هشتمین کنفرانس بین‌المللی انجمن رمز ایران
دانشگاه فردوسی مشهد - ۲۳ و ۲۴ شهریور ۱۳۹۰

# روشی برای تشخیص بات‌نت‌ها در مرحله فرمان و کنترل با استفاده از خوشه‌بندی برخط

موسی یحیی‌زاده، مهدی آبادی

گروه مهندسی کامپیوتر، دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس، تهران
{m.yahyazadeh, abadi}@modares.ac.ir

## چکیده

**امروزه بات‌نت‌ها به عنوان یکی از مهمترین تهدیدات امنیتی در مقابل زیرساخت اینترنت محسوب می‌شوند. برای تشخیص بات‌نت‌ها روش‌های مختلفی پیشنهاد شده است که اغلب آن‌ها به صورت غیربرخط عمل کرده و یا بات‌نت‌ها را در مرحله حمله از چرخه حیات آن‌ها تشخیص می‌دهند. در این مقاله، یک روش کلی، به نام OBD، برای تشخیص برخط بات‌نت‌ها در مرحله فرمان و کنترل پیشنهاد می‌شود. در این روش، در پایان هر دوره زمانی، ابتدا مجموعه‌ای از بردارهای جریان از ترافیک شبکه استخراج می‌شود. سپس این بردارهای جریان با استفاده از یک الگوریتم خوشه‌بندی با شعاع ثابت برخط، به نام OFWC، به تعدادی خوشه تقسیم شده و خوشه‌هایی که معیار شباهت درون خوشه‌ای آن‌ها از یک آستانه شباهت بیشتر باشد، به عنوان خوشه‌های حاوی ترافیک بات‌نت شناسایی می‌شوند. نتایج ارزیابی روش OBD در یک شبکه آلوده به بات‌نت HTTP نشان می‌دهد که این روش از نرخ تشخیص بالا و نرخ هشدار نادرست پایین برای تشخیص بات‌نت‌ها در مرحله فرمان و کنترل برخوردار است.**

## کلمات کلیدی

**تشخیص بات‌نت، چرخه حیات بات‌نت، مرحله فرمان و کنترل، خوشه‌بندی برخط، جریان دنباله‌ای.**

## ۱- مقدمه

امروزه حملات اینترنتی با انگیزه‌های مختلفی از قبیل کسب شهرت و درآمد، سرگرمی، خراب‌کاری و غیره انجام می‌شوند. تحقیقات نشان داده که بدافزارهای اینترنتی مهم‌ترین عوامل حملات در فضای اینترنت هستند که در سال‌های اخیر به سمت سازماندهی بهتر و سودمحوری بیشتر رشد کرده‌اند [۱]. در مرکز بیشتر این حملات، یک گروه از میزبان‌هایی قرار دارند که به تصرف[۱] مهاجم درآمده و توسط وی از راه دور هدایت می‌شوند. این گروه از میزبان‌ها یک بات‌نت[۲] را تشکیل می‌دهند.

کلمه بات[۳] که از روبات برگرفته شده با نام زامبی نیز شناخته می‌شود. مشابه روبات‌ها، از بات‌ها برای انجام عملیات از پیش تعریف شده‌ای استفاده می‌شود که به صورت خودکار اجرا می‌شوند [۲]. بات‌نت نیز به معنی شبکه‌ای از بات‌ها (به عبارت دیگر ارتش زامبی[۴]) می‌باشد. چرخه حیات هر بات‌نت شامل سه مرحله شکل‌گیری، فرمان و کنترل، و حمله است. در مرحله شکل‌گیری، مهاجم با تصرف تعداد زیادی از میزبان‌های آسیب‌پذیر و نصب یک برنامه مخرب، آن میزبان‌ها را به بات تبدیل کرده و شبکه‌ای از بات‌ها را برای خود راه‌اندازی می‌کند. در مرحله فرمان و کنترل، با ارسال فرامین بات‌ها را از راه دور هدایت کرده و در مرحله حمله، انواع مختلفی از حملات را به صورت هماهنگ و با قدرت تخریبی بسیار بالا بر روی قربانی سازماندهی می‌کند. بر اساس داده‌های ثبت شده در ظروف عسل، این حملات شامل جلوگیری از سرویس توزیع شده[۵]، ارسال هرزنامه[۶]، گسترش بدافزار، نشت اطلاعات[۷]، کلاهبرداری در تعداد کلیک‌ها[۸] و سرقت هویت[۹] می‌باشند [۳].

تاکنون روش‌های متعددی برای تشخیص بات‌نت‌ها پیشنهاد شده که اغلب آن‌ها تنها قادر به تشخیص نوع خاصی از یک بات‌نت بوده و به صورت غیربرخط عمل می‌کنند. در این مقاله، یک روش کلی برای تشخیص بات‌نت‌ها در مرحله فرمان و کنترل پیشنهاد می‌شود که به صورت برخط عمل می‌کند. بدین صورت که در پایان هر دوره زمانی، مجموعه‌ای از بردارهای جریان از ترافیک شبکه تحت نظارت استخراج می‌شود. سپس این بردارهای جریان با استفاده از یک الگوریتم خوشه‌بندی با شعاع ثابت برخط به تعدادی خوشه تقسیم می‌شوند. در نهایت، خوشه‌هایی که معیار شباهت درون خوشه‌ای

آن‌ها از یک آستانه شباهت بیشتر باشد، به عنوان خوشه‌های حاوی ترافیک باتنت علامت زده شده و خوشه‌هایی که معیار حذف آن‌ها از یک آستانه حذف بیشتر باشد، از مجموعه خوشه‌ها حذف می‌شوند. روش پیشنهادی دارای مزایای متعددی است:

۱) از نرخ تشخیص بالا و نرخ هشدار نادرست پایین برخوردار است.
۲) قبل از انجام هر گونه فعالیت بدخواهانه توسط باتهای عضو یک باتنت قادر است آن باتنت را در مرحله فرمان وکنترل تشخیص دهد.
۳) به صورت برخط عمل می‌کند.
۴) به فضای زیادی برای ذخیره‌سازی جریان‌های ترافیک شبکه نیاز ندارد.

