del servidor de 15,5 millones de respuestas a consultas observadas en pDNS global durante un período de 13 meses. A continuación, comparamos las firmas de Pupy para las cargas útiles cliente-servidor con los datos observados de Decoy Dog para comprender el comportamiento de los servidores.

Aunque comprobamos que la distribución general del tráfico coincidía con la de Pupy, existían diferencias claras. Los clientes de Decoy Dog utilizan un conjunto mayor de peticiones, o vocabulario, que el que se encuentra en Pupy por defecto.

La Figura 11 muestra las distribuciones relativas de pares de longitud de carga útil en todos los sistemas Decoy Dog. Al usar nuestras firmas de Pupy, como se detalla en el Apéndice D, podemos sacar algunas conclusiones inmediatas:

- Había más de las nueve cargas útiles de clientes esperadas.
- Hubo longitudes de carga útil del servidor que no habíamos observado en nuestro laboratorio.
- La mayoría de las comunicaciones estaban relacionadas con el mantenimiento de sesiones e intercambios de claves.
- Un gran porcentaje de las consultas a los servidores de Decoy Dog recibieron una respuesta de error y mostraron una variación consistente con el escaneado realizado por un tercero en lugar de un verdadero cliente. a mayoría de ellas se produjeron después de nuestros anuncios.

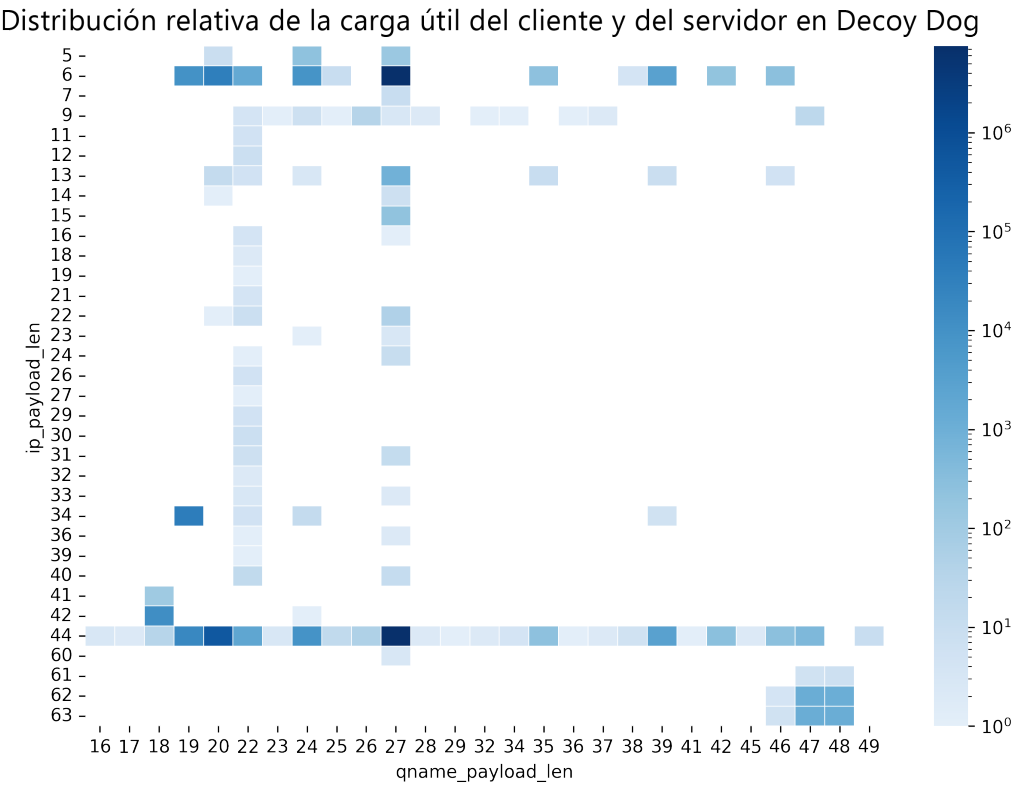

Figura 11. La distribución relativa de las longitudes de carga útil del cliente y del servidor según se observa en las comunicaciones de Decoy Dog.

Las cargas útiles únicas del cliente incluían las longitudes 20, 25, 38, 42 y 46. Algunos de estos pueden estar asociados con una configuración de clave diferente o un cambio en los parámetros de sondeo; no podemos determinar cuál era la comunicación, pero existe la variación. Además, hubo longitudes de carga de respuesta adicionales a las observadas en Pupy. En particular, Decoy Dog tiene una carga útil de servidor de 13 bytes que se observa a lo largo del tiempo en rachas de actividad. No podemos determinar cuál es esta carga útil, pero es consistente con un único comando que requiere 8 bytes de datos para ser transmitidos al cliente. También vimos una serie de respuestas del servidor que contienen una carga útil de 5 bytes, otra longitud no observada en nuestros datos de Pupy e indicativa de un solo comando que no requiere transferencia de datos al cliente. La Figura 12 resume los pares de carga útil únicos que se encuentran en Decoy Dog y no se ven en nuestros experimentos de Pupy.

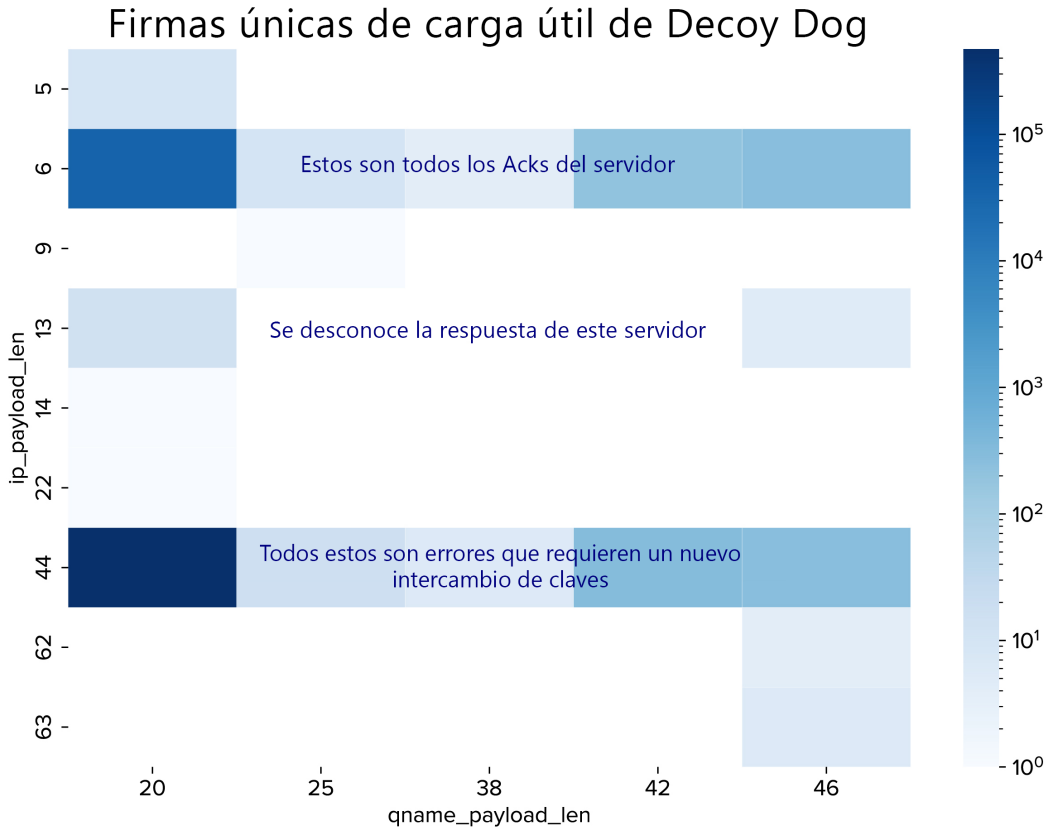

Figura 12. Un resumen de los pares de longitud de carga útil cliente-servidor observados en Decoy Dog y no encontrados en las comunicaciones predeterminadas de Pupy.

También utilizamos series temporales para identificar cambios en las configuraciones predeterminadas. En una sesión de Pupy establecida, el cliente se registrará cada 30 segundos. Mediante un análisis estadístico de la variación de las consultas de los latidos de los clientes, encontramos intervalos de latidos de 2 minutos y 30 minutos, además de los 30 segundos por defecto.

Como resultado de este análisis, pudimos entender la naturaleza de las comunicaciones para cada dominio de Decoy Dog, separando el mantenimiento rutinario de los comandos de acceso remoto. También pudimos aislar las posibles personalizaciones de Pupy utilizadas a través y dentro de subconjuntos de servidores Decoy Dog. Descubrimos que la mayor parte del tráfico de Decoy Dog son acuses de recibo y errores rutinarios, y que las comunicaciones de error eran desproporcionadas con respecto a lo que esperábamos ver basándonos en las observaciones de Pupy. Compartimos los resultados de nuestra investigación sobre este fenómeno de las respuestas de error en la siguiente sección.

COMPORTAMIENTO COMODÍN Y DE GEOFENCING

Informamos en nuestro documento técnico original que los servidores de Decoy Dog respondieron a las consultas de DNS reproducidas. Esto sigue siendo desconcertante. Cuando intentamos comprender cuándo y cómo respondería Decoy Dog a una consulta realizada días o semanas antes, descubrimos un comportamiento aún más sorprendente. Varios de los servidores de Decoy Dog no solo responden a las repeticiones, sino que también responden a cualquier consulta que sea coherente con la codificación de Pupy. En DNS, llamamos a esto una respuesta comodín. Mientras que un servidor Pupy normal devolvería una respuesta NXDOMAIN o SERVFAIL, el servidor Decoy Dog suele devolver 15 direcciones IP.

