نشریه علمی «علوم و فناوری‌های پدافند نوین»
سال سیزدهم، شماره ۱، بهار ۱۴۰۱؛ ص ۶۳- ۵۳

علمی- پژوهشی

طراحی و پیاده‌سازی سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی و شبکه همکارانه

کوروش داداش تبار احمدی*
استادیار دانشگاه صنعتی مالک اشتر، تهران
(دریافت: ۱۴۰۰/۸/۱۴، پذیرش: ۱۴۰۱/۲/۲۵)

چکیده

امروزه در شبکه‌های رایانه‌ای و به ویژه شبکه‌های محلی و زیرساخت فضای سایبری تعداد حملات سایبری به شدت افزایش پیدا کرده است و این حملات بسیار پیچیده‌تر شده است. برای تشخیص اینگونه از حملات سامانه‌های تشخیص نفوذ (IDS) و ناهنجاری‌های بسیاری در حال طراحی و توسعه هستند. در سال‌های اخیر با معرفی زنجیره بلوکی که یک پایگاه داده امن توزیع شده در شبکه‌های غیر متمرکز است، تحولی شگرف در شبکه‌های رایانه‌ای رخ داده است. این فناوری قابلیت ایجاد اجماع و اعتماد بین سامانه‌های تشخیص نفوذ را دارد به‌طوری که منجر به پایداری بیشتر شبکه همکاری بین سامانه‌های IDS نیز خواهد شد. بنابراین ترکیب این دو سامانه می‌تواند عملکرد بهتری را نسبت به نسل-های قبل IDS داشته باشد. فناوری زنجیره بلوکی به دلیل ویژگی‌هایی مانند حفظ صحت داده، دسترسی پذیری و مدیریت غیر متمرکز در دنیای رمزنگاری و امنیت شبکه کاربردهای زیادی دارد. امنیت اطلاعات در این شبکه برای کارکرد صحیح سامانه‌های تشخیص نفوذ و دیواره-های آتش ضروری است. چنین ویژگی‌هایی را می‌توان در شبکه هایپرلجر فبریک یافت. این شبکه با توجه به استفاده از رمزنگاری نامتقارن و زنجیره بلوکی اطلاعات را به‌صورت امن و با سرعت زیاد در شبکه انتقال و ثبت می‌نماید. در این تحقیق با استفاده از فناوری زنجیره بلوکی در تلاش برای ایجاد یک شبکه‌ای از گره‌های IDS هستیم که در آن هر گره‌ای بتواند قوانین مورد نظر خود را به پایگاه داده زنجیره بلوکی اضافه کند. با این کار سایر گره‌های زنجیره پس از اجماع و همگام‌سازی صورت گرفته از قوانین سایر گره‌ها برای تقویت سامانه تشخیص خود استفاده می‌کنند. همچنین به علت ویژگی غیر متمرکز بودن زنجیره بلوکی نیاز به یک کنترل هویت مرکزی در جهت تأیید/عدم تأیید گره‌ها و قوانین افزوده شده به پایگاه داده، مورد نیاز نیست و سازوکارهای اجماع این کار را انجام می‌دهند. در این سازوکار از کاهش سرعت اجرای هایپرلجر فبریک و ثبت تراکنش‌ها جلوگیری شده است و راندمان بالای رابط کاربری باعث بهتر شدن سامانه تشخیص نفوذ شده است.

کلیدواژه‌ها: زنجیره بلوکی، قراردادهای هوشمند، سامانه تشخیص نفوذ، هایپرلجر فبریک، شبکه همکارانه

Designing and Implementation of Blockchain-Based Collaborative Intrusion Detection System

K. Dadashtabar Ahmadi
Malek Ashtar University of Technology, Tehran
(Received: 05/11/2021, Accepted: 15/05/2022)

Abstract

Today, the number of cyberattacks on computer networks, especially local area networks and the Internet, has increased dramatically, and these attacks have become much more complex. Many intrusion detection systems (IDS) and signatures are being designed and developed to detect these types of attacks. In recent years, with the introduction of blockchain, which is a secure distributed database in decentralized networks, a dramatic change has occurred in computer networks. This technology can create consensus and trust between intrusion detection systems to increase the stability of the cooperating networks among IDS systems. Therefore, the combination of these two systems can have better performance than previous generations of IDS. Blockchain technology has many applications in the world of cryptography and network security due to features such as data integrity, availability, and decentralized management. Information security in this network is essential for the proper functioning of intrusion detection systems and firewalls. These features can be found in the Hyperledger Fabric network. Due to the use of asymmetric encryption and blockchain, this network transmits and records information securely and quickly in the network. In this project, using blockchain technology, we are trying to create a network of IDS nodes where each node can add its own rules to the blockchain database. In this way, the other nodes of the chain use the rules of other nodes to improve their intrusion detection system efficiency after consensus and synchronization. Also, due to the decentralized nature of the blockchain, a central identity control is not required to approve/disapprove the nodes and rules added to the database, and consensus mechanisms do this.

