فصلنامه علمی-ترویجی پدافند غیرعامل
سال هشتم، شماره ۱، بهار ۱۳۹۶، (پیاپی ۲۹): صص ٤٤-۳۵

شناسایی و مقابله با کانال کنترل و فرماندهی در بات‌نت‌های مبتنی بر شبکه‌های اجتماعی

حمید تنها[۱]، مهدی جوانمرد[۲*]

تاریخ دریافت: ۱۳۹۴/۱۰/۰۱
تاریخ پذیرش: ۱۳۹۵/۰۶/۰۸

چکیده

یک بات‌نت، مجموعه‌ای از بات‌ها است که به صورت جداگانه بر روی یک سامانه آلوده شده، اجرا شده و به دستورات ارسالی از طرف واحد کنترل و فرماندهی پاسخ می‌دهند. امروزه بات‌نت‌ها به چالش بزرگی در فضای سایبر تبدیل شده‌اند. گونه‌های جدید این خانواده از بدافزارها باسوء استفاده از شبکه‌های اجتماعی محبوب، از آن‌ها به عنوان کانال کنترل و فرماندهی استفاده می‌کنند. در این شیوه، سامانه‌های حفاظتی معمول نظیر IDSها قادر به شناسایی و مقابله با بات‌نت‌ها نخواهند بود، زیرا ترافیک شبکه تولید شده توسط بات ها همانند ترافیک کاربر سامانه می‌باشد. در این مقاله روش جدیدی برای شناسایی، رهگیری و مقابله با بات‌نت‌های مبتنی بر شبکه‌های اجتماعی ارائه شده است. در این روش، تشخیص بات‌نت بر اساس نظارت بر منابع سامانه انجام می‌شود. روش پیشنهادی پس از رهگیری اقدام به مقابله با بات نت مذکور می‌کند. این مقابله شامل ممانعت از اتصال بات به سرور کنترل و فرماندهی می‌باشد که روش پیشنهادی می تواند نزدیک به ۹۶ درصد وجود بات‌نت ها را تشخیص و با موفقت ۱۰۰ درصد سامانه را از وجود بات‌نت‌های کشف شده، تمیز دهد.

کلیدواژه‌ها: تشخیص بات نت، فرماندهی و کنترل، شبکه‌های اجتماعی، امنیت

۱- دانشجوی کارشناسی ارشد، دانشگاه پیام نور
۲- استادیار، دانشگاه پیام نور مرکزی، (javanmard@pnu.ac.ir)- نویسنده مسئول

۱- مقدمه

از دیر باز محبوبیت سیستم عامل ویندوز که در این مقاله به اختصار سیستم عامل ذکر می‌شود در بین کاربران سیستم‌های رایانه‌ای چه کاربران شخصی و چه کاربران اداری بسیار چشمگیر بوده است. این امر سبب شده تا سیستم عامل به عنوان هدف اصلی برنامه‌نویسان قرارگیرد. در این راستا برنامه‌نویسان بد خواه نیز عمده حملات خود را معطوف به این سیستم عامل نموده‌اند. از جمله خطرناک‌ترین حملات بات‌نت‌ها می‌باشند.

بات‌نت‌ها (شبکه‌ای از قربانیان) را به عنوان یکی از مخاطرات مهم امنیتی امروز می دانند. این‌گونه از بدافزارها در جهت حملات انکار سرویس توزیع شده، سرقت کلیک[1]، فیشینگ[2]، توزیع بدافزار، دست‌کاری نظرسنجی های آنلاین و بازی‌ها، سرقت هویت، هرزنامه و ... مورد استفاده قرار می‌گیرند [۱]. بات‌نت‌ها به دلیل استفاده از کانال‌های کنترل و فرماندهی با سایر گونه‌های بدافزار نظیر کرم‌ها متفاوت هستند [۲]. یک مدیر بات فعالیت بات‌نت را با استفاده از دستوراتی که از طریق کانال کنترل و فرماندهی منتقل می‌شود هدایت می‌کند. سه نوع کانال کنترل و فرماندهی وجود دارد: کانال‌های کنترل و فرماندهی مبتنی بر IRC[3]، کانال‌های کنترل و فرماندهی مبتنی بر HTTP و کانال‌های کنترل و فرماندهی مبتنی بر P2P[4][۳]. در این مقاله به بات‌نت‌های مبتنی بر شبکه‌های اجتماعی که بخشی از کانال مبتنی بر HTTP می‌باشد خواهیم پرداخت. از دیدگاه مهاجم، استفاده از شبکه‌های اجتماعی به‌عنوان کانال کنترل و فرماندهی مزایایی دارد: برخلاف معماری نمونه‌های قدیمی بات‌نت، مهاجم نیاز به دسترسی یا نصب یک سرور ندارد. صفحات شبکه‌های اجتماعی به‌سادگی ایجاد می‌شوند و به‌راحتی قابل‌دسترس هستند؛ اما مزیت عمده استفاده از شبکه‌های اجتماعی در دشواری تفاوت قائل شدن بین فعالیت‌های قانونی و فعالیت‌های بات‌نت می‌باشد [۴]. درصورتی‌که یک بات‌نت از الگوهای ارتباطی یک کاربر عادی در مشاهده صفحات شبکه‌های اجتماعی تقلید کند، شناسایی از طریق روش‌های سنتی IDS شبکه بسیار مشکل خواهد بود زیرا از آدرس، نام دامنه، پروتکل و یا درگاه غیرعادی استفاده نمی‌کند و بخش بزرگی از ترافیک سیستم‌های کامپیوتری متعلق به بازدید از شبکه‌های اجتماعی است. در ادامه این مقاله به شناسایی بات‌نت‌ها و ترافیک کانال کنترل و فرماندهی مبتنی بر شبکه‌های اجتماعی از طریق بررسی ترافیک شبکه، ورودی سیستم از سمت کاربر و ... خواهیم پرداخت. در نوشتار پیش رو که به سه بخش تقسیم گردیده، ابتدا در بخش ۲ مفاهیم پایه در خصوص شناسایی سرورهای کنترل و فرماندهی مبتنی بر شبکه‌های اجتماعی تحت عنوان «پیش‌زمینه» معرفی می‌گردد. در فصل ۳ با عنوان «کارهای پیشین» فعالیت‌های مشابه صورت گرفته را معرفی خواهیم نمود. درفصل ۴ تحت عنوان «سیستم پیشنهادی» الگوریتم و شیوه ترکیب پارامترهای مورد نظر جهت دست‌یابی به پاسخ مناسب معرفی شده و در پایان به جمع‌بندی و نتیجه‌گیری خواهیم پرداخت.

