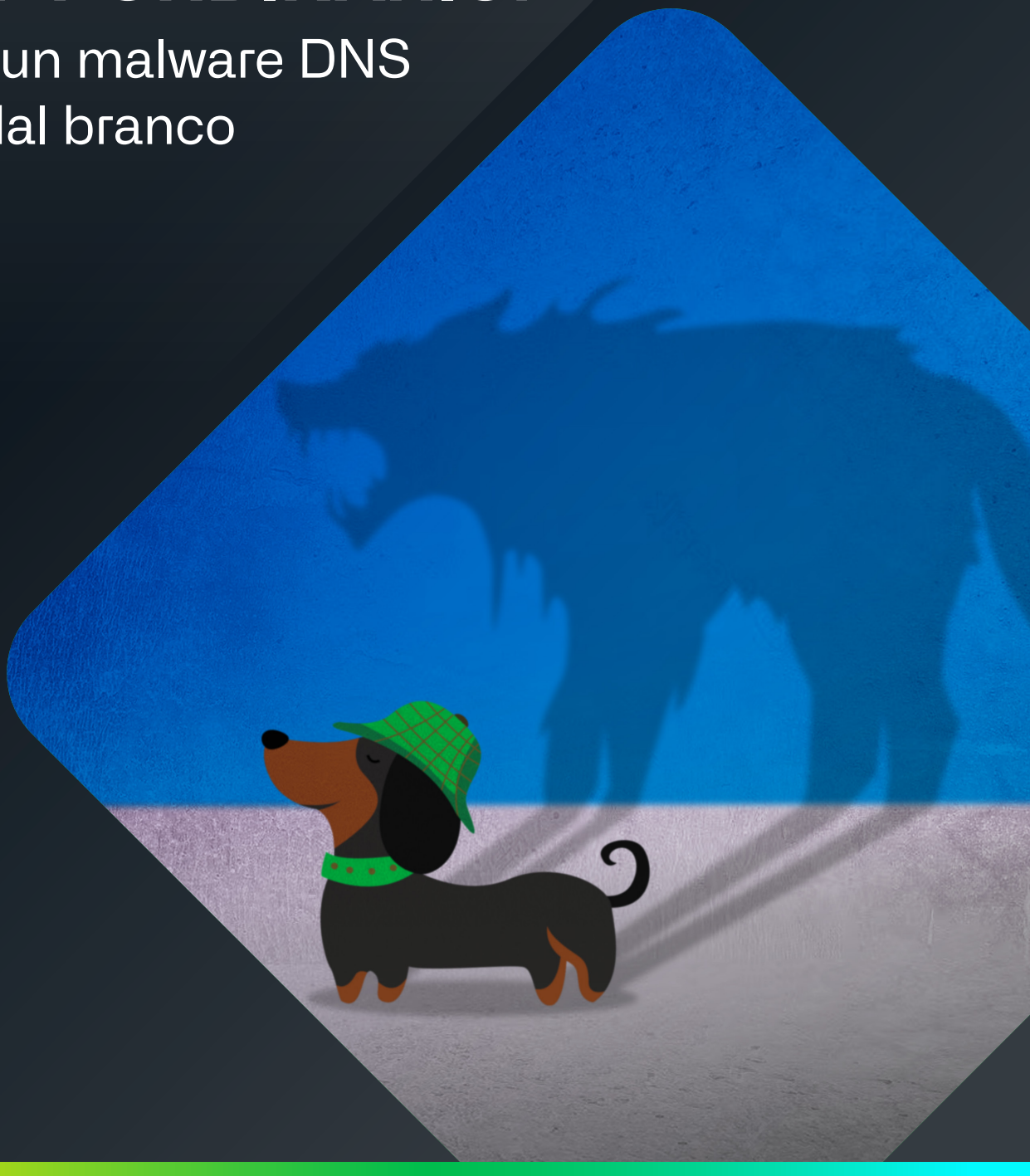
infoblox

DECOY DOG NON È UN PUPY ORDINARIO:

Separare un malware DNS subdolo dal branco



INDICE DEI CONTENUTI

Sommario esecutivo

Decoy Dog è un toolkit malware scoperto da Infoblox che utilizza il DNS (Domain Name System) per eseguire operazioni di comando e controllo (C2). Un client compromesso comunica con un controller e riceve indicazioni da quest'ultimo tramite query DNS. Tale controller è integrato in un name server DNS a cui vengono trasmesse le query attraverso il normale processo di risoluzione. Abbiamo rivelato l'esistenza di Decoy Dog nell'aprile 2023 e abbiamo pubblicato un rapporto dettagliato dei nostri risultati iniziali il 23 aprile. La scoperta si basava sul monitoraggio dei dati DNS. L'analisi di quel periodo ha confermato che il toolkit era stato costruito sulla base di un trojan di accesso remoto (RAT, Remote Access Trojan) noto come Pupy, ma non era noto quali sistemi venissero sfruttati, come il toolkit fosse stato distribuito o se Pupy fosse stato modificato.[1] Ci aspettavamo che, con i dettagli che abbiamo fornito, altri membri della comunità avrebbero individuato le macchine compromesse e che l'intera storia sarebbe diventata nota. Tuttavia, il mistero che circonda Decoy Dog non ha fatto che crescere.

Da aprile, Infoblox ha condotto ulteriori ricerche su Decoy Dog e Pupy. Questo report è il risultato di tale ricerca. Abbiamo appreso che Decoy Dog è un aggiornamento importante di Pupy che utilizza comandi e configurazioni che non sono presenti nel repository pubblico. Abbiamo sviluppato degli algoritmi per separare le comunicazioni del client Decoy Dog e dedurre una serie di altre proprietà su ogni controller. Ciò ci consente di concludere con elevata certezza che il toolkit si è diffuso ed è sotto il controllo di almeno tre attori. Sebbene l'attività che abbiamo osservato rimanga limitata alla Russia e all'Europa orientale, ci sono raggruppamenti distinti di tecniche, tattiche e procedure (TTP) all'interno dei controller coerenti con più attori.

Ogni attore di Decoy Dog ha risposto alle nostre rivelazioni di aprile in qualche modo, e le variazioni supportano la nostra valutazione di più operatori. Subito dopo il primo annuncio sui social media, alcuni name server sono stati disattivati. Tutti i rimanenti sono stati modificati per rimuovere il comportamento che abbiamo evidenziato nel nostro primo documento, anche se questo è stato realizzato in modi diversi a seconda del controller. Un gruppo di controller ha iniziato a limitare le risposte alle query a seconda del Paese di origine, una tecnica chiamata geofencing, mentre altri hanno modificato la risposta alle query per il sottodominio ping.

Un attore ha risposto così rapidamente alla nostra divulgazione su LinkedIn che inizialmente abbiamo pensato che i nuovi domini fossero registrazioni imitate da ricercatori di sicurezza. Ulteriori analisi, tuttavia, hanno dimostrato che si trattava di domini sostitutivi. Invece di interrompere l'operazione, l'attore ha trasferito i client compromessi esistenti ai nuovi controller. Si tratta di una risposta straordinaria che dimostra che l'attore ha ritenuto necessario mantenere l'accesso alle sue vittime esistenti. Ha creato una netta separazione tra le TTP di un set di domini Decoy Dog e tutti gli altri.

Nelle settimane successive al nostro annuncio, siamo rimasti sorpresi dal fatto che nessuno si sia fatto avanti per identificare il malware e la vulnerabilità sottostanti che hanno dato a Decoy Dog il suo punto d'appoggio per operare. Ma con il progredire della nostra ricerca, è diventato chiaro il motivo per cui le comunicazioni non sono state rilevate per oltre un anno. Gli attacchi che utilizzano Decoy Dog sono stati altamente mirati e ogni controller ha

un numero limitato di client attivi. Alcuni server hanno mantenuto costantemente da quattro a otto client attivi per mesi alla volta. Mentre altri hanno registrato un aumento del numero di client attivi contemporaneamente nel tempo, il numero totale di dispositivi interessati osservati in qualsiasi momento è stato inferiore a 100. Un piccolo set di vittime esclude in genere che gli attori siano mossi da motivi finanziari e la necessità di persistere su un dispositivo per un lungo periodo di tempo è coerente con gli attori altamente avanzati.

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

Siamo stati in grado di ricostruire parti delle comunicazioni di Decoy Dog identificando le firme del nostro traffico Pupy. Abbiamo creato un server Pupy su Internet, che, combinato con il reverse engineering selettivo del codice, ci ha permesso di correlare le query e le risposte DNS a specifici comandi Pupy. Da ciò siamo stati in grado di a) determinare che Decoy Dog contiene comandi non presenti in Pupy e b) caratterizzare la maggior parte delle comunicazioni. Inoltre, gli attori di Decoy Dog sembrano sfruttare Pupy per utilizzare altri livelli di trasporto al di fuori del DNS per funzioni come lo scambio di chiavi. Gli attori di minacce probabilmente considerano questo uno dei vantaggi di Pupy come trojan di accesso remoto (RAT).

La prima implementazione nota del toolkit Decoy Dog ha avuto luogo a fine marzo o inizio aprile 2022. È stata venduta o rubata poco dopo, come indicato dalla comparsa di un secondo controller, con TTP diverse, attivo a metà maggio. È stato poi registrato un terzo dominio nel luglio 2022 e fatto "invecchiare" strategicamente fino a settembre. È possibile che questi ultimi due controller siano di proprietà dello stesso attore, poiché condividono molte caratteristiche, tra cui l'hosting nello spazio IP russo. Tuttavia, presentano alcune differenze. Qualche mese dopo, sono stati registrati altri due domini, sempre con caratteristiche distinte dai precedenti controller. L'attore che ha registrato questi domini ha migrato i client immediatamente dopo la nostra divulgazione su nuovi domini. In totale, Infoblox sta monitorando 21 domini Decoy Dog, alcuni dei quali sono stati registrati e distribuiti nell'ultimo mese.

Dopo aver stabilito che Decoy Dog differiva significativamente da Pupy attraverso la nostra analisi dei log DNS, abbiamo esaminato i campioni binari correlati disponibili su VirusTotal per vedere se le differenze erano evidenti negli eseguibili. Il reverse engineering di questi campioni ha dimostrato che, sebbene siano stati rilevati come Pupy, sono molto più avanzati della versione open source. I campioni includono a) la capacità di eseguire codice Java arbitrario sul client, b) diversi nuovi meccanismi di trasporto e c) nuovi meccanismi DNS per garantire la persistenza. Un meccanismo è simile a un tradizionale algoritmo di generazione di domini DNS (DGA, Domain Generation Algorithm) e utilizza provider DNS dinamici gratuiti per connettersi ai cosiddetti controller di emergenza. Tutti i campioni condividono gli stessi aggiornamenti fondamentali, anche se uno di essi presenta delle capacità uniche non riscontrabili negli altri, legate all'utilizzo dei trasporti in streaming.

Per ragioni che rimangono poco chiare, Decoy Dog viola i principi fondamentali delle comunicazioni segrete, che generalmente mirano a evitare il rilevamento e il recupero dei contenuti da parte di un avversario. Mentre i normali server Pupy rifiutano le richieste di comunicazione ripetute da client compromessi, i server Decoy Dog non solo rispondono alle richieste DNS riprodotte, ma rispondono a qualsiasi query ben congegnata. Questo comportamento è simile alle configurazioni jolly nel DNS ed è stato un fattore significativo nel rilevamento di Decoy Dog da parte di Infoblox. Data la sofisticazione di Decoy Dog, ipotizziamo che la riproduzione e il comportamento jolly siano una scelta di progettazione; qualunque sia l'intenzione, la riproduzione diffusa del DNS è stata in parte responsabile dell'incapacità del settore di vedere Decoy Dog come un nuovo malware.

La scansione aggressiva di Internet da parte di un fornitore di sicurezza ha portato alla ritrasmissione di milioni di comunicazioni Decoy Dog attraverso reti globali, tra cui diversi nostri clienti. Ciò, a sua volta, ha portato alla scoperta del toolkit. L'incapacità del fornitore di identificare il traffico come malware, per evitare di riprodurre le query, ha attivato connessioni DNS da reti non infette ai controller Decoy Dog. Siamo certi che nessun cliente di Infoblox sia stato infettato e che le query ai nostri resolver siano state tutte il risultato di una scansione anomala dei fornitori. Nonostante l'assenza di una minaccia immediata per le reti dei nostri clienti, Decoy Dog rimane un toolkit sofisticato dalle origini incerte e potrebbe continuare a diffondersi.

Non solo Decoy Dog è stato osservato di recente in circolazione, ma, a nostra conoscenza, è il primo utilizzo del componente DNS C2 di Pupy in un'operazione dannosa. In parte, ciò è probabilmente dovuto alla difficoltà di stabilire un name server Pupy, che richiede la modifica del software nel repository e la corretta configurazione del DNS. La mancanza di

esposizione rende più difficile per il settore della sicurezza rilevare e difendersi sia da Pupy che da Decoy Dog. Per contribuire a interrompere le operazioni che utilizzano questi sistemi C2, stiamo fornendo alla comunità un set di dati di ricerca contenente il traffico DNS di Pupy acquisito dal nostro server e i dettagli del funzionamento interno del software. Questa documentazione è la prima del suo genere e permetterà ad altri di costruire algoritmi di rilevamento, oltre che di riprodurre le nostre scoperte.

La storia di Decoy Dog rivela la potenza del DNS come fonte di rilevamento e risposta alle minacce. Rivela anche una debolezza intrinseca dell'ecosistema di intelligence incentrato sul malware che domina il settore della sicurezza. Il toolkit è stato scoperto da algoritmi di rilevamento di minacce DNS e l'unica difesa contro di esso oggi è il DNS. Inoltre, avevamo segnalato diversi domini del controller come sospetti e li stavamo bloccando nei nostri resolver prima di renderci conto che stavano tutti utilizzando un malware comune. Questo tipo di protezione, che ostacola le attività dannose prima che vengano identificate e spesso prima che diventino operative, è esclusivo dei sistemi di rilevamento e risposta DNS.

In questo documento forniamo ai difensori le conoscenze per identificare Pupy e Decoy Dog. Anche se descriveremo il C2 DNS in modo approfondito, non forniremo informazioni che aiutino i malintenzionati a implementare Pupy, né divulgheremo la firma DNS completa di Decoy Dog. Spieghiamo alcuni comportamenti che abbiamo identificato nel nostro documento originale ed evidenziamo come Decoy Dog è distinto da Pupy. Inoltre, descriveremo la nostra analisi di grandi volumi di traffico DNS di Decoy Dog che ci ha permesso di stimare il numero di client e il traffico dei comandi senza possedere il malware stesso o controllare il name server. Descriviamo come i campioni Decoy Dog differiscono da Pupy. Infine, discutiamo il modo in cui gli operatori di Decoy Dog hanno reagito alle nostre rivelazioni e dimostriamo i tratti comuni tra i sottogruppi di controller. Le appendici contengono ulteriori informazioni tecniche di supporto.

CONTESTO

Infoblox ha scoperto Decoy Dog, un toolkit di comando e controllo (C2) che utilizza il DNS (Domain Name System) all'inizio di aprile 2023. Si basa su un trojan di accesso remoto (RAT) open source chiamato Pupy[2] e trasporta comunicazioni crittografate tra client e server o controller, tramite query di nomi di dominio e risposte di indirizzi IP. La scoperta è nata da algoritmi che monitorano le query DNS passive ai resolver Infoblox per comportamenti anomali. Le query per i domini Decoy Dog sono state effettuate dalle appliance di sicurezza in un piccolo numero di reti di clienti. Queste query hanno creato una firma coerente con il beaconing malware persistente e di basso profilo. L'analisi umana dell'attività è stata allarmante perché, sebbene il DNS fosse chiaramente utilizzato come canale di comunicazione riservato, i domini non sono stati identificati come C2 in nessun dato di intelligence disponibile al pubblico. In effetti, alcuni sono stati etichettati come "rispettabili" nei controlli di reputazione online. Abbiamo rilasciato una serie di domini il 13 aprile per aiutare la comunità a bloccare il traffico e identificare la natura della compromissione.

Durante la nostra ricerca originale, Infoblox ha identificato una firma DNS univoca indipendente dal software Pupy. Gli attori avevano implementato e gestito il loro sistema C2 in un modo molto specifico; per questo motivo, abbiamo identificato Decoy Dog come un toolkit distinto. Solo un numero limitato di domini in tutto il mondo condivideva questa firma, tutti name server Decoy Dog.

Il 23 aprile, abbiamo pubblicato parte della firma, l'analisi iniziale del DNS passivo e un sottoinsieme dei domini controller nel nostro report "Dog Hunt: Finding Decoy Dog Toolkit in Anomalous DNS Traffic".[3] Questo documento ha messo in evidenza un

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

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

comportamento specifico di Pupy, ovvero che restituiva una serie di risposte localhost a query per sottodomini specifici contenenti "ping". Inoltre, ha descritto una serie di tendenze all'interno delle comunicazioni DNS che all'epoca non potevamo spiegare completamente. In particolare, abbiamo identificato modelli sorprendenti negli indirizzi IP restituiti nelle risposte e il fatto che i server hanno risposto alle query riprodotte, il che è inaspettato per un sistema di comunicazione segreto.

In seguito agli annunci, un'ampia gamma di membri della comunità della sicurezza, tra cui venditori e altre organizzazioni, ci ha contattato. Molti di loro avevano visto del traffico correlato nelle proprie reti o in quelle dei clienti, ma nessuno aveva identificato i dispositivi compromessi o riconosciuto la portata dell'attività. Alcune di queste organizzazioni hanno fornito informazioni che ci hanno portato a isolare e confermare come è stato generato il traffico DNS nelle nostre stesse reti. Altri hanno contribuito a confermare l'ampiezza dell'attività e a verificare le ipotesi. Questa collaborazione informale è stata molto utile e ne siamo grati.

Per semplicità, usiamo il termine Pupy in questo documento per riferirci specificamente al componente DNS C2 di Pupy e non a Pupy in generale.

