

میراکا: تشخیص حملات بات‌نت مبتنی بر ترافیک DNS در اینترنت اشیا

محمدتقی بهنام^۱، رضا جلایی^{۲*}

۱- کارشناسی ارشد، ۲- استادیار، دانشگاه جامع امام حسین (ع)، تهران، ایران

(دریافت: ۱۴۰۳/۰۲/۲۱، بازنگری: ۱۴۰۳/۰۶/۲۲، پذیرش: ۱۴۰۳/۰۷/۱۶، انتشار: ۱۴۰۳/۰۸/۰۱)

DOR: <https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.3.3.5>

* این مقاله یک مقاله با دسترسی آزاد است که تحت شرایط و ضوابط مجوز Creative Commons Attribution (CC BY) توزیع شده است.

نویسندگان



ناشر: دانشگاه جامع امام حسین (ع)

چکیده

تجهیزات اینترنت اشیا امروزه به سرعت در حال افزایش است و انواع و کاربردهای گوناگون نیز باعث شده که مدیریت آن‌ها پیچیده و بستری برای رشد و گسترش بات‌نت‌ها شود. در چند سال اخیر حملات گسترده‌ای از طریق تجهیزات اینترنت اشیا انجام شده است که بات‌نت میرای جزء شاخص‌ترین آن‌ها است به گونه‌ای که این بات‌نت به الگوی بات‌نت‌های اینترنت اشیا تبدیل شده است به دلیل اینکه بردار حمله DNS جزء محبوب‌ترین و نمایان بردارهای حمله است، این مقاله ضمن بررسی حملات سرویس نام دامنه (DNS)، در شبکه اینترنت اشیا، روشی را برای تشخیص بات‌نت‌های مبتنی بر DNS پیشنهاد می‌دهد. روش پیشنهادی میراکا، بر اساس محاسبه اختلاف زمان درخواست و پاسخ ترافیک DNS و مقایسه آن با مقدار حد آستانه محاسبه شده، سرویس‌های نام دامنه مخرب را تشخیص می‌دهد. تفاوت میراکا با روش‌های ارائه شده، در بررسی ویژگی‌های نام سرویس دامنه و تحلیل متنی نام دامنه است. میراکا با سنجش شاخص‌های ماتریس درهم‌ریختگی و ارزیابی شد. نتایج ارزیابی نشان داد که روش پیشنهادی میراکا، با تشخیص ۰٫۹۹۵ و دقت ۰٫۹۷۷ بات‌های مبتنی بر اینترنت اشیا را تشخیص می‌دهد.

کلیدواژه‌ها: اینترنت اشیا، تشخیص بات‌نت، حملات DNS، بات‌نت میرای

۱- مقدمه

نصب نرم‌افزارهای تبلیغاتی^۴، ارسال هرزنامه‌های گسترده^۵، کلاهبرداری کلیک^۶ به خطر اندازند [۳].

اینترنت اشیا به طور چشمگیری بر زندگی روزمره در حوزه‌های مختلفی تأثیر می‌گذارد، و هدف آن ارتقای کیفیت زندگی انسان‌هاست با این وجود، اینترنت اشیا در برابر حملات سایبری مختلف آسیب‌پذیر بوده [۴] و نگرانی‌ها در مورد امنیت دستگاه‌های آن به عنوان یک مانع مهم به حساب می‌آید و سدی در مسیر پذیرش دستگاه‌های اینترنت اشیا در نظر گرفته می‌شود. با توجه به بهره‌برداری بات‌نت‌ها از دستگاه‌های اینترنت اشیا و عدم توانایی شناسایی بات‌نت‌ها، تشخیص آن‌ها یک مسئله پیچیده و چالشی مهم است. مخصوصاً که روند آن‌ها،

اینترنت اشیا از همگرایی فناوری‌های بی‌سیم، سیستم‌های میکروالکترومکانیکی، میکروسرویس‌ها و اینترنت تکامل یافته است. این همگرایی به از بین رفتن شکاف میان فناوری عملیاتی و فناوری اطلاعات کمک کرده است [۱].

امروزه بات‌نت به عنوان یکی از خطرناک‌ترین انواع حملات سایبری به شمار می‌آید که از راه دور توسط کارورها به وسیله کانال‌های فرمان و کنترل مدیریت و کنترل می‌شوند [۲]. آن‌ها می‌توانند با استفاده همزمان از گروه‌های هماهنگ بسیار بزرگی از بات‌نت‌ها منافع اقتصادی و امنیت را از طریق حملاتی مانند انکار خدمات توزیع شده، جستجوی فراگیر شکستن^۲، سرقت هویت^۳،

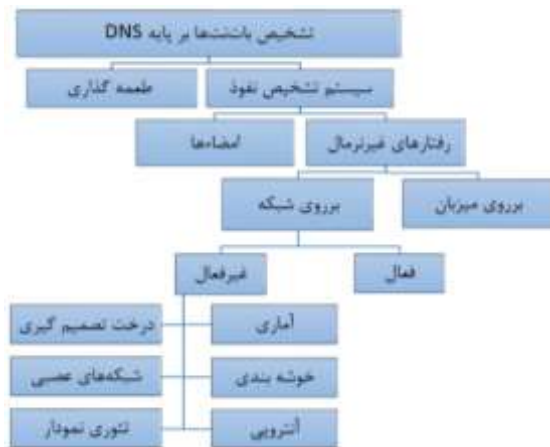
^۳ Identity Theft^۴ Adware Installation^۵ Mass Spamming^۶ Click Frauds

* رایانامه نویسنده مسئول: rjalaei@ihu.ac.ir

^۲ Brute-Force cracking

۲-۲- تشخیص بات‌نت‌ها مبتنی بر DNS

البان و همکارانش [۵] در خصوص تشخیص بات‌نت‌ها بر پایه DNS بر اساس خصوصیات، ویژگی بات‌نت‌ها و عملکرد آن‌ها، مطالعاتی انجام داده‌اند. شکل (۲) انواع طبقه‌بندی تشخیص بات‌نت‌ها در ترافیک DNS را نشان می‌دهد.



شکل (۲): ساختار درختی تشخیص بات‌نت‌ها بر اساس DNS [۸]

تشخیص بات‌نت‌ها در ترافیک DNS ابتدا به دودسته طعمه‌گذاری^۶ و سیستم تشخیص نفوذ^۷ تقسیم می‌شوند و خود سیستم تشخیص نفوذ نیز از دو بخش امضاها و رفتارهای غیرعادی تشکیل می‌شود. رفتارهای غیرنرمال بر اساس ترافیک میزبان و شبکه تقسیم‌بندی و ترافیک شبکه بر اساس دو حالت فعال^۸، غیرفعال^۹ بررسی می‌شود. در حالت فعال، بسته‌های خاص‌منظوره تولید و در شبکه تزریق و رفتار شبکه ارزیابی می‌شود؛ اما در حالت غیرفعال، ترافیک عبوری را از شش روش ارزیابی و بررسی می‌کند که شامل روش‌های آماری^{۱۰}، خوشه‌بندی^{۱۱}، آنتروپی^{۱۲}، درخت تصمیم‌گیری^{۱۳}، شبکه‌های عصبی^{۱۴} و تئوری گراف^{۱۵} است.

ابراهیم غفیر و همکارش [۹] به بررسی دامنه‌های مخرب^{۱۶}، با استفاده از فهرست سیاه پرداخته‌اند که یک روش محبوب در تشخیص بات‌نت‌ها است. برای همین منظور ترافیک DNS، پردازش شده و درخواست‌ها با لیست سیاه مطابقت داده می‌شود.

پنهان ساختن هویت‌ها خود، بویژه سرورهای فرمان و کنترل است [۵].

بات‌نت‌ها از سیستم DNS برای حملات خود استفاده می‌کنند و یکی از چالش‌های اینترنت اشیا تشخیص آن‌ها است [۵]. در این مقاله با تحلیل ترافیک‌های DNS اقدام به تشخیص بات‌نت‌های اینترنت اشیا می‌نماییم.

ساختار مقاله به این شرح است: در بخش ۲ به پژوهش‌های مرتبط اشاره شده و در بخش‌های ۳ و ۴ به انواع بات‌نت‌ها و سیر تکاملی آن‌ها در اینترنت اشیا پرداخته می‌شود و همچنین در بخش ۷ روش تشخیص پیشنهادی و در بخش ۸ و ۹ آزمایش و ارزیابی آزمایش‌ها بازگو خواهد شد و در بخش پایانی مقاله نتایج ارائه می‌شود.

۲- روش تحقیق پژوهش‌های مرتبط

در این بخش پژوهش‌های مرتبط با انواع حملات DNS و روش‌های تشخیص بات‌نت‌های مبتنی بر DNS مورد بررسی قرار می‌گیرد.