۲- پیش زمینه

در این بخش، مفاهیم پایه در زمینه تشخیص بات‌نت و روش پیشنهادی را تشریح می‌کنیم

۲-۱- بات‌نت

واژه بات مشتق شده از واژه روبات می‌باشد و اشاره به برنامه‌ای دارد که می‌تواند تا حدودی به شیوه مستقل عمل کند. یک سامانه‌کامپیوتری که از راه دور توسط مهاجم کنترل شود، بات یا دست نشانده[5] نامیده می‌شود[۵].

همانند گونه‌های قدیمی ویروس‌ها و کرم‌ها، یک بات، نرم‌افزاری خود انتشار است که در حین گسترش خود، اقدام به آلوده‌سازی میزبان‌های آسیب‌پذیر می‌نماید. شیوه‌های آلوده‌سازی بات همانند سایر گونه‌های بدافزار می‌باشد که با بهره‌گیری از آسیب‌پذیری‌های نرم‌افزار اقدام به در خدمت گرفتن سامانه‌های آسیب‌پذیر می‌کنند [۶] .

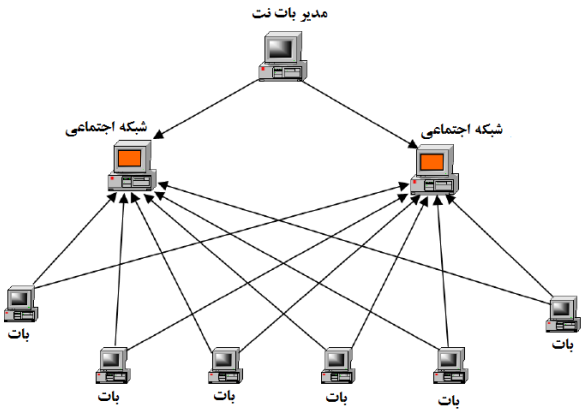
بات‌نت شبکه‌ای از ماشین‌های آلوده شده به بدافزار (بات‌ها) است که توسط یک موجودیت به نام مدیر بات‌نت کنترل می‌شود [۷]. اغلب بیش از یک مدیر بات‌نت کنترل بات‌نت را در دست دارند. این کار استفاده از بات‌نت را بیشتر می‌کند و ردیابی آن را سخت‌تر می‌کند [۸].

۲-۲- سرور کنترل و فرماندهی

یک سرور کنترل و فرماندهی، رایانه‌ای متمرکز است که وظیفه ارسال دستورات به بات‌های موجود در یک بات‌نت و دریافت پاسخ از آنها را بر عهده دارد.

در تصویر شماره ۱ معماری کلی یک بات‌نت (به صورت خاص بات‌نت مبتنی برشبکه اجتماعی) نمایش داده شده است.

1- Key Hijacking
2- Fishing
3-Internet Relay Chat
4- Peer to Peer
5- Zombie



شکل ۱- معماری بات‌نت مبتنی بر شبکه‌های اجتماعی [۹]

۲-۳- چرخه حیات بات‌نت

یک بات‌نت را می‌توان در پنج مرحله ایجاد و نگهداری کرد که شامل: آلوده‌سازی اولیه، تزریق ثانویه، اتصال، کنترل و فرماندهی، به‌روز رسانی و نگهداری می‌باشد.

در خلال فاز آلوده‌سازی اولیه، مهاجم جهت یافتن آسیب‌پذیری زیرشبکه هدف را اسکن می‌کند و ماشین‌های هدف را از طریق روش‌های گوناگون آلوده می‌کند. پس از آلوده‌سازی اولیه، در فاز تزریق ثانویه، میزبان آلوده شده اسکریپتی را تحت عنوان shell-code اجرا می‌کند. Shell-code فایل اجرایی اصلی بات را از مکانی مشخص از طریق FTP[1]، HTTP[2] ، یا P2P[3] دریافت می‌کند. فایل اجرایی اصلی خود را بر روی ماشین هدف نصب می‌کند. با نصب فایل اجرایی اصلی، سیستم هدف با عنوان دست‌نشانده شناخته می‌شود و کدهای بدخواهانه را اجرا می‌کند. برنامه بات در هر بار Boot شدن زامبی به صورت خودکار اجرا می‌شود. در فاز اتصال، بات اقدام به برقرای یک کانال کنترل و فرماندهی و اتصال زامبی به سرور کنترل و فرماندهی می‌نماید. به محض برقرای کانال کنترل و فرماندهی، زامبی بخشی از ارتش بات نت مهاجم خواهد بود. پس از فاز اتصال، فعالیت‌های اصلی کنترل و فرماندهی شروع خواهد شد. مدیر بات‌نت از کانال کنترل و فرماندهی جهت انتشار دستورات به ارتش باتها استفاده می‌کند. برنامه بات دستورات ارسال‌شده از مدیر بات‌نت را دریافت و اجرا می‌کند. کانال کنترل و فرماندهی مهاجم را قادر می‌سازد تا فعالیت شمار زیادی از باتها را جهت انجام دادن اعمال غیرقانونی از راه دور کنترل نماید. فاز آخر زنده نگه داشتن و بروزرسانی می‌باشد. در این فاز به بات دستور داده می‌شود که فایل اجرایی اصلی را به‌روزرسانی نماید. کنترل کنندگان بات‌نت ممکن است به دلایل مختلفی نیازمند به‌روزرسانی بات‌نت خود باشند. به عنوان مثال آنها ممکن است جهت مقابله با تکنیک‌های شناسایی اقدام به به‌روزرسانی فایل اجرایی اصلی نمایند. و یا ممکن است با هدف اضافه نمودن قابلیت‌های جدید به ارتش باتها خواهان به‌روزرسانی باشند. اغلب به‌روزرسانی فایل اجرایی بات با هدف اتصال باتها به سرور کنترل و فرماندهی جدید صورت می‌گیرد. این عمل مهاجرت سرور نامیده می‌شود و برای زنده نگه داشتن بات‌نتها از اهمیت بالایی برخوردار است [۶].