Pupy

UN TIPO RARO

Pupy è un trojan di accesso remoto (RAT) post-exploitation open source che presenta un complesso sistema di trasporto modulare.[4] Mentre la base di codice principale di Pupy è stata resa disponibile in GitHub nel 2015, il meccanismo DNS C2 non è stato aggiunto fino al 2019. Questo documento è la prima documentazione pubblica di Pupy C2. Inoltre, stiamo fornendo un set di dati in GitHub per consentire ad altri di riprodurre il nostro lavoro e creare difese per il futuro.

Sebbene Pupy sia open source, l'uso del protocollo DNS C2 è raro; non siamo stati in grado di identificarne l'uso al di fuori di Decoy Dog "in the wild".[5] Dai nostri resolver, che servono imprese e organizzazioni in tutto il mondo, non abbiamo trovato alcuna prova dell'uso del componente DNS C2 di Pupy storicamente. All'interno dei pDNS globali per i primi sei mesi del 2023, utilizzando i rilevatori DNS che abbiamo sviluppato per Pupy, non abbiamo trovato alcun utilizzo del software al di fuori di Decoy Dog. Infine, abbiamo interpellato privatamente un'ampia gamma di fornitori, nessuno dei quali ne ha visto l'utilizzo. Laddove è stato riportato l'uso di Pupy da parte di attori di minacce persistenti avanzate (APT, Advanced Persistent Threat), apparentemente non sono stati impiegati i componenti DNS C2.[6]

Il raro utilizzo di Pupy è probabilmente dovuto, almeno in parte, alla difficoltà di funzionamento del sistema. Stabilire le comunicazioni di Pupy attraverso il DNS globale non è facile. Richiede la corretta configurazione del name server e la modifica del codice nel repository GitHub. Inoltre, ci sono complessità nel DNS che variano tra i resolver ricorsivi che il software Pupy non gestisce correttamente. Queste sfide hanno probabilmente ostacolato la sua adozione sia da parte dei red team che degli hacker, a differenza di strumenti popolari come Cobalt Strike, che vediamo abbastanza frequentemente.[7]

---

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

5 L'espressione "in the wild" (in circolazione) è utilizzata nel gergo della sicurezza informatica per indicare l'impiego operativo e non come parte di un test di penetrazione isolato o di una ricerca.

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/

Sebbene il componente DNS C2 di Pupy sia oggi raro, l'uso di Decoy Dog si sta diffondendo e la probabilità che i difensori debbano affrontare Pupy in qualche forma è in aumento. Per aiutare la comunità a prepararsi, Infoblox ha svolto una ricerca significativa sia su Decoy Dog che su Pupy. Infoblox ha distribuito un server Pupy su Internet per confrontare il suo comportamento con quello di Decoy Dog. Abbiamo poi acquisito i dati dei pacchetti (pcap) e i log DNS passivi dai resolver Infoblox. Abbiamo utilizzato la nostra distribuzione di Pupy insieme al reverse engineering selettivo del codice per comprendere meglio la natura unica di Decoy Dog. In questa sezione, spieghiamo i componenti di Pupy che sono rilevanti per la nostra ricerca. Per semplicità, limitiamo questo articolo alle comunicazioni che utilizzano le risposte IPv4 (record A), anche se, quando disponibile, Pupy utilizzerà le risposte IPv6 (AAAA). La codifica delle query descritta nel documento è l'attuale impostazione predefinita di Pupy, versione 2 (se non diversamente specificato).[8]

COME FUNZIONA PUPY

Nel nostro precedente documento, abbiamo fornito una panoramica di Pupy e abbiamo evidenziato alcune caratteristiche insolite di Decoy Dog.[9] In questo documento, approfondiremo il protocollo di comunicazione di Pupy per dimostrare i suoi collegamenti con Decoy Dog e come sfruttare i DNS di Pupy raccolti passivamente per comprendere un'operazione in corso.

Pupy è stato progettato per fornire comunicazioni continue tra i client infetti e il server, in modo che quando l'attore vuole accedere in remoto al client, la connessione sia già stabilita. L'attore è in grado di monitorare i client connessi e di comandarli selettivamente per fornire un'ampia gamma di azioni. Il DNS viene utilizzato solo per le comunicazioni C2. Tutti i dati significativi esfiltrati dal client vengono inviati tramite una delle tante altre opzioni di trasporto offerte da Pupy. Di conseguenza, il client DNS di Pupy si limita a stabilire una connessione con il controller, a riconoscere i comandi, a fornire informazioni sul sistema e a una manciata di altre funzioni. Tra la gestione dei comandi dal server, il client entra in sospensione.

Le comunicazioni DNS vengono avviate e gestite dal client. Il client invia query tramite il normale percorso di risoluzione DNS o tramite DNS su HTTPS (DoH) quando è abilitato e disponibile.[10] Il controller invia comandi in risposta alle richieste del client sotto forma di indirizzi IP crittografati. Ogni query-risposta è una comunicazione completa, il che significa che né il client né il server possono suddividere i dati per un singolo comando in due query DNS. Questo protocollo si distingue dai comuni sistemi di tunneling DNS, ad esempio Iodine,[11] in cui il client stabilisce una sessione su DNS che può includere la ricostruzione di diversi pacchetti a entrambe le estremità per elaborare la comunicazione. Il client è obbligato a riconoscere la maggior parte dei comandi e il server risponde a ogni query del client valida con comandi o riconoscimenti. Il vocabolario del client è estremamente limitato. Ha nove tipi di query con cui gestisce le sessioni, riconosce i comandi, invia informazioni sul sistema e stabilisce le chiavi. I comandi personalizzati possono essere aggiunti scrivendo funzioni aggiuntive, ma richiedono una piena comprensione del software.

Alla riattivazione, il client interroga il server in due modi diversi, a seconda che sia stata stabilita o meno una chiave condivisa. Questa interrogazione fornisce al server informazioni aggiornate sul sistema e sullo stato del client Pupy, oppure effettua una semplice query

8 Una versione precedente di Pupy C2 non includeva le informazioni sull'host in ogni query. Ora sappiamo che Decoy Dog è alla versione 3 del client, ma la codifica della query sembra essere la stessa della versione 2.

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

10 Pupy utilizza i server Quad9 per impostazione predefinita per DoH.

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

che serve per avviare una nuova sessione crittografata. Sebbene sia possibile disabilitare le sessioni crittografate, questa non è l'impostazione predefinita e non è stata osservata in Decoy Dog. In risposta, il controller riconosce la richiesta, richiede al client di eseguire uno scambio di chiavi o invia nuovi comandi. Una volta completato il set completo di comandi, il client entrerà in sospensione per l'intervallo stabilito, per impostazione predefinita 60 secondi. Questo processo viene ripetuto mentre il client è in esecuzione. Una panoramica di alto livello delle comunicazioni client-server di Pupy è mostrata nella Figura 1 e una visione più dettagliata del processo del client è disponibile nell'Appendice A.

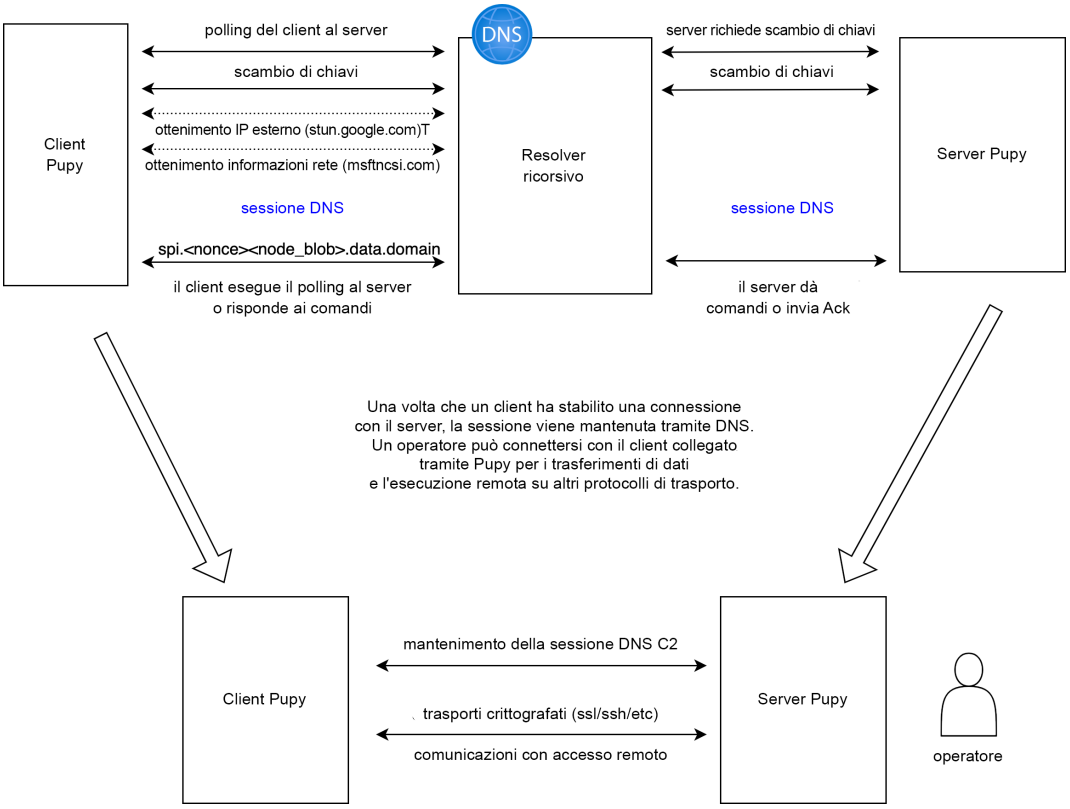


Figura 1. Una panoramica di alto livello delle comunicazioni Pupy.

L'attore Pupy interagisce con i client riga di comando del controller. Quando il client contatta il controller, tutti i comandi in coda verranno codificati nella risposta DNS. L'operatore stabilisce una connessione su una porta aperta del client e specifica il livello di trasporto da utilizzare per l'esfiltrazione. Le comunicazioni DNS del server sono ancora piuttosto limitate, anche se più estese rispetto al client. Esiste un'ampia gamma di comandi e questi possono essere concatenati in un'unica risposta al client. Mentre il client avvia lo scambio di comunicazioni, il server è responsabile di garantire la sicurezza delle comunicazioni. Lo fa imponendo le cosiddette sessioni con ciascun client, che servono a ruotare le chiavi di crittografia. Questa operazione è descritta nella sezione successiva.

INIZIO DELLA SESSIONE

Pupy richiede che venga stabilita una sessione crittografata tra il client e il controller prima di trasmettere i comandi dell'attore. Questa sessione scade quando si verifica il timeout delle comunicazioni client e potrebbe essere forzata a rinnovarsi per altri motivi, inclusi errori nella decodifica delle query DNS o un riavvio del client. Le sessioni sono identificate dalla presenza di un'etichetta SPI (Security Parameter Index) nella query e vengono crittografate utilizzando una chiave condivisa temporanea. Dal momento che i dettagli della comunicazione dipendono da una serie di fattori, tra cui il fatto che il client si sia precedentemente collegato al server, il protocollo esatto per l'inizializzazione della sessione

può differire, creando varianze negli scambi DNS osservati. Tuttavia, lo scambio tipico è il seguente:

- Il client stabilisce una connessione con il server senza una sessione stabilita o con una sessione scaduta (query 1).
- Il server risponde con un comando che richiede uno scambio di chiavi e informazioni sul sistema del client.[12]
- Il client riconosce la richiesta di informazioni di sistema (query 2).
- Il client genera una coppia di chiavi pubblica-privata casuale utilizzando un algoritmo a curva ellittica e le invia al server; il server fa lo stesso e risponde con la nuova chiave (query 3).
- Il client e il server utilizzano questo scambio per stabilire una nuova chiave di sessione condivisa, che viene utilizzata per crittografare i pacchetti con crittografia AES e anche per creare l'SPI per identificare la sessione.
- Il client raccoglie informazioni sulla sua rete, incluso il suo indirizzo IP esterno, utilizzando query DNS aggiuntive ad altri servizi.
- Il client trasmette queste informazioni utilizzando la chiave di crittografia condivisa e segnalando la presenza di una sessione attiva con l'inclusione dell'SPI nella query (query 4).
- Il client invia ulteriori informazioni sullo stato del sistema (query 5).

La chiave condivisa e l'SPI vengono in genere stabiliti dopo tre query, anche se lo scambio di chiavi è tecnicamente una singola query e risposta. Durante una sessione, ogni query e risposta verranno crittografate utilizzando questa chiave condivisa. La crittografia utilizza anche un nonce a 32 bit generato dal client che cambia per ogni query. Quando viene stabilita una nuova sessione, le chiavi vengono rigenerate, ma il valore nonce del client continua mentre il client è operativo. Questo argomento viene discusso ulteriormente nella sezione seguente.

CODIFICA DELLE QUERY

Il client genera query che contengono comunicazioni crittografate con il server. Queste possono includere informazioni sullo scambio di chiavi o una risposta ai comandi del server. In ogni query è possibile comunicare un massimo di 52 byte di dati trasmessi. Oltre ai dati trasmessi, ogni query include:

- nonce, un valore incrementale di 4 byte generato dal client
- versione, un valore di 1 byte che indica la versione DNS C2 di Pupy
- cid, un valore di 4 byte tratto dalla configurazione del client, che viene generato casualmente durante la creazione del client
- iid, un valore di 2 byte contenente gli ultimi 16 bit del processo client di Pupy
- node id, un valore di 6 byte tratto dal client, in genere l'indirizzo MAC del dispositivo
- opzionalmente, SPI, un valore di 4 byte generato durante lo scambio di chiavi e presente nelle query che rappresentano una sessione sul server per un determinato client.

Ogni query del client include questi 13 byte di informazioni sul client nonché un checksum di 4 byte sul payload sottostante. Il payload sottostante è crittografato ed è costituito da una serie di comandi e dati correlati.

12 Si tratta in genere di due comandi denominati Policy e Poll lato server.

Il client crittografa e codifica i dati da trasmettere al server come nome di dominio completo (FQDN, Fully Qualified Domain Name), chiamato nome di query (qname) nel protocollo DNS. L'intero processo, mostrato nella Figura 2 di seguito, include la crittografia, l'organizzazione e la codifica sia dei dati trasmessi che delle informazioni aggiuntive necessarie al server. Funziona come segue:

- Ai dati da trasmettere vengono aggiunte informazioni specifiche dell'host.
- Questa stringa di byte composita viene crittografata utilizzando una chiave simmetrica condivisa e il nonce corrente.
- I primi byte crittografati dei dati trasmessi, fino a 35 byte, vengono codificati e utilizzati per la prima etichetta, o quella più a destra, del qname.
- Il resto dei byte crittografati, che può contenere fino a 17 byte di dati trasmessi, viene preceduto dal valore nonce corrente e codificato per creare la seconda etichetta del qname.
- Se l'SPI (Security Parameter Index) esiste nel client, è codificato e utilizzato nella terza etichetta, o in quella più a sinistra, del qname; questo valore viene impostato dopo uno scambio di chiavi con il server.
- Il nonce viene incrementato della lunghezza dei dati crittografati all'interno del client per essere utilizzato per la query successiva.

La codifica da byte crittografati a un'etichetta di nome di dominio è stata descritta nel nostro articolo precedente. Utilizza una mappa personalizzata in combinazione con la codifica a 32 bit per garantire che il risultato finale sia un nome di dominio valido. La struttura del payload dei dati sottostante è descritta nell'Appendice B.

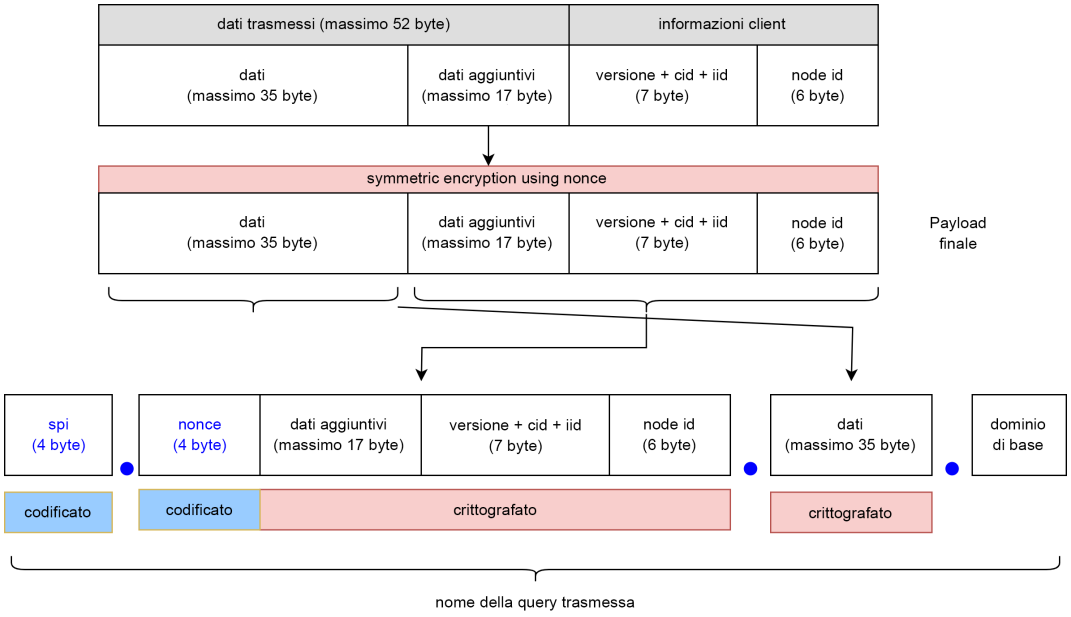


Figura 2. Il processo per convertire i dati dal client in un qname per una query DNS. Il dominio di base è il nome di dominio del server Pupy.

Pupy utilizza AES per impostazione predefinita quando crittografa le query DNS. Se è stata stabilita una chiave condivisa con il client, la utilizza per crittografare simmetricamente l'intera stringa di byte, altrimenti utilizza la chiave pubblica stabilita. In entrambi i casi, il nonce corrente viene utilizzato anche nella crittografia per garantire che la query codificata sia univoca, anche se i dati trasmessi sottostanti rimangono gli stessi in più query. Questo è un meccanismo standard per la protezione dagli attacchi crittografici. Di conseguenza,

il nome di dominio interrogato può essere decodificato per rivelare i dati crittografati, ma i dati crittografati non possono essere decrittografati senza la chiave. Il valore nonce viene inizializzato con un valore casuale a 32 bit e viene incrementato della lunghezza del payload su ogni query.

