infoblox

O DECOY DOG NÃO É UM PUPY COMUM:

Como separar um malware do DNS astuto da matilha

ÍNDICE

Resumo executivo

O Decoy Dog é um kit de ferramentas de malware descoberto pela Infoblox que usa o sistema de nomes de domínio (DNS) para executar comando e controle (C2). Um cliente comprometido se comunica com um controlador e recebe orientação dele por meio de consultas ao DNS. Esse controlador é integrado a um servidor de nomes DNS para o qual as consultas são transmitidas por meio do processo normal de resolução. Divulgamos a existência do Decoy Dog em abril de 2023 e lançamos um relatório detalhado de nossas descobertas iniciais em 23 de abril. A descoberta foi baseada no monitoramento de dados do DNS. Na época, a análise confirmou que o kit de ferramentas foi criado com base em um trojan de acesso remoto (RAT) conhecido como Pupy, mas não se sabia quais sistemas estavam sendo explorados, como o kit de ferramentas foi implantado ou se o Pupy havia sido modificado.[1] Esperávamos que, com os detalhes que fornecemos, outras pessoas da comunidade localizassem as máquinas comprometidas e a história completa se tornasse conhecida. No entanto, o mistério em torno do Decoy Dog só aumentou.

Desde abril, a Infoblox conduziu pesquisas adicionais sobre o Decoy Dog e o Pupy. Este relatório é o resultado dessas pesquisas. Aprendemos que Decoy Dog é uma atualização importante do Pupy, que usa comandos e configurações que não estão no repositório público. Desenvolvemos algoritmos para separar as comunicações do cliente do Decoy Dog e inferir uma série de outras propriedades sobre cada controlador. Isso nos permite concluir com grande confiança que o conjunto de ferramentas se espalhou e está sob controle de pelo menos três atores. Embora a atividade que observamos permaneça confinada à Rússia e ao Leste Europeu, existem agrupamentos distintos de técnicas, táticas e procedimentos (TTP) dentro dos controladores, consistentes com múltiplos atores.

Todos os atores do Decoy Dog responderam às nossas divulgações de abril de alguma forma, e as variações corroboram nossa avaliação de vários operadores. Logo após o primeiro anúncio nas mídias sociais, alguns dos servidores de nomes foram retirados do ar. Todos os demais foram modificados para remover o comportamento que destacamos em nosso primeiro artigo, embora isso tenha sido feito de maneiras diferentes, dependendo do controlador. Um conjunto de controladores começou a restringir as respostas a consultas dependendo do país de origem, uma técnica chamada geofencing, enquanto outros alteraram sua resposta a consultas para o subdomínio de ping.

Um ator respondeu tão rapidamente à nossa divulgação no LinkedIn que inicialmente pensamos que os novos domínios eram registros copiados por pesquisadores de segurança. Uma análise mais aprofundada, no entanto, mostrou que eram domínios de substituição. Em vez de encerrar a operação, o agente transferiu os clientes comprometidos existentes para os novos controladores. Essa é uma resposta extraordinária, que demonstra que o ator sentiu a necessidade de manter o acesso às suas vítimas existentes. Isso criou uma separação clara entre os TTPs de um conjunto de domínios do Decoy Dog e todos os outros.

Nas semanas que se seguiram ao nosso anúncio, ficamos surpresos com o fato de ninguém ter se manifestado para identificar o malware e a vulnerabilidade subjacentes que deram ao Decoy Dog sua base de operação. Mas, à medida que nossa pesquisa avançava, ficou claro por que as comunicações passaram despercebidas por mais de um ano. Os ataques usando o Decoy Dog têm sido altamente direcionados, e cada controlador tem um pequeno número de clientes ativos. Alguns servidores mantiveram de forma consistente de quatro a oito clientes ativos durante meses. Embora outros tenham observado um aumento no número de clientes ativos simultaneamente ao longo do tempo, o número total de dispositivos afetados observados em qualquer momento foi inferior a 100. Um pequeno conjunto de vítimas geralmente exclui atores com motivação financeira, e a necessidade de persistir em um dispositivo por um longo período de tempo é consistente com atores altamente avançados.

---

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

Conseguimos reconstruir partes das comunicações do Decoy Dog identificando assinaturas do nosso próprio tráfego do Pupy. Estabelecemos um servidor Pupy na Internet, que, quando combinado com a engenharia reversa seletiva do código, nos permitiu correlacionar consultas e respostas de DNS a comandos específicos do Pupy. Com isso, conseguimos a) determinar que o Decoy Dog contém comandos não encontrados no Pupy e b) caracterizar a maior parte das comunicações. Além disso, os atores do Decoy Dog parecem aproveitar o Pupy para utilizar outras camadas de transporte fora do DNS para funções como troca de chaves. Os agentes de ameaças provavelmente consideram essa uma das vantagens do Pupy como um trojan de acesso remoto (RAT).

A primeira implantação conhecida do kit de ferramentas Decoy Dog ocorreu no final de março ou início de abril de 2022. Ele foi vendido ou roubado logo em seguida, conforme indicado pelo surgimento de um segundo controlador, com TTPs diferentes, que estava ativo em meados de maio. Um terceiro domínio foi registrado em julho de 2022 e estrategicamente envelhecido até setembro. É possível que esses dois últimos controladores pertençam ao mesmo ator, pois compartilham muitas características, inclusive a hospedagem no espaço de IPs russo. No entanto, eles têm algumas diferenças. Alguns meses depois, mais dois domínios foram registrados, novamente com características distintas daquelas dos controladores anteriores. O ator que registrou esses domínios migrou os clientes imediatamente após nossa divulgação para novos domínios. No total, a Infoblox está monitorando atualmente 21 domínios do Decoy Dog, alguns dos quais foram registrados e implantados no último mês.

Depois de determinar que o Decoy Dog era significativamente diferente do Pupy por meio de nossa análise dos registros de DNS, examinamos amostras binárias relacionadas disponíveis no VirusTotal para ver se as diferenças eram aparentes nos executáveis. A engenharia reversa dessas amostras mostrou que, embora tenham sido detectadas como Pupy, elas são muito mais avançadas do que a versão de código aberto. As amostras incluem a) a capacidade de executar código Java arbitrário no cliente, b) vários novos mecanismos de transporte e c) novos mecanismos de DNS para garantir a persistência. Um mecanismo é semelhante a um algoritmo tradicional de geração de domínios de DNS (DGA) e usa provedores de DNS dinâmicos gratuitos para se conectar aos chamados controladores de emergência. Todas as amostras compartilham as mesmas atualizações fundamentais, embora uma das amostras tenha recursos exclusivos não vistos nas outras, relacionados ao uso de transportes de streaming.

Por motivos que ainda não estão claros, o Decoy Dog viola os princípios fundamentais das comunicações secretas, que geralmente visam evitar a detecção e a recuperação do conteúdo por um adversário. Enquanto os servidores Pupy normais rejeitam consultas de comunicação repetidas de clientes comprometidos, os servidores Decoy Dog não apenas respondem a consultas de DNS repetidas, mas também a qualquer consulta bem elaborada. Esse comportamento é semelhante às configurações curinga no DNS e foi um fator significativo na detecção do Decoy Dog pela Infoblox. Dada a sofisticação do Decoy Dog, especulamos que a repetição e o comportamento curinga são causados por design; qualquer que seja a intenção, a repetição generalizada do DNS foi parcialmente responsável pela incapacidade do setor de ver o Decoy Dog como um novo malware.

A varredura agressiva da Internet por um fornecedor de segurança levou à retransmissão de milhões de comunicações do Decoy Dog por meio de redes globais, incluindo vários de nossos clientes. Isso, por sua vez, levou à nossa descoberta do kit de ferramentas. A incapacidade do fornecedor de identificar o tráfego como malware, a fim de evitar a repetição das consultas, acionou conexões de DNS de redes não infectadas para os controladores do Decoy Dog. Estamos confiantes de que nenhum cliente da Infoblox foi infectado e que as consultas aos nossos resolvedores foram todas resultado de uma varredura anômala do fornecedor. Apesar da falta de ameaça imediata às redes de nossos clientes, o Decoy Dog continua sendo um kit de ferramentas sofisticado com origens incertas e pode continuar a se espalhar.

Não só o Decoy Dog foi recentemente observado “in the wild”, mas até onde sabemos, é o primeiro uso do componente C2 do DNS do Pupy em uma operação maliciosa. Em parte, isso provavelmente se deve à dificuldade de estabelecer um servidor de nomes do Pupy, o que requer a modificação do software no repositório e a configuração adequada do DNS. A falta de exposição torna mais difícil para o setor de segurança detectar e se defender contra o Pupy e o Decoy Dog. Para ajudar a interromper as operações que usam esses sistemas C2, estamos fornecendo à comunidade um conjunto de dados de pesquisa contendo o tráfego de DNS do Pupy capturado de nosso próprio servidor e detalhes do funcionamento interno do software. Essa documentação é a primeira desse tipo e permitirá que outros criem algoritmos de detecção, bem como reproduzam nossas descobertas.

A história do Decoy Dog revela o poder do DNS como fonte de detecção e resposta a ameaças. Ela também revela uma fraqueza inerente ao ecossistema de inteligência centrado em malware que domina o setor de segurança. O kit de ferramentas foi descoberto por algoritmos de detecção de ameaças ao DNS, e a única defesa contra isso hoje é o DNS. Além disso, sinalizamos vários domínios controladores como suspeitos e os bloqueamos em nossos resolvedores antes de perceber que todos estavam usando um malware comum. Esse tipo de proteção, que impede a atividade mal-intencionada antes que seja identificada e, muitas vezes, antes de ser operacionalizada, é exclusivo dos sistemas de detecção e resposta do DNS.

Neste documento, fornecemos aos defensores o conhecimento para identificar o Pupy e o Decoy Dog. Embora descrevamos detalhadamente o C2 do DNS, não forneceremos informações que ajudem os malfeitores a implantar o Pupy, nem divulgaremos a assinatura completa do DNS do Decoy Dog. Explicamos alguns comportamentos que identificamos em nosso artigo original e destacamos como o Decoy Dog é diferente do Pupy. Além disso, descreveremos nossa análise de grandes volumes de tráfego de DNS do Decoy Dog, o que nos permitiu estimar o número de clientes e o tráfego de comandos sem possuir o próprio malware ou controlar o servidor de nomes. Descrevemos como as amostras do Decoy Dog diferem das do Pupy. Por fim, discutimos como os operadores do Decoy Dog reagiram às nossas revelações e demonstramos características comuns entre os subgrupos de controladores. Os apêndices contêm informações técnicas de apoio adicionais.

HISTÓRICO

A Infoblox descobriu o Decoy Dog, um kit de ferramentas de comando e controle (C2) que usa o sistema de nomes de domínio (DNS) no início de abril de 2023. Ele é baseado em um trojan de acesso remoto (RAT) de código aberto chamado Pupy[2] e transporta comunicação criptografada entre clientes e servidores, ou controladores, por meio de consultas de nomes de domínio e respostas de endereços IP. A descoberta surgiu a partir de algoritmos que monitoram consultas passivas de DNS aos resolvedores da Infoblox em busca de comportamentos anômalos. As consultas aos domínios do Decoy Dog foram feitas a partir de dispositivos de segurança em um pequeno número de redes de clientes. Essas consultas criaram uma assinatura consistente com o sinalizador de malware persistente e de baixo perfil. A análise humana da atividade foi alarmante porque, embora o DNS estivesse claramente sendo usado como um canal de comunicação confidencial, os domínios não foram identificados como C2 em nenhum dado de inteligência disponível publicamente. Na verdade, alguns foram rotulados de “respeitáveis” em verificadores de reputação on-line. Lançamos um conjunto de domínios em 13 de abril para ajudar a comunidade a bloquear o tráfego e identificar a natureza do comprometimento.

Durante nossa pesquisa original, a Infoblox identificou uma assinatura de DNS exclusiva que era independente do software do Pupy. Os atores implantaram e operaram seu sistema C2 de uma maneira muito específica; por esse motivo, identificamos o Decoy Dog como um

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

kit de ferramentas distinto. Apenas um pequeno número de domínios em todo o mundo compartilhava essa assinatura, todos eles servidores de nomes do Decoy Dog.

Em 23 de abril, publicamos parte da assinatura, a análise inicial do DNS passivo e um subconjunto dos domínios controladores em nosso relatório "Dog Hunt: Finding Decoy Dog Toolkit in Anomalous DNS Traffic."[3] Esse documento destacou um comportamento específico do Pupy, no qual ele retornava uma série de respostas do localhost às consultas de subdomínios específicos contendo "ping". Ele também descreveu uma série de tendências nas comunicações de DNS que não pudemos explicar completamente na época. Em particular, identificamos padrões surpreendentes nos endereços IP que foram retornados nas respostas e no fato de os servidores responderem às consultas repetidas, o que é inesperado para um sistema de comunicação secreto.

Após os anúncios, vários membros da comunidade de segurança, incluindo fornecedores e outras organizações, entraram em contato conosco. Muitos deles tinham visto tráfego relacionado em suas próprias redes ou nas redes de seus clientes, mas ninguém havia identificado dispositivos comprometidos ou reconhecido o escopo da atividade. Algumas dessas organizações forneceram informações que nos levaram a isolar e confirmar como o DNS foi gerado em nossas próprias redes. Outros ajudaram a confirmar a amplitude da atividade e a testar hipóteses. Essa colaboração informal foi muito útil, e somos gratos por ela.

Para simplificar, usamos o termo Pupy neste artigo para nos referirmos especificamente ao C2 do DNS do Pupy, não ao Pupy em geral.

Pupy

UMA ESPÉCIE RARA

O Pupy é um trojan de acesso remoto (RAT) de código aberto pós-exploração que apresenta um sistema de transporte modular complexo.[4] Embora a base de código principal do Pupy tenha sido disponibilizada no GitHub em 2015, o mecanismo de C2 do DNS só foi adicionado em 2019. Este documento é a primeira documentação pública do Pupy C2. Além disso, estamos fornecendo um conjunto de dados no GitHub para que outros possam reproduzir nosso trabalho e criar defesas para o futuro.

Embora o Pupy seja de código aberto, o uso do protocolo C2 do DNS é raro; não conseguimos identificar seu uso "in the wild" fora do Decoy Dog.[5] A partir de nossos próprios resolvedores, que atendem a empresas e organizações em todo o mundo, não encontramos nenhuma evidência histórica do uso do C2 do DNS do Pupy. No pDNS global para os primeiros seis meses de 2023, usando detectores de DNS que desenvolvemos para o Pupy, não encontramos nenhum uso do software fora do Decoy Dog. Por fim, consultamos privadamente uma ampla gama de fornecedores; nenhum deles viu o uso do software. Quando o uso do Pupy por atores de ameaças persistentes avançadas (APT) foi relatado, aparentemente os componentes C2 do DNS não foram empregados.[6]

O uso raro do Pupy provavelmente se deve, pelo menos em parte, à dificuldade de operar o sistema. Estabelecer comunicações do Pupy pelo DNS global não é fácil. É necessário

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

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

5 A expressão "in the wild" é usada no vernáculo da segurança cibernética para significar implantado operacionalmente e não parte de testes de penetração ou pesquisas isoladas.

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

configurar corretamente o servidor de nomes e modificar o código no repositório do GitHub. Além disso, há complexidades no DNS que variam entre os resolvedores recursivos que o software Pupy não trata corretamente. Esses desafios provavelmente prejudicaram sua adoção tanto por red teams quanto por hackers, em contraste com ferramentas populares como o Cobalt Strike, que vemos com bastante frequência.[7]

Embora o C2 do DNS do Pupy seja raro atualmente, o uso do Decoy Dog está se espalhando, e a probabilidade de os defensores enfrentarem o Pupy de alguma forma está aumentando. Para ajudar a preparar a comunidade, a Infoblox realizou pesquisas significativas sobre o Decoy Dog e o Pupy. A Infoblox implantou um servidor do Pupy na Internet para comparar seu comportamento com o do Decoy Dog. Em seguida, capturamos dados de pacotes (pcap) e registros de DNS passivo dos resolvedores da Infoblox. Usamos nossa implantação do Pupy em conjunto com a engenharia reversa seletiva do código para entender melhor a natureza exclusiva do Decoy Dog. Nesta seção, explicamos os componentes do Pupy que são relevantes para nossa pesquisa. Para simplificar, limitamos este documento às comunicações que usam respostas IPv4 (registro A), embora, quando disponível, o Pupy use respostas IPv6 (AAAA). A codificação de consulta descrita no documento é o padrão atual do Pupy, versão 2 (a menos que especificado de outra forma).[8]

COMO O PUPY FUNCIONA

Em nosso artigo anterior, apresentamos uma visão geral do Pupy e destacamos algumas características incomuns do Decoy Dog.9 Neste artigo, vamos nos aprofundar no protocolo de comunicação do Pupy para demonstrar suas conexões com o Decoy Dog e como explorar o DNS do Pupy coletado passivamente para entender uma operação em andamento. [9]

O Pupy foi projetado para fornecer comunicações contínuas entre os clientes infectados e o servidor para que, quando o ator quiser acessar remotamente o cliente, a conexão já esteja estabelecida. O ator é capaz de monitorar os clientes conectados e comandá-los seletivamente para que forneçam uma ampla gama de ações. O DNS é usado somente para comunicações C2. Todos os dados significativos exfiltrados do cliente são enviados por uma das muitas outras opções de transporte oferecidas pelo Pupy. Como resultado, o cliente de DNS do Pupy se limita a fazer o check-in com o controlador, reconhecer comandos, fornecer informações sobre o sistema e algumas outras tarefas. Entre o tratamento de comandos do servidor, o cliente fica adormecido.

