Scientific Journal of Electronical & Cyber Defence
Vol. 10, No. 2, Summer 2022, Serial No. 38

A cooperative and independent deception system in the active cyber defense system

K. Dadash Tabar Ahmadi*, M. Mahmoud Babooyi
*Assistant Professor, Malik Ashtar University of Technology, Tehran, Iran

(Received: 23/08/2021, Accepted: 31/10/2021)

ABSTRACT

Cyber deception technology is a part of the process of identifying and responding to incidents. This technology helps the security team identify and analyze advanced threats by persuading an attacker to strike fake resources. The deception approach is to create a high-precision warning about high-risk behaviors. Deception occurs in a variety of ways, including an active defense approach. Active defense is an approach that is based on the establishment of measures to detect, analyze, identify and reduce threats to communication systems and networks in real time by default, which ultimately leads to cyber security. To better understand the techniques used in active defense, we can mention the Honeypot. The Honeypot is a trick that is deliberately placed on the net to be explored by an attacker in order to record, track and analyze the activities performed. In this project, we have used a low-interaction Honeypot to identify malicious activities. Using these technologies and strategies, we have designed an active cyber defense system (SDF). Taking into account the IP, this system has the capability of monitoring and real-time detection of abnormalities that occur in the form of functional level of attackers. Both the cyber deception and the honeypot concentrate on trapping the attacker by misleading, confusing, and etc. But active cyber deception (SDF) technology is an evolution of Honeypot, extending its limited capabilities.

Keywords: Honeypot, cyber deception, active cyber defense, low interaction.

* Corresponding Author Email: Dadashtabar@mut.ac.ir

نشریه علمی "پدافند الکترونیکی و سایبری"

سال دهم، شماره ۲، تابستان ۱۴۰۱، ص ۱۴۲-۱۲۹

علمی – پژوهشی

طراحی یک سامانه فریب همکارانه و مستقل در سامانه دفاع فعال سایبری

کوروش داداش تبار احمدی[1*]، محمد محمودبابویی[2]

۱- استادیار، ۲- دانشجوی کارشناسی ارشد، دانشگاه صنعتی مالک اشتر، تهران، ایران

(دریافت: ۱۴۰۰/۰۶/۰۱، پذیرش: ۱۴۰۰/۰۸/۰۹)

چکیده

فناوری فریب سایبری بخشی از فرآیند شناسایی و پاسخگویی به حوادث سایبری است. این فناوری مهاجمان را به سمت دارایی‌های دروغین IT هدایت کرده تا تهدیدات پیشرفته را شناسایی و تجزیه و تحلیل کند. هشدارهای ایجاد شده در سامانه فریب دارای صحت بالایی است. فریب به روش‌های مختلفی صورت می‌گیرد که رویکرد دفاع فعال از جمله آن‌هاست. دفاع فعال سایبری مجموعه اقداماتی را دربر می‌گیرد که ما را در رسیدن به امنیت سایبر هدایت می‌کند. این اقدامات شامل تشخیص، تجزیه و تحلیل، شناسایی و کاهش تهدیدات نسبت به سامانه و شبکه‌های ارتباطی در زمان واقعی را شامل می‌شود. از ابزارهای دفاع فعال می‌توان به تله عسل اشاره نمود. تله عسل فریبنده‌ای است که به عمد در شبکه قرار می‌گیرد تا توسط مهاجم کاوش شود و فعالیت‌های انجام گرفته را ثبت، ردیابی و تحلیل نماید. در این تحقیق به نوع کم تعامل آن پرداخته شده است که برای شناسایی فعالیت‌های مخرب مورد استفاده قرار می‌گیرد. با توجه به ابزار و استراتژهای موجود، سامانه دفاع فعال سایبری (سدف سایبری) طراحی شده است تا به‌صورت بلادرنگ به مانیتورینگ ناهنجاری رخ داده بپردازد. سدف توانایی تفکیک سطح عملکردی مهاجمین را با توجه به IP دارا است. مباحث مربوط به فریب سایبری و تله عسل بر روی به دام انداختن مهاجم از طریق گمراه کردن، گیج کردن و ... تمرکز دارد. در حقیقت فناوری به‌کار رفته در سدف نوع تکامل یافته تله عسل است بدین صورت که قابلیت‌های محدود آن را گسترش می‌دهد.

کلیدواژه‌ها: **تله عسل، فریب سایبری، دفاع فعال سایبری، کم تعامل**

۱- مقدمه

فناوری فریب[1] بخشی از فناوری شناسایی و پاسخگویی به حوادث است که به گروه‌های امنیتی کمک می‌کند، تا مهاجمان با تعاملی که با دارایی‌های دروغین بخش IT دارند مشغول شده و از این طریق تهدیدات پیشرفته را شناسایی، تجزیه و تحلیل و به دفاع بپردازند. رویکرد فریب می‌تواند هشدارهایی با صحت بالا را در مورد رفتارهای مخرب خاص به شما ارائه دهد. شناسایی این موارد چالش برانگیز توسط روش‌هایی مانند تجزیه و تحلیل log های سامانه یا ابزاری به نام SIEM به تنهایی انجام می‌گیرد. فایده این کار این است که شما می‌توانید فعالیت‌های مشکوک را در اوایل زنجیره حمله شناسایی کنید و همچنین دشمن خود را در شبکه داخلی گیج و هدایت نادرست نمایید. مدتی است که از ساخت اولین تله عسل[2] می‌گذرد و مفهوم فریب‌کاری به‌عنوان سازوکار بالقوه‌ای برای شناسایی، کاهش سرعت و ضد حمله به حریف درنظر گرفته می‌شود. تقریباً دو دهه است که برای پذیرش فریب سایبری تلاش‌هایی صورت می‌گیرد. در ادامه به معرفی یکی از ابزارهای کاربردی در دفاع فعال می‌پردازد شده است و پایه‌های اساسی آن را طبق مدل پیشنهادی مقاله دسته‌بندی می‌شود که در شکل (۱) مشاهده می‌شود. سپس به شرح سامانه پرداخته می‌شود و معماری لازم اعم از ساختار کلی، روند پردازشی، جمع‌آوری داده و آماده سازی اطلاعات ارائه می‌شود.. در آخر نیز به تشریح توابع کاربردی در قسمت ۵ پرداخته می‌شود و با سامانه‌های موجود مقایسه می‌شود. از رویکردهای نوین سدف می‌توان به به‌کارگیری سرویس‌های متنوع اشاره نمود که با استفاده از یادگیری ماشین با مهاجم تعامل دارد.

۲- تله عسل ابزاری برای دفاع فعال[3]

گسترش دامنه فضای مجازی به قدری فراگیر است که وزارت دفاع ایالات متحده فضای مجازی را هم تراز زمین، دریا و هوا به‌عنوان حوزه جنگی قرار داده است [۱]. سامانه‌ها در فضای مجازی هر روز دائماً با تهدیدهای سایبری روبه‌رو می‌شوند. در

* رایانامه نویسنده مسئول: Dadashtabar@mut.ac.ir

[1] Deception Technology
[2] Honeypot
[3] Active Defense

سال ۲۰۱۵ ، سیمانتک[1] ۵۴ آسیب‌پذیری روز صفر[2] را کشف کرد که ۱۲۵ درصد افزایش نسبت به سال قبل داشته است [۲]. از آنجا که تهدیدات سایبری را نمی‌توان به‌طور کامل حذف کرد، راهبرد ایمن‌سازی فضای مجازی و حذف بسیاری از آسیب‌پذیری‌ها تا حد امکان قبل از بهره‌برداری ممکن است [3]. تله عسل یک تأسیسات امنیتی حیاتی با هدف قربانی کردن منابع خود برای بررسی دسترسی‌های غیر مجاز به‌منظور کشف آسیب‌پذیری‌های بالقوه در سامانه‌های عملیاتی و کاهش خطرات است. به خاطر طراحی منحصر به فرد و ویژگی‌های کاربردی آن، به رفع کاستی‌های دیگر روش‌های امنیتی موجود کمک می‌کند.