Keywords: Blockchain, Smart Contract, Intrusion Detection System, Hyperledger Fabric, Collaborative Network

*Corresponding Author E-mail: Dadashtabar@mut.ac.ir

۱. مقدمه

در حال حاضر، حملات سایبری پیچیده‌تر و پیشرفته‌تر شده‌اند. برای کمک به شناسایی به موقع نفوذها، سامانه‌های تشخیص نفوذ[1] به‌طور گسترده در انواع مختلف شبکه اجرا می‌شوند. بر اساس مکان مستقر شده، یک سامانه تشخیص نفوذ را می‌توان در دو دسته مبتنی بر میزبان و یا مبتنی بر شبکه تقسیم‌بندی کرد. مورد اول به‌طور عمده ویژگی‌های یک سامانه محلی و رویدادهای سامانه را در یک میزبان برای فعالیت‌های مخرب کنترل می‌کند. مورد دوم با مقایسه تضاد ترافیک شبکه و تجزیه و تحلیل پروتکل-های شبکه و بارهای آن‌ها برای وقایع مشکوک بررسی می‌کند. چنین سامانه‌های تشخیصی، توانایی خود را در محافظت از شبکه-های مستقر در برابر آسیب‌های سایبری ثابت کرده‌اند. با این حال، با افزایش تعداد و پیچیدگی نفوذها، یک سامانه تشخیص نفوذ منفرد[2] یا ایزوله شده در بسیاری از سناریوها بی‌اثر می‌شود، یعنی می‌توان با حملات پیشرفته از آن عبور کرد. بدون شناسایی به موقع حملات سایبری، کل شبکه با خسارات مختلفی مواجه می-شود، حتی فلج کل شبکه نیز امکان پذیر است.

برای افزایش قابلیت تشخیص سامانه‌های تشخیص نفوذ، سامانه‌ها یا شبکه‌های تشخیص نفوذ مشارکتی[3] طراحی شده‌اند که به گره‌های سامانه‌های تشخیص نفوذ اجازه می‌دهند تا اطلاعات مورد نیاز را با یکدیگر جمع‌آوری و مبادله کنند. به‌عنوان مثال، با جمع‌آوری مشخصات ترافیک از حسگرهای مختلف تشخیصی، یک سرور مرکزی نسبت به ناهنجاری‌های شبکه حساسیت بیشتری نسبت به یک سامانه تشخیص نفوذ منفرد دارد. چارچوب‌های مشارکتی تشخیص نفوذ به دلیل افزایش عملکرد شناسایی، به‌طور گسترده‌ای در سازمان‌های مختلف پذیرفته شده و به‌کار گرفته شده‌اند، اما هنوز دو مسئله مهم باقی مانده است: اشتراک داده‌ها[4] و محاسبه اعتماد. در عمل، یک سرور مرکزی اغلب به‌عنوان یک نقطه قابل اعتماد برای کمک به مدیریت اشتراک داده‌ها و محاسبه اعتماد برای سامانه‌های تشخیص نفوذ در بین طرف‌های همکاری استفاده می‌شود، حتی اگر این سرور به ضعیف‌ترین نقطه برای امنیت شبکه تبدیل شود. برای رفع چالش‌های فوق، به فناوری‌های جدیدی در زمینه تشخیص نفوذ نیاز است. در سال‌های اخیر، فناوری زنجیره بلوکی[5] به دلیل نوآوری بسیار مورد توجه دانشگاهیان و صنعت قرار گرفته است، که به طرفین بی اعتماد متقابل اجازه می‌دهد داده‌های مالی را بدون نیاز به شخص ثالث قابل اعتماد مبادله کنند. این ویژگی دقیقاً مطلوب برای شناسایی مشارکتی است که فرصتی را برای حل مشکلات مربوط به اشتراک داده و مدیریت اعتماد فراهم می‌کند [۱].

هدف از این پروژه طراحی و پیاده‌سازی یک سامانه تشخیص نفوذ است که در آن گره‌های سامانه در بستر زنجیره بلوکی با یکدیگر در تعامل و در حال همکاری هستند. این سامانه می‌تواند صحت داده[6]، دسترسی‌پذیری[7] و مدیریت غیر متمرکز[8] سامانه‌های تشخیص نفوذ را برای شبکه‌های رایانه‌ای فراهم آورد. این سامانه کارایی بالایی در اینترنت اشیاء نیز می‌تواند داشته باشد که هر گره‌ای در آن دارای یک سامانه تشخیص نفوذ مخصوص به خود است و این در حالی است که شبکه اینترنت اشیاء دارای میلیون‌ها گره کوچک و بزرگ است و مدیریت متمرکز آن مشکل و ناپایدار است. عملکرد این سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی در یک شبکه آزمایشگاهی مورد ارزیابی قرار خواهد گرفت.

در ادامه این مقاله و در بخش دوم به بررسی روش تحقیق پرداخته شده و مدل ارائه شده برای سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی ارائه می‌گردد. بخش سوم به نتایج ارزیابی پرداخته و درنهایت بخش چهارم به نتیجه‌گیری مباحث این نوشتار می‌پردازد.

۲. روش تحقیق

در این بخش ابتدا مفاهیم مورد نیاز جهت ارائه مدل طراحی شده برای سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی توضیح داده می‌شود. پس از آن مدل ارائه شده برای سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی ارائه می‌گردد.