La siguiente figura 13 muestra respuestas a consultas aleatorias. En este caso, hemos colocado la expresión “comodín” dentro del nombre de la consulta y hemos recibido 15 respuestas en respuesta de dos servidores para perros Decoy diferentes. Las respuestas son diferentes a cada consulta y se ajustan al esquema de codificación de Pupy. A través de nuestra investigación, nos enteramos de que Decoy Dog está gestionando casi todos los errores de esta manera en lugar de devolver las respuestas esperadas NXDOMAIN. Consulte el Apéndice E para obtener información adicional sobre el manejo de errores.

```
; <<>> DiG diggui.com <<>> @ns1.rtupdates.net wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22151
;; flags: qr aa rd; QUERY: 1, ANSWER: 15, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. IN A

;; ANSWER SECTION:
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 64.88.80.242
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 131.163.188.250
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 68.221.203.220
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 198.206.187.196
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 200.37.65.250
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 75.195.241.234
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 141.67.92.44
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 142.153.85.81
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 209.92.80.161
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 147.26.100.52
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 213.83.7.105
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 150.143.51.118
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 153.171.88.194
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 219.226.5.44
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.rtupdates.net. 60 IN A 157.111.237.108

;; Query time: 150 msec
;; SERVER: 5.252.179.232#53(5.252.179.232)
;; WHEN: Sat Jun 03 15:29:11 UTC 2023
;; MSG SIZE  rcvd: 321
```

```
; <<>> DiG diggui.com <<>> @ns1.allowlisted.net wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33023
;; flags: qr aa rd; QUERY: 1, ANSWER: 15, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. IN A

;; ANSWER SECTION:
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 64.88.161.73
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 67.179.145.230
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 69.153.193.38
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 71.14.146.226
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 73.22.176.2
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 138.151.231.153
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 141.232.226.212
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 79.241.118.178
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 209.158.29.150
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 147.248.180.89
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 148.158.234.156
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 215.63.12.236
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 153.141.240.250
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 219.18.219.74
wildcard5yt1j46ra2gwossxhw1q9999.2vzxfwild3999999.allowlisted.net. 60 IN A 156.250.150.9

;; Query time: 151 msec
;; SERVER: 83.166.240.52#53(83.166.240.52)
;; WHEN: Sat Jun 03 15:31:15 UTC 2023
;; MSG SIZE  rcvd: 323
```

Figura 13. Comportamiento de respuesta comodín de dos servidores autorizados de Decoy Dog. En ambos casos, los servidores respondieron con 15 direcciones IP compatibles con la codificación Pupy a la misma consulta aleatoria que contenía la cadena "comodín".

Y lo que es aún más sorprendente, algunos de los servidores Decoy Dog también responden de forma diferente en función de la dirección IP del resolver recursivo que realiza la consulta en nombre del cliente. En la Figura 14, mostramos la repetición de una consulta al dominio Decoy Dog nsdps[.]cc, que se produjo originalmente varias semanas antes. Al realizar la consulta a través de los solucionadores públicos de Yandex, recibimos una respuesta que contiene 15 direcciones IP. También recibimos 15 direcciones IP de los solucionadores públicos rusos TimeWeb. Sin embargo, de los más de treinta solucionadores públicos que probamos, ninguno devolvió una respuesta. Este tipo de comportamiento es coherente con el geofencing, en el que un servidor responde a las consultas de DNS en función de la geolocalización de la dirección IP. Descubrimos este comportamiento en junio de 2023, y comprobamos que algunos de los servidores sólo respondían cuando dirigíamos las consultas DNS a través de direcciones IP rusas, mientras que otros respondían a cualquier consulta bien formulada desde cualquier ubicación. Este tipo de respuesta selectiva garantiza que el controlador solo se comunique con clientes que parezcan estar en Rusia. Sabemos que esta funcionalidad se añadió después de la divulgación porque los controladores habían resuelto previamente consultas de los resolutores recursivos de Infoblox.

```
; <<>> DiG diggui.com <<>> @77.88.8.8 qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42579
;; flags: qr rd ra; QUERY: 1, ANSWER: 15, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. IN A

;; ANSWER SECTION:
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 46 IN A 72.11.125.198
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 36 IN A 203.92.202.218
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 45 IN A 76.74.229.130
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 44 IN A 207.26.86.188
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 31 IN A 80.154.112.164
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 43 IN A 146.160.113.9
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 52 IN A 148.235.159.60
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 41 IN A 151.103.182.130
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 54 IN A 89.76.7.130
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 45 IN A 218.111.60.250
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 42 IN A 93.43.159.18
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 49 IN A 128.88.84.164
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 45 IN A 195.161.207.129
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 51 IN A 68.172.178.156
qnomwvxags5u1pawkzmkllkmuoda9999.abp11i4yk5y1fqyd4tpzm1q9.nsdps.cc. 49 IN A 199.24.240.30

;; Query time: 459 msec
;; SERVER: 77.88.8.8#53(77.88.8.8)
;; WHEN: Tue Jun 20 14:31:11 UTC 2023
;; MSG SIZE  rcvd: 335
```

```
; <<>> DiG diggui.com <<>> @74.82.42.42 hoxlgxq9.yopzgoha3rlp4pdcclosfb63yodq9999.enueh2eluu6uqnjtjpid4lq9.nsdps.cc A
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
```

```
; <<>> DiG diggui.com <<>> @ns2.nsdps.ns2.name hoxlgxq9.yopzgoha3rlp4pdcclosfb63yodq9999.enueh2eluu6uqnjtjpid4lq9.nsdps.c
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
```

Figura 14. Comparación de las respuestas a una consulta de Decoy Dog reproducida de los solucionadores públicos de Yandex, los solucionadores públicos de Hurricane Electric y el solucionador autoritativo. Estas consultas se realizaron sucesivamente a través de un navegador Tor. Solo la consulta a través de Yandex recibió una respuesta.

Cuando se realiza una consulta para un nombre de dominio que no se puede decodificar con la codificación predeterminada de Pupy (agregamos caracteres adicionales para esta prueba), los servidores nsdps[.]cc devuelven una dirección IP que, en esencia, es un sumidero. Como se muestra en la Figura 15 a continuación, modificamos ligeramente la consulta para que no se pueda decodificar correctamente. En este caso, se devolvió una dirección IP aleatoria dentro del rango 172.0.0.0/8. Normalmente, Pupy devolvería una respuesta NXDOMAIN.

```
; <<>> DiG diggui.com <<>> @77.88.8.1 hoxlgxq9.yopzgoha3rlp4pdcclosfb63yodq9999.wildenueh2eluu6uqnjtjpid4lq9.nsdps.cc A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2019
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;hoxlgxq9.yopzgoha3rlp4pdcclosfb63yodq9999.wildenueh2eluu6uqnjtjpid4lq9.nsdps.cc. IN A

;; ANSWER SECTION:
hoxlgxq9.yopzgoha3rlp4pdcclosfb63yodq9999.wildenueh2eluu6uqnjtjpid4lq9.nsdps.cc. 32 IN A 172.67.132.113

;; Query time: 1695 msec
;; SERVER: 77.88.8.1#53(77.88.8.1)
;; WHEN: Tue Jun 20 23:44:46 UTC 2023
;; MSG SIZE  rcvd: 124
```

Figura 15. Una consulta para un nombre de dominio Pupy no válido para el controlador nsdps[.]cc devolverá una dirección IP aleatoria en el intervalo 172.0.0.0/8 en lugar de la respuesta NXDOMAIN esperada.

Parte de este comportamiento puede explicarse como un artefacto de resolución DNS del cliente. Cuando se consulta un host en el DNS, algunos solucionadores intentan resolver nombres de dominio potencialmente relacionados para prepararse para posibles consultas futuras. Por ejemplo, un solucionador recursivo que recibe una consulta para www[.]baddomain[.]com, puede intentar resolver baddomain[.]com además de www[.]baddomain[.]com. Vimos este comportamiento en nuestro propio servidor Pupy al enrutar las consultas de los clientes a través de algunos solucionadores públicos.

RESPUESTAS DE ETIQUETA ÚNICA

Por defecto, Pupy rechaza las peticiones entrantes a etiquetas que no coinciden con la estructura de una comunicación cliente o una consulta ping establecida. Sin embargo, como explicamos en la sección "Gestión de nombres de dominio especial" anterior, la función de solicitud de activación de DNS permite a un actor configurar el servidor de Pupy para que responda a consultas para recursos personalizados. En los registros globales de pDNS, identificamos consultas con un solo subdominio de etiqueta. El único subdominio de este tipo era "m" y nuestra hipótesis era que la resolución de estos dominios era posible a través de la función de activación. Por la naturaleza de la función hash del activador, se debe devolver una única dirección IP estática para estas consultas. Encontramos este comportamiento en 4 dominios: hsdps[.]cc, nsdps[.]cc, j2update[.]cc, y ads-tm-glb[.]click, y es otra característica compartida de este conjunto de dominios que no se ve en ningún otro controlador. Cada uno de ellos devolvió una única dirección IP; sin embargo, en lugar de la dirección IP estática esperada, encontramos 104 direcciones únicas en las respuestas. Esto parece indicar una diferencia en la función de Pupy predeterminado, pero no conocemos el propósito.