۳- نحوه تعامل با جامعه هدف

جامعه Blackhat از هوش کافی برای ایجاد تهدیدهای ناشناخته جدید برخوردار است. یک راه خوب برای بررسی تهدیدهای جدید، ثبت گام به گام اقدامات مخربی هست که باعث به خطر افتادن سامانه می‌شود. از این رو تله عسل‌ها می‌توانند با در اختیار گذاشتن یک سامانه قربانی ارزش تحقیقاتی که بر روی حملات انجام می‌گیرد را تکمیل کنند. به علاوه، لازم است مشاهده شود که دشمنان در سامانه مورد تهاجم چه می‌کنند، مانند برقراری ارتباط با مهاجمان دیگر و بارگذاری rootkits جدید، همچنین تله عسل‌ها می‌توانند حملات خودکار را به دام بیندازند [۴ و ۵]. با توجه به این واقعیت که حملات خودکار اغلب کل شبکه را هدف قرار می‌دهند، تله عسل‌ها می‌توانند به سرعت آن‌ها را برای آنالیز، ضبط کنند. از این رو با توجه به الزامات امنیتی متفاوت انواع تله عسل‌ها پیشنهاد می‌شود از جمله تله عسل‌های اختصاصی، تله عسل‌های اشتراکی و تله عسل‌های هیبرید را می‌توان نام برد [۶]. با این وجود، یک روش مشخص وجود ندارد که بتواند به سرعت نکات کلیدی تله عسل را به‌دست آورد و بتواند بینش‌های جدیدی را کشف کند [۴، ۵، ۷ و ۸].

کار ما ارائه این مشکلات است. سهم اصلی این مقاله را می‌توان به شرح زیر خلاصه کرد:

- ✓ دو عنصر اساسی در تله عسل که شامل تله[3] و برنامه‌های امنیتی است. نحوه به دام انداختن مهاجمان و بیان سازمان‌دهی آن یک دید کلی برای تجزیه و تحلیل تله عسل با تعاملات مختلف به ما می‌دهد.
- ✓ یک طبقه‌بندی جدید بر اساس تله و برنامه‌ریزی امنیتی برای تعریف فناوری تله عسل پیشنهاد شده است.
- ✓ بر اساس طبقه‌بندی، چندین روند توسعه شناسایی شده‌اند.

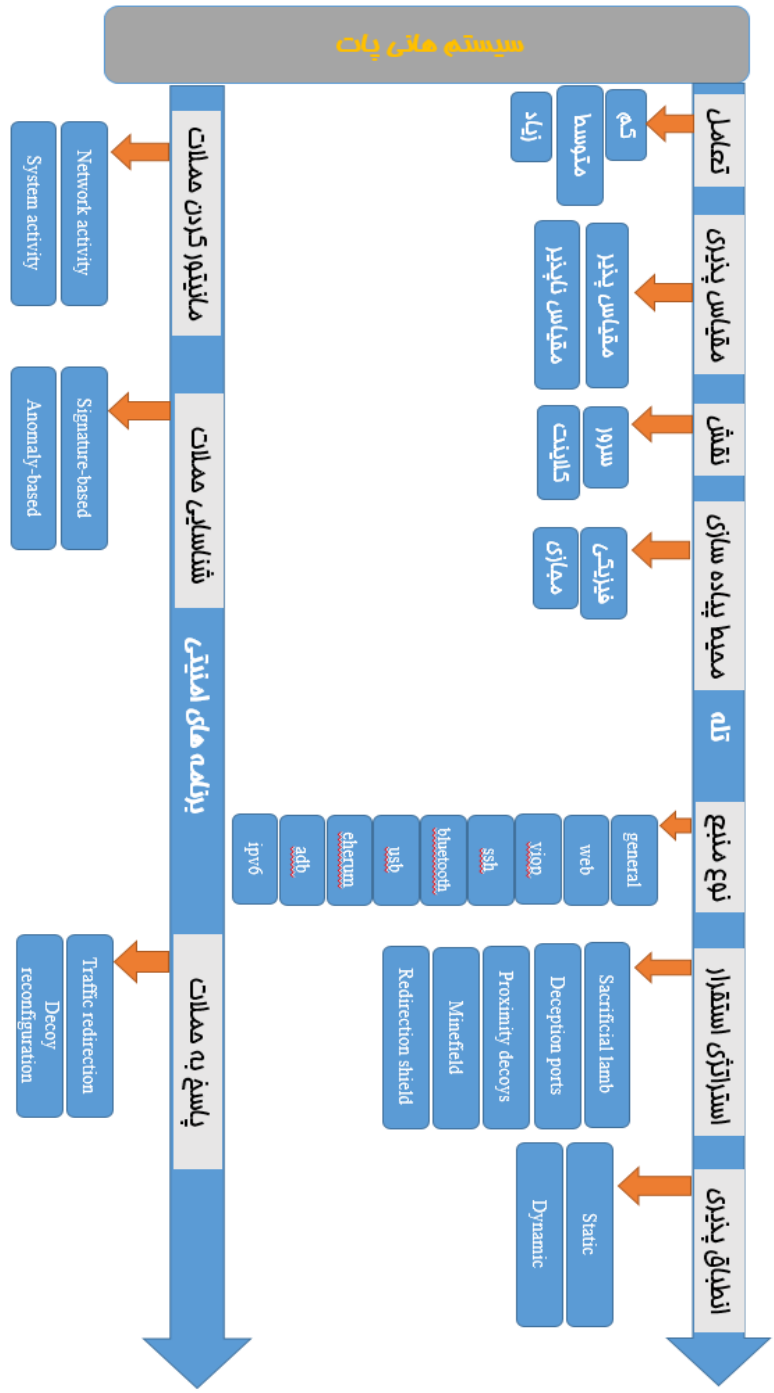
۴- طبقه‌بندی پیشنهادی و کارهای گذشته

این بخش طبقه‌بندی مبتنی بر D-P[4] را پیشنهاد می‌کند که شمای کلی آن در شکل (۱) نشان داده شده است. طرح طبقه‌بندی به دودسته تقسیم می‌شود. دسته اول شامل ویژگی‌های یک طعمه است و دسته دوم شامل عملکردهای یک برنامه امنیتی است. طبقه بندی مبتنی بر D-P به‌عنوان مدل مفهومی اساسی به‌منظور بررسی فناوری تله عسل استفاده می‌شود. تحت این چارچوب طبقه‌بندی، تله عسل‌های معمولی و روش‌های مربوط به تله عسل‌های خاص مرور می‌شود. اصطلاحات در این قسمت از گزارش به روشی فنی شرح داده شده استکه می‌تواند تعاریف، آن‌ها را متمایز کند و فهم آسانی به ما بدهد.

تله عسل یک عنصر اطلاعاتی است که دارای دو عنصر برنامه‌های امنیتی و تله است. با توجه به برنامه‌های امنیتی استفاده غیر مجاز و غیر قانونی که مبتنی بر اهداف تحقیقاتی امنیتی است، منابع اطلاعاتی به‌صورت عمدی در اختیار قرار می‌گیرند. تله می‌تواند هر نوع سامانه اطلاعاتی باشد و برنامه امنیتی عملکردهای مربوط به امنیت را کنترل می‌کند، مانند نظارت بر حمله، جلوگیری، شناسایی، پاسخ و profiling. علاوه بر این، برنامه‌های امنیتی باید به‌صورت پنهان در حال اجرا باشند تا از شناسایی شدنشان جلوگیری شود. در میان پروژه‌های موجود تله عسل و کار تحقیقاتی که روی آن صورت گرفته اصطلاحات ناسازگاری وجود دارد. تعدادی از تله‌ها به تله عسل اشاره دارد. برای مثال یک تله می‌تواند یک موجودیت مصنوعی دیجیتال باشد. اصطلاح فنی برای موجودیت دیجیتال که به‌عنوان تله عمل می‌کند honeytoken است. در کتاب The Cuckoo's Egg گیرافتادن با honeytoken توسعه داده شده، یعنی با فایل‌های دیجیتال به ردیابی هکر آلمانی پرداخته است [۹-۱۱]. بنابراین honeytoken یک تله است، اما یک سامانه به دام انداختن، Honeypot system است. تعریف ما این مطلب را روشن می‌کند که یک سامانه بدون برنامه امنیتی یک تله است نه تله عسل، اما اگر به برنامه امنیتی مجهز شد باز هم ما آن را تله عسل نمی‌نامیم. سازمان‌دهی دو عنصر اساسی را می‌توان تقریباً بر اساس درجه اتصال آن‌ها طبقه‌بندی کرد: سست و محکم[5]. کوپلینگ[6] به مقدار دانش مستقیم اشاره دارد که یک مؤلفه دارای یک مؤلفه دیگر است. کوپلینگ شل[7] به ترکیبی گفته می‌شود که