۲-۱. مفاهیم اولیه

۲-۱-۱. زنجیره بلوکی

زنجیره بلوکی یک دفترکل[9] توزیع شده[10] غیر متمرکز و بدون نیاز به اعتمادسازی است که توسط همه اعضاء نگهداری می‌شود. بعضی از اعضاء به‌عنوان معدن‌کاو[11] در این شبکه نظیر به نظیر فعالیت می‌کنند و یک نسخه کامل از زنجیره بلوکی را نگه می‌دارند. این اعضاء تمام تراکنش‌هایی که توسط کاربران امضاء و ارسال شده است را جمع‌آوری می‌کنند و پس از اعتبارسنجی امضاءها، آن‌ها را در داخل یک بلوک قرار می‌دهند. هر بلوک اطلاعاتی از تراکنش‌ها را به همراه چکیده[12] بلوک قبلی نگه می‌دارد و لیست ترتیبی از این بلوک‌ها یک زنجیره بلوکی را تشکیل می‌دهند. زمانی که یک بلوک توسط یک معدن‌کاو تولید می‌شود، تمام معدن‌کاوان موجود در شبکه باید برای قبول کردن و اضافه کردن آن به زنجیره به اجماع کلی برسند. بدین ترتیب یک

[1] Intrusion Detection System
[2] Individual
[3] Collaborative IDS
[4] Data Sharing
[5] Blockchain

[6] Data Integrity
[7] Availability
[8] Distributed
[9] Ledger
[10] Distributed Ledger
[11] Miner
[12] Hash

سازوکار اجماع توزیع شده در شبکه زنجیره بلوکی مورد نیاز است [۲]. در حالت ایده‌آل، تمام گره‌ها برای رسیدن به اجماع شروع به رأی دادن می‌کنند. در حوزه رمزارزها مسئله اجماع یک عنصر مهم از هر شبکه زنجیره بلوکی است. الگوریتم‌های اجماع وظیفه حفظ یکپارچگی و امنیت را در این سامانه توزیع شده برعهده دارند. در شبکه زنجیره بلوکی عمومی به دلیل عدم وجود یک مقام مرکزی برای مدیریت و اعتبارسنجی داده‌ها، گره‌های توزیع شده باید بر روی هر داده‌ای به توافق برسند. سازوکارهای مختلفی برای رسیدن به اجماع کلی ارائه شده است که هر یک دارای مزایا و معایب خاص خود هستند. پراستفاده‌ترین و معمول‌ترین این الگوریتم‌ها اثبات کار و اثبات سهم است که اکثر رمزارزها از این دو الگوریتم بهره برده‌اند [۳].

۲-۱-۲. رمزارزها و قرارداد هوشمند

در چند سال اخیر ارزهای دیجیتالی مختلفی با عنوان رمزارز[1] ارائه شده است که اساس کار آن‌ها فناوری زنجیره بلوکی و رمزنگاری است. این فناوری به کاربران اجازه می‌دهد تا در یک شبکه عمومی، توزیع شده و غیر قابل اعتماد اقدام به تبادل ارزهای خود کنند. چهار مؤلفه امنیت یعنی انکارناپذیری[2]، دسترسی‌پذیری، صحت داده و احراز هویت[3] در اکثر رمزارزهای معرفی شده وجود دارد [۱]. در مقابل، محرمانگی و حریم خصوصی ویژگی‌هایی هستند که در تضاد با فناوری زنجیره بلوکی قرار دارند و تأمین آن‌ها سربار زیادی به شبکه تحمیل می‌کند [۴].

پس از رمزارز بیت‌کوین[4] [۵] که تراکنش مالی شفاف و بدون حریم خصوصی را برای افراد در شبکه فراهم کرد، رمزارز زی‌کَش[5] [۶] ارائه گردید. این رمزارز شامل هسته بیت‌کوین است و امکان انجام تراکنش‌های مالی خصوصی را برای کاربران فراهم می‌کند. شایان ذکر است که رمزارز بیت‌کوین سطحی از ناشناس بودن را برای کاربران ایجاد می‌کند اما رمزگذاری نشدن تراکنش‌ها باعث می‌شود امکان نقض حریم خصوصی کاربران با تحلیل تراکنش‌ها وجود داشته باشد. در بیت‌کوین و زی‌کش امکان اجرای قرارداد هوشمند[6] وجود ندارد و به همین دلیل گنجانده شدن این ویژگی در رمزارز اتریوم[7] [۷] باعث رشد سریع و فراگیر شدن این رمزارز در بین عموم مردم گردید است. قراردادهای هوشمند قابلیتی است که در آن کاربران می‌توانند قطعه کدهای تورینگ-کامل[8] را بر روی زنجیره بلوکی اجرا کنند. نتیجه این قطعه کدها در بلوک‌ها ثبت و نگهداری می‌شوند. در این سامانه، تمامی اعضاء به گزارش‌ها و داده‌های قراردادهای هوشمند دسترسی دارند. یک تراکنش تنها در صورتی در زنجیره بلوکی ثبت می‌گردد که مورد توافق اکثریت اعضای شبکه قرار گرفته باشد. این نحوه انتشار اطلاعات و وجود سازوکاری برای اجماع باعث ایجاد اطمینان از صحت داده‌ها و دسترسی‌پذیری می‌شود.

