نشریه علمی "پدافند الکترونیکی و سایبری"
سال هفتم، شماره ۴، زمستان ۱۳۹۸، ص ۱۵-۱

# تشخیص بات‌نت‌ها با کانال‌های فرمان و کنترل پنهان زمانی

رضا جلایی[۱]، محمدرضا حسنی‌آهنگر[۲*]
۱- استادیار، ۲- دانشیار، دانشگاه جامع امام حسین(ع)
(دریافت: ۹۷/۰۸/۰۱، پذیرش: ۹۷/۱۲/۱۴)

## چکیده

امروزه بات‌نت‌ها به‌عنوان یک ناهنجاری در فرآیند تبادل اطلاعات و آسیب‌رساندن به منابع شبکه تبدیل شده‌اند. روش‌های تشخیص آن‌ها همواره با چالش‌هایی روبرو بوده است و به‌عنوان یک موضوع تحقیق مورد بررسی و به‌روز شدن قرار گرفته است. اصلی‌ترین جزء یک بات‌نت، کانال فرمان و کنترل آن است و مدیربات توسط این کانال، فرمان‌های خود را برای اجرا روی سامانه قربانی ارسال می‌کند. در صورت تشخیص کانال فرمان و کنترل یک بات‌نت، عملا ارتباط با مدیر بات برقرار نشده و دستورات مدیربات اجرا نمی‌شوند. به همین دلیل مدیر بات با استفاده از انواع روش‌های فرار سعی می‌کند احتمال کشف کانال را پایین نگه دارد. کانال پنهان فرمان و کنترل مفهومی است که بات‌نت‌های نسل جدید برای مخفی‌سازی ارتباط خود به‌کار می‌برند. در این مقاله یک مدل انتزاعی از بات‌نت پیشنهاد شده است که در آن فرمان‌های مدیر بات، مبتنی بر تاخیر زمانی بین بسته‌های و توالی آن‌ها ارسال می‌شوند. این فرمان‌ها از طریق کانال فرمان و کنترل پنهان زمانی ارسال می‌شوند. در ادامه با استفاده از مفهوم فعالیت گروهی بات‌ها؛ روشی برای تشخیص این بات‌نت پیشنهاد شده است. معماری روش تشخیص، از سه لایه جمع‌آوری و پردازش ترافیک، پردازش الگوها و تشخیص دومرحله‌ای تشکیل شده است. با استفاده از روش تشخیص دو مرحله‌ای که شامل ماتریس شباهت و آنتروپی است، میزبان‌های آلوده به بات تشخیص داده می‌شوند. برای ارزیابی روش، پنج کانال زمانی معتبر شبیه‌سازی شده و هر کدام برای ارسال فرمان‌های مدیربات مورد استفاده قرار می‌گیرند. نتایج آزمایش‌ها، کارایی روش تشخیص با وجود حداقل دو بات در شبکه را نشان می‌دهد.

**کلیدواژه‌ها:** بات‌نت، کانال فرمان و کنترل، کانال پنهان زمانی، ماتریس شباهت، آنتروپی، آنتروپی شرطی اصلاح‌شده

## ۱. مقدمه

بات‌نت مجموعه‌ای از بات‌ها یا مجموعه‌ای از رایانه‌های آسیب‌پذیر است که به‌وسیله مدیر بات کنترل می‌شوند [۱]. بات‌نت‌ها از ماشین‌های آلوده برای اجرای بدافزارهای دیگر و همچنین آسیب‌پذیری‌های نرم‌افزاری استفاده می‌کنند [۲]. این کار از طریق ایجاد زیرساخت سرویس‌دهنده‌های فرمان و کنترل بین ماشین‌های آلوده و مدیربات انجام می‌گیرد. مدیر بات کنترل ماشین‌ها را به‌دست گرفته و اقدام به فعالیت‌های مخرب مثل سرقت اطلاعات، هویت کاربران، حملات منع خدمات[۱]، ارسال پیغام‌های حجیم و دیگر فعالیت‌های غیرقانونی می‌کند [۳]. کانال‌های فرمان و کنترل، مهم‌ترین جزء بات‌نت‌ها هستند و به‌شکل ساختاری به متمرکز، غیرمتمرکز و ترکیبی تقسیم‌بندی می‌شوند [۴]. در ساختار متمرکز، یک نقطه مرکزی از سوی مدیر بات به‌عنوان سرویس‌دهنده برای بات‌های عضو انتخاب شده و از طریق IRC یا HTTP با بات‌ها ارتباط برقرار می‌کنند. در ساختار غیرمتمرکز، مدیر بات از یک پروتکل نظیر به نظیر[۲] برای برقراری ارتباط با بات‌ها استفاده می‌کند و در ساختار ترکیبی، مدیر بات، هر دو ساختار متمرکز و توزیع‌شده را با یکدیگر ترکیب کرده و براساس یک ساختار ترکیبی عمل می‌کند.

در سال‌های اخیر، روش‌های مختلفی برای تشخیص کانال‌های فرمان و کنترل ارایه شده است. همچنین، مدیران بات نیز روش‌های جدیدی را برای فرار از تشخیص و پایداری بات‌های خود ایجاد کرده و یا توسعه داده‌اند. از این‌رو، کانال‌های ارتباطی پنهان جدید به‌طور مداوم درحال ظهور هستند [۵]. برای مثال می‌توان به استفاده از کانال‌های پنهان مبتنی‌بر تاخیر بین بسته‌ها[۳] اشاره کرد. در این کانال، پیام‌های مبادله شده بین مبدا و مقصد توسط یک الگو که نوعی توالی تاخیر بین بسته‌ای است انجام می‌شود. به بیان دیگر فرستنده، اطلاعات زمانی ارسال بسته‌های شبکه را به نحوی دستکاری می‌کند که گیرنده بتواند از طریق مشاهده فواصل زمانی بین بسته‌ها، اطلاعات را کدبرداری کند.

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

[1] DoS
[2] Peer to Peer
[3] Inter Packet Delay

ادامه مقاله به ترتیب زیر نگارش شده است. در بخش دوم مفاهیم پایه بیان می‌شوند. در بخش سوم پژوهش‌های حوزه باتنت و کانال پنهان مورد بررسی قرار می‌گیرند، بخش چهارم به مفهوم باتنت زمانی، اجزا و معماری روش تشخیص پیشنهادی پرداخته است. در بخش پنجم آزمایش‌های انجام شده ارایه شده است. در ادامه و در بخش ششم، نتایج ارزیابی روش پیشنهادی ارایه شده است و در انتها و در بخش هفتم، بحث و نتیجه‌گیری نگارش شده است.

## ۲. مفاهیم پایه

در این بخش مفاهیم پایه مورد استفاده در این مقاله بیان می‌شوند.

### ۲-۱. کانال‌های پنهان زمانی

کانال‌های پنهان به‌عنوان مسیری برای تبادل اطلاعات بین دو نقطه مبدا و مقصد مورد استفاده قرار می‌گیرند. لامپسون، کانال‌های پنهان را در سیستم‌های یکپارچه، ساز وکاری تعریف می‌کند که در آن یک فرآیند با سطح امنیتی بالا، اطلاعات را برای یک فرآیند با سطح امنیتی پایین و عدم اجازه دسترسی به این اطلاعات فاش می‌کند [۶]. در تعریفی دیگر، کانال پنهان هر روش ارتباطی است که برای انتقال غیرمجاز اطلاعات مورداستفاده قرار می‌گیرد [۷]. هدف تمامی آن‌ها رساندن سالم پیام‌های ارسالی به مقصد است به گونه‌ای که از افشای محتوای پیام جلوگیری شود.

کانال‌های پنهان به دو دسته انبارشی و زمانی تقسیم می‌شوند [۵]. مطالعات درخصوص کانال‌های انبارشی بیشتر است و در میان آن‌ها کانال‌های شبکه‌ای فراوانی بیشتری در پیاده‌سازی و استفاده دارند. در این کانال‌ها، رسانه انتقال، محیط شبکه یعنی، مسیریاب‌ها، دیواره‌های آتش و مانند آن است.

تعریف ۱ (کانال پنهان زمانی): کانال پنهان زمانی شامل یک پردازه فرستنده است که اطلاعات را به پردازه دیگری از طریق تنظیم و میزان استفاده از منابع سیستم (به‌عنوان مثال زمان پردازنده) ارسال می‌کند به نحوی که این تغییرات در زمان جواب واقعی مشاهده شده توسط پردازه دیگر قابل مشاهده است [۸].

### ۲-۲. آنتروپی شرطی[1]

آنتروپی، توصیف میانگین ابهام یک متغیر تصادفی است [۹]. میانگین آنتروپی به‌ازای هر متغیر تصادفی نرخ آنتروپی و معیاری برای پیچیدگی یا نظم است [۱۱-۱۰]. بر همین اساس نرخ آنتروپی را می‌توان آنتروپی شرطی دنباله‌ای با طول بی‌نهایت تعریف کرد که فرآیندی مستقل، توزیع‌شده و برابر با آنتروپی درجه اول است. یک فرآیند با نظم بالا آنتروپی کوچکی دارد و در فرآیندی، که الگوهای تکراری دارد صفر است.

### ۲-۳. آنتروپی شرطی اصلاح شده[2]

نرخ آنتروپی شرطی با اندازه‌گیری نمونه‌های متناهی به دست نمی‌آید و مقدار آن تخمین زده می‌شود [۱۲]. از طرفی تخمین آنتروپی یا آنتروپی شرطی مبتنی‌بر تابع چگالی احتمال تجربی[3] است که به ترتیب با $EN$ و $CE$ نشان داده می‌شوند. رابطه (۱) آنتروپی شرطی را نشان می‌دهد.

$$CCE(X_m|X_1,\dots,X_{m-1}) = CE(X_m|X_1,\dots,X_{m-1}) + perc(X_m)\cdot EN(X_1) \quad (۱)$$

که در آن، $perc(X_m)$ درصدی از الگوهای منحصربه‌فرد به طول $m$ و $EN(X_1)$ آنتروپی الگوهای به طول یک است.

تخمین مقدار $CE(X_m|X_1,\dots,X_{m-1})$ به تعداد مقادیر $m$ وابسته است. با توجه به محدودیت داده‌ها، آنتروپی شرطی با افزایش $m$ به سمت صفر میل می‌کند. اگر تنها یک نمونه از یک الگوی خاص به طول $m-1$ در داده‌ها پیدا شود، یک نمونه از بسط این الگو به طول $m$ را نیز می‌توان یافت. بنابراین، الگوی به طول $m$ را می‌توان با دیدن الگوی به طول $m-1$ پیش‌بینی کرد و بنابراین الگوهای به طول $m$ و $m-1$ نادیده گرفته می‌شوند و اگر هیچ الگویی به طول $m$ در داده‌ها پیدا نشود حتی برای فرآیندهای مستقل و یکسان توزیع‌شده، $CE(X_m|X_{m-1})$ صفر است. برای گذر از این محدودیت داده‌ها، بدون توجه به طول $m$، از آنتروپی شرطی اصلاح‌شده استفاده می‌شود.