[1] Symantec
[2] Zero Day Vulnerability
[3] Decoy
[4] Decoy and Security Program
[5] Loose and Tight
[6] Coupling
[7] Loose Coupling

هر جزء آن به‌صورت جداگانه است کار می‌کند یا از هم دانش کمی دارند. این اجزاء را قادر می‌سازد تا در حالی که هنوز با یکدیگر ارتباط برقرار می‌کنند کاملاً مستقل و از یکدیگر بی‌خبر باشند. در مقابل، کوپلینگ محکم زمانی است که گروهی از اجزای سازنده وابستگی زیادی به یکدیگر داشته یا برای انجام وظیفه در همان واحد تعبیه شده باشند. تله عسل مستقل[1] به آن اشاره می‌کند که از کوپلینگ محکم استفاده می‌کند و تله عسل تعاونی[2] نشان می‌دهد که از کوپل شل استفاده می‌کند. ناروکی و همکاران تعدادی از تله عسل‌ها را مورد بررسی قرار داده‌اند، اما در سامانه‌های پیچیده‌ای مانند honeynet و سامانه‌های ترکیبی، تله عسل مشترک هستند [۸-۴].



شکل (۱): شمای کلی ساختار تله عسل بر اساس تعامل، عملکرد و نحوه به‌کارگیری پیشنهادی

[1] Independent Honeypot

[2] Cooperative Honeypot

۴-۱- ویژگی‌های تله

هدف از این تله، ثبت داده‌ها از طریق حمله است. چندین ویژگی اولیه وجود دارند که طراحی یک تله را تشکیل می‌دهند.

۴-۲- تعامل‌پذیری

این نشان دهنده میزان دقت یک منبع سامانه اطلاعاتی است که توسعه دهنده تله در اختیار مهاجم می‌گذارد. این تعامل را به سه سطح کم، متوسط، زیاد طبقه‌بندی می‌کنند [۴، ۵، ۷ و ۱۹-۱۲].

۴-۳- مقیاس پذیر

مقیاس پذیری[1] نشان دهنده توانایی فراهم کردن تعدادی از طعمه‌ها یا پتانسیل آن‌ها برای بزرگ شدن است تا بتواند رشد کند، که خود به دو دسته طبقه‌بندی می‌شود: مقیاس پذیر و مقیاس ناپذیر. یک تله عسل مقیاس ناپذیر دارای تعداد محدودی طعمه است (یک یا چند تا) به‌عنوان مثال Argos فقط می‌تواند یک طعمه مجازی را رصد کند. اما یک تله عسل مقیاس پذیر می‌تواند چندین طعمه را باهم رصد و پایش کند [۸، ۱۲، ۱۵ و ۲۰].

۴-۴- تطبیق‌پذیری

به قابلیت تنظیم مجدد جهت انطباق وضعیت طعمه با شرایط تغییر یافته تطبیق‌پذیری[2] گفته می‌شود. دارای دو سطح ایستا و پویا است. تله عسل ایستا سنتی (specter و dionea) از جمله تله عسل‌های سنتی ایستا هستند و نیاز به یک محقق امنیتی جهت آماده‌سازی پیکره‌بندی یا پیکره‌بندی مجدد دارند [۲۳-۲۰].

۴-۵- نقش

این مطلب را روشن می‌کند که در چه قسمتی از فریب به نقش[3] بازی کردن در معماری چند لایه می‌پردازد. تله عسل دو نقش را بازی می‌کند: سرور یا کلاینت. این بدین معنی است که تله عسل به‌صورت منفعلانه یا فعال به شناسایی برنامه‌های مخرب می‌پردازد یا خیر. بیشتر تله عسل‌ها مانند Honeyd و dionaea سمت سرور هستند یعنی به‌صورت منفعلانه با مهاجم برخورد می‌کنند، بدین ترتیب مهاجمین با استفاده از ابتکار عملشان آن‌ها را پیدا می‌کنند [۲۲-۲۰ و ۲۴].

۴-۶- فیزیکی/ مجازی

این مطلب نشان می‌دهد تله در آنجا حضور دارد و به دو دسته فیزیکی و مجازی تقسیم‌بندی می‌شوند. تله عسل فیزیکی به یک سامانه رایانه‌ای واقعی گفته می‌شود که روی دستگاه فیزیکی کار می‌کند و به‌عنوان طعمه عمل می‌کند. در واقع، تله عسل فیزیکی اغلب دلالت بر تعامل با درجه بالا دارد، اما می‌تواند عملکرد بالاتری نسبت به HIH[4] مجازی داشته باشد [۳۸-۲۵].

۴-۷- استراتژی استقرار

ارائه استقرار محل طعمه‌های فریب را بیان می‌کند. استراتژی استقرار طعمه پنج شیوه متداول دارد:

- ✓ قربانی کردن بره[5]
- ✓ پورت‌های فریب[6]
- ✓ طعمه‌های نزدیک[7]
- ✓ میدان مین[8]
- ✓ سپر تغییر مسیر[9]

۴-۷-۱- قربانی کردن بره

یک سامانه عادی است، اما بدون اتصال به شبکه های تولید، که منتظر است توسط مهاجمان به خطر بیفتد، به‌عنوان مثال Argos و Cuckoo Sandbox [۳۹]. این می‌تواند یک رایانه تجاری خارج از قفسه (COTS)، یک روتر یا یک سوئیچ و غیره باشد. پیاده‌سازی معمول شامل بارگیری سیستم‌عامل، پیکربندی برخی از برنامه ها و سپس رها کردن آن در شبکه برای دیدن اتفاقات است. بره‌های قربانی شده میانگین تجزیه و تحلیل یک سامانه به خطر افتاده تا آخرین بایت را ارائه می دهند. تجزیه و تحلیل اغلب به چندین ابزار شخص ثالث نیاز دارد. آن ها همچنین امکانات محدود کنترل ترافیک را فراهم نمی کنند، بنابراین نیاز به ملاحظات شبکه بیشتری نیز دارند [۱۲ و ۱۳].

۴-۷-۲- پورت‌های فریب

نشان دادن خدمات شبیه‌سازی شده به همراه خدمات شناخته شده در سامانه‌های تولیدی را بر عهده دارند. این‌ها اساساً LIH[10] یا MIH[11] هستند، مانند Spectre و Dionaea، که از خدمات

[1] Scalability
[2] Adaptability
[3] Role

[4] High Interaction Honeypot
[5] Sacrificial Lamb
[6] Deception Ports
[7] Proximity Decoys
[8] Minefield
[9] Redirection Shield
[10] Low Interaction Honeypot
[11] Medium Interaction Honeypot

مختلف در پورت‌های مختلف سامانه تقلید می‌کنند. به‌عنوان مثال HTTP در پورت ۸۰ به تقلید می‌پردازد. تله عسل ابتدا سیستم عامل را به خوبی مشاهده نموده و سپس تقلید می‌کند. ایده اصلی فریب این است که زمانی مهاجم به دام افتاد و خواهان خروج از شبکه بود بتواند آن را به دام بیندازد [۲۲ و ۲۳].

۴-۷-۳- تله‌های نزدیک

نشان می‌دهد که طعمه‌ها در همان شبکه، سامانه‌های تولیدی مستقر می شوند و احتمالاً پیکربندی سامانه های تولیدی را شبیه‌سازی می‌کنند. هیچ مشکلی از لحاظ قانونی به جهت رصد کردن تله‌ها وجود ندارد، چرا که آن‌ها زیرمجموعه سامانه سرورهای تولیدی ما می‌باشند و این اجازه صادر می‌شود که بر روی شبکه نظارت داشته باشند. همچنین، هنگامی که برخی از حملات مخرب در سامانه‌های تولیدی شناسایی می‌شوند، مسیریابی مجدد آن‌ها را به تله عسل هدایت می‌کند یا آن‌ها را به دام می‌اندازد، زیرا آن‌ها در مجاورت سامانه‌های تولید هستند. honeyd می‌تواند آدرس‌های IP آزاد شبکه را در سامانه‌های تولیدی شبکه جهت توسعه و ادغام تله‌ها مورد استفاده قرار دهد که این استراتژی استقرار این نوع تله عسل است [۲۰].