ANÁLISIS DE MUESTRAS BINARIO

Después de nuestros descubrimientos de DNS, analizamos las muestras binarias disponibles en VirusTotal para determinar si la fuente de las diferencias con Pupy era evidente. Al analizar las tablas de importaciones y funciones de dos muestras Decoy Dog, identificamos una firma única específica de los implantes Decoy Dog que nos permitió descubrir otras muestras Decoy Dog. La ingeniería inversa de estas muestras confirmó aún más nuestros hallazgos de que Decoy Dog es sustancialmente diferente de Pupy, y que el código más maduro puede haber sido creado por un segundo desarrollador. El cliente se actualiza a Python 3.8 e incluye varios nuevos transportes, cifrado actualizado, comandos personalizados y nueva funcionalidad de DNS. La muestra relacionada con un controlador, claudfront[.]net, contiene características que no se encuentran en los demás. En esta sección se describen algunas de las principales conclusiones y el proceso; en el Apéndice F se ofrecen más detalles técnicos. Los datos analíticos relacionados con los binarios también se añadirán a nuestro repositorio de GitHub.

La primera muestra se subió en septiembre de 2022 y las demás en 2023; tres de ellas tras nuestra divulgación. Extrajimos y comparamos las configuraciones de las diferentes muestras de Decoy Dog, que mostraron que las claves de cifrado difieren entre servidores. Todas las muestras que se comunicaron con cbox4[.]ignorelist[.]com contienen las mismas claves RSA y SSL, lo que indica que la existencia de diferentes muestras no está relacionada con los cambios de clave del servidor. Se puede encontrar una lista completa de las claves descifradas en el repositorio de Github que se detalla en el Apéndice I. El primer certificado SSL de las muestras se generó el 26 de diciembre de 2021 y pertenece a cbox4[.]ignorelist[.]com, el primer controlador observado.

Un descubrimiento significativo fue que Decoy Dog incluye código personalizado en su cliente Pupy que permite a los atacantes enviar y ejecutar módulos Java en tiempo de ejecución inyectándolos en un hilo JVM (Java Virtual Machine). Esta capacidad no existe en las versiones estándar de Pupy. Este código se ha encontrado en todas las muestras de Decoy Dog y es idéntico en todas las instancias. Las funciones binarias restantes en todos los ejemplos de clientes conocidos de Decoy Dog son idénticas a las funciones de los clientes base de Pupy.

La inclusión de módulos Java plantea más preguntas que respuestas. De forma predeterminada, Pupy ya es altamente capaz y admite el uso de módulos Python desde el primer momento. Expandir estas capacidades y escribir módulos de Python es un proceso sencillo que no requiere modificaciones en el lado del servidor ni cambios en el binario del cliente. Se podría crear fácilmente un módulo de Python para ejecutar y ejecutar módulos de Java. Por el contrario, la inyección de módulos Java en tiempo de ejecución sin usar jni.h (o el resto de la API estándar de Java/C) no es una tarea trivial y requiere conocimientos especializados. Por lo tanto, es probable que la adición de estos módulos Java permita a los atacantes dirigirse a sistemas que no ejecutan Python, sistemas que ejecutan una máquina virtual Java privilegiada o no supervisada, o escenarios donde los atacantes intentan evitar dejar pruebas en la máquina no creando archivos.

Los clientes también tienen nuevas funcionalidades, que maduraron con el tiempo. El software cliente se crea marcando un archivo de configuración de Python en un binario determinado. El archivo de configuración incluye los ajustes, todas las claves necesarias para las comunicaciones (RSA, certificados SSL, contraseñas, etc.) y los módulos cliente Python. Los módulos encontrados en las muestras, que son descomprimidos y ejecutados por los dispositivos comprometidos, son muy diferentes del código Pupy disponible públicamente.

La extracción y el análisis de los módulos incrustados dibujan una fascinante historia de desarrollos y cambios personalizados de Decoy Dog. En primer lugar, un número considerable de módulos Pupy han sido simplemente eliminados de Decoy Dog, posiblemente porque los atacantes los consideraron inútiles. En segundo lugar, las muestras similares muestran un gran número de diferencias en los módulos, a veces con capacidades muy diferentes. En tercer lugar, el gran número de cambios y la complejidad añadida por las nuevas funcionalidades muestran un tiempo de desarrollo considerable y un ajuste preciso de los recursos de Pupy. Además, la base de código y los módulos de Pupy fueron portados desde Python 2.7 a Python 3.8, que mejoró la calidad del código, la estabilidad de las operaciones de memoria y la compatibilidad con Windows. Los ejemplos incluyen una versión de cliente que cambia de 3 a 4 con el tiempo; el cliente Pupy más reciente disponible es la versión 2. En la Figura 16 a continuación se muestra un cronograma que resume las fechas de envío en comparación con la madurez del código y las características clave.

Al analizar la naturaleza y el número de módulos modificados, pudimos identificar que, desde una perspectiva de madurez del código, la muestra con el hash ad186df91282cf78394ef3bd60f04d859bcacccbcdcbfb620cc73f19ec0cec64 es el primer binario de Decoy Dog disponible públicamente. Se comunica con el servidor de nombres cbox4[.]ignorrelist[.]com Aunque comparte la mayor cantidad de código con Pupy, esta muestra no se subió a VirusTotal hasta el 27 de abril de 2023, varios días después de que se publicara nuestro artículo. Sin embargo, basándose en el certificado SSL incluido, esta muestra podría remontarse a diciembre de 2021. El desarrollador añadió una función de sondeo específica, una función XOR, nuevos transportes y soporte total para las comunicaciones de red multiproceso. Curiosamente, una serie de módulos nuevos se dirigen específicamente a Win32, aunque todas las muestras hasta ahora son bibliotecas Linux. En este ejecutable, el código responsable de gestionar las comunicaciones DNS es el mismo que el Pupy predeterminado.

Con el paso del tiempo, las muestras que se comunicaban con cbox4[.]ignorelist[.]com se volvieron más complejas. A lo largo de una serie de tres muestras, se fue añadiendo un número creciente de módulos de comunicaciones, incluido un módulo completo para comunicarse utilizando flujos bidireccionales sobre HTTP síncrono (BOSH), así como reescrituras completas de los módulos SSL, TCP y UDP. Los actores detrás de Decoy Dog también agregaron una serie de scripts para portar los módulos de explotación y

comunicación existentes a las plataformas de Windows, reescribieron el cliente picocmd responsable de las comunicaciones DNS e implementaron una serie de mejoras de calidad de vida y estabilidad en el código anterior. Las referencias a Windows en el código apuntan a la existencia de un cliente de Windows actualizado que incluye las nuevas capacidades de Decoy Dog, aunque todos los ejemplos actuales están dirigidos a Linux.

Las versiones posteriores también incluyen un módulo de emergencia que permite que una máquina comprometida se comunique con un servidor DNS de terceros si se impide que el malware se comunique con el servidor C2 durante un período prolongado de tiempo. Este módulo utiliza un DGA para seleccionar dominios para que el cliente los consulte dentro de los servicios DNS dinámicos gratuitos. Estas versiones también permiten arrancar para localizar el controlador C2, el establecimiento de dominios beacon e incorporar consultas CNAME en el servicio de emergencia. Los mecanismos de persistencia extensiva, que se encuentran a partir de la versión 3 del cliente, son capacidades que se asocian con mayor frecuencia a las operaciones de inteligencia que a las llevadas a cabo por actores con motivaciones financieras o equipos rojos.

El código más maduro, que se conecta al controlador claudfront[.]net, incluye dos nuevos comandos denominados AlterDnsCncDomain y CompromisedNode. Como se ha descrito anteriormente, determinamos mediante el análisis de los valores de los clientes que algunos de los actores de Decoy Dog habían hecho la transición de sus clientes a nuevos controladores tras nuestra revelación. Basándonos en el código fuente de Pupy disponible públicamente no vimos cómo esto era posible sin el uso de comandos personalizados. Parece probable que el comando AlterDnsCnCDomain sea la fuente de esas transiciones de cliente y, por tanto, los controladores asociados a nsdps[.]cc pueden estar usando el código más avanzado. La gran desviación de este código con respecto al resto puede indicar que se trataba de un nuevo desarrollador. El código incluye la versión 4 del cliente.

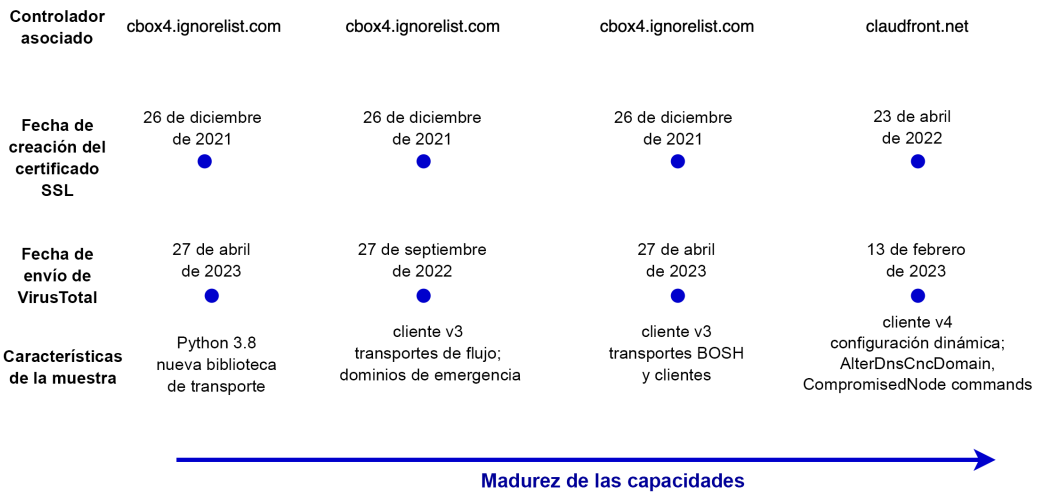

Figura 16. Una cronología de los envíos relacionados con VirusTotal Decoy Dog y la madurez del código.