۲-۴- قلاب‌اندازی[4] و انواع آن

قلاب‌اندازی یک مفهوم استفاده شده برای به‌دست آوردن کنترل جریان اجرایی برنامه بدون تغییر و هم گردانی[5] مجدد کد منبع آن است [۱۰]. این کار توسط متوقف‌سازی فراخوانی تابع و هدایت مجدد آن به کد سفارشی شده به‌دست می‌آید. با ارائه کد سفارشی، هر عملیاتی را می‌توان اجرا نمود. پس از آن قابلیت‌های اصلی تابع می‌تواند اجرا شده و در نتیجه می‌تواند یا به سادگی برگشت داده شود یا تغییر داده شده و برای انتقال کنترل به کدی که تابع قلاب شده را فراخوانی کرده، برگشت داده شود. بنابراین قلاب‌سازی یک ابزار مناسب و کامل برای شناسایی و تحلیل کدهای بدخواه به صورت پویا فراهم می‌کند. قلاب‌های درون خط[6] به طور مستقیم، بایت‌های کد تابع را در حافظه بازنویسی می‌کنند. به طور خاص، تنها چند دستورالعمل با یک دستورالعمل پنج بایتی پرش (jmp) به تابع قلاب جایگزین می‌شود. دستورالعمل‌های جایگزین شده در تابع trampoline که به عنوان نقطه ورودی جدید به فراخوانی API اصلی استفاده می‌شود، ذخیره می‌شوند. در تابع قلاب، ثبت اطلاعات و اصلاح آرگومان‌ها قبل از اجرای API اصلی می‌تواند انجام شود. این نوع از قلاب‌اندازی مستلزم تغییر در فایل‌های سیستم عامل است. بنابراین، پیاده‌سازی آن بسیار سخت بوده و گاهی سبب ناسازگاری در سیستم عامل می‌شود.

روش دیگر در قلاب اندازی، تغییر آدرس وقفه‌های سامانه‌ای است. اما پیاده‌سازی این روش بسیار سخت و در بسیاری از موارد غیر ممکن است. زیرا به ندرت اطلاعاتی در خصوص وقفه‌های سامانه‌ای منتشر شده است و نمی‌توان دریافت که یک وقفه با یک کد خاص برای چه رویدادی تعریف شده است. از آنجا که جدول توصیف‌گر این وقفه‌ها[7] پایین‌ترین سطح در هسته سیستم‌عامل قرار دارد، تعداد آنها نیز زیاد است. شناسایی آنها با راه‌کارهایی نظیر اشکال‌زدایی هسته سامانه توسط ابزارهایی مثل Windbg هم در عمل غیرممکن است.

1- File Transfer Protocol
2- Hypertext Transfer Protocol
3- Peer-to-peer

4- Hooking
5- Compile
6- Inline
7- Interrupt Descriptor Table

روش دیگری که بسیاری از بدافزارهای برای قلاب‌اندازی استفاده می‌کنند، تغییر آدرس توابع سامانه‌ای[1] است. این آدرس‌ها در جدولی به‌نام جدولتوصیف‌گر توابع سامانه‌ای[2] (SSDT) و جدول توصیف‌گر توابع سامانه‌ای سایه[3] ذخیره می‌شوند. بنابراین، یکی از مناسب‌ترین نقاط برای قلاب‌اندازی جدول توصیف‌گر توابع سامانه‌ای و جدول توصیف‌گر توابع سامانه‌ای سایه می‌باشد. این دو جدول مشخص کننده آدرس توابع سامانه‌ای سطح هسته در حافظه سامانه است. بنابراین، تمام درخواست‌های گذار از سطح کاربر، رسیدن به سطح هسته نیاز به یافتن آدرس توابع مورد نظرشان در این جدول‌ها هستند [۱۱].

تفاوت دو جدول توصیف‌گر توابع سامانه‌ای و جدول توصیف‌گر توابع سامانه‌ایسایه در این است که جدول توصیف‌گر توابع سامانه‌ای نگاهدارنده آدرس توابع سامانه‌ای برای کار با هسته سیستم عامل و جدول توصیف‌گر توابع سامانه‌ای سایه حاوی آدرس توابع سامانه‌ای مرتبط با عملیات گرافیکی و پنجره‌های سیستم عامل است [۱۲].

در این پژوهش با قلاب‌اندازی به این جدول‌ها و جایگزینی آدرس توابع موجود در آن‌ها با آدرس توابع مورد نظر که خود از پیش مهیا کرده‌ایم کنترل را به‌دست گرفته و درخواست‌های بدخواهانه را از بین می‌بریم.

۲-۵- فیلتر درایور[4]

سیستم‌عامل ویندوز به صورت پیش فرض یک درایور برای مدیریت صفحه کلید و موشواره نصب می‌کند. تمام صفحه کلیدها و موشواره‌ها با درایور استاندارد سیستم عامل سازگار هستند. برخی از صفحه کلیدها و موشواره‌ها که دارای دکمه‌های اضافی برای عملکردهای خاص منظوره هستند که درایور استاندارد سیستم عامل پاسخگوی امکانات اضافی آن‌ها نیست، ناچار هستند برای استفاده از امکانات اضافی درایور خاص خودشان را نصب کنند. اما این درایورها چگونه عمل می‌کنند؟ بسیاری از آن‌ها فیلتر درایورهایی هستند که با قرار گرفتن در بالای درایور اصلی صفحه کلید یا موشواره در فضای حافظه مدیر ورودی/خروجی سیستم عامل با استفاده از امکانات سکوی فیلترینگ سیستم عامل اقدام به ضبط کردن بسته‌های ارسال شده به سمت دستگاه سخت‌افزار و تفسیر آنها می‌کنند [۱۳].

ما نیز با علم به این موضوع و وجود چنین امکانی در سیستم عامل شروع به نوشتن فیلتر درایورهایی برای مدیریت صفحه کلید و موشواره می‌کنیم این فیلتر درایورها با قرار گرفتن بالاتر از درایور استاندارد سامانه در فضای حافظه مدیر ورودی/خروجی سیستم‌عامل برای صفحه کلید یا موشواره تمام درخواست‌های دریافت شده از سمت کاربر ار ضبط می‌کنند.

در این پژوهش استفاده از فیلتر درایورها با هدف تشخیص زمان و کلیدهای فشرده شده صفحه کلید و موشواره صورت می‌گیرد.

۲-۶- اجرای خودکار[5]

یک نام عمومی برای مکان‌هایی در سیستم عامل ویندوز می‌باشد که به برنامه‌ها و سرویس‌هایی که خود را در این مکان‌ها ثبت کرده‌اند اجازه می‌دهد پس از هر بار بارگذاری سیستم عامل، به صورت خودکار اجرا شوند [۱۴]. این سرویس با هدف اجرای خودکار خدمات حیاتی نظیر بارگذاری کتابخانه های اتصال پویا ، شروع خودکار درایورها، احراز هویت و ... فراهم گردیده است. با این وجود برنامه های ثانویه نیز می توانند خود را در یکی از مکان‌های مناسب جهت اجرای خودکار قرار دهند.