۴-۷-۴- میدان مین

به معنای استقرار تعداد نسبتاً زیاد تله عسل در محیط یا خط مقدم شبکه محافظت شده برای ایفای نقش مین‌های زمینی است که (هنگام تماس منفجر می‌شوند) و منظور ما این است که عملکرد ضبط داده را هنگام تماس روشن کنید. هر اسکن یا آشکارسازهای آسیب‌پذیر می‌توانند از محتویات تله عسل بهره‌برداری کند و از کمبودهای سامانه تولیدی جلوگیری کند. بنابراین این استراتژی استقرار می‌تواند برای گرفتن مقدار زیادی از داده‌ها مورد استفاده قرار گیرد. همان‌طور که گفته شد IDSها در محیط قرار می‌گیرند، جایی که می‌توانند از محتویات تله عسل برای کاهش احتمال تولید هشدارهای کاذب دروغین استفاده کنند. گودال‌ها[1]، مانند تلسکوپ های شبکه[2]، اغلب از این استراتژی استقرار استفاده می‌کنند [۱۴].

۴-۷-۵- سپر تغییر مسیر

از هدایت مجدد پورت[3] یا مسیریابی مجدد ترافیک[4] برای انتقال داده‌های مخرب به تله عسل استفاده می‌کند. این استراتژی برای ارزیابی ترافیک شبکه به فناوری تشخیص نفوذ نیاز دارد. اگر ترافیک جالب باشد، برای جلوگیری از حمله به سامانه تولیدی، به تله عسل محافظ هدایت می‌شود. سپر و شبکه تولید باید محکم یا شل[5] باشند. تله عسل‌ها می‌تواند در فضای آدرس همان شبکه تولید، در زیر شبکه دیگری در کنار شبکه تولید، و یا حتی به‌صورت دسترسی از راه دور حضور داشته باشند. به‌عنوان مثال، Shadow Honeypots [۴۱] به دنبال این استراتژی استقرار از برنامه سایه[6] به عنوان سپری برای مقابله با ترافیک مخرب برای تشخیص مبتنی بر ناهنجاری استفاده می‌کند [۴۲].

۴-۸- نوع منبع

نشان دهنده نوع منبع اطلاعات موجود برای حملات است. اکثر تله عسل‌ها منابع عمومی را ارائه کرده یا از آن تقلید می‌کنند، و هدف آن‌ها شناسایی بیش از یک تکنیک حمله است [۲۱ و ۴۷-۴۳].

۴-۹- برنامه‌های امنیتی

همان‌طور که قبلاً گفته شد، هدف برنامه امنیتی این است که کلیه عملکردهای مربوط به امنیت مانند نظارت بر حمله، پیشگیری، شناسایی، واکنش و profiling[7] را انجام دهد.

۴-۹-۱- رصد کردن حملات

این هدف برای ثبت همه وقایع نفوذ و رفتارهای مخرب است تا تحقیقات بیشتری در این زمینه انجام دهد. دو لایه مهم داده را می‌توان شناسایی کرد: فعالیت شبکه (هر اتصال ورودی و خروجی، بسته و هدر، همچنین میزان بار آن و غیره) و فعالیت سامانه (ضربه زدن به کلید، تماس سامانه، روت کیت ها و غیره) [۴۸-۵۱].

۴-۹-۲- جلوگیری کردن از حمله

هدف آن‌ها بازدارندگی یا جلوگیری از نفوذ است. این عملکرد را می‌توان با رویکردهای مختلف فیلتر کردن داده‌ها، محدودسازی و مهار کردن انجام داد. فیلتر کردن شامل دور انداختن ترافیک داده است. این مورد با استفاده از قوانین فیلتر کردن استفاده می‌شود. دو سازوکار فیلتر کردن وجود دارد، مبتنی بر منبع مقصد و مبتنی بر محتوا [۴، ۱۷ و ۲۲].

۴-۹-۳- تشخیص حمله

این تابع با هدف شناسایی نفوذ و تولید هشدارها انجام می شود.

[1] Sinkholes
[2] Network Telescopes
[3] Port Redirection
[4] Traffic Re-routing

[5] Loosely Coupled or Tightly
[6] Shadow
[7] سرویسی است که ارتباطات ماشین مجازی را زیر نظر دارد و از وضعیت ارتباطی گزارش هایی ارائه می‌دهد.

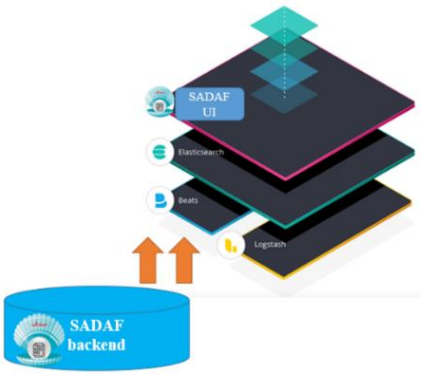
دو رویکرد تشخیص مشترک وجود دارد، مبتنی بر امضاء و مبتنی بر ناهنجاری [۱۲، ۲۲ و ۵۵-۵۲].

۴-۹-۴- پاسخ‌دهی به حملات

این مربوط به اقدامات انجام شده برای پاسخ به حملات و سازگار شدن با حوادث نفوذ بر اساس الزامات از پیش تعریف‌شده است. این تله عسل‌ها می‌توانند دو نوع واکنش نشان دهند: تغییر مسیر ترافیک و تغییر شکل پیکربندی [۴ و ۶۰-۵۶].

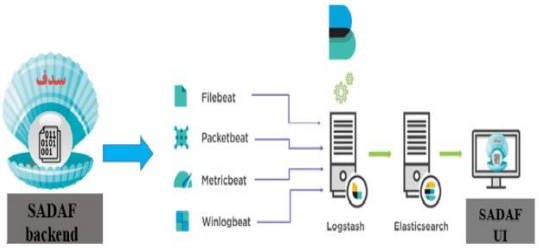
۵- سامانه دفاع فعال سایبری[1]

سامانه دفاع فعال یا سدف[2]، سامانه‌ای است که می‌تواند در شبکه کارگذاری شده و تعاملات مهاجمین با حسگرهای فریب‌کار گذاری شده در شبکه را به نمایش بگذارد و امکان جمع‌آوری و بررسی اطلاعات را به ما بدهد که در شکل (۴)تا (۷)نشان داده شده است. طرح کلی پیاده‌سازی تله سایبری ما از معماری ELK بهره‌برداری می‌کند. این معماری به مانند آنچه در شکل (۲) نشان داده شده است شامل سه لایه کلی است (شکل (۲)). لایه اول شامل دو قسمت Beat و Logstash می‌شود. لایه دوم Elasticsearch است که اطلاعات به‌دست آمده از logfile های به‌دست آمده را به رابط کاربری جهت نمایش و پایش می‌دهد. اغلب ابزارهای تجاری برای ارزیابی و سنجش تهدیدات، از هشدارهای خاصی و یا هشدارهای رخ داده بر روی ماشین‌های مشابه بهره می‌برند. بنابراین به رویکرد جدیدی نیاز است تا اثرات و تهدیدات ناشی از حملات سایبری گسترده شده بر روی چندین ماشین نیز مورد ارزیابی و سنجش قرار گیرد [۶۱].



شکل (۲): معماری لایه‌ای سدف

Beats فرستنده‌های داده منبع آزاد هستند که ما به‌عنوان عامل بر روی سرورها نصب می‌کنیم تا داده‌های عملیاتی را به Elasticsearch بدهد. Beats می‌تواند داده‌ها را مستقیماً به Elasticsearch یا از طریق Logstash ارسال کند، جایی که داده‌ها را از قبل در سدف تجسم کرده‌ایم تا پردازش شود.