Quando il name server Pupy riceve una query, decodifica il nome di dominio per rivelare il valore SPI, il nonce e il payload crittografato. Per assicurarsi che riceva comunicazioni client valide, il server verifica se l'SPI è valido quando presente e che il nonce sia maggiore di quello precedente registrato per il client. Effettua diversi altri controlli sui dati, incluso un controllo sul numero di versione, che è crittografato nel payload. Se uno qualsiasi di questi controlli fallisce, restituirà un errore al client.

In particolare, Pupy non risponde due volte alla stessa query e qualsiasi server Pupy non modificato risponderà a una query già ricevuta in passato con una risposta NXDOMAIN (dominio non esistente). Abbiamo convalidato questo comportamento con il nostro server Pupy, tentando di interrogare un nome di dominio precedentemente interrogato. Questo è importante perché è una caratteristica di Decoy Dog che risponde alle query DNS riprodotte con risposte coerenti con il protocollo Pupy C2.

Poiché la query DNS contiene una codifica reversibile del nonce e il nonce aumenta della lunghezza del payload in ogni query, possiamo ricostruire i thread di query associati a un singolo client. Come vedremo più avanti in questo articolo, attraversp la raccolta dei dati DNS passivi per un dominio Pupy o Decoy Dog, possiamo usare questa ricostruzione per stimare il numero di client e la natura della comunicazione in determinati casi.

GESTIONE SPECIALE DEI NOMI DI DOMINIO

Alla ricezione di una query, il server analizza il nome della query e determina se corrisponde alla struttura appropriata per un pacchetto crittografato proveniente da un client. Ci sono alcuni casi speciali che hanno un'elaborazione unica. A parte questi casi speciali, il server rifiuterà qualsiasi richiesta che non soddisfi il formato previsto. Uno di questi casi speciali sono le richieste di ping, che abbiamo descritto nel nostro documento precedente. Una query per un sottodominio pingN, dove

N è un numero intero, restituirà una sequenza di risposte localhost lunga N. Una query per il ping stesso restituisce 15 risposte di questo tipo, mentre una query per il dominio di base restituisce una sola risposta localhost, cioè 127.0.0.1.

Al di fuori delle richieste di ping, il server può essere configurato per rispondere a query con etichetta singola con un singolo indirizzo IP. Lo scopo di questa funzionalità è sconosciuto e non sembra essere utilizzata nel client; nel codice sorgente viene indicata come richiesta di attivazione DNS. Questa funzionalità non è documentata e per utilizzarla un attore dovrebbe capire come funziona il software del server.

La gestione speciale per i sottodomini con etichetta singola viene eseguita configurando le voci di "attivazione", che sono coppie di stringhe chiave-valore. Il valore viene quindi utilizzato insieme alla chiave privata del server per creare un indirizzo IP di risposta. Questa risposta viene creata utilizzando una funzione hash unidirezionale e non può essere invertita. L'hash distingue tra maiuscole e minuscole ed è definito come

MD5(subdomain_label + activation_value + private_key)

CODIFICA DELLA RISPOSTA

Quando il server riceve una query da un client, decodifica, decrittografa, controlla i risultati ed elabora i dati del client. In particolare, una comunicazione con il client formattata correttamente deve contenere due o tre etichette, come descritto in precedenza nella sezione sulla codifica delle query. Il server assemblerà quindi una risposta al client contenente uno o più comandi. Sebbene possa restituire query IPv4 (A) o IPv6 (AAAA), limiteremo la nostra descrizione alle query IPv4 (A) per semplicità.

La risposta del server è una stringa binaria crittografata che viene poi codificata in uno o più record A.13 Il processo per questa codifica è mostrato nella Figura 3 qui sotto. Il numero massimo di byte nella risposta è 64, che vengono codificati in segmenti di 3 byte, il che comporta un massimo di 22 indirizzi IPv4 nella risposta. [13]

- Nel primo passaggio, il server calcola la lunghezza della risposta e la antepone ai dati della risposta. Aggiunge quindi byte casuali per creare una stringa composita di lunghezza multipla di 3 byte.[14] Chiamiamo questa stringa composita payload.
- Nella seconda fase, gli indirizzi IPv4 vengono creati in modo iterativo da segmenti di 3 byte del payload. Ogni indirizzo IPv4 è rappresentato da un valore a 32 bit, dove il bit 0 è il bit alto.
- I primi 3 bit di ciascun indirizzo sono casuali.
- Ogni segmento ha un indice, che consente al client di ordinare i dati al momento della ricezione; questo è rappresentato da 5 bit. Questo indice si trova nei bit 3-7 del risultato.
- Il segmento del payload è nei bit 8-30, il che impone che il bit più alto del segmento del payload sia il bit inferiore del primo ottetto dell'indirizzo IPv4.
- Infine, il bit meno significativo, il bit 31, è un bit di controllo generato nel segmento del payload. A causa della natura di questo checksum, questo bit è 1 nel 75% degli indirizzi IPv4.
- La stringa a 32 bit risultante viene interpretata come indirizzo IPv4 e aggiunta alla risposta.

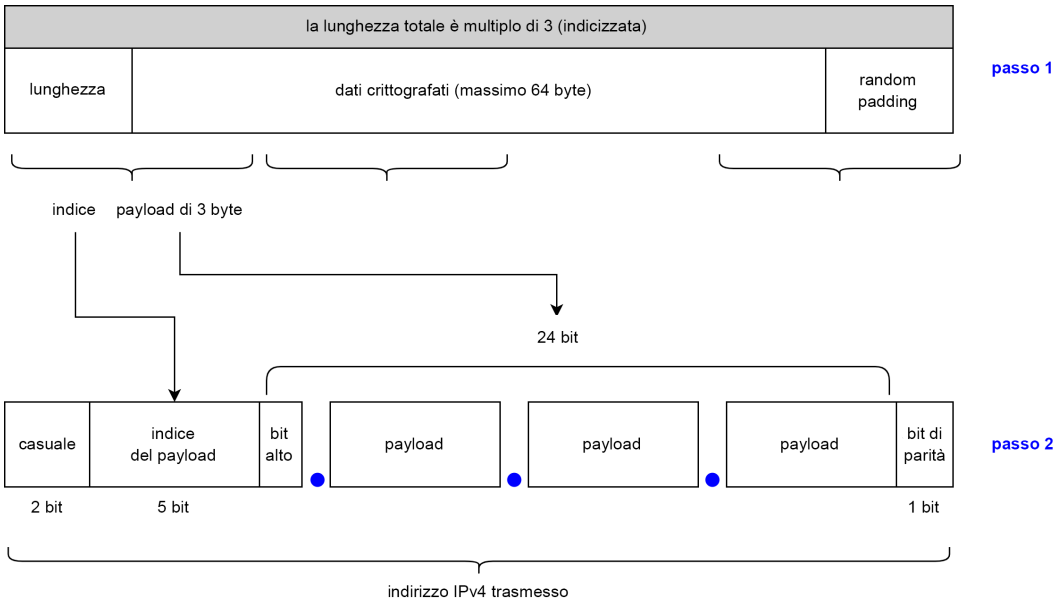


Figura 3. Codifica del server Pupy di una risposta IPv4 a una query del client. I dati sono codificati in una serie di indirizzi IPv4 utilizzando 3 byte del payload in ciascun indirizzo.

Nel nostro precedente articolo, abbiamo notato che Decoy Dog ha una distribuzione sorprendente delle risposte IPv4. Ora sappiamo che si trattava di un artefatto della codifica

13 Una serie di comandi viengono assemblati e quindi crittografati utilizzando una chiave condivisa e il nonce corrente prima della codifica, a condizione che lo scambio di chiavi sia stato completato. In caso contrario, la chiave privata del server viene utilizzata, insieme al nonce, per crittografare i dati con un algoritmo a curva ellittica a chiave pubblica.

14 Nel codice, questo processo è più complicato ma ha lo stesso risultato.

della risposta Pupy. L'uso di tre bit casuali e di un indice incrementale come primi 7 bit del primo ottetto in ogni risposta garantisce che gli indirizzi IPv4 risultanti siano in intervalli specifici e che tali intervalli siano direttamente correlati al numero di risposte individuali nella risposta completa alla query, che a sua volta è determinato dalla dimensione dei dati trasmessi al client. In particolare, il primo indirizzo IP sarà sempre compreso tra 64.0.0.0/8, 128.0.0.0/8 o 192.0.0.0/8.

Ogni volta che l'indice viene aumentato, le scelte per il primo ottetto dell'indirizzo IP vengono spostate di due. Specificamente:

- Il primo indirizzo IP inizierà con 64, 128 o 192 perché l'indice è 0 e la lunghezza è al massimo 64. Di conseguenza, solo i primi 3 bit vengono impostati nel primo indirizzo IP della risposta.
- Il secondo indirizzo IP inizierà con 66, 67, 130, 131, 194 o 195 perché l'indice è 1, che aggiunge 2 ai 3 bit superiori generati casualmente, e il bit superiore del payload di dati può essere 0 o 1.
- Il terzo indirizzo IP inizierà con 68, 69, 132, 133, 196 o 197, ecc.

Possiamo vedere il risultato di questo algoritmo per un numero crescente di risposte nella Figura 4 qui sotto. In particolare, utilizziamo una mappa di Hilbert per dimostrare come il primo ottetto degli indirizzi IP sia correlato al numero totale di risposte complete per 3, 12 e 15 risposte individuali.

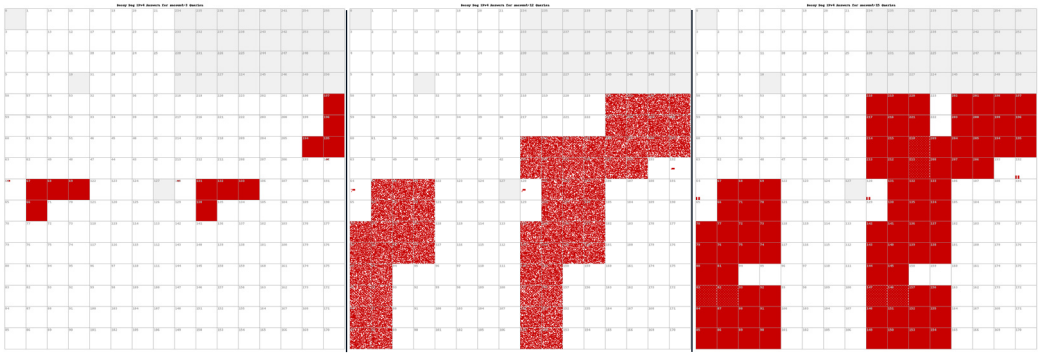


Figura 4. Mappe di Hilbert che dimostrano la distribuzione degli indirizzi IPv4 nelle risposte complete di Pupy contenenti rispettivamente 3, 12 e 15 risposte individuali.

La struttura degli indirizzi IPv4 consente a chiunque osservi la risposta completa di ricostruire i dati trasmessi. Sebbene questi dati siano crittografati, le risposte possono essere profilate utilizzando l'analisi della lunghezza e delle serie temporali. Questo tipo di analisi può rivelare informazioni sulle comunicazioni, come vedremo più avanti in questo documento.

ANALISI PASSIVA DEI DATI

Sebbene le comunicazioni Pupy siano fortemente crittografate, le informazioni necessarie per decrittografare e tracciare i pacchetti sono codificate in modo reversibile. Se le query e le risposte DNS vengono raccolte, possono essere analizzate in forma aggregata per ricavare informazioni sulla distribuzione di Pupy e sui client. La raccolta passiva dei dati DNS, che prende il nome di DNS passivo (anche pDNS), avviene in molti luoghi di Internet, tra cui i resolver aziendali, i resolver ricorsivi pubblici, nonché i server root e TLD. Nelle sezioni che seguono, mostriamo come la raccolta DNS passiva delle query Pupy può essere sfruttata per ottenere informazioni sulle comunicazioni.

Possiamo recuperare una grande quantità di informazioni su un controller Pupy e sui suoi client dal DNS passivo. In particolare, possiamo recuperare:

- il numero approssimativo di client attivi in qualsiasi momento,
- i tipi di scambi che avvengono tra il server e i client,
- le firme della distribuzione, ad esempio l'intervallo di sospensione del client e
- una cronologia degli scambi di chiavi dei client e dell'attività complessiva.

Abbiamo utilizzato queste tecniche per analizzare il traffico proveniente dal nostro server e dai server di Decoy Dog. Questo ci ha permesso di capire quanto sia simile Decoy Dog a Pupy e quanto siano simili i server tra loro. In definitiva, queste tecniche ci hanno permesso di profilare ogni distribuzione di Decoy Dog. I dettagli tecnici dei metodi utilizzati sono ulteriormente trattati nell'Appendice C.

FIRME DEL PAYLOAD DI PUPY

La natura delle comunicazioni tra un client e un server può essere dedotta in una certa misura utilizzando l'analisi passiva dei dati. Il vocabolario del client, ossia i payload distinti che può creare, è molto limitato: ci sono solo nove tipi di comunicazioni client. Due tipi condividono la stessa lunghezza del payload, mentre un altro tipo può avere più lunghezze. Un attore può creare eventi personalizzati in Pupy, creando potenzialmente un'ulteriore diversità di lunghezza del payload.

Il server ha un vocabolario più flessibile ed è in grado di trasmettere più comandi in un'unica risposta DNS, il che rende più difficile la profilazione. Tuttavia, la stragrande maggioranza delle comunicazioni in un sistema Pupy riguarda l'inizializzazione della sessione, lo scambio di chiavi e gli heartbeat del client con il server. Le comunicazioni del server sono dominate da riconoscimenti delle richieste dei client, messaggi di errore, inclusa la necessità di stabilire una nuova sessione e scambi di chiavi.

Di conseguenza, è possibile creare firme per diversi tipi di comunicazioni utilizzando la lunghezza dei payload sottostanti delle query e delle risposte DNS. Queste firme ci consentono di separare le attività di manutenzione comuni dai comandi significativi del server e di isolare l'uso di tipi di eventi personalizzati. Questi possono essere usati per profilare il comportamento generale di un client e server Pupy osservato passivamente, incluso Decoy Dog.

Nella Figura 5 qui sotto, mostriamo una mappa di calore delle lunghezze dei payload osservate nelle query del client e nelle risposte del server nei nostri dati di Pupy. Mentre le lunghezze dei server hanno più variazioni a causa degli argomenti dei comandi e dei comandi concatenati, le comunicazioni del client sono ben definite. Per le comunicazioni di profilazione, utilizziamo la lunghezza del payload sottostante, inclusi i checksum e le informazioni sui nodi. Di conseguenza, ad esempio, il riconoscimento client (Ack) è lungo 19 byte e l'Ack del server è lungo 6 byte. L'Appendice D contiene tabelle per le lunghezze comuni dei payload di client e server e la loro relazione con i comandi.

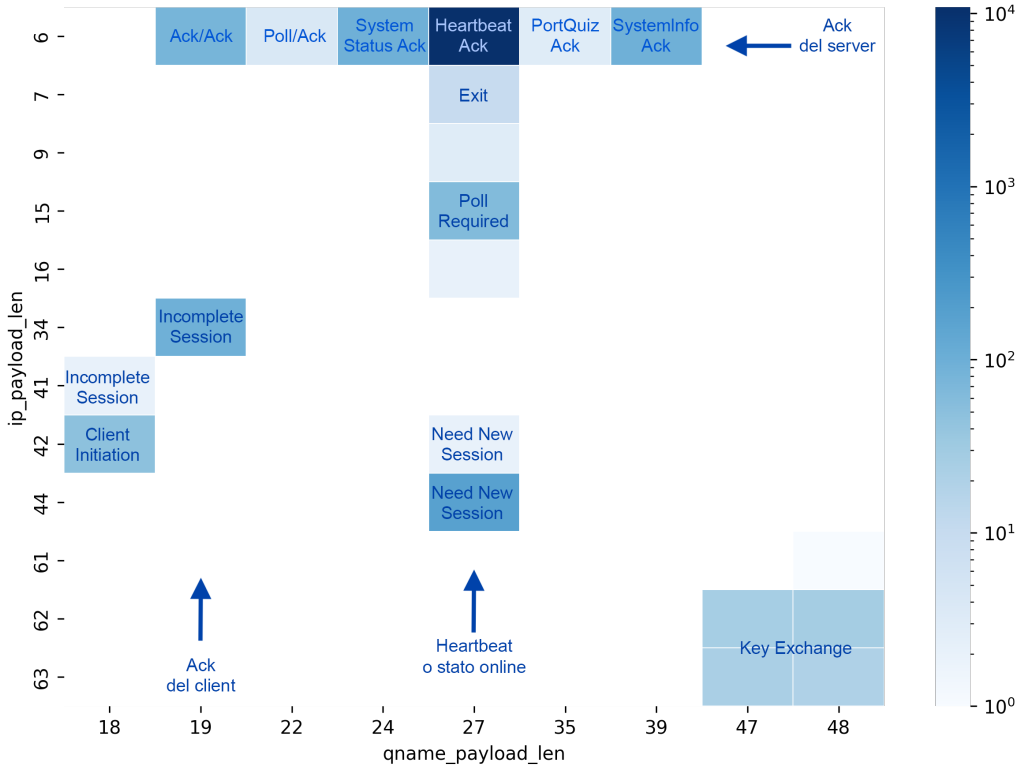


Figura 5. Una distribuzione annotata delle coppie di lunghezze di payload comuni osservate nel traffico Pupy. Il payload è costituito dai dati crittografati trasmessi nella query o nella risposta. Questo grafico non include comandi DNS C2 complessi dal server e le celle senza annotazione non sono completamente identificate. La lunghezza è in byte.