Vale la pena señalar que, a pesar de todas las mejoras de Decoy Dog, las reglas YARA desarrolladas para versiones más básicas de Pupy aún logran detectar el malware. Sin embargo, no pueden detectar que las muestras se desvían sustancialmente del código y las capacidades conocidas. Esto puede llevar a los investigadores de malware a la falsa suposición de que las muestras de Decoy Dog son solo Pupy básico, ya que ambos tipos de malware están marcados por la misma regla. Por esta razón, hemos incluido una nueva regla YARA para Decoy Dog en el Apéndice G.

COMPARANDO CONTROLADORES

Infoblox está rastreando actualmente 21 dominios de Decoy Dog. Algunos de estos han tenido poca o ninguna actividad observable C2 y no los estamos divulgando en este momento. Algunos controladores cambiaron tras nuestra divulgación inicial en las redes sociales, y el resto cambió después de que publicáramos nuestro primer documento. Todos ellos respondieron cesando las operaciones, trasladando a los clientes a nuevos controladores o modificando el comportamiento "ping" que habíamos descrito en el documento. Algunos incluso agregaron geocercas. Estas respuestas, junto con otros TTP utilizados, nos permiten concluir que hay al menos tres actores que utilizan el kit de herramientas en este momento. En la Tabla 1 a continuación, hemos agrupado un subconjunto de dominios de controladora en función de su comportamiento y características similares.

Grupo de dominios	Características
cbox4.ignorelist[.]com	• primer dominio activo y fuente probable de herramientas de Decoy Dog • desactivado tras la divulgación • uso de DNS dinámico Afraid • intervalo de latido 30 segundos • no geocercado • al menos tres iteraciones distintas del software de cliente • observado por primera vez por nosotros a fines de marzo de 2022, pero es posible que haya estado presente ya en diciembre de 2021. • cliente v2 y v3
claudfront[.]net allowlisted[.]net maxpatrol[.]net atlas-upd[.]com	• segundo conjunto de controladores activos, a partir de mayo de 2022 • operaciones continuas después de la divulgación • registrado con Namecheap • consultas a ping12.<domain> antes de que se viera por primera vez la comunicación cifrada remota • cambió la respuesta ping a una respuesta NODATA • alojamiento IP ruso • intervalo de latidos del corazón de 30 segundos • no geocercado • cliente V3 y V4 • hay algunas diferencias entre allowlisted[.]net y claudfront[.]net que pueden indicar diferentes actores

hsps[.]cc nsdps[.]cc j2update[.]cc ads-tm-glb[.]click	• tercer conjunto de controladores activos, a partir de diciembre de 2022 • movió clientes entre controladores después de la divulgación. • controladores originales aparcados • intervalos de latidos de 2 minutos y 30 minutos • geocercado después de la divulgación • cambió la respuesta ping a una única dirección IP loopback no local • uso de una sola etiqueta de dominio: M • posiblemente cliente v4
rcmsf100[.]net	• observada por primera vez en junio de 2023 • comparte alojamiento con allowlisted[.]net • respuesta ping de NODATA • geocercado

Tabla 1. Una comparación de varios controladores de Decoy Dog.

DECOY DOG EN REDES INFOBLOX

Infoblox ha determinado que nuestros solucionadores los activó un escáner de un proveedor de seguridad que reproducía las consultas de Decoy Dog. Una combinación del comportamiento del escáner y del Decoy Dog creó la señal detectada. El escaneo de Internet se ha convertido en un negocio importante y el escaneo ahora representa una gran cantidad de tráfico de Internet. La interpretan actores legítimos y malintencionados. Un estudio reciente utilizó un telescopio de red oscura para entender el impacto de estos escaneos.[17] Si bien la mayoría de los escaneos se limitan a los escaneos de puertos, que intentan identificar puertos abiertos en todo el espacio IP global, existe una amplia gama de otras actividades de escaneo en el entorno. Por ejemplo, hay escáneres que buscan directorios abiertos y solucionadores de DNS abiertos. Algunas organizaciones documentan completamente su actividad de escaneo, pero muchas no.

El "escaneo agresivo" es una actividad de escaneo no autorizada o de gran volumen que puede degradar el rendimiento de una red. Puede crear una denegación de servicio a una red o, como en el caso de Decoy Dog, crear eventos de seguridad falsos.[18] El escaneo agresivo beneficia al operador, a expensas de las redes cuyos propietarios no están de acuerdo con la actividad. En abril de 2023, los equipos de seguridad de redes con detecciones de Decoy Dog dedicaron importantes recursos a intentar encontrar la causa principal de estas consultas de DNS para asegurarse de que sus sistemas no estaban comprometidos. Estas consultas fueron particularmente alarmantes, ya que se originaron principalmente en firewalls, y el sector de los cortafuegos ha expresado su creciente preocupación por los ataques a los firewalls en los últimos meses.[19]

17 Aggressive Internet Wide Scanners: Network Impact and Longitudinal Characterization, May 2023, Anand, Dainotti, Sippe, Kallitsis. https://arxiv.org/pdf/2305.07193.pdf

18 https://live.paloaltonetworks.com/t5/general-topics/spurious-hits-from-the-expanse-webcrawler/td-p/447239 , consultado por última vez el 11 de junio de 2023

19 https://blog.talosintelligence.com/state-sponsored-campaigns-target-global-network-infrastructure/, consultado por última vez el 11 de junio de 2023

La forma en que las consultas de Decoy Dog llegaron a nuestros solucionadores y por qué provocaron una señal similar a una baliza C2 de malware dirigida es complicada. Para ayudar a los defensores a reconocer actividades similares, proporcionaremos una breve explicación y una ilustración en la Figura 17.

Para que Infoblox reciba consultas DNS de Decoy Dog, una red de cliente debe tener Infoblox como su proveedor de DNS. demás, el cliente debe disponer de dispositivos de seguridad, como cortafuegos, que tengan configurado tanto el filtrado de URL entrantes como el reenvío de DNS desde ese dispositivo a nuestros solucionadores. Estos criterios por sí solos son restrictivos. Cuando se cumplen, se produce la siguiente secuencia:

- El escáner intenta recuperar el contenido del malware C2 directamente desde una dirección IP dentro de la red. Lo hace a pesar de que estas comunicaciones DNS C2 no son contenido web.
- El dispositivo de seguridad intercepta la solicitud e intenta resolver el nombre de dominio.
- La solicitud de DNS se reenvía a Infoblox, que resuelve la consulta y devuelve la respuesta. Si el dominio está en una lista de bloqueo de DNS configurada por el cliente, no devolverá resultados.
- Si el dominio escaneado por el proveedor no es Decoy Dog u otro malware, se resolverá y, dependiendo de las reglas del firewall, el contenido del sitio web se devolverá al escáner.

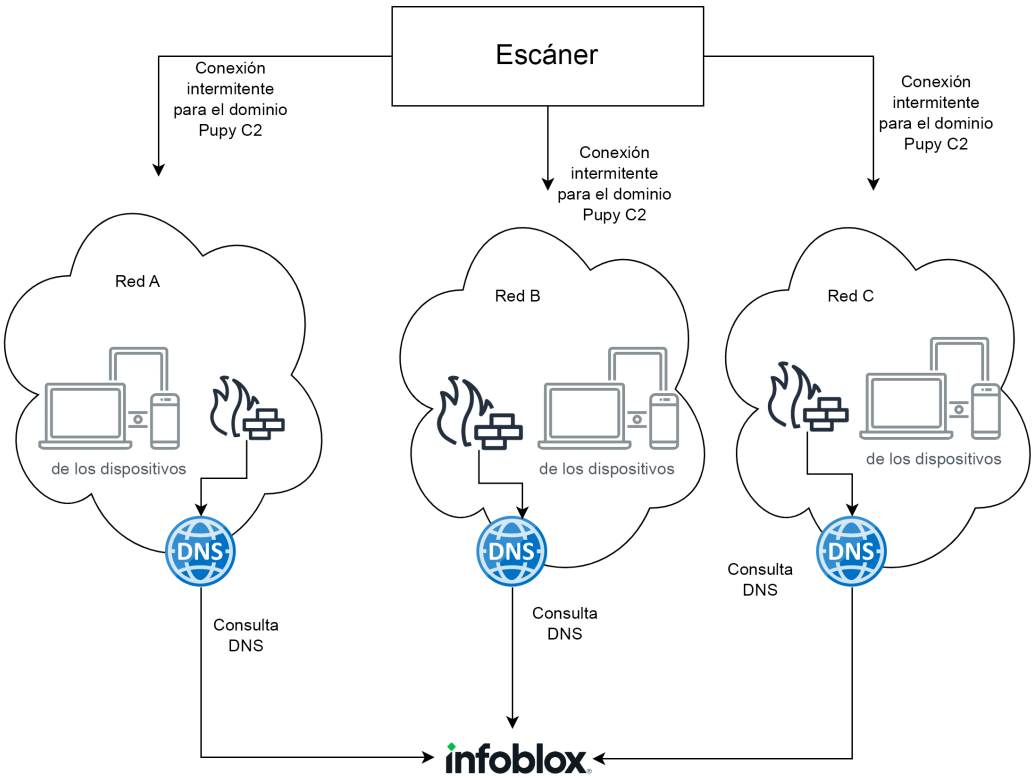

Figura 17. Las consultas de dominios DNS C2 de Decoy Dog se realizaron a los resolutores de Infoblox desde dispositivos dentro de diferentes redes. Los causó un escáner comercial y se activaron de forma intermitente.

Infoblox ha determinado que el proveedor realiza escaneos incluso si la dirección IP no tiene puertos abiertos conocidos y que utilizará puertos raros además de puertos comunes. No sabemos cómo decide el proveedor qué direcciones IP y puertos utilizar. La consecuencia de un escaneo agresivo indiscriminado de esta naturaleza es que dispositivos muy sensibles pueden parecer comprometidos cuando no lo están. Aunque el proveedor parece buscar contenido de forma amplia y constante, Infoblox solo observó

consultas de DNS cuando se cumplieron los criterios anteriores. Como resultado, aunque el número de exploraciones realizadas por el proveedor era muy grande, lo que concuerda con una exploración agresiva, sólo resolvimos un pequeño número de consultas, de forma intermitente a lo largo del tiempo. Este tipo de configuración también introduce la capacidad de un actor para realizar un reconocimiento en ciertas redes; describimos esto en el Apéndice H.