As comunicações de DNS são iniciadas e mantidas pelo cliente. O cliente envia consultas por meio de seu caminho de resolução de DNS normal ou por meio de DNS sobre HTTPS (DoH), quando ele está habilitado e disponível.[10] O controlador envia comandos em resposta a solicitações do cliente na forma de endereços IP criptografados. Cada consulta-resposta é uma comunicação completa, o que significa que nem o cliente nem o servidor podem dividir dados para um único comando em duas consultas de DNS. Esse protocolo é diferenciado dos sistemas comuns de encapsulamento de DNS, como por exemplo, o Iodine,[11] onde o cliente estabelece uma sessão sobre DNS que pode incluir a reconstrução de vários pacotes em cada extremidade para processar a comunicação. O cliente é obrigado a

---

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

8 Uma versão anterior do C2 do Pupy não incluía informações de host em cada consulta. Agora sabemos que o Decoy Dog é a versão 3 do cliente, mas a codificação da consulta parece ser a mesma da versão 2.

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

10 O Pupy usa servidores Quad9 por padrão para DoH.

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

reconhecer a maioria dos comandos, e o servidor responde a cada consulta de cliente válida com comandos ou confirmação. O vocabulário do cliente é extremamente limitado. Ele tem nove tipos de consultas pelas quais gerencia sessões, reconhece comandos, envia informações do sistema e estabelece chaves. Comandos personalizados podem ser adicionados escrevendo funções adicionais, mas exigem uma compreensão completa do software.

Ao acordar, o cliente consulta o servidor de uma de duas maneiras diferentes, dependendo de ter sido estabelecida ou não uma chave compartilhada. Essa consulta fornece ao servidor informações atuais sobre o sistema e o estado do cliente do Pupy ou faz uma consulta simples que serve para iniciar uma nova sessão criptografada. Embora seja possível desativar as sessões criptografadas, esse não é o padrão e não foi observado no Decoy Dog. Em resposta, o controlador confirma a solicitação, exige que o cliente realize uma troca de chaves ou envia novos comandos. Quando o conjunto completo de comandos estiver concluído, o cliente ficará adormecido durante o intervalo estabelecido, por padrão 60 segundos. Esse processo é repetido enquanto o cliente estiver em execução. Uma visão geral de alto nível das comunicações cliente-servidor do Pupy é mostrada na Figura 1, e uma visão mais detalhada do processo do cliente pode ser encontrada no Apêndice A.

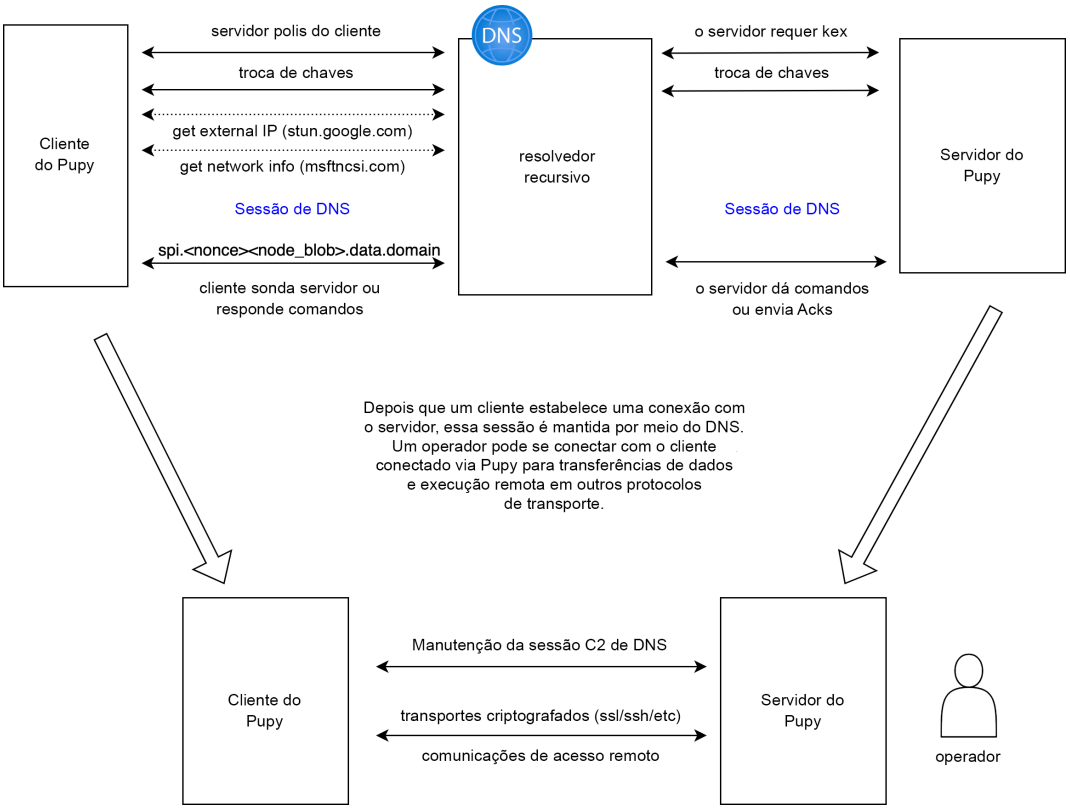

Figura 1. Uma visão geral de alto nível das comunicações do Pupy.

O ator do Pupy interage com os clientes a partir do utilitário de linha de comando do controlador. Quando o cliente entra em contato com o controlador, todos os comandos em fila são codificados na resposta do DNS. O operador estabelece uma conexão em uma porta aberta do cliente e especifica a camada de transporte a ser usada para a exfiltração. As comunicações do DNS do servidor ainda são bastante restritas, embora mais abrangentes do que as do cliente. Há uma grande variedade de comandos, e eles podem ser concatenados em uma única resposta ao cliente. Considerando que o cliente inicia a troca de comunicações, o servidor é responsável por garantir a segurança das comunicações. Ele faz isso impondo as chamadas sessões com cada cliente, que servem para o rodízio de chaves de criptografia. Isso é descrito na próxima seção.

INÍCIO DA SESSÃO

O Pupy exige o estabelecimento de uma sessão criptografada entre o cliente e o controlador antes de transmitir os comandos do ator. Essa sessão expira quando as comunicações do cliente atingem o tempo limite e pode ser forçada a ser renovada por outros motivos, incluindo erros na decodificação da consulta de DNS ou uma reinicialização do cliente. As sessões são identificadas pela presença de um rótulo de índice de parâmetros de segurança (SPI) na consulta e são criptografadas usando uma chave compartilhada efêmera. Como os detalhes da comunicação dependem de vários fatores, inclusive se o cliente já se conectou anteriormente ao servidor, o protocolo exato para a inicialização da sessão pode diferir, criando uma variação nas trocas de DNS observadas. No entanto, a troca típica é a seguinte:

- O cliente faz check-in no servidor sem uma sessão estabelecida ou com uma sessão expirada (consulta 1).
- O servidor responde com um comando que exige uma troca de chaves e informações do sistema do cliente.[12]
- O cliente reconhece a necessidade de informações do sistema (consulta 2).
- O cliente gera um par aleatório de chaves públicas-privadas usando um algoritmo de curva elíptica e o envia ao servidor; o servidor faz o mesmo e responde com sua nova chave (consulta 3).
- O cliente e o servidor usam essa troca para estabelecer uma nova chave de sessão compartilhada, que é usada para criptografar pacotes com criptografia AES, e também criam o SPI para identificar a sessão.
- O cliente coleta informações sobre sua rede, incluindo seu endereço IP externo, usando consultas adicionais de DNS a outros serviços.
- O cliente transmite essas informações usando a chave de criptografia compartilhada e sinalizando a presença de uma sessão ativa com a inclusão do SPI na consulta (consulta 4).
- O cliente envia informações adicionais sobre o status do sistema (consulta 5).

A chave compartilhada e o SPI são normalmente estabelecidos após três consultas, embora a troca de chaves seja tecnicamente uma única consulta e resposta. Durante uma sessão, cada consulta e resposta será criptografada usando essa chave compartilhada. A criptografia também usa um nonce de 32 bits gerado pelo cliente, que muda a cada consulta. Quando uma nova sessão é estabelecida, as chaves são regeneradas, mas o valor do nonce do cliente continua enquanto o cliente estiver operando. Isso é discutido mais detalhadamente na seção abaixo.

CODIFICAÇÃO DE CONSULTAS

O cliente gera consultas que contêm comunicações criptografadas com o servidor. Isso pode incluir informações de trocas de chaves ou uma resposta a comandos do servidor. Há um máximo de 52 bytes de dados transmitidos que podem ser comunicados em cada consulta. Além dos dados transmitidos, cada consulta inclui:

- nonce, um valor incremental de 4 bytes gerado pelo cliente
- versão, um valor de 1 byte que indica a versão do C2 do DNS do Pupy
- cid, um valor de 4 bytes da configuração do cliente, que é gerado aleatoriamente ao criar o cliente

12 Normalmente, isso ocorre na forma de dois comandos denominados Policy e Poll no lado do servidor.

- iid, um valor de 2 bytes que contém os 16 bits inferiores do processo do cliente do Pupy
- id do nó, um valor de 6 bytes do cliente, normalmente o endereço MAC do dispositivo
- opcionalmente, SPI, um valor de 4 bytes gerado durante a troca de chaves e presente em consultas que representam uma sessão no servidor para um determinado cliente.

Cada consulta do cliente inclui esses 13 bytes de informações do cliente, bem como uma soma de verificação de 4 bytes sobre a carga útil subjacente. A carga útil subjacente é criptografada e consiste em uma série de comandos e dados relacionados.

O cliente criptografa e codifica os dados a serem transmitidos ao servidor como um nome de domínio totalmente qualificado (FQDN), chamado de nome de consulta (qname) no protocolo DNS. Todo o processo, mostrado na Figura 2 abaixo, inclui a criptografia, a organização e a codificação dos dados transmitidos e das informações adicionais necessárias para o servidor. Funciona da seguinte maneira:

- Os dados a serem transmitidos são anexados às informações específicas do host.
- Essa string de bytes composta é criptografada usando uma chave simétrica compartilhada e o nonce atual.
- Os primeiros bytes criptografados dos dados transmitidos, até 35 bytes, são codificados e usados para o primeiro rótulo, ou o mais à direita, do qname.
- O restante dos bytes criptografados, que podem conter até 17 bytes dos dados transmitidos, é acrescido do valor nonce atual e codificado para criar o segundo rótulo do qname.
- Se o índice de parâmetro de segurança (SPI) existir no cliente, ele será codificado e usado no terceiro rótulo, ou o mais à esquerda, do qname; esse valor é definido após uma troca de chaves com o servidor.
- O nonce é incrementado pelo comprimento dos dados criptografados dentro do cliente a ser usado na próxima consulta.

A codificação de bytes criptografados para um rótulo de nome de domínio foi descrita em nosso artigo anterior. Ela usa um mapa personalizado em combinação com a codificação de 32 bits para garantir que o resultado final seja um nome de domínio válido. A estrutura de carga útil de dados subjacente é descrita no Apêndice B.

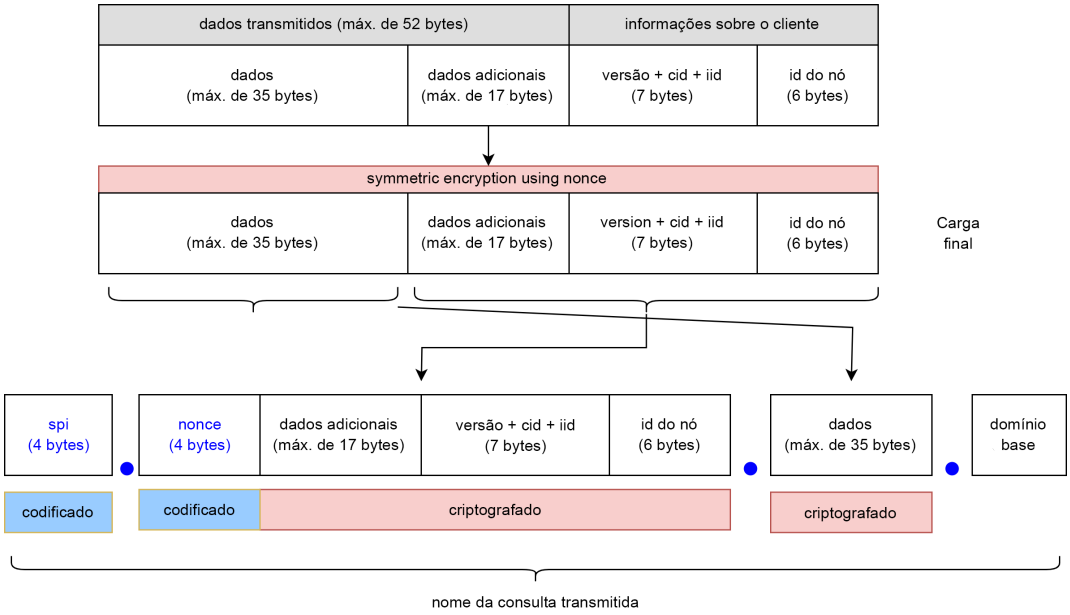

Figura 2. O processo de conversão de dados do cliente em um qname para uma consulta de DNS. O domínio base é o nome de domínio do servidor Pupy.

O Pupy usa o AES por padrão ao criptografar consultas de DNS. Se uma chave compartilhada foi estabelecida com o cliente, ele a usa para criptografar a string de bytes completa simetricamente; caso contrário, usa a chave pública estabelecida. Em ambos os casos, o nonce atual também é usado na criptografia para garantir que a consulta codificada seja exclusiva, mesmo que os dados subjacentes transmitidos permaneçam os mesmos em várias consultas. Esse é um mecanismo padrão de proteção contra ataques criptográficos. Como resultado, o nome de domínio consultado pode ser decodificado para revelar os dados criptografados, mas os dados criptografados não podem ser decodificados sem a chave. O valor do nonce é inicializado com um valor aleatório de 32 bits e é incrementado pelo comprimento da carga útil em cada consulta.

Quando o servidor de nomes do Pupy recebe uma consulta, ele decodifica o nome do domínio para revelar o valor SPI, o nonce e a carga criptografada. Para garantir que está recebendo comunicações válidas do cliente, o servidor verifica se o SPI é válido quando presente e se o nonce é maior do que o anterior registrado para o cliente. Ele faz várias outras verificações nos dados, incluindo uma verificação do número da versão, que é criptografada na carga útil. Se alguma dessas verificações falhar, ela retornará um erro ao cliente.

Em particular, o Pupy não responde à mesma consulta duas vezes, e qualquer servidor do Pupy não modificado responderá a uma consulta que já tenha recebido no passado com uma resposta NXDOMAIN (no such domínio). Validamos esse comportamento com nosso próprio servidor do Pupy, tentando consultar um nome de domínio consultado anteriormente. Isso é importante porque uma característica do Decoy Dog é que ele responde a consultas de DNS reproduzidas com respostas consistentes com o protocolo C2 do Pupy.

Como a consulta de DNS contém uma codificação reversível do nonce, e o nonce é incrementado pelo comprimento da carga útil em cada consulta, podemos reconstruir threads de consultas associadas a um único cliente. Como veremos mais adiante neste documento, dada a coleta de DNS passivo para um domínio do Pupy ou Decoy Dog, podemos usar essa reconstrução para estimar o número de clientes, bem como a natureza da comunicação em determinados casos.

TRATAMENTO DE NOMES DE DOMÍNIOS ESPECIAIS

Ao receber uma consulta, o servidor disseca o nome da consulta e determina se ele corresponde à estrutura apropriada para um pacote criptografado de um cliente. Há alguns casos especiais que têm processamento exclusivo. Com exceção desses casos especiais, ele rejeitará qualquer solicitação que não atenda ao formato esperado. Um desses casos especiais são as solicitações de ping, que descrevemos em nosso artigo anterior. Uma consulta para um subdomínio pingN, onde N é um número inteiro, retorna uma sequência de respostas do localhost com comprimento N. Uma consulta para o próprio ping retorna 15 respostas desse tipo, e uma consulta para o domínio base retorna uma única resposta localhost, ou seja, 127.0.0.1.

Além das solicitações de ping, o servidor pode ser configurado para responder a consultas de rótulo único com um único endereço IP. A finalidade dessa funcionalidade é desconhecida e não parece ser usada no cliente; ela é mencionada no código-fonte como uma solicitação de ativação de DNS. Esse recurso não está documentado e, para utilizá-lo, um ator precisaria entender como o software do servidor funciona.

O tratamento especial para subdomínios de rótulo único é realizado por meio da configuração de entradas de "ativação", que são pares de strings de caracteres de valores-chave. O valor é então usado em conjunto com a chave privada do servidor para criar um endereço IP de resposta. Essa resposta é criada usando uma função de hash unidirecional e não pode ser invertida. O hash diferencia maiúsculas de minúsculas e é definido como

MD5(subdomain_label + activation_value + private_key)

CODIFICAÇÃO DE RESPOSTA

Quando o servidor recebe uma consulta de um cliente, ele decodifica, descriptografa, verifica os resultados e processa os dados do cliente. Em particular, uma comunicação com o cliente devidamente formatada deve conter dois ou três rótulos, conforme descrito anteriormente na seção sobre codificação de consulta. O servidor então reunirá uma resposta para o cliente contendo um ou mais comandos. Embora ele possa retornar consultas IPv4 (A) ou IPv6 (AAAA), limitaremos nossa descrição às consultas IPv4 (A) para simplificar.