Decoy Dog

Le comunicazioni di Decoy Dog sono state osservate non solo presso i resolver Infoblox, ma anche presso molti resolver pubblici e commerciali. Per comprendere meglio le operazioni di Decoy Dog e in che modo il toolkit differisce da Pupy, abbiamo usato altre raccolte DNS passive per aumentare le nostre. In totale, la nostra analisi copre oltre 15 milioni di eventi DNS nel periodo di tempo che va dal 29 marzo 2022 fino al 16 giugno 2023. Inoltre, abbiamo sondato attivamente i name server e confrontato il traffico DNS raccolto passivamente con quello generato dal nostro client e server Pupy.

Abbiamo usato una serie di tecniche per comprendere meglio Decoy Dog e le sue operazioni. Abbiamo anche eseguito il reverse engineering di campioni trovati nel repository pubblico VirusTotal, che ha convalidato le nostre scoperte DNS e svelato altre funzionalità. Nelle sezioni che seguono, descriveremo la nostra analisi in dettaglio e mostreremo i risultati. I punti salienti di questo lavoro sono:

- Decoy Dog non è Pupy, ma un grande refactor che estende in modo significativo le capacità del malware e aiuta a garantire la persistenza su un dispositivo compromesso.
- È gestito da una manciata di attori, che impiegano TTP distinte e hanno risposto in modo diverso alla nostra rivelazione del toolkit dell'aprile 2023.
- Il numero complessivo di dispositivi interessati è ridotto, con un minimo di quattro su un singolo controller.

- I nuovi controller registrati da aprile 2023 si sono adattati per mitigare le caratteristiche delineate nel nostro documento originale; ciò include meccanismi di geofencing per limitare le risposte agli indirizzi IP dei client a determinate posizioni.
- L'analisi DNS si è rivelata uno strumento potente non solo per rilevare Decoy Dog, ma anche per comprenderne l'uso e separarlo da Pupy, che, combinato con il reverse engineering selettivo, fornisce un quadro solido di Decoy Dog e della minaccia che rappresenta.

SCAMBI DI CHIAVI

Come descritto in precedenza, una sessione inizia quando lo scambio di chiavi è completato e il valore SPI è impostato. In teoria, una singola sessione crittografata può continuare indefinitamente, ma in pratica ci sono una serie di condizioni in base alle quali il controller richiederà che venga stabilita una nuova sessione. Pertanto, una singola istanza in esecuzione del client può avere molte sessioni. Utilizzando le firme del payload Pupy, possiamo determinare quando sono state generate le chiavi condivise tra un client e un server e fare stime approssimative del numero di inizializzazioni del client, da una nuova compromissione o da un riavvio del client, per ciascun controller nel tempo.

La Figura 6 di seguito mostra la sequenza temporale degli scambi di chiavi per diversi controller Decoy Dog. Esistono lacune negli scambi di chiavi osservati per alcuni controller. L'ultimo scambio di chiavi per claudfront[.]net è stato osservato nel dicembre 2022, anche se l'attività dei client non solo è proseguita, ma è aumentata nel 2023; oltre il 70% di tutti i valori SPI unici è stato osservato per la prima volta nel 2023. Allo stesso modo, il controller allowlisted[.]net non ha avuto scambi di chiavi da dicembre 2022 fino a dopo la nostra divulgazione nell'aprile 2023. Infine, cbox4[.]ignorelist[.]com mostra anche un lungo periodo di tempo senza scambi di chiavi, con un numero limitato che si verifica direttamente prima che il dominio smettesse di funzionare. Sospettiamo che gli attori abbiano riconfigurato i client per eseguire lo scambio di chiavi su un trasporto diverso da DNS.

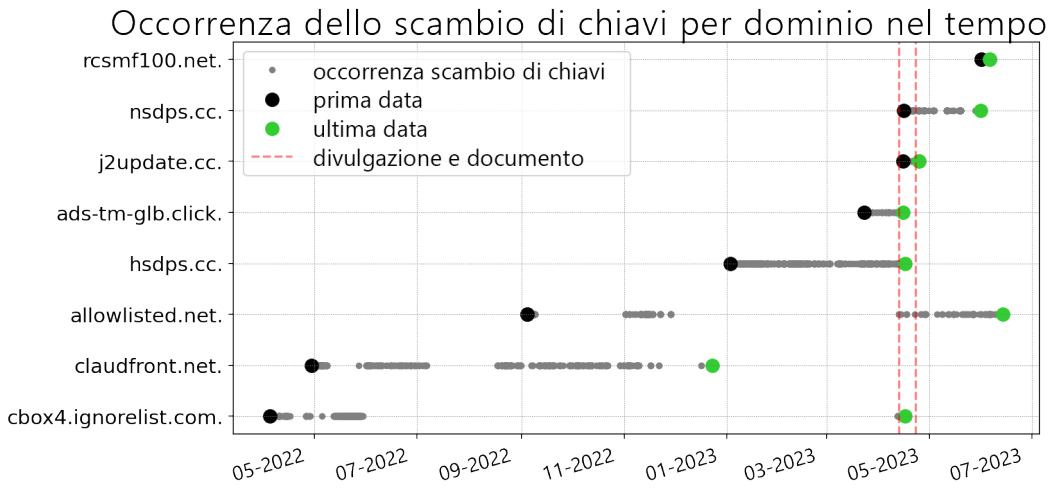


Figura 6. Cronologia degli scambi di chiavi osservati per domini Decoy Dog selezionati.

TEMPISTICHE DEI CLIENT

Oltre al numero complessivo di client, volevamo determinare quanti client attivi gestiva ogni controller alla volta e per quanto tempo i client comunicavano attivamente con il server. È stato utilizzato il metodo di raggruppamento dei valori nonce descritto nell'Appendice C. Questa analisi ha fornito informazioni chiave sulle operazioni di Decoy Dog per un lungo periodo di tempo, come dimostrato nei grafici che seguono. In particolare:

- Tutti i controller gestiscono un piccolo numero di client contemporaneamente, con alcuni che ne controllano solo quattro e tutti probabilmente meno di cinquanta.
- Il dominio originale, cbox4[.]ignorelist[.]com, è uno dei controller più grandi e presenta un aumento dei client in più momenti. Mantiene inoltre un numero limitato di client con esecuzione molto prolungata.
- Il secondo controller da osservare, claudfront[.]net, ha registrato un accentuato aumento dell'attività nel febbraio 2023.
- Il terzo controller da osservare, allowlisted[.]net, ha costantemente mantenuto un piccolo numero di client simultanei.
- I controller ads-tm-glb[.]click e hsdps[.]cc hanno trasferito i client a nuovi controller in seguito alla nostra divulgazione.
- Claudfront[.]net e allowlisted[.]net non hanno modificato le operazioni in risposta alla nostra divulgazione, cbox4[.]ignorelist[.]com ha cessato le operazioni e sia hsdps[.]cc che ads-tm-glb[.]click hanno trasferito i client a nuovi domini.

Sebbene sia difficile stimare il numero totale di client in qualsiasi momento, il numero ridotto di client attivi contemporaneamente indica che queste operazioni sono altamente mirate. Spiega anche perché i fornitori di sicurezza non hanno rilevato l'attività e non hanno ancora trovato i dispositivi infetti. I client infetti sono presenti in un numero molto ridotto di reti, apparentemente quelle che non sono in grado di identificare e bloccare le comunicazioni C2 nel DNS.

Nei diagrammi a linee che seguono, rappresentiamo l'attività di un singolo client come una linea e la chiamiamo thread del client. L'asse y mostra i thread di client distinti identificati da una catena di nonce. Quando un client Pupy viene riavviato, attraverso un riavvio o in altro modo, verrà generato un nuovo nonce e verrà osservato un nuovo thread. In alcuni diagrammi, ci sono chiare interruzioni dell'attività che probabilmente indicano il riavvio del client. L'asse x indica il tempo.

La Figura 7 mostra l'attività del client per il dominio iniziale di Decoy Dog cbox4[.]ignorelist[.]com. Il primo thread del client inizia alla fine di marzo 2022 e il thread più lungo è durato quasi un anno. Possiamo vedere che questo controller inizialmente aveva solo pochi client, ma a metà maggio 2022 si è verificato un cambiamento che ha portato a quasi 40 client attivi contemporaneamente. Aumenti simili nei thread dei client si sono verificati periodicamente, con il maggiore aumento mensile nell'agosto 2022; tuttavia, con l'inizio di nuovi thread dei client, altri sono terminati. Durante l'intero anno di attività, il numero di client simultanei sembra essere sempre inferiore a 50. È inoltre possibile vedere dalla Figura 7 che un quarto dei thread dei client è persistito per sei mesi o più, coerentemente con un'operazione sostenuta. Tutte le comunicazioni sono cessate in seguito al post di LinkedIn e non sono state più osservate.

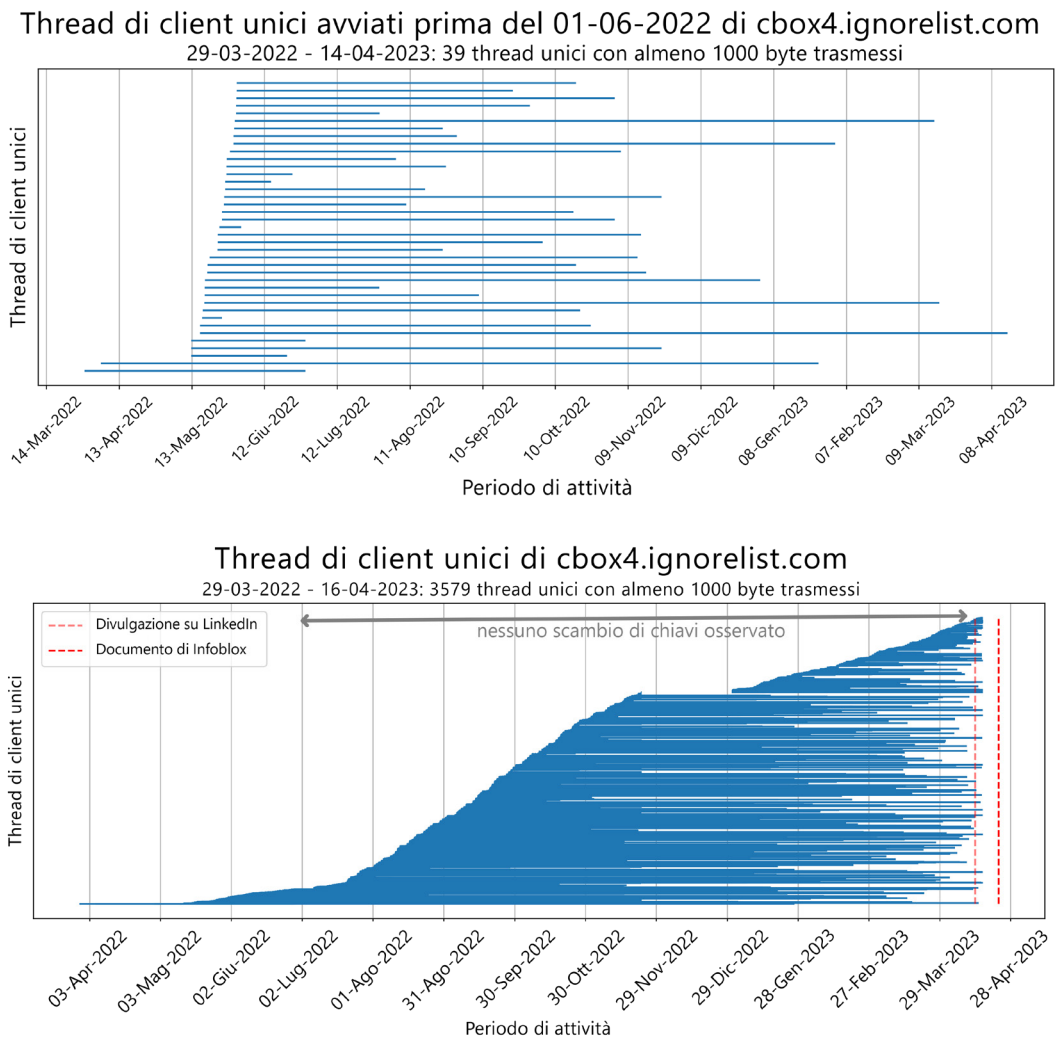


Figura 7. La figura in alto mostra i client presenti prima del 1° giugno 2022, mentre la figura in basso mostra i thread dei client nel tempo.

L'attività DNS per claudfront[.]net, cronologicamente il secondo dominio di Decoy Dog ad apparire, è abbastanza diversa da cbox4. Come mostrato nella Figura 8 di seguito, fino all'inizio di febbraio 2023 c'erano meno di dieci client attivi contemporaneamente su questo controller. Successivamente, il numero di client è aumentato sostanzialmente, anche se non nella misura in cui ci si aspetterebbe da un'infezione diffusa. Il momento di questo aumento è poco prima dell'invio di un campione binario contenente il dominio del controller a VirusTotal il 13 febbraio.[15] A differenza di cbox4[.] ignorelist[.]com, non ci sono stati cambiamenti degni di nota nelle query di claudfront[.]net a seguito della nostra divulgazione.

15 0375f4b3fe011b35e6575133539441009d015ebecbee78b578c3ed04e0f22568, inviato per la prima volta nel 2023

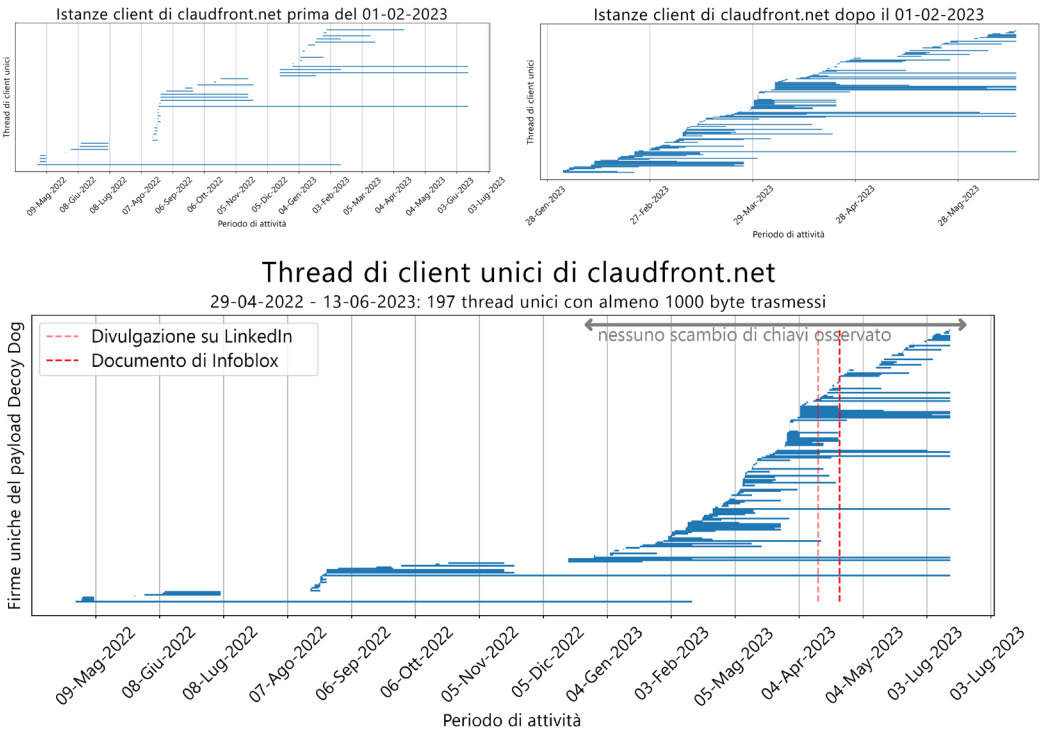


Figura 8. Thread dei client per claudfront[.]net basati sui thread di nonce nel tempo. C'è un cambiamento significativo all'inizio di febbraio 2023, che viene ingrandito con immagini separate che mostrano periodi di tempo distinti.

Il terzo dominio, allowlisted[.]net, mostra un'altra variazione nel comportamento. In questo caso, il numero di client è costantemente ridotto: meno di dieci in un dato momento. A differenza di claudfront[.] net non ci sono cambiamenti nel febbraio 2023 e non è disponibile alcun campione binario contenente allowlisted[.]net noto. Non si vedono scambi di chiavi da metà novembre 2022 fino a poco dopo la nostra divulgazione, il che coincide con la brusca fine dell'attività dei client e il riavvio di diversi thread nell'aprile 2023, come mostrato nella Figura 9.

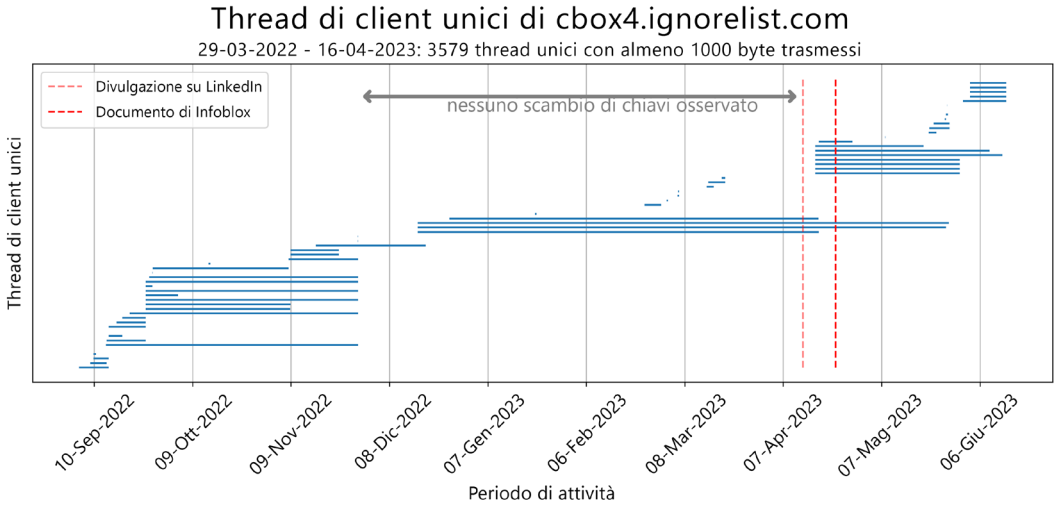


Figura 9. Thread dei client per allowlisted[.]net. C'è un numero molto piccolo di client su allowlisted[.]net storicamente e questo non è cambiato dopo la divulgazione.

