BESCHERM JE EIGEN STUKJE IOT

DE VELE MILJARDEN IOT-APPARATEN DIE AL WORDEN GEBRUIKT, ZIJN EEN KWETSBAAR DOELWIT VAN BOTNETS. HET CONCEPT 'DNS RESOLUTION REQUIRED' ZIEN JELTE JANSEN, CASPAR SCHUTIJSER EN ELMER LASTDRAGER ALS EEN MOGELIJKE OPLOSSING VOOR DIT PROBLEEM. MAAR DAN WEL ALLEEN VOOR BEPERKTE SCENARIO'S ALS IOT-APPARATEN IN EEN NETWERK THUIS.

door Jelte Jansen, Caspar Schutijser en Elmer Lastdrager
beeld Marc Kolle

REACTIES EN BIJDRAGEN
Voor reacties en nieuwe bijdragen van IT-experts:
Tanja de Vrede
020-2356415
t.d.vrede@agconnect.nl

DNS-FIREWALLS WORDEN INGEZET OM PHISHING EN MALWARE TEGEN TE GAAN, maar er zijn botnets die deze firewalls weten te omzeilen. Zo zijn er voorbeelden van malware die zelf de DNS-resolutie (het proces waarbij een IP-adres vertaald wordt naar een domeinnaam) uitvoeren (bijvoorbeeld Cutwail), de DNS-resolutie simpelweg helemaal overslaan (zoals Nugache) of zelfs de DNS-configuratie van het systeem volledig veranderen (bijvoorbeeld DNSChanger). In dit soort scenario's is de malware niet afhankelijk van de DNS-resolver (de server waarop de daadwerkelijke vertaling van domeinnaam naar IP-adres plaatsvindt) die door de systeembeheerder of ISP is ingesteld. DNS-firewalls bieden geen enkele bescherming tegen dergelijke methoden en daarom zijn er aanvullende beveiligingsmaatregelen nodig.

DRR ALS CONCEPT

Het concept van *DNS Resolution Required* (DRR) kan hierbij een oplossing vormen. In een notendop: DRR is een nieuw mechanisme voor edge-netwerken dat een client alleen toestaat om een netwerkverbinding met een extern eindpunt te initiëren als er een DNS-zoekactie op een geautoriseerde DNS-server aan vooraf is gegaan. Oftewel: een netwerk met DRR zorgt ervoor dat elke verbinding tussen een client en een externe dienst pas tot stand komt na een DNS-antwoord van een van de geautoriseerde DNS-resolvers van het netwerk, dus de resolvers die door de gebruiker, systeembeheerder of ISP zijn geconfigureerd.
Hoe zou dit concept er in praktijk uit kunnen zien? DRR analyseert al het DNS-verkeer op het netwerk. Standaard wordt al het uitgaande verkeer naar elk willekeurig IP-adres geblokkeerd, behalve DNS-verkeer naar specifieke geconfigureerde DNS-resolvers. Als er in reactie op een DNS-verzoek van een client een DNS-antwoord van een van die resolvers wordt waargenomen, wordt de firewall van het lokale systeem of lokale netwerk dynamisch geopend voor de IP-adressen in dat specifieke antwoord.

Belangrijk om te benadrukken is dat het DRR-systeem een aanvulling is op een DNS-firewall. Zonder firewall biedt DRR bescherming tegen DNS-lekken, malware die DNS-wijzigingen aanbrengt en malware die de lokale DNS-configuratie probeert te omzeilen, bijvoorbeeld om zichzelf te beschermen tegen inbraakdetectie op basis van DNS.
Maar de echte waarde van DRR ligt in de combinatie met een DNS-resolver die blokkeringslijsten met malware implementeert. DNS-query's kunnen alleen naar die resolver worden gestuurd en verkeer wordt alleen toegestaan als die resolver een geldig antwoord heeft geretourneerd.
In onderstaand figuur wordt geschetst hoe een DRR-systeem opereert.