A resposta do servidor é uma string binária criptografada que é, então, codificada em um ou mais registros A.[13] O processo dessa codificação é mostrado na Figura 3 abaixo. O número máximo de bytes na resposta é 64, que são codificados em segmentos de 3 bytes, resultando em um máximo de 22 endereços IPv4 na resposta.

- Na primeira etapa, o servidor calcula o comprimento da resposta e o anexa aos dados da resposta. Em seguida, ele acrescenta bytes aleatórios para criar uma string composta que é um múltiplo de 3 bytes de comprimento.[14] Chamamos essa string composta de carga útil.
- Na segunda etapa, os endereços IPv4 são criados iterativamente a partir de segmentos de 3 bytes da carga útil. Cada endereço IPv4 é representado por um valor de 32 bits, onde o bit 0 é o bit mais alto.
- Os 3 primeiros bits de cada endereço são aleatórios.

13 Uma série de comandos é montada e, em seguida, criptografada usando uma chave compartilhada e o nonce atual antes da codificação, se uma troca de chaves tiver sido concluída. Caso contrário, a chave privada do servidor é usada, juntamente com o nonce, para criptografar os dados com um algoritmo de curva elíptica de chave pública.

14 No código, esse processo é mais complicado, mas tem o mesmo resultado.

- Cada segmento tem um índice, que permite que os dados sejam ordenados pelo cliente no recebimento; isso é representado por 5 bits. Esse índice está nos bits 3-7 do resultado.
- O segmento de carga útil está nos bits 8-30, o que força o bit mais alto do segmento de carga útil a ser o bit inferior do primeiro octeto no endereço IPv4.
- Por fim, o bit menos significativo, o bit 31, é um bit de verificação gerado no segmento de carga útil. Devido à natureza dessa soma de verificação, esse bit é de 1 em 75% dos endereços IPv4.
- A string de 32 bits resultante é interpretada como um endereço IPv4 e anexada à resposta.

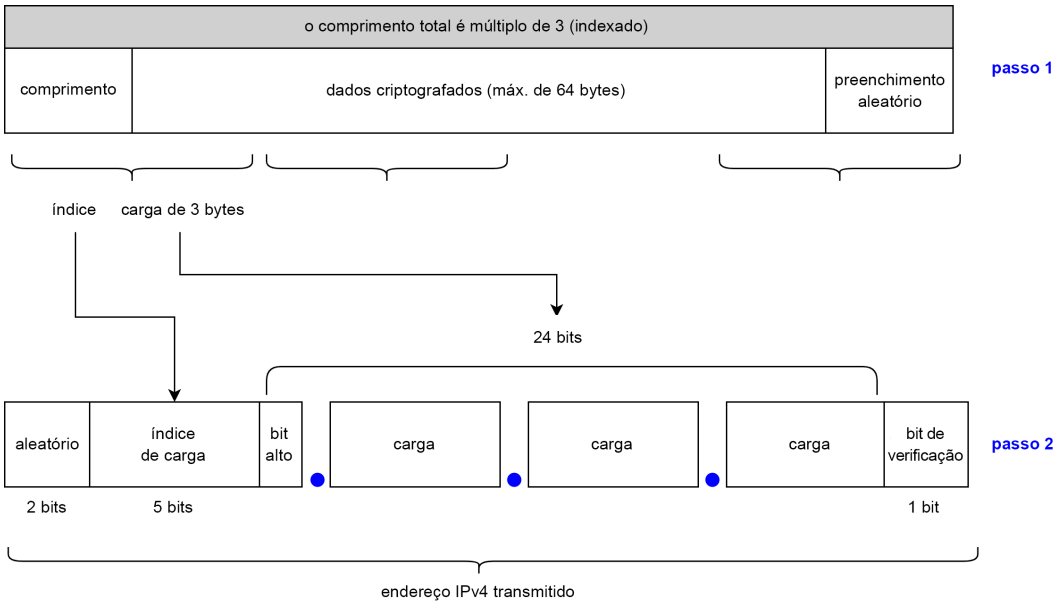

Figura 3. Codificação do servidor do Pupy de uma resposta de IPv4 a uma consulta do cliente. Os dados são codificados em uma série de endereços IPv4 usando 3 bytes da carga útil em cada endereço.

Em nosso artigo anterior, observamos que o Decoy Dog tem uma distribuição surpreendente de respostas de IPv4. Agora, sabemos que isso era um artefato da codificação de resposta do Pupy. O uso de três bits aleatórios e um índice incremental, como os 7 bits principais do primeiro octeto em cada resposta, garante que os endereços IPv4 resultantes estão em intervalos específicos e que esses intervalos estão diretamente correlacionados ao número de respostas na resposta, o que, por sua vez, é determinado pelo tamanho dos dados que estão sendo transmitidos ao cliente. Em particular, o primeiro endereço IP sempre estará no intervalo 64.0.0.0/8, 128.0.0.0/8 ou 192.0.0.0/8.

Cada vez que o índice é aumentado, as opções para o primeiro octeto do endereço IP são deslocadas em dois. Especificamente:

- O primeiro endereço IP começará com 64, 128 ou 192, porque o índice é 0 e o comprimento é de no máximo 64. Como resultado, apenas os 3 primeiros bits são definidos no primeiro endereço IP da resposta.
- O segundo endereço IP começará com 66, 67, 130, 131, 194 ou 195, porque o índice é 1, o que adiciona 2 aos 3 bits superiores gerados aleatoriamente, e o bit superior da carga de dados pode ser 0 ou 1.
- O terceiro endereço IP começará com 68, 69, 132, 133, 196 ou 197 etc.

Podemos ver o resultado desse algoritmo para um número crescente de respostas na Figura 4 abaixo. Em particular, usamos um mapa de Hilbert para demonstrar como o primeiro octeto dos endereços IP está correlacionado com o número total de respostas para 3, 12 e 15 respostas.

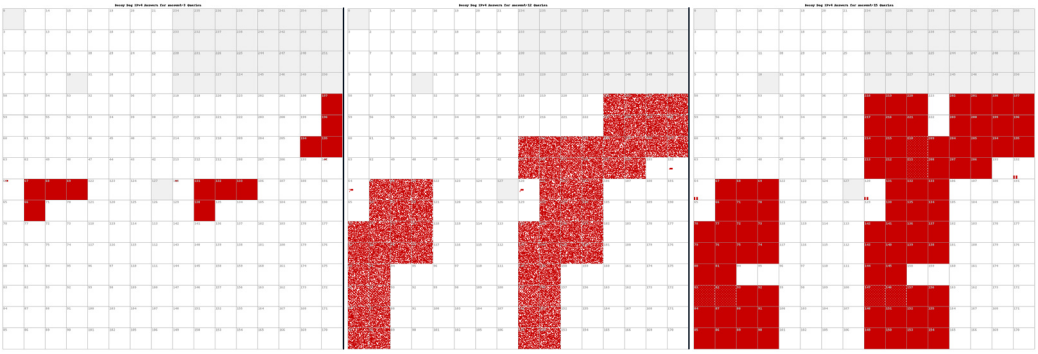

Figura 4. Mapas de Hilbert demonstrando a distribuição de endereços IPv4 em respostas de Pupy contendo 3, 12 e 15 respostas, respectivamente.

A estrutura dos endereços IPv4 permite que qualquer pessoa que observe a resposta completa reconstrua os dados transmitidos. Enquanto esses dados são criptografados, as respostas podem ser perfiladas usando a análise de comprimento e série temporal. Esse tipo de análise pode revelar informações sobre as comunicações, como veremos mais adiante neste artigo.

ANÁLISE PASSIVA DE DADOS

Embora as comunicações do Pupy sejam fortemente criptografadas, as informações necessárias para descriptografar e rastrear os pacotes são codificadas de forma reversível. Se as consultas e respostas de DNS forem coletadas, elas poderão ser analisadas de forma agregada para obter informações sobre a implantação e os clientes do Pupy. A coleta passiva de dados de DNS, comumente chamada de DNS passivo (também conhecido como pDNS), ocorre em muitos locais da Internet, incluindo resolvedores corporativos, resolvedores recursivos públicos, bem como servidores raiz e TLD. Nas seções a seguir, mostramos como a coleta passiva de consultas do Pupy pelo DNS pode ser explorada para obter informações sobre as comunicações.

Podemos recuperar uma grande quantidade de informações sobre um controlador do Pupy e seus clientes a partir do DNS passivo. Em particular, podemos recuperar

- o número aproximado de clientes ativos em um determinado momento,
- os tipos de trocas que ocorrem entre o servidor e os clientes,
- assinaturas da implantação, como o intervalo de adormecimento do cliente e
- um cronograma de trocas de chaves de clientes e atividades gerais.

Usamos essas técnicas para analisar o tráfego de nosso próprio servidor, bem como dos servidores do Decoy Dog. Isso nos permitiu entender o quanto o Decoy Dog é semelhante ao Pupy e o quanto os servidores são semelhantes entre si. Por fim, essas técnicas nos permitiram traçar o perfil de cada implantação do Decoy Dog. Os detalhes técnicos dos métodos usados são abordados em mais detalhes no Apêndice C.

ASSINATURAS DE CARGAS ÚTEIS DO PUPY

A natureza das comunicações entre um cliente e um servidor pode ser inferida até certo ponto por meio da análise passiva de dados. O vocabulário do cliente, ou seja, as diferentes cargas úteis que ele pode criar, é altamente restrito: há apenas nove tipos de comunicações com o cliente. Dois tipos compartilham o mesmo comprimento de carga útil, enquanto outro tipo pode ter vários comprimentos. Um ator pode criar eventos personalizados no Pupy, possivelmente criando diversidade adicional de comprimento de carga útil.

O servidor tem um vocabulário mais flexível e é capaz de transmitir vários comandos em uma única resposta de DNS, o que torna o perfil mais desafiador. No entanto, a grande maioria das comunicações em um sistema do Pupy está relacionada à inicialização da sessão, troca de chaves e pulsações do cliente para o servidor. As comunicações do servidor são dominadas por confirmações de solicitações de clientes, mensagens de erro, incluindo a necessidade de estabelecer uma nova sessão e trocas de chaves.

Como resultado, as assinaturas de diferentes tipos de comunicações podem ser criadas usando os comprimentos das cargas subjacentes das consultas e respostas do DNS. Essas assinaturas nos permitem separar a atividade de manutenção comum dos comandos significativos do servidor e isolar o uso de tipos de eventos personalizados. Elas podem ser usadas para traçar o perfil do comportamento geral de um cliente e servidor do Pupy observado passivamente, incluindo o Decoy Dog.

Na Figura 5 abaixo, mostramos um mapa de calor dos comprimentos de carga útil observados em consultas de clientes e respostas de servidores em nossos próprios dados do Pupy. Embora os comprimentos do servidor tenham mais variações devido a argumentos de comando e comandos concatenados, as comunicações do cliente são bem definidas. Para traçar o perfil das comunicações, usamos o comprimento da carga útil subjacente, incluindo somas de verificação e informações de nós. Como resultado, por exemplo, a confirmação do cliente (Ack) tem 19 bytes, e a confirmação do servidor tem 6 bytes. O Apêndice D contém tabelas para comprimentos comuns de carga útil de cliente e servidor e sua relação com os comandos.

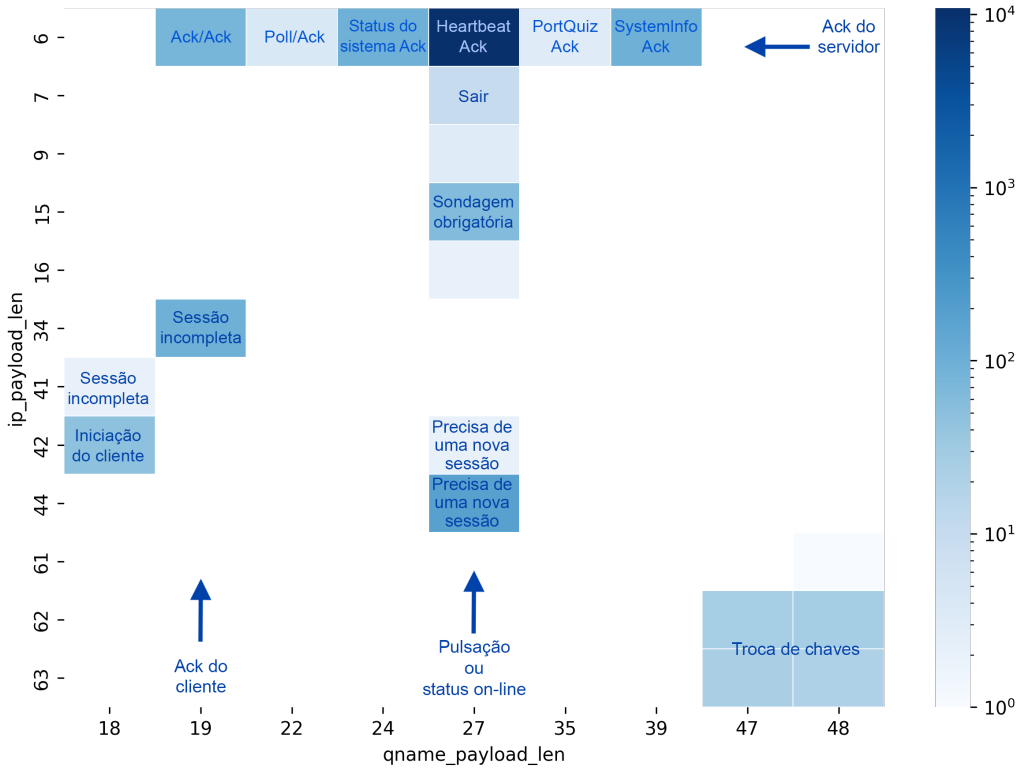

Figura 5. Uma distribuição anotada de pares de comprimentos de carga útil comuns observados no tráfego do Pupy. A carga útil são os dados criptografados transmitidos na consulta ou na resposta. Esse gráfico não inclui comandos complexos de C2 do DNS do servidor e as células sem anotação não são totalmente identificadas. O comprimento é em bytes.

Decoy Dog

As comunicações do Decoy Dog foram observadas não apenas nos resolvedores da Infoblox, mas em muitos resolvedores públicos e comerciais. Para entender melhor as operações do Decoy Dog e como o kit de ferramentas difere do Pupy, usamos outras coleções de DNS passivo para aumentar as nossas. No total, nossa análise cobre mais de 15 milhões de eventos de DNS durante o período de 29 de março de 2022 até 16 de junho de 2023. Além disso, sondamos ativamente os servidores de nomes e comparamos o tráfego de DNS coletado passivamente com o gerado por nosso próprio cliente e servidor do Pupy.

Utilizamos uma série de técnicas para entender melhor o Decoy Dog e suas operações. Também fizemos engenharia reversa de amostras encontradas no repositório público VirusTotal, o que validou nossas descobertas de DNS e revelou outros recursos. Nas seções a seguir, descreveremos nossa análise em detalhes e mostraremos os resultados. Os destaques desse trabalho são:

- O Decoy Dog não é o Pupy, mas uma grande refatoração que amplia significativamente os recursos do malware e ajuda a garantir a persistência em um dispositivo comprometido.
- Ele é operado por um punhado de atores, que empregam TTPs distintos e responderam de forma diferente à nossa revelação do kit de ferramentas em abril de 2023.
- O número total de dispositivos afetados é pequeno, com apenas quatro em um único controlador.
- Os novos controladores registrados desde abril de 2023 se adaptaram para mitigar as características descritas em nosso artigo original; isso inclui mecanismos de geofencing para limitar as respostas aos endereços IP do cliente a determinados locais.
- A análise de DNS provou ser uma ferramenta poderosa não apenas para detectar o Decoy Dog, mas também para entender seu uso e separá-lo do Pupy, o que, combinado com a engenharia reversa seletiva, fornece uma imagem robusta do Decoy Dog e da ameaça que ele representa.

TROCAS DE CHAVES

Conforme descrito anteriormente, uma sessão começa quando a troca de chaves é concluída e o valor SPI é definido. Em teoria, uma única sessão criptografada pode continuar indefinidamente, mas, na prática, há várias condições sob as quais o controlador exigirá que uma nova sessão seja estabelecida. Assim, uma única instância em execução do cliente pode ter muitas sessões. Usando as assinaturas de carga útil do Pupy, podemos determinar quando as chaves compartilhadas foram geradas entre um cliente e um servidor e fazer estimativas aproximadas do número de inicializações de clientes, seja a partir de um novo comprometimento ou de uma reinicialização do cliente, para cada controlador ao longo do tempo.

A Figura 6 abaixo mostra a cronologia das trocas de chaves para vários controladores do Decoy Dog. Há lacunas nas trocas de chaves observadas para alguns controladores. A última troca de chaves para o claudfront[.]net foi observada em dezembro de 2022, embora a atividade do cliente não só tenha continuado, mas também aumentado em 2023; mais de 70% de todos os valores únicos de SPI foram observados pela primeira vez em 2023. Da mesma forma, o controlador allowlisted[.]net não teve trocas de chaves de dezembro de 2022 até depois de nossa divulgação em abril de 2023. Por fim, o cbox4[.]ignorelist[.]com também mostra um longo período de tempo sem trocas de chaves, com um pequeno número ocorrendo diretamente antes de o domínio parar de funcionar. Suspeitamos que os atores reconfiguraram os clientes para realizar a troca de chaves em um transporte diferente do DNS.