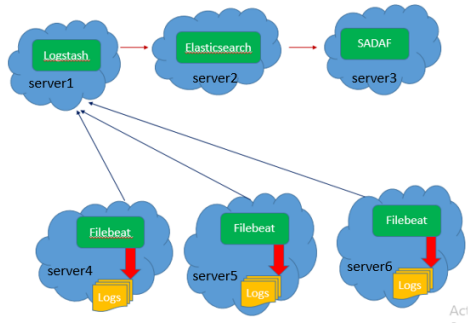
شکل (۳): روند خطی جمع‌آوری اطلاعات سدف

به‌طور کلی Beats حمل کننده داده‌های سبک وزن می‌باشند که ما به‌عنوان عامل بر روی سرور نصب می‌کنیم، تا از این طریق انواع خاصی از داده‌های عملیاتی را به Logstash بفرستیم. سپس ما چندین سرور filebeats داریم. این‌ها پرونده log را می‌خوانند و به Logstash می‌فرستند.

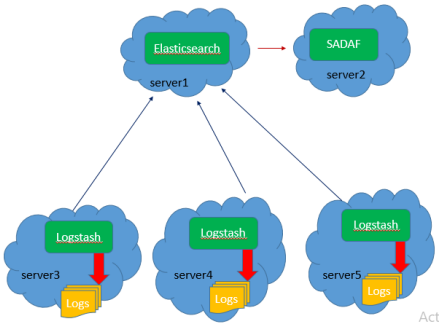
دو راه پیش روی ما جهت طراحی وجود دارد:

- در هر سرور فرستنده داده یک Logstash تعبیه می‌کنیم تا اطلاعات را به Elasticsearch ارسال نماید.
- یک پایگاه مرکزی جهت دریافت filebeatها ایجاد و از آن مکان به ارسال داده به Elasticsearch اقدام نماییم.

با توجه به آن‌که Logstash منابع زیادی را به خود اختصاص می‌دهد و از کارایی و سرعت عمل سامانه می‌کاهد ما از یک پایگاه مرکزی جهت دریافت filebeats استفاده می‌کنیم.



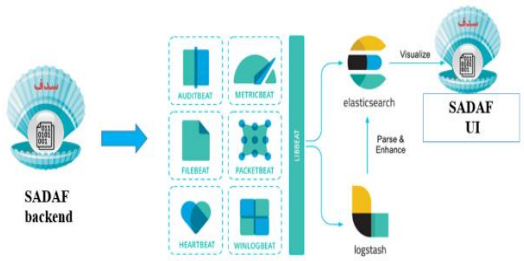
شکل (۴): جمع‌آوری اطلاعات به‌صورت توزیع شده در سدف



شکل (۵): جمع‌آوری اطلاعات به‌صورت متمرکز در سدف

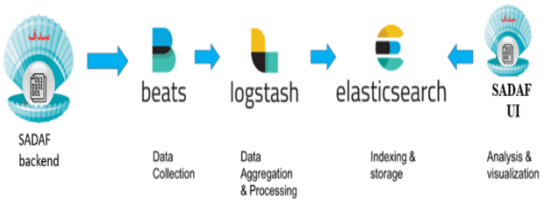
[1] Active Cyber Defense
[2] سامانه دفاع فعال سایبری

اکنون ما از filebeats برای خواندن پرونده ورود به سامانه و ارسال آن به Logstash برای index گذاری به Elastiucsearch استفاده می‌کنیم.



شکل (۶): روند کلی جمع‌آوری اطلاعات سدف

Filebeats با یکسری ماژول‌ها جهت مشاهده و به‌عنوان منابع داده‌های امنیتی به‌صورت مجموعه‌ای ساده برای آنالیز، و مجازی‌سازی به‌صورت یک فرمت عادی log به‌عنوان دستور واحد فرستاده می‌شود.



شکل (۷): روند آماده‌سازی اطلاعات سدف

قرار دادن اطلاعات در کانتینرها بسیار ساده است. به علاوه ویژگی Beats Autodiscover کانتینرهای جدیدی را تشکیل می‌دهد و به‌طور تطبیقی آن‌ها را با ماژول filbeats مناسب ارزیابی می‌کند.

۵-۱- فایل‌های پیکره‌بندی سدف

در داخل، فایل‌های نرم‌افزار، ۲ فایل اصلی جهت پیکربندی و تنظیمات مخصوص docker و سدف قرار دارد که به ترتیب زیر است:

۱- فایل config.toml: در این فایل قسمت‌های مختلف، برای تعریف پورت‌های شبکه ایجاد شده است تا بتوان نوع سرویس‌های فریب کم تعامل و محدوده پورت آن را مشخص نمود.

این فایل دارای ۲ قسمت اصلی است:

الف- service: در این قسمت با توجه به اینکه سرویس‌های telnet، ssh، adb، ethereum، http و https برای سدف تعریف شده است، باید به‌صورت الگوی "service_name".service سرویس مورد نظر را فعال کنیم و اگر تنظیماتی با توجه به نوع سرویس اگر وجود داشت، در این قسمت اعمال کنیم، مثلاً می‌توان نوع پسورد و روت در سرویس ssh را تعریف نمود.

ب- port: در قسمت پورت در ادامه هر سرویس تعریف شده، باید شماره پورتی که می‌خواهیم سرویس‌های فریب در آن قرار گیرد را تعریف می‌کنیم به‌عنوان مثال "tcp/'port_number'" =port تعریف می‌شود.

```
[listener]
type="socket"
[service.ssh-simulator]
type="ssh-simulator"
credentials=["root:root" ,"root:password"]
[[port]]
port="tcp/8022"
services=["ssh-simulator"]
[service.telnet]
type="telnet"
prompt=">>"
[[port]]
port="tcp/5900"
services=["telnet"]
[services.adb]
type="adb"
[[port]]
port="tcp/5902"
services=["adb"]
[services.ethereum]
type "=ethereum"
[[port]]
port="tcp/5903"
services "]=ethereum["
[service.http]
type="http"
server=" Nginx"
[[port]]
port="tcp/5905"
services=["http"]
[services.https]
type="https"
[[port]]
port="tcp/5906"
services=["https"]
[channel.console]
type=" console"
```

```
[channel.elasticsearch]
type=" elasticsearch"
url="http://elasticsearch:9200/honeytrap"
[[filter]]
channel =[" console" ," elasticsearch" ]

[logging]
output="stdout"
level="debug"
```

۲- فایل sadaf_run_in_docker_compose.yml: این فایل از فایل‌های compose برای داکر است تا بتوان یک مجموعه مجزا را با پیکربندی مشخص و یکپارچه ساخت و به اجرا درآورد. در این فایل می‌توان نام imageها، containerها و پورت‌هایی که در اختیار سدف قرار می‌گیرد را مشخص نمود و نوع وابستگی به elk به سدف را تعریف کرد و همچنین تعیین کرد کدام IPها برای فریب در سدف استفاده شود.

۵-۲- توابع و الگوهای شبیه‌سازی

برای اینکه بتوان سرویس‌های فریب را شبیه‌سازی نمود، از زبان Go برای پیاده‌سازی سرویس‌های کم تعامل telnet، ssh، adb، ethereum، http و https استفاده شده است که هر کدام با پسوند go. در سدف قرار گرفته‌اند. توابع و الگوی رفتاری سرویس‌ها در ادامه به آن‌ها پرداخته می‌شود.

۵-۲-۱- سرویس ADB

سرویس android debug bridge از سرویس‌ها مورد استفاده برای کنترل و برنامه‌نویسی اندروید است که به‌صورت فریب کم تعامل این سرویس شبیه‌سازی شده است و نشان دهنده حضور تلفن همراه هوشمند در شبکه است.

جدول (۱): سرویس کنترل و برنامه اندروید

عملکرد تابع در adb	نام تابع
این تابع وظیفه ایجاد سرویس adb در شبکه را دارد و ساختار کلی فریب adb را تعریف می‌کند.	**adbService**
وظیفه این تابع بررسی تعاملات صورت گرفته با سرویس adb و پاسخ به آن است، به‌عنوان نمونه در این سامانه یک تلفن همراه سامسونگ مدل SM-G950F تعریف شده است.	**makeAdbPacket**

۵-۲-۲- سرویس Ethereum

یکی از مزایای استفاده از سدف، شبیه‌سازی شبکه بلاکچینی است، و این‌گونه به مهاجم القا می‌کند که در شبکه سازوکار بلاکچین وجود دارد و این رغبت را برای مهاجم ایجاد می‌کند که با آن تعامل داشته باشد. درنتیجه یکی از مطرح‌ترین آن‌ها یعنی شبکه Ethereum شبیه‌سازی‌شده است.