VOORWAARDEN

Er zijn enkele voorwaarden om het DRR-systeem goed te kunnen laten functioneren. Zo moet het systeem inzicht hebben in het DNS-verkeer, vooral de DNS-antwoorden van de geconfigureerde resolver. In zijn meest elementaire vorm vereist dat het gebruik van standaard, onversleuteld DNS-verkeer over poort 53. Het gebruik van DNS over HTTPS (DoH) of DNS over TLS (DoT) zou resulteren in verbindingsfouten, omdat DRR de firewall niet zou openen voor de omgezette domeinnamen. Om met DoH of DoT te kunnen werken, moet het DRR-systeem ofwel een DNS-proxy zijn, ofwel een out-of-band-verbinding met de resolver nodig hebben om de benodigde informatie op te halen.

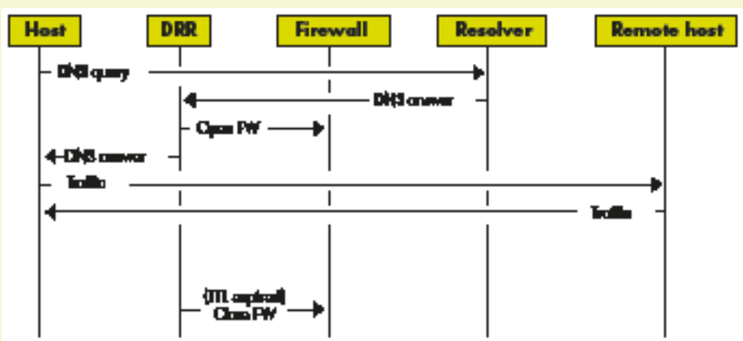
Ook moet het DRR-systeem kunnen communiceren met de firewall en geïntegreerd worden in de software van de firewall. Of het moet zo geconfigureerd zijn dat het de juiste open- en sluitopdrachten kan sturen naar de firewall voor omgezette domeinnamen en IP-adressen. Een modulair ontwerp van een gegeneraliseerd DRR-systeem is daarvoor noodzakelijk. Op die manier kan met verschillende firewallimplementaties worden gewerkt.

DRR heeft als gevolg dat verkeer dat geen gebruikmaakt van DNS om IP-adressen te vinden, wordt geblokkeerd. Dit omdat het DRR-systeem alleen verkeer toestaat van en naar IP-adressen die zijn waargenomen in DNS-antwoorden. Dit betekent dat peer-to-peersoftware en een aantal VoIP-oplossingen standaard zullen worden geblokkeerd. Hierbij worden namelijk IP-adressen binnen eigen protocollen gecommuniceerd, en die adressen zullen standaard door een DRR-systeem worden geblokkeerd.

PROTOTYPE

Tot zover de theorie. Tijd om het DRR-systeem in praktijk te testen. Hiervoor implementeerden wij DRR in een prototype, waarbij we een typische thuisomgeving met IoT-apparaten nabootsten. Hiervoor is gekozen omdat de meerwaarde van DRR vooral ligt in netwerken met IoT-apparatuur.

In een typische thuisomgeving met IoT-apparaten kan het verkeer worden gefilterd op de router, aangezien al het verkeer van de IoT-apparatuur via de router naar het internet gaat. Onderstaand figuur toont een veel voorkomen-



Figuur 1: Sequence diagram

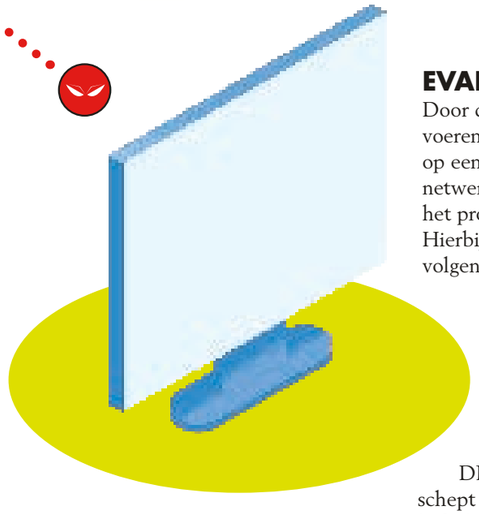
Beheer kan door ISP of andere dienstverlener worden gedaan

de opstelling, waarbij de DNS-resolver zich buiten het lokale netwerk bevindt. Deze resolver kan worden beheerd door de ISP of een DNS-dienstverlener.