در ادامه، در بخش ۲ به کارهای مرتبط اشاره شده و در بخش ۳ مرحله فرمان و کنترل از چرخه حیات باتنت‌ها معرفی می‌گردد. در بخش ۴ روش OBD برای تشخیص برخط باتنت‌ها در مرحله فرمان و کنترل شرح داده می‌شود. در بخش ۵ نتایج آزمایش‌های انجام شده برای ارزیابی کارآیی روش OBD ارائه می‌شود و در نهایت در بخش ۶ نتیجه‌گیری به عمل می‌آید.

## ۲- کارهای مرتبط

برای تشخیص باتنت‌ها روش‌های مختلفی پیشنهاد شده است که در ادامه به برخی از آن‌ها اشاره می‌شود:

Choi و همکاران [۴] یک روش تشخیص باتنت مبتنی بر ناهنجاری ارائه کرده‌اند که با نظارت بر فعالیت‌های گروهی در ترافیک DNS، باتنت‌ها را در مراحل مختلف از چرخه حیات آن‌ها تشخیص می‌دهد. این فعالیت گروهی بر اساس پرس‌وجوهای DNS ارسال شده به صورت هم‌زمان توسط باتهای توزیع شده شکل می‌گیرد. در این روش، از ویژگی‌های متمایزکننده بین پرس‌وجوهای DNS قانونی و پرس‌وجوهای DNS باتنت برای تشخیص استفاده می‌شود. اگر باتهای عضو یک باتنت تنها در مرحله شکل‌گیری از پرس‌وجوهای DNS استفاده کرده و یا از آدرس‌های IP به‌جای نام‌های دامنه استفاده کنند، روش فوق قادر به تشخیص آن باتنت نخواهد بود.

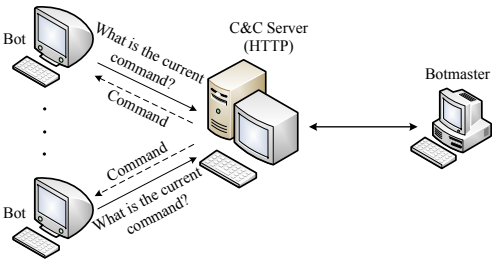
Gu و همکاران [۵] یک روش مبتنی بر خوشه‌بندی برای تشخیص باتنت‌ها در مرحله حمله ارائه کرده‌اند. در این روش، ابتدا ترافیک ارتباطی مشابه و ترافیک بدخواهانه مشابه خوشه‌بندی شده و سپس یک همبستگی بین خوشه‌ای[۱۰] انجام می‌شود تا میزبان‌های دارای هر دو الگوی ارتباطی مشابه و الگوی فعالیت بدخواهانه مشابه شناسایی شوند. روش فوق به صورت غیربرخط عمل می‌کند که در سیستم‌های تشخیص باتنت یک ضعف عمده به‌شمار می‌آید. همچنین، در صورتی که باتهای عضو یک باتنت در مرحله حمله فعالیت بدخواهانه جدیدی را انجام دهند، این روش قادر به تشخیص آن باتنت نخواهد بود.

Xiaocong و همکاران [۶] روشی پیشنهاد کرده‌اند که در آن از تحلیل خوشه‌بندی وفق‌پذیر نسبت به داده، برای تشخیص برخط باتنت‌های متمرکز در مرحله فرمان و کنترل استفاده می‌شود. در این روش، ابتدا جریان‌های ترافیک شبکه به دنباله‌هایی از ویژگی‌ها تبدیل می‌شوند. سپس خوشه‌بندی وفق‌پذیر نسبت به داده بر روی آن‌ها اعمال شده تا خوشه‌هایی از دنباله‌های ویژگی با مشابهت بالا تولید شوند. خوشه‌ها تنها در صورت تغییر عمده جریان‌های عضو آن‌ها به‌روزرسانی می‌شوند. در صورتی‌که جریان‌ها در یک خوشه از شباهت بالایی نسبت به هم برخوردار باشند و میزبان‌های تولیدکننده آن‌ها به یک نقطه مرکزی متصل شوند، این میزبان‌ها آلوده به بات تشخیص داده می‌شوند. عیب این روش این است که تنها قادر به تشخیص باتنت‌های متمرکز است و خوشه‌بندی ارائه شده فقط ویژگی‌های عددی را در بر می‌گیرد. همچنین، در این روش مدت زمان نگهداری جریان‌ها در هر خوشه به‌صورت شفاف بیان نشده و چنین به نظر می‌رسد که فضای ذخیره‌سازی به‌صورت نامحدود در نظر گرفته شده است. بنابراین از کارآیی زیادی برای تشخیص برخط باتنت‌ها برخوردار نیست.

## ۳- فرمان و کنترل باتنت‌ها

تفاوت اصلی بین باتنت‌ها و سایر انواع بدافزارها در وجود زیرساخت فرمان و کنترل (C&C[۱۱]) برای آن‌ها است، به‌طوری که مدیر بات با ارسال فرامین از طریق این زیرساخت، باتنت خود را به صورت یکپارچه هدایت می‌کند [۷].

مدیر بات یک میزبان (که معمولاً کامپیوتری با پهنای باند بالا است) را به عنوان نقطه مرکزی (سرویس‌دهنده C&C) برای همه باتها انتخاب می‌کند. سپس بر روی این میزبان، سرویس‌های شبکه‌ای خاصی (از قبیل IRC و یا HTTP) را برای برقراری ارتباط با باتها اجرا کرده و از این طریق باتنت خود را هدایت می‌کند. شکل ۱ نمونه‌ای از یک کانال فرمان و کنترل متمرکز را نمایش می‌دهد که با استفاده از یک سرویس‌دهنده وب ایجاد شده است.