جدول (۲): سرویس Ethereum

عملکرد تابع در ethereum	نام تابع
این تابع وظیفه ایجاد سرویس ethereum در شبکه را دارد و ساختار کلی فریب ethereum را تعریف می‌کند.	**ethereumService**
در این قسمت انواع کاربران، نوع کدینگ، نوع ماینر، میزان تبادل اطلاعات و سایر ویژگی‌های اختصاصی Ethereum به‌صورت فریب ایجاد می‌شود و پکت‌های اشتباهی به مهاجم ارسال می‌کند.	**ethereumMethods**

۵-۲-۳- سرویس Telnet

این سرویس از سرویس‌هایی تبادل اطلاعات رمز نشده است که برای مهاجمان پر ریسک و خطرپذیر می‌تواند مناسب باشد تا با این سامانه تعامل برقرار کنند.

جدول (۳): سرویس Telnet

عملکرد تابع در telnet	نام تابع
این تابع وظیفه ایجاد سرویس telnet در شبکه را دارد و ساختار کلی فریب telnet را تعریف می‌کند.	**telnetService**
این دو، وظیفه دریافت نام کاربری و پسورد درخواستی مهاجم را دارند و بعد از درخواست، مهاجم را به یک محیط خالی هدایت می‌کند تا در صورت قراردهی کد آلوده آن‌ها را ثبت و ضبط کند.	**annel, term setCh**

۵-۲-۴- سرویس SSH

در این سرویس، رمزنگاری صورت گرفته و ورود به این سرویس سخت است، هر چند با ایجاد نام کاربری و پسورد چند لایه و قابل پیش‌بینی، این امکان را در سدف ایجاد کردیم که مهاجم بتواند با آن تعامل برقرار کند و عملیات حمله را پیاده‌سازی کند.

جدول (۴): سرویس SSH

عملکرد تابع در ssh	نام تابع
این تابع وظیفه ایجاد سرویس ssh در شبکه را دارد و ساختار کلی فریب ssh را تعریف می‌کند.	**sshSimulatorService**
در این دو تابع پیکربندی شبکه ssh را انجام می‌دهیم، نام کاربری و رمز عبور را مشخص می‌کنیم، همچنین مهاجم را با شبکه‌ای رمز شده از فریب برخوردار می‌کنیم و payloadها و دستورات مهاجم ثبت و ضبط می‌کنیم.	**PayloadDecoder, config**

۵-۲-۵- سرویس HTTP و HTTPS

در این سرویس‌ها، سرورهایی مانند nginx یا Apache در شبکه ایجاد می‌کنیم تا مهاجم قصد حمله به آن را داشته باشد و با آن تعامل برقرار کند.

جدول (۵): سرویس HTTP و HTTPS

عملکرد تابع در http و https	نام تابع
این تابع وظیفه ایجاد سرویس در شبکه را دارد و ساختار کلی فریب را تعریف می‌کند.	httpService, httpsService
شبیه‌سازی سرورهای مختلف در این قسمت تعریف می‌شود و رمزنگاری، ایجاد کوکی‌های مختلف و پاسخ کم تعامل با مهاجم را انجام می‌دهد.	Servicer, Cookies

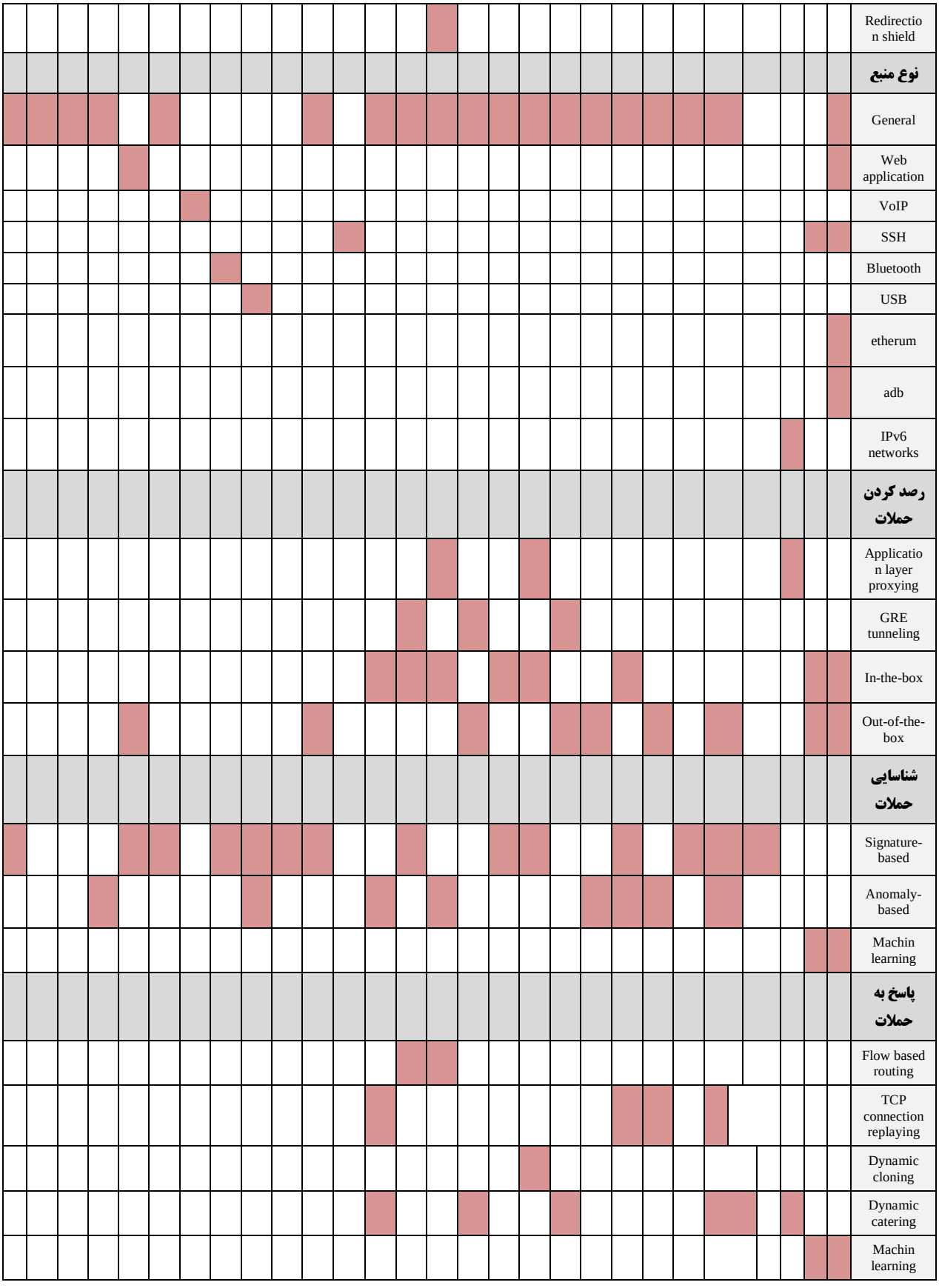
در سدف امکان مشاهده و بررسی تعاملات صورت گرفته با سرویس‌های قرار داده شده در شبکه، بر اساس برچسب‌های اطلاعاتی در بازه‌های زمانی دلخواه را خواهیم داشت. همچنین می‌توان به‌صورت دسته‌بندی شده رویدادهای صورت گرفته را مشاهده و بر اساس آن نمودارهای مختلفی را جهت آنالیز مشاهده نمود.

از دیگر امکانات دیگر سدف می‌توان به فیلتر کردن اطلاعات با توجه به رویدادهای موجود در شبکه فریب اشاره نمود، و درنهایت به شناسایی ناهنجاری‌ها و بررسی اطلاعات ایجاد شده در شبکه فریب بر اساس IP از امکاناتی است که در این سامانه تعبیه شده است.