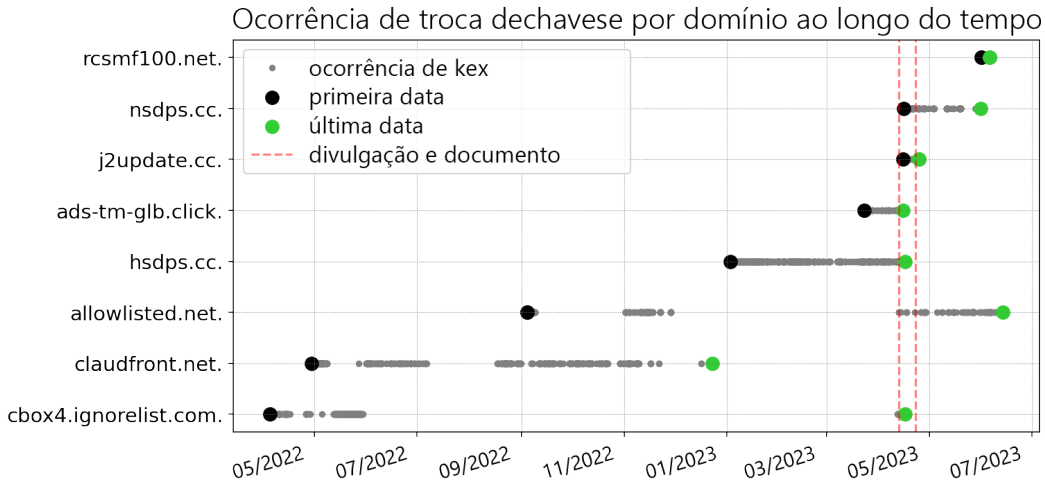

Figura 6. Cronologia das trocas de chaves observadas para domínios selecionados do Decoy Dog.

CRONOLOGIAS DOS CLIENTES

Além do número de clientes gerais, queríamos determinar quantos clientes ativos cada controlador mantinha por vez e por quanto tempo os clientes estavam se comunicando ativamente com o servidor. Utilizou-se o método de agrupamento dos valores de nonce descrito no Apêndice C. Essa análise resultou em insights importantes sobre as operações do Decoy Dog durante um longo período de tempo, conforme demonstrado nos gráficos a seguir. Em particular:

- Todos os controladores estão gerenciando um pequeno número de clientes ao mesmo tempo, sendo que alguns controlam apenas quatro, e todos, provavelmente menos de cinquenta.
- O domínio original, cbox4[.]ignorelist[.]com, é um dos maiores controladores e apresenta um salto nos clientes em vários pontos no tempo. Ele também mantém um pequeno número de clientes de longa data.
- O segundo controlador a ser observado, o claudfront[.]net, teve um aumento impressionante na atividade em fevereiro de 2023.
- O terceiro controlador a ser observado, o allowlisted[.]net, tem mantido consistentemente um pequeno número de clientes simultâneos.
- Os controladores ads-tm-glb[.]click e hsdps[.]cc transferiram clientes para novos controladores após nossa divulgação.
- O claudfront[.]net e o allowlisted[.]net não modificaram as operações em resposta à nossa divulgação, o cbox4[.]ignorelist[.]com cessou suas operações e tanto o hsdps[.]cc quanto o ads-tm- glb[.]click transferiram clientes para novos domínios.

Embora seja difícil estimar o número total de clientes durante todo o tempo, o pequeno número de clientes ativos simultaneamente indica que essas operações são altamente direcionadas. Isso também explica por que os fornecedores de segurança não detectaram a atividade e ainda não encontraram dispositivos infectados. Os clientes infectados estão presentes em um número muito pequeno de redes, aparentemente aquelas que não são capazes de identificar e bloquear as comunicações C2 no DNS.

Nos diagramas de gráfico de linha a seguir, representamos a atividade de um único cliente como uma linha e a chamamos de thread do cliente. O eixo y mostra threads de clientes distintos identificados por uma cadeia de nonce. Quando um cliente do Pupy é reiniciado, por meio de uma reinicialização ou de algum outro meio, um novo nonce será gerado e um novo thread será observado. Em alguns diagramas, há pausas claras na atividade, que provavelmente indicam reinicializações do cliente. O eixo x indica o tempo.

A Figura 7 mostra a atividade do cliente para o domínio inicial do Decoy Dog cbox4[.]ignorelist[.]com. O primeiro thread de clientes começou no final de março de 2022, e o thread mais longo durou quase um ano. Podemos ver que esse controlador inicialmente tinha apenas alguns clientes, mas uma mudança ocorreu em meados de maio de 2022, resultando em quase 40 clientes ativos simultaneamente. Aumentos semelhantes em threads de clientes ocorreram periodicamente, com o maior aumento mensal em agosto de 2022; no entanto, à medida que novos threads de clientes começaram, outros terminaram. Durante todo o ano de atividade, o número de clientes simultâneos parece estar abaixo de 50 em todos os momentos. Também podemos ver na Figura 7 que um quarto dos threads de clientes persistiram por seis meses ou mais, consistente com uma operação sustentada. Todas as comunicações foram interrompidas após a publicação no LinkedIn e não foram observadas novamente.

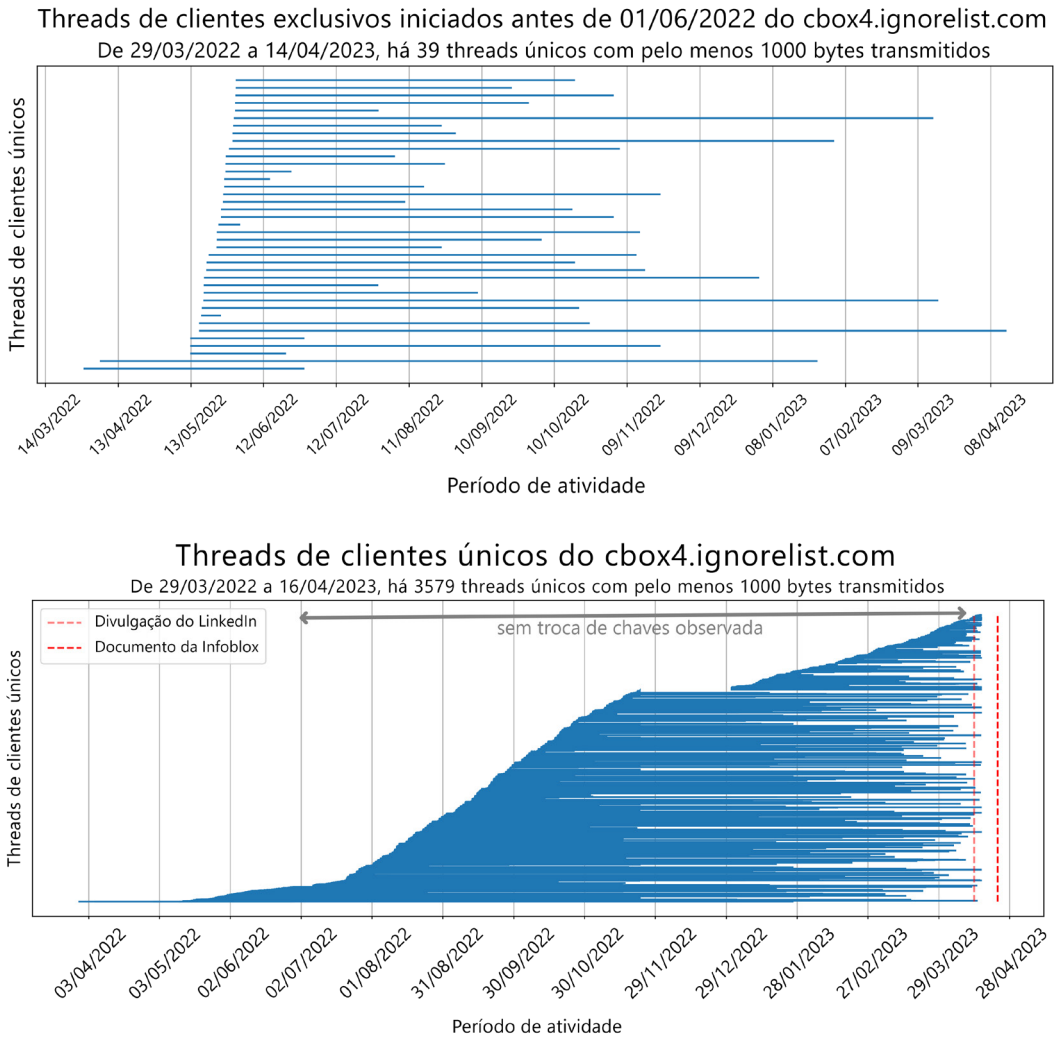

Figura 7. A figura superior mostra os clientes que estavam presentes antes de 1º de junho de 2022, e a figura inferior mostra os threads de clientes ao longo do tempo.

A atividade de DNS do claudfront[.]net, cronologicamente o segundo domínio do Decoy Dog a aparecer, é bem diferente da do cbox4. Conforme mostrado na Figura 8 abaixo, havia menos de dez clientes ativos simultaneamente nesse controlador até o início de fevereiro de 2023. Após esse período, o número de clientes aumentou substancialmente, embora não na medida que seria de se esperar de uma infecção generalizada. O momento desse aumento é pouco antes do envio de uma amostra binária contendo o domínio do controlador ao

VirusTotal em 13 de fevereiro.[15] Diferentemente do cbox4[.]ignorelist[.]com, não houve nenhuma alteração notável nas consultas do claudfront[.]net após nossa divulgação.

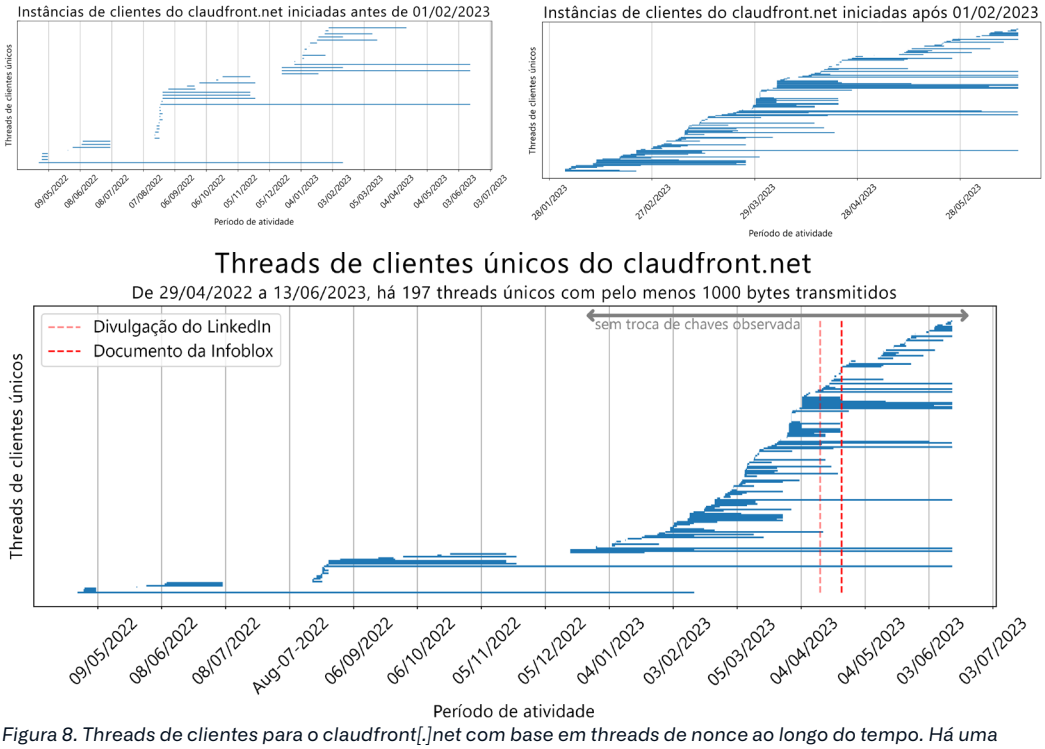

Figura 8. Threads de clientes para o claudfront[.]net com base em threads de nonce ao longo do tempo. Há uma mudança significativa no início de fevereiro de 2023, que é ampliada com imagens separadas mostrando períodos de tempo distintos.

O terceiro domínio, allowlisted[.]net, mostra outra variação no comportamento. Neste caso, o número de clientes é consistentemente pequeno: menos de dez em um determinado momento. Ao contrário do claudfront[.] net, não há mudança em fevereiro de 2023 e não há nenhuma amostra binária conhecida contendo o allowlisted[.]net disponível. Não há trocas de chaves observadas de meados de novembro de 2022 até pouco depois de nossa divulgação, o que coincide com o término acentuado da atividade do cliente e o reinício de vários threads em abril de 2023, conforme mostrado na Figura 9.

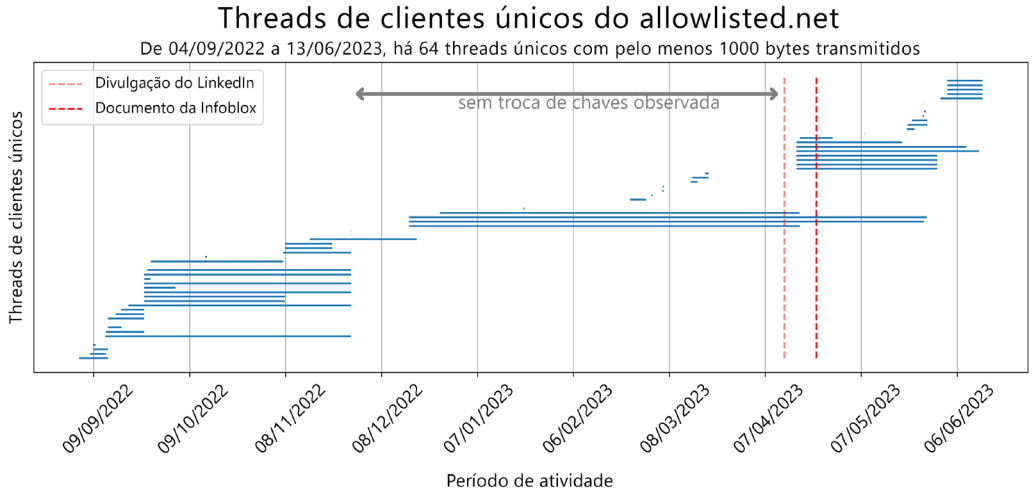

Figura 9. Threads de cliente para o allowlisted[.]net. Há um número muito pequeno de clientes no allowlisted[.]net historicamente, e isso não mudou desde a divulgação.

15 0375f4b3fe011b35e6575133539441009d015ebecbee78b578c3ed04e0f22568, enviado pela primeira vez em 2023

Por fim, observamos atividades relacionadas pelo hsdps[.]cc, nsdps[.]cc, ads-tm-glb[.]click e j2update[.]cc. Os domínios hsdps[.]cc e ads-tm-glb[.]click deixaram de operar após nossa divulgação nas mídias sociais, mas vários de seus clientes foram transferidos para nsdps[.]cc e j2update[.]cc, respectivamente. Descobrimos isso criando cadeias de nonce em todos os domínios ao longo do tempo e identificando threads que começaram a se comunicar com um controlador e terminaram com outro.[16]

Os novos domínios, nsdps[.]cc e j2update[.]cc, foram registrados menos de 48 horas após nossos anúncios nas mídias sociais. Podemos ver nos diagramas de threads de clientes que um conjunto de domínios cessa a atividade enquanto outros iniciam. Os controladores começaram a se comunicar ativamente com os clientes quase imediatamente depois disso. Após a descoberta da transferência de clientes por meio da análise de DNS, encontramos evidências em amostras binárias de um comando para fazer essa alteração, conforme descreveremos mais adiante.

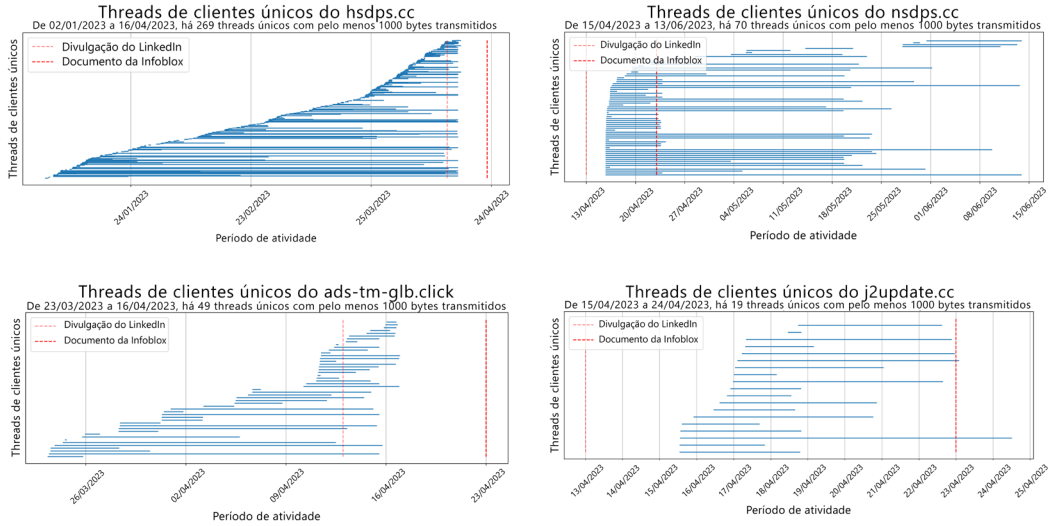

Figura 10. Uma comparação cronológica de quatro domínios do controlador do Decoy Dog. Os controladores hspds[.]cc e ads-tm-glb[.]. click cessaram as comunicações após a divulgação da Infoblox, e os domínios nsdps[.]cc e j2update[.]cc iniciaram as comunicações. Também observamos transferências de clientes entre esses domínios.