Infine, abbiamo osservato attività correlate da parte di hsdps[.]cc, nsdps[.]cc, ads-tm-glb[.] click e j2update[.]cc. I domini hsdps[.]cc e ads-tm-glb[.]click hanno cessato di operare dopo la nostra divulgazione sui social media, ma molti dei loro client sono stati trasferiti rispettivamente a nsdps[.]cc e j2update[.]cc. Lo abbiamo scoperto creando catene di nonce

su tutti i domini nel corso del tempo e identificando i thread che iniziavano a comunicare con un controller e terminavano con un altro.[16]

I nuovi domini, nsdps[.]cc e j2update[.]cc, sono stati registrati meno di 48 ore dopo i nostri annunci sui social media. Possiamo vedere dai diagrammi dei thread dei client che un set di domini cessa l'attività mentre altri vengono avviati. I controller hanno iniziato a comunicare attivamente con i client quasi subito dopo. Dopo la scoperta del trasferimento dei client tramite analisi DNS, abbiamo trovato prove in esempi binari di un comando per apportare questa modifica, come descriveremo più avanti.

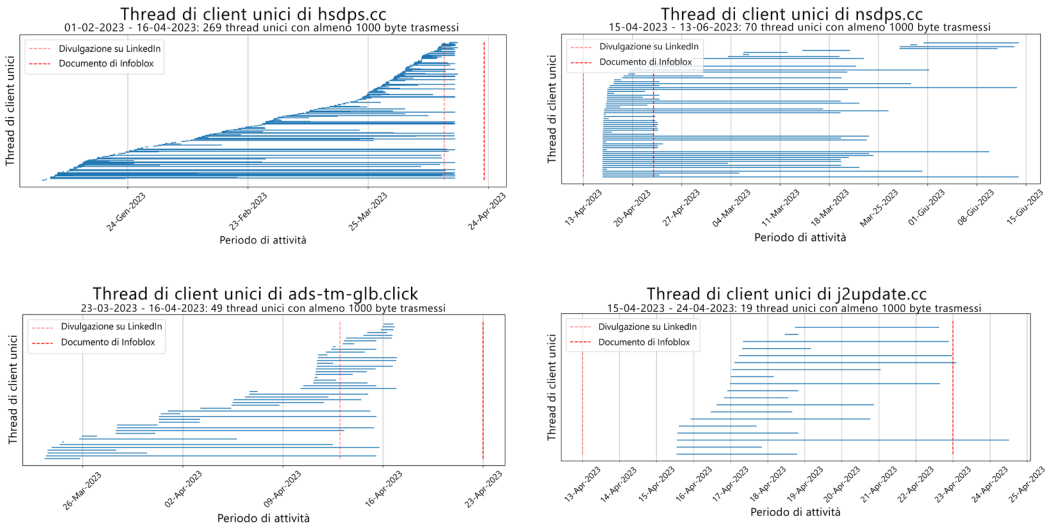


Figura 10. Un confronto temporale di quattro domini di controller Decoy Dog. I controller hspds[.]cc e ads-tm-glb[.]click interrompono le comunicazioni a seguito della divulgazione di Infoblox e i domini nsdps[.]cc e j2update[.]cc iniziano le comunicazioni. Abbiamo anche osservato i trasferimenti di client tra questi domini.

Dal nostro documento originale, abbiamo visto l'attivazione di controller aggiuntivi, ognuno con un numero molto ridotto di client. Il comportamento dei client mostrato qui, insieme alla risposta al nostro annuncio, indica che il toolkit Decoy Dog viene utilizzato da più attori.

FIRME DEL PAYLOAD DI DECOY DOG

Abbiamo decodificato le lunghezze del payload di client e server di 15,5 milioni di risposte alle query osservate nel pDNS globale in un periodo di 13 mesi. Abbiamo poi confrontato le firme di Pupy per i payload client-server con i dati osservati da Decoy Dog per capire il comportamento dei server. Anche se abbiamo scoperto che le distribuzioni complessive del traffico erano in linea con Pupy, c'erano delle differenze nette. I client Decoy Dog utilizzano un set più ampio di richieste, o vocabolario, rispetto a quello che si trova in Pupy predefinito.

La Figura 11 mostra le distribuzioni relative delle coppie di lunghezze del payload in tutti i sistemi Decoy Dog. Utilizzando le nostre firme Pupy, come dettagliato nell'Appendice D, possiamo trarre alcune conclusioni immediate:

- Erano presenti più dei nove payload client previsti.
- C'erano lunghezze di payload del server che non avevamo osservato nel nostro laboratorio.
- La maggior parte delle comunicazioni riguarda il mantenimento della sessione e lo scambio di chiavi.

16 La probabilità che ciò avvenga in modo casuale con un nonce casuale a 32 bit è estremamente bassa e il numero di "trasferimenti" di nonce da un controller all'altro per questi domini era elevato.

- Un'ampia percentuale delle query ai server Decoy Dog ha ricevuto una risposta di errore e ha mostrato variazioni coerenti con la scansione da parte di una terza parte piuttosto che di un vero client. La maggior parte di queste è avvenuta dopo i nostri annunci.

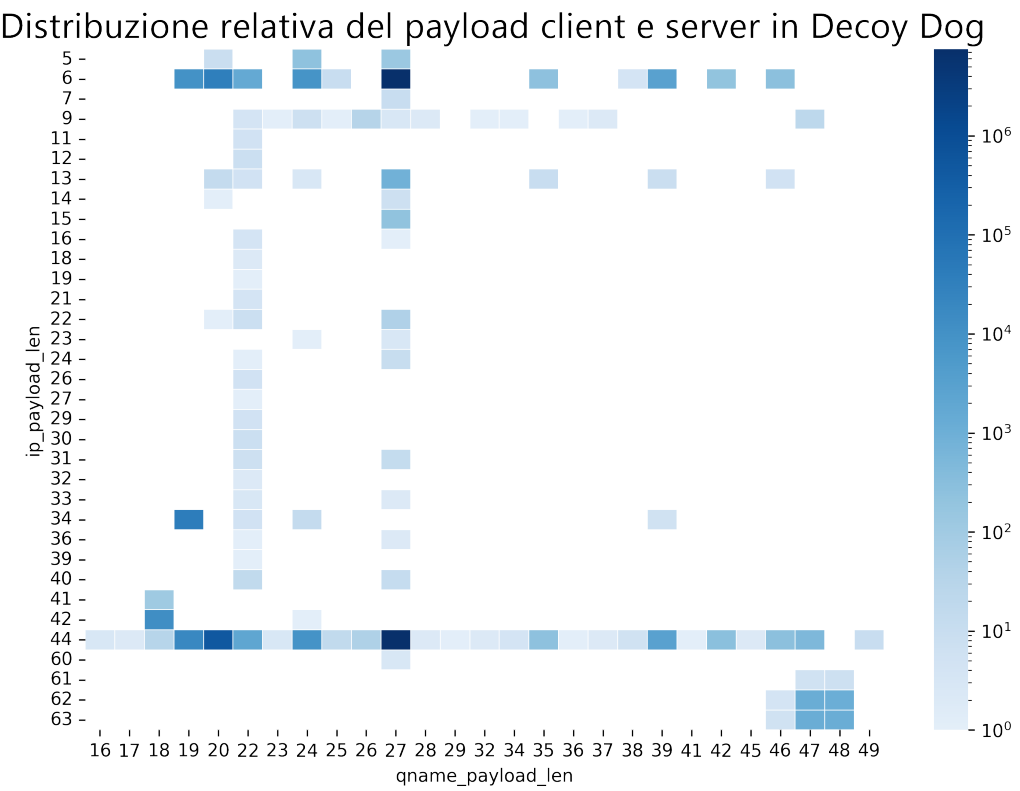


Figura 11. La distribuzione relativa delle lunghezze del payload di client e server osservata nelle comunicazioni Decoy Dog.

I payload unici per i client includevano le lunghezze 20, 25, 38, 42 e 46. Alcuni di questi possono essere associati a una diversa configurazione delle chiavi o a una modifica dei parametri di polling; non possiamo determinare quale fosse la comunicazione, ma la variazione esiste. Inoltre, c'erano lunghezze del payload di risposta aggiuntive oltre a quelle osservate in Pupy. In particolare, Decoy Dog ha un payload del server di 13 byte che viene rilevato nel tempo in periodi di attività. Non siamo in grado di determinare cosa sia questo payload, ma è coerente con un singolo comando che richiede 8 byte di dati da trasmettere al client. Abbiamo anche visto una serie di risposte del server contenenti un payload di 5 byte, un'altra lunghezza non osservata nei nostri dati Pupy e indicativa di un singolo comando che non richiede il trasferimento di dati al client. La Figura 12 qui sotto riassume le coppie di payload uniche trovate in Decoy Dog e non viste nei nostri esperimenti con Pupy.

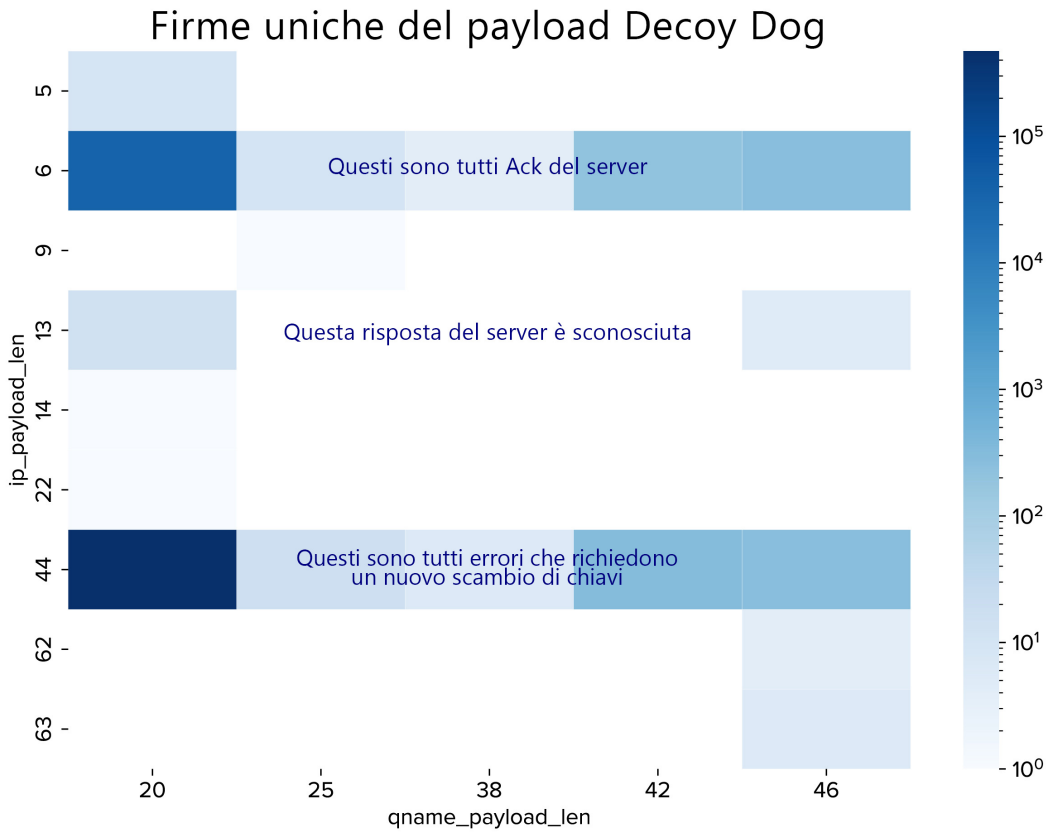


Figura 12. Un riepilogo delle coppie di lunghezza del payload client-server osservate in Decoy Dog e non presenti nelle comunicazioni predefinite di Pupy.

Abbiamo utilizzato anche le serie temporali per identificare le modifiche alle configurazioni predefinite. In una sessione Pupy prestabilita, il client stabilirà una connessione ogni 30 secondi. Utilizzando l'analisi statistica sulla variazione delle query heartbeat del client, sono stati trovati intervalli di heartbeat di 2 minuti e 30 minuti oltre ai 30 secondi predefiniti.

Come risultato di questa analisi, siamo stati in grado di comprendere la natura delle comunicazioni per ogni dominio Decoy Dog, separando la manutenzione ordinaria dai comandi di accesso remoto. Siamo stati anche in grado di isolare le probabili personalizzazioni di Pupy utilizzate all'interno dei sottoinsiemi di server Decoy Dog. Abbiamo scoperto che la stragrande maggioranza del traffico di Decoy Dog è costituito da riconoscimenti ed errori di routine e che le comunicazioni di errore erano sproporzionate rispetto a ciò che ci aspettiamo di vedere in base alle osservazioni di Pupy. Condivideremo i risultati della nostra indagine su questo fenomeno delle risposte di errore nella prossima sezione.

COMPORTAMENTO JOLLY E GEOFENCING

Nel nostro documento tecnico originale abbiamo riportato che i server di Decoy Dog hanno risposto a query DNS riprodotte. Questo lascia perplessi. Mentre cercavamo di capire quando e come Decoy Dog avrebbe risposto a una query originariamente fatta giorni o settimane prima, abbiamo scoperto un comportamento ancora più sorprendente. Diversi server Decoy Dog non solo rispondono alle query riprodotte, ma rispondono a qualsiasi query che sia coerente con la codifica Pupy. Nel DNS, questa è una risposta jolly. Mentre un normale server Pupy restituirebbe una risposta NXDOMAIN o SERVFAIL, il server Decoy Dog restituisce in genere 15 indirizzi IP.