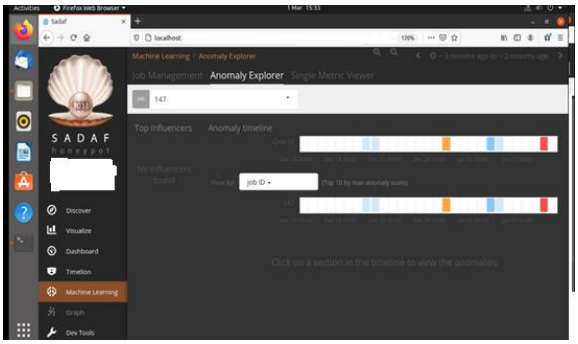
سعی شده در سامانه سدف سرویس هایی در کنار هم قرار گیرند که ترکیبی جدید را ارائه کرده تا کاربر در محیطی انعطاف پذیر به مدیریت فضای مورد استفاده بپردازد. در جدول‌های (۱ الی ۵) به معرفی سرویس‌های به‌کار رفته پرداخته‌ایم و عملکرد توابع تشریح شده است. سپس در ادامه طبق دسته‌بندی پیشنهادی مقاله که در شکل (۱) نشان داده شد، از مزایای بارز سدف که در جدول (۶) نمایان است، به‌کارگیری سرویس‌های متعددی است که در محیطی انعطاف پذیر جهت استفاده آسان در ساختار شبکه‌های مختلف می‌توان از آن بهره برد. دیگر مزیت استفاده از سدف استفاده از روش‌های نوین جمع‌آوری log و پاسخ‌دهی مناسب به مهاجم با استفاده از یادگیری ماشین است.

جدول (۶): مقایسه سدف با دیگر سامانه‌های موجود

مشخصات	سامانه همکار از سامانه‌های مستقل				سامانه‌های فریب همکارانه																سامانه‌های فریب مستقل												
	سدف (SDF)		Hyhoneydvg	Io TPOT	vm honeymon	li system	sgmet	honeybrid	sweetbait	GA	artail system	gen2 honey	potemkin	shadow honeypot	collapsar	bailey system	cowrie	cuckoo	conpot	ghost	bluepot	artemisa	dianaea	glastopf	argos	honeyd	net telsco	specter
تعامل پذیری																												
کم	■	■	■	■		■	■	■	■	■	■					■			■			■		■		■	■	■
متوسط																	■			■	■		■					
زیاد	■		■	■	■	■	■	■	■	■	■	■	■	■	■	■		■							■			
محیط پیاده‌سازی																												
مجازی	■	■															■	■			■	■	■			■	■	■
فیزیکی		■		■					■	■		■																
مقیاس پذیری																												
مقیاس پذیر	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		■		■	■	■	■		■	■	■
مقیاس ناپذیر																		■		■			■		■			
انطباق پذیری																												
ایستا	■	■										■					■	■	■	■	■	■	■		■		■	■
پویا	■		■	■	■	■	■	■	■	■	■	■	■	■	■	■						■		■		■		
نقش																												
سرور	■	■	■	■		■	■	■	■	■	■	■	■	■	■	■		■			■	■	■	■	■	■	■	■
کلاینت																	■		■	■								
استراتژی استقرار																												
Sacrificial lamb			■	■			■	■	■	■	■	■	■		■	■		■		■					■			
Deception ports	■	■		■	■								■				■		■		■	■	■	■				■
Proximity decoys			■			■	■	■	■		■	■				■						■				■		
Minefield										■			■		■												■	


Redirection shield
نوع منبع
General
Web application
VoIP
SSH
Bluetooth
USB
etherum
adb
IPv6 networks
رصد کردن حملات
Application layer proxying
GRE tunneling
In-the-box
Out-of-the-box
شناسایی حملات
Signature-based
Anomaly-based
Machin learning
پاسخ به حملات
Flow based routing
TCP connection replaying
Dynamic cloning
Dynamic catering
Machin learning

همان‌طورکه در شکل (۸) مشاهده می‌شود بر اساس رویدادهای متفاوت می‌توان شبکه را رصد نمود.



شکل (۸): رصد و پایش سدف

۶- نتیجه‌گیری

تله عسل به‌عنوان فناوری که به سرعت در حال توسعه است، به یک موضوع مهم در زمینه امنیت رایانه و شبکه تبدیل شده‌اند. سدف (سامانه دفاع فعال) سایبری دارای مؤلفه کم تعامل برای سرویس‌های HTTP، HTTPS، SSH، Telnet، ADB و Blockchain است که بر پایه زبان Golang نوشته شده و در محیط Docker پیاده‌سازی شده است. از قابلیت‌های اجمالی آن می‌توان به پیاده‌سازی در بستر شبکه‌های سازمانی اشاره نمود. با سدف می‌توان انواع سرویس‌های مختلف فریب را ایجاد و به شناسایی هکرهای درون سازمانی و برون سازمانی پرداخت به گونه‌ای که رصد و پایش بلادرنگ با توجه به تعاملات صورت گرفته به کاربر نمایش داده می‌شود. به وسیله رابط کاربری می‌توان نمایش عملکرد سرویس‌های فریب را مشاهده کرد و به‌صورت نموداری تاریخچه اطلاعات (حجم، IPها، سرویس‌ها)، فیلتر کردن اطلاعات (به‌عنوان مثال اطلاعات دستورات SSH) و نمایش ناهنجاری را بر اساس IP مشاهده نمود. از مزایای سدف می‌توان به طراحی سبک، توسعه کم هزینه به وسیله Docker، سهولت مدیریت با بهره‌گیری از سامانه جمع‌آوری Log هوشمند و رابط کاربری مناسب اشاره نمود. درنهایت با ایجاد سامانه سدف، ما به افزایش هوش سایبری دست یافتیم.

۷- مراجع

[1] S. Brandes, "The Newest Warfighting Domain: Cyberspace," *Synesis: A J. Sci. , Technol. , Ethics, Policy*, vol 4, bll G90-95, 2013.

[2] M. Fossi., "Symantec Internet Security Threat Report Trends for 2010,", Volume XVI, 2011.

[3] G. J. Rattray, "An Environmental Approach to Understanding Cyberpower," Cyberpower and National Security, vol 10, National Defense University Press Washington, DC, bll 253–274, 2009.

[4] M. Bailey, E. Cooke, D. Watson, F. Jahanian, and N. Provos, "A Hybrid Honeypot Architecture for Scalable Network Monitoring," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep. CSE-TR-499-04, 2004.

[5] H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, "A Hybrid Honeypot Framework for Improving Intrusion Detection Systems in Protecting Organizational Networks," Computers & Security, vol. 25, no. 4, pp. 274–288, 2006.

[6] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A Survey on Honeypot Software and Data Analysis," arXiv preprint arXiv:1608. 06249, 2016.

[7] T. K. Lengyel, J. Neumann, S. Maresca, B. D. Payne, and A. Kiayias, "Virtual Machine Introspection in a Hybrid Honeypot Architecture," In CSET, 2012.

[8] L. Spitzner, "The Honeynet Project: Trapping the Hackers," IEEE Security & Privacy, vol. 1, no. 2, pp. 15–23, 2003.

[9] L. Spitzner, "Honeypots: Catching the Insider Ihreat," In 19th Annual Computer Security Applications Conference, Proceedings, 2003, pp. 170–179.

[10] B. Cheswick, "An Evening with Berferd in which a Cracker is Lured, Endured, and Studied," In Proc. Winter USENIX Conference, San Francisco, 1992, pp. 20–24.

[11] C. Stoll, The cuckoo's egg: tracking a spy through the maze of computer espionage. Simon and Schuster, 2005.

[12] G. Portokalidis, A. Slowinska, and H. Bos, "Argos: An Emulator for Fingerprinting Zero-day Attacks for Advertised Honeypots with Automatic Signature Generation," ACM SIGOPS Operating Systems Review, vol. 40, no. 4, pp. 15–27, 2006.

[13] R. Rajabioun, "Cuckoo Optimization Algorithm," Applied Soft Computing, vol. 11, no. 8, pp. 5508–5518, 2011.

[14] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Network Telescopes: Technical Report," Cooperative Association for Internet Data Analysis (CAIDA), 2004.

[15] A. Kirkby, "Honeynet Phase Two: Knowing Your Enemy More", Computer Fraud & Security, vol. 2001, no. 12, pp. 8–9, 2001.