Desde nosso artigo original, vimos outros controladores se tornarem ativos, cada um com um número muito pequeno de clientes. O comportamento do cliente mostrado aqui, em conjunto com a resposta ao nosso anúncio, indica que o kit de ferramentas do Decoy Dog está sendo usado por vários agentes.

ASSINATURAS DE CARGA ÚTIL DO DECOY DOG

Decodificamos os comprimentos de carga útil de clientes e servidores de 15,5 milhões de respostas de consulta observadas no pDNS global em um período de 13 meses. Em seguida, comparamos as assinaturas do Pupy para cargas úteis cliente-servidor com os dados observados do Decoy Dog para entender o comportamento dos servidores. Embora tenhamos descoberto que as distribuições gerais de tráfego estavam alinhadas com o Pupy, havia diferenças definitivas. Os clientes do Decoy Dog utilizam um conjunto maior de solicitações, ou vocabulário, do que o encontrado no Pupy padrão.

A Figura 11 mostra as distribuições relativas de pares de comprimento de carga útil em todos os sistemas do Decoy Dog. Usando nossas assinaturas do Pupy, conforme detalhado no Apêndice D, podemos tirar algumas conclusões imediatas:

- Mais do que as nove cargas úteis esperadas do cliente estavam presentes.

16 A probabilidade de isso ocorrer aleatoriamente com um nonce aleatório de 32 bits é extremamente baixa, e o número de "transferências" de nonce de um controlador para outro para esses domínios foi alto.

- Havia comprimentos de carga útil do servidor que não havíamos observado em nosso laboratório.
- A maioria das comunicações está relacionada à manutenção de sessões e trocas de chaves.
- Um grande percentual das consultas aos servidores do Decoy Dog recebeu uma resposta de erro e mostrou uma variação consistente com a varredura por terceiros em vez de um cliente verdadeiro. A maioria delas ocorreu após nossos anúncios.

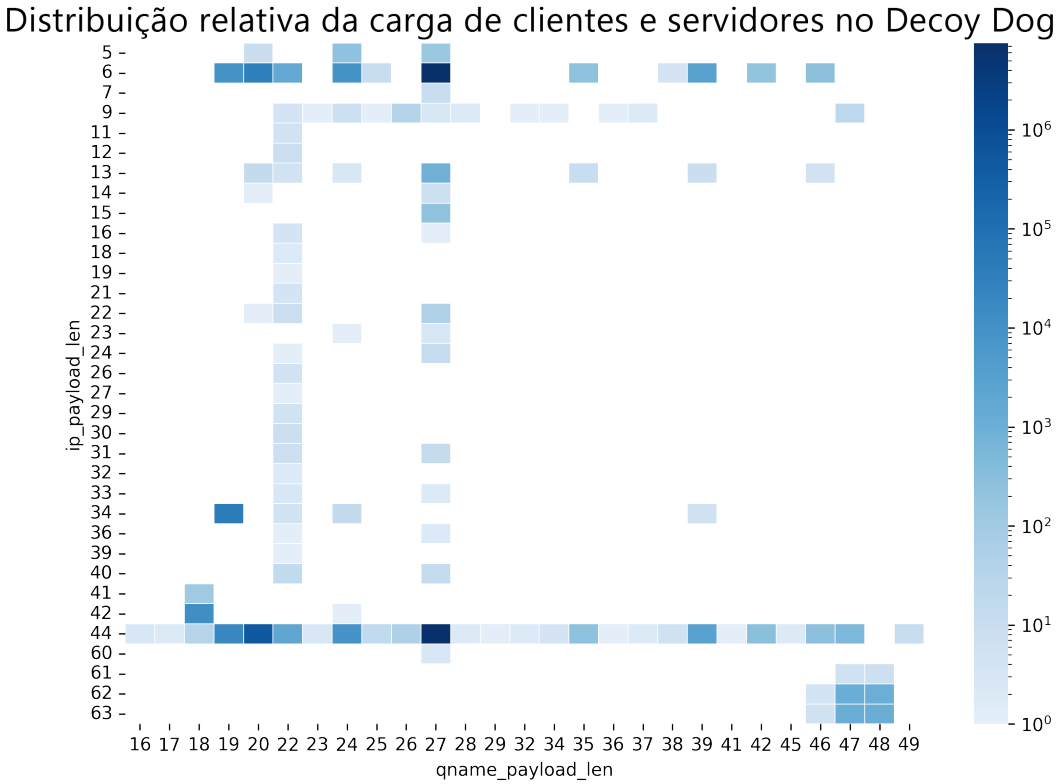

Figura 11. A distribuição relativa dos comprimentos de carga útil de clientes e servidores, conforme observado nas comunicações do Decoy Dog.

As cargas úteis exclusivas do cliente incluíam os comprimentos 20, 25, 38, 42 e 46. Algumas delas podem estar associadas a uma configuração de chave diferente ou a uma alteração nos parâmetros de pesquisa; não podemos determinar qual foi a comunicação, mas a variação existe. Além disso, houve comprimentos adicionais de carga útil de resposta além dos observados no Pupy. Mais notavelmente, o Decoy Dog tem uma carga útil de servidor de 13 bytes, que é vista ao longo do tempo em surtos de atividade. Não conseguimos determinar qual é essa carga, mas ela é consistente com um único comando que exige 8 bytes de dados para ser transmitido ao cliente. Também vimos várias respostas do servidor contendo uma carga útil de 5 bytes, outro comprimento não observado em nossos dados do Pupy e indicativo de um único comando que não exige transferência de dados para o cliente. A Figura 12 abaixo resume os pares de carga útil exclusivos encontrados no Decoy Dog e não vistos em nossos experimentos com o Pupy.

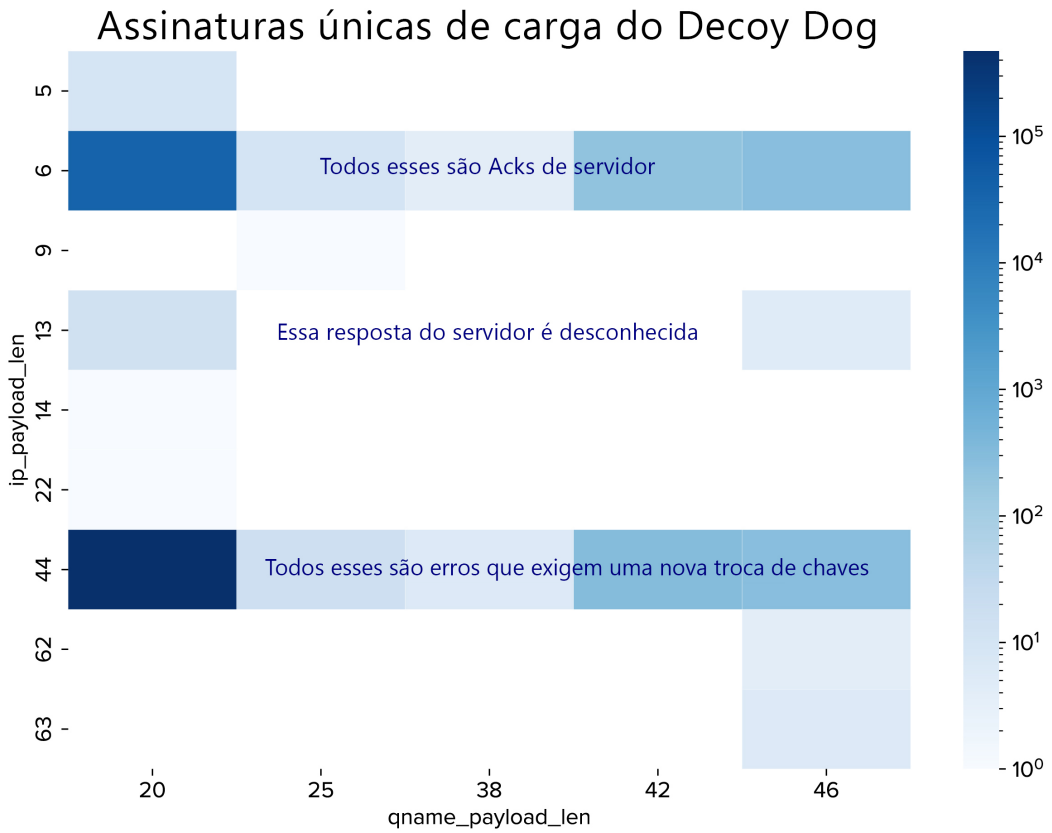

Figura 12. Um resumo dos pares de comprimento de carga útil cliente-servidor observados no Decoy Dog e não encontrados nas comunicações padrão do Pupy.

Também usamos séries temporais para identificar alterações nas configurações padrão. Em uma sessão estabelecida do Pupy, o cliente fará check-in a cada 30 segundos. Usando a análise estatística sobre a variação das consultas de pulsação do cliente, encontramos intervalos de pulsação de 2 minutos e 30 minutos, além dos 30 segundos padrão.

Como resultado dessa análise, conseguimos entender a natureza das comunicações de cada domínio do Decoy Dog, separando a manutenção de rotina dos comandos de acesso remoto. Também conseguimos isolar as personalizações prováveis do Pupy usadas entre e dentro de subconjuntos de servidores do Decoy Dog. Descobrimos que a grande maioria do tráfego do Decoy Dog são confirmações e erros de rotina, e que as comunicações de erro eram desproporcionais ao que esperávamos ver com base nas observações do Pupy. Compartilhamos os resultados de nossa investigação sobre esse fenômeno de respostas a erros na próxima seção.

COMPORTAMENTO CURINGA E GEOFENCING

Em nosso artigo técnico original, informamos que os servidores do Decoy Dog responderam a consultas de DNS repetidas. Isso continua sendo desconcertante. Ao tentarmos entender quando e como o Decoy Dog responderia a uma consulta que havia sido feita originalmente dias ou semanas antes, descobrimos um comportamento ainda mais surpreendente. Vários dos servidores do Decoy Dog não só respondem a repetições, mas também a qualquer consulta que seja consistente com a codificação do Pupy. No DNS, chamamos isso de resposta curinga. Enquanto um servidor do Pupy normal retornaria uma resposta NXDOMAIN ou SERVFAIL, o servidor do Decoy Dog normalmente retorna 15 endereços IP.