۲-۱- حملات DNS

مرجع [۶] حملات DNS را به چهار دسته تقسیم کرده است، دستکاری داده‌های DNS^۱، داده سیل‌آسای DNS^۲، سوء استفاده از DNS^۳ و ساختار سرور DNS^۴ و شکل (۱) فهرستی از ۱۱ حمله DNS را نشان می‌دهد.



شکل (۱): تقسیم‌بندی حملات DNS [۶]

حملات دیگری نیز وجود دارند که از آن جمله می‌توان به حمله سیل‌آسای DNS^۲، زیر دامنه تصادفی^۴ و حمله NXDOMAIN^۵ اشاره کرد.

حملات توزیع شده یک تهدید برای امنیت و ثبات ارتباطات و خدمات ارائه شده در شبکه‌ها هستند [۷].

^۶ Honey
^۷ Intrusion Detection System
^۸ Active
^۹ Passive
^{۱۰} Statistical
^{۱۱} Clustering
^{۱۲} Entropy
^{۱۳} Decision Tree
^{۱۴} Neural Networks
^{۱۵} Graph Theory
^{۱۶} malicious

^۱ DNS Data Tampering
^۲ DNS Data Flooding
^۳ DNS Flooding Attack
^۴ Random Subdomain
^۵ Non-Existent Domain

درگاه‌های دیگر، اندازه حجم بسته‌ها، آنتروپی بین بسته‌ها و توزیع حروف به‌عنوان ورودی به سیستم داده می‌شود و سپس با استفاده از الگوریتم رگرسیون لجستیک باتوجه به هوشمندی خود، تشخیص بات‌نت‌ها را انجام می‌دهد. در این روش به مجموعه‌ای از ویژگی‌های خاص حجمی پرداخته شده و آن‌ها را مورد بررسی قرار می‌دهد. ضعف روش پیشنهادی پروکوفیف و همکارانش عدم بررسی عامل‌های نظیر فهرست سیاه، تعداد نشانی‌های IP متمایز، نسبت دسترسی^۵، تعداد حروف نام دامنه، تکرار نام دامنه، تعداد نقطه در نام دامنه، تعداد اعداد است.

در بررسی مقاله‌ای دیگر برای تشخیص بات‌نت‌ها در تجهیزات اینترنت اشیا میدان^۶ و همکارانش [۱۴] از طریق بررسی عمیق کل ترافیک، یک سری ویژگی‌هایی را استخراج کرده و از طریق خودرمزگذار^۷ بر اساس شبکه عصبی به تشخیص ناهنجاری و بات‌نت‌ها در ترافیک شبکه تجهیزات اینترنت اشیا پرداخته می‌شود. در این روش، ترافیک DNS به طور اختصاصی بررسی نشده است و تشخیص بر اساس الگوریتم‌های در نظر گرفته شده است.

۳-۲- سیر تکاملی بات‌نت‌ها در اینترنت اشیا

با افزایش تعداد تجهیزات اینترنت اشیا، تعداد بات‌نت‌ها، حملات سایبری نیز به همان نسبت افزایش یافته است که این موضوع باعث ایجاد یک دسته‌بندی متفاوت در بات‌نت‌های سنتی و اینترنت اشیا شده است [۱۵].

۳-۲-۱- بات‌نت‌های سنتی

دارندگان بات‌نت‌ها یا هادایتگران آنها می‌توانند با استفاده از یک کانال مخفی مانند چت رله اینترنتی^۸ یا نظیر به نظیر^۹ ماشین‌های آلوده به بات‌نت را کنترل کنند. این روش کنترلی دسترسی را برای انجام فعالیت‌های مخرب مانند حمله‌های توزیع شده، انکار سرویس، هرزنامه‌ها یا سرقت اطلاعات صادر می‌کند [۳].

۳-۲-۱- بات‌نت‌های اینترنت اشیا

بات‌نت اینترنت اشیا مجموعه‌ای از تجهیزات اینترنت اشیا به خطر افتاده است، مانند دوربین‌های امنیتی، مسیرپاها، دوربین‌های مداربسته دیجیتال، پوشیدنی‌ها^{۱۰} و سایر فناوری‌های تعبیه شده^{۱۱}، آلوده به بدافزار. این بدافزارها به مهاجم اجازه می‌دهد تا

در ادامه، لیست سیاه دامنه‌های مخرب به طور خودکار به‌روزرسانی شده و ردیابی انجام می‌شود.

ضعف این روش این است که بررسی‌ها را فقط بر اساس لیست دامنه‌ها انجام می‌دهد.

خوزه سلوی و همکارانش [۱۰] روشی ارائه کرده‌اند که بر اساس تحلیل نام دامنه کار می‌کند. روش پیشنهادی آن‌ها یک رویکرد یادگیری ماشین با استفاده از جنگل تصادفی معرفی می‌کند، که از نام واژگان دامنه برای تشخیص دامنه‌های تولید شده استفاده می‌کند. روش پیشنهادی، از n -گرام همراه با سایر آمارهای به دست آمده از نام دامنه برای تشخیص آن‌ها استفاده می‌کند.

در روش n -گرام که مدل‌های تک‌کلمه^۱ و دوکلمه^۲، تا n کلمه‌ای بررسی می‌شود، نام دامنه از روش‌های تحلیل یادگیری ماشین بررسی می‌شود. سپس بر اساس عامل‌های خود نمره‌دهی می‌شود و در انتها نیز بر اساس معیار تعیین شده، مخرب بودن آن مشخص می‌گردد.

ضعف این روش این است که فقط تحلیل نام دامنه و به عنوان یک تحلیل گر متن در نظر گرفته می‌شود و دیگر ویژگی‌ها را مورد بررسی قرار نمی‌دهد.

یکی دیگر از روش‌ها که توسط جی هیون لی و همکارش [۱۱] ارائه شده است، تشخیص فعالیت بدافزار مبتنی بر گراف با تجزیه و تحلیل ترافیک DNS است. این روش یک سازوکار تشخیص فعالیت بدافزار مبتنی بر گراف^۳ است و از همبستگی پیوندی بین نام دامنه استفاده می‌کند.

در این روش بر اساس ارتباطات بین DNS ها و همبستگی بین آن‌ها و درخت تصمیم^۴ تشخیص داده می‌شود.

دیوید و بالزاروتی و همکارانش [۱۲] به بررسی یک سرویس تحلیل DNS غیرفعال برای تشخیص و گزارش دامنه‌های مخرب پرداخته‌اند. این روش، داده‌هایی از DNS سرورها جمع‌آوری می‌کند و آن‌ها را به‌صورت غیرفعال و غیربرخط و بر اساس ۱۵ ویژگی شامل چهار دسته بررسی می‌نماید. سپس مبتنی بر یک لیست، یک سری ویژگی‌های تشخیص، آموزش داده و نتایج توسط الگوریتم ژنتیک تحلیل می‌شود تا میزان مخرب بودن دامنه تشخیص داده شود.

برای تشخیص بات‌نت‌ها در تجهیزات اینترنت اشیا پروکوفیف و همکارانش [۱۳] روشی پیشنهاد می‌کنند که بر اساس آن پارامترهای نظیر درگاه مقصد، تعداد درخواست، تعداد درخواست‌های زوج، میانگین بین درخواست‌ها، درخواست بر روی

⁵ Access ratio

⁶ Y. Meidan

⁷ Auto-Encoders

⁸ Internet Relay Chat (IRC)

⁹ Peer-To-Peer

¹⁰ Wearables

¹¹ Embedded Technologies

¹ 1-gram

² 2-gram

³ Graph-Based Malware Activity Detection by DNS Traffic Analysis

⁴ Decision Tree

آلوده کردن تجهیزات اینترنت اشیا است. گونه جدیدی از این بدافزار مسیریاب‌های کوچک (SOHO Roter) اداری با نام‌های معروف مانند Huawei و Asus را مورد هدف قرار داده است. و در نمونه جدید، عملکرد اصلی بدافزار Gafgyt حمله به تجهیزات اینترنت اشیا و تحت کنترل گرفتن آن‌ها با بهره‌برداری از آسیب‌پذیری‌های شناخته شده برای انجام حمله انکار خدمات توزیع شده است.

۳-۴-۲- میرای (Mirai)

میرای [۱۸] یک شبکه ربات بات‌نت است، که هدف اصلی آن، دستگاه‌های برخط مانند دوربین‌های نظارت تصویری و مسیریاب‌های خانگی است.