**شکل (۱) : کانال فرمان و کنترل متمرکز HTTP**

همان طور که در این شکل ملاحظه می‌شود، مدیر بات فرامین مورد نظر خود را در یک سرویس‌دهنده وب قرار داده و میزبان‌های آلوده به بات با سرکشی دوره‌ای به این سرویس‌دهنده، فرامین جاری را واکشی کرده و آن‌ها را اجرا می‌کنند. این روش یکی از متداول‌ترین روش‌های مورد استفاده توسط مدیران بات است، زیرا آن‌ها را قادر می‌سازد تا ترافیک فرمان و کنترل خود را در ترافیک عادی HTTP میزبان‌های آلوده به بات مخفی کنند.

## ٤- تشخیص برخط باتنت‌ها در مرحله فرمان و کنترل

در این بخش، روشی به نام OBD برای تشخیص برخط باتنت‌ها در مرحله فرمان وکنترل (C&C) پیشنهاد می‌شود. هدف از این روش، تشخیص گروهی از میزبان‌های آلوده به بات است که به عنوان بخشی از یک باتنت محسوب می‌شوند. در ادامه، ابتدا به بیان مسأله و فرضیات آن پرداخته شده و سپس مراحل روش OBD شرح داده می‌شوند.

### ٤-۱- بیان مسأله و فرضیات

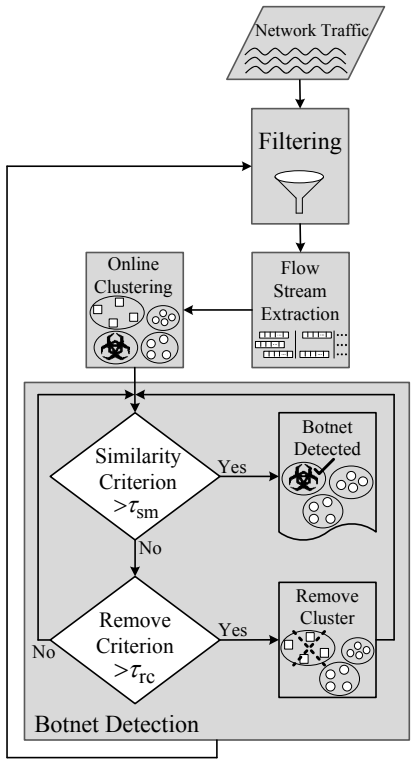
روش OBD برای تشخیص باتنت‌هایی است که در تعریف این مقاله از باتنت قرار می‌گیرند.

**تعریف ۱ – باتنت.** هر باتنت یک گروه هماهنگ از باتهایی است که از طریق کانال‌های فرمان و کنترل هدایت شده و فعالیت‌های بدخواهانه‌ای را انجام می‌دهند [۵].

گروه هماهنگ از باتها بدین معنی است که چندین بات (حداقل دو بات) داخل یک باتنت، از الگوی ارتباطی فرمان و کنترل مشابه با هم پیروی می‌کنند. در صورتی که مدیر بات فرامین متفاوتی را به صورت جداگانه برای هر بات ارسال کند، این باتها در تعریف این مقاله از باتنت قرار نمی‌گیرند. از طرفی یک باتنت زمانی برای مدیر بات موثر است که شامل چندین بات بوده و امکان هدایت آن‌ها به‌صورت یکپارچه وجود داشته باشد.

## ٤-٢- مراحل روش OBD

روش OBD شامل چهار مرحله اصلی فیلترسازی ترافیک، استخراج جریان‌های دنباله‌ای، خوشه‌بندی برخط و تشخیص باتنت است (شکل ۲).




**شکل (۲) : مراحل روش OBD**

در مرحله فیلترسازی ترافیک، با استفاده از یک فهرست سفید[۱۲] شامل میزبان‌های مورد اعتماد (به عنوان مثال سرویس‌دهندگان موتورهای جستجو از قبیل گوگل و یاهو) ترافیک شبکه پالایش می‌شود. بدین منظور، ابتدا ترافیک شبکه ضبط می‌شود. سپس بسته‌ها به/از سمت میزبان‌های موجود در فهرست سفید از این ترافیک حذف شده و سایر بسته‌های باقی‌مانده به عنوان ترافیک ورودی به مرحله بعدی داده می‌شوند. مزیت استفاده از فهرست سفید کاهش میزان محاسبات و فضای مورد نیاز برای ذخیره‌سازی ترافیک شبکه تحت نظارت می‌باشد. همچنین، با استفاده از فهرست سفید نرخ هشدار نادرست روش پیشنهادی کاهش می‌یابد، به دلیل این که دو میزبان در شبکه تحت نظارت ممکن است به طور همزمان درخواست‌های مشابهی را به سمت یک میزبان مورد اعتماد ارسال کنند و این رفتار به عنوان رفتار گروهی باتها تشخیص داده شود. در مرحله استخراج جریان‌های دنباله‌ای، در پایان هر دوره زمانی ترافیک دریافتی از مرحله قبل پردازش شده و مجموعه‌ای از بردارهای جریان از این ترافیک استخراج می‌شود. سپس این بردارهای جریان در مرحله خوشه‌بندی برخط با استفاده از الگوریتم خوشه‌بندی با شعاع ثابت برخط (OFWC) به تعدادی خوشه تقسیم می‌شوند. در مرحله تشخیص باتنت، خوشه‌هایی که دارای حداقل دو عضو بوده و معیار شباهت درون خوشه‌ای آن‌ها از یک آستانه شباهت $\tau_{sm}$ بیشتر باشد، به‌عنوان خوشه‌های حاوی ترافیک باتنت علامت زده می‌شوند. همچنین، در این مرحله خوشه‌هایی که معیار حذف آن‌ها از یک آستانه حذف $\tau_{rc}$ بیشتر باشد، از مجموعه خوشه‌ها حذف می‌شوند. در شکل ۳ شبه‌کد روش OBD نمایش داده شده است.