### ۲-۴. روش توزیع سطل‌ها

برای تعیین چگونگی توزیع داده‌ها و نیز سطح دانه‌بندی آن‌ها روش توزیع سطل تعریف می‌شود. در این روش، تعداد سطل‌ها و توزیع داده‌ها از طریق سطح‌های هم احتمال تعیین می‌شوند. پورتا[4] [۱۰] نشان داد که مساحت زیر نمودار تمامی سطل‌ها با هم برابر است. بنابراین، احتمال وقوع در این سطل‌ها نیز با هم برابر است. همچنین توزیع داده‌ها در سطل‌های هم‌احتمال بسیار موثراست [۱۱].

## ۳. تاریخچه پژوهش

در سال‌های اخیر، روش‌های متفاوتی برای تشخیص کانال‌های فرمان و کنترل باتنت‌ها پیشنهاد شده است. از این‌رو، مدیران بات سعی می‌کنند کانال‌های فرمان و کنترل جدیدی برای فرار از روش‌های تشخیص موجود توسعه دهند. به همین دلیل کانال‌های

[1] Conditional Entropy (CE)

[2] Corrected Conditional Entropy (CCE)
[3] Empirical probability density function
[4] Porta

فرمان و کنترل نسل‌های آینده بات‌نت‌ها درحال تحول به سمت ارتباطات پنهان و پیچیده است. استفاده از کانال‌های پنهان زمانی در بات‌نت‌های نسل‌های آینده این پیچیدگی را افزایش داده و کانال‌ها را مخفی‌تر می‌کند، چرا که تشخیص ترافیک آلوده از ترافیک سالم مشکل است.

روش‌های مبتنی بر امضاها و الگوهای رایج شامل اولین روش‌های تشخیص بودند که در حال حاضر نیز اساس عملکرد تعداد زیای از نسل‌های سیستم‌های تشخیص نفوذ هستند [۵]. این روش‌ها در تشخیص بات‌نت‌های ناشناخته کارایی لازم را نداشتند، لذا روش‌های تشخیص ناهنجاری مطرح شدند که اساس کار آن‌ها ناهنجاری‌های ناشی از تاثیر بر عملکرد شبکه بود، این روش‌ها دارای اشکالات فراوانی هستند که از آن جمله می‌توان به وابستگی زیاد به تغییرات شبکه و عامل‌های انسانی اشاره کرد. با شکل‌گیری بات‌نت‌هایی که از روش‌های پیچیده برای مبهم‌سازی ارتباط‌های خود و همچنین مخفی‌سازی سرورهای فرمان و کنترل استفاده می‌کنند، روش‌های مبتنی‌بر رفتار گروهی بات‌ها ظهور کردند [۵].

بات‌نت‌های نسل آینده، رفتار گروهی متغیر دارند و به سرعت رفتار خود را در برقراری ارتباط با سرورهای فرمان و کنترل تغییر می‌دهند، کانال‌های پنهان متنوعی برای برقراری ارتباط دارند و ترافیک مجاز شبکه را برای فرار از تشخیص تقلید می‌کنند [۱۳].

## ۳-۱. پژوهش‌های حوزه کانال‌های فرمان و کنترل بات‌نت‌ها

در این بخش، پژوهش‌های مرتبط با روش‌های تشخیص بات‌نت‌ها و روش‌های تشخیص کانال‌های فرمان و کنترل آن‌ها مورد بررسی قرار می‌گیرند.

بات ماینر[1] [۱۴] روشی است که برای تشخیص ترافیک فرمان و کنترل بات‌نت، از روش‌های داده‌کاوی استفاده می‌کند. این بات روش بات اسنیفر[2] [۱۵] را ارتقا می‌بخشد. بات ماینر ارتباطات مشابه و ترافیک مخرب را خوشه‌بندی می‌کند؛ سپس همبستگی میان‌دسته‌ای را برای شناسایی میزبان‌هایی اجرا می‌کند که هم ارتباطات مشابه و هم الگوهای فعالیت مخرب دارند. بات ماینر ابزار پیشرفته تشخیص بات‌نت است که مستقل از پروتکل و ساختار بات‌نت است و می‌تواند بات‌نت‌های IRC، HTTP و P2P را با درصد خطای پایین تشخیص دهد.

اغلب بات‌نت‌های جدید از شبکه‌های فست فلاکس[3] [۱۸-۱۶] به‌عنوان روش فرمان و کنترل خود استفاده می‌کنند. فست فلاکس روشی مبتنی‌بر DNS است که برای پنهان کردن حملات فیشینگ[4] و سایت‌های توزیع بدافزار استفاده می‌شود. این شبکه‌ها پشت شبکه‌ای از میزبان‌های به‌خطرافتاده استفاده می‌شوند که همواره در حال تغییر هستند و به‌عنوان پروکسی[5] عمل می‌کنند. آقای شریف‌نیا و همکاران [۱۹] روشی مبتنی‌بر شهرت ارایه کرده‌اند که بات‌نت‌هایی را تشخیص می‌دهد که از الگوریتم‌های تولید نام دامنه برای کانال فرمان و کنترل خود استفاده می‌کنند.

ابراهیم غفیر[6] و همکاران [۲۰] روشی دومرحله‌ای به نام BotDet برای تشخیص ترافیک کانال فرمان و کنترل بات‌نت‌ها ارایه کرده‌اند. مرحله اول این روش، طراحی چهار ماژول است. شامل ماژول تشخیص آدرس آی پی مخرب (MIPD)، ماژول تشخیص گواهی نامه‌های مخرب SSL (MSSLD)، ماژول تشخیص دامنه فلاکس (DFD) و ماژول تشخیص اتصالات تُر (TorD). مرحله دوم نیز چارچوبی برای همبستگی بین این چهار ماژول است که باعث کاهش نرخ هشدار نادرست می‌شود. آن‌ها در روش خود قابلیت تشخیص بی‌درنگ[7] را بهبود دادند.

دایتریش[8] و همکاران [۲۱] فیدربات[9] را ارایه کرده‌اند که کانال فرمان و کنترل آن، رکوردهای TXT از DNS است. ترافیک فرمان و کنترل این بات به تعدادی قطعه تقسیم می‌شود و در فیلد rdata، پاسخ DNS از رکورد TXT ارسال می‌شوند و با استفاده از الگوریتم Base 64 کدگذاری شده است؛ پیام‌های ارسالی نیز با استفاده از رمز جریانی RC4 رمزگذاری می‌شوند.

نازریو[10] [۲۲] بات‌نتی ارایه کرده است که از توییتر برای کانال فرمان و کنترل خود استفاده می‌کند. بات‌ها به‌روزرسانی‌ها را از طریق فیدهای RSS می‌خوانند و پیام‌ها را کدگشایی می‌کنند. URLهایی که به واسطه کدگشایی پیام ایجاد می‌شوند نیز برای دانلود محتوای مخرب و بر روی ماشین آلوده استفاده می‌شوند.

استگوبات[11] بات‌نتی است که کانال فرمان و کنترل آن مبتنی‌بر نهان‌نگاری در تصاویر اشتراکی کاربران شبکه اجتماعی است. [۲۳]. نحوه ارتباط و آلوده‌سازی در این بات‌نت به این شکل است که میزبان آلوده به بات تنها درصورتی می‌تواند با یک میزبان آلوده به بات دیگر ارتباط برقرار کند که بین کاربران این

[1] BotMiner
[2] BotSniffer
[3] Fast-flux
[4] Phishing Attacks
[5] Proxy
[6] Ibrahim Ghafir
[7] Real time
[8] Dietrich
[9] Feider
[10] Nazario
[11] Stegobot

میزبان‌ها رابطه دوستی وجود داشته باشد. شبکه فیس‌بوک زیرساخت ارتباطی است که اطلاعات را از سمت بات‌ها به مدیر بات منتقل می‌کند.

نوع دیگری از کانال‌های پنهان مورد استفاده در بات‌نت‌ها، شبکه تُر[1] است. اونیون‌بات[2] یک بات‌نت شبکه تُر است که توسط امیرعلی صنعتی‌نیا [۲۴] و همکاران پیشنهاد شده است.

ریچر[3] و همکاران [۲۵] روشی برای شناسایی کانال‌های فرمان و کنترل ارایه کرده‌اند که مبتنی بر اندازه‌گیری منظم بودن جریان داده‌ها است. در این روش از برآورد آنتروپی نزدیکترین همسایگی[4] استفاده شده است. آن‌ها میانگین، انحراف استاندارد و آنتروپی فواصل زمانی و اندازه داده‌های ارتباطی بین مبداء و مقصد را محاسبه کرده، سپس برای تشخیص فعالیت کانال‌های فرمان و کنترل به‌عنوان ورودی به یک ماشین بردار پشتیبان[5] دادند. دسته‌بندی که آن‌ها ارایه کرده‌اند ترافیک آلوده به بات را تشخیص می‌دهد.

## ۳-۲. پژوهش‌های مرتبط با کانال‌های پنهان زمانی

کانال پنهان زمانی ساده توسط کابوک و همکاران [۲۶] ارایه شده است که در آن فرستنده و گیرنده بر روی یک فاصله زمانی t توافق می‌کنند. بر اساس آن، برای کدگذاری بیت "۱"، یک بیت ارسال و برای کدگذاری بیت "۰" ، بیتی ارسال نمی‌شود. مزیت اصلی این کانال آن است که وقتی یک بسته گم می‌شود، بیت، پرش می‌کند ولی هم‌زمانی به‌هم نمی‌خورد. در حقیقت زمان t و تعداد بیت‌های صفر بین دو بیت‌های ۱، توزیع تاخیر بین بسته‌ای کانال را تعیین می‌کنند.

همین نویسنده در [۲۷] کانال پنهان زمانی دیگری به نام کانال پنهان زمانی تکرار زمان[6] پیشنهاد کرد که مبتنی بر حمله تکرار است. این کانال مجموعه‌ای از ترافیک سالم $S_{\text{in}}$ به‌عنوان ورودی در نظر می‌گیرد و برای ارسال پیغام پنهان بازبخش می‌کند. $S_{\text{in}}$ از طریق تعریف یک مقدار منقطع، به دو سطل مساوی $s_1$ و $s_2$ تقسیم می‌شود. این کانال بیت "۱" را از طریق بازپخش تصادفی یک تاخیر بین بسته‌ای از سطل $s_1$ و بیت "۰" را از طریق بازپخش تصادفی یک تاخیر بین بسته‌ای از سطل $s_2$ ارسال می‌کند.