این سرویس یکی از سرویس‌های جذاب و پر استفاده برای بدافزار نویسان است زیرا می‌توانند پس از هر بار بارگذاری سیستم عامل به صورت خودکار و در پس زمینه و دور از چشم کاربر اجرا شوند.

۳- کارهای پیشین

شیوه‌های شناسایی کانال کنترل و فرماندهی در بات نت‌ها را می‌توان به سه دسته روش شبکه محور، روش میزبان محور و روش تقاضا محور تقسیم نمود.

۳-۱- روش شبکه محور

این روش معطوف به شناسایی باتنت‌ها با بررسی همبستگی ترافیک شبکه بین گروهی از سیستم‌های کامپیوتری نظیر بررسی آدرس IP مقصد، نام‌های Server، محتویات بسته[6]، ازدحام پاسخ و مواردی از این دست می‌باشد [۵ و ۱۵-۲۳]. این شیوه به‌خصوص در مواردی که تنها ترافیک شبکه موجود باشد کاربرد دارد.

۳-۲- روش میزبان محور

این روش متمرکز بر تفاوت قائل شدن بین پردازه‌های بدخواهانه و پردازه‌های سودبخش بر روی یک سیستم میزبان با در نظر گرفتن این موضوع است که پردازه‌های بات از داده‌های دریافت‌شده از شبکه به‌عنوان پارامتر در فراخوانی‌های سیستمی خود استفاده می‌کنند [۲۴]. روش تشخیص دیگری بر پایه نرخ بالای تلاش‌های ناموفق برای اتصال به یک سرور راه دور در [۲۵] ارائه گردیده است،

1- System Function Address Patching
2- System Service Descriptor Table (SSDT)
3- SSDT Shadow
4- Filter Driver
5- AutoRun
6- Packet

که برای باتنت‌هایی که ما در این مقاله مورد بحث قرار می‌دهیم کاربرد ندارد. زیرا در اینجا اتصالات به وب‌سایت‌های شبکه‌های اجتماعی مشهور می‌باشد که همیشه با موفقیت برقرار می‌شود. این روش اغلب با توجه عمیق به Stack نرم‌افزارها همراه است [۲۵].

۳-۳- روش تقاضا محور

این روش بر مبنای تقاضا برای تعاملات خاص استوار است. اخیرا تمرکز بر روی باتنت‌های مبتنی بر IRC استوار بوده است [۵]. امروزه امکان بهره‌گیری از پست الکترونیک‌ها به‌عنوان کنترل و فرماندهی (C&C) مورد بررسی قرار گرفته است [۲۶]. و امکان شناسایی چنین باتنت‌هایی از طریق ترافیک هرزنامه تولید شده در [۲۹-۲۷] ارائه گردیده است. نوع دیگری از این کلاس باتنت‌های مبتنی بر شبکه‌های اجتماعی هستند. مفهوم کنترل و فرماندهی مبتنی بر شبکه‌های اجتماعی به سال ۲۰۰۷ برمی‌گردد [۳۳-۳۰]، اما چنین باتنت‌هایی پس از این سال به صورت واقعی مشاهده گردید [۳۵- ۳۴].

۴- سامانه پیشنهادی

طرح پیشنهادی از دوفاز پایه، فاز شناسایی کانال کنترل و فرماندهی و فاز مقابله تشکیل گردیده است. در فاز شناسایی هدف بهره‌گیری از پارامترها و الگوریتم‌هایی است که به ما جهت تصمیم‌گیری در خصوص بدخواهانه ویا خیرخواهانه بودن درخواست‌های تولید شده در سطح شبکه یاری می‌رساند. در فاز مقابله نیز تصمیم‌گیری در خصوص راه‌کار مناسب جهت مقابله موثر با درخواست‌های بدخواهانه تولیدشده درسطح شبکه صورت می‌گیرد.

۴-۱- فاز شناسایی

در این مرحله هدف تصمیم‌گیری در خصوص بدخواهانه یا خیرخواهانه بودن درخواست‌های تولیدشده در سطح شبکه سیستم رایانه‌ای کاربر می‌باشد. ومبنای تصمیم‌گیری نیز به این صورت است که درخواست‌هایی که تشخیص داده شود توسط کاربر سیستم تولید گردیده است را در دسته درخواست‌های خیرخواهانه قرار می‌دهیم و درخواست‌هایی که توسط کاربر تولید نگردیده باشند و مبدا آن برنامه‌ای اجرایی باشد که دارای ویژگی‌های خاصی باشد را در دسته درخواست‌های بدخواهانه خواهیم گنجاند. فاز شناسایی از دو بخش مهم تعیین پارامترهایی که مورد ارزیابی قرار می‌گیرند و الگوریتمی که پارامترها رو مورد ارزیابی قرار می‌دهد تشکیل گردیده است.

۴-۱-۱- پارامترها

پارامترهای تعیین شده به صورت زیر تعریف و به کار گرفته شد:

- یک پنجره زمانی که مشخص کننده بازه زمانی قابل قبول برای تولید درخواست شبکه پس از فعالیت کاربر است.
- وضعیت پردازه تولیدکننده درخواست شبکه از لحاظ در پیش زمینه یا پس زمینه اجرا شدن.
- وجود یا عدم وجود پردازه در مکان‌های تعریف‌شده سیستم عامل تحت عنوان اتوران (اجرای خودکار).

۴-۱-۱-۱- پنجره زمانی و بررسی تعامل کاربر با سامانه

یکی از تفاوت‌های موجود بین تعامل کاربر با سیستم و بدافزارها با سیستم کامپیوتری در این است که کاربران بر خلاف بدافزارها تعامل بسیار زیادی با تجهیزات ورودی و خروجی دارند. این در حالی است که بدافزارها اعمال خود را بدون نیاز به دستگاه‌های ورودی خروجی و به‌صورت دیکته‌شده در بدنه فایل اجرایی به سیستم اعمال می‌کنند.

این تفاوت مبنای نخستین پارامتر انتخابی جهت ارائه به الگوریتم تصمیم‌گیرنده می‌باشد.

در این پژوهش آن دسته از تعاملات کاربر با دستگاه‌های ورودی و خروجی که موجب تولید درخواست مشاهده صفحه شبکه اجتماعی می‌شود می‌بایست ثبت و به الگوریتم تصمیم‌گیرنده ارائه گردد.

جهت برآورده کردن این خواسته از تکنیک قلاب‌اندازی استفاده گردیده است.