بات‌نهایی مانند Masuta، Owari، DaddysMirai و Orion شامل کد حمله میرای هستند. همچنین سایر بات‌نهای اینترنت اشیا مانند IoT_Reaper/IoTroop و Satori نیز اگرچه رویکردهای متفاوتی دارند، بر مبنای چارچوب میرای هستند. در ادامه عملکرد و بردارهای حمله بات‌نت میرای از مرحله آلوده‌سازی تا مرحله حمله، بررسی می‌گردد.

۲-۵- عملکرد بات‌نت میرای

عملکرد بات‌نت میرای از مرحله آلوده‌سازی تا مرحله حمله از طریق هفت مرحله تشریح و وضعیت هر یک از مراحل در شکل (۳) مشاهده می‌شود [۱۹].

دستگاه‌ها را کنترل کند، دقیقاً مانند یک بات‌نت سنتی، اما برخلاف بات‌نهای سنتی، تجهیزات اینترنت اشیا به دنبال گسترش بدافزار خود هستند و دستگاه‌های بیشتری را آلوده می‌کنند. [۱۶].

۴-۲- انواع بات‌نهای اینترنت اشیا

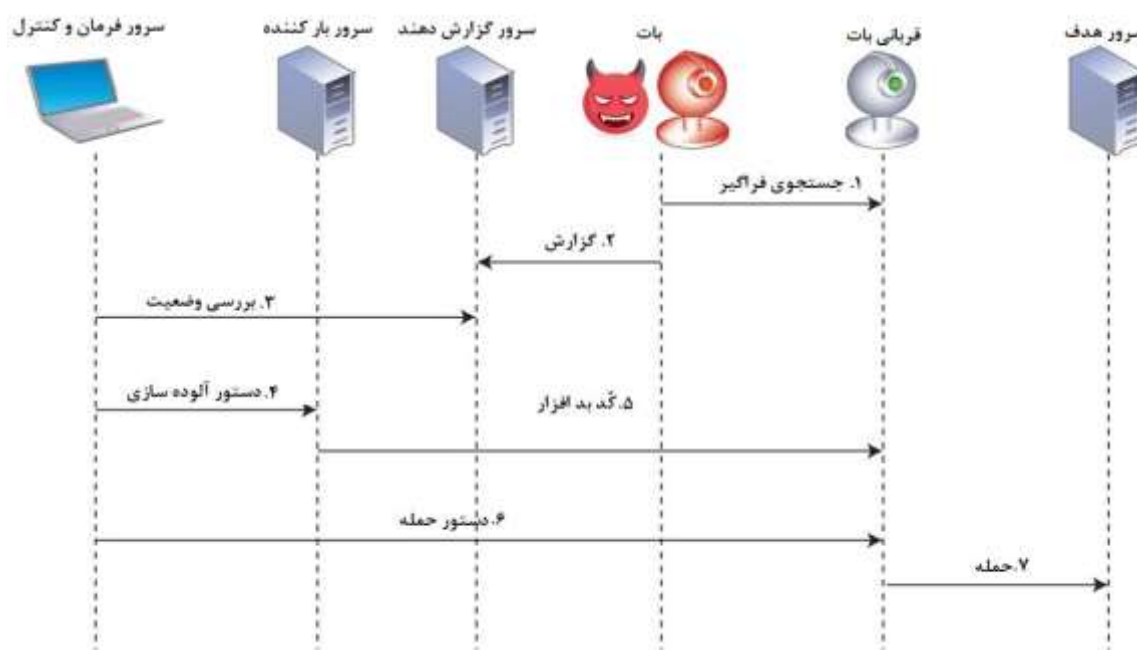
گسترش روزافزون اینترنت اشیا منجر به رشد مجموعه‌ای گسترده و روبه‌رشدی از تجهیزات شبکه‌ای مثل، شمارگرهای هوشمند، دستگاه‌های پزشکی، سنسورهای ایمنی عمومی و مانند این‌ها شده است. همین امر و همچنین تنوع کاربردهای این تجهیزات، منجر به گسترش و توسعه بات‌ن‌ها در آن‌ها شده است. در ادامه، برخی از بات‌نهای شاخص این حوزه مورد بررسی قرار می‌گیرد.

۲-۴-۱- Linux.Aidra

بات‌نت Linux.Aidra که با عنوان Linux.Lightaidra نیز شناخته می‌شود در سال ۲۰۱۲ توسط محققان امنیتی کشف شد. این بات‌نت اولین بار هنگامی مشاهده شد که در تعداد زیادی از حملات مبتنی بر Telnet به تجهیزات اینترنت اشیا مشارکت داشت.

۲-۴-۲- Bashlite

Gafgyt [۱۷] که با نام Bashlite نیز شناخته می‌شود یکی از معمول‌ترین انواع بدافزار است که از سال ۲۰۱۴ در حال



شکل (۳): عملکرد بات‌نت میرای [۱۹]

ابزار انجام شده است، حمله HTTP GET است که یک حمله GET سنتی محسوب می‌شود، HTTP Flood حمله به برنامه‌های در حال اجرا است و قربانی معمولاً نمی‌تواند ترافیک TCP درگاه ۸۰ را مسدود کند.

۲-۶- Linux/IRCTelnet

Linux/IRCTelnet (new Aidra) یک نوع جدید از بدافزار لینوکس بر روی تجهیزات اینترنت اشیا و تبدیل آن‌ها به ربات‌هایی با قابلیت حمله‌های توزیع شده است [۲۰].

۲-۷- Hajime

باتنت [۲۱] Hajime مشابه میرای از طریق پراکنده کردن خود توسط تجهیزات اینترنت اشیا غیرایمن که درگاه Telnet باز دارند فعالیت می‌کند. این باتنت با استفاده از کلمات عبور پیش‌فرض و همچنین لیست همانند ترکیب نام‌های کاربری و کلمات عبوری که در میرای است، فعالیت می‌کند.

باتنت Hajime برخلاف باتنت میرای پس از اینکه یک تجهیز اینترنت اشیا را آلوده می‌کند، با مسدودکردن چهار درگاه ۲۳، ۷۵۴۷، ۵۵۵۵ و ۵۳۵۸ که بیشترین درگاه‌های مورد استفاده برای آلوده‌سازی تجهیزات اینترنت اشیا هستند، اقدام به ایمن‌سازی این تجهیزات از حمله میرای و دیگر تهدیدات مشابه می‌نماید.

۳- روش تشخیص پیشنهادی

روش پیشنهادی مبتنی بر تحلیل فاصله زمانی درخواست‌های DNS است.

در این روش، زمان‌های پاسخ و دریافت درخواست‌های DNS محاسبه می‌شود. بر اساس اندازه حد آستانه به‌دست‌آمده، سالم یا ناسالم بودن این درخواست‌ها تشخیص داده می‌شود.

۳-۱- معماری روش تشخیص

برای تشخیص بات‌نت‌های مبتنی بر اینترنت اشیا، یک معماری سه‌مرحله‌ای مطابق *(Error! Reference source not found.)* پیشنهاد می‌شود. در این معماری، در مرحله اول که جمع‌آوری ترافیک نامیده می‌شود، ترافیک شبکه جمع‌آوری می‌شود، در مرحله دوم که پردازش ترافیک نام دارد، ابتدا با استفاده از روش‌های تحلیل ترافیک، درخواست‌های DNS جداسازی می‌شوند؛ در ادامه زمان بین ارسال و پاسخ این درخواست‌ها محاسبه می‌شوند. و در مرحله سوم یا تشخیص، نتایج تحلیل زمان درخواست‌ها با هم و از طریق محاسبه یک حد آستانه مقایسه می‌گردند. اگر این زمان از حد آستانه کوچکتر باشد، درخواست مربوط به DNS، سالم است و توسط یک سرویس معتبر DNS پاسخ داده شده است و اگر این زمان از حد آستانه

مرحله ۱: در ابتدا آدرس‌های آی‌پی عمومی تصادفی را از طریق پروتکل TCP درگاه ۲۳ یا ۲۳۲۳ اسکن می‌کند. آنگاه در یک حمله جستجوی فراگیر، رمز پیش‌فرض تجهیزات اینترنت اشیا که پیکربندی ضعیف داشته‌اند از طریق ۶۲ نام کاربری و رمز عبور ممکن بررسی می‌گردد.

مرحله ۲: ربات با کشف اعتبار صحیح به‌دست آمده، دسترسی و گزارش از آن، به سرور گزارش، ارسال می‌شود.

مرحله ۳: سرور فرمان و کنترل به‌صورت دوره‌ای، وضعیت سرور گزارش را بررسی می‌کند.