جیان وچیو و همکاران [۲۸] کانال پنهان زمان مبتنی بر مدل[7] را ارایه کرده‌اند. این کانال دارای چهار بخش فیلتر، تحلیلگر، کدگذار و انتقال دهنده است که برای تقلید خواص آماری ترافیک مجاز طراحی شده است، طرز عملکرد این کانال بدین شکل است که ابتدا بخش فیلتر، ترافیک خروجی را به نوع خاصی از ترافیک که بهترین مدل آماری است جدا می‌سازد. سپس بخش تحلیل‌گر با استفاده از خطای میانگین مربع ریشه[8] بهترین مدل توزیع را پیدا کرده و آن را برای تقلید ترافیک شبکه انتخاب می‌کند. به دلیل تغییر توزیع ترافیک در ارسال‌ها و دریافت‌های متوالی، این مدل به‌صورت خودکار بعد از هر ۱۰۰ بسته به‌روز رسانی می‌شود. در بخش کدگذار، با استفاده از تابع توزیع معکوس[9]، مدل نهایی انتخاب شده و کدگشایی با استفاده از تابع توزیع تجمعی[10] انجام می‌شود. شکل ترافیک این کانال تقریباً مشابه ترافیک مجاز است ولی از آنجا که هیچ همبستگی بین تاخیرهای بین بسته‌های متوالی وجود ندارد؛ ترافیک آن منظم است.

کوتاری و همکاران [۲۹] کانال پنهان زمانی Mimic را ارایه کرده‌اند که قادر است شکل و نظم ترافیک مجاز را با استفاده ویژگی‌های آماری ترافیک تقلید کند. ایده اصلی Mimic، تقلید باز رخداد الگوهای تکراری است. Mimic، دو بخش اصلی شکل[11] و نظم[12] دارد. بخش شکل، توزیع ترافیک مجاز را تقلید می‌کند و بخش نظم، میزان بی‌نظمی ترافیک سالم را با حفظ آنتروپی ترافیک مجاز و استفاده از درخت نظم، تقلید می‌کند.

جیترباگ[13] کانال غیرفعالی است که نحوه افشای اطلاعات محرمانه مثل رمزهای عبور را بررسی می‌کند [۳۰]. یکی از ویژگی‌های این کانال این است که برای ارسال اطلاعات پنهان احتیاجی به آلوده‌کردن ماشین میزبان نیست و این کار تا حد زیادی در عدم توانایی تشخیص آن تاثیر دارد. این کانال با اضافه کردن تاخیر کوچکی به زمان‌های ترافیک برنامه‌هایی مانند SSH، Telnet و RDP اطلاعات را کدگذاری کرده و ارسال می‌کند. برای انجام عمل کدگذاری، فرستنده تاخیر کوچکی به زمان‌های ترافیک اضافه می‌کند. گیرنده نیز با محاسبه تاخیر بین بسته‌های متوالی، اطلاعات را رمزکدگشایی می‌کند.

## ۴. بات‌نت با کانال فرمان و کنترل پنهان زمانی

همان‌طور که در بخش ۱ اشاره شد هر بات‌نت برای ارتباط با

[1] Tor
[2] Onion
[3] Richer
[4] Nearest neighbor
[5] Support Vector Mashine(SVM)
[6] Time reply Covert Timing Channel(TRCTC)
[7] Model-Based Covert Timing Channel(MBCTC)
[8] Root Mean Square Error(RMSE)
[9] Inverse Distribution Function
[10] Cumulative Distribution Function (CDF)
[11] Shape
[12] Regularity
[13] Jitterbug

بات‌های خود از یک کانال استفاده می‌کند. بات‌نت زمانی که در این بخش ارایه می‌شود بات‌نتی است که برای برقراری ارتباط و ارسال فرمان به بات‌ها از کانال فرمان و کنترل پنهان زمانی استفاده می‌کند. کانال‌های پنهان زمانی به دو دسته‌های اصلی مبتنی بر تاخیر بین بسته‌ای[1] و مبتنی بر مرتب‌سازی دوباره بسته‌ها[2] تقسیم می‌شوند. در این مقاله از کانال پنهان زمانی مبتنی بر تاخیر بین بسته‌ای استفاده شده است.

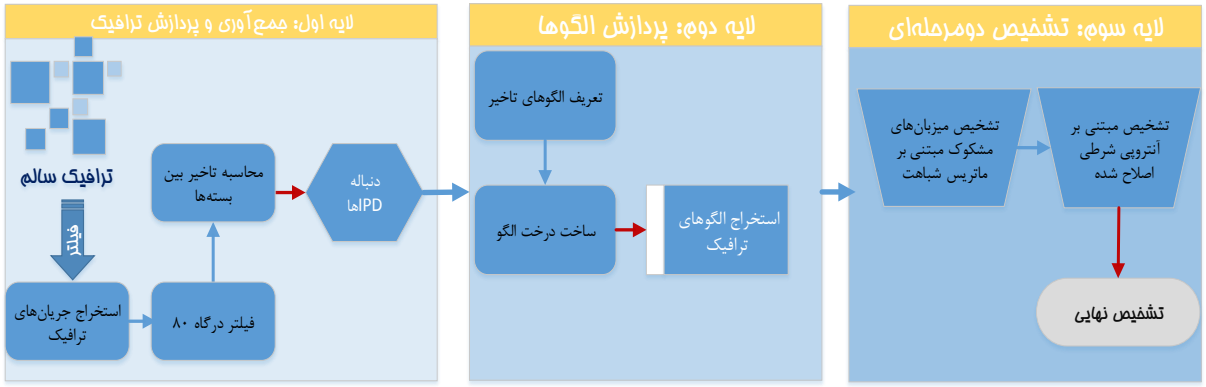
مدل بات‌نت پیشنهادی، مطابق ساختار بات‌نت‌های متمرکز است که در آن مدیر بات از پروتکل HTTP برای برقراری ارتباط با سرویس‌دهنده‌ها و بات‌ها استفاده می‌کند. این بات‌نت، برای ارتباط با بات‌های خود از کانال پنهان زمانی استفاده می‌کند. در این مدل، با دستکاری تاخیر زمانی بین بسته‌ها، فرمان‌های مدیر بات به ماشین‌های آلوده ارسال می‌شوند.

## ۴-۱. روش تشخیص بات‌نت زمانی

روش پیشنهادی تشخیص بات‌نت زمانی، مبتنی بر یافتن الگوهای تکراری است. در این روش، که مستقل از ساختار و پروتکل استفاده‌شده در کانال فرمان و کنترل است، ضمن بررسی و تشخیص بازرخداد الگوهای تکراری، فعالیت‌های گروهی ماشین‌های آلوده به بات در یک شبکه تحت نظارت، تشخیص داده می‌شوند.

## ۴-۲. معماری روش تشخیص پیشنهادی

شکل (۱)، معماری روش تشخیص پیشنهادی را نشان می‌دهد که شامل سه لایه است. لایه اول، جمع‌آوری و پردازش ترافیک، لایه دوم، پردازش الگوها و لایه سوم، تشخیص دومرحله‌ای است. در ادامه، هر کدام از لایه‌ها توضیح داده می‌شوند.




**شکل (۱):** معماری روش تشخیص پیشنهادی

### ۴-۲-۱. لایه اول: جمع‌آوری و پردازش ترافیک

در این مرحله ابتدا با توجه به ترافیک ورودی، ترافیک HTTP، فیلتر شده و ترافیک درگاه ۸۰ ذخیره می‌شود. سپس جریان‌های موجود در این ترافیک استخراج می‌شوند. در انتها نیز تاخیر بین بسته‌های جریان‌های استخراج شده، محاسبه می‌شوند.

تعریف ۲ (جریان ترافیک): جریان ترافیک، دنباله‌ای از بسته‌ها با آدرس آی پی مبداء، درگاه مبداء، آدرس آی پی مقصد، درگاه مقصد و پروتکل یکسان در خلال یک پنجره زمانی مشخص است.

### ۴-۲-۲. لایه دوم: پردازش الگوها

در این مرحله، پردازش الگوهای استخراج شده، پردازش‌شده و شباهت آن‌ها مقایسه می‌شوند.

تعریف ۳ (الگوهای تاخیر): دنباله‌ای منحصربه‌فرد از تاخیرهای بین بسته‌هاست. بسته‌هایی که در یک دنباله قرار می‌گیرند حامل ویژگی‌های یکسانی هستند.

با استفاده از روش توزیع سطل که در بخش ۲-۴ ارایه شد، تاخیرهای محاسبه شده در سطل‌های هم احتمال توزیع شده و شماره سطل متناظر با تاخیرها مشخص می‌شود.

تعریف ۴ (مجموعه الگوها): با فرض داشتن دنباله‌ای از الگوها، مجموعه تمامی الگوهایی که طول‌های متفاوت دارند، از رابطه (۲) تعریف می‌شوند:

$$\mathcal{P} = \{p_1, p_2, \dots, p_\ell\} \qquad (۲)$$

که در آن، $p_1$ نشان‌دهنده الگوهای به طول یک، $p_2$ نشان‌دهنده الگوهای به طول دو و به همین ترتیب $p_\ell$ نشان‌دهنده الگوهای به طول $\ell$ است.

برای ذخیره این مجموعه الگوها در هر میزبان و پردازش و مقایسه با میزبان‌های دیگر از ساختار درخت k-تایی استفاده

[1] Inter Packet Delay(IPD)
[2] Packet reordering

شده است. به عبارت دیگر تمامی جریان‌های هر میزبان تشکیل یک درخت $k$-تایی ریشه دار می‌دهند. که در آن $k$ برابر است با $Q$، و نشان‌دهنده حداکثر تعداد فرزندی است که هر گره می‌تواند داشته باشد؛ این مقدار برابر است با تعداد سطل‌های انتخاب‌شده در روش توزیع سطل‌ها؛ عمق درخت $\ell$ است، که برابر با حداکثر طول الگوها است. در این درخت، تمامی برگ‌ها در عمق $\ell$ قرار می‌گیرند و هر الگو به طول $\ell$، مسیری منحصربه‌فرد به طول $\ell$ از ریشه تا عمق $\ell$ در درخت ایجاد می‌کند. بنابراین، تمامی الگوهای به طول یک، مسیری به طول یک، الگوهای به طول دو، مسیری به طول دو، و به همین ترتیب الگوهای به طول $\ell$، مسیری به طول $\ell$ در درخت ایجاد می‌کنند.

### ۴-۲-۳. لایه سوم: تشخیص

این لایه روش تشخیص دو مرحله‌ای را از طریق دو مولفه تعیین شباهت الگوها و محاسبه آنتروپی ارایه می‌کند. ابتدا نرخ شباهت الگوی میزبان‌ها با یکدیگر محاسبه می‌شوند. از طریق تعیین حد آستانه $\gamma_t$، میزبان‌های مشکوک به بات مشخص شده و برای تشخیص نهایی بوسیله مولفه آنتروپی مورد آزمون قرار می‌گیرند. در ادامه جزییات این روش ارایه می‌شود.

- **مولفه شباهت الگوها**

این مولفه وظیفه تعیین میزان شباهت الگوهای هر میزبان با میزبان‌های دیگر را بر عهده دارد.

تعریف ۵ (الگوهای مشابه): به دو یا چند الگو، مشابه گفته می‌شود اگر تک‌تک اعضای این الگوها، با هم برابر باشند.