**procedure** OBD

**input**:
  $\sigma$ : cluster radius
  $\tau_{sm}$ : similarity threshold
  $\tau_{rc}$ : remove threshold
**begin**
  **for** each time period $t$ **do**
    Extract a set $S(t)$ of flow vectors from $S$
    $C(t)=ofwc(S(t),C(t-1),\sigma)$
    **for** each cluster $c_i \in C(t)$ **do**
      Calculate similarity criterion $sm(c_i)$ using equation (9)
      **if** ($|c_i|>2$ **and** $sm(c_i)>\tau_{sm}$) **then**
        Mark flow vectors in cluster $c_i$ as suspicious
        Alert "*Bot Detected!*"
      **end if**
      Calculate remove criterion $rc(c_i)$ using equation (12)
      **if** $rc(c_i)>\tau_{rc}$ **then**
        $C(t) := C(t)-\{c_i\}$
      **end if**
    **end for**
  **end for**
**end procedure**

**شکل (۳) : شبه‌کد روش تشخیص برخط باتنت OBD**

## ٤-٣- الگوریتم OFWC

روش OBD[۱۳] بر پایه تحلیل ترافیک شبکه به صورت جریان[۱۴] می باشد. جریان، به مجموعه‌ای از بسته‌ها اطلاق می‌شود که آدرس IP مبدا، درگاه مبدا، آدرس IP مقصد، درگاه مقصد و پروتکل آن‌ها یکسان است. روش‌هایی که به صورت غیربرخط عمل می‌کنند جریان‌ها را پس از اتمام آن‌ها در یک مجموعه داده تجمیع کرده و سپس آن‌ها را تحلیل می‌کنند. اما در روش OBD که به صورت برخط عمل می‌کند، جریان ترافیک به صورت دنباله‌ای در نظر گرفته می‌شود.

**تعریف ۲ – جریان دنباله‌ای.** جریان دنباله‌ای $\vec{F}_i$ یک توالی از بردارهای ویژگی استخراج شده برای هر جریان $i$ در دوره‌های زمانی متفاوت با طول یکسان می‌باشد:

$$\vec{F}_i = \langle f_i(t_1), f_i(t_2), f_i(t_3), \ldots, f_i(t_j), \ldots \rangle \qquad (۱)$$

که $f_i(t_j)$ بردار ویژگی استخراج شده برای جریان $i$ در پایان دوره زمانی $t_j$ است و بردار جریان $i$ در این دوره زمانی نامیده می‌شود:

$$f_i(t_j) = (x_1, x_2, x_3, ..., x_p) \quad (۲)$$

که $p$ تعداد ویژگی‌ها و $x_k$ ویژگی $k$ام از جریان $i$ در پایان دوره زمانی $t_j$ است و به صورت $f_i^k(t_j)$ نیز نمایش داده می‌شود.

**تعریف ۳ – مجموعه جریان.** مجموعه همه جریان‌های دنباله‌ای موجود در ترافیک شبکه مجموعه جریان نامتناهی نامیده شده و با $S$ نمایش داده می‌شود:

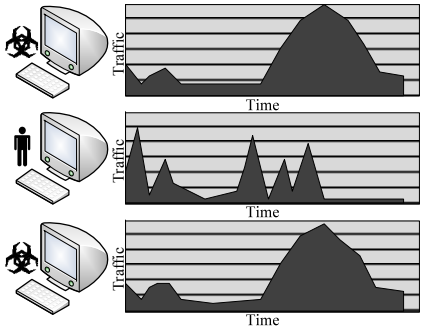
$$S = \{..., \vec{F}_{i-1}, \vec{F}_i, \vec{F}_{i+1}, ...\} \quad (۳)$$

مجموعه همه بردارهای جریان در دوره زمانی $t$ با $S(t) \subseteq S$ نمایش داده می‌شود:

$$S(t) = \{f_i(t), f_{i+1}(t), f_{i+2}(t), ..., f_{i+n}(t)\} \quad (۴)$$

با توجه به این که مقادیر ویژگی‌های هر جریان دائماً در حال تغییر است، خوشه‌بندی جریان‌ها به صورت برخط امکان پذیر نیست. بنابراین با در نظر گرفتن هر جریان به صورت دنباله‌ای، مقادیر ویژگی‌های آن جریان در پایان هر دوره زمانی محاسبه شده و با یک بردار جریان نمایش داده می‌شود.

همان طور که اشاره شد، میزبان‌های آلوده به بات در یک باتنت به خاطر از پیش کد شدن، رفتار ترافیکی مشابهی را در دوره‌های زمانی نزدیک به هم از خود نشان می‌دهند. در مقابل، رفتار ترافیکی سایر میزبان‌هایی که توسط انسان‌ها کنترل می‌شوند این‌گونه نیست (شکل۴). بنابراین، در صورت خوشه‌بندی جریان‌های ترافیکی میزبان‌های شبکه، میزبان‌های آلوده به بات در یک خوشه یکسان قرار می‌گیرند.




**شکل (٤) : مقایسه رفتار ترافیکی میزبان‌های آلوده به بات با میزبان‌های غیرآلوده**

میزان تشابه دو بردار جریان نسبت به هم با استفاده از معیاری به نام فاصله جریان سنجیده می‌شود.

**تعریف ٤ – فاصله جریان.** فاصله دو بردار جریان $f_i(t)$ و $f_j(t)$ در دوره زمانی $t$ با محاسبه اختلاف بین ویژگی‌های متناظر آن‌ها حاصل می‌شود:

$$d(f_i(t), f_j(t)) = \frac{\sum_{k=1}^{p} \delta_{i,j}^k \, \Delta(f_i^k(t), f_j^k(t))}{\sum_{k=1}^{p} \delta_{i,j}^k} \quad (۵)$$

که $\delta_{i,j}^k$ یک نشانگر است و به صورت زیر تعریف می‌شود:

$$\delta_{i,j}^k = \begin{cases} 0 & \text{if } f_i^k(t) \text{ or } f_j^k(t) \text{ is missed} \\ 1 & \text{otherwise} \end{cases} \quad (۶)$$