Infoblox Intelligence mantiene registros históricos de toda la actividad de DNS y los utiliza para crear y mantener estadísticas agregadas de la actividad de dominio en nuestras redes y en DNS global. Utilizamos estas agregaciones para identificar una amplia gama de amenazas, incluido el comportamiento anómalo que es coherente con las balizas C2 del malware. En concreto, buscamos dominios cuyas consultas, a lo largo del tiempo, se produzcan en un número anormal de redes de clientes, tengan subdominios coherentes con la exfiltración de datos y que presenten un número bajo de consultas en relación con su comportamiento esperado. Para ello, utilizamos las estadísticas de todos los dominios que hemos observado a lo largo de varios años y billones de consultas DNS.

Una vez descubiertas, las balizas C2 de Decoy Dog y otros programas maliciosos parecen muy sospechosas, pero detectarlas es muy difícil. Por su naturaleza, el tráfico DNS es muy variable y contiene un gran porcentaje de valores atípicos, es decir, dominios que rara vez se ven y tienen una estructura de nombres de dominio coherente con la exfiltración de datos. Sin embargo, la exfiltración y el balizamiento de DNS son muy raros fuera de las actividades de prueba de penetración establecidas. Además, la firma DNS de las pruebas de penetración es muy distinta de las balizas C2 de malware. Aunque Decoy Dog resultó ser DNS C2 de una variante del RAT Pupy, un sistema de gran volumen, parecía ser una baliza de bajo perfil porque el proveedor de seguridad inyectaba el tráfico en las redes.

Aunque las consultas de Decoy Dog a nuestros solucionadores fueron iniciadas por el escáner, se detectaron debido al comportamiento inusual de los servidores de nombres de Decoy Dog. Como revelamos en nuestro artículo anterior, los servidores de nombres de Decoy Dog respondieron a consultas repetidas, aunque a veces intermitentemente. Esto no es coherente con Pupy y otros protocolos de comunicación cifrados. Hemos aprendido que los controladores responden a cualquier consulta bien formada. El comportamiento combinado hizo que nuestros sistemas detectaran una baliza intermitente de bajo volumen. Este tipo de comportamiento de escaneo y reenvío de DNS abierto dentro de una red plantea riesgos de seguridad adicionales para una empresa. Al permitir que una parte externa active consultas DNS desde dentro de una red, un atacante puede realizar un reconocimiento contra una red. Describimos esta vulnerabilidad con más detalle en el Apéndice H.

Conclusión

Decoy Dog es claramente una amenaza grave. Un puñado de actores de amenazas ha estado utilizando el kit de herramientas durante más de un año con las únicas detecciones documentadas resultantes de la supervisión de los datos DNS. Se utiliza en operaciones altamente dirigidas y solo hemos observado que sus programadores interactúan con un número muy limitado de clientes activos. Aunque hemos podido aprender mucho sobre Decoy Dog, seguirá siendo una amenaza grave hasta que se identifiquen y mitiguen las vulnerabilidades utilizadas para establecer su fuga.

Después de nuestra revelación inicial de Decoy Dog, los actores de amenazas respondieron de diversas maneras para garantizar el acceso continuo a los sistemas de las víctimas. Estas respuestas incluían cambiar el comportamiento de respuesta DNS de los controladores, añadir restricciones de geofencing a los controladores y trasladar a los clientes a nuevos controladores. A pesar de estas adaptaciones, Infoblox ha seguido rastreándolas y aprendiendo más sobre Decoy Dog y en qué se diferencia de Pupy RAT.

Los cambios realizados en Pupy para crear Decoy Dog son considerables y son indicativos de un sofisticado actor de amenazas. Estos cambios incluyen:

- Pupy fue escrito en Python 2.7. Decoy Dog requiere Python 3.8 e incluye numerosas mejoras, incluida la compatibilidad con Windows y operaciones de memoria mejoradas.
- Pupy tiene un vocabulario de comunicación muy limitado. Decoy Dog amplía significativamente ese vocabulario mediante la incorporación de múltiples módulos de comunicación nuevos.
- Decoy Dog responde a las repeticiones de consultas de DNS anteriores donde Pupy no.
- Pupy no responde a las solicitudes de DNS comodín, pero Decoy Dog sí. Básicamente, esto duplica el número de resoluciones observadas en el DNS pasivo. De hecho, Decoy Dog responde a solicitudes DNS que no coinciden con la estructura de comunicación válida con un cliente.
- Decoy Dog añade la capacidad de ejecutar código Java arbitrario inyectándolo en un subproceso JVM y añade varios métodos nuevos para mantener la persistencia en el dispositivo de una víctima.

La sofisticación de estos cambios hace que la decisión de Decoy Dog de re-sponder a cualquier consulta bien elaborada sea aún más curiosa. Aunque esta decisión parezca un error a primera vista, es probable que exista al-guna razón aún desconocida para ello. En la actualidad, es sólo otro mis-terio de Decoy Dog.

En el futuro, ya que se investigan aún más estos misterios que rodean a Decoy Dog, los de-fensores deben tener en cuenta lo siguiente:

- Las direcciones IP tanto en Pupy como en Decoy Dog son datos encriptados. No representan direcciones IP reales utilizadas para la comunicación. Cualquier conexión a IP reales asociada con malware es falsa.
- Aunque las IP devueltas en las respuestas del DNS no son significativas, las propias consultas y respuestas del DNS contienen información significativa que se puede utilizar para el seguimiento. Sin embargo, el volumen de comunicación es bajo, lo que significa que se necesita un largo historial de registro para realizar un seguimiento de las comunicaciones detectadas.
- Las respuestas comodín del kit de herramientas combinadas con un escaneo agresivo por parte del proveedor de seguridad pueden dar la apariencia de compromiso donde no lo hay.
- Hay disponible una regla YARA que puede detectar el cliente Decoy Dog en una máquina víctima. Es capaz de diferenciar a Decoy Dog de la versión pública de Pupy.

Decoy Dog se detectó únicamente utilizando algoritmos de detección de amenazas de DNS. Hasta la fecha, no hay divulgación pública que describa las detecciones del propio malware y el alcance completo de sus capacidades aún no se conoce. El hecho de que haya operado sin ser detectado durante tanto tiempo pone de manifiesto una debilidad que se produce cuando la industria confía excesivamente en la detección basada en malware. La detección y respuesta DNS es actualmente la única forma de defenderse contra Decoy Dog y puede ser la mejor opción incluso después de que las vulnerabilidades de las víctimas y el propio Decoy Dog se comprendan completamente.

Indicadores

Los indicadores Decoy Dog relacionados con los controladores y los ejemplos descritos en este informe aparecen a continuación y están disponibles en nuestro repositorio abierto de Github.[20]

20 https://github.com/infobloxopen/threat-intelligence/tree/main/cta_indicators

Grupo de dominios	Características
ads-tm-glb[.]click	Decoy Dog C2 domain
allowlisted[.]net	Decoy Dog C2 domain
atlas-upd[.]com	Decoy Dog C2 domain
cbox4[.]ignorelist[.]com	Decoy Dog C2 domain
claudfront[.]net	Decoy Dog C2 domain
hsdps[.]cc	Decoy Dog C2 domain
j2update[.]cc	Decoy Dog C2 domain
maxpatrol[.]net	Decoy Dog C2 domain
nsdps[.]cc	Decoy Dog C2 domain
rcmsf100[.]net	Decoy Dog C2 domain
13[.]248[.]169[.]48	IP del servidor de nombres De-coy Dog C2
156[.]154[.]132[.]200	IP del servidor de nombres De-coy Dog C2
194[.]31[.]55[.]85	IP del servidor de nombres De-coy Dog C2
5[.]199[.]173[.]4	IP del servidor de nombres De-coy Dog C2
5[.]252[.]176[.]63	IP del servidor de nombres De-coy Dog C2
5[.]252[.]176[.]22	IP del servidor de nombres De-coy Dog C2
5[.]252[.]179[.]18	IP del servidor de nombres De-coy Dog C2
67[.]220[.]81[.]190	IP del servidor de nombres De-coy Dog C2
69[.]65[.]50[.]194	IP del servidor de nombres De-coy Dog C2
69[.]65[.]50[.]223	IP del servidor de nombres De-coy Dog C2
70[.]39[.]97[.]253	IP del servidor de nombres De-coy Dog C2
83[.]166[.]240[.]52	IP del servidor de nombres De-coy Dog C2

4996180b2fa1045aab5d36f46983e91dadeebf d4f765d69fa50eba4edf310acf	Decoy Dog binario SHA256
ab8e333ef9bc5c5a7d1ed4cab08335861e150 b0639d3d0ca4c30b7def5cdccde	Decoy Dog binario SHA256
ad186df91282cf78394ef3bd60f04d859bcaccc bcdcbfb620cc73f19ec0cec64	Decoy Dog binario SHA256
6c8f413111f1abfee788dad4ee7cca37e0c259 7cca66d155af958c535faf55cc	Decoy Dog binario SHA256
0375f4b3fe011b35e65751335394410 09d015 ebecbee78b578c3ed04e0f22568	Decoy Dog binario SHA256
6c8f413111f1abfee788dad4ee7cca37e0c259 7cca66d155af958c535faf55cc	Decoy Dog binario SHA256
t1fde0f101c9395f39ecd16430b41041a59107 c73c904087309fb8d0e8d87e0077129f3f	Firma de Decoy Dog Telfhash[21]

APÉNDICE A: PROCESAMIENTO DE COMANDOS DE CLIENTE

La figura 18 ilustra el ciclo operativo del cliente descrito en el documento. El cliente pasa repetidamente de dormir a sondear al servidor y responder a comandos.