درخت هر میزبان شامل تمامی الگوهای آن میزبان است، برای ذخیره تعداد الگوهای مشابه هر میزبان با دیگر میزبان‌ها، از یک ماتریس دو بعدی $n \times \ell$ استفاده می‌شود که $n$ نشان‌دهنده تعداد میزبان‌ها و $\ell$ نشان‌دهنده حداکثر طول الگوها است. این ماتریس با $\mathcal{M}_{n\times\ell}$ نشان داده می‌شود. بنابراین، سطر $i$ ماتریس، تعداد الگوهای مشابه به طول یک تا $\ell$ از میزبان $i$ام را با دیگر میزبان‌ها نشان می‌دهد. به‌عنوان‌مثال، عنصر $\mathcal{M}_{i,j}$، برابر تعداد الگوهای مشابه به طول $j$ است که میزبان $i$-ام با دیگر میزبان‌ها غیر از خودش دارد.

الگوریتم (۱)، محاسبه شباهت درخت الگوی هر میزبان و مقایسه الگوهای میزبان‌ها با یکدیگر را نشان می‌دهد. در الگوریتم (۱)، $temp.child[p_\ell^k]$ نشان‌دهنده فرزند $p_\ell^k$ام از گره $temp$ در درخت، و $p_\ell^k$ نشان‌دهنده عنصر $k$-ام از الگوی $p_\ell$، و $p_\ell$ یک الگو به طول $\ell$ است. برای مثال اگر $p_6 = \{2,3,1,1,5,4\}$ باشد، آن‌گاه $p_6^2 = 3$ است. فرض کنیم الگوی میزبان $j$-ام $p_6 = \{2,3,1,1,5,5\}$ باشد. در این‌صورت ابتدا بررسی می‌شود که آیا مسیری برای الگوی {2} در درخت وجود دارد یا خیر. اگر مسیری وجود داشته باشد تعداد الگوهای به طول یک میزبان‌های $i$-ام و $j$-ام، یک واحد افزایش می‌یابد یعنی $\mathcal{M}_{i,1} = \mathcal{M}_{i,1} + 1$ و $\mathcal{M}_{j,1} = \mathcal{M}_{j,1} + 1$ است. برای الگوی {2,3} نیز این کار صورت می‌گیرد و اگر مسیری در درخت پیدا شود، تعداد الگوهای به طول دو میزبان‌های $i$-ام و $j$-ام، یک واحد افزایش می‌یابد یعنی $\mathcal{M}_{i,2} = \mathcal{M}_{i,2} + 1$ و $\mathcal{M}_{j,2} = \mathcal{M}_{j,2} + 1$ است. برای برای یافتن مسیرهای مشابه بزرگ‌تر نیز به همین ترتیب ادامه می‌یابد.

**الگوریتم (۱):** محاسبه شباهت میزبان‌ها

**Algorithm 1:** Calculation of Similarity Hosts

```
n: number of hosts
ℓ: pattern length
1.  begin
2.    for i ← 1 to n − 1
3.      root ← tree_host i
4.      for all p_ℓ ∈ P_i
5.          Insert p_ℓ into tree_host i
6.      end for
7.      for j ← i + 1 to n
8.        for all p_ℓ ∈ P_j
9.          temp ← root
10.         for k ← 1 to ℓ
11.           if temp.child[p_ℓ^k] <> null then
12.             M_i,k ← M_i,k + 1
13.             M_j,k ← M_j,k + 1
14.             temp ← temp.child[p_ℓ^k]
15.           else
16.             break
17.           end if
18.         end for
19.     end for
20.   end for
21. End
```

پس از محاسبه الگوهای مشابه و به‌دست آوردن ماتریس $\mathcal{M}_{n\times\ell}$، احتمال رخداد الگوها به‌ازای هر میزبان از رابطه (۳) محاسبه می‌شود.

$$\mathcal{S}_{i,j} = \frac{\mathcal{M}_{i,j}}{\sum_{i=1}^{n} \mathcal{M}_{i,j}} \quad (1 \le i \le n\,, 1 \le j \le \ell) \qquad (۰)$$

$\mathcal{S}_{i,j}$ مقدار احتمال الگوهای به طول $j$ است که میزبان $i$-ام با دیگر میزبان‌ها غیر از خود دارد.

- **وزن‌دهی به الگوها**

ارزیابی میزان تاثیر الگوها با طول متفاوت از طریق وزن‌دهی به آن‌ها انجام می‌شود. الگوهای مشابه کوتاه‌تر، نسبت به الگوهای مشابه بلندتر فرکانس رخداد بیشتری دارند. به‌عنوان مثال، تعداد الگوهای مشابه به طول یک، بیشتر یا مساوی تعداد الگوهای مشابه به طول دو، و تعداد الگوهای مشابه به طول $\ell - 1$، بیشتر

یا مساوی تعداد الگوهای مشابه به طول $\ell$ است. به همین دلیل نیز احتمال یافتن الگوهای مشابه کوتاه‌تر بیشتر از احتمال یافتن الگوهای مشابه بلندتر است. برای مثال، احتمال یافتن دو الگوی مشابه به طول یک در دو میزبان، بیشتر از احتمال یافتن دو الگوی مشابه با طول ۵۰ است. بنابراین، الگوهای کوتاه‌تر نسبت به الگوهای بلندتر، تاثیر کمتری در میزان شباهت میزبان‌ها خواهند داشت. بر این اساس، در وزن‌دهی به الگوها، الگوهای کوتاه‌تر، با وزن کمتر و الگوهای بلندتر، با وزن بیشتری محاسبه می‌شوند.

تعریف ۶ (نرخ تشابه): جمع وزن‌دار الگوهای مشابه با طول متفاوت، نرخ تشابه نامیده می‌شود. این نرخ از رابطه (۴) تعریف می‌شود.

$$\gamma_i = \sum_{j=1}^{\ell} w_j |S_{i,j}| \qquad 1 \le i \le n \ , \quad 1 \le j \le \ell, w_{j-1} < w_j \tag{۰}$$

که در آن، $|S_{i,j}|$ نشان‌دهنده احتمال شباهت الگوهای به طول $j$ میزبان $i$-ام با دیگر میزبان‌ها، $w_j$ نشان‌دهنده وزن الگوهای به طول $j$، $\ell$ نشان‌دهنده حداکثر طول الگوها و $\gamma_i$ نشان‌دهنده نرخ شباهت میزبان $i$-ام با دیگر میزبان‌ها است.

تعریف ۷ (میزبان‌های مشابه): به دو یا چند میزبان، مشابه گفته می‌شود، اگر نرخ شباهت آن‌ها از یک حد آستانه $\gamma_t$ بیشتر باشد.

- **مولفه آنتروپی**

این مولفه وظیفه تشخیص میزبان‌های آلوده را بر اساس محاسبه آنتروپی الگوها بر عهده دارد.

در روش تشخیص مبتنی بر آنتروپی، ابتدا آنتروپی جریان‌های سالم محاسبه شده و مقدار مرجع برای آنتروپی محاسبه می‌گردد. سپس آنتروپی میزبان‌های مشکوک به بات که در مولفه شباهت الگو تشخیص داده شده‌اند نیز محاسبه می‌گردند. اگر مقدار آنتروپی میزبان‌های مظنون به بات، از حد آستانه تعیین شده برای آنتروپی ترافیک مجاز تجاوز کند، این میزبان‌ها به‌عنوان میزبان‌های آلوده به بات تشخیص داده می‌شوند.

- **محاسبه آنتروپی شرطی اصلاح‌شده میزبان‌های مشکوک**

در روش آنتروپی برای تشخیص میزبان‌های آلوده از مفهوم درخت نظم استفاده شده است. این مفهوم اولین بار در پژوهش [۲۸] ارایه شد و در مرجع [۲۹] نیز توسعه داده شده است. درخت نظم برای کنترل میزان بی‌نظمی در کانال پنهان پیشنهاد شده است. در این مقاله برای محاسبه آنتروپی شرطی اصلاح شده از این مفهوم استفاده شده است.

نحوه کار به این شکل است که الگوهای به طول $\ell$ هر یک از میزبان‌های مشکوک در درخت مختص به خود درج می‌گردند. میانگین گره‌های ظاهر شده در هر سطح نیز مطابق الگوریتم ارایه شده در مرجع [۲۹] تعیین می‌شوند. بر این اساس، ریشه درخت در سطح صفر قرار گرفته و حاوی هیچ الگویی نیست. سطح اول درخت دربردارنده تمامی گره‌های مجاز الگوهای به طول یک، سطح دوم درخت دربردارنده تمامی گره‌های مجاز الگوهای به طول دو و به همین ترتیب، سطح $\ell$ درخت دربردارنده تمامی گره‌های مجاز الگوهای به طول $\ell$ است. آنتروپی شرطی اصلاح‌شده، از طریق رابطه (۱)، و با داشتن الگوهای موجود در جریان‌های ترافیک، میزان رخداد هر الگو و همچنین تعداد الگوهای منحصربه‌فرد محاسبه می‌شود.

## ۵. آزمایش‌ها

در این بخش، نتایج روش پیشنهادی با انجام مجموعه‌ای از آزمایش‌ها جهت تشخیص میزبان‌های آلوده ارایه می‌شود. برای ارزیابی روش پیشنهادی، پنج نوع کانال پنهان زمانی IPCTC، TRCTC، MBCTC، Mimic و Jitterbug مورد استفاده قرار می‌گیرند.

تمرکز روش تشخیص، ایجاد تمایز بین میزبان‌های آلوده به بات و میزبان‌های سالم است. برای افزایش کارایی آزمون‌ها، ترافیک سالم و داده‌های پَرت[1] اهمیت زیادی دارند زیرا در صورت وجود هم‌پوشانی قابل توجه بین مقادیر حاصل از آزمون برای نمونه‌های سالم و آلوده، روش تشخیص پیشنهادی نرخ خطای تشخیص پایینی خواهد داشت.

### ۵-۱. مجموعه داده

داده‌های مورد استفاده در آزمایش‌ها از بسته‌های مبادله شده در شبکه و از مجموعه داده دارپا[2] [۳۱]، جمع‌آوری شده است. بسته‌های HTTP و SSH این مجموعه به ترتیب ۴۸۳۵۶۶۰ و ۷۸۵۴۲۸، جداسازی شد (جدول (۱)). در مرحله بعد از میان این بسته‌ها، جریان‌ها به صورت یک‌سویه از سمت سرورهای وب به سمت میزبان‌های شبکه استخراج می‌گردد.

**جدول (۱):** بسته‌های استخراج‌شده از مجموعه داده دارپا

| شاخص بسته‌ها | SSH | HTTP |
|---|---|---|
| **تعداد جریان‌ها** | ۱۲۲۸ | ۴۶۸۴ |
| **مجموع بسته‌ها در کل جریان‌ها** | ۷۸۵۴۲۸ | ۴۸۳۵۶۶۰ |
| **میانگین تعداد بسته‌ها در هر جریان** | ۶۳۹ | ۱۰۳۲ |

[1] Outlier
[2] DARPA

جدول (۲)، تقسیم‌بندی جریان‌ها را به زیرمجموعه‌های آموزشی و آزمون نشان می‌دهد. برای ترافیک HTTP، هر زیرمجموعه متشکل از ۷۰۰ جریان و در حدود ۷۰۰۰۰۰ بسته است. همچنین هر زیرمجموعه SSH حاوی ۴۰۰ جریان و در حدود ۲۵۵۰۰۰ بسته است. برای افزایش دقت محاسبه سطوح آستانه، هر آزمایش به تعداد ۱۰۰ بار تکرار می‌شود.