La Figura 13 di seguito mostra le risposte alle query randomizzate. In questo caso, abbiamo inserito le frasi "wild" e "wildcard" (jolly) nel nome della query e abbiamo ricevuto 15 risposte da due diversi server Decoy Dog. Le risposte sono diverse per ogni query e sono conformi allo schema di codifica Pupy. Attraverso la nostra ricerca, abbiamo appreso che Decoy Dog gestisce quasi tutti gli errori in questo modo invece di restituire le risposte NXDOMAIN previste. Consultare l'Appendice E per ulteriori informazioni sulla gestione degli errori.

```
; <<>> 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. Comportamento di risposta jolly da due server autorevoli di Decoy Dog. In entrambi i casi i server hanno risposto con 15 indirizzi IP coerenti con la codifica Pupy alla stessa query randomizzata contenente le stringhe "wild" e "wildcard".

Ancora più sorprendente, alcuni server Decoy Dog rispondono anche in modo diverso a seconda dell'indirizzo IP del resolver ricorsivo che effettua la query per conto del client. Nella Figura 14 viene illustrata la riproduzione di una query al dominio Decoy Dog nsdps[.]cc, che

è avvenuta originariamente diverse settimane prima. Quando abbiamo effettuato la query tramite i resolver pubblici di Yandex, abbiamo ricevuto una risposta contenente 15 indirizzi IP. Abbiamo anche ricevuto 15 indirizzi IP dai resolver pubblici russi di TimeWeb. Tuttavia, degli oltre trenta resolver pubblici che abbiamo provato, nessun altro ha restituito una risposta. Questo tipo di comportamento è coerente con il geofencing, in cui un server risponde alle richieste DNS in base alla geolocalizzazione dell'indirizzo IP. Abbiamo scoperto questo comportamento nel giugno 2023 e abbiamo osservato che alcuni server rispondevano solo quando indirizzavamo le query DNS tramite indirizzi IP russi, mentre altri rispondevano a qualsiasi query ben formata da qualsiasi luogo. Questo tipo di risposta selettiva assicura che il controller comunichi solo con i client che sembrano essere in Russia. Sappiamo che questa funzionalità è stata aggiunta dopo la divulgazione, perché i controller avevano precedentemente risolto le query dai resolver ricorsivi di 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. Un confronto tra le risposte a una query Decoy Dog riprodotta dai resolver pubblici Yandex, dai resolver pubblici Hurricane Electric e dal resolver autorevole. Queste query sono state fatte in successione tramite un browser Tor. Solo la richiesta tramite Yandex ha ricevuto una risposta.

Quando viene effettuata una richiesta per un nome di dominio che non può essere decodificato utilizzando la codifica predefinita di Pupy (abbiamo aggiunto caratteri extra per questo test), i server di nsdps[.]cc restituiscono un indirizzo IP che è essenzialmente un sinkhole. Come mostrato nella Figura 15 di seguito, abbiamo leggermente modificato la query in modo che non possa essere decodificata correttamente. In questo caso, è stato restituito un indirizzo IP casuale compreso nell'intervallo 172.0.0.0/8. Normalmente Pupy restituirebbe una risposta 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 query per un nome di dominio Pupy non valido per il controller nsdps[.]cc restituirà un indirizzo IP casuale nell'intervallo 172.0.0.0/8 anziché la risposta NXDOMAIN prevista.

Parte di questo comportamento può essere spiegato come un artefatto della risoluzione DNS da parte del client. Quando un host viene interrogato nel DNS, alcuni resolver tenteranno di risolvere nomi di dominio potenzialmente correlati per prepararsi a potenziali query future. Ad esempio, un resolver ricorsivo che riceve una query per www[.]baddomain[.]com potrebbe tentare di risolvere baddomain[.]com oltre a www[.]baddomain[.]com. Abbiamo riscontrato questo comportamento nel nostro server Pupy durante il routing delle query del client tramite alcuni resolver pubblici.

RISPOSTE A ETICHETTA SINGOLA

Per impostazione predefinita, Pupy rifiuta le richieste in ingresso alle etichette che non corrispondono alla struttura di una comunicazione client o di una query ping stabilita. Tuttavia, come abbiamo spiegato nella sezione "Gestione speciale dei nomi di dominio", la funzione di richiesta di attivazione DNS consente all'attore di configurare il server Pupy in modo che risponda alle query per le risorse personalizzate. Nei log pDNS globali sono state identificate le query con un sottodominio a una singola etichetta. L'unico sottodominio di questo tipo era "m" e abbiamo ipotizzato che la risoluzione di questi domini fosse possibile tramite la funzione di attivazione. Per la natura della funzione hash dell'attivatore, per queste query deve essere restituito un singolo indirizzo IP statico. Abbiamo riscontrato questo comportamento in 4 domini: hsdps[.]cc, nsdps[.]cc, j2update[.]cc, e ads-tm-glb[.]click, ed è un'altra caratteristica condivisa di questo insieme di domini che non si vede in nessun altro controller. Ognuno di questi ha restituito un singolo indirizzo IP; tuttavia, invece dell'indirizzo IP statico previsto, abbiamo trovato 104 indirizzi univoci nelle risposte. Questo sembra indicare una differenza nella funzione rispetto al Pupy predefinito, ma non ne conosciamo lo scopo.

ANALISI DEI CAMPIONI BINARI

Dopo le nostre scoperte sul DNS, abbiamo esaminato i campioni binari disponibili in VirusTotal per determinare se l'origine delle differenze rispetto a Pupy fosse immediatamente evidente. Analizzando le importazioni e le tabelle delle funzioni di due campioni di Decoy Dog, abbiamo identificato una firma univoca specifica per gli impianti Decoy Dog che ci ha permesso di scoprire altri campioni di Decoy Dog. Il reverse engineering di questi campioni ha ulteriormente confermato le nostre scoperte secondo cui Decoy Dog è sostanzialmente diverso da Pupy e che il codice più maturo potrebbe essere stato creato da un secondo sviluppatore. Il client viene aggiornato a Python 3.8 e include una serie di nuovi trasporti, crittografia aggiornata, comandi personalizzati e nuove funzionalità DNS. Il campione relativo a un controller, claudfront[.]net, contiene funzionalità non presenti negli altri. Questa sezione descrive alcuni dei principali risultati e del processo; ulteriori dettagli tecnici sono disponibili nell'Appendice F. I dati analitici relativi ai file binari verranno aggiunti anche al nostro repository GitHub.

Il primo campione è stato caricato a settembre 2022 e gli altri sono stati caricati nel 2023; tre di loro dopo la nostra divulgazione. Abbiamo estratto e confrontato le configurazioni dei diversi campioni di Decoy Dog, il che ha dimostrato che le chiavi di crittografia differiscono tra i server. Tutti i campioni che hanno comunicato con cbox4[.]ignorelist[.]com contengono le stesse chiavi RSA e SSL, indicando che l'esistenza di campioni diversi non è correlata alle modifiche delle chiavi del server. Un elenco completo delle chiavi decrittografate è disponibile nel repository Github dettagliato nell'Appendice I. Il primo certificato SSL tra gli esempi è stato generato il 26 dicembre 2021 e appartiene a cbox4[.]ignorelist[.]com, il primo controller osservato.

Una scoperta significativa è stata che Decoy Dog include codice personalizzato nel suo client Pupy che consente agli aggressori di inviare ed eseguire moduli Java in fase di esecuzione iniettandoli in un thread JVM (Java Virtual Machine). Questa funzionalità non esiste nelle versioni standard di Pupy. Questo codice è stato trovato in tutti i campioni di Decoy Dog ed è identico in tutti i casi. Le funzioni binarie rimanenti in tutti i campioni noti di client Decoy Dog sono identiche alle funzioni dei client Pupy di base.

L'inclusione dei moduli Java solleva più domande che risposte. Per impostazione predefinita, Pupy è già molto capace e supporta l'uso di moduli Python pronti all'uso. L'espansione di queste funzionalità e la scrittura di moduli Python è un processo semplice che non richiede modifiche sul lato server o modifiche al file binario del client. Si potrebbe facilmente creare un modulo Python per eseguire moduli Java. Al contrario, l'injection di moduli Java in fase di esecuzione senza utilizzare jni.h (o il resto dell'API Java/C standard) non è un compito banale e richiede conoscenze specialistiche. Quindi, è probabile che l'aggiunta di questi moduli Java consenta agli aggressori di colpire sistemi che non eseguono Python, sistemi che eseguono una macchina virtuale Java privilegiata o non monitorata, o scenari in cui gli aggressori mirano a evitare di lasciare prove sulla macchina, non creando file.

I client dispongono inoltre di nuove funzionalità, maturate nel tempo. Il software client viene creato eseguendo il marshalling di un file di configurazione Python in un determinato file binario. Il file di configurazione include le impostazioni, tutte le chiavi necessarie per le comunicazioni (RSA, certificati SSL, password, ecc.), e i moduli Python del client. I moduli presenti negli esempi, che vengono decompressi ed eseguiti dai dispositivi compromessi, sono molto diversi dal codice Pupy disponibile al pubblico.

L'estrazione e l'analisi dei moduli integrati descrive una storia affascinante di sviluppi e modifiche personalizzate di Decoy Dog. Innanzitutto, un numero considerevole di moduli Pupy è stato semplicemente rimosso da Decoy Dog, forse perché gli aggressori li hanno ritenuti inutili. In secondo luogo, campioni simili presentano un gran numero di differenze nei moduli, a volte con capacità molto diverse. In terzo luogo, l'elevato numero di modifiche e la complessità aggiunta dalle nuove funzionalità mostrano un notevole tempo di sviluppo e di messa a punto delle risorse di Pupy. Inoltre, la base di codice e i moduli di Pupy sono stati trasferiti da Python 2.7 a Python 3.8, il che ha migliorato la qualità del codice, la stabilità delle operazioni di memoria e la compatibilità con Windows. I campioni includono una versione del client che cambia da 3 a 4 nel corso del tempo; il client Pupy più recente disponibile è la versione 2. Una sequenza temporale che riassume le date di invio rispetto alla maturità del codice e alle caratteristiche chiave si trova nella Figura 16 qui sotto.

Analizzando la natura e il numero di moduli modificati, siamo stati in grado di identificare che dal punto di vista della maturità del codice, il campione con l'hash ad186df91282cf78394ef3bd60f04d859bcacccbcdcbf620cc73f19ec0cec64 è il primo binario di Decoy Dog disponibile pubblicamente. Comunica con il name server cbox4[.] ignorelist[.]com. Sebbene condivida la maggior parte del codice con Pupy, questo esempio non è stato caricato su VirusTotal fino al 27 aprile 2023, diversi giorni dopo la pubblicazione del nostro documento. Tuttavia, in base al certificato SSL incluso, questo campione potrebbe risalire al dicembre 2021. Lo sviluppatore ha aggiunto funzionalità di polling specifiche, una funzione XOR, nuovi trasporti e il supporto completo per le comunicazioni di rete multithread. È interessante notare che alcuni nuovi moduli si rivolgono specificamente

a Win32, anche se finora tutti gli esempi sono librerie Linux. In questo eseguibile, il codice responsabile della gestione delle comunicazioni DNS è lo stesso del Pupy predefinito.

Col passare del tempo, i campioni che comunicano con cbox4[.]ignorelist[.]com sono diventati più complessi. In una serie di tre campioni, è stato aggiunto un numero crescente di moduli di comunicazione, incluso un intero modulo per comunicare utilizzando flussi bidirezionali su HTTP sincrono (BOSH, Bidirectional-streams Over Synchronous HTTP), oltre a riscritture complete dei moduli SSL, TCP e UDP. Gli attori dietro Decoy Dog hanno anche aggiunto una serie di script per trasferire i moduli di exploit e comunicazione esistenti su piattaforme Windows, hanno riscritto il client picocmd responsabile delle comunicazioni DNS e hanno implementato una serie di miglioramenti della qualità di vita e della stabilità del vecchio codice. I riferimenti a Windows nel codice suggeriscono l'esistenza di un client Windows aggiornato che include le nuove funzionalità di Decoy Dog, sebbene tutti i campioni attuali siano destinati a Linux.

Le versioni successive includono anche un modulo di emergenza che consente a una macchina compromessa di contattare un server DNS di terze parti se al malware viene impedito di comunicare con il server C2 per un periodo di tempo prolungato. Questo modulo utilizza un DGA per selezionare i domini su cui il client può eseguire query all'interno dei servizi DNS dinamici gratuiti. Queste versioni consentono anche il bootstrapping per individuare il controller C2, la creazione di domini beacon e l'incorporazione di query CNAME nel servizio di emergenza. I meccanismi di persistenza estesi, disponibili a partire dalla versione 3 del client, sono funzionalità spesso associate alle operazioni di intelligence piuttosto che a quelle condotte da attori o red team motivati finanziariamente.

Il codice più maturo, che si connette al controller claudfront[.]net, include due nuovi comandi chiamati AlterDnsCncDomain e CompromisedNode. Come descritto in precedenza, abbiamo determinato tramite l'analisi dei valori nonce dei client che alcuni degli attori di Decoy Dog avevano trasferito i client a nuovi controller dopo la nostra divulgazione. Sulla base del codice sorgente Pupy disponibile pubblicamente, non abbiamo visto come ciò sia stato possibile senza l'uso di comandi personalizzati. Sembra probabile che il comando AlterDnsCncDomain sia l'origine di tali transizioni di client e quindi i controller associati a nsdps[.]cc potrebbero utilizzare il codice più avanzato. Il grande distacco di questo codice dal resto può indicare che è stato coinvolto un nuovo sviluppatore. Il codice include la versione 4 del client.

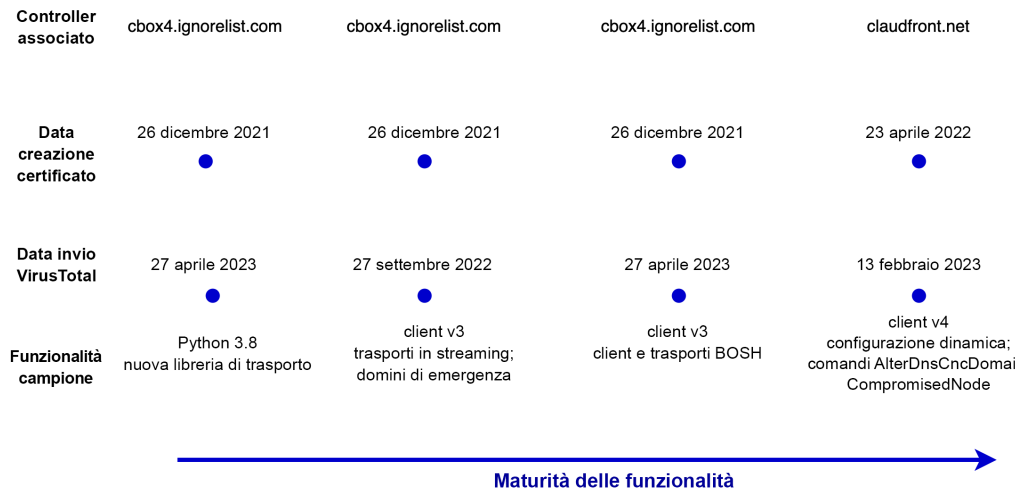


Figura 16. Una sequenza temporale degli invii a VirusTotal relativi a Decoy Dog e della maturità del codice.

Vale la pena notare che, nonostante tutti i miglioramenti di Decoy Dog, le regole YARA sviluppate per le versioni più basilari di Pupy riescono comunque a rilevare il malware. Tuttavia, non sono in grado di rilevare che i campioni si discostano in modo sostanziale dal codice e dalle funzionalità note. Ciò potrebbe portare i ricercatori di malware a supporre erroneamente che i campioni di Decoy Dog siano solo Pupy di base poiché entrambi i tipi di malware sono contrassegnati dalla stessa regola. Per questo motivo, abbiamo incluso una nuova regola YARA per Decoy Dog nell'Appendice G.

CONFRONTO TRA I CONTROLLER

Infoblox sta attualmente monitorando 21 domini Decoy Dog. Alcuni di questi hanno avuto poca o nessuna attività C2 osservabile e non li riveleremo in questo momento. Alcuni controller sono cambiati dopo la nostra divulgazione iniziale sui social media, mentre gli altri sono cambiati dopo la pubblicazione del nostro primo documento. Tutti hanno risposto interrompendo le operazioni, spostando i client su nuovi controller o modificando il comportamento "ping" descritto nel documento. Alcuni hanno addirittura aggiunto il geofencing. Queste risposte, insieme ad altre TTP utilizzate, ci permettono di concludere che ci sono almeno tre attori che utilizzano il toolkit in questo momento. Nella Tabella 1 seguente abbiamo raggruppato un sottoinsieme di domini controller in base al loro comportamento e a caratteristiche simili.

Gruppo di domini	Caratteristiche
cbox4.ignorelist[.]com	• primo dominio attivo e probabile fonte del toolkit Decoy Dog • disattivato dopo la divulgazione • utilizzo del DNS dinamico Afraid • intervallo di heartbeat di 30 secondi • nessun geofencing • almeno tre iterazioni distinte del software client • osservato da noi per la prima volta alla fine di marzo 2022, ma potrebbe essere stato presente già a dicembre 2021 • client v2 e v3
claudfront[.]net allowlisted[.]net maxpatrol[.]net atlas-upd[.]com	• secondo set di controller attivi, a partire da maggio 2022 • continuazione dell'attività dopo la divulgazione • registrati con Namecheap • query a ping12.<domain> prima che la comunicazione crittografata remota fosse vista per la prima volta • modifica della risposta ping in una risposta NODATA • hosting IP russo • intervallo di heartbeat di 30 secondi • nessun geofencing • client v3 e v4 • ci sono alcune differenze tra allowlisted[.]net e claudfront[.]net che potrebbero indicare attori diversi

hsps[.]cc nsdps[.]cc j2update[.]cc ads-tm-glb[.]click	• terzo set di controller attivi, a partire da dicembre 2022 • spostamento dei client tra i controller dopo la divulgazione • controller originali parcheggiati • intervalli di heartbeat di 2 minuti e 30 minuti • geofencing dopo la divulgazione • modificata la risposta ping a un singolo indirizzo IP di loopback non locale • utilizzo di un'unica etichetta di dominio: m • possibilmente client v4
rcmsf100[.]net	• osservato per la prima volta nel giugno 2023 • condivide l'hosting con allowlisted[.]net • risposta ping di NODATA • geofencing

Tabella 1. Un confronto tra diversi controller Decoy Dog.

DECOY DOG NELLE RETI INFOBLOX

Infoblox ha stabilito che i nostri resolver sono stati attivati da uno scanner di un fornitore di sicurezza che riproduceva le query di Decoy Dog. Una combinazione del comportamento dello scanner e del comportamento di Decoy Dog ha creato il segnale rilevato. La scansione di Internet è diventata un'attività importante e rappresenta oggi una grande quantità di traffico Internet. Viene eseguita sia da attori legittimi che malintenzionati. Uno studio recente ha utilizzato un telescopio darknet per comprendere l'impatto di queste scansioni.[17] Sebbene la maggior parte delle scansioni sia limitata alle scansioni delle porte, che tentano di identificare le porte aperte nello spazio IP globale, esiste un'ampia gamma di altre attività di scansione nell'ambiente. Ad esempio, ci sono scanner che cercano directory aperte e resolver DNS aperti. Alcune organizzazioni documentano in modo completo la propria attività di scansione, ma molte non lo fanno.

Per "scansione aggressiva" si intende un'attività di scansione non autorizzata o ad alto volume che potenzialmente degrada le prestazioni di una rete. Può creare un denial of service per una rete o, come nel caso di Decoy Dog, creare falsi eventi di sicurezza.[18] Una scansione aggressiva avvantaggia l'operatore a scapito delle reti i cui proprietari non hanno acconsentito all'attività. Nell'aprile 2023, i team di sicurezza per le reti con rilevamenti Decoy Dog hanno speso risorse significative nel tentativo di trovare la causa principale di queste query DNS per garantire che i loro sistemi non fossero compromessi. Queste query erano particolarmente allarmanti in quanto originate prevalentemente da firewall e il settore dei firewall ha espresso crescenti preoccupazioni per gli attacchi ai firewall negli ultimi mesi.[19]

17 Aggressive Internet Wide Scanners: Network Impact and Longitudinal Characterization, maggio 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 , ultimo accesso 2023-06-11

19 https://blog.talosintelligence.com/state-sponsored-campaigns-target-global-network-infrastructure/, ultimo accesso 2023-06-11

Il modo in cui le query di Decoy Dog sono arrivate ai nostri resolver e il motivo per cui hanno provocato un segnale simile a un beacon C2 di malware mirato è complicato. Al fine di supportare il riconoscimento da parte dei difensori di un'attività simile, forniremo una breve spiegazione e un'illustrazione nella Figura 17.