21 https://github.com/trendmicro/telfhash

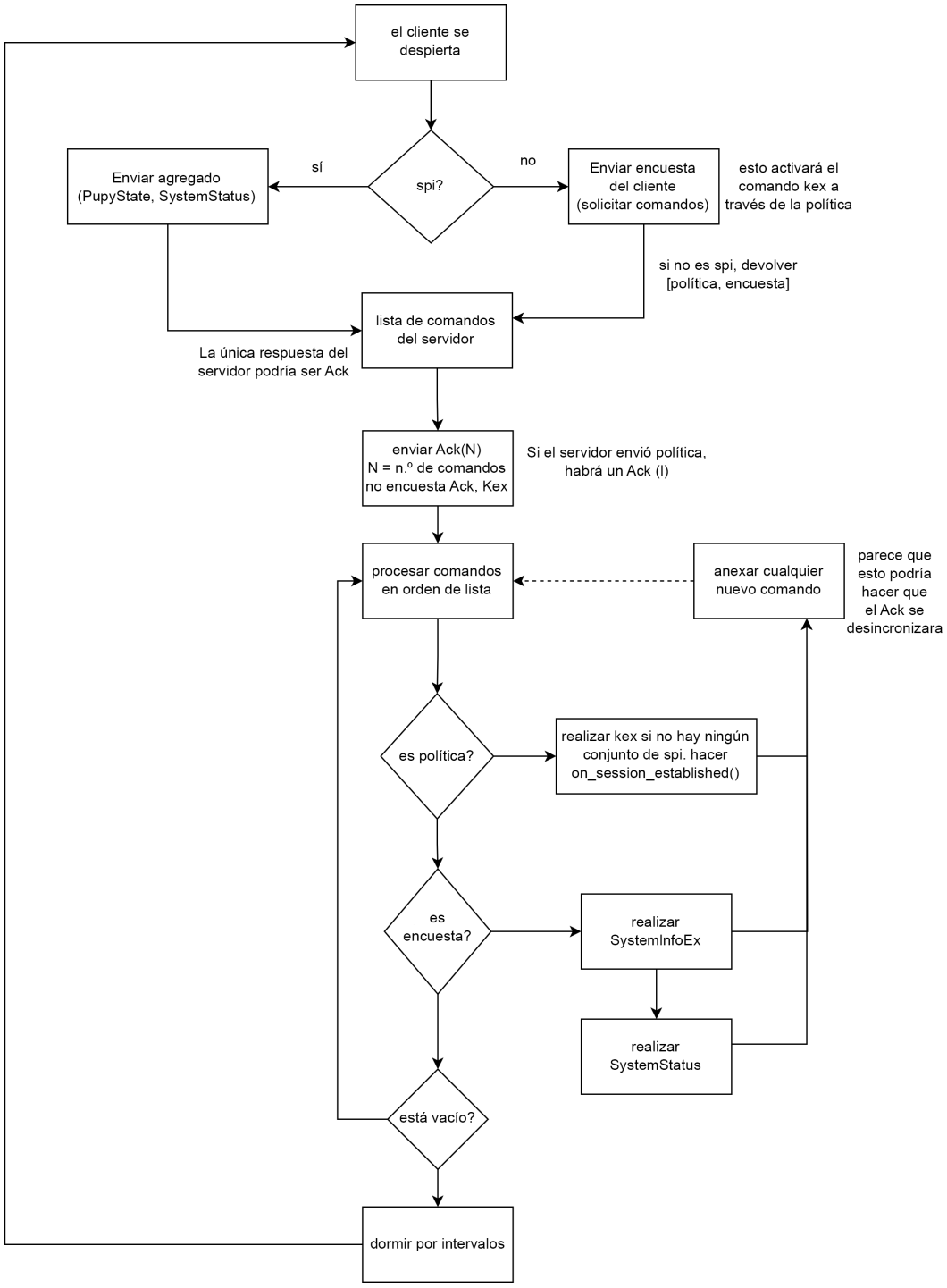

Figura 18. Flujo de trabajo del cliente.

APÉNDICE B: ESTRUCTURA DE LA CARGA ÚTIL DE COMUNICACIONES

La estructura de la carga cifrada para el cliente y el servidor es idéntica, pero existen diferencias en su procesamiento. En concreto, el cliente incluye 13 bytes de información de cliente en cada consulta junto con la carga útil de datos, como se ha descrito anteriormente.

Tanto el cliente como el servidor utilizan el término comando para el tipo de información que transmiten al receptor. Por lo tanto, cuando el cliente se pone en contacto con el servidor al despertarse, se considera un comando cliente. Los comandos se registran para que el cliente o el servidor puedan aplicar un procesamiento específico a los datos. Puede haber más de un comando en una misma comunicación, aunque desde el cliente esto es poco frecuente.

La carga útil que se envía para codificación y transmisión tiene la siguiente forma:

- una suma de comprobación de 4 bytes,
- Paquetes de comandos concatenados, que contienen una identificación de comando de 1 byte y una parte de datos variable dependiente del comando.

La longitud total de la carga útil no puede superar los 52 bytes.

APÉNDICE C: RECONSTRUCCIÓN DE CLIENTES A PARTIR DE DATOS PASIVOS

Como se ha descrito antes, las consultas de Pupy incluyen datos cifrados y dos valores codificados, ¡el nonce y SPI, que proporcionan cierta seguridad y permiten al servidor ordenar las comunicaciones del cliente. El valor SPI se usa específicamente para identificar una sesión en curso dentro del servidor y está presente en las consultas después de un intercambio de claves correcto. Como resultado, las consultas que contengan el mismo SPI y que se produzcan cerca en el tiempo tienen casi garantizado que proceden del mismo cliente. Por otro lado, un solo cliente tendrá muchas sesiones y muchos valores de SPI a lo largo del tiempo, por lo que el SPI por sí solo no puede distinguir a los clientes. En cambio, utilizamos los valores nonce para separar las comunicaciones con los clientes.

Cuando el cliente se inicializa, genera aleatoriamente un valor nonce de 32 bits que sirve como punto de partida. Con cada paquete, este nonce se incrementa en la longitud de los datos que se transmiten. El servidor utiliza el nonce como una comprobación de seguridad menor, asegurándose de que aumenta con cada consulta recibida, pero su uso principal es descifrar e interpretar correctamente la comunicación subyacente. A partir de una serie de consultas de Pupy observadas, podemos decodificar estos valores de nonce y calcular el siguiente nonce de la serie, como se muestra en la Figura 19 a continuación.

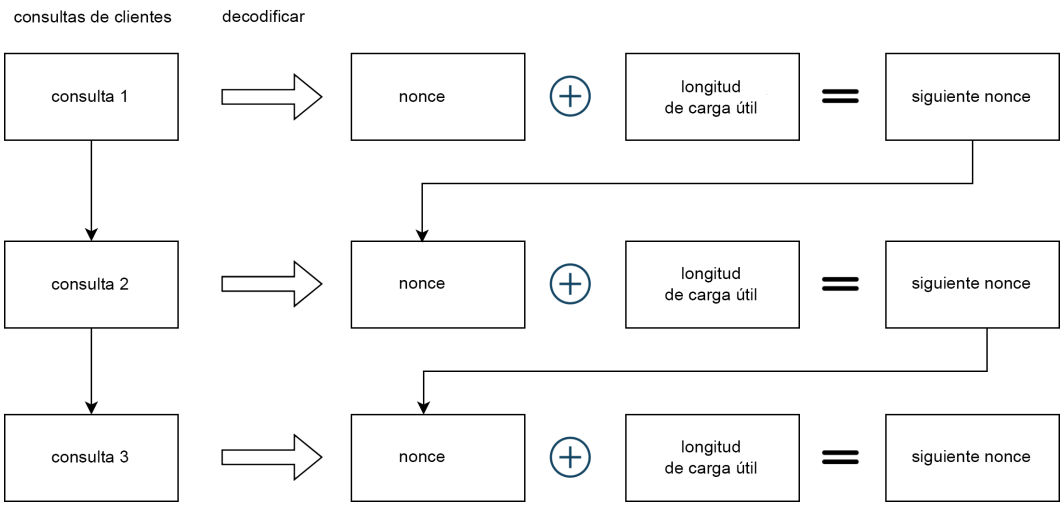

Figura 19. Relación de valores nonce dentro de una serie de consultas de Pupy.

Como resultado, podemos ordenar consultas de un solo cliente y confirmar que una serie de consultas pertenecen a un solo cliente. En la recopilación pasiva de una implementación de Pupy, las consultas pueden originarse en muchos clientes y superponerse en el tiempo. Sin

embargo, aún podemos separar estas observaciones en actividades de clientes separadas con un alto grado de confianza debido a la construcción del nonce. Debido a que el nonce se utiliza para cifrar la carga útil, el desarrollador utilizó un potente generador de números aleatorios para crearlo. Esto garantiza que cada cliente generará valores nonce iniciales únicos.[22] El nonce se recrea cada vez que el cliente se reinicia.

La seguridad adicional para el cifrado también proporciona un mecanismo para distinguir a los clientes en observaciones agregadas. Para hacer esto, calculamos tanto el nonce codificado como el siguiente valor del nonce para cada consulta. A continuación, encadenamos las consultas utilizando los valores nonce secuenciales, como se muestra en la figura 20 a continuación. Aunque los datos subyacentes permanecen cifrados, podemos estimar el número de clientes y realizar observaciones sobre la duración de su actividad. Además, podemos inferir información sobre la comunicación misma utilizando las longitudes de carga útil y comparando series temporales entre clientes.