۲-۱-۳. هایپرلجر

هایپرلجر[9] پروژه متن باز[10] مؤسسه لینوکس است که مجموعه ابزار و چارچوب‌های مختلف و پروژه‌هایی برای توسعه دهندگان و کسب و کارها ارائه می‌دهد تا شبکه‌های زنجیره بلوکی را ایجاد کنند [۸]. هدف پروژه هایپرلجر تسهیل همکاری بین کسب و کارها، توسعه دهندگان و سایر فعالان حوزه فناوری دفترکل توزیع شده است. هایپرلجر به یکی از رایج‌ترین و غالب‌ترین ابزارها و محیط‌ها برای نهادها و طرفین تبدیل شده است تا بیشتر درباره فناوری زنجیره بلوکی یاد بگیرند و در جامعه پویا شرکت کنند. قابل ذکر است که هایپرلجر اساساً برای ایجاد راه حل‌های دفترکل توزیع شده برای زنجیره بلوکی‌های دارای مجوز و شبکه‌های کنسرسیوم[11] متمرکز است [۹]. هایپرلجر محبوبیت و اعتبار زیادی در میان شرکت‌های مالی و فناوری بزرگ به‌دست آورده است و باعث شده که زنجیره بلوکی بیشتر از سایر پروژه‌ها مورد توجه واقع شود. در سال‌های اخیر هایپرلجرهای مختلفی با امکانات متنوع ارائه شده‌اند که هر کدام کاربردهای خاص خود را دارا هستند. در ادامه به دو مورد از مهم‌ترین هایپرلجرها اشاره می‌شود.

- هایپرلجر ساتوس[12]: یک پلتفرم ماژولار[13] برای اجرای دفترکل-های توزیع شده است که نوآوری‌های فنی را برای شرکت‌ها و کنسرسیوم‌ها فراهم می‌کند تا درباره پلتفرم خود تصمیمات مستقل بگیرند. به‌طور کل بعضی از ویژگی‌های اصلی این چارچوب عبارتند از: اجماع دینامیک[14]، اجماع اثبات زمان سپری شده[15]، اجرای تراکنش موازی و تراکنش‌های خصوصی [۱۰].
- هایپرلجر فبریک[16]: به جای یک زنجیره بلوکی ساده، مبنایی برای توسعه راه حل‌های مبتنی بر زنجیره بلوکی با یک معماری پیمانه‌ای است. با استفاده از فبریک، اجزای مختلف زنجیره بلوکی (مانند سرویس‌های عضویت و یا نظرسنجی گروهی) می‌توانند وصل و اجرا شوند. در واقع پروژه فبریک ساختاری ارائه می‌دهد تا شرکت‌ها بتوانند شبکه‌های زنجیره بلوکی اختصاصی خود را با سرعتی بالا (بیش از ۱۰۰۰ تراکنش در

[1] Cryptocurrency
[2] Non-Repudiation
[3] Authentication
[4] Bitcoin
[5] Zcash
[6] Smart Contract
[7] Ethereum
[8] Turing-Complete

[9] Hyperledger
[10] Open-Source
[11] Consortium
[12] Hyperledger Sawtooth
[13] Modular
[14] Consensus Dynamics
[15] Proof of Elapsed Time
[16] Hyperledger Fabric

ثانیه) در آن قرار دهند. این ساختار در محیط Go پیاده‌سازی شده است و می‌تواند زنجیره بلوکی کنسرسیومی با امکان فعال‌سازی انواع مجوز را ایجاد کند [۹]. فبریک به میزان قابل توجهی به یک سامانه قرارداد هوشمند به نام Chaincode متکی است که در آن هر یک از همتاهای شبکه‌ها در مخزن[1] داکر[2] اجرا می‌شود. فبریک به سرمایه گذاران این امکان را می‌دهد تا قسمت‌هایی از زنجیره بلوکی را بسازند. شرکت کنندگان می‌بایست برای دریافت مجوز اتصال و صدور تراکنش‌ها بر روی یک زنجیره بلوکی مبتنی بر فبریک، ثبت نام کنند. بر خلاف زنجیره بلوکی عمومی رمزارزها، فبریک به شرکت کنندگان اجازه می‌دهد که کانال مجزایی برای دارایی‌های خود ساخته و از این طریق تراکنش‌ها را از یک دفترکل تفکیک کنند. با این روش، Chaincode که نیاز به خواندن و تغییر حالت یک دارایی دارد، تنها بر روی همتایان درگیر در این مورد کسب و کار به خصوص نصب می‌شود. درست مانند برنامه‌های چت، زنجیره بلوکی فبریک به کاربر اجازه می‌دهند که در هر دو تعاملات خصوصی و غیر خصوصی، شرکت کنند.