Als de hosts hun queries rechtstreeks naar een externe DNS-resolver sturen, gebruikmakend van standaard plain-text DNS, is het mogelijk om DRR te implementeren door het DNS-verkeer op te vangen bij de router. In dit scenario beschermt DRR elk apparaat in het lokale netwerk en hoeven de apparaten zelf niet te worden aangepast of bijgewerkt.

In ons prototype wordt DNS-verkeer onderschept en doorgegeven naar de userland daemon van het prototype, de omgeving waar op de achtergrond de configuratie van de firewall wordt bijgewerkt, waarna het DNS-antwoord wordt verstuurd naar de client.

Dit betekent dat DNS-antwoorden worden vertraagd met de tijd die het kost om de firewallregels bij te werken. Een alternatieve aanpak zou geweest zijn om het DNS-antwoord onmiddellijk aan de client door te geven en de firewallupdate op de achtergrond uit te voeren. Maar dat zou tot gevolg hebben dat als de clientapplicatie contact probeert te maken met het externe IP-adres voordat de firewall is bijgewerkt, het verkeer nog steeds geblokkeerd zou worden.



EVALUATIE PROTOTYPE

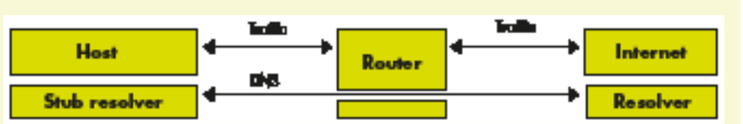
Door dagelijkse activiteiten uit te voeren terwijl de applicatie draait op een werkstation in een thuisnetwerkomgeving, hebben wij het prototype kunnen evalueren. Hierbij liepen we tegen de volgende problemen aan.

1. Vertraging

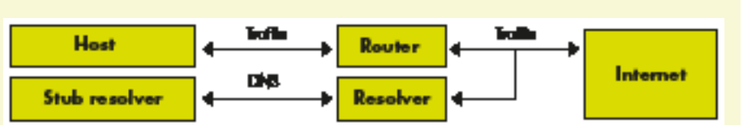
Browsen met het DRR-prototype was merkbaar langzamer dan wanneer DRR niet was ingeschakeld. Dat is niet vreemd, omdat elk DNS-antwoord werd onderschept en verwerkt voordat het werd afgeleverd bij onze browser. Dit vertraagt elke verbinding waarvoor een DNS-lookup nodig is en de relevante gegevens niet lokaal in de cache zijn opgeslagen. Aangezien het laden van een webpagina veel DNS-query's nodig kan hebben, is de vertraging aanzienlijk als een website voor de eerste keer wordt bezocht. In gevallen waarin een cliënt maar contact hoeft te maken met een paar externe diensten, zoals in een IoT-scenario, heeft deze vertraging echter geen merkbaar effect. Het doel waarvoor DRR ontwikkeld is - het beschermen van IoT-apparatuur - ondervindt hier dus geen last van.