ساختار قلاب‌اندازی به دستگاه‌های ورودی خروجی (در اینجا صفحه کلید) به صورت زیر است:

```
LRESULT CALLBACK keyboardHookProc(int nCode,
WPARAM wParam, LPARAM lParam)
  {
    PKBDLLHOOKSTRUCT                    p=
(PKBDLLHOOKSTRUCT) (lParam);
  if (wParam == WM_KEYDOWN)
     {
      // process pressed keys and record last time it
      }
   }
  return CallNextHookEx(NULL, nCode, wParam, lParam);
  }
```

درساختار بالا با استفاده از روش قلاب‌اندازی تعامل کاربر با صفحه کلید رهگیری و شناسایی می‌گردد. هدف از این رهگیری، شناسایی آن دسته از فعالیت‌های در ارتباط با صفحه کلید کاربر است که موجب تولید درخواست مشاهده صفحه شبکه اجتماعی می‌گردد.

۴-۱-۱-۲- اجرای خودکار

همان‌طور که پیش از این نیز اشاره شد، بخش اجرای خودکار یکی از امکانات سیستم عامل است که برای بارگذاری خودکار برنامه‌ها، سرویس‌ها، درایورها و... پس از هر بار بوت شدن سیستم کاربر تعبیه

گردیده است. این سرویس از دیدگاه بدافزار نویسان بسیار جذاب می‌باشد چرا که ضمن مخفی ماندن بدافزار در پس زمینه امکان اجرای خودکار و بدون دخالت کاربر مهیا می‌گردد.

در این پژوهش مکان‌هایی از سیستم عامل را که به برنامه‌نویسان امکان اجرای خودکار برنامه‌ها را می‌دهد جمع‌آوری گردیده و مورد استفاده قرار گرفته است.

۴-۱-۱-۳- وضعیت پردازه از نظر اجرا در پیش‌زمینه یا پس‌زمینه

از دیگر پارامترهایی که توسط روش پیشنهادی مورد بررسی قرار می‌گیرد بررسی وضعیت پردازه می‌باشد. از آن جهت که نرم‌افزارهای قانونی و مشروع (بخصوص نرم‌افزارهای در ارتباط با کاربر تولید کننده درخواست‌های مشاهده صفحات اجتماعی) در پیش زمینه اجرا می‌گردند. اما برخلاف نرم‌افزارهای قانونی و مشروع بدافزارها با هدف مخفی ماندن و تمدید موجودیت خود بر روی سیستم قربانی سعی در اجرای خود در پس زمینه می‌نمایند. از این تفاوت آشکار می‌توان به عنوان یکی از شاخصه‌های تعیین‌کننده بدخواه یا خیرخواه بودن درخواست استفاده نمود.

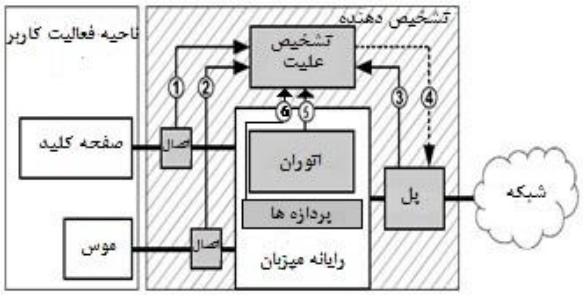
با استفاده از واسط برنامه‌نویسی کاربردی زیر از کتابخانه user32 می‌توان تعداد منابع گرافیکی به‌کاررفته شده در یک پردازه را مشخص نمود. در صورتی که پردازه از منابع گرافیکی استفاده ننماید و در پس زمینه اجرا گردد، خروجی این واسط برنامه نویسی کاربردی صفر خواهد بود.

```
DWORD WINAPI GetGuiResources (_In_ HANDLE
hProcess, _In_ DWORD  uiFlags)
```

۴-۲- الگوریتم و واحد تصمیم‌گیرنده

پس از معرفی پارامترهای مورد نیاز و شیوه پیاده‌سازی هر کدام، به تشریح ساختار تصمیم‌گیرنده و الگوریتم روش پیشنهادی خواهیم پرداخت.

درخواست مشاهده یک صفحه اجتماعی به خودی خود به وجود نمی آید. یک کاربر سیستم کامپیوتری در ایجاد درخواست و تولید ترافیک شبکه مربوط به این درخواست دخالت دارد. این عمل با استفاده از کلیدهای صفحه کلید و موش واره انجام می‌گیرد. درصورتی‌که یک بات‌جهت دریافت دستورات جدید و یا قرار دادن اطلاعات جدید به یک صفحه شبکه اجتماعی متصل شود و ترافیک متناسب با آن را تولید کند، تعامل با صفحه کلید و موس رخ نخواهد داد. از سوی دیگر بات ها پس آلوده‌سازی سیستم های کامپیوتری جهت اجرای مجدد در هر بار شروع به کار سیستم عامل خود را در بخش‌های Autorun سیستم‌عامل جاسازی می‌کنند. در حالی‌که مرورگرهای اینترنتی خاصیت اجرای خود کار نداشته و توسط کاربر سیستم اجرا می‌گردند. ما بر پایه این تفاوت‌ها اقدام به طراحی چارچوب سیستم تشخیص بات‌ها نموده‌ایم. در شکل (۲) نمای کلی سیستم تشخیص طراحی شده، تشریح گردیده است.



شکل ۲- نمای کلی سیستم پیشنهادی

اندیس‌های ۱ و ۲، مشخص‌کننده ورودی‌های کاربر هستند که از صفحه کلید و موس دریافت می‌شوند. اندیس ۵، بیان‌کننده وضعیت پردازه درخواست‌دهنده و تولیدکننده ترافیک شبکه در واحدAutoRun سیستم‌عامل می‌باشد. اندیس ۶، بررسی‌کننده وضعیت اجرای پردازه در پیش‌زمینه یا پس‌زمینه می‌باشد. اندیس ۳، دریافت‌کننده ترافیک شبکه می‌باشد و متناسب با الگوریتم تشخیص اگر درخواست امن تشخیص داده شود از طریق مسیر ۴، اجازه ادامه داده می‌شود.

آن دسته از فعالیت‌های کاربر که موجب تولید درخواست صفحه شبکه اجتماعی وبه دنبال آن ایجاد ترافیک شبکه می‌شود را می‌توان موارد زیر دانست:

- کلیک چپ ماوس
- کلید Enter صفحه‌کلید
- کلید F5 صفحه‌کلید

رهگیری و ثبت کلیدهای فشرده‌شده ماوس و صفحه‌کلید را می‌توان از طریق Hook کردن بخش‌های مربوطه سیستم عامل انجام داد.