$\Delta(f_i^k(t), f_j^k(t))$ اختلاف بین ویژگی $k$ام در بردارهای جریان $f_i(t)$ و $f_j(t)$ است. در صورتی که این ویژگی اسمی باشد، $\Delta(f_i^k(t), f_j^k(t))$ با استفاده از رابطه (۷) محاسبه می‌شود:

$$\Delta(f_i^k(t), f_j^k(t)) = \begin{cases} 0 & \text{if } f_i^k(t) = f_j^k(t) \\ 1 & \text{otherwise} \end{cases} \quad (۷)$$

و در غیر این صورت با استفاده از رابطه (۸) محاسبه می‌گردد:

$$\Delta(f_i^k(t), f_j^k(t)) = \frac{|f_i^k(t) - f_j^k(t)|}{f_{\max}^k(t-1) - f_{\min}^k(t-1)} \quad (۸)$$

که $f_{\min}^k(t-1)$ و $f_{\max}^k(t-1)$ به ترتیب کمترین و بیشترین مقدار ویژگی $k$ام مجموعه بردارهای جریان در دوره زمانی قبلی است. دلیل استفاده از این مقادیر این است که اختلاف ویژگی‌ها با مقادیر بزرگتر بر اختلاف ویژگی‌ها با مقادیر کوچکتر غلبه نکنند.

**procedure** OFWC

**input**:
  $S(t)$ : Set of flow vectors
  $\sigma$ : Cluster radius
  $C(t-1)$ : Set of clusters
**output**:
  $C(t)$ : Set of clusters
**begin**
  **for** $k = 1$ to $p$ **do**
    Calculate $f_{\max}^k(t-1)$ and $f_{\min}^k(t-1)$
  **end for**
  **for** each flow vector $f_i(t) \in S(t)$ **do**
    **if** $f_i(t-1) \in c_j$ for some $c_j \in C(t-1)$ **then**
      **if** $d(f_i(t), \mu_j) < \sigma$ **then**
        $c_j := (c_j - \{f_i(t-1)\}) \cup f_i(t)$
        Update centroid $\mu_j$
      **else**
        $c_j := c_j - \{f_i(t-1)\}$
        Update centroid $\mu_j$
      **end if**
    **end if**
    **if** $f_i(t) \notin c_j$ for all $c_j \in C(t-1)$ **then**
      Find the nearest cluster $c_l \in C(t-1)$ to $f_i(t)$
      **if** $d(f_i(t), \mu_l) < \sigma$ **then**
        $c_l := c_l \cup \{f_i(t)\}$
      **else**
        Make a new cluster $c_k$ with centroid $\mu_k$
        $C(t-1) := C(t-1) \cup \{c_k\}$
        $\mu_k := \{f_i(t)\}$
        $\beta_k := t$
      **end if**
    **end if**
  **end for**
  $C(t) := C(t-1)$
**end procedure**

**شکل (٥) : شبه‌کد الگوریتم خوشه‌بندی با شعاع ثابت برخط OFWC**

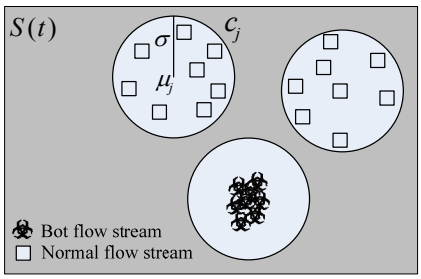
**تعریف ٥ – خوشه.** هر خوشه شامل مجموعه‌ای از بردارهای جریان است که فاصله آن‌ها تا مرکز خوشه از یک شعاع ثابت $\sigma$ کمتر است. هر خوشه $c_j$ با دوتایی $(\mu_j, \beta_j)$ نمایش داده می‌شود که $\mu_j$ مرکز و $\beta_j$ دوره

زمانی تولد خوشه است. $\mu_j$ یک بردار ویژگی $p$ بُعدی است. اگر ویژگی $\mu_j^k \in \mu_j$ اسمی باشد، مقدار آن از مقدار ویژگی $k$ام با بیشترین تکرار و در غیر این صورت از میانگین مقادیر این ویژگی در بردارهای جریان عضو خوشه $c_j$ حاصل می‌شود. مجموعه خوشه‌ها در هر دوره زمانی $t$ با $C(t)$ نمایش داده می‌شود.

در الگوریتم خوشه‌بندی با شعاع ثابت برخط OFWC[۱۵]، ابتدا مجموعه $S(t)$ از بردارهای جریان در دوره زمانی جاری $t$ و مجموعه $C(t-1)$ از خوشه‌ها در دوره زمانی قبلی به عنوان ورودی دریافت می‌شود. سپس بردارهای جریان در هر خوشه $c_j \in C(t-1)$ با مقادیر جدید آن‌ها در $S(t)$ به‌روزرسانی می‌شود. در صورتی که فاصله یک بردار جریان عضو خوشه $c_j$ تا مرکز این خوشه از شعاع $\sigma$ کوچکتر باشد، مرکز $c_j$ به‌روزرسانی می‌شود. در غیر این صورت، این بردار جریان از خوشه $c_j$ حذف شده و به نزدیکترین خوشه‌ای که فاصله تا مرکز آن خوشه از شعاع $\sigma$ کمتر باشد اضافه می‌شود. شبه‌کد الگوریتم OFWC در شکل ۵ نمایش داده شده است.

## ٤-٤- شباهت درون خوشه‌ای

همان طور که اشاره شد، پس از خوشه‌بندی جریان‌های دنباله‌ای مشابه درون خوشه‌های یکسان قرار می‌گیرند. مهمترین تفاوت بین خوشه‌های حاوی جریان‌های دنباله‌ای تولید شده توسط باتها و سایر خوشه‌ها در میزان تراکم آن‌ها است. باتهای عضو یک باتنت به دلیل از پیش‌کد شدن دارای الگوی ارتباطی مشابهی برای دریافت فرامین هستند. بنابراین جریان‌های دنباله‌ای تولید شده توسط این باتها به میزان زیادی مشابه هم بوده و خوشه حاوی این جریان‌های دنباله‌ای از تراکم بالایی برخوردار است (شکل ۶).