2. Toegankelijkheid

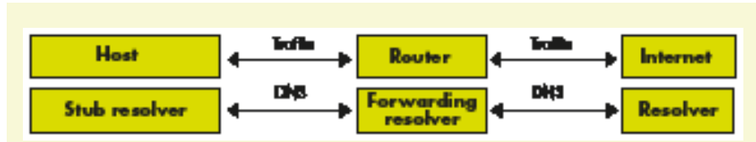
Het tweede probleem waar het prototype tegenaan liep had te maken met de codeertaal. Het prototype is namelijk geschreven in Python en gebruikt nfqueue om de DNS-pakketten te vertragen.
De Python-nfqueue-wrapper is niet direct beschikbaar op veel routers, wat het prototype moeilijk inzetbaar maakt op de meeste netwerken.
Het draaien ervan op onze werkstations was makkelijk, maar het inzetten op de OpenWRT-router was niet zo eenvoudig. Een implementatie in een meer low-level taal, of in een taal waar nfqueue-modules direct beschikbaar zijn zou dit probleem oplossen.



Figuur 2: Scenario A: remote resolver



Figuur 3: Scenario B: local resolver



Figuur 4: Scenario C: forwarding resolver

AUTEURS

JELTE JANSEN
is research engineer bij SIDN Labs.

CASPAR SCHUTIJSER
is research engineer bij SIDN Labs.

ELMER LASTDRAGER
is research engineer bij SIDN Labs.

3. Firewallregels

Een derde probleem was het goed regelen van de duur van de geldigheid van IP-adressen. De firewallregels die verkeer naar bepaalde IP-adressen toelaten, zouden niet voor altijd geldig moeten blijven. Daarom heeft een DRR-implementatie een mechanisme nodig om de regels op een bepaald moment weer in te trekken. Een voor de hand liggende mogelijkheid is om de Time to Live (TTL) van een DNS-antwoord te gebruiken als de geldigheidsperiode van de firewallregel. Dit zou echter mis kunnen gaan bij verbindingen die lang openblijven, aangezien er in die situatie geen nieuwe DNS-query's worden gedaan.

Om dit laatste probleem op te lossen, is ervoor gekozen om TCP-verkeer van bestaande verbindingen altijd toe te staan. TCP-verbindingen worden geïnitieerd door middel van een handshake, dus in theorie zou de standaardblokkeerregel een uitzondering kunnen maken voor pakketjes die geen deel uitmaken van deze handshake. Het is een immers voortzetting van een bestaande verbinding. Stateful firewalls houden deze informatie al bij, en laten pakketten van bestaande verbindingen door of kunnen dat relatief eenvoudig inschakelen.
Deze aanpak heeft het voordeel dat een DRR-implementatie de huidige status van verbindingen niet zelf bij hoeft te houden. Want dat zou een alternatieve oplossing zijn: het verkeer direct bijhouden.

Alternatieve scenario's

Naast de opstelling zoals die is gekozen voor de typische thuisopstelling, zijn er ook andere scenario's te bedenken. De resolver kan bijvoorbeeld door de router zelf worden gedraaid, en bevindt zich dan niet buiten het lokale netwerk. In dit scenario zou DRR ook kunnen worden geïmplementeerd als uitbreiding van de resolver zelf. Dan heeft DRR de beschikking over alle benodigde DNS-informatie zonder deze zelf te hoeven onderscheppen. Dit heeft als grote voordeel dat het niet afhankelijk is van het gebruik van plain-text DNS, waardoor DRR ook kan werken als het DNS-verkeer versleuteld is met bijvoorbeeld DNS-over-HTTPS (DoH) of DNS-over-TLS (DoT).

In een ander mogelijk scenario draait er een forwarding resolver op de router. In dat geval kunnen de filterregels worden gebaseerd op de resultaten van de forwarding resolver, en DRR zou kunnen worden geïmplementeerd als een uitbreiding op deze forwarding resolver.

Welk scenario ook wordt gekozen, de evaluatie van het prototype laat zien dat het concept van *DNS Resolution Required* kan werken in praktijk. Hoewel het enkele beperkingen heeft die het onpraktisch maakt voor algemeen gebruik op pc's, kan het een aanzienlijke netwerkbeveiliging toevoegen in meer beperkte scenario's, bijvoorbeeld bij bescherming van IoT-apparaten in een thuisnetwerk.