**جدول (۲):** تقسیم‌بندی جریان‌ها به زیرمجموعه‌های تحت آزمون

| | HTTP | SSH |
|---|---|---|
| مجموعه آموزش | ۷۰۰ | ۴۰۰ |
| مجموعه آزمون | ۷۰۰ | ۴۰۰ |
| IPCTC | ۷۰۰ | - |
| TRCTC | ۷۰۰ | - |
| MBCTC | ۷۰۰ | - |
| Mimic | ۷۰۰ | - |
| Jittebug | - | ۴۰۰ |

## ۵-۲. پیاده‌سازی بات‌نت‌های زمانی

بات‌های زمانی از طریق کدهای C#.Net و روش‌های کدگذاری کانال‌های IPCTC، TRCTC، MBCTC، Jitterbug و Mimic پیاده‌سازی شده و محاسبات آماری آنها نیز با استفاده از زبان R انجام می‌شود. تاخیر بین بسته‌ها نیز از طریق محاسبه زمان تولید بسته‌ها به‌دست می‌آید.

## ۵-۳. سکوی آزمایش

برای انجام آزمایش‌ها شبکه‌ای شامل سرویس‌دهنده و میزبان در نظر گرفته می‌شود و برخی از این میزبان‌ها به بات‌های زمانی آلوده می‌شوند. قطعه کدی به زبان C# شامل دو نسخه سرویس‌دهنده و مشتری تولید شد که نسخه سرویس‌دهنده، نقش کانال فرمان و کنترل را دارد و فرمان‌ها را از طریق این کانال ارسال می‌کند. میزبان آلوده یا بات نیز با استفاده از الگوی کدگشایی کانال پنهان زمانی (الگوی تاخیرهای زمانی بین بسته‌ها)، فرمان‌ها را دریافت می‌کند.

نسخه سرویس‌دهنده، بر روی یک سرویس‌دهنده وب و بر روی ماشین بیرون از شبکه داخلی قرار می‌گیرد و نسخه‌های مشتری نیز بر روی هفت میزبان ماشین مجازی نصب می‌شود. برای جمع‌آوری ترافیک بات‌های زمانی، سه فرمان ۱۶۰ بیتی تولید شده و با استفاده از طرح کدگذاری کانال‌های پنهان به تاخیرهای زمانی تبدیل می‌شوند. برای جلوگیری از ایجاد الگوهای خاص، از قبیل حضور بیت‌های تکراری ناشی از استفاده از کدهای اسکی[1]، فرمان‌ها به صورت تصادفی تولید می‌شوند.

مطابق جدول (۲)، مجموعه‌های ۷۰۰تایی به ۱۰۰ زیرمجموعه ۷تایی، و مجموعه‌های ۴۰۰تایی به ۱۰۰ زیرمجموعه ۴تایی تقسیم می‌شوند و در هر زیرمجموعه ۷تایی، دو جریان به بات زمانی آلوده می‌شوند و پنج جریان دیگر بدون تغییر باقی می‌مانند. در زیرمجموعه‌های ۴تایی نیز دو جریان به بات زمانی آلوده می‌شوند و دو جریان دیگر نیز بدون تغییر باقی می‌مانند. آزمایش‌ها برای هر بات زمانی و به تعداد ۱۰۰ مرتبه تکرار، انجام می‌شود.

## ۵-۴. نحوه تشخیص

هدف روش تشخیص، ایجاد تمایز بین میزبان‌های آلوده به بات و میزبان‌های سالم است. برای این منظور آزمایش‌های تشخیص میزبان‌های آلوده به بات به گونه‌ای تنظیم شد که مبنای نرخ مثبت نادرست، ۰/۰۱ باشد. برای دست‌یابی به این نرخ، نقاطی که در آن‌ها نمونه سالم یا آلوده به بات تصمیم‌گیری می‌شود بر روی یک درصد ابتدایی ( صفر الی ۱ درصد) یا یک درصد انتهایی (۹۹ الی ۱۰۰ درصد) مقادیر نمونه سالم تنظیم می‌شود. پس از تعریف سطوح آستانه، میزبان‌هایی که مقدار جریان‌های آنها خارج از این حد آستانه باشد، به‌عنوان میزبان آلوده و بقیه میزبان‌ها نیز سالم گزارش می‌شوند.

در روش پیشنهادی با استفاده از مجموعه آموزشی، ابتدا حد آستانه نرخ شباهت برای جریان‌های سالم محاسبه می‌شود. بیشترین مقدار به‌دست آمده در ۱۰۰ آزمایش انجام شده بر روی نمونه‌های سالم، به‌عنوان حد آستانه نرخ شباهت در نظر گرفته می‌شود. با توجه به این که میزبان‌های آلوده به بات در فعالیت‌های گروهی شرکت می‌کنند، انتظار می‌رود که نسبت به سایر میزبان‌های شبکه، شباهت بیشتری با یکدیگر داشته باشند و از الگوی خاصی پیروی کنند. مطابق معماری روش تشخیص دومرحله‌ای، تمامی جریان‌هایی که نرخ شباهت آن‌ها از حد آستانه فراتر رود مشکوک به بات تشخیص داده شده و برای بررسی بیشتر به مولفه تشخیص آنتروپی ارسال می‌شوند.

در مولفه تشخیص آنتروپی، با استفاده از مجموعه‌های آموزشی و آزمون، حد آستانه بالا و پایین آنتروپی شرطی اصلاح شده برای میزبان‌های سالم محاسبه می‌شود. با توجه به شباهت ترافیک میزبان‌های آلوده به بات و حضور الگوهای مشابه در آن‌ها، انتظار می‌رود که آنتروپی دو یا چند میزبان آلوده به بات پایین‌تر یا بالاتر از حد مجاز باشد. همچنین انتظار می‌رود که با افزایش تعداد بات‌ها، این آنتروپی کاهش پیدا کند. به همین جهت، اگر مقدار آنتروپی میزبان‌ها از حد آستانه تجاوز کند، به‌عنوان میزبان آلوده به بات تشخیص داده می‌شوند.

[1] ASCII

## ۶. نتایج آزمایش‌ها و ارزیابی

در این بخش، نتایج حاصل از روش تشخیص پیشنهادی برای تشخیص میزبان‌های آلوده مورد بررسی و مقایسه قرار می‌گیرد.

### ۶-۱. آزمایش‌ها

برای ارزیابی نتایج، آزمون‌های آنتروپی شانون، آنتروپی شرطی و آنتروپی شرطی اصلاح‌شده روی داده‌های کانال‌های IPCTC، TRCTC، MBCTC، Jitterbug و Mimic انجام می‌شود.

- **آزمون آنتروپی.** در این آزمون تعداد سطل‌ها برابر ۶۵,۵۳۶ و ارتفاع الگوها برابر یک است. اگر مقدار آنتروپی پایین باشد، نشان‌دهنده آلوده بودن جریان‌های مورد بررسی به بات زمانی است. زیرا توزیع نمونه مورد آزمایش، مشابه توزیع ترافیک سالم نیست.

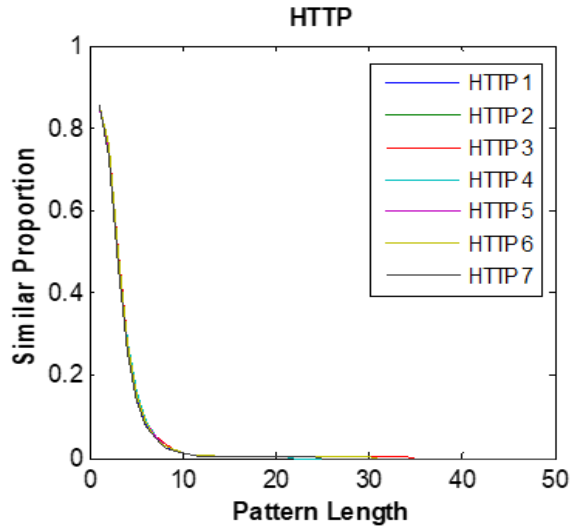
**آزمون آنتروپی شرطی اصلاح شده.** در این آزمون تعداد سطل‌ها برابر ۵ است. اگر مقدار آنتروپی شرطی اصلاح شده کمتر از مقدار آستانه باشد، نشان‌دهنده وجود بات زمانی است. هر چه نمونه مورد آزمایش منظم‌تر باشد، مقدار آنتروپی شرطی کمتر است. همچنین آنتروپی شرطی در میزبان‌هایی که در مولفه تشخیص الگوهای مشابه، مشکوک به بات شناخته شده‌اند کاهش می‌یابد.

### ۶-۲. سنجه‌های ارزیابی

سنجه‌های ارزیابی برای روش پیشنهادی، نرخ تشخیص[1] و نرخ هشدار نادرست[2] است. نرخ تشخیص، نسبت تعداد بات‌های صحیح تشخیص داده شده به تعداد کل نمونه بات‌ها است. نرخ هشدار نادرست، نسبت تعداد نمونه‌های عادی که به اشتباه بات تشخیص داده شده‌اند، به تعداد کل نمونه‌های عادی است.

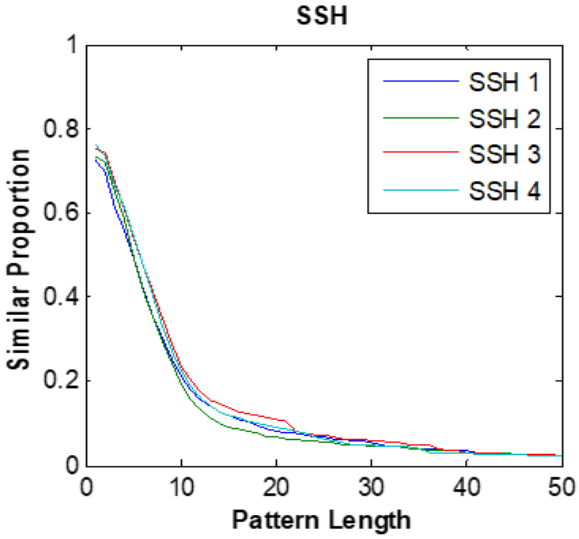
#### ۶-۲-۱. مولفه تشخیص مبتنی‌بر الگوهای مشابه

مولفه تشخیص مبتنی بر الگوهای مشابه، میزبان‌های مشکوک به بات را با توجه به میزان شباهت، شناسایی می‌کند. شکل (۲)، میانگین میزان شباهت هفت الگوی HTTP را با یکدیگر نشان می‌دهد. میانگین به‌دست‌آمده حاصل از انجام آزمایش‌ها به تعداد ۱۰۰ مرتبه، به‌ازای ۱۰۰ حالت مختلف از الگوهای سالم HTTP است. همانگونه که در شکل (۲) دیده می‌شود میزان شباهت الگوهای سالم با یکدیگر، و برای کل الگوها برای طول‌های بیشتر از ۱۰ به سمت صفر میل کرده است. به عبارت دیگر تغییرات در ترافیک HTTP از الگوی خاصی پیروی نمی‌کند.