فاصله زمانی بین درخواست صفحه مجازی ایجاد شده و کلید ثبت‌شده در این سیستم از اهمیت بالایی بر خوردار است. در نتیجه مطابق با جدول ۱ می بایست یک پنجره زمانی تعیین گردد تا در صورت مشاهده یک درخواست صفحه شبکه اجتماعی تعیین گردد که آیا در این بازه فعالیتی از سوی کاربر که موجب تولید درخواست صفحه شبکه اجتماعی شده، وجود داشته است یا خیر؟. در این پژوهش پنجره زمانی را در حالت‌های بار پردازشی کم سیستم، بار پردازشی متوسط و بار پردازشی زیاد به صورت تجربی محاسبه

گردیده است:

جدول ۱- پنجرهای زمانی بار شبکه درحالت‌های مختلف

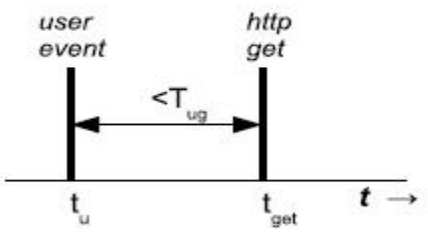
Time	Value (ms)
Tmax	۱۴۹
Tavg	۴/۸
Tmin	۰/۳

در این مرحله سه پارامتر Tu،Tget وTug را مورد استفاده قرار داده‌ایم. Tu بیانگر زمان ثبت‌شده فعالیت کاربر (کلیدهای فشرده‌شده صفحه‌کلید و ماوس)، Tget زمان ثبت شده درخواست مشاهده صفحه شبکه اجتماعی و Tug زمان میانگین است که از جدول بالا استخراج می‌شود. بار پردازشی کم، متوسط و زیاد سیستم طبق جدول (۲) محاسبه می‌گردد.

جدول ۲- بررسی وضعیت سیستم در سه حالت مختلف

	استفاده از پردازنده٪	پردازندهGHz
بار پردازشی کم	۲۰	۱/۶
بار پردازشی متوسط	۷۰	۲/۲
بار پردازشی زیاد	۹۰	۰/۳

بر اساس شکل ۳، یک ترافیک قانونی شبکه اجتماعی که توسط کاربر تولید گردیده است می‌بایست از رابطه $tget - tu < Tug$ پیروی کند که پس از دریافت یک درخواست مشاهده صفحه شبکه اجتماعی در صورتی که متناسب با بار پردازشی سیستم کامپیوتری فعالیتی از سوی کاربر روی نداده باشد می‌توان گفت این درخواست از سوی یک بات تولید گردیده است؛ اما در صورتی که در این پنجره زمانی فعالیتی دال بر تعامل کاربر با سیستم ثبت گردد می‌بایست اطمینان حاصل کرد که بات اقدام به جعل تعامل کاربر با سیستم نکرده باشد. به همین جهت و در ادامه سیستم پیشنهادی می‌بایست مبدأ تولیدکننده درخواست را بررسی کند. درصورتی‌که پروسه تولیدکننده درخواست یا پروسه‌های والد، خود را در بخش‌های Autorun سیستم عامل قرار داده باشند می‌توان نتیجه گرفت که درخواست تولید شده از سوی یک بات بوده است.



شکل۳- دیاگرام زمانی یک فعالیت کاربر و تولید درخواست صفحه شبکه اجتماعی

از مهم‌ترین مکان‌هایی که بات‌ها می‌توانند جهت اجراشدن در هر بار راه اندازی مجدد سیستم از آن بهره بگیرند می‌توان به کلیدهای رجیستری، سرویس‌ها، درایورها، فراهم‌کننده LSA و ... اشاره نمود. در این مرحله پارامتر atrn را تعریف می‌کنیم که نشان دهنده استفاده یا عدم استفاده از سرویس‌های اجرای خودکار سیستم عامل توسط برنامه درخواست‌دهنده صفحه می‌باشد.

با ادغام تمامی پارامترهای تعریف شده جهت شناسایی درخواست‌های اتصال به سرور کنترل و فرماندهی باتنت می‌توان عملکرد سیستم پیشنهادی را در قالب شبه کد زیر نمایش داد:

```
If (tget − tu > Tug and atrn==1 andbkg==1) then
Request ← unsafe
Else
Request← safe
End
```

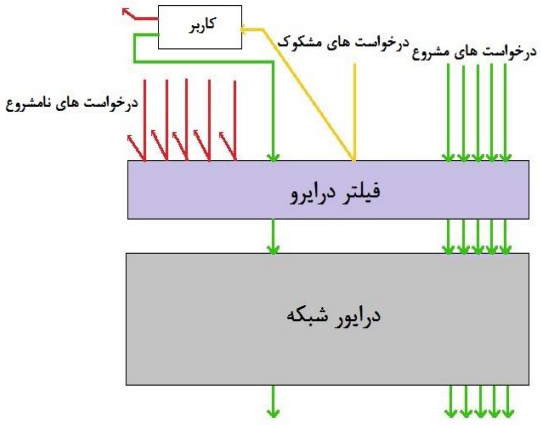
متناسب با الگوریتم بالا تصمیم‌گیری شامل دسته‌بندی درخواست‌ها در سه دسته مجزا می‌گردد. آن دسته از درخواست‌هایی که توسط پردازه‌های اجرا شونده در پیش زمینه تولید گردند و در بخش اجرای خودکار سیستم عامل موجود نباشند و در یک بازه زمانی مناسب فعالیتی دال بر این تعامل با سیستم ثبت گردد، درخواست قانونی تشخیص داده خواهد شد. در مقابل آن دسته از درخواست‌هایی که توسط پردازه‌های موجود در پس زمینه تولید گردد و پردازه درخواست‌دهنده در بخش اجرای خودکار سیستم ثبت گردیده باشد و تعاملی از جانب کاربر با سیستم موجود نباشد در دسته درخواست‌های غیرقانونی جای داده می‌شود. در این بین درخواست‌هایی که دارای برخی از این ویژگی‌ها باشند و فاقد برخی دیگر باشند به عنوان درخواست‌های مشکوک دسته‌بندی می‌گردند. در جدول (۳) دسته‌بندی درخواست‌ها ارائه گردیده است.