**شکل (٦) : خوشه‌های حاوی جریان‌های دنباله‌ای تولید شده توسط باتها و جریان‌های دنباله‌ای عادی**

در روش OBD برای پیدا کردن خوشه‌های حاوی جریان‌های دنباله‌ای باتها از معیاری به نام شباهت درون خوشه‌ای استفاده می‌شود.

**تعریف ٦ – معیار شباهت درون خوشه‌ای.** فرض کنید خوشه $c_j$ شامل $m$ بردار جریان و $\mu_j$ مرکز این خوشه باشد. معیار شباهت درون خوشه‌ای برای $c_j$ با استفاده از رابطه (۹) محاسبه می‌شود:

$$sm(c_j) = e^{-\frac{\bar{d}_j}{1+o_j}} \qquad (۹)$$

که $\bar{d}_j$ متوسط فاصله همه بردارهای جریان عضو خوشه $c_j$ تا $\mu_j$ است و متوسط فاصله درون خوشه‌ای نامیده می‌شود:

$$\bar{d}_j = \frac{1}{m}\sum_{i=1}^{m} d(f_i(t), \mu_j) \qquad (۱۰)$$

و $o_j$ طول عمر خوشه $c_j$ است:

$$o_j = t - \beta_j \qquad (۱۱)$$

که $t$ دوره زمانی جاری و $\beta_j$ دوره زمانی تولد خوشه $c_j$ است.

## ٤-٥- حذف خوشه‌ها

با ورود جریان‌های دنباله‌ای جدید به شبکه در طول دوره‌های زمانی متفاوت، تعداد خوشه‌ها به صورت خطی افزایش می‌یابد که این مسأله منجر به افزایش فضای مورد نیاز برای ذخیره‌سازی و از همه مهم‌تر افزایش محاسبات می‌شود. از آن جا که روش OBD به صورت برخط عمل می‌کند، با افزایش تعداد خوشه‌ها، تعداد مقایسه‌های انجام شده برای هر بردار جریان افزایش می‌یابد. بنابراین، در روش OBD برای حل این مشکل در پایان هر دوره زمانی تعدادی از خوشه‌ها انتخاب شده و از مجموعه خوشه‌ها حذف می‌شوند. برای انتخاب این خوشه‌ها از معیار حذف استفاده می‌شود.

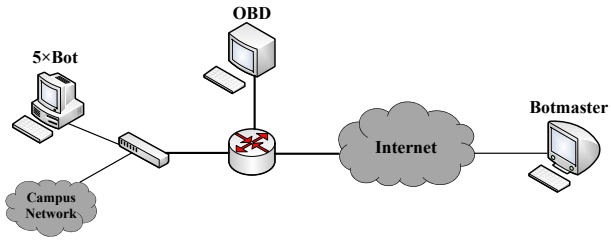
**تعریف ۷ – معیار حذف خوشه‌ها.** فرض کنید $\bar{d}_j$ متوسط فاصله درون خوشه‌ای و $o_j$ طول عمر خوشه $c_j$ باشد. معیار حذف برای $c_j$ با استفاده از رابطه (۱۲) محاسبه می‌شود:

$$rc(c_j) = \bar{d}_j \cdot o_j \qquad (۱۲)$$

از آن جا که جریان‌های دنباله‌ای تولید شده توسط باتها در یک بازه زمانی کوتاه به طور هماهنگ ایجاد شده و از شباهت زیادی نسبت به هم برخوردار هستند، لذا نگهداری خوشه‌ها با متوسط فاصله درون خوشه‌ای زیاد به مدت طولانی قابل قبول نیست. از طرفی با در نظر گرفتن طول عمر $o_j$ در معیار حذف، فرصت کافی به هر خوشه $c_j$ داده می‌شود تا چنانچه حاوی جریان‌های دنباله‌ای ناتمام از باتها باشد، از مجموعه خوشه‌های $C$ حذف نشده و در دوره‌های زمانی بعدی شباهت درون خوشه‌ای خود را افزایش دهد.

## ٥- نتایج آزمایش‌ها

در آزمایش‌های انجام شده کارآیی روش OBD با استفاده از ترافیک شبکه محلی دانشگاه مورد ارزیابی قرار گرفت. این ترافیک شامل انواع مختلفی از پروتکل‌ها از قبیل HTTP، SMTP و FTP بود. برای جمع‌آوری جریان‌های موجود در این ترافیک از نرم‌افزار متن‌باز Argus [۸] استفاده شد. در پایان هر دوره زمانی، برای هر جریان یک بردار جریان شامل هفت ویژگی آدرس IP مبدا، شماره درگاه مبدا، آدرس IP مقصد، شماره درگاه مقصد، پروتکل، تعداد کل بایت‌ها و تعداد کل بسته‌های منتقل شده استخراج شد. شکل ۷ نمایی از بستر شبکه مورد استفاده در آزمایش‌ها را نمایش می‌دهد.



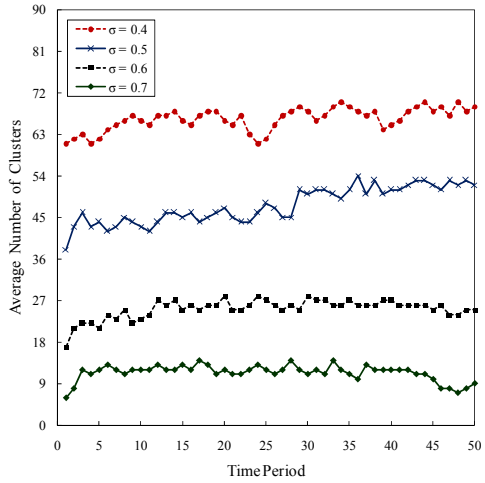

**شکل (۷) : بستر شبکه مورد استفاده برای ارزیابی کارآیی روش OBD**

در آزمایش‌ها از پنج میزبان آلوده به بات HTTP استفاده شد که از دوره زمانی پنجم فعالیت خود را آغاز کردند. در جدول ۱ بهترین تنظیمات برای پارامترهای مورد استفاده در روش OBD نمایش داده شده است.