مرحله ۴: پس از تصمیم در مورد این که کدام ابزار آسیب‌پذیرتر و مستعد برای آلوده‌سازی است، مدیر بات یک فرمان آلوده حاوی تمام جزئیات لازم آدرس آی‌پی و معماری سخت افزار را صادر می‌کند مرحله ۵: فایل آلوده به دستگاه موردنظر وارد شده و به آن دستور می‌دهد، نسخه باینری مربوطه را بارگیری و اجرا کند. در این مرحله، نمونه جدید بات برای دریافت دستورهای حمله، با سرور فرمان و کنترل خود ارتباط برقرار می‌کند.

مرحله ۶: مدیر بات به تمام بات‌ها برای آغاز حمله به سرور هدف با صدور فرمانی ساده و با پارامترهای متناظر بر اساس نوع حمله، زمان حمله، آی‌پی آدرس، حمله به یک سرور هدف را شروع می‌کند.

مرحله ۷: بات‌ها از طریق یکی از بردارهای حمله^۱، حمله‌های توزیع شده از سرویس مورد نظر به سرور هدف را انجام می‌دهند.

۱-۵-۲- بردارهای حمله میرای

بردارهای اصلی حمله میرای [۱۸] که شامل ۱۰ نوع حمله است بر اساس بیشترین رتبه بهره‌برداری تشریح می‌گردند. در ادامه، به برخی از این بردارهای حمله‌ها اشاره می‌شود.

• حمله DNS

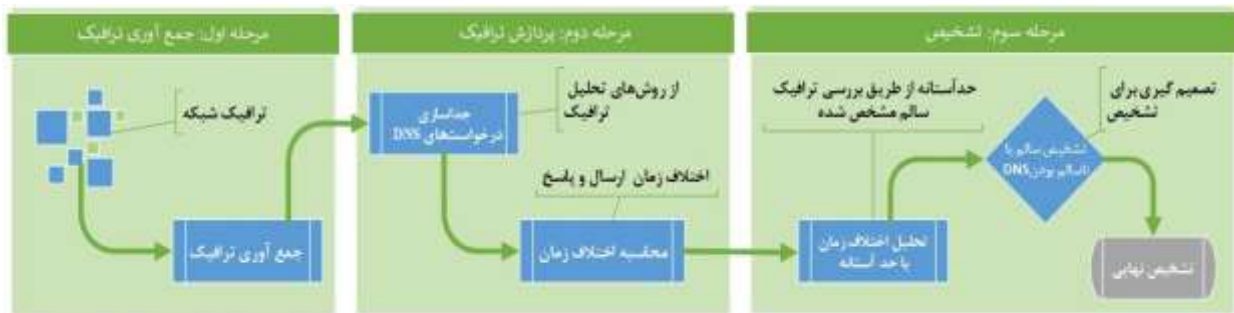
در این حمله، آدرس آی‌پی مشخص شده استفاده نمی‌شود، در عوض حمله‌کننده باید دامنه مورد حمله را مشخص کرده و سیل درخواست‌های DNS در دامنه مشخص شده ایجاد می‌کند. این حمله به‌خودی‌خود سیلی از دامنه‌های فرعی در دامنه مشخص شده با فرمت \$STRING.domain.com است. دفاع در برابر این بردار حمله بسیار دشوار بوده؛ زیرا یک سازمان نمی‌تواند به‌سادگی درگاه ۵۳ و آی‌پی مبدأ را مسدود نماید.

• HTTP

حمله HTTP در میرای بسیار انعطاف‌پذیر و با چندین حمله سفارشی‌سازی شده است که دفاع از آن بدون استفاده از ابزارهای مناسب دشوار است. بارزترین حمله‌ای که برای اولین بار با این

¹ Attack Vectors

بزرگ‌تر باشد، درخواست ناسالم است و توسط یک سرویس DNS نامعتبر ارسال شده است.



شکل (۴): معماری روش تشخیص پیشنهادی

می‌گیرد؛ بدین ترتیب که اگر این فاصله از حد آستانه بیشتر باشد، ترافیک DNS، ناسالم اگر این فاصله کم باشد، ترافیک DNS، سالم است. ترافیک سالم به این معنا است که در خواست‌های DNS ارسالی از مبدأ تجهیزات اینترنت اشیا معتبر است و درخواست‌ها بر اساس نیازمندی‌های تجهیزات در مبدأ انجام شده است. درحالی‌که در ترافیک ناسالم، ترافیک مربوط به مبدأ درخواست‌کننده نامعتبر است و توسط بات‌نت آلوده شده است.

۳-۲- محاسبه حد آستانه

حد آستانه، تشخیص ترافیک سالم از ناسالم بر مبنای محاسبه اختلاف زمان درخواست و پاسخ سرویس DNS است.

برای محاسبه حد آستانه که با پارامتر σ نشان داده می‌شود؛ به‌ازای هر درخواست DNS دامنه‌های سالم و معتبر، اختلاف زمان بین ارسال و دریافت آن‌ها از رابطه ۱ محاسبه می‌شود. سپس میانگین زمان این درخواست‌ها به‌ازای تعداد انبوهی از آن‌ها محاسبه و به‌عنوان حد آستانه σ تعیین می‌شود.

$$\sigma = \sum_{n=1}^N N \frac{R_n - r_n}{n} \quad (1)$$

که در آن، R زمان پاسخ و r زمان درخواست هر دامنه DNS، N ، تعداد دفعات تکرار محاسبه درخواست دامنه‌ها و n ، تعداد نمونه‌ها است.

با محاسبه حد آستانه، و آموزش ترافیک DNS دامنه‌های سالم، و مقایسه با دامنه‌های ترافیک DNS، نرخ هشدار نادرست^۱ و نرخ تشخیص^۱ به دست می‌آید.

۳-۱-۱- جمع آوری ترافیک

وظیفه این مرحله، جمع‌آوری ترافیک درگاه‌های خروجی تجهیزات اینترنت اشیا در شبکه مبتنی بر آی پی است. نحوه کار به این شکل است که کلیه بسته‌های ارسالی و دریافتی از این تجهیزات، در بازه‌های زمانی مختلف، جمع‌آوری و جهت پردازش مرحله بعدی، ذخیره می‌شوند.

۳-۱-۲- پردازش ترافیک

این مرحله، وظیفه جداسازی ترافیک DNS از کل ترافیک جمع‌آوری شده و پردازش و محاسبه زمان‌های بین درخواست‌های DNS را برعهده دارد. نحوه کار به این صورت است که ترافیک DNS بر اساس پروتکل UDP و درگاه ۵۳ فیلتر شده و این ترافیک از دیگر پروتکل‌های موجود ردوبدل شده جداسازی می‌شود.

پس از این جداسازی، با توجه به ساختار ترافیک DNS، تأخیر زمانی بین درخواست و پاسخ‌های متناظر با هم و مربوطه به یک مبدأ و مقصد، محاسبه می‌شود. برای این منظور، اختلاف بین زمان پاسخ و زمان درخواست محاسبه می‌شود. این محاسبه به تفکیک هر درخواست و پاسخ DNS ذخیره و برای مرحله بعدی آماده‌سازی می‌گردد.

۳-۱-۳- تشخیص

همان‌گونه که در مرحله قبلی اشاره شد، اختلاف زمان درخواست و پاسخ مربوط به هر یک از درخواست‌ها محاسبه می‌شود. در خصوص اینکه ترافیک DNS سالم یا ناسالم است، معیاری به نام حد آستانه تعریف می‌شود. بر اساس این معیار، میزان فاصله از حد آستانه، برای تشخیص ترافیک سالم از ناسالم ملاک قرار

¹ False Alarm Rate

الگوریتم (۱): شبه کد محاسبه حد آستانه σ را نشان می‌دهد.

الگوریتم ۱: محاسبه حد آستانه σ

Input:

Benign DNS files

Output:

σ : threshold

```

1.   $DNS[NumberOfDNSAdd] \leftarrow GetBenignDNSFile();$ 
2.   $Count \leftarrow GetCount();$ 
3.   $totalDNSAddress \leftarrow 0$ 
4.   $repeat \leftarrow 0$ 
5.   $sumDiff \leftarrow 0$ 
6.  while ( $repeat \neq Count$ ) do
7.    while ( $totalDNSAddress \neq NumOfDNSAdd$ ) do
8.       $Packet \leftarrow SendGetDNSPacket(DNS[totalDNSAddress]);$ 
9.       $reqTime \leftarrow getRequestTime(Packet);$ 
10.      $resTime \leftarrow getResopnseTime(Packet);$ 
11.      $difTime \leftarrow (resTimes - reqTimes);$ 
12.      $sumDiff \leftarrow (sumDiff + difTime);$ 
13.      $totalDNSAddress \leftarrow totalDNSAddress + 1;$ 
14.   end
15.    $repeat \leftarrow repeat + 1;$ 
16. end
17.  $\sigma \leftarrow (sumDiff / (NumOfDNSAdd \times Count));$ 

```

الگوریتم (۲): شبه کد روش تشخیص پیشنهادی را نشان می‌دهد.