[16] D. Song, "A snapshot of global Internet worm activity," The 14th Annual FIRST Conference on Computer Security Incident Handling and Response, 2002.

[17] V. Yegneswaran, P. Barford, and D. Plonka, "On the Design and Use of Internet Sinks for Network Abuse Monitoring," In International Workshop on Recent Advances in Intrusion Detection, 2004, pp. 146–165.

[18] K. M. Aghaei, S. Farshchi, and H. Shirazi, "A New Architecture for Impact Projection of Cyber-attacks Based on High Level Information Fusion in Cyber Command and Control," Volume 9, No. 36, pp. 125-

and Virtualization Management: Standards and the Cloud (SVM), pp. 1–8, 2011.

[33] W. Fan, D. Fernández, and Z. Du, "Versatile Virtual Honeynet Management Framework," IET Information Security, vol. 11, no. 1, pp. 38–45, 2017.

[34] W. Y. Chin, E. P. Markatos, S. Antonatos, and S. Ioannidis, "HoneyLab: Large-scale Honeypot Deployment and Resource Sharing," Third International Conference on Network and System Security, pp. 381–388, 2009.

[35] B. Sobesto, M. Cukier, M. A. Hiltunen, D. Kormann, G. Vesonder, and R. Berthier, "DarkNOC: Dashboard for Honeypot Management," In LISA, 2011.

[36] W. Han, Z. Zhao, A. Doupé, and G. J. Ahn, "Honeymix: Toward SDN-based Intelligent Honeynet," In Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 1–6, 2016.

[37] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A Survey on Automated Dynamic Malware-analysis Techniques and Tools," ACM computing surveys (CSUR), vol. 44, no. 2, pp. 1–42, 2008.

[38] L. Spitzner, "Know Your Enemy: Genii Honeynets," The Honeynet Alliance, 2005.

[39] W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an Anatomic View to Investigate Honeypot Systems: A Survey," IEEE Systems Journal, vol. 12, no. 4, pp. 3906–3919, 2017.

[40] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis, "Detecting Targeted Attacks Using Shadow Honeypots," 14th USENIX Security Symposium, 2005.

[41] S. Schindler, B. Schnor, and T. Scheffler, "Hyhoneydv6: A Hybrid Honeypot Architecture for IPV6 Networks," International Journal of Intelligent Computing Research, vol. 6, No. 2, pp. 562-570, 2015.

[42] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the Rise of IoT Compromises," In 9th ${USENIX} Workshop on Offensive Technologies ({WOOT}$ 15), 2015.

[43] A. Pashaei, M. E. Akbari, M. Z. Lighvan, and H. A. Teymorzade, "Improving the IDS Performance through Early Detection Approach in Local Area Networks Using Industrial Control Systems of Honeypot," In 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), pp. 1–5, 2020.

[44] A. Podhradsky, C. Casey, and P. Ceretti, "The Bluetooth Honeypot Project: Measuring and Managing Bluetooth Risks in the Workplace," International Journal of Interdisciplinary Telecommunications and Networking (IJITN), vol. 4, no. 3, pp. 1–22, 2012.

140, 2015.

[19] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, D. Watson, and Others, "The Internet MotionS-a Distributed Blackhole Monitoring System," In NDSS, 2005.

[20] N. Provos and Others, "A Virtual Honeypot Framework," In USENIX Security Symposium, vol. 173, pp. 1–14, 2004.

[21] B. Mphago, O. Bagwasi, B. Phofuetsile, and H. Hlomani, "Deception in Dynamic Web Application Honeypots: Case of Glastopf," In Proceedings of the International Conference on Security and Management (SAM), p. 104, 2015.

[22] W. Schulze, E. D. Schulze, I. Schulze, and R. Oren, "Quantification of Insect Nitrogen Utilization by the Venus Fly Trap Dionaea Muscipula Catching Prey with Highly Variable Isotope Signatures," Journal of experimental botany, vol. 52, no. 358, pp. 1041–1049, 2001.

[23] L. Spitzner, "Specter: A Commercial Honeypot Solution for Windows," Acesso em, vol. 26, no. 08, 2003.

[24] S. Poeplau and J. Gassen, "A Honeypot for Arbitrary Malware on USB Storage Devices," 7th International Conference on Risks and Security of Internet and Systems (CRiSIS), pp. 1–8, 2012.

[25] N. Provos and T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Pearson Education, 2007.

[26] L. K. Yan, "Virtual Honeynets Revisited," In Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, pp. 232–239, 2005.

[27] A. Capalik, "Next-generation Honeynet Technology with Real-time Forensics for US Defense," In MILCOM 2007-IEEE Military Communications Conference, pp. 1–7, 2007.

[28] N. Memari, S. J. B. Hashim, and K. B. Samsudin, "Towards Virtual Honeynet Based on LXC Virtualization," IEEE Region 10 Symposium, pp. 496–501, 2014.

[29] D. Sever and T. Kišasondi, "Efficiency and Security of Docker Based Honeypot Systems," 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1167–1173, 2018.

[30] F. Galán and D. Fernández, "Use of VNUML in Virtual Honeynets Deployment," IX Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Barcelona, Spain, 2006.

[31] F. Stumpf, A. Görlach, F. Homann, and L. Brückner, "NoSE-building Virtual Honeynets Made Easy," In Proceedings of the 12th International Linux System Technology Conference, Hamburg, Germany, 2005.

[32] D. Fernández , "Distributed Virtual Scenarios Over Multi-host Linux Environments," 5th International DMTF Academic Alliance Workshop on Systems

[45] R. Do Carmo, M. Nassar, and O. Festor, "Artemisa: An Open-source Honeypot Back-end to Support Security in VoIP Domains," In 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, pp. 361–368, 2011.

[46] L. Spitzner, "Know Your Enemy: Sebek2 A Kernel Based Data Capture Tool," Recuperado a partir de http://www. honeynet. org, 2003.

[47] C. Song, B. Hay, and J. Zhuge, "Know Your Tools: Qebek--Conceal the Monitoring," The Honeynet Project (www. honeynet. org/sites/default/files/files/KYT-Qebek-final_v1. pdf), 2010.

[48] C. Willems, T. Holz, and F. Freiling, "Toward Automated Dynamic Malware Analysis Using Cwsandbox," IEEE Security & Privacy, vol. 5, no. 2, pp. 32–39, 2007.

[49] X. Jiang and X. Wang, "'Out-of-the-box' Monitoring of VM-based High-Interaction Honeypots," In International Workshop on Recent Advances in Intrusion Detection, pp. 198–218, 2007.

[50] J. Newsome and D. X. Song, "Dynamic Taint Analysis for Automatic Detection, Analysis, and SignatureGeneration of Exploits on Commodity Software," In NDSS, vol. 5, pp. 3–4, 2005.

[51] C. Kreibich and J. Crowcroft, "Honeycomb: Creating Intrusion Detection Signatures Using Honeypots," ACM SIGCOMM Computer Communication Review, vol. 34, no. 1, pp. 51–56, 2004.

[52] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," Eai Endorsed Transactions on Security and Safety, vol. 3, no. 9, p. e2, 2016.

[53] R. Sekar, A.Gupta and S.Zhou. , "Specification-based Anomaly Detection: A New Approach for Detecting Network Intrusions", In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 265–274, 2002.

[54] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," ACM Transactions on Computer Systems (TOCS), vol. 18, no. 3, pp. 263–297, 2000.

[55] C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang, "Enabling Security Functions with SDN: A Feasibility Study," Computer Networks, vol. 85, pp. 19–35, 2015.

[56] R. Berthier and M. Cukier, "Honeybrid: A Hybrid Honeypot Architecture," In USENIX Security Symposium, vol. 2008, 2008.

[57] R. Kundel P. Stiegele, D. Tran, J. Zobel, O.Abboud, R. Hark and R.Steinmetz, "User Space Packet Schedulers: Towards Rapid Prototyping of Queue-Management Algorithms," Electronic Communications of the EASST, vol. 80, 2021.

[58] Y.-D. Lin, T.-B. Shih, Y.-S. Wu, and Y.-C. Lai, "Secure and Transparent Network Traffic Replay, Redirect, and Relay in a Dynamic Malware Analysis Environment," Security and Communication Networks, vol. 7, no. 3, pp. 626–640, 2014.