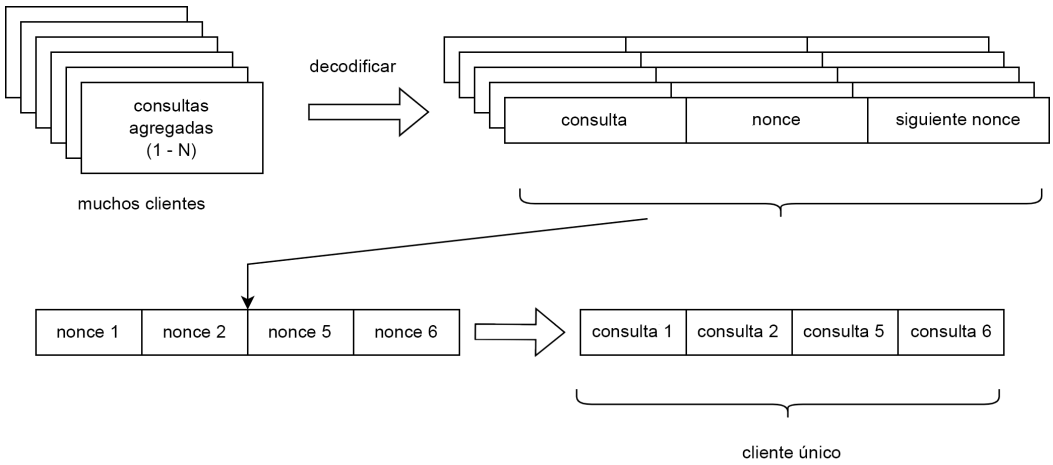

Figura 20. Separación de un subproceso de cliente de consultas de un conjunto agregado de observaciones utilizando los valores nonce.

Este tipo de explotación tiene dos desafíos: los cambios en la resolución de DNS del cliente infectado y la caída de paquetes. PPor defecto, Pupy utiliza el solucionador DNS predeterminado del cliente y la eleccion del solucionador puede no estar bajo el control del actor. Si el cliente está en itinerancia, puede que utilice diferentes solucionadores recursivos según el entorno local. En las redes empresariales, pueden utilizar la infraestructura DNS de proveedores como Infoblox, en la que las consultas de DNS se imponen a los solucionadores recursivos empresariales independientemente de la configuración del cliente.[23] Además, cuando el DNS se transporta por UDP, la pérdida de paquetes es inevitable. El resultado es que es poco probable que observemos todas las consultas solo en el DNS pasivo, lo que crea huecos en la cadena nonce recuperada que podrían tener un tamaño significativo.

Sin embargo, aún podemos reconstruir los hilos de clientes aprovechando el hecho de que el nonce es un valor generado aleatoriamente. El desarrollador utilizó un generador de números fuertes que asegura que los clientes independientes de Pupy tienen muy pocas probabilidades de compartir un valor nonce. Además, dado que solo se pueden transmitir 52 bytes de datos a la vez, y el valor nonce se incrementa por la carga útil, es poco probable que se superpongan dos cadenas nonce generadas de forma independiente. Como

22 Existen probabilidades raras de que dos clientes diferentes puedan generar el mismo nonce al mismo tiempo.

23 Who is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path, Baojun Liu, et al. 2018, https://www.usenix.org/conference/usenixsecurity18/presentation/liu-baojun

resultado, los clientes se pueden separar ordenando valores nonce y agrupando aquellos que son estadísticamente similares. Un solo cliente tiene un único nonce a la vez, lo que nos permite estimar el número de clientes activos en un momento dado. Como mostramos en el cuerpo principal del papel, descubrimos que esta técnica es muy eficaz para recuperar las cadenas de consulta de clientes de Decoy Dog.

APÉNDICE D: FIRMAS DE CARGA ÚTIL

Las tablas de esta sección incluyen las longitudes de carga útil para comandos específicos que se observan comúnmente en las comunicaciones de Pupy. En particular, proporciona la longitud de carga útil cifrada para cada comando de cliente estándar. Las cargas útiles del servidor son más flexibles que las de los clientes; a continuación se muestran las más comunes.

Comando del cliente	Longitud de la carga útil
Registro del cliente (inicial)	18
Ack	19
Registro de clientes (variante rara)	22
Estado del sistema	24
Estado en línea	27
Registro del cliente (en sesión)	27
Cuestionario de puerto	35
Información del sistema ampliada	39
Intercambio de claves	47, 48

Tabla 2. Comandos de cliente y longitudes de carga útil.

Comando del servidor	Longitud de carga
Ack	6
Necesita sesión: política, encuesta	42
Sesión incompleta: ack, política	34
Error: mensaje, política, encuesta	44
Necesita información del sistema: encuesta	15
Intercambio de claves	62, 63
Salir	7

Tabla 3. Longitud de carga útil y comandos comunes del servidor.

APÉNDICE E: MANEJO DE ERRORES

Pupy contiene un manejo personalizado para una variedad de errores que el servidor puede encontrar. Un dominio que no se descodifique correctamente o se reproduzca dará como resultado una respuesta NXDOMAIN del servidor. El siguiente fragmento de código muestra el procesamiento de consultas del servidor. Si no se devuelve ninguna respuesta, devolverá una respuesta NXDOMAIN.

```
answers = self.process(qtype, qname.stripSuffix(self.domain).idna()[:-1])
klass = SUPPORTED_METHODS[qtype]


if answers:
    for answer in answers:
        reply.add_answer(RR(qname, qtype, rdata=klass(answer), ttl=600))


    if self.edns:
        reply.add_ar(EDNS0(udp_len=512))
else:
    reply.header.rcode = RCODE.NXDOMAIN
```

Figura 21. Código fuente del servidor Pupy que procesa las consultas de los clientes.

En Decoy Dog, muchas consultas de clientes que deberían dar como resultado un NXDOMAIN del servidor devuelven en su lugar una respuesta, normalmente 15 direcciones IP. Esto parece deberse a un cambio en el código, mientras que Decoy Dog responde a una gran variedad de posibles errores con un DnsCommandServerException internamente. DnsCommandServerException dará como resultado una respuesta al cliente, especificando el tipo de error encontrado e indicando al cliente que realice un nuevo intercambio de claves seguido de la transmisión de información del sistema. A continuación se muestra el bloque de código para este control de errores.

```
except DnsCommandServerException as e:
    nonce = e.nonce
    version = e.version
    responses = [e.error, Policy(self.interval, self.kex), Poll()]
    emsg = 'Server Error: {} (v={})'.format(e, version)
    logger.debug(emsg)
    if node:
        node.warning = emsg
```

Figura 22. Código fuente del servidor Pupy que devuelve un error al cliente.

En las comunicaciones normales entre un servidor Pupy y un cliente, este tipo de excepción se generará cuando no haya una sesión activa para un cliente conocido. También se utiliza cuando la carga útil del cliente no es válida o tiene una suma de comprobación incorrecta. En todos los demás casos, el resultado es un NXDOMAIN.

APÉNDICE F: ANÁLISIS DE MUESTRAS BINARIAS

Binarios del clientes Pupy

Cuando el servidor Pupy se configura por primera vez, compila los archivos de la biblioteca Pupy y crea un archivo de plantilla estático para cada arquitectura. Estos archivos de plantilla se comprimen, están muy ofuscados y se eliminan todos los símbolos.

Los binarios cliente pueden entonces crearse manualmente utilizando pupygen.py en el servidor. El script crea archivos binarios específicos de C2 mediante la cálculo de referencias de bytes de configuración específicos (host remoto, tipo de transporte, indicador de depuración, etc.) en la plantilla estática correspondiente a la arquitectura de destino y al tipo de archivo.

Los binarios del cliente Pupy ofrecen una variedad de funcionalidades avanzadas y pueden dirigirse a prácticamente todas las plataformas, incluidas Windows, macOS, Linux, Solaris y Android. En concreto, son capaces de permanecer residentes en memoria, interactuar con el servidor, ofrecer capacidades completas de shell inverso, crear copias sin archivos, etc. Cuando el binario se ejecuta, crea copias de sí mismo en la memoria para evitar ser detectado y ser más resistente a las técnicas de destrucción de procesos.

Ejemplo de función de inyección de Java

Los binarios de Decoy Dog incluyen una serie de nuevas funciones relacionadas con la inyección de Java. Este es un ejemplo de una de esas funciones.

```
undefined8 FUN_00105903(void)
{
  int iVar1;
  long lVar2;
  long lVar3;
  long lVar4;
  undefined8 uVar5;
  char *pcVar6;
  undefined local_20 [8];
  undefined8 local_18;

  local_18 = 0;
  if (DAT_005fbda0 == 0) {
     pcVar6 = "JVM was not loaded yet";
  }
  else {
       jvm_address = check_jvm_is_running(0);

       if (jvm_address == 0) {
       return 0;
       }
       classloader_address = find_classloader(lVar2);
       if (classloader_address == 0) {
       pcVar6 = "Preferred classloader was not found";
       }
       else {
       thread_class_address = find_jv_thread(lVar2);
       if (thread_class_address == 0) {
       pcVar6 = "Could not find Thread class";
       }
       else {
       iVar1 =
inject_in_thread(jvm_address,thread_class_address,"currentThread","()Ljava/lang/Thread;",&lo
cal_18);
       if (iVar1 == 0) {
       iVar1 = inject_in_class(jvm_address,local_18,"setContextClassLoader","(Ljava/lang/ClassLoader;)V",
                                          local_20,classloader_address);
       if (iVar1 == 0) {
          uVar5 = (*DAT_005fb748)(1);
          return uVar5;
       }
      pcVar6 = "Iteration failed";
      }
      else {
      pcVar6 = "Could not find current JVM Thread";
      }
  return 0;
```

Figura 23. Función Decoy Dog parcialmente desensamblada, intentando encontrar el hilo JVM en ejecución actual para la inyección.