Affinché Infoblox riceva le richieste DNS di Decoy Dog, una rete di clienti deve avere Infoblox come provider DNS. Inoltre, il cliente deve disporre di appliance di sicurezza, come i firewall, che abbiano configurato sia il filtraggio degli URL in entrata, sia l'inoltro dei DNS da tale dispositivo ai nostri resolver. Questi criteri sono di per sé restrittivi. Quando vengono soddisfatti, si verifica la seguente sequenza:

- Lo scanner tenta di recuperare il contenuto del malware C2 direttamente da un indirizzo IP all'interno della rete. Lo fa anche se queste comunicazioni DNS C2 non sono contenuti web.
- L'appliance di sicurezza intercetta la richiesta e tenta di risolvere il nome di dominio.
- La richiesta DNS viene inoltrata a Infoblox, che risolve la query e restituisce la risposta. Se il dominio si trova in una blocklist DNS configurata dal cliente, non restituirà risultati.
- Se il dominio scansionato dal fornitore non è Decoy Dog o altro malware, verrà risolto e, a seconda delle regole del firewall, il contenuto del sito web verrà restituito allo scanner.

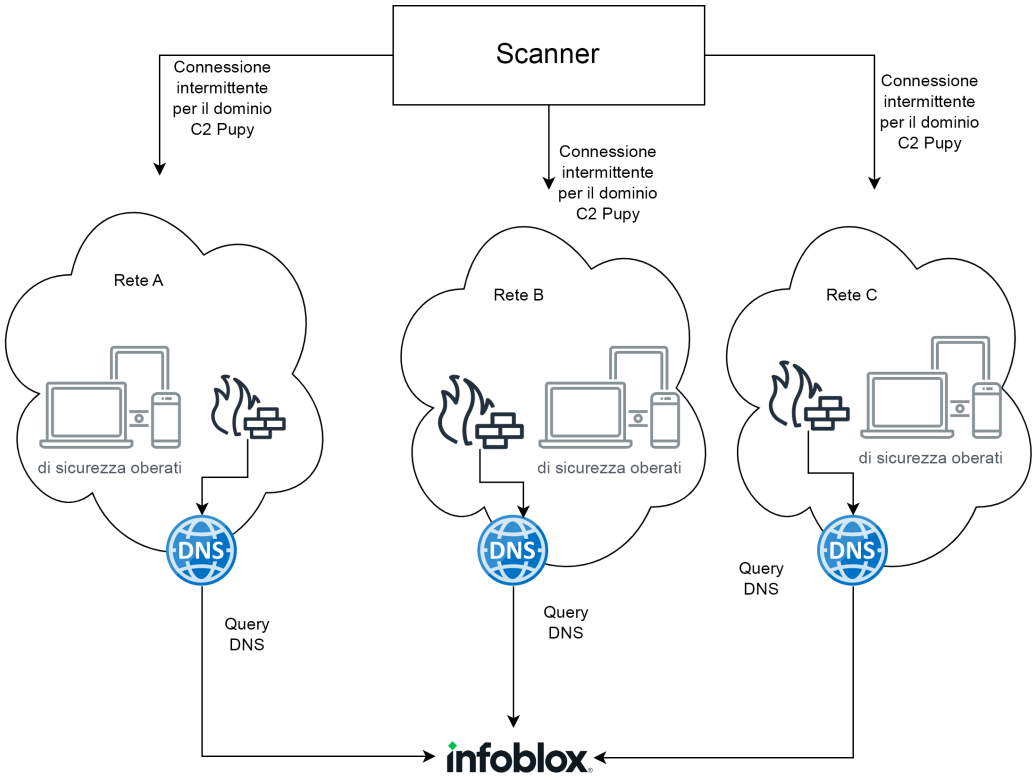


Figura 17. Le query per i domini DNS C2 di Decoy Dog sono state effettuate ai resolver Infoblox da dispositivi all'interno di reti diverse. Queste erano causate da uno scanner commerciale e si attivavano a intermittenza.

Infoblox ha stabilito che il fornitore esegue scansioni anche se l'indirizzo IP non ha porte aperte note e che utilizzerà porte rare oltre alle porte comuni. Non sappiamo come il fornitore decida quali indirizzi IP e porte utilizzare. La conseguenza di una scansione aggressiva e indiscriminata di questo tipo è che i dispositivi molto sensibili possono apparire compromessi quando non lo sono. Sebbene il fornitore sembri eseguire una scansione ampia e costante dei contenuti, Infoblox ha osservato le query DNS solo quando i criteri di

cui sopra sono stati soddisfatti. Di conseguenza, mentre il numero di scansioni effettuate dal fornitore era molto elevato, il che è coerente con una scansione aggressiva, abbiamo risolto solo un piccolo numero di query, a intermittenza nel tempo. Questo tipo di configurazione introduce anche la capacità di un attore di eseguire reconnaissance su determinate reti; lo descriviamo nell'Appendice H.

Infoblox Intelligence conserva i record storici di tutte le attività DNS e li utilizza per creare e mantenere statistiche aggregate sull'attività dei domini nelle nostre reti e nel DNS globale. Utilizziamo queste aggregazioni per identificare un'ampia gamma di minacce, tra cui i comportamenti anomali che sono coerenti con i beacon C2 del malware. In particolare, stiamo cercando domini per i quali le query, nel tempo, si verificano in un numero anomalo di reti di clienti, hanno sottodomini compatibili con l'esfiltrazione dei dati e che hanno un numero basso di query rispetto al comportamento previsto. A tale scopo, utilizziamo le statistiche di ogni dominio che abbiamo osservato nel corso di diversi anni e trilioni di query DNS.

Una volta scoperti, i beacon C2 di Decoy Dog e altri malware sembrano molto sospetti, ma rilevarli è molto difficile. Per sua natura, il traffico DNS è molto variabile e contiene un'ampia percentuale di valori anomali, ovvero domini che vengono visualizzati raramente e hanno una struttura dei nomi di dominio coerente con l'esfiltrazione dei dati. Tuttavia, l'esfiltrazione DNS e il beaconing sono molto rari al di fuori delle attività di pen testing consolidate. Inoltre, la firma DNS del pen testing è ben diversa dai beacon C2 del malware. Nonostante Decoy Dog abbia dimostrato di essere un esempio di DNS C2 da una variante del RAT Pupy, un sistema ad alto volume, sembrava essere un beacon di basso profilo perché il traffico è stato iniettato nelle reti dal fornitore di sicurezza.

Sebbene le query di Decoy Dog ai nostri resolver siano state avviate dallo scanner, sono state rilevate a causa del comportamento insolito dei name server Decoy Dog. Come rivelato nel nostro articolo precedente, i name server di Decoy Dog hanno risposto a query ripetute, anche se a volte in modo intermittente. Ciò non è coerente con Pupy e altri protocolli di comunicazione crittografati. Ora abbiamo inoltre appreso che i controller rispondono a qualsiasi query ben formulata. Il comportamento combinato ha indotto i nostri sistemi a rilevare un beacon intermittente a basso volume. Questo tipo di scansione e il comportamento di inoltro DNS aperto all'interno di una rete comportano ulteriori rischi per la sicurezza di un'azienda. Consentendo a un soggetto esterno di attivare query DNS dall'interno di una rete, un aggressore può effettuare una reconnaissance contro una rete. Descriviamo ulteriormente questa vulnerabilità nell'Appendice H.

Conclusione

Decoy Dog è chiaramente una seria minaccia. Una manciata di autori utilizza il toolkit da oltre un anno con gli unici rilevamenti documentati derivanti dal monitoraggio dei dati DNS. Viene utilizzato in operazioni altamente mirate e abbiamo osservato i suoi controller interagire solo con un numero molto limitato di client attivi. Anche se siamo riusciti a imparare molto su Decoy Dog, rimarrà una minaccia seria fino a quando non verranno identificate e mitigate le vulnerabilità utilizzate per stabilire il suo punto d'appoggio.

Dopo la nostra divulgazione iniziale di Decoy Dog, gli attori di minacce hanno risposto in vari modi per assicurarsi un accesso continuo ai sistemi delle vittime. Queste risposte includevano la modifica del comportamento di risposta DNS dei controller, l'aggiunta di restrizioni di geofencing ai controller e il trasferimento dei client a nuovi controller. Nonostante questi adattamenti, Infoblox ha continuato a seguirli e a imparare di più su Decoy Dog e su come differisce dal RAT Pupy.

Le modifiche apportate a Pupy per creare Decoy Dog sono considerevoli e indicano un sofisticato attore di minacce. Queste modifiche includono:

- Pupy è stato scritto in Python 2.7. Decoy Dog richiede Python 3.8 e include numerosi miglioramenti tra cui la compatibilità con Windows e operazioni di memoria migliorate.
- Pupy ha un vocabolario comunicativo molto limitato. Decoy Dog espande significativamente quel vocabolario attraverso l'aggiunta di più nuovi moduli di comunicazione.
- Decoy Dog risponde alle riproduzioni di precedenti query DNS mentre Pupy non lo fa.
- Pupy non risponde alle richieste DNS jolly, ma Decoy Dog sì. Questo raddoppia essenzialmente il numero di risoluzioni viste nel DNS passivo. In effetti, Decoy Dog risponde alle richieste DNS che non corrispondono alla struttura di una comunicazione valida con un client.
- Decoy Dog aggiunge la capacità di eseguire codice Java arbitrario iniettandolo in un thread JVM e aggiunge una serie di nuovi metodi per mantenere la persistenza sul dispositivo della vittima.

La sofisticazione di queste modifiche rende ancora più curiosa la scelta di Decoy Dog di rispondere a qualsiasi query ben formulata. Anche se a prima vista questa decisione sembrerebbe un errore, è probabile che vi sia una motivazione ancora sconosciuta. Al momento, è solo un altro mistero di Decoy Dog.

In futuro, man mano che questi misteri che circondano Decoy Dog verranno ulteriormente indagati, i difensori dovrebbero tenere presente quanto segue:

- Gli IP sia in Pupy che in Decoy Dog sono dati crittografati. Non rappresentano IP reali utilizzati per la comunicazione. Qualsiasi connessione a IP reali associati al malware è falsa.
- Sebbene gli IP restituiti nelle risposte DNS non siano significativi, le query e le risposte DNS stesse contengono informazioni significative che possono essere utilizzate per il tracciamento. Tuttavia, il volume delle comunicazioni è basso, il che significa che è necessaria una lunga cronologia di log per tracciare le comunicazioni rilevate.
- Le risposte jolly del toolkit, combinate con una scansione aggressiva da parte del fornitore di sicurezza, possono dare l'impressione di una compromissione che non c'è.
- È disponibile una regola YARA in grado di rilevare il client Decoy Dog su un computer vittima. È in grado di differenziare Decoy Dog dalla versione pubblicamente disponibile di Pupy.

Decoy Dog è stato rilevato esclusivamente utilizzando algoritmi di rilevamento delle minacce DNS. Ad oggi, non esiste alcuna divulgazione pubblica che descriva i rilevamenti del malware stesso e l'intera portata delle sue capacità non è ancora nota. Il fatto che abbia operato inosservato per così tanto tempo evidenzia una debolezza che si verifica quando il settore si affida eccessivamente al rilevamento basato sul malware. Il rilevamento e la risposta DNS sono attualmente l'unico modo per difendersi da Decoy Dog e potrebbero essere l'opzione migliore anche dopo che le vulnerabilità delle vittime e Decoy Dog stesso saranno state pienamente comprese.

Indicatori

Gli indicatori Decoy Dog relativi ai controller e ai campioni descritti in questo report sono elencati di seguito e disponibili nel nostro repository Github aperto.[20]

Gruppo di domini	Caratteristiche
ads-tm-glb[.]click	Dominio C2 Decoy Dog
allowlisted[.]net	Dominio C2 Decoy Dog
atlas-upd[.]com	Dominio C2 Decoy Dog
cbox4[.]ignorelist[.]com	Dominio C2 Decoy Dog
claudfront[.]net	Dominio C2 Decoy Dog
hsdps[.]cc	Dominio C2 Decoy Dog
j2update[.]cc	Dominio C2 Decoy Dog
maxpatrol[.]net	Dominio C2 Decoy Dog
nsdps[.]cc	Dominio C2 Decoy Dog
rcmsf100[.]net	Dominio C2 Decoy Dog
13[.]248[.]169[.]48	IP name server C2 Decoy Dog
156[.]154[.]132[.]200	IP name server C2 Decoy Dog
194[.]31[.]55[.]85	IP name server C2 Decoy Dog
5[.]199[.]173[.]4	IP name server C2 Decoy Dog
5[.]252[.]176[.]63	IP name server C2 Decoy Dog
5[.]252[.]176[.]22	IP name server C2 Decoy Dog
5[.]252[.]179[.]18	IP name server C2 Decoy Dog
67[.]220[.]81[.]190	IP name server C2 Decoy Dog
69[.]65[.]50[.]194	IP name server C2 Decoy Dog
69[.]65[.]50[.]223	IP name server C2 Decoy Dog
70[.]39[.]97[.]253	IP name server C2 Decoy Dog
83[.]166[.]240[.]52	IP name server C2 Decoy Dog
4996180b2fa1045aab5d36f46983e91dadeebf d4f765d69fa50eba4edf310acf	SHA256 binario Decoy Dog

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

ab8e333ef9bc5c5a7d1ed4cab08335861e150 b0639d3d0ca4c30b7def5cdccde	SHA256 binario Decoy Dog
ad186df91282cf78394ef3bd60f04d859bcaccc bcdcbfb620cc73f19ec0cec64	SHA256 binario Decoy Dog
6c8f413111f1abfee788dad4ee7cca37e0c259 7cca66d155af958c535faf55cc	SHA256 binario Decoy Dog
0375f4b3fe011b35e6575133539441009d015 ebecbee78b578c3ed04e0f22568	SHA256 binario Decoy Dog
6c8f413111f1abfee788dad4ee7cca37e0c259 7cca66d155af958c535faf55cc	SHA256 binario Decoy Dog
t1fde0f101c9395f39ecd16430b41041a59107 c73c904087309fb8d0e8d87e0077129f3f	Firma Telfhash Decoy Dog[21]

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

APPENDICE A: ELABORAZIONE DEI COMANDI DEL CLIENT

La Figura 18 illustra il ciclo operativo del client descritto nel documento. Il client passa ripetutamente dalla sospensione al polling del server e alla risposta ai comandi.

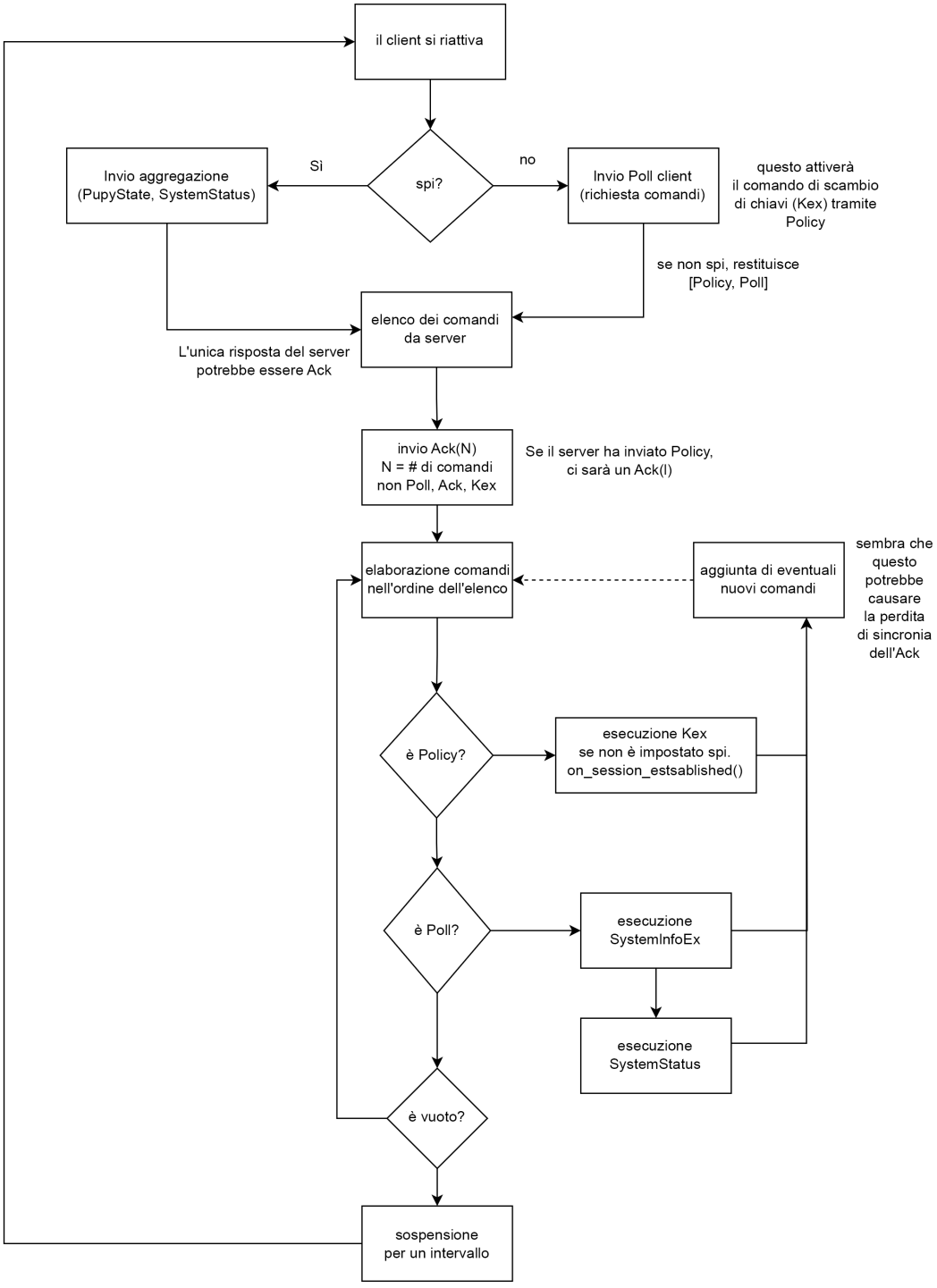


Figura 18. Flusso di lavoro del client.

APPENDICE B: STRUTTURA DEL PAYLOAD DI COMUNICAZIONE

La struttura del payload crittografato per client e server è identica, ma ci sono differenze nella loro elaborazione. In particolare, il client include 13 byte di informazioni sul client in ogni query, insieme al payload dei dati, come descritto in precedenza.

Sia il client che il server utilizzano il termine comando per indicare il tipo di informazioni che stanno trasmettendo al destinatario. Pertanto, quando il client contatta il server al momento della riattivazione, viene considerato un comando client. I comandi sono registrati in modo che il client o il server possa applicare un'elaborazione specifica ai dati. Ci può essere più di un comando in una singola comunicazione, anche se dal client questo è raro.

Il payload inviato per la codifica e la trasmissione ha il seguente formato:

- un checksum di 4 byte,
- pacchetti di comandi concatenati, contenenti un'identificazione del comando a 1 byte e una parte dati dipendente dal comando variabile.

La lunghezza totale del payload non può superare i 52 byte.

APPENDICE C: RICOSTRUZIONE CLIENT DA DATI PASSIVI

Come descritto in precedenza, le query di Pupy includono dati crittografati e due valori codificati, il nonce e l'SPI, che forniscono una certa sicurezza e consentono al server di ordinare le comunicazioni con i client. Il valore SPI viene utilizzato specificamente per identificare una sessione in corso all'interno del server ed è presente nelle query successive a uno scambio di chiavi riuscito. Di conseguenza, è quasi garantito che le query che contengono lo stesso SPI e che vengono eseguite in tempi ravvicinati provengano dallo stesso client. D'altra parte, un singolo client avrà molte sessioni e molti valori SPI nel tempo, quindi l'SPI da solo non è in grado di distinguere i client. Utilizziamo invece i valori nonce per separare le comunicazioni client.

Quando il client viene inizializzato, genera in modo casuale un valore nonce a 32 bit che funge da punto di partenza. Con ogni pacchetto, questo nonce viene incrementato in base alla lunghezza dei dati trasmessi. Il server utilizza il nonce come controllo di sicurezza minore, assicurandosi che aumenti con ogni query ricevuta, ma il suo uso principale è quello di decrittografare e interpretare correttamente la comunicazione sottostante. Da una serie di query Pupy osservate, possiamo decodificare questi valori nonce e calcolare il nonce successivo della serie, come mostrato nella Figura 19 di seguito.

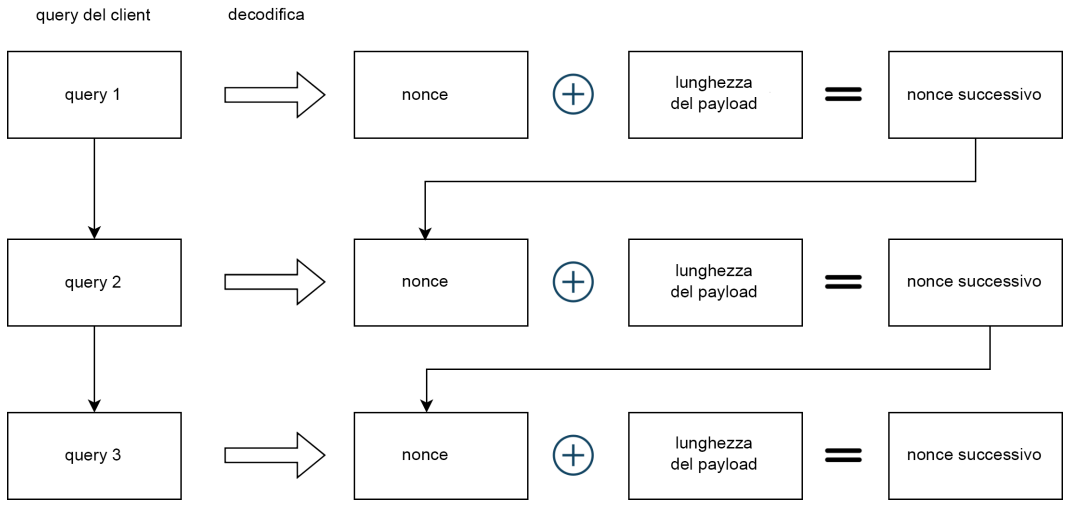


Figura 19. La relazione dei valori nonce all'interno di una serie di query Pupy.

Di conseguenza, possiamo sia ordinare le query da un singolo client, sia confermare che una serie di query appartengono ad un singolo client. Nella raccolta passiva di una distribuzione di Pupy, le query possono provenire da molti client e sovrapporsi nel tempo. Tuttavia, possiamo ancora separare queste osservazioni in attività client distinte con un alto grado di fiducia, grazie alla costruzione del nonce. Poiché il nonce viene utilizzato per crittografare il payload, lo sviluppatore ha utilizzato un generatore di numeri sicuri casuali per crearlo. Questo garantisce che ogni client generi valori nonce iniziali unici.[22] Il nonce viene ricreato ogni volta che il client viene riavviato.

La sicurezza aggiuntiva per la crittografia fornisce anche un meccanismo per distinguere i client nelle osservazioni aggregate. Per fare ciò, calcoliamo sia il valore nonce codificato che il valore nonce successivo per ogni query. Poi concateniamo le query tra loro utilizzando i valori nonce sequenziali, come mostrato nella Figura 20 qui sotto. Sebbene i dati sottostanti rimangano crittografati, possiamo stimare il numero di client e fare osservazioni sulla durata della loro attività. Inoltre, possiamo dedurre informazioni sulla comunicazione stessa utilizzando le lunghezze del payload e confrontando le serie temporali tra i client.

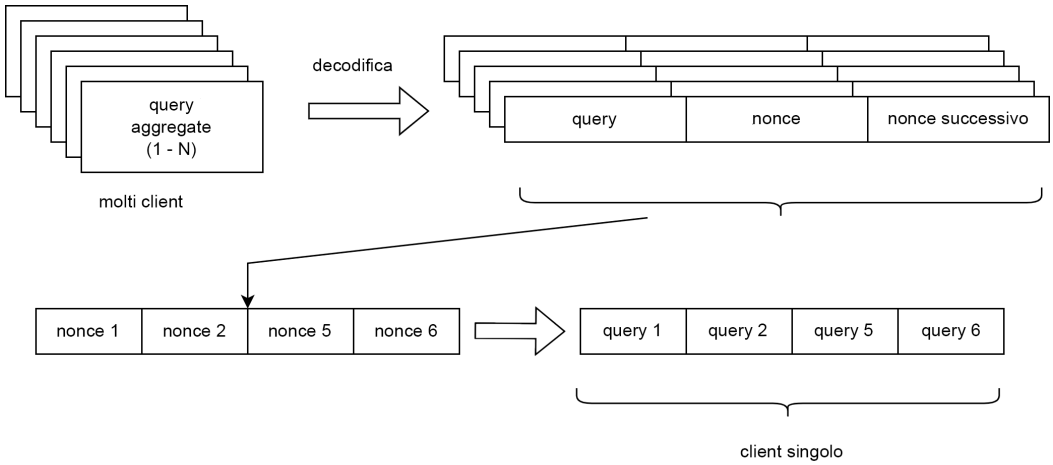


Figura 20. Separazione di un thread di query del client da un set aggregato di osservazioni utilizzando i valori nonce.

Ci sono due sfide con questo tipo di utilizzo: le modifiche nel resolver DNS del client infetto e le perdite di pacchetti. Per impostazione predefinita, Pupy utilizza il resolver DNS predefinito del client e la scelta del resolver potrebbe non essere sotto il controllo dell'attore. Se il client esegue il roaming, può utilizzare resolver ricorsivi diversi a seconda dell'ambiente locale. Nelle reti aziendali, possono utilizzare l'infrastruttura DNS di fornitori come Infoblox, in cui le query DNS verranno forzate sui resolver ricorsivi aziendali indipendentemente dalle impostazioni del client.[23] Inoltre, quando il DNS viene trasportato tramite UDP, la perdita di pacchetti è inevitabile. Il risultato è che è improbabile che si osservi ogni query solo nel DNS passivo, creando così lacune nella catena di nonce recuperati che potrebbero essere di dimensioni significative.

Possiamo comunque ricostruire i thread dei client, sfruttando il fatto che il nonce è un valore generato in modo casuale. Lo sviluppatore ha utilizzato un generatore di numeri sicuri che

22 Ci sono rare probabilità che lo stesso nonce possa essere generato contemporaneamente da due client diversi.

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

garantisce che i client Pupy indipendenti abbiano un'estrema probabilità di condividere un valore nonce. Inoltre, poiché è possibile trasmettere solo 52 byte di dati alla volta e il valore nonce aumenta in base al payload, è improbabile che due catene nonce generate in modo indipendente si sovrappongano. Di conseguenza, i client possono essere separati ordinando valori nonce e raggruppando quelli statisticamente simili. Un singolo client ha un solo nonce alla volta, permettendoci di stimare il numero di client attivi in un dato momento. Come mostriamo nel corpo principale del documento, abbiamo riscontrato che questa tecnica è molto efficace per recuperare le catene di query del client Decoy Dog.

APPENDICE D: FIRME DEL PAYLOAD

Le tabelle di questa sezione includono le lunghezze del payload per comandi specifici che vengono comunemente osservati nelle comunicazioni Pupy. In particolare, fornisce la lunghezza del payload crittografato per ogni comando client standard. I payload dei server sono più flessibili di quelli dei client; i più comuni sono mostrati di seguito.

Comando del client	Lunghezza del payload
Client check-in (initial)	18
Ack	19
Client check-in (rare variant)	22
System status	24
Online status	27
Client check-in (in session)	27
Port quiz	35
System information extended	39
Key exchange	47, 48

Tabella 2. Comandi client e lunghezze del payload.

Comando del server	Lunghezza del payload
Ack	6
Need session: policy, poll	42
Session incomplete: ack, policy	34
Error: message, policy, poll	44
Need system info: poll	15
Key exchange	62, 63
Exit	7

Tabella 3. Comandi del server comuni e lunghezze del payload.

APPENDICE E: GESTIONE DEGLI ERRORI

Pupy contiene una gestione personalizzata per una serie di errori che il server può incontrare. Un dominio che non viene decodificato correttamente o viene riprodotto genererà una risposta NXDOMAIN dal server. Il frammento di codice seguente mostra l'elaborazione della query del server. Se non viene restituita alcuna risposta, verrà restituita una risposta 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. Codice sorgente del server Pupy che elabora le query del client.

In Decoy Dog, molte query del client che dovrebbero generare un NXDOMAIN dal server restituiscono invece una risposta, in genere 15 indirizzi IP. Ciò sembra essere dovuto a un cambiamento nel codice, in cui Decoy Dog risponde a una grande varietà di possibili errori con un DnsCommandServerException internamente. DnsCommandServerException genererà una risposta al client, specificando il tipo di errore riscontrato e istruendo il client a eseguire un nuovo scambio di chiavi seguito dalla trasmissione delle informazioni di sistema. Il blocco di codice per questa gestione degli errori è mostrato di seguito.