۲-۱-۴. سامانه تشخیص نفوذ

هدف از تشخیص نفوذ نمایش، بررسی و ارائه گزارش از فعالیت شبکه است. این سامانه روی بسته‌های داده که از ابزار کنترل دسترسی عبور کرده‌اند، عمل می‌کند. به دلیل وجود محدودیت‌های اطمینان پذیری، تهدیدهای داخلی و وجود شک و تردید مورد نیاز، پیشگیری از نفوذ باید به بعضی از موارد مشکوک به حمله اجازه عبور دهد تا احتمال تشخیص‌های مثبت نادرست کاهش یابد [۱۱]. از طرف دیگر، روش‌های تشخیص نفوذ با هوشمندی همراه هستند و از روش‌های مختلفی برای تشخیص حملات بالقوه، نفوذها و سوءاستفاده‌ها بهره می‌گیرند. یک سامانه تشخیص نفوذ معمولاً به گونه‌ای از پهنای باند استفاده می‌کند که می‌تواند بدون تأثیر گذاشتن روی معماری‌های محاسباتی و شبکه‌ای به کار خود ادامه دهد. این سامانه‌ها می‌توانند در دو حالت منفرد و همکارانه در شبکه فعالیت کنند. در حالت منفرد سامانه تشخیص نفوذ، به تنهایی و به‌صورت ایزوله بر اساس قوانین و الگوهای از قبل تعریف شده و در حال تجزیه و تحلیل ترافیک شبکه است تا بسته‌های مشکوک را شناسایی و به کاربر اعلام کند [۱۲]. این حالت از سامانه‌های تشخیص نفوذ همواره در خطر حملات سایبری است و به راحتی می‌توانند با حملات جدید و ناشناخته به خطر بیافتند. همچنین به‌روزرسانی قوانین و الگوهای تشخیص نفوذ در این سامانه‌ها برای مدیران شبکه امری دشوار است. همکاری میان سامانه‌های تشخیص نفوذ باعث می‌شود هر سامانه‌ای با استفاده از اطلاعات اشتراکی و استفاده از تجربه دیگر سامانه‌ها برای رسیدن به تشخیص دقیق‌تر نفوذ، بهتر عمل کند. شبکه‌ای که سامانه‌های تشخیص نفوذ برای تبادل اطلاعات با یکدیگر به آن متصل هستند، شبکه تشخیص نفوذ همکارانه گفته می‌شود. این نوع شبکه به‌عنوان یک راه حل توافقی درنظر گرفته می‌شود که از اطلاعات منابع متعدد برای به‌دست آوردن درک بهتر هدف و تأثیر حملات اینترنتی پیچیده استفاده می‌کند. همچنین برای حل مشکلات کلاسیک سامانه‌های تشخیص نفوذ مانند حملات صفر روزه[3] و نرخ‌های هشدار بالا و چالش‌های معماری طراحی متمرکز کمک می‌کنند [۱۲ و ۱۳].

۲-۲. مدل پیشنهادی

در این مدل از یک دفترکل توزیع شده امن برای تبادل قوانین تأییده شده بین گره‌های همکار، توسط فناوری زنجیره بلوکی استفاده شده است. به‌عنوان مثال داده‌های قوانین خام تولید شده توسط سامانه‌های مانیتورینگ به‌عنوان تراکنش در زنجیره بلوکی ذخیره می‌شود و در میان گره‌های شرکت کننده شبکه تکثیر می‌شوند. گره‌های درگیر، یک پروتکل اجماع را برای تضمین اعتبار تراکنش‌ها قبل از اضافه کردن آن‌ها در یک بلوک اجرا می‌کنند. این فرآیند تضمین می‌کند که فقط قوانین معتبر و مفید در زنجیره بلوکی گنجانده می‌شوند و از هر تداخل و دستکاری در قوانین ثبت شده در زنجیره بلوکی جلوگیری به عمل می‌آید. به این ترتیب شرکت کنندگان در قبال اقدامات خود پاسخگو هستند، زیرا اقدامات اخیر آن‌ها برای همه گره‌های شبکه شفاف هست. در این سامانه یکپارچگی داده‌ها (قوانین) نیز تضمین می‌شود. دو مشکل عمده در اشتراک‌گذاری داده‌ها وجود دارد: اعتماد متقابل و حریم خصوصی داده‌ها. اعتماد متقابل به این معنی است که هنگام به اشتراک گذاری داده‌ها، طرفین همکاری باید به داده‌هایی که با یکدیگر به اشتراک می‌گذارند اعتماد کنند. به‌عنوان مثال، دو سازمان می‌خواهند نتایج تجزیه و تحلیل حملات چند ماه اخیر خود را که به قوانین سامانه‌های تشخیص نفوذ تبدیل نموده‌اند به اشتراک بگذارند. طرفین باید از صحت داده‌های تبادل شده اطمینان داشته باشند تا آن‌ها را در سامانه‌های تشخیص نفوذ خود استفاده کنند. در این حالت یا سازمان‌ها و گره‌های موجود در شبکه باید احراز هویت و تأیید شده باشند که در این صورت اگر سامانه احراز هویت به‌صورت تک واحدی (مرکزی) باشد، سامانه تشخیص نفوذ همکارانه را تشکیل می‌دهد. در صورتی که بخواهید سامانه احراز هویت به‌صورت غیر متمرکز باشد، باید از فناوری زنجیره بلوکی استفاده کرد. زنجیره بلوکی بر سه نوع خصوصی، عمومی و کنسرسیومی است. در زنجیره بلوکی خصوصی تمام گره‌های موجود در شبکه احراز هویت شده و به شبکه اضافه می‌شوند. اما در زنجیره بلوکی عمومی احراز

[1] Container
[2] Docker
[3] Zero-Day Attacks

هویتی وجود ندارد و همه گره‌ها بدون هویت در آن فعالیت می‌کنند. زنجیره بلوکی کنسرسیومی ترکیبی از گره‌های احراز هویت شده و نشده است که در آن گره‌های احراز هویت شده وظیفه اعتبارسنجی تراکنش‌ها و ثبت آن‌ها را بر عهده دارند. اعتبار داده‌های به اشتراک‌گذاری شده در هر سه مدل زنجیره بلوکی توسط معدن کاوها تضمین می‌شود.