**شکل (۲):** احتمال شباهت الگوهای سالم HTTP

شکل (۳) میانگین میزان شباهت الگوهای سالم SSH را نشان می‌دهد. همان‌گونه که مشاهده می‌شود میزان الگوهای مشابه میزبان‌ها با یکدیگر در ترافیک SSH در مقایسه با ترافیک HTTP، بیشتر است. دلیل آن وجود تغییرات در رفتار انسان، مثل تغییر در نحوه تایپ کردن و یا سرعت آن است. بنابراین، به دلیل وابستگی ترافیک‌های تعاملی همانند SSH به رفتار انسان [۳۰]، پیروی از الگوهای خاص بیشتر دیده می‌شود.




**شکل (۳):** میانگین میزان شباهت الگوهای سالم SSH

- **آلودگی با یک بات**

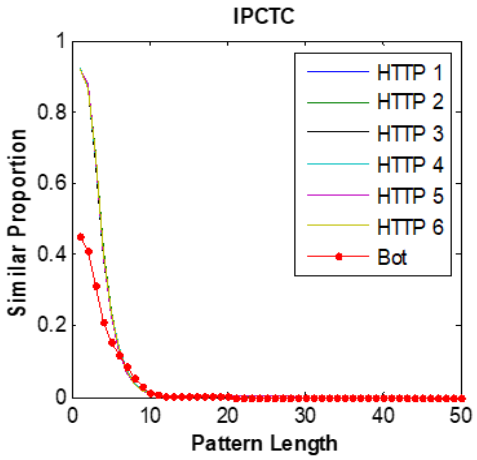
در این مرحله یک میزبان شبکه آلوده می‌شود. شکل (۴- الف) تا (۴- هـ)، میزان شباهت الگوهای یک میزبان آلوده به بات را با الگوهای میزبان‌های سالم در کانال‌های IPCTC، MBCTC، Jitterbug، Mimic، Jitterbug و

[1] Detection Rate
[2] False Alarm Rate

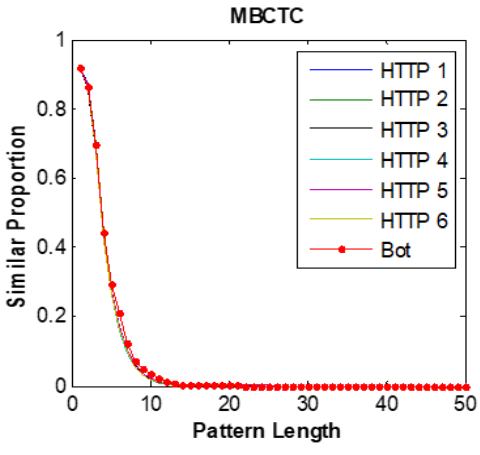
TRCTC نشان می‌دهد. این میزان شباهت برای طول‌های متفاوت، برابر میانگین مقادیر به‌دست‌آمده از ۱۰۰ آزمایش است. با استفاده از این معیار، اثبات وجود یک میزبان آلوده به بات ممکن نیست. در باتنت IPCTC و برای طول‌های کمتر از ۱۰، میزان شباهت با میزبان‌های شبکه، کمتر از ۰,۵ است و دلیل آن استفاده از دو تاخیر زمانی برای ارسال بیت‌های "۰" و "۱" است که باعث شده میزان تغییرات الگوی این بات، کمتر از میزان تغییرات الگوی میزبان‌های سالم باشد.

میزان شباهت بات‌های MBCTC و Mimic در الگوهای کمتر از ۱۰ میزبان به‌دلیل تقلید توزیع ترافیک سالم، به‌طور تقریبی مطابق با میزان تغییرات ترافیک سالم است. بنابراین رفتار این بات‌ها در الگوهای کوتاه، مشابه با رفتار میزبان‌های سالم است.

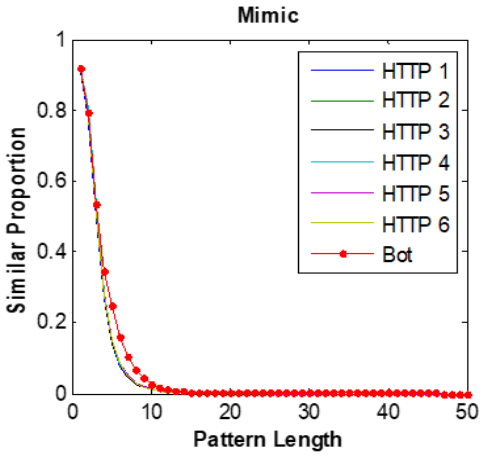
در باتنت TRCTC نیز به دلیل استفاده از دو مجموعه برای ارسال بیت‌های ۰ و ۱، تاخیرهای زمانی، بی‌نظم هستند. این بی‌نظمی باعث می‌شود که در الگوهای با طول‌های کوتاه، الگوهای مشابه کمتری با الگوهای میزبان‌های سالم مشاهده شود.



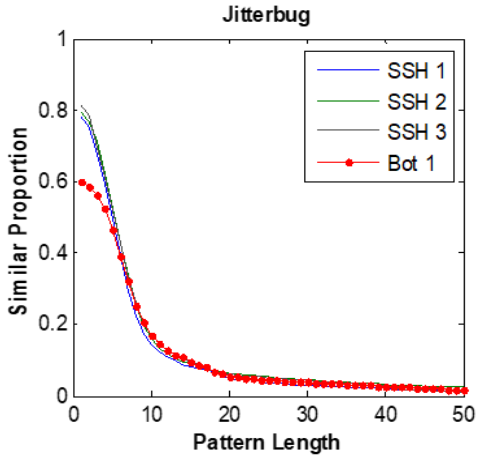

الف



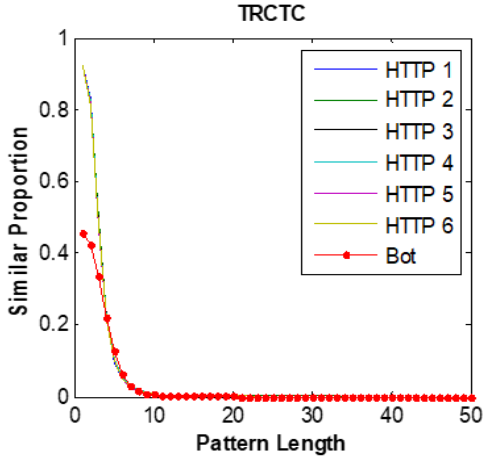

ب




ج




د

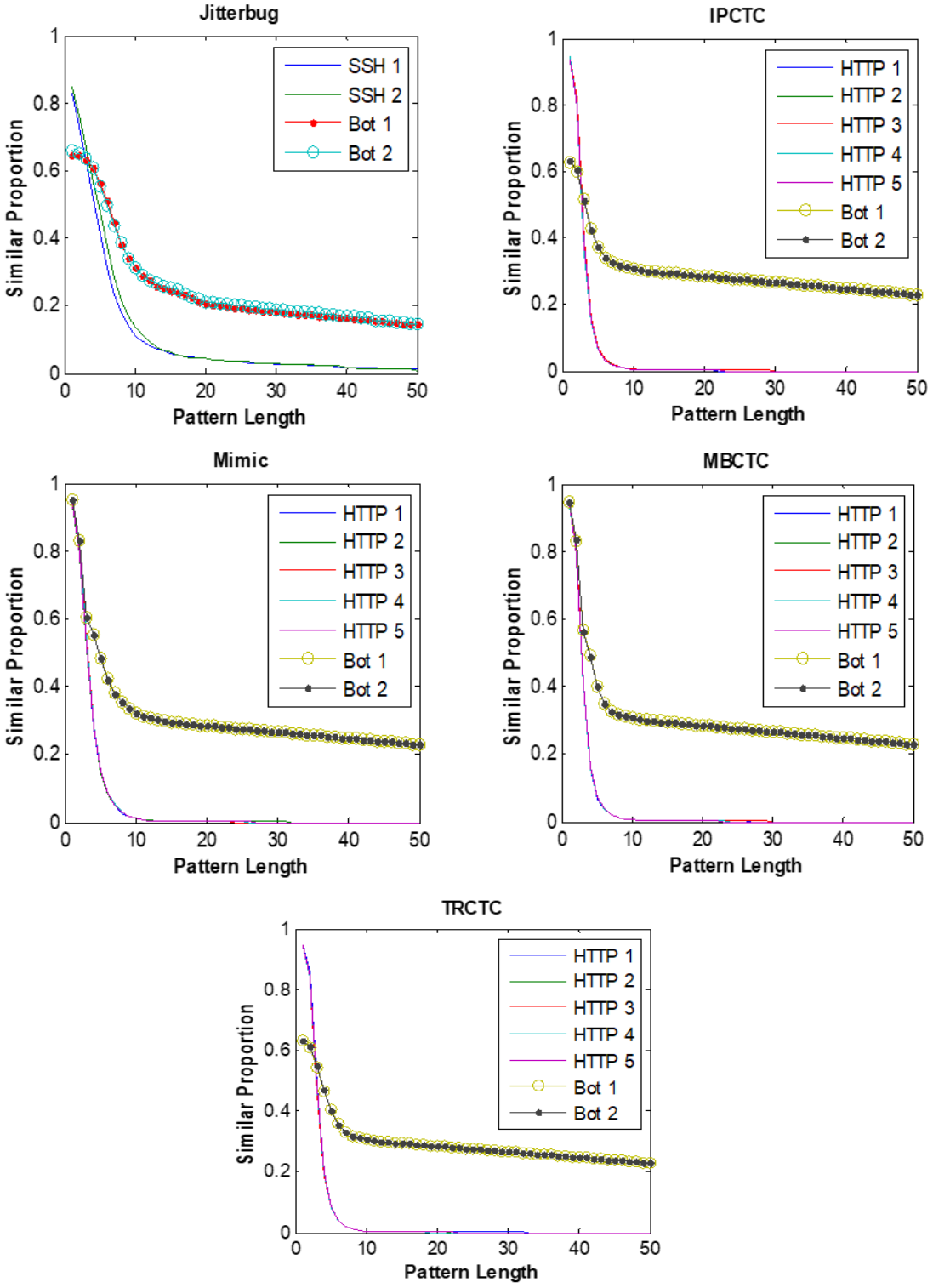



هـ

**شکل (۴):** الف) میزان شباهت الگوهای یک میزبان آلوده به باتنت زمانی IPCTC با الگوهای میزبان‌های سالم ب) میزان شباهت الگوهای یک میزبان آلوده به باتنت زمانی MBCTC با الگوهای میزبان‌های سالم ج) میزان شباهت الگوهای یک میزبان آلوده به باتنت زمانی Mimic با الگوهای میزبان‌های سالم د) میزان شباهت الگوهای یک میزبان آلوده به باتنت زمانی Jitterbug با الگوهای میزبان‌های سالم هـ) میزان شباهت الگوهای یک میزبان آلوده به باتنت زمانی TRCTC با الگوهای میزبان‌های سالم

- **آلودگی با دو بات**

در این مرحله دو میزبان شبکه آلوده شد. شکل (۵) میزان شباهت الگوهای دو میزبان آلوده به بات را با الگوهای میزبان‌های سالم نشان می‌دهد. به دلیل وجود رفتارهای گروهی مشابه، شباهت الگوهای آن‌ها نسبت به میزبان‌های سالم بیشتر است.