الگوریتم ۲: روش تشخیص پیشنهادی

Input: σ

output: Benign DNS detection

```

1.   $thershold \leftarrow thershold$ 
2.   $packet \leftarrow packet$ 
3.  While ( $GetDNSPacket( ) \neq 0$ ) do
4.     $packet = getDNSPacket( )$ ;
5.     $reqTime \leftarrow getRequestTime(packet)$ 
6.     $resTime \leftarrow getresponceTime(packet);$ 
7.     $difTime \leftarrow (resTime - reqTime);$ 
8.    If  $difTime < \sigma$  then
9.      return DNS Traffic Benign
10.   else
11.   return DNS Traffic Malicious
12.   end
13. end

```

الگوریتم (۲)، ابتدا حد آستانه و بسته‌های ترافیک DNS را به‌عنوان ورودی می‌گیرد و سپس DNS‌های آلوده را از نوع بی‌خطر آن، تشخیص می‌دهد. ابتدا مجموعه‌ای از ورودی‌های DNS (خط ۱ تا ۲)، سپس به ترتیب زمان ارسال درخواست DNS، زمان پاسخ DNS و اختلاف زمان ارسال و پاسخ به ازای هر درخواست ذخیره می‌گردد (خط ۳ تا ۷)، تا اتمام دریافت تمامی بسته‌های DNS، اختلاف بین زمان درخواست و زمان پاسخ DNS محاسبه شده و با حد آستانه مقایسه می‌شود (خط ۸)، در صورتی که مقدار آن کمتر از حد آستانه باشد، درخواست DNS سالم بوده در غیر این‌صورت درخواست DNS بات تشخیص داده می‌شود (خط ۹ تا ۱۳).

۴- آزمایش‌ها

در این بخش، مبتنی بر روش پیشنهادی و به‌منظور تشخیص آلودگی ترافیک DNS، آزمایش‌هایی بر روی داده‌های واقعی انجام می‌شود. این آزمایش‌ها بر اساس سنجه‌های ارزیابی که در ادامه معرفی می‌شوند، مورد ارزیابی قرار می‌گیرند.

۴-۱- مجموعه داده^۲

مجموعه داده مورد استفاده در روش پیشنهادی، باید بتواند داده لازم را در خصوص پرسش‌های DNS مدیریت کرده و پاسخ دهد.

الگوریتم (۱) مجموعه‌ای از دامنه‌های سالم DNS را به‌عنوان ورودی می‌گیرد و حد آستانه تشخیص DNS‌های سالم از ناسالم را محاسبه می‌کند. ابتدا مجموعه‌ای از نام دامنه‌های DNS دریافت می‌شود (خط ۱). از خط (۲) تا خط (۵)، متغیرهای لازم برای محاسبه تمام آدرس‌های DNS، تکرار لیست و جمع اختلاف زمان‌ها تعریف می‌گردد. تعداد مراحل تکرار و تعداد DNS وارد شده به ترتیب نام DNS براساس لیست وارد شده و زمان ارسال درخواست DNS، زمان پاسخ DNS و اختلاف زمان ارسال و پاسخ به ازای هر درخواست ذخیره می‌گردد، و این اقدام برای تمام DNS‌ها یک مرحله انجام می‌گردد و پس از انجام یک مرحله به تعداد DNS تمام این فرآیند به تعداد مقدار وارد شده برای تکرار انجام می‌گردد (خط ۷ تا ۱۶)، در نهایت برای محاسبه حد آستانه، میانگین اختلاف زمان DNS‌ها به عنوان حد آستانه مشخص می‌گردد. (خط ۱۷).

۳-۳- الگوریتم روش تشخیص پیشنهادی

روش تشخیص پیشنهادی در الگوریتم (۲) تشریح می‌گردد.

² Data Sets

¹ Detection Rate

تولید شده است. از طرفی عملکرد بات‌نت اینترنت اشیا بر اساس حملات توزیع شده است، به عبارت دیگر حمله از چندین نقطه به یک هدف انجام می‌گیرد. به همین دلیل، مجموعه آزمایش‌ها بر مبنای عملکرد بات‌نت میرای در نظر گرفته می‌شود و نظر به محدودیت در اجرای آن در روش پیاده‌سازی، برای تولید داده‌های ناسالم و ارزیابی در برنامه تولیدی از روش چند نخه سیستم‌عامل، برای پردازش موازی و توزیع شده استفاده می‌گردد.

۴-۳- سکوی آزمایش

برای انجام آزمایش‌ها، بعد از پیاده‌سازی بات‌نت میرای، در ساختار سرویس DNS، ترافیک مربوط به آن به دودسته تقسیم می‌شود، یک دسته درخواست و دسته دیگر، پاسخ‌های دریافتی از سرور است. به عبارت دیگر، به‌ازای هر درخواست، متناظر با آن یک پاسخ از طرف مقصد برای مبدأ ارسال می‌گردد. برای ارزیابی دقیق‌تر، در تولید مجموعه داده، زمان ارسال و دریافت‌ها به‌دقت مورد سنجش قرار می‌گیرد و بر اساس میکروثانیه در نظر گرفته می‌شود.

۴-۴- پیاده‌سازی محیط آزمایشی

برای تولید مجموعه داده و پیاده‌سازی محیط آزمایشگاهی، برنامه‌های با زبان C++ طراحی و تولید شد. این ابزار محیط واقعی اجرای بات میرای را در یک شبکه بات تا شبیه‌سازی می‌کند.

برنامه شبیه‌ساز تولیدی دارای دو بخش کلی است. در بخش اول، داده‌های ناسالم به همراه بردار حمله مورد نظر و نتایج آن‌ها تولید می‌شود، در این بخش نحوه تولید مجموعه داده ناسالم و شبیه‌سازی بات‌نت میرای از طریق بردار حمله DNS انجام می‌گردد و در بخش دوم، داده‌های سالم و نتایج آن‌ها به تفکیک ارائه می‌شود. در این بخش، تولید مجموعه داده سالم برای شبیه‌سازی رفتار و عملکرد یک یا تعداد زیادی درخواست و پاسخ DNS انجام می‌شود.

۴-۵- تنظیمات ابزار

در این بخش، تنظیمات موردنیاز هر بخش که برای شبیه‌سازی محیط آزمایشگاهی با محیط واقعی انجام می‌شود به همراه حالت‌های مختلف اجرای نرم‌افزار، بررسی و ارائه می‌شود.

در بخش اول، شبیه‌سازی بات‌نت میرای، بردار حمله DNS مطابق شکل (۵) با توجه به درخواست‌های DNS، به طور مدام بر روی یک دامنه مورد نظر با تولید زیردامنه‌های تصادفی بسیار زیاد انجام می‌گیرد و همچنین با توجه به اینکه این حمله به صورت توزیع شده و از چندین محل مختلف انجام می‌گردد، تنظیمات چند نخه سیستم‌عامل در این بخش برای شبیه‌سازی

از ویژگی‌های مورد نیاز در این مجموعه داده نیز، میزان اعتبار پرسش‌ها و درخواست‌های DNS است. از آنجاکه مجموعه داده با ویژگی‌های مورد نظر از چالش‌های این نوع تحقیق‌ها است. برای تهیه دودسته از مجموعه داده‌های سالم و ناسالم، از دامنه‌های سالم و آلوده استفاده می‌شود. در ادامه نحوه تولید این دسته از داده‌ها ارائه می‌شوند.

۴-۱-۱- مجموعه داده سالم^۱

برای تولید داده‌های سالم، ابتدا عملکرد بات میرای مورد بررسی قرار گرفت. باتوجه به اینکه بردار حمله DNS این بات، عموماً بر روی زیر دامنه‌ها^۲ است، فهرستی از دامنه‌های معروف، به همراه زیر دامنه‌های آن‌ها از وبگاه گیت هاپ^۳ انتخاب شد. برای اطمینان از کارکرد صحیح هر کدام از این دامنه‌ها، وضعیت آن‌ها مورد ارزیابی قرار گرفت و از نظر اعتبار و قابل‌دسترس بودن، به طور مجزا بررسی شد. در نهایت، فهرستی از زیر دامنه‌های سالم برای استفاده در فرایند ارزیابی و آزمایش‌ها روش پیشنهادی، انتخاب و تعیین شد.