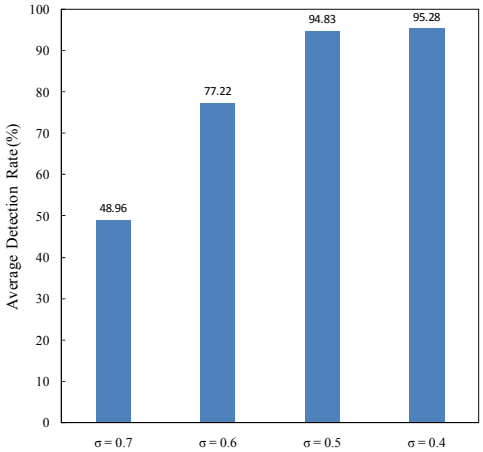
**جدول (۱) : پارامترهای تنظیم شده**

| | |
|---|---|
| Time period ( $t$ ) | 5(s) |
| Fixed width cluster radius ( $\sigma$ ) | 0.5 |
| Similarity threshold ( $\tau_{sm}$ ) | 0.95 |
| Remove threshold ( $\tau_{rc}$ ) | 10 |

در مرحله فیلترسازی ترافیک، از یک فهرست سفید شامل ۱۰۰ سایت برتر گزارش شده توسط Alexa [۹] استفاده شد. برای ارزیابی تاثیر هر یک از پارامترها بر روی کارآیی روش OBD هر آزمایش ده بار تکرار شد. در شکل‌های ۸ و ۹ به ترتیب تاثیر مقادیر مختلف شعاع ثابت $\sigma$ بر روی متوسط تعداد خوشه‌های تولید شده در مرحله خوشه‌بندی برخط و متوسط نرخ تشخیص در مرحله تشخیص باتنت نمایش داده شده است.



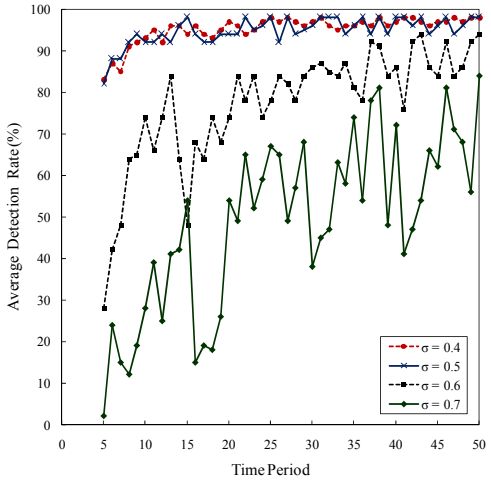

**شکل (۸) : تاثیر مقادیر مختلف شعاع ثابت ($\sigma$) بر روی متوسط تعداد خوشه‌های تولید شده در مرحله خوشه‌بندی برخط**




**شکل (۹) : تاثیر مقادیر مختلف شعاع ثابت ($\sigma$) بر روی متوسط نرخ تشخیص در مرحله تشخیص باتنت**

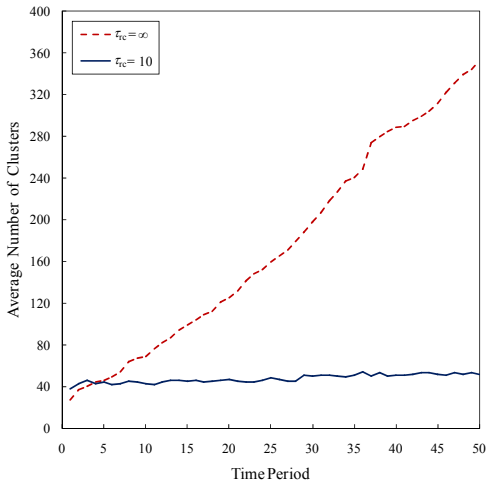
با توجه به شکل‌های فوق مشخص می‌شود که با کاهش مقدار $\sigma$ تعداد کل خوشه‌ها در هر دوره زمانی افزایش می‌یابد که این مسأله منجر به افزایش محاسبات هنگام خوشه‌بندی بردارهای جریان در دوره‌های زمانی بعدی می‌شود. با افزایش مقدار $\sigma$ جریان‌های دنباله‌ای غیربات درون خوشه‌های حاوی جریان‌های دنباله‌ای بات قرار می‌گیرند که این مسأله منجر به کاهش معیار شباهت برای این خوشه‌ها و در نتیجه کاهش نرخ تشخیص در مرحله تشخیص باتنت می‌شود. با انتخاب $\sigma = 0.5$ توازن بهتری میان تعداد خوشه‌های تولیدشده و نرخ تشخیص برقرار می‌شود.

در شکل ۱۰ تاثیر مقادیر مختلف شعاع ثابت $\sigma$ بر روی متوسط نرخ تشخیص به تفکیک دوره‌های زمانی مختلف نمایش داده شده است. با توجه به این شکل مشخص می‌شود که متوسط نرخ تشخیص در دوره‌های زمانی یکسان برای $\sigma = 0.5$ در مقایسه با سایر مقادیر این پارامتر بالاتر بوده و در تعدادی از دوره‌های زمانی به نرخ ۱۰۰ درصد نزدیک می‌شود.




**شکل (۱۰) : تاثیر مقادیر مختلف شعاع ثابت ($\sigma$) بر روی متوسط نرخ تشخیص در مرحله تشخیص باتنت به تفکیک دوره‌های زمانی**

در شکل ۱۱ تاثیر وجود معیار حذف خوشه‌ها بر روی متوسط تعداد خوشه‌های تولید شده در مرحله خوشه‌بندی برخط نمایش داده شده است. با توجه به این شکل مشخص می‌شود که در صورت عدم حذف خوشه‌ها با طول عمر و متوسط فاصله درون خوشه‌ای زیاد متوسط تعداد خوشه‌ها در هر دوره زمانی به صورت خطی افزایش می‌یابد که خود باعث افزایش محاسبات در مراحل خوشه‌بندی برخط و تشخیص باتنت می‌شود.