جدول ۳- وضعیت‌های ممکن در بررسی پارامترها

موجود در اجرای خودکار	اجرا در پس زمینه	تعامل کاربر	
خیر	خیر	بله	درخواست‌های قانونی
بله	بله	خیر	درخواست‌های غیرقانونی
خیر	بله	بله	درخواست‌های مشکوک
بله	بله	بله	درخواست‌های مشکوک
بله	خیر	بله	درخواست‌های مشکوک
خیر	بله	خیر	درخواست‌های مشکوک
خیر	خیر	خیر	درخواست‌های مشکوک
بله	خیر	خیر	درخواست‌های مشکوک

۴-۳- مقابله

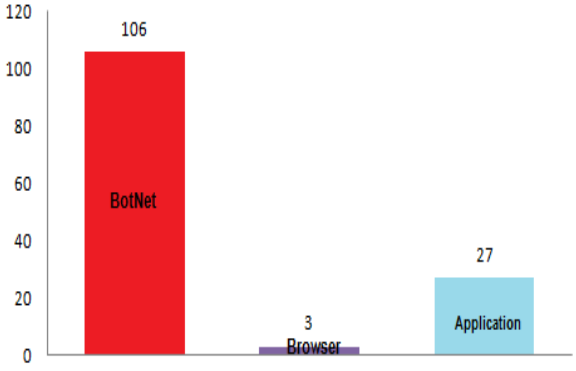
پس از رهگیری درخواست‌ها و بررسی آنها می‌بایست اقدامات لازم صورت بگیرد. همان‌طور که در شکل ۴ مشاهده می‌گردد، درخواست‌هایی که توسط واحد تصمیم‌گیرنده قانونی تشخیص داده شوند اجازه اجرا به آنها داده می‌شود. در مقابل درخواست‌هایی که نامشروع و بدخواهانه باشند از بین می‌روند. آن دسته از درخواست‌هایی که مشکوک تشخیص داده شوند پس از اطلاع به کاربر و تصمیم‌گیری در خصوص آنها از سمت کاربر عملیات لازم انجام خواهد گرفت. اجرای عملیات به معنی از بین بردن و یا ارسال درخواست به بیرون بر عهده فیلتر درایوری است که پیش از این تشریح گردید؛ می‌باشد.



شکل ۴. ساختار مدیریت درخواست ها

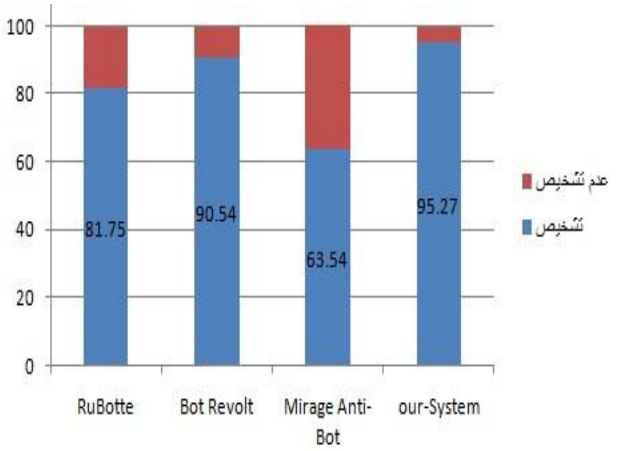
۵- نتایج و بحث

در این قسمت به ارزیابی میزان دقت روش مذکور در شناسایی بات‌نت‌ها پرداخته و روش پیشنهادی را با چند نرم‌افزار شناسایی کننده بات‌نت مقایسه می‌کنیم. براساس شکل ۵ در ارزیابی روش پیشنهادی از ۱۰۶ نمونه بات‌نت از مراجع [۳۷-۳۶] استفاده گردیده است. همچنین روش پیشنهادی امکان رهگیری، شناسایی و مقابله همزمان با چند بات‌نت را دارد.



شکل ۵- میزان بات‌نت، مرورگر و سایر نرم‌افزارهای مورد ارزیابی قرارگرفته

نمونه بات‌نت‌ها و سایر برنامه‌های فوق بر روی سیستم‌عامل ویندوز ۷، نسخه ۳۲ بیتی پس از نصب برنامه‌های امنیتی زیر اجرا شده است. هدف از مقایسه روش پیشنهادی با نرم‌افزارهای مشابه، تعیین میزان موفقیت در تشخیص و شناسایی ترافیک‌های شبکه تولیدشده از سوی بات‌نت‌ها برای اتصال به سرور کنترل و فرماندهی خود می‌باشد. نتایج این مقایسه در شکل ۶ آورده شده است.



شکل ۶- مقایسه روش پیشنهادی با چند برنامه ضد بات‌نت.

در شکل ۶، نرخ دقت، شاخص تعداد بات‌نت‌های کشف‌شده می‌باشد. این ارزیابی و مقایسه حکایت از دقت بیش از ۹۵ درصدی روش پیشنهادی در شناسایی و مقابله با کانال کنترل و فرماندهی در بات‌نت‌های مبتنی بر شبکه‌های اجتماعی دارد.

۶- نتیجه‌گیری

در این مقاله روشی کارا برای شناسایی، رهگیری و مقابله با بات‌نت‌های مبتنی بر شبکه‌های اجتماعی توضیح داده شد. روش پیشنهادی با استفاده از نظارت بر منابع سیستم وجود بات‌نت‌ها را تشخیص می‌دهد. روش پیشنهادی پس از تشخیص وجود بات‌نت اقدام به مقابله و جلوگیری از اتصال بات به سرور کنترل و فرماندهی می‌کند. در روش پیشنهادی شناسایی بات‌نت شامل نظارت بر ورودی‌های کاربر، ترافیک شبکه، پردازه تولیدکننده درخواست و وجود یا عدم وجود پردازه در بخش اجرای خودکار سیستم‌عامل می‌باشد. نحوه پیاده‌سازی روش پیشنهادی برای تولید ابزاری مؤثر در کشف و مقابله با بات‌نت‌ها تا حد امکان تشریح شد. در انتها روش پیشنهادی از جهات مختلف ارزیابی شد که حاصل آن دقت نزدیک به ۹۶ درصد در تشخیص و موفقیت ۱۰۰ درصدی در مقابله با بات‌نت‌ها بود و برای اثبات کارایی با نرم‌افزارهای ضد بات‌نت مقایسه شد.