۴-۱-۲- مجموعه داده ناسالم^۴

روش تولید مجموعه داده ناسالم نیز مانند داده سالم بر اساس عملکرد بات میرای است که در بردار حمله DNS بررسی شده است. همانند داده‌های سالم در تولید این نوع داده نیز بر اساس کد ارائه شده این بات در وبگاه گیت هاپ عمل شده است. زیر دامنه‌های جعلی از ترکیب حرف الفبا و اعداد با طول متغیر و به شکل تصادفی تولید می‌شوند، به سمت سرور DNS ارسال می‌شوند و منتظر پاسخ آن‌ها می‌مانند. از آنجایی که چنین دامنه‌هایی وجود ندارند، حمله توزیع شده ممانعت از سرویس به زیر ساخت DNS شکل می‌گیرد. از همین منطق برای تولید این زیر دامنه‌های ناسالم استفاده شده است.

روش چند نخه^۵ سیستم‌عامل که برای توسعه کد برنامه استفاده شده است، به نحو مطلوبی حملات بات میرای را که باعث ممانعت از سرویس توزیع شده می‌شود، شبیه‌سازی می‌کند.

۴-۲- پیاده‌سازی بات‌نت

برای انجام آزمایش‌ها، بات‌نت میرای انتخاب می‌شود. با بررسی‌های انجام شده، مشخص شد که حملات زیادی در اینترنت اشیا از طریق بات‌نت میرای انجام می‌گردد. همچنین خانواده مختلفی از این بات‌نت پس از انتشار، به روش‌های مختلف

¹ Benign Datasets

² Subdomain

³ GitHub

⁴ Malicious Datasets

⁵ Multi-Threading

برنامه، مجموعه داده ناسالم به همراه گزارش اجرای آن در محل اجرایی برنامه در پوشه Tab1 با قالب متنی قابل دسترس است.

در بخش دوم، همان گونه که در خصوص مجموعه داده سالم بیان شد، ابتدا فهرستی از زیر دامنه های سالم، تهیه و در قالب یک فایل متنی در برنامه وارد می شود. سپس تعداد مراحل اجرای فرایند درخواست DNS فایل مورد نظر، در قسمت Number of Loop به تعداد دلخواه وارد می شود. نتایج برنامه در محل فایل اجرایی برنامه پوشه Tab2 ایجاد و در قالب متنی به همراه گزارش برنامه قابل دسترس است.

۵- نتایج آزمایش ها و ارزیابی

در این بخش، نتایج حاصل عملکرد روش تشخیص پیشنهادی برای تشخیص میزبان های آلوده و درخواست های ناسالم DNS مورد بررسی و ارزیابی قرار می گیرد.

۵-۱- آزمایش ها

تعداد ۱۰ آزمایش به این نحو انجام شد که برای هر کدام از آنها، ابتدا داده سالم و ناسالم تولید می شود، سپس با استفاده از محاسبه زمان ارسال و دریافت بسته های DNS، و مقایسه نتایج با حد آستانه تشخیص محاسبه شده، دامنه های ناسالم تشخیص داده می شوند. این فرآیند در ادامه، نشان داده می شود.

۵-۱-۱- تولید داده های ناسالم

در این مرحله طبق توضیحات ارائه شده در خصوص نحوه تولید مجموعه داده های ناسالم، طبق جدول (۱) تعداد ۶۰۰۰ داده ناسالم تولید و برچسب گذاری می گردد.

جدول (۱): تولید مجموعه داده ناسالم

تعداد بسته (در ثانیه)	نام دامنه	طول رشته دامنه
۱۰۰۰	Ali.com	۱۰

۵-۱-۲- تولید داده های سالم

تعداد ۳۰۰ داده های سالم بر اساس روش ارائه شده مطابق جدول (۲) تولید و برچسب گذاری می گردد.

جدول (۲): تولید مجموعه داده سالم

تعداد نام دامنه	تعداد تکرار فرایند
۱۰۰	۳

۵-۱-۳- محاسبه حد آستانه

برای درخواست های موازی در نظر گرفته شده است. شکل (۶) نمایی از ابزار و تنظیمات آن را نشان می دهد.

Tab 1	Tab 2
Number of threads:	7
packets per Thread:	10000
Domain:	ali.com
Random string lenght:	7

شکل (۵): تنظیمات شبیه ساز حمله بات‌نت میرای

Number Of Threads: این گزینه برای تنظیمات چند نخه

سیستم عامل در نظر گرفته شده و باتوجه به قدرت پردازنده،

رایانه از عدد یک تا n در نظر گرفته می شود.

Packets Per Thread: در قسمت تعداد بسته ارسالی در ثانیه

برای تکمیل شبیه سازی حمله با درخواست های فراوان انجام

می گردد.

Domain: در این قسمت نام دامنه مورد نظر مشخص می شود.

Random String Length: در این قسمت تعداد کاراکترهای

مربوط به تولید تصادفی زیر دامنه صورت می گیرد که از

ترکیب حروف الفبا و اعداد از یک تا عدد مورد نظر فاکتوریل

است.

در شکل (۶) وضعیت اجرای برنامه در لحظه نشان داده شده است. در ادامه هر کدام از این وضعیت ها تشریح می شوند.

Thread	Request	Response	Success	Failed
0	10000	10000	10000	0
1	10000	10000	10000	0
2	10000	10000	10000	0
3	10000	10000	10000	0
4	10000	10000	10000	0
5	10000	10000	10000	0
6	10000	10000	10000	0

شکل (۶): وضعیت اجرای حمله بات‌نت میرای

Therad: وضعیت هر یک از نخ های سیستم عامل

Request: تعداد درخواست های ارسالی

Response: تعداد پاسخ های دریافتی

Success: تعداد تکمیل شدگان فرایند درخواست و پاسخ

Failed: تعداد تکمیل نشدگان فرایند درخواست و پاسخ

در صورتی که فرایند اجرای این قسمت کامل انجام گردد، تعداد

اعداد هر یک از فیلدها و ستون های Request, Response, Success

یک عدد ثابت نشان داده می شود. در پایان خروجی

داده می‌شود. بر اساس الگوریتم تشخیص روش پیشنهادی، دامنه‌های آلوده تشخیص داده می‌شود و ضمن برجسب‌گذاری، نتایج آن در قالب فایل متن خروجی تولید می‌گردد. جدول (۳) نمونه‌ای از تشخیص داده‌های سالم از ناسالم را مطابق حد آستانه محاسبه شده، نشان می‌دهد. در این جدول، Mal، نشان‌دهنده فایل آلوده و Ben، فایل سالم است.

حد آستانه مطابق الگوریتم ارائه شده در بخش ۵-۲، از مجموعه داده‌های سالم محاسبه می‌گردد.

۵-۱-۴- تشخیص دامنه‌های سالم و ناسالم

در این بخش داده‌های سالم و ناسالم تولید شده با هم ترکیب می‌شود و در قالب یک فایل متنی، مجموعه‌ای از داده‌ها به همراه مقدار حد آستانه محاسبه شده به‌عنوان ورودی به برنامه تشخیص

جدول (۳): تشخیص دامنه‌های سالم از ناسالم

نام دامنه	اختلاف زمان دریافت و ارسال	محاسبه اختلاف زمان با مقدار حد آستانه	مقایسه با حد آستانه (۳۲۴۷۵۴۱۷۳)	وضعیت واقعی	نتیجه ارزیابی
OjJS.ali.com	6419598900	6094844727	بزرگ‌تر	Mal	Mal
u9D.ali.com	6466031600	6141277427	بزرگ‌تر	Mal	Mal
NA.ali.com	832500	-323921673	کوچک‌تر	Mal	Ben
autodiscover.whatsapp.com	508906600	184152427	بزرگ‌تر	Ben	Mal
ayuda.linkedin.com	164635100	-160119073	کوچک‌تر	Ben	Ben
azure.microsoft.com	225975200	-98778973	کوچک‌تر	Ben	Ben
beta.yahoo.com	456774400	132020227	بزرگ‌تر	Ben	Mal

براساس آن جدول (۵) نیز شاخص دقت، نرخ هشدار صحیح و نرخ هشدار نادرست را محاسبه می‌کند.

۵-۲-۵- ارزیابی

برای بررسی و ارزیابی روش پیشنهادی، داده‌های سالم و ناسالم برجسب‌گذاری شده و بر اساس ماتریس درهم‌ریختگی^۱ [۲۲] مورد ارزیابی قرار می‌گیرد.

۵-۲-۱- معیارهای ارزیابی

ارزیابی روش پیشنهادی با استفاده از معیار دقت^۲ [۲۳]، نرخ تشخیص صحیح^۳ و نرخ هشدار نادرست^۴ مقایسه می‌شوند [۲۴].

۵-۲-۲- محاسبه معیارهای سنجش

به‌منظور ارزیابی و سنجش روش تشخیص بر اساس مقادیر به‌دست‌آمده از ماتریس درهم‌ریختگی، شاخص دقت، نرخ هشدار صحیح و نرخ هشدار نادرست و ناحیه زیر منحنی محاسبه می‌شود.