**شکل (۱۱) : تاثیر وجود معیار حذف خوشه‌ها بر روی متوسط تعداد خوشه‌های تولید شده در مرحله خوشه‌بندی برخط**

## ٦- نتیجه‌گیری

در این مقاله، روشی به نام OBD برای تشخیص برخط باتنت‌ها در مرحله فرمان و کنترل پیشنهاد شد که از رفتار هماهنگ و گروهی بات‌ها در فرآیند تشخیص استفاده می‌کند. این روش، شامل چهار مرحله اصلی فیلترسازی ترافیک، استخراج جریان‌های دنباله‌ای، خوشه‌بندی برخط و تشخیص باتنت است. در مرحله فیلترسازی ترافیک، با استفاده از یک فهرست سفید شامل میزبان‌های مورد اعتماد ترافیک شبکه پالایش شده و در مرحله استخراج جریان‌های دنباله‌ای، از این ترافیک مجموعه‌ای از بردارهای جریان استخراج می‌شود. در مرحله خوشه‌بندی برخط، بردارهای جریان استخراج شده با استفاده از الگوریتم خوشه‌بندی با شعاع ثابت برخط (OFWC) به تعدادی خوشه تقسیم می‌شوند. در مرحله تشخیص باتنت، خوشه‌هایی که دارای حداقل دو عضو بوده و معیار شباهت درون خوشه‌ای آن‌ها از یک آستانه شباهت بیشتر باشد، به‌عنوان خوشه‌های حاوی ترافیک باتنت تشخیص داده می‌شوند. نتایج آزمایش‌های انجام شده نشان می‌دهند که روش OBD از نرخ تشخیص بالا و نرخ هشدار نادرست پایین برای تشخیص باتنت‌ها برخوردار است.

**جدول (۲) : مقایسه روش OBD با سایر روش‌های تشخیص باتنت‌ها**

| روش تشخیص باتنت‌ها | تشخیص باتنت‌های ناشناخته | تشخیص باتنت‌ها با C&C رمزشده | تشخیص برخط باتنت‌ها | نرخ هشدار نادرست پایین |
|---|---|---|---|---|
| BotGAD [4] | ✓ | ✓ | ✓ | ✓ |
| BotMiner [5] | – | ✓ | – | ✓ |
| DataAdaptive [6] | ✓ | – | ✓ | ✓ |
| Rishi [10] | – | – | ✓ | – |
| BotProbe [11] | – | – | ✓ | ✓ |
| BotSniffer [12] | ✓ | ✓ | – | ✓ |
| OBD | ✓ | ✓ | ✓ | ✓ |

روش OBD قادر به تشخیص باتنت‌های ناشناخته است، به دلیل این که هر باتنت را بر اساس رفتار گروهی هماهنگ و مشابه بات‌های عضو آن باتنت تشخیص می‌دهد که این ویژگی در بین انواع باتنت‌ها مشترک است. همچنین، این روش قادر به تشخیص باتنت‌ها با کانال فرمان و کنترل رمز شده است، به‌دلیل این که بردارهای جریان برای هر جریان دنباله‌ای تنها بر اساس سرآیند بسته‌های آن جریان دنباله‌ای استخراج می‌شوند. در جدول ۲ جایگاه روش OBD در مقایسه با سایر روش‌های تشخیص باتنت‌ها نمایش داده شده است.

## سپاسگزاری

این تحقیق با حمایت مالی مرکز تحقیقات مخابرات ایران و تحت قرارداد با کد شناسایی ۹۰-۰۱-۰۴ انجام شده است.

## مراجع

[1] P. Wang, S. Sparks, and C. Zou, "An Advanced Hybrid Peer-to-Peer Botnet", *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 2, pp. 113–127, 2010.
[2] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of Botnet and Botnet Detection", in *Proceedings of the 3rd Conference on Emerging Security Information Systems and Technologies*, Athens, Greece, 2009.
[3] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A Multifaceted Approach to Understanding the Botnet Phenomenon", in *Proceedings of the 6th ACM Internet Measurement Conference,* Rio de Janeriro, Brazil, 2006.
[4] H. Choi, H. Lee, and H. Kim, "BotGAD: Detecting Botnets by Capturing Group Activities in Network Traffic", in *Proceedings of the 4th International ICST Conference on Communication System Software and Middleware*, Dublin, Ireland, 2009.
[5] 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 USENIX Security Symposium*, San Jose, CA, USA, 2008.
[6] Y. Xiaocong, D. Xiaomei, Y. Ge, Q. Yuhai, and Y. Dejun. "Data-Adaptive Clustering Analysis for Online Botnet Detection", in *Proceedings of the 3th IEEE International Joint Conference on Computational Science and Optimization*, Anhui, China, 2010.
[7] H. Zeidanloo and A. Manaf, "Botnet Command and Control Mechanisms", in *Proceedings of the 2nd Conference on Computer and Electrical Engineering*, Dubai, UAE, 2009.
[8] Argus – Auditing Network Activity, http://www.qosient.com/argus
[9] Alexa – The Web Information Company, http://www.alexa.com
[10] J. Goebel and T. Holz, "Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation", in *Proceedings of 1st Workshop on Hot Topics in Understanding Botnets*, Cambridge, MA, USA, 2007.
[11] G. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee, "Active Botnet Probing to Identify Obscure Command and Control Channels", in *Proceedings of the 25th Annual Computer Security Applications Conference*, Honolulu, HI, USA, 2009.
[12] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic", in *Proceedings of the 15th Annual Network and Distributed System Security Symposium*, San Diego, CA, USA, 2008.

## زیرنویس‌ها

[1] Compromised
[2] Botnet
[3] Bot
[4] Zombie army
[5] Distributed Denial of Service (DDoS)
[6] Spamming
[7] Information leakage
[8] Click fraud
[9] Identity thief
[10] Cross-cluster correlation
[11] Command and Control
[12] White listing
[13] Online Botnet Detection (OBD)
[14] Flow
[15] Online Fixed Width Clustering