A Figura 13 abaixo mostra respostas a consultas aleatórias. Neste caso, colocamos a frase "wild" e "wildcard" dentro do nome da consulta e recebemos 15 respostas em resposta de dois servidores do Decoy Dog diferentes. As respostas são diferentes para cada consulta e estão em conformidade com o esquema de codificação do Pupy. Por meio de nossa pesquisa, aprendemos que o Decoy Dog está lidando com quase todos os erros dessa maneira em vez de retornar as respostas esperadas do NXDOMAIN. Consulte o Apêndice E para obter informações adicionais sobre tratamento de erros.

```
; <<>> 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 de resposta curinga de dois servidores autorizados do Decoy Dog. Em ambos os casos, os servidores responderam com 15 endereços IP consistentes com a codificação do Pupy para a mesma consulta aleatória, contendo as strings "wild" e "wildcard".

Ainda mais surpreendente, alguns dos servidores do Decoy Dog também respondem de forma diferente, dependendo do endereço IP do resolvedor recursivo que faz a consulta em nome do cliente. Na Figura 14, mostramos a repetição de uma consulta ao domínio do Decoy Dog nsdps [.] cc, que ocorreu originalmente várias semanas antes. Ao fazer a consulta por meio dos resolvedores públicos do Yandex, recebemos uma resposta contendo 15 endereços IP. Também recebemos 15 endereços IP dos resolvedores públicos russos TimeWeb. No entanto, dos mais de trinta resolvedores públicos que tentamos, nenhum outro retornou uma resposta. Esse tipo de comportamento é consistente com o geofencing, em que um servidor responde às consultas de DNS com base na geolocalização do endereço IP. Descobrimos esse comportamento em junho de 2023 e constatamos que alguns dos servidores respondiam somente quando roteávamos consultas de DNS por meio de endereços IP russos, enquanto outros respondiam a qualquer consulta bem formada de qualquer local. Esse tipo de resposta seletiva garante que o controlador se comunique apenas com clientes que parecem estar na Rússia. Sabemos que essa funcionalidade foi adicionada após a divulgação porque os controladores já haviam resolvido consultas dos resolvedores recursivos da 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. Uma comparação das respostas a uma consulta do Decoy Dog reproduzida dos resolvedores públicos da Yandex, dos resolvedores públicos da Hurricane Electric e do resolvedor autoritativo. Essas consultas foram feitas em sucessão por meio de um navegador Tor. Somente a consulta via Yandex recebeu uma resposta.

Quando uma consulta é feita para um nome de domínio que não pode ser decodificado usando a codificação padrão do Pupy (adicionamos caracteres extras para esse teste), os servidores nsdps[.]cc retornam um endereço IP que é essencialmente um sinkhole. Conforme mostrado na Figura 15 abaixo, alteramos ligeiramente a consulta para que ela não possa ser decodificada corretamente. Nesse caso, foi retornado um endereço IP aleatório no intervalo 172.0.0.0/8. Normalmente, o Pupy retornaria uma resposta 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. Uma consulta para um nome de domínio Pupy inválido para o controlador nsdps [.] cc retornará um endereço IP aleatório no intervalo 172.0.0.0/8 em vez da resposta NXDOMAIN esperada.

Parte desse comportamento pode ser explicada como um artefato da resolução de DNS do cliente. Quando um host é consultado no DNS, alguns resolvedores tentam resolver nomes de domínio potencialmente relacionados para se preparar para possíveis consultas futuras. Por exemplo, um resolvedor recursivo que recebe uma consulta para www[.]baddomain[.]com, pode tentar resolver baddomain[.]com além de www[.]baddomain[.]com. Vimos esse comportamento em nosso próprio servidor do Pupy ao rotear consultas de clientes por meio de alguns resolvedores públicos.

RESPOSTAS DE RÓTULO ÚNICO

Por padrão, o Pupy rejeita solicitações de entrada para rótulos que não correspondem à estrutura de uma comunicação com o cliente ou de uma consulta de ping estabelecida. No entanto, como explicamos na seção “Manuseio de nomes de domínio especiais” acima, o recurso de solicitação de ativação de DNS permite que um ator configure o servidor do Pupy para que ele responda a consultas de recursos personalizados. Nos logs globais do pDNS, identificamos consultas com um único subdomínio de rótulo. O único subdomínio desse tipo era o “m”, e levantamos a hipótese de que a resolução desses domínios era possível por meio da função de ativação. Pela natureza da função hash do ativador, um único endereço IP estático deve ser retornado para essas consultas. Encontramos esse comportamento em quatro domínios: hsdps[.]cc, nsdps[.]cc, j2update[.]cc e ads-tm-glb[.]click, e essa é outra característica compartilhada desse conjunto de domínios que não é vista em nenhum outro controlador. Cada uma delas retornou um único endereço IP; no entanto, em vez do endereço IP estático esperado, encontramos 104 endereços exclusivos nas respostas. Isso parece indicar uma diferença no recurso em relação ao Pupy padrão, mas não sabemos a finalidade.

ANÁLISE DE AMOSTRAS BINÁRIAS

Após nossas descobertas de DNS, examinamos as amostras binárias disponíveis no VirusTotal para determinar se a origem das diferenças em relação ao Pupy era facilmente visível. Ao analisar as importações e as tabelas de funções de duas amostras do Decoy Dog, identificamos uma assinatura exclusiva específica dos implantes do Decoy Dog que nos permitiu descobrir outras amostras do Decoy Dog. A engenharia reversa dessas amostras confirmou ainda mais nossas descobertas de que o Decoy Dog é substancialmente diferente do Pupy e que o código mais maduro pode ter sido criado por um segundo desenvolvedor. O cliente foi atualizado para o Python 3.8 e inclui uma série de novos transportes, criptografia atualizada, comandos personalizados e nova funcionalidade de DNS. A amostra relacionada a um controlador, claudfront[.]net, contém recursos não encontrados nos outros. Esta seção descreve algumas das principais descobertas e o processo; mais detalhes técnicos estão disponíveis no Apêndice F. Dados analíticos relacionados aos binários também serão adicionados ao nosso repositório do GitHub.

A primeira amostra foi carregada em setembro de 2022, e as outras foram carregadas em 2023, três delas após nossa divulgação. Extraímos e comparamos as configurações das diferentes amostras do Decoy Dog, o que mostrou que as chaves de criptografia diferem entre os servidores. Todas as amostras que se comunicaram com o cbox4[.]ignorelist[.]com contêm as mesmas chaves RSA e SSL, indicando que a existência de amostras diferentes não está relacionada a alterações de chaves de servidores. Uma lista completa das chaves decriptografadas pode ser encontrada no repositório do Github detalhado no Apêndice I. O certificado SSL mais antigo nas amostras foi gerado em 26 de dezembro de 2021 e pertence ao cbox4[.]ignorelist[.]com, o primeiro controlador observado.

Uma descoberta importante foi a de que a Decoy Dog inclui um código personalizado em seu cliente do Pupy que permite que os invasores enviem e executem módulos Java em tempo de execução, injetando-os em um thread JVM (Java Virtual Machine). Esse recurso não existe nas versões padrão do Pupy. Esse código foi encontrado em todas as amostras do Decoy Dog e é idêntico em todas as instâncias. As funções binárias restantes em todas as amostras conhecidas do cliente do Decoy Dog são idênticas às funções dos clientes do Pupy básicos.

A inclusão de módulos Java levanta mais perguntas do que respostas. Por padrão, o Pupy já é altamente capaz e é compatível com o uso de módulos Python prontos para uso. Expandir esses recursos e escrever módulos Python é um processo simples que não requer modificações no lado do servidor ou alterações no binário do cliente. Pode-se facilmente criar um módulo Python para executar e operar módulos Java. Por outro lado, injetar módulos Java em tempo de execução sem usar o jni.h (ou o restante da API Java/C padrão) não é uma tarefa trivial e requer conhecimento especializado. Portanto, é provável que a adição desses módulos Java permita que os invasores tenham como alvo sistemas que não executam Python, sistemas que executam uma máquina virtual Java privilegiada ou não monitorada ou cenários no qual os invasores buscam evitar deixar evidências na máquina ao não criar arquivos.

Os clientes também têm novas funcionalidades, que amadureceram com o tempo. O software do cliente é criado por meio da transformação de um arquivo de configuração Python em um determinado binário. O arquivo de configuração inclui definições, todas as chaves necessárias para as comunicações (RSA, certificados SSL, senhas etc.) e módulos Python do cliente. Os módulos encontrados nas amostras, que são descompactados e executados pelos dispositivos comprometidos, são muito diferentes do código do Pupy disponível publicamente.

A extração e a análise de módulos incorporados mostram uma história fascinante de desenvolvimentos e alterações personalizadas do Decoy Dog. Primeiro, um número considerável de módulos do Pupy foi simplesmente removido do Decoy Dog, possivelmente porque os atacantes os consideraram inúteis. Em segundo lugar, amostras semelhantes apresentam um grande número de diferenças nos módulos, às vezes com recursos muito diferentes. Em terceiro lugar, o grande número de alterações e a complexidade adicionada pelas novas funcionalidades mostram um tempo de desenvolvimento considerável e o ajuste fino dos recursos do Pupy. Além disso, a base de código e os módulos do Pupy foram portados do Python 2.7 para Python 3.8, o que melhorou a qualidade do código, a estabilidade das operações de memória e a compatibilidade com o Windows. As amostras incluem uma versão de cliente que muda de 3 para 4 ao longo do tempo; o cliente do Pupy mais recente disponível é a versão 2. Uma cronologia que resume as datas de envio em comparação com a maturidade do código e os principais recursos é encontrada na Figura 16 abaixo.

Ao analisar a natureza e o número de módulos alterados, conseguimos identificar que, de uma perspectiva de maturidade do código, a amostra com o hash ad186df91282cf78394ef3bd60f04d859bcacccbcdcbfb620cc73f19ec0cec64 é o primeiro binário do Decoy Dog disponível publicamente. Ele se comunica com o servidor de nomes cbox4[.]ignorelist[.]com. Embora compartilhe a maior parte do código com o Pupy,

essa amostra não foi carregada no VirusTotal até 27 de abril de 2023, vários dias após a publicação de nosso artigo. No entanto, com base no certificado SSL incluído, essa amostra pode ser datada de dezembro de 2021. O desenvolvedor adicionou uma funcionalidade específica de pesquisa, uma função XOR, novos transportes e compatibilidade total com comunicações de rede multithreaded. É interessante notar que vários módulos novos visam especificamente o Win32, embora todas as amostras até agora sejam bibliotecas do Linux. Nesse executável, o código responsável pelo tratamento das comunicações de DNS é o mesmo do Pupy padrão.

Com o passar do tempo, amostras se comunicando com o cbox4 [.] ignorelist [.] com se tornaram mais complexas. Em uma série de três amostras, um número crescente de módulos de comunicação foi adicionado, incluindo um módulo inteiro para comunicação usando fluxos bidirecionais sobre HTTP síncrono (BOSH), bem como reescritas completas dos módulos SSL, TCP e UDP. Os responsáveis pelo Decoy Dog também adicionaram vários scripts para portar os módulos de exploração e comunicação existentes para plataformas Windows, reescreveram o cliente picocmd responsável pelas comunicações DNS e implementaram várias melhorias de qualidade de vida e estabilidade no código antigo. As referências ao Windows no código sugerem a existência de um cliente Windows atualizado que inclui os novos recursos do Decoy Dog, embora todas as amostras atuais sejam voltadas para o Linux.

As versões posteriores também incluem um módulo de emergência, que permite que uma máquina comprometida entre em contato com um servidor de DNS de terceiros se o malware estiver sendo impedido de se comunicar com o servidor de C2 por um longo período de tempo. Esse módulo usa um DGA para selecionar domínios para o cliente consultar em serviços DNS dinâmicos gratuitos. Essas versões também permitem o bootstrapping para localizar o controlador C2, o estabelecimento de domínios de beacon e a incorporação de consultas CNAME no serviço de emergência. Mecanismos de persistência extensos, encontrados a partir da versão 3 do cliente, são recursos mais frequentemente associados a operações de inteligência do que àquelas conduzidas por agentes com motivação financeira ou red teams.

O código mais maduro, conectado ao controlador claudfront [.] net, inclui dois novos comandos chamados AlterDnsCncDomain e CompromisedNode. Conforme descrito anteriormente, determinamos, por meio da análise dos valores de nonce do cliente, que alguns dos atores do Decoy Dog haviam feito a transição de clientes para novos controladores após nossa divulgação. Com base no código-fonte do Pupy disponível publicamente, não vimos como isso era possível sem o uso de comandos personalizados. Parece provável que o comando AlterDnsCnCDomain seja a fonte dessas transições de clientes e, portanto, os controladores associados ao nsdps[.]cc podem estar usando o código mais avançado. A grande diferença entre esse código e os demais pode indicar que um novo desenvolvedor estava envolvido. O código inclui a versão 4 do cliente.

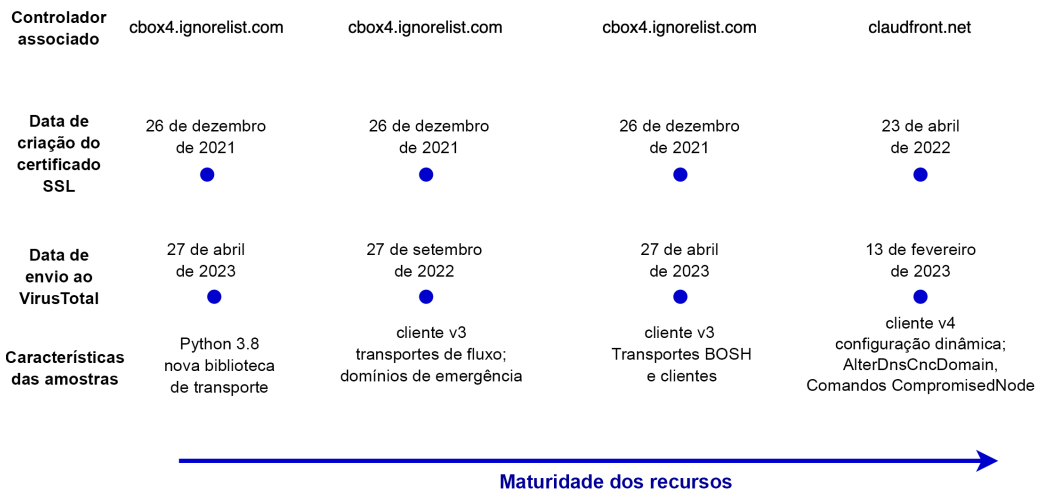

Figura 16. Uma cronologia de envios relacionados ao Decoy Dog no VirusTotal e a maturidade do código.

Vale a pena observar que, apesar de todos os aprimoramentos do Decoy Dog, as regras do YARA desenvolvidas para versões mais básicas do Pupy ainda conseguem detectar o malware. No entanto, elas não conseguem detectar se as amostras se desviam substancialmente do código e dos recursos conhecidos. Isso pode levar os pesquisadores de malware à falsa suposição de que as amostras do Decoy Dog são apenas Pupy básico, pois ambos os tipos de malware são sinalizados pela mesma regra. Por esse motivo, incluímos uma nova regra do YARA para o Decoy Dog no Apêndice G.

COMPARAÇÃO DE CONTROLADORES

A Infoblox está atualmente rastreando 21 domínios do Decoy Dog. Alguns deles tiveram pouca ou nenhuma atividade de C2 observável, e não os estamos divulgando no momento. Alguns controladores mudaram após nossa divulgação inicial nas mídias sociais, e o restante mudou depois que lançamos nosso primeiro artigo. Todos eles responderam interrompendo as operações, transferindo os clientes para novos controladores ou modificando o comportamento de "ping" que descrevemos no artigo. Alguns até adicionaram geofencing. Essas respostas, em conjunto com outras TTPs usadas, nos permitem concluir que há pelo menos três atores utilizando o kit de ferramentas no momento. Na Tabela 1 abaixo, agrupamos um subconjunto de domínios de controladores com base em seu comportamento e características semelhantes.

Grupo de domínios	Características
cbox4.ignorelist[.]com	• primeiro domínio ativo e provável fonte do kit de ferramentas do Decoy Dog • desativado após a divulgação • uso do DNS dinâmico do Afraid • intervalo de pulsação de 30 segundos • sem geofencing • pelo menos três iterações distintas de software do cliente • observado pela primeira vez por nós no final de março de 2022, mas pode ter estado presente já em dezembro de 2021 • cliente v2 e v3

claudfront[.]net allowlisted[.]net maxpatrol[.]net atlas-upd[.]com	• segundo conjunto de controladores ativos, a partir de maio de 2022 • operações continuadas após a divulgação • registrado na Namecheap • consultas ao ping12.<domain> antes que a comunicação criptografada remota fosse vista pela primeira vez • alterou a resposta do ping para uma resposta NODATA • Hospedagem de IPs russos • intervalo de pulsação de 30 segundos • sem geofencing • cliente v3 e v4 • existem algumas diferenças entre o allowlisted [.] net e o claudfront[.]net que podem indicar diferentes atores
hsps[.]cc nsdps[.]cc j2update[.]cc ads-tm-glb[.]click	• terceiro conjunto de controladores ativos, com início em dezembro de 2022 • transferiu clientes entre controladores após a divulgação • controladores originais estacionados • intervalos de pulsação de 2 minutos e 30 minutos • com geofencing após a divulgação • resposta de ping alterada para um único endereço IP de loopback não local • uso de um único rótulo de domínio: m • possivelmente cliente v4
rcmsf100[.]net	• observada pela primeira vez em junho de 2023 • compartilha hospedagem com o allowlisted[.]net • resposta de ping do NODATA • com geofencing

Tabela 1. Uma comparação de vários controladores do Decoy Dog.

DECOY DOG EM REDES DA INFOBLOX

A Infoblox determinou que nossos resolvedores foram acionados por um scanner de fornecedor de segurança que reproduziu as consultas do Decoy Dog. Uma combinação do comportamento do scanner e do comportamento do Decoy Dog criou o sinal detectado. A varredura na Internet tornou-se um negócio proeminente, e a varredura agora é responsável por uma grande quantidade de tráfego na Internet. Ele é realizado por agentes legítimos e mal-intencionados. Um estudo recente usou um telescópio darknet para entender o impacto dessas varreduras.[17] Embora a maior parte das varreduras seja limitada a varreduras de portas, que tentam identificar portas abertas no espaço IP global, há uma

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

ampla gama de outras atividades de varredura no ambiente. Por exemplo, há scanners que procuram diretórios abertos e resolvedores de DNS abertos. Algumas organizações documentam totalmente suas atividades de varredura, mas muitas não o fazem.

"Varredura agressiva" é uma atividade de varredura não autorizada ou de alto volume que pode degradar o desempenho de uma rede. Ela pode criar uma negação de serviço para uma rede ou, como no caso do Decoy Dog, criar eventos de segurança falsos.[18] A varredura agressiva beneficia o operador em detrimento das redes cujos proprietários não concordaram com a atividade. Em abril de 2023, as equipes de segurança de redes com detecções do Decoy Dog gastaram recursos significativos tentando encontrar a causa raiz dessas consultas de DNS para garantir que seus sistemas não fossem comprometidos. Essas consultas foram particularmente alarmantes, pois se originaram predominantemente de firewalls, e o setor de firewall expressou maiores preocupações sobre ataques a firewalls nos últimos meses. [19]

A maneira como as consultas do Decoy Dog chegaram aos nossos resolvedores e por que elas causaram um sinal semelhante a um sinalizador de C2 de malware direcionado é complicada. Para ajudar os defensores a reconhecer atividades semelhantes, forneceremos uma breve explicação e uma ilustração na Figura 17.

Para que a Infoblox receba consultas de DNS do Decoy Dog, a rede do cliente deve ter a Infoblox como seu provedor de DNS. Além disso, o cliente deve ter dispositivos de segurança, como firewalls, que tenham a filtragem de URLs de entrada configurada e o encaminhamento de DNS a partir desse dispositivo para nossos resolvedores. Esses critérios, por si só, são restritivos. Quando são atendidos, ocorre a seguinte sequência:

- O scanner tenta recuperar o conteúdo do malware de C2 diretamente de um endereço IP dentro da rede. Ele faz isso mesmo que essas comunicações de C2 de DNS não sejam conteúdo da Web.
- O appliance de segurança intercepta a solicitação e tenta resolver o nome de domínio.
- A solicitação de DNS é encaminhada para a Infoblox, que resolve a consulta e retorna a resposta. Se o domínio estiver em uma lista de bloqueio de DNS configurada pelo cliente, ele não retornará resultados.
- Se o domínio que está sendo varrido pelo fornecedor não for o Decoy Dog ou outro malware, ele será resolvido e, dependendo das regras do firewall, o conteúdo do site será devolvido ao scanner.

18 https://live.paloaltonetworks.com/t5/general-topics/spurious-hits-from-the-expanse-webcrawler/td-p/447239 , último acesso em 11/06/2023

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

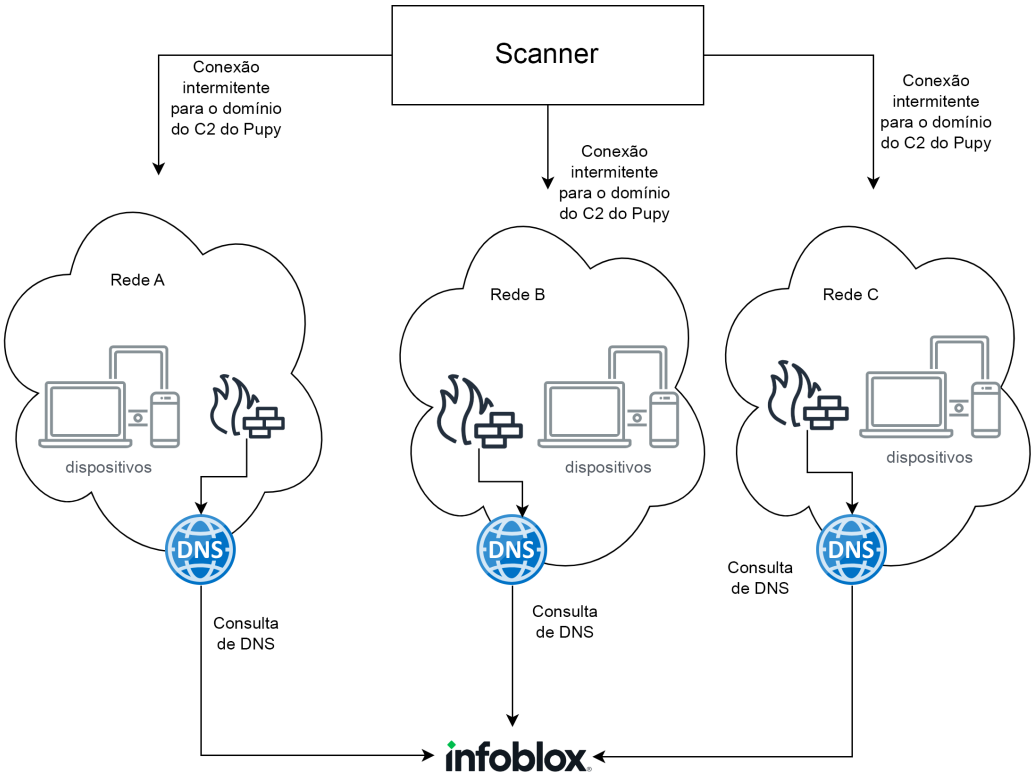