جدول (۴) مقادیر مثبت صحیح، منفی صحیح، مثبت غلط و منفی غلط محاسبه شده در آزمایش شماره ۱ را نشان می‌دهد. و

جدول (۴): نتایج محاسبه عناصر ماتریس درهم‌ریختگی آزمایش شماره ۱

TP	FP	FN	TN	آزمایش
۵۹۷۸	۱۰۴	۲۱	۱۹۶	۱

دقت	جدول (۵): محاسبه شاخص‌های سنجش آزمایش شماره ۱
۰٫۹۸۰۲	
نرخ هشدار نادرست	۰٫۳۴۶۷
نرخ تشخیص درست	۰٫۹۹۶۵

به منظور ارزیابی و سنجش روش پیشنهادی، آزمایش شماره ۱ به تعداد ۱۰ بار تکرار می‌شود و نتایج تمام ۱۰ آزمایش محاسبه معیارهای تشخیص و میانگین آن‌ها در جدول (۶). نتایج آزمایش‌ها نشان داده می‌شود.

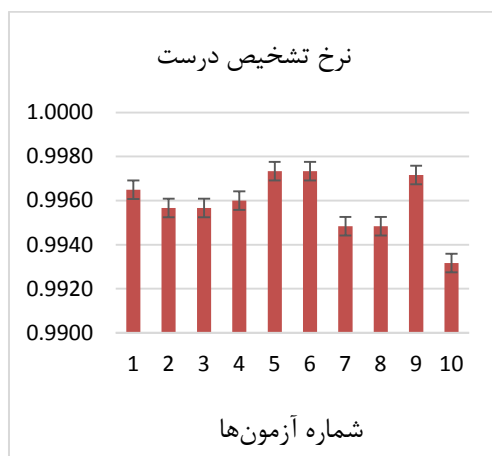
همچنین با استفاده از مفهوم انحراف معیار، میزان قابل اتکا بودن هر یک از ویژگی‌ها نشان داده می‌شوند. در این مفهوم، هر چه انحراف معیار کم‌تر باشد آن ویژگی در محیط‌های مختلف، رفتار باثبات‌تری از خود نشان می‌دهد. برای همین منظور محاسبه انحراف معیار شاخص‌ها در شکل (۷): شاخص دقت، نرخ تشخیص

¹ Confusion matrix

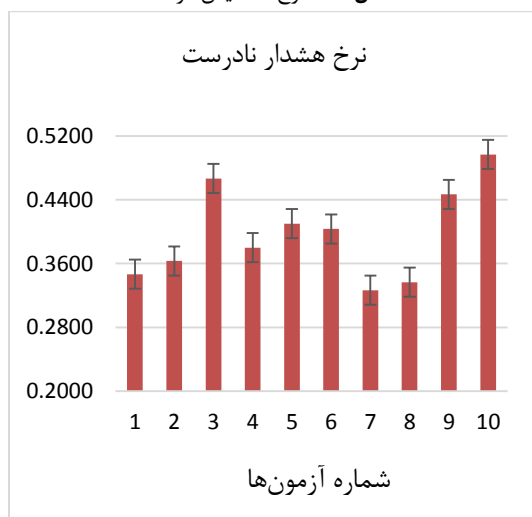
² Accuracy (ACC)

³ True Positive Rate (TPR)

⁴ False Alarm Rate (FAR)



شکل (۸): نرخ تشخیص درست



شکل (۹): نرخ هشدار نادرست

۶- مقایسه روش‌های تشخیص

برای تکمیل ارزیابی، روش‌های تشخیص مختلف بررسی و در جدول (۶) نشان داده شده است. معیارهایی که برای این مقایسه انتخاب شده است، رویکرد تشخیص، دقت و وابستگی به دامنه‌های شناخته شده است. معیار وابستگی به دامنه‌های شناخته شده که در روش میراکا وجود دارد از آن جهت دارای ارزش بالایی است که تشخیص را منحصر به دامنه‌هایی نمی‌کند که قبلاً تشخیص داده شده و مخرب بوده‌اند. بلکه مبتنی بر رفتار آن دامنه است.

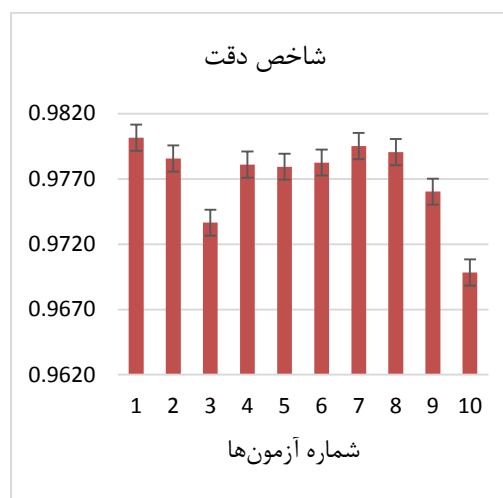
جدول (۶) مقایسه روش‌های تشخیص بات‌نت

ردیف	روش تشخیص	رویکرد	دقت	عدم وابستگی به دامنه‌های شناخته‌شده
۱	ابراهیم غفیری و همکارش [9]	فهرست سیاه	--	x
۲	خوزه سلوی و همکارانش [10]	یادگیری ماشین	۰,۹۸۹۱	x
۳	جی هیون لی و	درخت	۰,۹۱۲۶	x

درست شکل (۸) و نرخ هشدار نادرست در شکل (۹) نشان داده شده برای اثبات روش تشخیص پیشنهادی است.

جدول (۶): نتایج آزمایش‌ها

آزمایش‌ها	TP	FP	ACC	FAR(FPR)	PR
شماره ۱	۵۹۷۸	۱۰۴	۰,۹۸۰۲	۰,۳۴۶۷	۰,۹۹۶۵
شماره ۲	۵۹۷۳	۱۰۹	۰,۹۷۸۶	۰,۳۶۳۳	۰,۹۹۵۷
شماره ۳	۵۹۷۳	۱۴۰	۰,۹۷۳۶	۰,۴۶۶۷	۰,۹۹۵۷
شماره ۴	۵۹۷۵	۱۱۴	۰,۹۷۸۱	۰,۳۸۰۰	۰,۹۹۶۰
شماره ۵	۵۹۸۳	۱۲۳	۰,۹۷۷۹	۰,۴۱۰۰	۰,۹۹۷۳
شماره ۶	۵۹۸۴	۱۲۱	۰,۹۷۸۳	۰,۴۰۳۳	۰,۹۹۷۳
شماره ۷	۵۹۶۹	۹۸	۰,۹۷۹۵	۰,۳۲۶۷	۰,۹۹۴۸
شماره ۸	۵۹۶۹	۱۰۱	۰,۹۷۹۰	۰,۳۳۶۷	۰,۹۹۴۸
شماره ۹	۵۹۸۲	۱۳۴	۰,۹۷۶۰	۰,۴۴۶۷	۰,۹۹۷۲
شماره ۱۰	۵۹۵۸	۱۴۹	۰,۹۶۹۸	۰,۴۹۶۷	۰,۹۹۳۲
میانگین نتایج			۰,۹۷۷۱	۰,۳۹۷۷	۰,۹۹۵۸



شکل (۷): شاخص دقت

لیست سیاه و تحلیل نام دامنه نیازمند به‌روزرسانی دائم هستند، درحالی‌که میراکا، حد قابل‌قبولی از یک ویژگی ذاتی و مستقل ارائه می‌کند.