```
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. Codice sorgente del server Pupy che restituisce un errore al client.

Nelle normali comunicazioni tra un server Pupy e un client, questo tipo di eccezione verrà sollevata quando non c'è una sessione attiva per un client noto. Viene utilizzato anche quando il payload del client non è valido o presenta un checksum errato. In tutti gli altri casi, il risultato è un NXDOMAIN.

APPENDICE F: ANALISI DEI CAMPIONI BINARI

File binari del client Pupy

Quando il server Pupy viene configurato per la prima volta, compila i file della libreria Pupy e crea un file modello statico per ogni architettura. Questi file modello sono compressi, fortemente offuscati e privati di tutti i simboli.

I file binari del client possono quindi essere creati manualmente utilizzando pupygen.py sul server. Lo script crea file binari specifici per C2 inserendo byte di configurazione specifici (host remoto, tipo di trasporto, flag di debug, ecc.) nel modello statico corrispondente all'architettura e al tipo di file di destinazione.

I file binari del client Pupy offrono una varietà di funzionalità avanzate e sono in grado di indirizzarsi praticamente a tutte le piattaforme, tra cui Windows, macOS, Linux, Solaris e Android. In particolare, sono in grado di rimanere residenti in memoria, interagire con il server, offrire funzionalità di reverse shell complete, creare copie senza file, ecc. Quando il file binario viene eseguito, crea copie di se stesso in memoria, nel tentativo di evitare il rilevamento e di rendersi più resistente alle tecniche di eliminazione dei processi.

Esempio di injection Java

I file binari di Decoy Dog includono una serie di nuove funzioni relative all'injection di Java. Questo è un esempio di una di queste funzioni.

```
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. Funzione Decoy Dog parzialmente disassemblata, in cerca del thread JVM corrente per l'injection.

APPENDICE G: REGOLA YARA PER DECOY DOG

La seguente regola YARA può essere utilizzata per rilevare i campioni di Decoy Dog che abbiamo osservato a luglio 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. Regola YARA per il rilevamento dei campioni di Decoy Dog.

APPENDICE H: VULNERABILITÀ DI SICUREZZA ESPOSTE

Quando un dispositivo è configurato per eseguire una query DNS su una connessione in entrata, consente a un'entità esterna di controllare parzialmente il proprio comportamento e le proprie risorse.[24] In particolare, questa configurazione può fornire agli attori di minacce un mezzo per la reconnaissance, la risoluzione aperta e la potenziale partecipazione ad un attacco denial of service. Poiché il DNS è complesso, sia i venditori che gli operatori di rete potrebbero non comprendere questi rischi. Sebbene le appliance di sicurezza che hanno trasmesso le richieste che abbiamo rilevato fossero destinate ad avere nuove funzionalità, l'uso del DNS in tali funzionalità espone la rete alla reconnaissance e potenzialmente ad altre minacce.

24 https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000PLRaCAO, ultimo accesso 11-06-2023

Un dispositivo all'interno di una rete che fornisce query DNS a qualsiasi entità esterna è noto come resolver aperto. In alcuni casi, un dispositivo può restituire risposte ma non risolvere completamente le richieste DNS esterne a causa di un'ampia gamma di circostanze. In entrambi i casi, tali dispositivi rappresentano un rischio per la rete stessa e per l'utilizzo della rete per amplificare gli attacchi DDOS (Distributed Denial of Service). I rischi dei resolver DNS aperti sono stati ben documentati e i resolver aperti sono vietati da molti contratti di servizio, compresi quelli di Infoblox, a causa di questi rischi.

Nel caso delle query Decoy Dog, le appliance di sicurezza non erano resolver aperti, ma consentivano comunque a una parte esterna di attivare le query DNS. Questo tipo di configurazione non può essere utilizzata per un attacco di amplificazione, ma può essere utilizzata da un attore di minacce per altri scopi. Ad esempio, un attore di minacce può eseguire una reconnaissance contro una rete; come mostrato nella Figura sotto. L'attore crea un dominio e configura il name server corrispondente per registrare le query in arrivo. L'attore utilizza poi un meccanismo di scansione per inviare nomi di dominio su misura per connettersi alla rete. Nel caso di una ricerca con resolver aperto, potrebbe trattarsi di query DNS. Nel caso di Decoy Dog, si trattava di connessioni HTTPS. In entrambi i casi, il dispositivo interno genera una query DNS che viene inviata al name server controllato dall'attore. L'attore è quindi in grado di collegare il nome del dominio e l'indirizzo IP originale alla query ricevuta. Sebbene questi tipi di attacchi ottengano una quantità limitata di informazioni in ogni tentativo, sono meccanismi ben consolidati per mappare le reti interne per un attacco successivo.

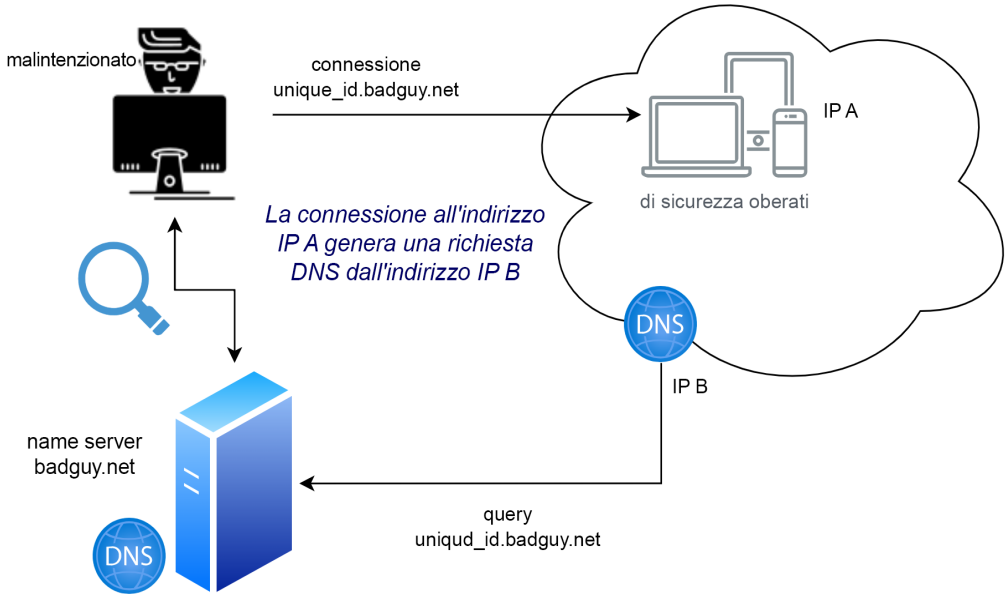


Figura 25. Un attore esegue la reconnaissance su una rete creando nomi di dominio univoci che creano query DNS sul proprio name server.

APPENDICE I: DATI DI RICERCA

Per la nostra ricerca, abbiamo creato un server Pupy e indirizzato le comunicazioni tra il server e i client tramite i nostri resolver ricorsivi. Abbiamo raccolto questi log di query DNS per la nostra analisi e li stiamo rendendo disponibili per la ricerca. I dati coprono diversi giorni di attività variabile. La maggior parte delle volte, controllavamo i client stabilendo un proxy inverso e i comandi venivano inviati tramite SSL. Sospettiamo che questo sia il caso anche di Decoy Dog. Tuttavia, abbiamo esercitato tutti i comandi disponibili tramite le risposte DNS del server. Inoltre, esistono periodi di tempo con più client attivi contemporaneamente e numerosi riavvii dei client. L'ambito di attività incluso dovrebbe consentire di ricreare i risultati qui descritti.

I dati sono disponibili nel nostro repository pubblico GitHub infobloxopen: threat-intelligence.[25] I registri delle query-risposte contengono i risultati dei record A e sono confezionati in un file csv che contiene i seguenti campi:

- timestamp: l'ora della query in secondi Unix
- query: il nome di dominio completo trasmesso nella query del client
- response: l'insieme di indirizzi IP restituiti dal server
- client_payload_len: il numero di byte di payload all'interno della query, comprese le informazioni sull'host
- server_payload_len: il numero di byte di payload all'interno della risposta

Il repository comprende anche gli indicatori di questo documento; ulteriori indicatori sono a disposizione dei difensori su richiesta come informazioni TLP:RED. Inoltre, stiamo fornendo dati risultanti da campioni binari di reverse engineering disponibili su VirusTotal. Questo include:

- Parametri di configurazione incorporati per ogni campione
- Chiavi crittografiche e password integrate per ogni campione
  - BIND_PAYLOADS_PASSWORD
  - DCONFIG_PUBLIC_KEY (solo per 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 regola YARA e un hash TELF in grado di rilevare i file binari di Decoy Dog

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

Infoblox unisce networking e sicurezza per offrire prestazioni e protezione senza pari. Scelti dalle aziende Fortune 100 e dagli innovatori emergenti, forniamo visibilità e controllo in tempo reale su chi e cosa si connette alla tua rete, in modo che la tua organizzazione funzioni più velocemente e blocchi le minacce prima che avvengano.

Sede centrale
2390 Mission College Blvd, Ste. 501
Santa Clara, CA 95054

+1.408.986.4000
www.infoblox.com

RP-072023 v1