Figura 17. As consultas aos domínios de C2 de DNS do Decoy Dog foram feitas aos resolvedores da Infoblox a partir de dispositivos dentro de diferentes redes. Esses problemas foram causados por um scanner comercial e foram acionados de forma intermitente.

A Infoblox determinou que o fornecedor realiza varreduras mesmo que o endereço IP não tenha portas abertas conhecidas e que utilizará portas raras além das portas comuns. Não sabemos como o fornecedor decide quais endereços IP e portas usar. A consequência de uma varredura agressiva e indiscriminada dessa natureza é que dispositivos muito sensíveis podem parecer comprometidos, quando na verdade não estão. Embora o fornecedor pareça fazer uma varredura ampla e constante em busca de conteúdo, a Infoblox só observou consultas de DNS quando os critérios acima foram atendidos. Como resultado, embora o número de varreduras feitas pelo fornecedor fosse muito grande, o que é consistente com varreduras agressivas, resolvemos apenas um pequeno número de consultas de forma intermitente ao longo do tempo. Esse tipo de configuração também introduz a capacidade de um ator realizar reconhecimento em determinadas redes; descrevemos isso no Apêndice H.

A Infoblox Intelligence mantém registros históricos de todas as atividades de DNS e os utiliza para criar e manter estatísticas agregadas de atividade de domínio em nossas redes e no DNS global. Usamos essas agregações para identificar uma ampla gama de ameaças, incluindo comportamentos anômalos consistentes com os beacons de C2 de malware. Em particular, procuramos domínios para os quais as consultas, ao longo do tempo, ocorram em um número anormal de redes de clientes, tenham subdomínios consistentes com a exfiltração de dados e tenham um baixo número de consultas em relação ao comportamento esperado. Para isso, usamos estatísticas de todos os domínios que observamos ao longo de vários anos e trilhões de consultas de DNS.

Uma vez descobertos, o Decoy Dog e outros beacons de C2 de malware parecem altamente suspeitos, mas detectá-los é um grande desafio. Por sua natureza, o tráfego de DNS é altamente variável e contém um grande percentual de outliers, ou seja, domínios que

raramente são vistos e têm uma estrutura de nome de domínio consistente com a exfiltração de dados. No entanto, a exfiltração e o beaconing de DNS são muito raros fora das atividades de pen testing estabelecidas. Além disso, a assinatura de DNS do pen testing é bem diferente dos beacons de C2 de malware. Embora o Decoy Dog tenha provado ser o C2 de DNS de uma variante do Pupy RAT, um sistema de alto volume, ele parecia ser um beacon de baixo perfil, porque o tráfego foi injetado nas redes pelo fornecedor de segurança.

Embora as consultas do Decoy Dog aos nossos resolvedores tenham sido iniciadas pelo scanner, elas foram detectadas devido ao comportamento incomum dos servidores de nomes do Decoy Dog. Conforme revelado em nosso artigo anterior, os servidores de nomes do Decoy Dog responderam a consultas repetidas, embora às vezes de forma intermitente. Isso é inconsistente com o Pupy e outros protocolos de comunicação criptografados. Aprendemos agora que os controladores respondem a qualquer consulta bem formada. O comportamento combinado fez com que nossos sistemas detectassem um beacon intermitente de baixo volume. Esse tipo de varredura e comportamento de encaminhamento de DNS aberto em uma rede representa riscos adicionais à segurança de uma empresa. Ao permitir que uma parte externa acione consultas de DNS de dentro de uma rede, um invasor pode realizar o reconhecimento da rede. Descrevemos essa vulnerabilidade em mais detalhes no Apêndice H.

Conclusão

O Decoy Dog é claramente uma séria ameaça. Alguns agentes de ameaças usam o kit de ferramentas há mais de um ano, com as únicas detecções documentadas resultantes do monitoramento de dados de DNS. Ele é usado em operações altamente direcionadas, e observamos apenas seus controladores interagindo com um número muito limitado de clientes ativos. Embora tenhamos aprendido muito sobre o Decoy Dog, ele continuará sendo uma séria ameaça até que as vulnerabilidades usadas para estabelecer sua posição sejam identificadas e mitigadas.

Após nossa divulgação inicial do Decoy Dog, os agentes de ameaças responderam de várias maneiras para garantir o acesso contínuo aos sistemas das vítimas. Essas respostas incluíram a alteração do comportamento da resposta de DNS dos controladores, a adição de restrições de geofencing aos controladores e a transferência de clientes para novos controladores. Apesar dessas adaptações, a Infoblox continuou a rastreá-los e a aprender mais sobre o Decoy Dog e como ele difere do Pupy RAT.

As alterações feitas no Pupy para criar o Decoy Dog são consideráveis e indicam um agente de ameaças sofisticado. Essas mudanças incluem:

- O Pupy foi escrito em Python 2.7. O Decoy Dog requer o Python 3.8 e inclui vários aprimoramentos, inclusive compatibilidade com o Windows e operações de memória aprimoradas.
- O Pupy tem um vocabulário de comunicação muito limitado. O Decoy Dog amplia significativamente esse vocabulário com a adição de vários novos módulos de comunicação.
- O Decoy Dog responde a repetições de consultas de DNS anteriores, o que não acontece com o Pupy.
- O Pupy não responde às solicitações de DNS curinga, mas o Decoy Dog sim. Isso basicamente dobra o número de resoluções vistas no DNS passivo. Na verdade, o Decoy Dog responde a solicitações de DNS que não correspondem à estrutura de comunicação válida com um cliente.
- O Decoy Dog adiciona a capacidade de executar código Java arbitrário injetando-o em um thread JVM e adiciona uma série de novos métodos para manter a persistência no dispositivo da vítima.

A sofisticação dessas mudanças torna ainda mais curiosa a escolha do Decoy Dog para responder a qualquer consulta bem elaborada. Embora essa decisão pareça ser um erro à primeira vista, é provável que haja alguma justificativa ainda desconhecida para ela. No momento, é apenas mais um mistério do Decoy Dog.

No futuro, à medida que esses mistérios em torno do Decoy Dog forem sendo investigados, os defensores devem estar atentos ao seguinte:

- IPs no Pupy e no Decoy Dog são dados criptografados. Eles não representam IPs reais usados para comunicação. Todas as conexões com IPs reais associadas a malware são espúrias.
- Embora os IPs retornados nas respostas do DNS não sejam significativos, as próprias consultas e respostas do DNS têm informações significativas que podem ser usadas para rastreamento. No entanto, o volume de comunicação é baixo, o que significa que é necessário um longo histórico de registros para rastrear as comunicações detectadas.
- As respostas curingas do kit de ferramentas, combinadas com a análise agressiva do fornecedor de segurança, podem dar a aparência de comprometimento, quando na verdade não existe.
- Há uma regra YARA disponível que pode detectar o cliente do Decoy Dog em uma máquina vítima. Ela é capaz de diferenciar o Decoy Dog da versão publicamente disponível do Pupy.

O Decoy Dog foi detectado apenas usando algoritmos de detecção de ameaças de DNS. Até o momento, não há nenhuma divulgação pública que descreva as detecções do malware em si, e o escopo completo de suas capacidades ainda não é conhecido. O fato de ele ter operado sem ser detectado por tanto tempo destaca um ponto fraco que ocorre quando o setor confia excessivamente na detecção baseada em malware. A detecção e resposta de DNS são atualmente a única maneira de se defender contra o Decoy Dog e podem ser a melhor opção mesmo depois que as vulnerabilidades da vítima e do próprio Decoy Dog forem totalmente compreendidas.

Indicadores

Os indicadores do Decoy Dog relacionados aos controladores e amostras descritos neste relatório estão listados abaixo e disponíveis em nosso repositório aberto no Github.[20]

Grupo de domínios	Características
ads-tm-glb[.]click	Domínio de C2 do Decoy Dog
allowlisted[.]net	Domínio de C2 do Decoy Dog
atlas-upd[.]com	Domínio de C2 do Decoy Dog
cbox4[.]ignorelist[.]com	Domínio de C2 do Decoy Dog
claudfront[.]net	Domínio de C2 do Decoy Dog
hsdps[.]cc	Domínio de C2 do Decoy Dog
j2update[.]cc	Domínio de C2 do Decoy Dog
maxpatrol[.]net	Domínio de C2 do Decoy Dog

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

nsdps[.]cc	Domínio de C2 do Decoy Dog
rcmsf100[.]net	Domínio de C2 do Decoy Dog
13[.]248[.]169[.]48	IP do servidor de nomes de C2 do Decoy Dog
156[.]154[.]132[.]200	IP do servidor de nomes de C2 do Decoy Dog
194[.]31[.]55[.]85	IP do servidor de nomes de C2 do Decoy Dog
5[.]199[.]173[.]4	IP do servidor de nomes de C2 do Decoy Dog
5[.]252[.]176[.]63	IP do servidor de nomes de C2 do Decoy Dog
5[.]252[.]176[.]22	IP do servidor de nomes de C2 do Decoy Dog
5[.]252[.]179[.]18	IP do servidor de nomes de C2 do Decoy Dog
67[.]220[.]81[.]190	IP do servidor de nomes de C2 do Decoy Dog
69[.]65[.]50[.]194	IP do servidor de nomes de C2 do Decoy Dog
69[.]65[.]50[.]223	IP do servidor de nomes de C2 do Decoy Dog
70[.]39[.]97[.]253	IP do servidor de nomes de C2 do Decoy Dog
83[.]166[.]240[.]52	IP do servidor de nomes de C2 do Decoy Dog
4996180b2fa1045aab5d36f46983e91dadeebf d4f765d69fa50eba4edf310acf	SHA256 binário do Decoy Dog
ab8e333ef9bc5c5a7d1ed4cab08335861e150 b0639d3d0ca4c30b7def5cdccde	SHA256 binário do Decoy Dog
ad186df91282cf78394ef3bd60f04d859bcaccc bcdcbfb620cc73f19ec0cec64	SHA256 binário do Decoy Dog
6c8f413111f1abfee788dad4ee7cca37e0c259 7cca66d155af958c535faf55cc	SHA256 binário do Decoy Dog
0375f4b3fe011b35e6575133539441009d015 ebecbee78b578c3ed04e0f22568	SHA256 binário do Decoy Dog
6c8f413111f1abfee788dad4ee7cca37e0c259 7cca66d155af958c535faf55cc	SHA256 binário do Decoy Dog
t1fde0f101c9395f39ecd16430b41041a59107 c73c904087309fb8d0e8d87e0077129f3f	Assinatura Telfhash do Decoy Dog[21]

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

APÊNDICE A: PROCESSAMENTO DE COMANDOS DOS CLIENTES

A Figura 18 ilustra o ciclo operacional do cliente descrito no documento. O cliente transita repetidamente entre adormecer, sondar o servidor e responder aos comandos.

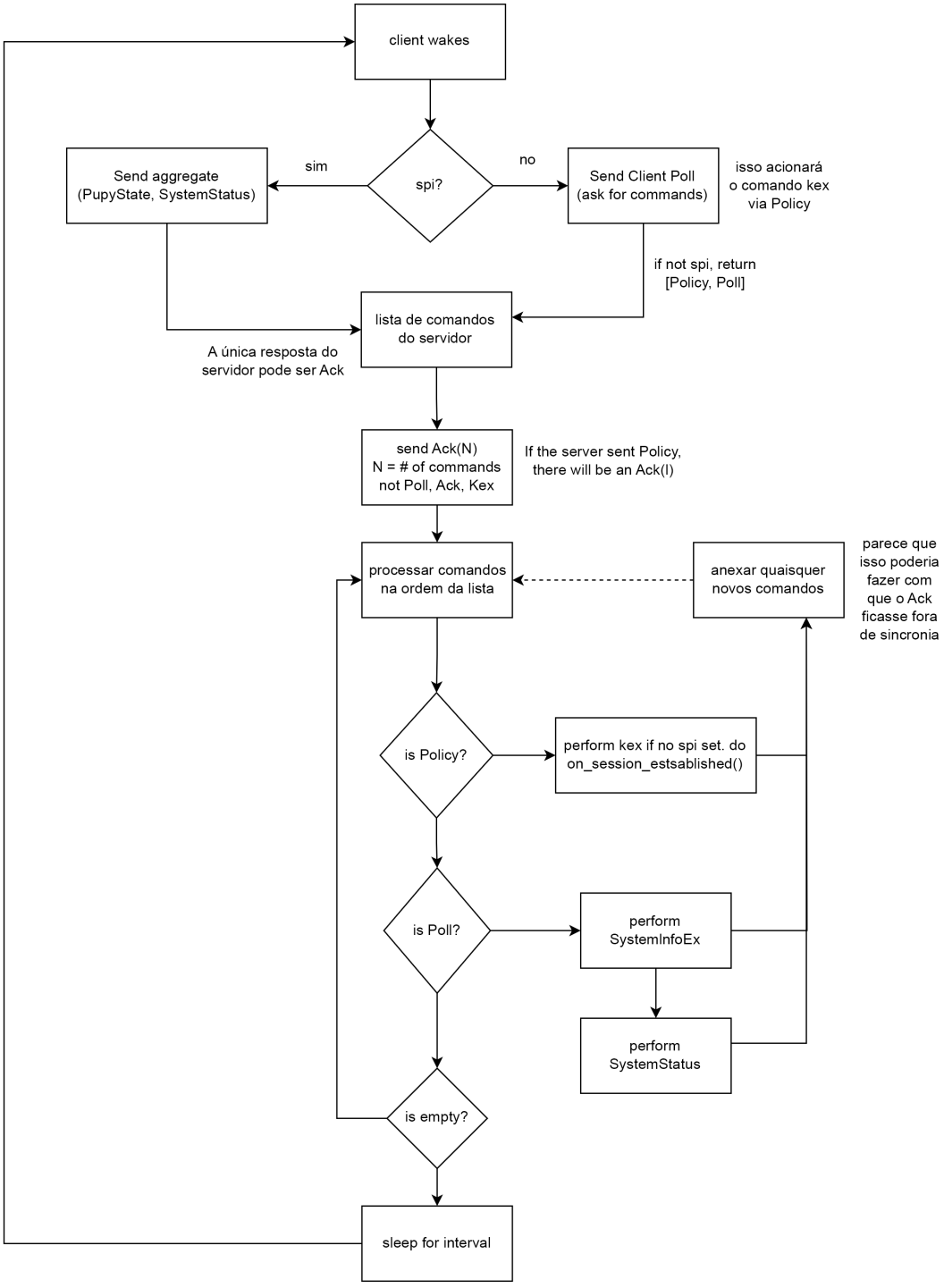

Figura 18. Fluxo de trabalho dos clientes.

APÊNDICE B: ESTRUTURA DA CARGA DE COMUNICAÇÃO

A estrutura da carga criptografada para o cliente e o servidor é idêntica, mas há diferenças em seu processamento. Em particular, o cliente inclui 13 bytes de informações do cliente em cada consulta, juntamente com a carga de dados, conforme descrito anteriormente.

Tanto o cliente quanto o servidor usam o termo comando para o tipo de informação que estão transmitindo ao receptor. Portanto, quando o cliente entra em contato com o servidor ao despertar, isso é considerado um comando do cliente. Os comandos são registrados para que o cliente ou o servidor possa aplicar um processamento específico aos dados. Pode haver mais de um comando em uma única comunicação, embora isso seja raro no cliente.

A carga enviada para codificação e transmissão tem o seguinte formato:

- uma soma de verificação de 4 bytes,
- pacotes de comandos concatenados, contendo uma identificação de comando de 1 byte e uma parte de dados variável dependente do comando.

O comprimento total da carga não pode exceder 52 bytes.

APÊNDICE C: RECONSTRUÇÃO DE CLIENTES A PARTIR DE DADOS PASSIVOS

Conforme descrito anteriormente, as consultas do Pupy incluem dados criptografados e dois valores codificados - o nonce e o SPI - que fornecem alguma segurança e permitem que o servidor solicite comunicações com o cliente. O valor do SPI é usado especificamente para identificar uma sessão em andamento no servidor e está presente em consultas após uma troca de chaves bem-sucedida. Como resultado, as consultas que contêm o mesmo SPI e que ocorrem próximas em termos de tempo são quase garantidas como sendo do mesmo cliente. Por outro lado, um único cliente terá muitas sessões e muitos valores do SPI ao longo do tempo, de modo que o SPI sozinho não pode distinguir clientes. Em vez disso, usamos os valores de nonce para separar as comunicações do cliente.

Quando o cliente é inicializado, ele gera aleatoriamente um valor de nonce de 32 bits para servir como ponto de partida. A cada pacote, esse nonce é incrementado pelo comprimento dos dados que estão sendo transmitidos. O servidor usa o nonce como uma pequena verificação de segurança, garantindo que ele aumente a cada consulta recebida, mas seu principal uso é descriptografar e interpretar corretamente a comunicação subjacente.
A partir de uma série de consultas observadas do Pupy, podemos decodificar esses valores de nonce e calcular o próximo nonce da série, conforme mostrado na Figura 19 abaixo.

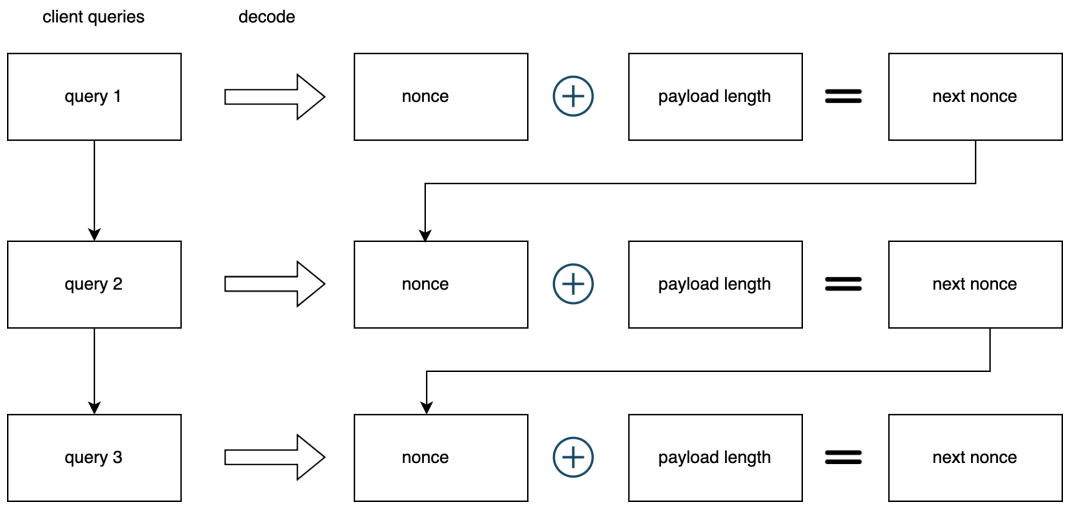

Figure 19. The relationship of nonce values within a series of Pupy queries.

Como resultado, podemos solicitar consultas de um único cliente e confirmar se uma série de consultas pertence a um único cliente. Na coleta passiva de uma implantação do Pupy, as consultas podem se originar de muitos clientes e se sobrepor no tempo. No entanto, ainda podemos separar essas observações em atividades separadas do cliente com um alto grau de confiança devido à construção do nonce. Como o nonce é usado para criptografar a carga, o desenvolvedor usou um gerador de números aleatórios fortes para criá-lo. Isso garante que cada cliente gerará valores nonce iniciais exclusivos.[22] O nonce é recriado cada vez que o cliente é reiniciado.

A segurança extra da criptografia também fornece um mecanismo para distinguir clientes em observações agregadas. Para fazer isso, calculamos o valor codificado de nonce e o próximo valor de nonce para cada consulta. Em seguida, encadeamos as consultas usando os valores sequenciais de nonce, conforme mostrado na Figura 20 abaixo. Embora os dados subjacentes permaneçam criptografados, podemos estimar o número de clientes e fazer observações sobre a duração de suas atividades. Além disso, podemos inferir informações sobre a própria comunicação usando os comprimentos de carga e comparando séries temporais entre clientes.

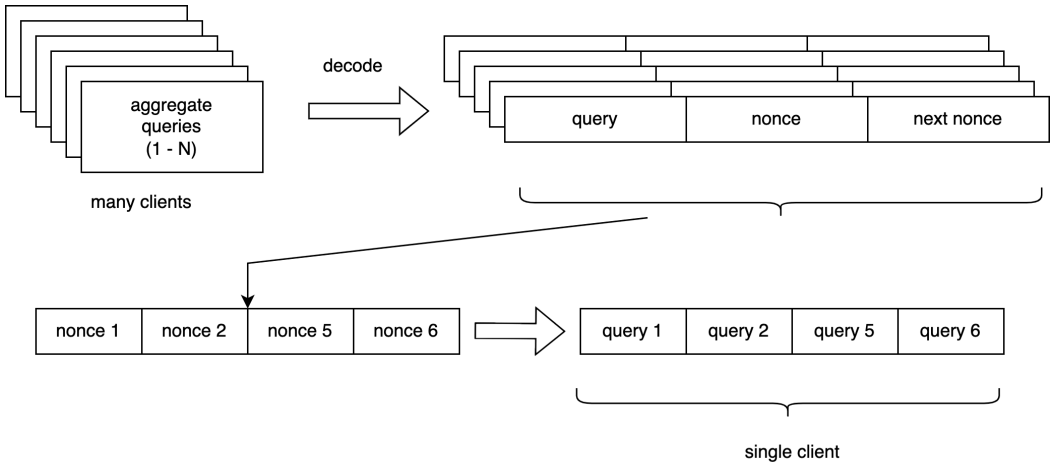

Figura 20. Separação de um thread de consultas do cliente de um conjunto agregado de observações usando os valores de nonce.

Há dois desafios com esse tipo de exploração: alterações no resolvedor de DNS do cliente infectado e perdas de pacotes. Por padrão, o Pupy usa o resolvedor de DNS padrão do cliente, e a escolha do resolvedor pode não estar sob o controle do ator. Se o cliente estiver em roaming, ele poderá utilizar resolvedores recursivos diferentes, dependendo do ambiente local. Em redes corporativas, eles podem usar a infraestrutura de DNS de fornecedores como a Infoblox, na qual as consultas de DNS serão forçadas sobre os resolvedores recursivos corporativos, independentemente das configurações do cliente.[23] Além disso, quando o DNS é transportado por UDP, a perda de pacotes é inevitável. O resultado é que é improvável que observemos todas as consultas apenas no DNS passivo, criando lacunas na cadeia nonce recuperada que podem ter um tamanho significativo.

No entanto, ainda podemos reconstruir os threads do cliente, aproveitando o fato de que o nonce é um valor gerado aleatoriamente. O desenvolvedor usou um gerador de números fortes, o que garante que é extremamente improvável que clientes independentes do Pupy

22 Há raras probabilidades de que o mesmo nonce possa ser gerado ao mesmo tempo por dois clientes diferentes.

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

compartilhem um valor de nonce. Além disso, como apenas 52 bytes de dados podem ser transmitidos por vez, e o valor de nonce aumenta de acordo com a carga, é improvável que duas cadeias de nonce geradas de forma independente se sobreponham. Como resultado, os clientes podem ser separados ordenando valores de nonce e agrupando aqueles que são estatisticamente similares. Um único cliente tem apenas um nonce por vez, o que nos permite estimar o número de clientes ativos em um determinado momento. Como mostramos no corpo principal do artigo, descobrimos que essa técnica é muito eficaz na recuperação de cadeias de consultas de clientes do Decoy Dog.

APÊNDICE D: ASSINATURAS DE CARGAS

As tabelas desta seção incluem os comprimentos de carga para comandos específicos que são comumente observados nas comunicações do Pupy. Em particular, elas fornecem o comprimento da carga criptografada para cada comando padrão do cliente. As cargas do servidor são mais flexíveis do que as dos clientes; as mais comuns são mostradas abaixo.

Comando do cliente	Comprimento da carga
Check-in do cliente (inicial)	18
Ack	19
Check-in do cliente (variante rara)	22
Status do sistema	24
Status on-line	27
Check-in do cliente (em sessão)	27
Teste de portas	35
Informações do sistema ampliadas	39
Troca de chaves	47, 48

Tabela 2. Comandos do cliente e comprimentos de cargas.

Comando do servidor	Comprimento da carga
Ack	6
Precisa de sessão: política, sondagem	42
Sessão incompleta: ack, política	34
Erro: mensagem, política, sondagem	44
Precisa de informações do sistema: sondagem	15
Troca de chaves	62, 63
Sair	7

Tabela 3. Comandos de servidores e comprimentos de cargas comuns.

APÊNDICE E: TRATAMENTO DE ERROS

O Pupy contém um tratamento personalizado para uma variedade de erros que o servidor pode encontrar. Um domínio que não seja decodificado corretamente ou que seja repetido resultará em uma resposta NXDOMAIN do servidor. O trecho de código abaixo mostra o processamento da consulta ao servidor. Se nenhuma resposta for retornada, ele retornará uma resposta NXDOMAIN.

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


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


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

Figura 21. Código-fonte do servidor do Pupy que processa as consultas de clientes.

No Decoy Dog, muitas consultas de clientes que deveriam resultar em um NXDOMAIN do servidor, em vez disso, retornam uma resposta, normalmente 15 endereços IP. Isso parece ser devido a uma alteração no código, em que o Decoy Dog responde a uma grande variedade de possíveis erros com um DnsCommandServerException internamente. O DnsCommandServerException resultará em uma resposta ao cliente, especificando o tipo de erro encontrado e instruindo o cliente a realizar uma nova troca de chaves seguida da transmissão de informações do sistema. O bloco de código para esse tratamento de erros é mostrado a seguir.

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

Figura 22. Código-fonte do servidor do Pupy que retorna um erro para o cliente.

Em comunicações normais entre um servidor do Pupy e um cliente, esse tipo de exceção será gerado quando não houver uma sessão ativa para um cliente conhecido. Também é usado quando a carga do cliente é inválida ou tem uma soma de verificação incorreta. Em todos os outros casos, o resultado é um NXDOMAIN.

APÊNDICE F: ANÁLISE DE AMOSTRAS BINÁRIAS

Binários do cliente do Pupy

Quando o servidor do Pupy é configurado pela primeira vez, ele compila os arquivos da biblioteca do Pupy e cria um arquivo de modelo estático para cada arquitetura. Esses arquivos de modelos são compactados, altamente ofuscados e desprovidos de todos os símbolos.

Os binários dos clientes podem, então, ser criados manualmente usando o pupygen.py no servidor. O script cria binários específicos do C2 agrupando bytes de configuração específicos (host remoto, tipo de transporte, sinalizador de depuração etc.) no modelo estático correspondente à arquitetura de destino e ao tipo de arquivo.

Os binários do clientes do Pupy oferecem uma variedade de funcionalidades avançadas e são capazes de atingir praticamente todas as plataformas, incluindo Windows, macOS, Linux, Solaris e Android. Em particular, eles podem permanecer residentes na memória, interagir com o servidor, oferecer recursos completos de shell reverso, criar cópias sem arquivo etc. Quando o binário é executado, ele cria cópias de si mesmo na memória, em um esforço para evitar a detecção e se tornar mais resistente às técnicas de eliminação de processos.

Exemplo de função de injeção de Java

Os binários do Decoy Dog incluem uma série de novas funções relacionadas à injeção de Java. Este é um exemplo de uma dessas funções.