**شکل (۵):** میزان شباهت الگوهای دو میزبان آلوده به بات با الگوهای میزبان‌های سالم

- **تشخیص میزبان‌های آلوده از طریق تاثیر طول الگوها بر شباهت آن‌ها**

با افزایش طول الگوها، نرخ شباهت جریان‌های آلوده به بات افزایش می‌یابد. در حالی‌که جریان‌های سالم به دلیل ماهیت غیر ایستایی خود از الگوی خاصی پیروی نمی‌کنند و بنابراین افزایش طول الگوها، تاثیر قابل ملاحظه‌ای بر نرخ شباهت جریان‌های سالم ندارد. به عبارت دیگر به دلیل تغییرات زیاد در ترافیک سالم، احتمال وجود الگوهای بلندتر کم است؛ اما در ترافیک بات‌ها به دلیل رفتارهای مشابه، احتمال وجود الگوها با طول بلندتر بیشتر است. رابطه (۴) این تاثیر را با اختصاص ضریب بیشتر به طول‌های بلندتر، انجام می‌دهد. پس از بررسی‌های انجام شده به ازای طول‌های متفاوت، طول الگو برای محاسبه نرخ شباهت، برابر ۵۰ در نظر گرفته شده است. در صورتی‌که شباهت میزبان‌های مشکوک به بات بیشتر از حد آستانه تعیین شده برای ترافیک سالم باشد؛ برای بررسی بیشتر به مولفه تشخیص مبتنی بر آنتروپی ارسال می‌شوند.

### ۳-۶. مولفه تشخیص مبتنی بر آنتروپی

در این مولفه، آنتروپی شرطی اصلاح‌شده میزبان‌های مشکوک به بات محاسبه می‌شوند. در صورتی که آنتروپی میزبان‌های دریافتی از حد آستانه تعیین شده برای آنتروپی سالم کمتر شود، این میزبان‌ها به‌عنوان میزبان‌های آلوده تشخیص داده می‌شوند.

حد بالای آنتروپی شرطی اصلاح شده، آنتروپی مرتبه اول و حد پایین آن صفر است. هر چه ترافیک سیستم مورد بررسی بی‌نظم‌تر باشد آنتروپی آن بیشتر است و به سمت آنتروپی مرتبه اول میل می‌کند. در مقابل اگر بی‌نظمی کمتری وجود داشته باشد آنتروپی آن نیز کمتر بوده و به سمت صفر میل می‌کند. به همین دلیل، انتظار می‌رود که آنتروپی دو یا چند میزبان آلوده به بات، به علت وجود الگوهای مشابه میان آن‌ها، کمتر از آنتروپی میزبان‌های سالم باشد.

در شکل (۶)، مقدار آنتروپی میزبان‌های سالم HTTP و میزبان‌های آلوده به باتنت‌های IPCTC، MBCTC، TRCTC و Mimic نشان داده شده است. محور افقی در این نمودارها نشان‌دهنده تعداد میزبان‌ها و محور عمودی نشان‌دهنده مقدار آنتروپی شرطی اصلاح‌شده است. نمودار HTTP شکل (۶)، نشان می‌دهد که مقدار آنتروپی با افزایش تعداد میزبان‌ها با شیب کمی کاهش یافته است.

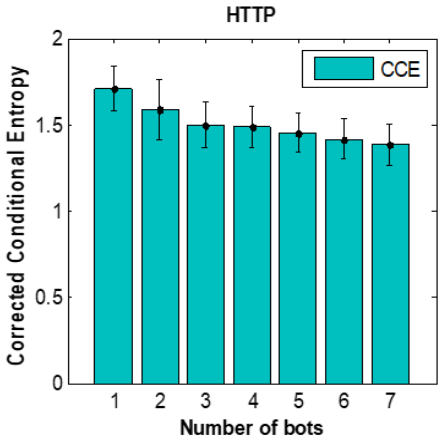
در این نمودارها، خطوط برش، نشان‌دهنده حدود آستانه برای تفکیک میزبان‌های آلوده به بات هستند. نمودار باتنت IPCTC شکل (۶) نشان می‌دهد که آنتروپی میزبان‌های آلوده به بات، به علت وجود الگوهای مشابه، کاهش یافته است. به همین ترتیب، در نمودار باتنت‌های MBCTC، Mimic و TRCTC نیز این تغییر دیده می‌شود.

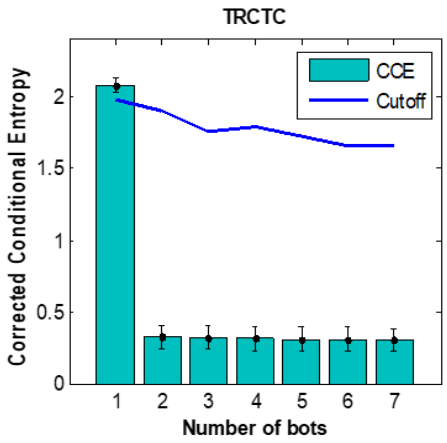
همان‌گونه که در نمودار باتنت Mimic نیز دیده می‌شود مقدار آنتروپی یک میزبان آلوده به باتنت Mimic بین حد آستانه پایین و حد آستانه بالا قرار گرفته است و دلیل آن تقلید بی‌نظمی و شکل موجود در ترافیک سالم است. روش‌های تشخیص پیشین، این کانال را تشخیص نمی‌دهند. در حالی که در رفتار گروهی میزبان‌ها، در صورت وجود بیش از یک میزبان آلوده به بات، ترافیک سالم، تشخیص داده می‌شود.

در شکل (۷) آنتروپی اصلاح شده برای بات‌های مبتنی بر SSH نشان داده شده است. نمودار این شکل نشان می‌دهد که با افزایش تعداد میزبان‌ها، آنتروپی شرطی اصلاح شده تغییر محسوسی نکرده است به نحوی که آنتروپی چهار میزبان بیشتر از آنتروپی محاسبه شده برای یک میزبان است. در مقابل، با افزایش تعداد میزبان‌ها، انحراف معیار آنتروپی کاهش پیدا کرده است. نتایج حاصل از بات Jitterbug در شکل (۸) نشان داده شده است.

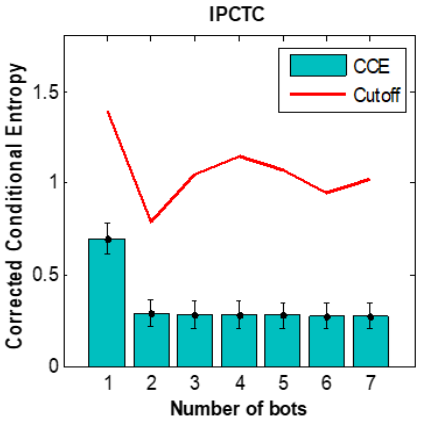
با توجه به این که برای تولید این نوع باتنت، پنجره زمانی ۲۰ میلی ثانیه در نظر گرفته شده است، در صورت یکنواخت بودن پیام، در حدود ۱۰ میلی‌ثانیه به میانگین تاخیرهای زمانی اضافه می‌شود. بنابراین، به دلیل ایجاد تغییر کوچکی در ترافیک سالم برای ارسال پیام‌ها، این نوع باتنت قادر است حداکثر همبستگی موجود در ترافیک سالم را حفظ کند. علاوه بر این، Jitterbug به دلیل این که تاخیرهای زمانی را به مقدار کمی افزایش می‌دهد، تاثیر کمی بر توزیع ترافیک دارد. به همین جهت، Jitterbug توزیع و نظم مشابه با ترافیک سالم دارد، و تشخیص آن مشکل است. همان‌طور که شکل (۸) نیز نشان می‌دهد، با افزایش تعداد میزبان‌های آلوده به بات، مقدار آنتروپی کاهش می‌یابد.

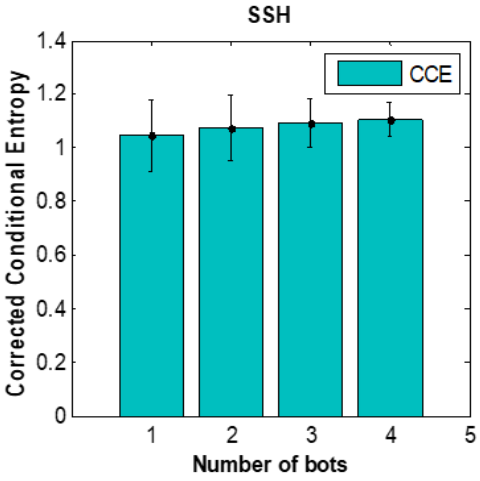
از نتایج به‌دست‌آمده برای میزبان‌های آلوده به بات‌های مبتنی بر HTTP و میزبان‌های آلوده به بات‌های مبتنی‌بر SSH می‌توان نتیجه گرفت که درصورت وجود بیش از یک بات در شبکه، آنتروپی بات‌های مبتنی‌بر HTTP در مقایسه با بات‌های مبتنی‌بر SSH، کاهش بیشتری دارند.



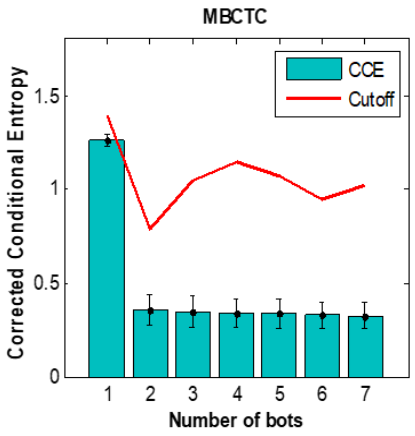



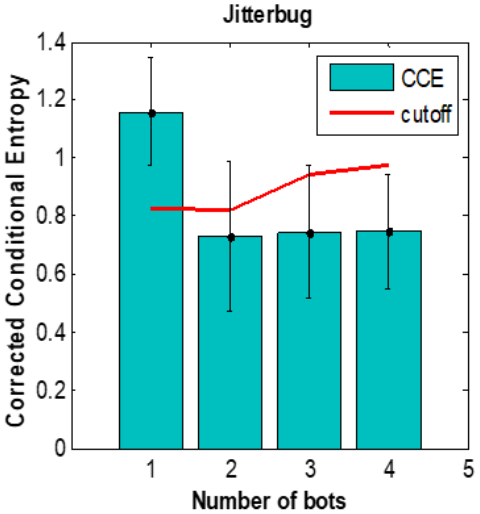

**شکل (۶)**: آنتروپی شرطی اصلاح شده بات‌های مبتنی بر HTTP



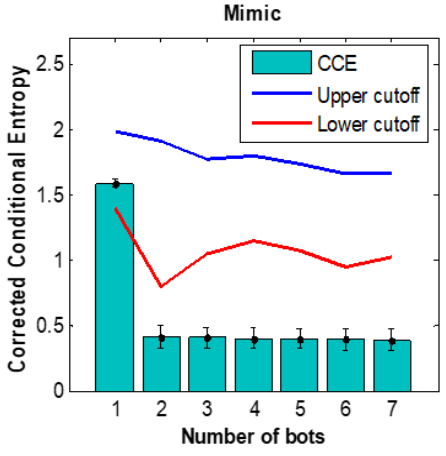




**شکل (۷)**: آنتروپی شرطی اصلاح‌شده بات مبتنی بر SSH




**شکل (۸)**: نتایج حاصل از بات Jitterbug




جدول (۳) نرخ تشخیص و نرخ هشدار نادرست را به‌ازای یک بات، دوبات و هفت بات نشان می‌دهد.