APÉNDICE G: REGLA YARA PARA DECOY DOG

La siguiente regla de YARA se puede utilizar para detectar las muestras de Decoy Dog que hemos observado a partir de julio de 2023.

```
/*
This rule only detects Decoy Dog. It was adapted from Florian Roth's Pupy Rule
original author : Florian Roth / @neo23x0
original link : https://github.com/Neo23x0/signature-base/blob/master/yara/gen_pupy_rat.yar
*/

/* Rule Set ----------------------------------------------------------------- */
import "elf"
import "pe"

rule DecoyDog_Backdoor {
   meta:
     description = "Detects Decoy Dog backdoor"
     license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-
base/blob/master/LICENSE"
     author = "Infoblox Inc."
     reference = "https://github.com/n1nj4sec/pupy-binaries"
     date = "2023-07-11"

   strings:
     $x1 = "reflectively inject a dll into a process." fullword ascii
     $x2 = "ld_preload_inject_dll(cmdline, dll_buffer, hook_exit) -> pid" fullword ascii
     $x3 = "LD_PRELOAD=%s HOOK_EXIT=%d CLEANUP=%d exec %s 1>/dev/null 2>/dev/null" fullword ascii
     $x4 = "reflective_inject_dll" fullword ascii
     $x5 = "ld_preload_inject_dll" fullword ascii
     $x6 = "get_pupy_config() -> string" fullword ascii
     $x7 = "[INJECT] inject_dll. OpenProcess failed." fullword ascii
     $x8 = "reflective_inject_dll" fullword ascii
     $x9 = "reflective_inject_dll(pid, dll_buffer, isRemoteProcess64bits)" fullword ascii
     $x10 = "linux_inject_main" fullword ascii
     $x11 = "jvm.PreferredClassLoader" fullword ascii
     $x12 = "jvm.JNIEnv capsule is invalid" fullword ascii

   condition:
     (3 of them and $x11 ) or (3 of them and $x12)
     or (uint16(0) == 0x5a4d and pe.imphash() == "84a69bce2ff6d9f866b7ae63bd70b163" and
$x11) or (elf.telfhash() ==
"t1fde0f101c9395f39ecd16430b41041a59107c73c904087309fb8d0e8d87e0077129f3f")
}
```

Figura 24. Regla YARA para detectar muestras de Decoy Dog.

APÉNDICE H: VULNERABILIDADES DE SEGURIDAD EXPUESTAS

Cuando un dispositivo está configurado para realizar una consulta DNS en una conexión entrante, permiten a una entidad externa controlar parcialmente su comportamiento y recursos.[24] En particular, esta configuración puede proporcionar a los actores de amenazas un medio para el reconocimiento, la resolución abierta y posible participación en un ataque de denegación de servicio. Dado que el DNS es complejo, es posible que tanto los proveedores como los operadores de red no comprendan estos riesgos. Si bien los dispositivos de seguridad que transmitieron las consultas que detectamos estaban destinados a tener características novedosas, el uso de DNS en esas funciones expone la red al reconocimiento y potencialmente a otras amenazas.

Un dispositivo dentro de una red que atiende consultas DNS a cualquier entidad externa se conoce como solucionador abierto. En algunos casos, un dispositivo puede devolver respuestas pero no resolver del todo las consultas de DNS externas debido a una amplia

24 https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000PLRaCAO, consultado por última vez el 11 de junio de 2023

gama de circunstancias. En cualquier caso, estos dispositivos representan un riesgo para la propia red y para el uso de la red para amplificar ataques de denegación de servicio distribuida (DDOS). Los riesgos de los resolvedores DNS abiertos están bien documentados y muchos contratos de servicios, incluidos los de Infoblox, prohíben los resolvedores abiertos debido a estos riesgos.

En el caso de las consultas de Decoy Dog, los dispositivos de seguridad no estaban abiertos, pero permitieron a una parte externa activar consultas de DNS. Este tipo de configuración no puede ser utilizado para un ataque de amplificación, pero puede ser utilizado por un actor de amenazas para otros fines. Por ejemplo, un actor de amenazas puede realizar el reconocimiento contra una red; se muestra en la figura de abajo. El actor crea un dominio y configura el servidor de nombres correspondiente para registrar las consultas entrantes. A continuación, el actor utiliza un mecanismo de escaneo para enviar nombres de dominio personalizados para conectarse a la red. En el caso de una búsqueda de resolución abierta, estas pueden ser consultas de DNS. En el caso de Decoy Dog, eran conexiones HTTPS. En cualquier caso, el dispositivo interno genera una consulta DNS que se envía al servidor de nombres controlado por actores. A continuación, el actor puede vincular el nombre de dominio y la dirección IP original a la consulta que recibió. Si bien este tipo de ataques obtienen una cantidad limitada de información en cada intento, son mecanismos bien establecidos para mapear redes internas para ataques posteriores.

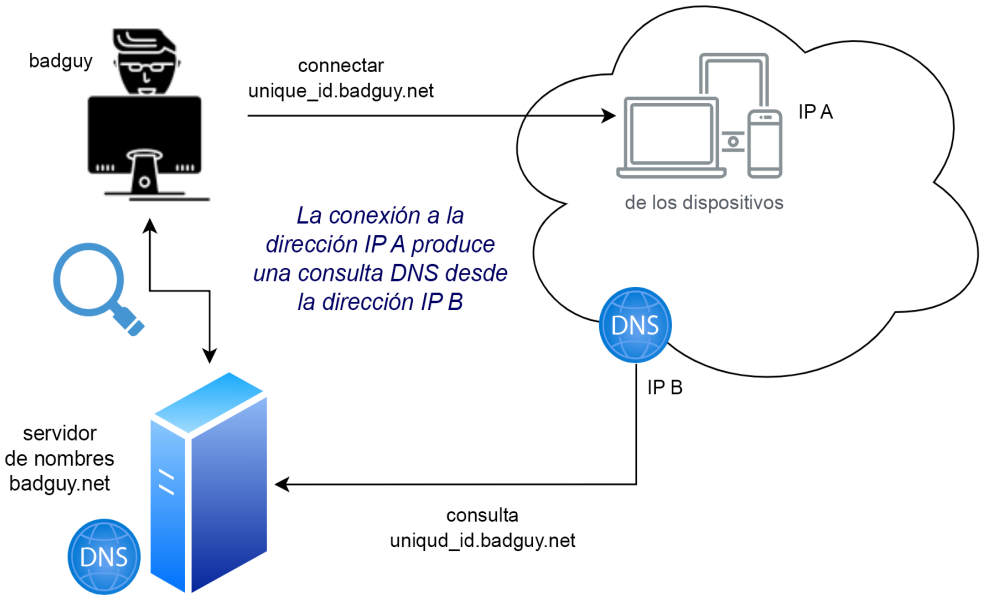

Figura 25. Un actor realiza un reconocimiento en una red mediante la creación de nombres de dominio únicos que crean consultas DNS a su servidor de nombres.

APÉNDICE I: DATOS DE LA INVESTIGACIÓN

Para nuestra investigación, establecimos un servidor Pupy y enrutamos las comunicaciones entre el servidor y los clientes a través de nuestros resolutores recursivos. Recopilamos esos registros de consultas DNS para nuestro análisis y estamos haciendo que los registros estén disponibles para su investigación. Los datos cubren varios días de actividad variable. La mayoría de las veces, controlábamos a los clientes estableciendo un proxy inverso y los comandos se enviaban a través de SSL. Sospechamos que este también es el caso de Decoy Dog. Sin embargo, ejercemos todos los comandos disponibles a través de respuestas DNS del servidor.

Además, hay períodos en los que varios clientes están activos simultáneamente y numerosos reinicios de clientes. El ámbito de la actividad incluida debería permitir recrear los resultados aquí descritos.

Los datos están disponibles en nuestro repositorio público de GitHub infobloxopen: threat-intelligence.[25] Los registros de consulta-respuesta contienen resultados de registro A y se empaquetan en un archivo csv que contiene los siguientes campos:

- marca de tiempo: el tiempo de la consulta en segundos de época Unix
- consulta: el nombre de dominio completo transmitido en la consulta del cliente
- respuesta: conjunto de direcciones IP devueltas por el servidor
- client_payload_len: número de bytes de carga útil dentro de la consulta, incluida la información del host
- server_payload_len: el número de bytes de carga dentro de la respuesta

El repositorio también incluye los indicadores en este documento; hay indicadores adicionales disponibles para los defensores previa solicitud como información TLP:RED. Además, proporcionamos datos que resultaron de muestras binarias de ingeniería inversa disponibles en VirusTotal. Esto incluye:

- Parámetros de configuración integrados para cada muestra
- Claves criptográficas incrustadas y contraseña para cada muestr
  - » BIND_PAYLOADS_PASSWORD
  - » DCONFIG_PUBLIC_KEY (only for client v4)
  - » DNSCNC_PUB_KEY_V2
  - » ECPV_RC4_PRIVATE_KEY
  - » ECPV_RC4_PUBLIC_KEY
  - » SCRAMBLESUIT_PASSWD
  - » SIMPLE_RSA_PUB_KEY
  - » SIMPLE_RSA_PRIV_KEY
  - » SSL_BIND_CERT
  - » SSL_BIND_KEY
  - » SSL_CA_CERT
  - » SSL_CLIENT_CERT
  - » SSL_CLIENT_KEY
- Una regla de la YARA y un hash de TELF que pueden detectar los binarios de Decoy Dog

25 https://github.com/infobloxopen/threat-intelligence