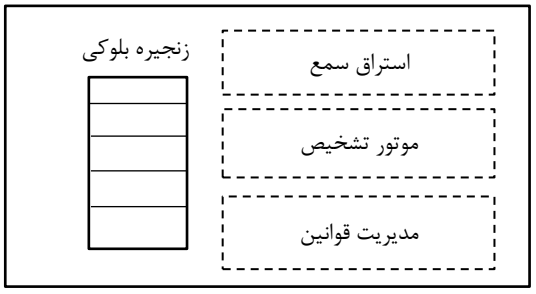
۲-۳. معماری سامانه

در این معماری گره‌های موجود در شبکه به دو گروه مختلف تقسیم می‌شوند:

- گره‌های Trusted: این گره‌ها وظیفه شرکت در عملیات اجماع، تأیید و رد بلاک (حاوی قوانین) را دارند.
- گره‌های Participating: این نوع از گره‌ها تنها می‌توانند قوانین خود را در قالب بلوک‌های زنجیره بلوکی برای تأیید یا رد به شبکه ارسال کنند.

این معماری بر اساس زنجیره بلوکی کنسرسیومی ساخته شده است که در آن هم اعضای احراز هویت شده وجود دارد و هم افراد ناشناس می‌توانند فعالیت کنند. با این حال تنها اعضای Trusted در آن حق تغییر پایگاه داده الگوها و قوانین را دارند. این پایگاه داده که اساس آن بلوک‌های زنجیره بلوکی است به‌صورت تراکنش، قوانین را در خود نگه می‌دارند.

همان‌طور که در شکل (۱) نشان داده شده است، هر گره سامانه تشخیص نفوذ علاوه بر زنجیره بلوکی شامل سه جزء دیگر است. این اجزاء در سامانه‌های تشخیص نفوذ مانند اسنورت تعبیه شده است [۱۴ و ۱۵].



شکل ۱. ساختار پیشنهادی برای گره‌های سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی

- استراق سمع[1]: تمام ترافیک شبکه را می‌خواند و به جزءهای کوچک تقسیم می‌کند و به پس از آن هر جزء را به موتور تشخیص می‌دهد.
- موتور تشخیص[2]: بسته‌های دریافتی از بخش قبل را با الگوها و قوانین تطابق می‌دهد. در صورت وجود تطابق، طبق عملیات تعریف شده برای آن نوع از بسته‌ها، وظیفه خود را انجام می‌دهد.
- مدیریت قوانین[3]: وظیفه نگهداری و تبادل قوانین و الگوها با سایر میزبان‌ها را بر عهده دارد.

در این سامانه هر کدام از گره‌ها برای افزودن داده‌های (قوانین) خود به زنجیره بلوکی باید مراحل زیر را طی کنند. این مراحل تنها در عملیات‌هایی که منجر به تغییر در زنجیره بلوکی می‌شوند را شامل می‌شود.

۱. گره T(trusted) یا P(participating) درخواست افزودن قوانین نگاشتی خود را می‌دهد.
۲. این درخواست به تمام گره‌های T ارسال می‌شود.
۳. گره‌های T در یک چارچوب زمانی از پیش تعیین شده به بلاک مورد نظر رأی مثبت یا منفی می‌دهند.
۴. رأی تمام گره‌ها توسط سامانه شمارش و تأیید یا رد بلوک مورد نظر اعلام می‌شود.

به دلیل تغییر ناپذیری بلوک‌ها، می‌توان هم‌زمان به چندین درخواست رأی داد. درخواست‌های تأیید شده به ترتیب دسته‌ای اجرا می‌شوند.
هر گره موجود در شبکه زنجیره بلوکی دارای یک سری متد برای کار با زنجیره بلوکی است. این متدها برای همه گره‌های Trusted و Participate فعال است. با این حال برخی از این متدها برای اعمال در زنجیره بلوکی نیاز به اجماع و تأیید گره‌های Trusted دارد.

- CreateRule: افزودن یک قانون تشخیص نفوذ به زنجیره بلوکی
- InsertBulkRule: افزودن انبوه قوانین تشخیص نفوذ به زنحیره بلوکی
- RemoveRule: حذف کردن یک قانون از زنجیره بلوکی
- QueryAllRules: پرس‌وجوی تمام قوانین موجود در زنجیره بلوکی
- UpdateRuleOwner: به‌روزرسانی مالکیت یک قانون با شناسه خاص

قوانین ثبت شده در زنجیره بلوکی که توسط هر گره موجود در شبکه تولید و در بلوک‌ها ذخیره می‌شوند تا گره‌های تشخیص نفوذ از آن استفاده کنند، دارای یک استاندارد یکسان است.

Rule = {RuleAction, DocType, Protocol, SourceIP, SourcePort, Direction, DestIP, DestPort, MSG, Sid, Revision, ClassType, Reference, DetectionOption, RuleOwner}

[1] Sniffing
[2] Detection Engine
[3] Rule Manager

که در آن:

- RuleAction: عملکرد گره تشخیص نفوذ در برابر تطابق ترافیک عبوری با این قانون را بیان می‌کند که می‌تواند Alert، Log، Pass و غیره باشد.
- DocType: نوع سند را مشخص می‌کند. سامانه تشخیص نفوذ ما از سه نرم‌افزار تشخیص نفوذ اسنورت[1] [۱۶]، سوریکاتا[2] و ساگان[3] پشتیبانی می‌کند. در این فیلد یکی از این سه مقدار قرار می‌گیرد.
- Protocol: پروتکل ترافیک مورد نظر را مشخص می‌کند که می‌تواند tcp، udp، icmp و غیره باشد.
- Direction: جهت ترافیک را که قانون بر روی آن اعمال می‌شود را مشخص می‌کند. علامت < > نشان دهنده دوطرفه بودن جهت ترافیک است. یعنی ترافیک از مبدأ به مقصد و از مقصد به منبع بررسی می‌شود.
- SourceIP and port: آدرس port و Ip مبدأ ترافیک را مشخص می‌کند.
- DestIP and port: آدرس port و Ip مقصد ترافیک را مشخص می‌کند.
- Msg: متن پیغامی که در Alert و log چاپ می‌شود را نشان می‌دهد.
- Sid and Revision: یک شناسه منحصر به فرد برای هر قانون است. این اطلاعات به پلاگین‌های خروجی اجازه می‌دهد تا قوانین را به راحتی شناسایی کنند.
- Classtype: نوع کلاس حمله اتفاق افتاده را مشخص می‌کند. این فیلد مقادیر زیادی دارد، به‌طور مثال مقدار attempted-admin به معنای حمله برای دسترسی به مجوزهای مدیر است. مقدار denial-of-service نشان دهنده حمله اختلال در سرویس است.
- Reference: منابع خارجی مرتبط با حمله مورد نظر را مشخص می‌کند. به‌طور مثال می‌توان از مقدار McAfee برای ارجاع به منبع مورد نظر استفاده کرد.
- DetectionOption: متشکل شده از یکسری پارامترهای تشخیص ترافیک‌های مخرب و عبارات منطقی در جهت فیلتر ترافیک بر اساس نیازهای سامانه است.
- RuleOwner: صاحب قانون را که این قانون را تولید کرده است، مشخص می‌کند.

[1] Snort
[2] Suricata
[3] Sagan

۲-۴. پیاده‌سازی

در این پروژه برای پیاده‌سازی مدل ارائه شده از هایپرلجر فبریک به‌عنوان زیرساخت شبکه استفاده شده است. همان‌طور که در بخش ۲-۳ گفته شد، هایپرلجر فبریک زیرساخت و ابزارهای لازم برای ساخت زنجیره بلوکی مورد نظر توسعه دهندگان را فراهم می‌کند. با استفاده از این ابزارها می‌توان شبکه زنجیره بلوکی، گره‌های Trusted و Participant را تولید نمود و در کنار این ماژول‌ها از نرم‌افزارهای تشخیص نفوذ مانند اسنورت، سوریکاتا و ساگان برای شناسایی و تشخیص حملات استفاده کرد.

در این سامانه، قرارداد هوشمند جزء اصلی برای اجرای صحیح سامانه است. هایپرلجر فبریک نیز از ویژگی قرارداد هوشمند به نام Chaincode پشتیبانی می‌کند. همه تراکنش‌ها در قرارداد هوشمند هایپرلجر فبریک اجرا شده و نتایج آن در زنجیره بلوکی ثبت یا به کاربر برگردانده می‌شود. بنابراین اجرای صحیح قرارداد هوشمند باعث ثبت داده‌های درست در زنجیره بلوکی شده و نتایجی که کاربر دریافت می‌کند دارای ویژگی صحت داده است. همان‌طور که در بخش‌های قبل توضیح داده شد، قرارداد هوشمند تنها در گره‌های تأیید کننده[4] نصب و اجرا می‌شود. اگر نتایج اجرای قرارداد هوشمند توسط اکثریت گره‌های تأیید کننده، تأیید شود، نتایج در زنجیره بلوکی ثبت می‌شود. برنامه‌هایی که تراکنش آن‌ها فراخوانی از زنجیره بلوکی است نیازی به تأیید گره‌های تأیید کننده ندارند و با اجرای قرارداد هوشمند و استخراج نتایج از زنجیره بلوکی آن را به کاربر برمی‌گردانند.

قرارداد هوشمند پیاده‌سازی شده دارای چندین تابع برای پاسخگویی به تراکنش‌های دریافتی است:

- تابع InitLedger: این تابع بعد از نصب قرارداد هوشمند در گره‌های تأیید کننده و ایجاد یک نمونه از آن در شبکه، اجرا می‌شود. نتیجه اجرای این تابع تولید تعدادی قانون سامانه تشخیص نفوذ برای ثبت در زنجیره بلوکی است. این قوانین در تعدادی تراکنش در بلوک اولیه زنجیره بلوکی ثبت می‌شود.
- تابع CreateRule: این تابع با دریافت پارامترهای ورودی مورد نیاز یک قانون تشخیص نفوذ، آن‌ها را تبدیل به یک تراکنش نموده و خروجی مورد نظر را تولید می‌کند. این خروجی یک تراکنش است که بعد از تأیید اکثریت گره‌های تأیید کننده، در زنجیره بلوکی ثبت می‌شود. برای ایجاد تراکنش مورد نظر، ابتدا