**جدول (۳):** نرخ تشخیص و نرخ هشدار نادرست روش پیشنهادی باتنت زمانی با کانال‌های پنهان زمانی مختلف

| | یک بات | | دو بات | | هفت بات | |
|---|---|---|---|---|---|---|
| | نرخ تشخیص | نرخ هشدار نادرست | نرخ تشخیص | نرخ هشدار نادرست | نرخ تشخیص | نرخ هشدار نادرست |
| **IPCTC** | ۱۰۰ | ۱/۶۰ | ۱۰۰ | ۱/۰۷ | ۱۰۰ | ۱/۰۲ |
| **MBCTC** | ۸۵ | ۲/۰۳ | ۱۰۰ | ۱/۰۹ | ۱۰۰ | ۱/۰۲ |
| **TRCTC** | ۹۴ | ۲/۰۱ | ۱۰۰ | ۱/۰۹ | ۱۰۰ | ۱/۰۲ |
| **Mimic** | ۰ | _ | ۱۰۰ | ۱/۰۹ | ۱۰۰ | ۱,۰۴ |
| | | | | | چهار بات | |
| **Jitterbug** | ۸ | _ | ۹۲ | ۱/۰۹ | ۱۰۰ | ۱/۰۴ |

## ۷. نتیجه‌گیری

باتنت‌ها ضمن این‌که به یک ناهنجاری در فرآیند تبادل اطلاعات و آسیب‌رساندن به منابع شبکه تبدیل شده‌اند دارای روش‌های تشخیص متنوعی هستند. اصلی‌ترین جزء یک باتنت، کانال فرمان و کنترل آن است که مدیریات توسط این کانال، فرمان‌های خود را روی ماشین قربانی ارسال و اجرا می‌کند. کانال پنهان فرمان و کنترل مفهومی است که باتنت‌های نسل جدید برای مخفی‌سازی ارتباط خود به‌کار می‌برند. در این مقاله ضمن استفاده از یک کانال پنهان زمانی موجود؛ با استفاده از مفهوم فعالیت گروهی که در باتنت‌ها وجود دارد روشی برای تشخیص آن‌ها ارایه شد و مورد بررسی و آزمایش قرار گرفت. در این روش، یک معماری سه لایه‌ای با مولفه‌های تشخیص دو مرحله‌ای شامل ماتریس شباهت و آنتروپی ارایه شد. از طریق شبیه‌سازی پنج کانال زمانی موجود، فرمان‌های مدیریات به یک شبکه مفروض ارسال شد. نتایج آزمایش‌ها نرخ تشخیص بالا را حتی در صورت وجود دو بات در شبکه نشان داد. بنابراین رفتار گروهی میزبان‌های آلوده به بات، موجب تشخیص میزبان‌های آلوده حتی برای کانالی مثل Mimic شد که با روش‌های موجود تشخیص داده نمی‌شوند.

## ۸. مراجع


[1] H. R. Zeidanloo, A. B. Manaf, P. Vahdani, F. Tabatabaei, and M. Zamani, "Botnet detection based on traffic monitoring," In Proceedings of the 2010 International Conference on Networking and Information Technology, pp. 97-101, 2010.

[2] C. Li, W. Jiang, and X. Zou, "Botnet: Survey and Case Study," In Proceedings of the 4th International Conference on Innovative Computing, Information and Control, 2009.

[3] P. Bacher, T. Holz, M. Kotter, and G. Wicherski, "Know Your Enemy: Tracking Botnets (using honeynets to learn more about bots)," Technical Report, The Honeynet Project, 2008.

[4] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han, "Botnet research survey," In Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC'08), pp. 967–972, 2008.

[5] R. Jalaei and M. R. Hasani Ahangar, "An Analytical Survey on Botnet and Detection Methods," Journal of Electronical & Cyber Defence, vol. 4, no. 4, 2017. (In Persian)

[6] B. W. Lampson, "A note on the confinement problem," Communication of the ACM, vol. 16, no. 10, pp. 613–615, 1973.

[7] L. Qiu, Y. Zhang, F. Wang, M. Kyung, and H. R. Mahajan, "Trusted computer system evaluation criteria," In Proceedings of the National Computer Security Center, 1985.

[8] C. Serdar, "Network covert channels: design, analysis, Detection and elimination," Ph.D. dissertation, Purdue University, 2006.

[9] C. E. Shannon, "A note on the concept of entropy," Bell system technical journal, vol. 27, pp. 379–423, 1948.

[10] A. Porta, G. Baselli, D. Liberati, N. Montano, C. Cogliati, T. Gnecchi-Ruscone, A. Malliani, and S. Cerutti, "Measuring regularity by means of a corrected conditional entropy in sympathetic outflow," Biological Cybernetics, vol. 78, no. 1, pp. 71–78, 1998.

[11] R. Moddemeijer, "On estimation of entropy and mutual information of continuous distributions," Signal Processing, vol. 16, no. 3, pp. 233–248, 1989.

[12] S. Gianvecchio and H. Wang, "An entropy-based approach to detecting covert timing Channels," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 6, pp. 785–797, 2011.

[13] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M.Salles, "Botnets: A Survey," Computer networks, Elsevier, vol. 57, no. 2, pp. 378-403, 2012.

[14] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: clustering analysis of network traffic for protocol-and structure-independent botnet detection," In Proceedings of the 17th Conference on Security Symposium, USENIX Association, pp. 139– 154, 2008.

[15] G. Gu, J. Zhang, and W. Lee, "BotSniffer: detecting botnet command and control channels in network traffic," In Proceedings of the 15th Annual Network & Distributed System Security Symposium, The Internet Society (ISOC), 2008.

[16] E. Middelesch, "Anonymous and hidden communication channels: A perspective on future developments," Master Thesis, University of Twente, 2015.

[17] "Channels: a perspective on future developments," M.S. thesis, University of Twente, 2015.

[18] J. Nazario and T. Holz, "As the net churns: fast-flux botnet observations," In Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE), pp. 24–31, 2008.

[19] A. Caglayan, M. Toothaker, D. Drapaeau, D. Burke, and G. Eaton, "Behavioral analysis of fast flux service networks," In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, CSIIRW'09, ACM, vol. 48, pp. 1-4, 2009.

[20] R. Sharifnya and M. Abadi, "A novel reputation system to detect DGA-based botnets," In Proceedings of the ICCKE 2013, Mashhad, pp. 417-423, 2013. (In Persian)

[21] I. Ghafir, V. Prenosil, M. Hammoudeh, T. Baker, S. Jabbar, S. Khalid, and S. Jaf, "BotDet: A system for real time botnet command and control traffic detection," IEEE Access, vol. 4, pp. 2169-3536, 2018.

[22] C. j. Dietrich, C. Rossow, F. C. Freiling, H. Bos, M. V. Steen, and N. Pohlmann, "On Botnets That Use DNS for Command and Control," In Proceedings of the 7th European Conference on Computer Network Defense, pp. 9-16, IEEE Computer Society, 2011.

[23] J. Nazario, "Twitter-Based Botnet Command Channel," 2009. [Online]. Availible: http://asert.arbornetworks.com/2009/08/twitter-based-botnet-command channel.

[24] S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, and N. Borisov, "Stegobot: A covert social network botnet," In Proceedings of the 13th International Conference on Information Hiding, pp. 299 –313, 2011.

[25] A. Sanatinia and G. Noubir, "Onionbots: Subverting privacy infrastructure for cyber attacks," in Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 69-80, 2015.

[26] T. J. Richer, "Entropy-based detection of botnet command and control," In Proceedings of the Australasian Computer Science Week Multiconference, ACSW '17, ACM, p. 75, 2017.

[27] S. Cabuk, C. E. Brodley, and C. Shields, "IP Covert Timing Channels : Design and Detection," In Proceedings of the 11th ACM conference on Computer and communications security, pp. 178–187, 2004.

[28] S. Cabuk, "Network covert channels: Design, analysis, detection, and elimination," Ph.D dissertation, Purdue University, West Lafayette, USA, 2006.

[29] S. Gianvecchio, H. Wang, D. Wijesekera, and S. Jajodia, "Model-Based Covert Timing Channels: Automated Modeling and Evasion," In Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection, pp. 211–230, 2008.

[30] K. Kothari and M. Wright, "Mimic: An active covert channel that evades regularity-based detection," Computer Networks, vol. 57, no. 3, pp. 647–657, Feb. 2013.

[31] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," In Proceedings of the 2006 USENIX Security Symposium, July–August 2006.

[32] DARPA, "Intrusion Detection Evaluation Data Set," 1999. [Online]. Avalible: https://www.ll.mit.edu/ideval/data/1999data.html.

Journal of Electronical & Cyber Defence
Vol. 7, No. 4, 2020, Serial No. 28



# Detecting Botnets with Timing-Based Covert Command and Control Channels

**R. Jalaei, M. R. Hasani Ahangar***

*Imam Hossein Comprehensive University

(Received: 23/10/2018, Accepted: 05/03/2019)

**ABSTRACT**

*Nowadays, botnets have become an inconsistency in the process of exchanging information and tampering network resources. Botnet detection methods have always faced challenges and have been investigated and promoted as subjects of research. The main characteristics of botnets is the command and control (C&C) channel through which a botmaster sends malicious commands to the victim's system. By detecting the C&C channel of a botnet, the botnet is not essentially able to communicate with the botmaster and loses its efficiency. For this reason, botmasters try to evade detection by using a variety of methods. Covert command and control channel is a concept that the new generation of botnets use to hide their communications. In this paper, a Botnet is proposed, in which botmaster's commands are sent by using Inter Packet Delays (IPDs) and their sequences. The commands are sent via a timing-based covert command and control channel. In the following, a detection method is proposed by applying the concept of group activity of bots. A three-layer architecture is proposed which consists of traffic data collection and processing, pattern processing, and two-step detection methods. Using the two-step detection method including similarity matrix and entropy, hosts infected with the bot are detected. To evaluate the method, five covert timing channels are simulated and each of them is used to send botmaster commands. The results of the experiments showed the effectiveness of the detection method with the minimum number of two bots in the network.*

**Keywords:** Botnet, Covert channel, Covert timming channel, Similarity matrix, Entropy, Corrected conditional entropy

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