```
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. Função do Decoy Dog parcialmente desmontada, tentando encontrar o thread de JVM em execução no momento para injeção.

APÊNDICE G: REGRA YARA PARA O DECOY DOG

A regra YARA a seguir pode ser usada para detectar as amostras do Decoy Dog que observamos a partir de julho de 2023.

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

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

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

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

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

Figura 24. Regra YARA para detectar amostras do Decoy Dog.

APÊNDICE H: VULNERABILIDADES DE SEGURANÇA EXPOSTAS

Quando um dispositivo é configurado para realizar uma consulta de DNS em uma conexão de entrada, ele permite que uma entidade externa controle parcialmente seu comportamento e seus recursos.[24] Em particular, essa configuração pode fornecer aos agentes de ameaças um meio de reconhecimento, resolução aberta e possível participação em um ataque de negação de serviço. Como o DNS é complexo, tanto os fornecedores quanto os operadores de rede podem não entender esses riscos. Embora os dispositivos de segurança que transmitiram as consultas que detectamos tivessem novos recursos, o uso de DNS nesses recursos expõe a rede ao reconhecimento e, potencialmente, a outras ameaças.

24 https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000PLRaCAO, último acesso em 11/06/2023

Um dispositivo em uma rede que serve consultas de DNS a qualquer entidade externa é conhecido como resolvedor aberto. Em alguns casos, um dispositivo pode retornar respostas, mas não resolver totalmente as consultas de DNS externas devido a uma ampla gama de circunstâncias. Em ambos os casos, esses dispositivos representam um risco para a própria rede e para o uso da rede para ampliar ataques de negação de serviço distribuída (DDOS). Os riscos dos resolvedores de DNS abertos foram bem documentados, e os resolvedores abertos são proibidos em muitos contratos de serviço, incluindo os da Infoblox, devido a esses riscos.

No caso das consultas do Decoy Dog, os dispositivos de segurança não eram resolvedores abertos, mas ainda permitiam que uma parte externa acionasse consultas de DNS. Esse tipo de configuração não pode ser usado para um ataque de amplificação, mas pode ser usado por um agente de ameaças para outros fins. Por exemplo, um agente de ameaças pode realizar reconhecimento contra uma rede; mostrado na Figura abaixo. O ator cria um domínio e configura o servidor de nomes correspondente para registrar as consultas recebidas. Em seguida, o ator usa um mecanismo de varredura para enviar nomes de domínios personalizados para se conectar à rede. No caso de uma pesquisa de resolvedor aberto, essas podem ser consultas de DNS. No caso do Decoy Dog, eram conexões HTTPS. Em ambos os casos, o dispositivo interno gera uma consulta de DNS que é enviada ao servidor de nomes controlado pelo ator. O ator pode, então, vincular o nome de domínio e o endereço IP original à consulta recebida. Embora esses tipos de ataques obtenham uma quantidade limitada de informações em cada tentativa, são mecanismos bem estabelecidos para mapear redes internas para ataques posteriores.

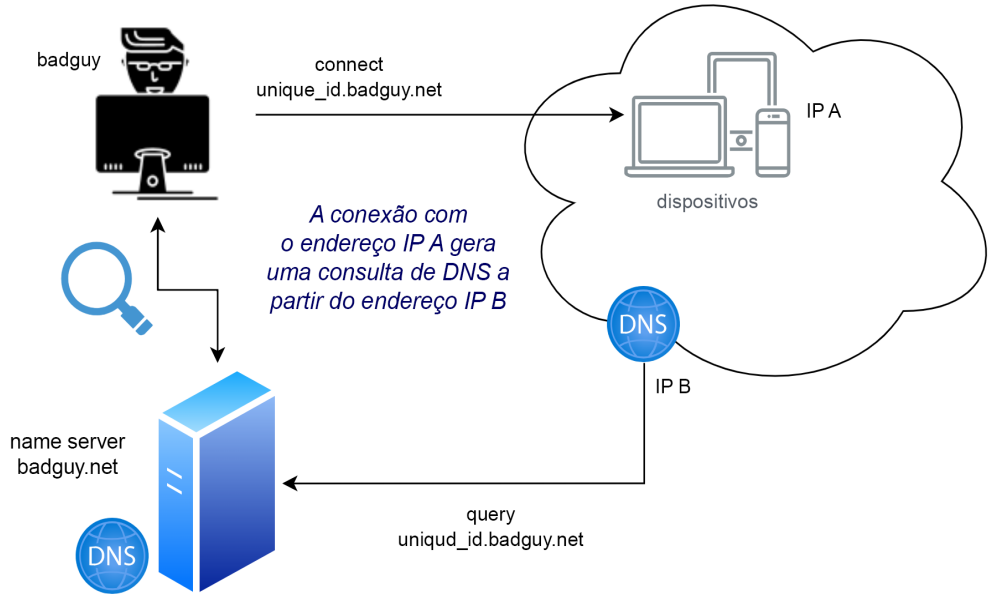

Figura 25. Um ator realiza o reconhecimento em uma rede criando nomes de domínios exclusivos que criam consultas de DNS ao seu servidor de nomes.

APÊNDICE I: DADOS DE PESQUISAS

Para nossa pesquisa, estabelecemos um servidor do Pupy e roteamos as comunicações entre o servidor e os clientes por meio de nossos resolvedores recursivos. Coletamos esses logs de consulta de DNS para nossa análise e estamos disponibilizando os logs para pesquisa. Os dados abrangem vários dias de atividade variável. Na maioria das vezes, controlávamos os clientes estabelecendo um proxy reverso, e os comandos eram enviados através de SSL. Suspeitamos que este também seja o caso do Decoy Dog. No entanto, exercemos todos os comandos disponíveis por meio de respostas de DNS do servidor.

Além disso, há períodos de tempo com vários clientes ativos simultaneamente e várias reinicializações de clientes. O escopo da atividade incluída deve permitir que os resultados descritos aqui sejam recriados.

Os dados estão disponíveis em nosso repositório público do GitHub infobloxopen: threat-intelligence.[25] Os logs de resposta a consultas contêm resultados de registros A e são empacotados em um arquivo csv que contém os seguintes campos:

- timestamp: a hora da consulta em segundos de época do Unix
- query: o nome de domínio totalmente qualificado transmitido na consulta do cliente
- resposta: o conjunto de endereços IP retornados pelo servidor
- client_payload_len: o número de bytes de carga na consulta, incluindo as informações do host
- server_payload_len: o número de bytes de carga na resposta

O repo também inclui os indicadores neste documento; indicadores adicionais estão disponíveis para os defensores mediante solicitação como informações TLP:RED. Além disso, estamos fornecendo dados resultantes de amostras binárias de engenharia reversa disponíveis no VirusTotal. Isso inclui:

- Parâmetros de configuração incorporados para cada amostra
- Chaves criptográficas e senha incorporadas para cada amostra
  - BIND_PAYLOADS_PASSWORD
  - DCONFIG_PUBLIC_KEY (only for client v4)
  - DNSCNC_PUB_KEY_V2
  - ECPV_RC4_PRIVATE_KEY
  - ECPV_RC4_PUBLIC_KEY
  - SCRAMBLESUIT_PASSWD
  - SIMPLE_RSA_PUB_KEY
  - SIMPLE_RSA_PRIV_KEY
  - SSL_BIND_CERT
  - SSL_BIND_KEY
  - SSL_CA_CERT
  - SSL_CLIENT_CERT
  - SSL_CLIENT_KEY
- Uma regra YARA e um hash TELF que podem detectar binários do Decoy Dog

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

infoblox.

O Infoblox une rede e segurança para oferecer desempenho e proteção incomparáveis. Reconhecida por empresas presentes na lista Fortune 100 e por inovadores emergentes, fornecemos visibilidade e controle em tempo real sobre quem e o que se conecta à sua rede, para que sua organização opere com maior velocidade e detecte ameaças mais cedo.

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

+1.408.986.4000
www.infoblox.com

RP-072023 v1