۷- مراجع

1. E. Stinson and J. C. Mitchell, “Characterizing the Remote Control Behavior of Bots,” Lecture Notes in Computer Science, vol. 4579, p. 20, 2007.
2. C. J. Dietrich, “Identification and Recognition of Remote-Controlled Malware,” master, computer science, mannheim, 2013.
3. G. Fedynyshyn, M. C. Chuah, and G. Tan, “Detection and Classification of Different Botnet C&C Channels,” in Autonomic and Trusted Computing: 8th International Conference, ATC 2011, Banff, Canada, September 2-4, 2011. Proceedings, J. M. A. Calero, L. T. Yang, F. G. Mármol, L. J. García Villalba, A. X. Li, and Y. Wang, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 228-242, 2011.
4. S. Ashutosh, “Social networking for botnet command and control,” master Project, Computer Science, San Jose State University, 2012.
5. J. Goebel and T. Holz, “Rishi: identify bot contaminated hosts by IRC nickname evaluation,” presented at the HotBots'07 Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets CA, USA, 2007.
6. S. Shah and V. M. Lomte, “The Survey Paper on ASP2P: An Advanced Botnet Based on Social Networks over Hybrid P2P,” International Journal of Science and Research (IJSR), vol. 3, p. 6, December 2014.
7. P. Jaikumar and A. C. Kak, “A graph-theoretic framework for isolating botnets in a network,” Security and Communication Networks, vol. 8, p. 19, 28 February 2012.
8. S. Chang and T. E. Daniels, “P2P botnet detection using behavior clustering & statistical tests,” 2nd ACM workshop on Security and artificial intelligence, p. 8, 2009.
9. P. Wang, S. Sparks, and C. C. Zou, “An Advanced Hybrid Peer-to-Peer Botnet,” IEEE Transactions on Dependable and Secure Computing, vol. 7, p. 113, 2010.
10. A. blue, Hooks, 2011. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/ms632589%28v=vs.85%29.aspx
11. M. Russinovich and D. A. Solomon, “Windows Internals Part 1: Microsoft,” 2012.
12. B. Blunden, “The Rootkit Arsenal: WordWar,” 2009.
13. Microsoft, Filter Drivers, 2015. Available: https://msdn.microsoft.com/en-us/library/windows/hardware/ff545890%28v=vs.85%29.aspx
14. R. Mark. Autoruns for Windows, 2015. Available: https://technet.microsoft.com/en-us/sysinternals/bb963902.aspx
15. J. Binkley and S. Singh, “An algorithm for anomaly-based botnet detection,” In: Proc. Reducing Unwanted Traffic on the Internet, 2007. Available: http://static.usenix.org/events/sruti06/tech/full_papers/binkley/binkley_html/
16. P. P. G. Gu, V. Yegneswaran, and M. Fong, “BotHunter: Detecting malware infection through ids-driven dialog correlation,” 2007. Available: http://static.usenix.org/legacy/events/sec07/tech/full_papers/gu/gu_html/
17. G. Gu, R. Perdisci, J. Zhang, and W. Lee, “BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection,” USENIX Security Symposium, 2008.
18. G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic,” presented at the Computer Science and Engineering Faculty Publications, 2008.
19. M. P. Collins and M. K. Reiter, “Hit-List Worm Detection and Bot Identification in Large Networks Using Protocol Graphs,” Software Engineering Institute p. 20, September 2007.
20. X. Hu and M. Knysz, “Rb-seeker: Auto-detection of redirection botnets,” presented at the Proceedings of the Network and Distributed System Security Symposium, San Diego, California, USA, 2009.
21. M. P. Collins, “Using uncleanliness to predict future botnet addresses,” presented at the 7th ACM SIGCOMM conference on Internet measurement 2007.
22. A. Karasaridis and B. Rexroad, “Wide-scale botnet detection and characterization,” presented at the HotBots'07 Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets CA, USA, 2007.
23. E. Cooke, “The Zombie roundup: understanding, detecting, and disrupting botnets,” presented at the SRUTI'05 Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet, CA, USA, 2005.
24. E. Stinson and J. C. Mitchell, “Characterizing Bots’ Remote Control Behavior,” presented at the 4th International Conference, DIMVA 2007 Lucerne, Switzerland, July 12-13, 2007
25. C. Yan, T. D. Dimitriou, and J. Zhou, “Using Failure Information Analysis to Detect Enterprise Zombies,” presented at the 5th International ICST Conference, Greece, Athens, Secure Comm. 2009.
26. K. Singh, A. Srivastava, J. Giffin, and W. Lee, “Evaluating email’s feasibility for botnet command and control,” presented at the Dependable Systems and Networks with FTCS and DCC, DSN 2008, IEEE International Conference on, 2008.
27. Y. Zhao, Y. Xie, and F. Yu, “BotGraph: Large Scale Spamming Botnet Detection” The 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI '09), April 1, 2009 2009.
28. L. Zhuang, J. Dunagan, and D. R. Simon, “Characterizing botnets from email spam records,” presented at the LEET'08 Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, CA, USA, 2008.
29. Y. Xie, F. Yu, and R. Panigrahy, “Spamming Botnet: Signatures and Characteristics,” presented at the ACM SIGCOMM 2008, Seattle, WA USA, 2008.
30. S. Poland, “How to create a twitter bot,” 2007. Available: http://blog.stevepoland.com/how-to-create-a-twitter-bot
31. DigiNinja, Kreiosc2: Poc using twitter as its command and control channel, 2008. Available: http://www.digininja.org
32. J. P. John, A. Moshchuk, and S. D. Gribble, “Studying spamming botnets using Botlab,” Presented at the NSDI'09 Proceedings of the 6th USENIX symposium on Networked systems design and implementation CA, USA, 2009.
33. F. S. Inc, “Web security trends report q4 .Technical report,” 2007. Available: http://www.finjan.com/Content.aspx?id=827
34. J. Baltazar, J. Costoya, and R. Flores, “The Real Face of KOOBFACE: The Largest Web 2.0 Botnet Explained,” Trend Micro Threat Research, p. 18, 2009.
35. T. Easton and K. Johnson, “Social zombies,” presented at the DEFCON, CA, USA, 2009.
36. V. S. M. D. Base, “Virus Sign Malware Data Base,” Ed, 2014.
37. C. SandBox, “CW_Sand Box Data,” 2014. Available: http://pi1.informatik.uni-mannheim.de/malheur/

Detection and Defense with the Command-and-Control Channels in the Social Network-Based BotNet

H. Tanha, M. Javanmard*

Abstract

A BotNet is a collection of Bots that are run separately on infected Computers and respond to commands sent from command and control unit. Nowadays, BotNets are a major challenge in cyberspace . New species of this family of malware abuses the popular social networking sites as command and control channels. In this way, the usual protective systems such as IDS will not be able to identify and deal with BotNets, because the network traffic generated by BotNets is similar to the user traffic system. In this paper, a new method is presented to detect, intercept and deal with BotNets based social networks. In this method, BotNet detection is done based on the system resource monitoring. After tracing, the proposed method intercepts with the BotNet. This interception includes preventing from the command and control server. the proposed method can detect nearly 96% of BotNets and can caus the system to discover 100% of BotNets successfully.

Key Words: *BotNet detection, command and control, social network, Security*

* Imam Hussein Comprehensive University (javanmard@pnu.ac.ir)- Writer-in-Charge