۸- مراجع

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols and Applications," IEEE Communications Surveys & Tutorials, vol. 17, pp. 2347-2376, 2015.
- [2] M. A. R. Jalaei, "Detecting Botnets with Timing-Based Covert Command and Control," Journal of Electronical & Cyber Defence, vol. 7, no. 4, pp. 1-15, 2020.
- [3] G. Doyen, M. Charalambides, S. Latré, B. Stiller, "Towards Incentivizing ISPs to Mitigate Botnets," in International Conference on Autonomous Infrastructure, Brno, 2014.
- [4] M. Litoussi, N. Kannouf, K. El Makkaoui, A. Ezzatia, M. Fartitchou, "IoT security: challenges and countermeasures," Published by Elsevier B.V., vol. 177, p. 503-508, 2020.
- [5] I. Elzen, J. Heugten, "Techniques for detecting compromised," University of Amsterdam, Amsterdam, February 12, 2017.
- [6] T. Hyun Kim, D. Reeves, "A survey of domain name system vulnerabilities and attacks," Journal of Surveillance, Security and Safety, pp. 34-60, 2020.
- [7] L. Fang, H. Wu, K. Qian, W. Wang, L. Han, "A Comprehensive Analysis of DDoS attacks based on DNS," in International Conference on Computer Vision and Data Mining (ICVDM 2021), 2021.
- [8] K. Alieyan, A. Almomani, A. Manasrah, M. Kadhum, "A survey of botnet detection based on DNS," Springer, 2015.
- [9] I. Ghafir, V. Prenosil, "DNS Traffic Analysis for Malicious Domains Detection," in 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 2015.
- [10] J. Selvi, J. Ricardo, b. Rodríguez, E. Soria-Olivas, "Detection of algorithmically generated malicious domain names using masked N-grams," Expert Systems With Applications, p. 156-163, 2019.
- [11] J. Lee, H. Lee, "Graph-based Malware Activity Detection by DNS traffic analysis," Computer Communications, p. 33-47, 2014.
- [12] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, C. Kruegel, "A passive DNS analysis service to detect and report malicious," Article in ACM Transactions on Information and System Security, 2014.
- [13] A. Prokofiev, Y. Smirnova, V. Surov, "A Method to Detect Internet of Things Botnets," National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), 2018.
- [14] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, "N-BalIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," IEEE Pervasive Computing, pp. 12-22, July–September 2018.
- [15] Radware, "A Quick History of IoT Botnets," Radware, 1 March 2018. [Online]. Available: <https://blog.radware.com/uncategorized/2018/03/history-of-iot-botnets/>.
- [16] S. Daniel, "IoT Botnets on the Rise," Radware, 2 October 2018. [Online]. Available: <https://blog.radware.com/security/2018/10/iot-botnets-on-the-rise/>.
- [17] P. Danny, "This aggressive IoT malware is forcing Wi-Fi routers to join its botnet army," ZDNET, 31 October 2019. [Online]. Available: <https://www.zdnet.com/article/this-aggressive-iot-malware-is-forcing-wi-fi-routers-to-join-its-botnet-army/>.
- [18] R. Winward, "IoT Attack Handbook," Radware, U.S., 2018.
- [19] C. Koliass, G. Kambourakis, A. Stavrou, J. Voas, "Mirai and Other Botnets," IEEE Computer Security, 2017 IEEE.

ردیف	همکارش [11]	تصمیم	میراکا
۴	دیوید بالزاروتی و همکارانش [12]	الگوریتم ژنتیک	۰,۹۸۰۴
۵	پروکوفیف و همکارانش [13]	روش‌های آماری	۰,۹۷۳۰
۶	میدان و همکارانش [14]	شبکه عصبی	---
۷	میراکا (روش پیشنهادی)	دسته‌بندی تک کلاسه	۰,۹۷۷۱

همان‌گونه که در جدول (۷) مشاهده می‌شود، روش‌های تشخیصی صرفاً دامنه‌های شناخته شده را تشخیص می‌دهند. درحالی‌که میراکا، وابستگی به دامنه‌های شناخته شده ندارد. از نظر دقت نیز میراکا دقت بالایی دارد و صرفاً روش [۱۵] دقت بالاتری دارد ولی به دلیل استفاده از الگوریتم تکاملی ژنتیک سربار پردازشی بالایی دارد.

۷- نتیجه گیری

در این مقاله ابتدا حملات DNS بررسی و پژوهش‌های تشخیص بات‌نت‌های مبتنی بر DNS ارائه شد. روش‌های ارائه شده به‌طور کلی از روش‌های لیست سیاه، تحلیل محتوایی نام DNS و تحلیل ویژگی‌های آن استفاده می‌کنند.

همچنین باتوجه به بررسی و مطالعه انواع بات‌نت‌های اینترنت اشیا مشخص شد بات‌نت میرای باتی است با انواع بردارهای حمله که کد برنامه آن به‌صورت عمومی انتشار یافته است. این امر باعث شده است که انواع گوناگونی از بات‌ها با عملکردی شبیه به آن تولید و انتشار یافته و به‌عنوان پایه و اساس بات‌نت‌های دیگر به کار برده شود. همچنین دیگر بات‌نت‌ها باتوجه به اهداف اقتصادی و امنیتی، هدف از تولید، کد و تشریح عملکرد خود را انتشار نمی‌دهند و صرفاً طبق گزارشاتی که از طریق مراکز امنیتی به دست می‌آید، مورد بررسی و ارزیابی قرار می‌گیرند. برای همین منظور باتوجه به نقش محوری بات‌نت میرای برای ارزیابی روش پیشنهادی، این بات‌نت با بردار حمله DNS مورد بررسی قرار گرفته است.

در روش پیشنهادی میراکا، بر اساس عملکرد DNS که شامل دو بخش ارسال و پاسخ است؛ از طریق محاسبه اختلاف زمان بین آن‌ها باتوجه به مقدار به‌دست‌آمده از حد آستانه، سالم و یا ناسالم بودن درخواست DNS تشخیص داده می‌شود.

باتوجه به روش میراکا و نتایج به‌دست‌آمده از ارزیابی‌ها مشخص شد زمان پاسخ‌گویی به درخواست‌های سالم کمتر از درخواست‌های ناسالم است؛ لذا میراکا با بهره‌گیری از این ایده، بار پردازشی کمتری نسبت به روش‌های لیست سیاه، تحلیل نامه و ویژگی درخواست‌ها را موجب می‌شود. همچنین روش‌های

- [20] Z.Zorz, "Linux/IRCTelnet creates new, powerful IoT DDoS botnet," Help Net Security, 2 November 2016. [Online]. Available: <https://www.helpnetsecurity.com/2016/11/02/linuxircnetelnet-iot-ddos-botnet/>.
- [21] S.Edwards, I.Profetis, "Hajime – Friend or Foe?," ERT Threat Advisory -Radware, USA, April 26, 2017.
- [22] F.Bagheri,M.Rezvani,M.Fateh,E.Tahanian, "Malicious Domain Detection Using DNS Records," Journal of Electronical & Cyber Defence, vol. 9, pp. 83-97, 2021. (in persian).<https://dor.isc.ac/dor/20.1001.1.23224347.1400.9.3.7.8>
- [23] A.Jafar Gholi Beik,M.E.Shiri Ahmad Abadi,A.Rezakhani, "The Presentation of a Hybrid Anomaly Detection Model Using Inverse Weight Clustering and Machine Learning in Cloud Environments," Journal of Electronical & Cyber Defence, vol. 9, pp. 21-29, 2022. (in persian).<https://dor.isc.ac/dor/20.1001.1.23224347.1400.9.4.2.5>
- [24] B. Wójcicki, R.Dąbrowski, "Applying Machine Learning to Software Fault Prediction," e-Informatica Software Engineering Journal, vol. vol. 12, no. 1, p. 199–216, 2018.
- [25] M.Singh,M.Singh,S.Kaur, "Detecting bot-infected machines using DNS fingerprinting," Digital Investigation, pp. 14-33, 24 December 2018.
- [26] M.Aldwairi,F.Belqasmi, "Malware Detection using DNS Records and Domain Name Features," in International Conference on Future Networks and Distributed Systems, June 26-27,2018, Amman, Jordan. ACM, NewYork, NY, USA, June 2018.



Detection of DNS based botnets using traffic analysis in IoT

M.T. Behnam¹, R. Jalaei^{2*}

^{2*} Associate Professor, Imam Hossein University, Tehran, Iran

(Received: 2024 /05/20, Revised: 2024/09/12, Accepted: 2024/10/07, Published: 2024/10/22)

DOR: <https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.3.3.5>

Abstract

Application of IOT based equipment using different technologies are increasing on a day-to-day basis. On the other hand, due to light protocols, versatility of applications, and geographic prevalence, and management by non-specialists, a thorough, secure configuration, and proper update of these equipment is not handled properly. Thus, such equipment are easy targets for various hackers' attacks.

Moreover, crafting bot networks for destructive activities is available more than ever. Using IOT equipment with such weaknesses and shortcomings, detecting bot networks remains as a serious challenge.

In this work, after surveying relevant works, different attack vectors in IOT networks were studied. Then DNS attack vectors that may be leveraged by bots were identified. Finally, Miraka, a method for detecting DNS attacks in IP layer of IOT networks was proposed. For practical purposes, Mirai, a notorious IP based bot common to IOT as well as TCP/IP networks was studied, and different results were obtained using comprehensive traffic generation patterns and scenarios.

The advantage of the proposed method relies on faster detection of contaminated traffic due to lexical domain name analysis, and less reliance on domain name attributes. Empirical results show a success rate of %99.5 and precision of %99.7 in Mirai based bot detection in IP based IOT networks which is superior to other competing methods.

Keywords: : Botnet, Internet of Things, DNS Attacks, Mirai.

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Publisher: Imam Hussein University

Authors



*Corresponding Author Email: rjalaei@ihu.ac.ir