[4] Endorsement Node

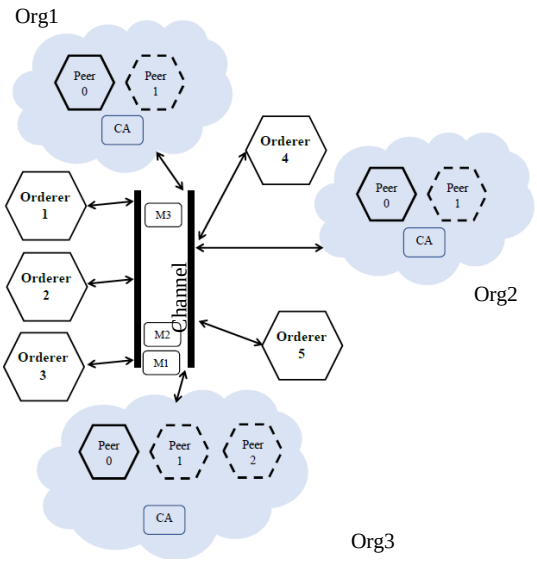
باید شماره آخرین قانون ثبت شده در زنجیره بلوکی استخراج شود و سپس شماره قانون جدید با افزایش یک واحدی آن در تراکنش ایجادی ثبت می‌شود.

- تابع CreateBulkRules: این تابع برای افزودن انبوه قوانین تشخیص نفوذ در زنجیره بلوکی است. ورودی این تابع دو مقدار Rules و RuleOwner است. مقدار متغیر Rules قوانین مد نظر برای افزودن به زنجیره بلوکی است که در فرمت JSON به تابع ارسال شده است. متغیر RuleOwner نیز اسم فردی است که این تابع از قرارداد هوشمند را فراخوانی نموده است و خواهان افزودن قوانین است. خروجی این تابع یک سری تراکنش است که درون هر کدام تعدادی قانون تشخیص نفوذ نوشته شده است. هر قانون نیز دارای یک شماره است که از افزایش یک واحدی شماره قانون قبلی به‌دست می‌آید. مقدار متغیر Rules که به‌صورت JSON ارسال شده است، بعد از دریافت، تجزیه شده و پارامترهای قوانین از آن استخراج می‌شود. در هر قانون پارامترهای Action، Protocol، SourceIP، SourcePort، Direction، DestIP، DestPort، MSG، Metadata، SID، Revision، ClassType و Reference وجود دارد (البته برخی از قوانین منبع مشخصی ندارند و به همین منظور پارامتر Reference آن‌ها Null است). سایر پارامترها در هر قانونی متناسب با آن قانون متغیر هستند. برخی از این پارامترها عبارتند از: flow، flowbits، content، pcre، service و غیره. هر کدام از این پارامترها ممکن است چندین بار در قانون تکرار شود و هر بار مقادیر مختلف داشته باشد. این پارامترها برای مؤلفه DetectionOption طراحی شده در سامانه ما، کاربرد دارند.
- تابع ChangeRuleOwner: این تابع برای تغییر صاحب قانون ثبت شده در زنجیره بلوکی به‌کار می‌رود. با دریافت دو ورودی شماره قانون و اسم صاحب جدید، این تابع قانون مورد نظر را از زنجیره بلوکی فراخوانی نموده و مقدار RuleOwner آن را تغییر می‌دهد و تراکنش جدید را تولید و به زنجیره ارسال می‌کند. با ثبت نسخه جدید از این قانون، نسخه قبل بی اعتبار شده و در فراخوانی‌ها نسخه جدید آن برگردانده می‌شود. با این حال نسخه قبلی در بلوک‌های قبلی باقی می‌ماند و حذف نمی‌شود.
- تابع DeleteRule: این تابع برای حذف قانون از زنجیره بلوکی استفاده می‌شود. به دلیل اینکه زنجیره بلوکی غیر قابل دستکاری است، داده‌های اضافه شده را نمی‌توان از بلوک‌ها حذف نمود. در هایپرلجر فبریک به دلیل تشکیل دفترکل از دو جزء زنجیره بلوکی و وضعیت جهانی، حذف داده‌ها از زنجیره بلوکی نیز امکان‌پذیر است. با این تفاوت که داده‌ها از زنجیره بلوکی حذف نمی‌گردد بلکه وضعیت آن در پایگاه داده وضعیت جهانی حذف می‌شود و مقدار آن در زنجیره بلوکی معتبر نیست. این عمل باید به تأیید اکثریت گره‌های تأیید کننده برسد.
- تابع QueryRule: این تابع برای فراخوانی قوانین از زنجیره بلوکی به‌کار می‌رود. تابع فراخوانی نیاز به ایجاد بلوک جدید و تأیید اکثریت گره‌های تأیید کننده ندارد.
- تابع QueryAllRule: این تابع همچون تابع QueryRule برای فراخوانی استفاده می‌شود. در این تابع همه قوانین موجود در زنجیره بلوکی استخراج شده و در فرمت JSON به برنامه کاربر برگردانده می‌شود. اساس استخراج داده از زنجیره بلوکی شناسه داده (قوانین) است. اگر تنها یک قانون را بخواهید از زنجیره بلوکی بخوانید، کافیست با شناسه قانون مورد نظر فراخوانی صورت گیرد. با این حال برای خواندن تمام قوانین لازم است یک محدوده را مشخص کنید. تمام قوانینی که شناسه آن‌ها داخل این بازه باشد از زنجیره بلوکی خوانده شده و به کاربر برگردانده می‌شود.

۲-۴-۱. ساختار شبکه سامانه تشخیص نفوذ

شبکه زنجیره بلوکی یک زیرساخت فنی است که خدمات دفترکل و قرارداد هوشمند را به برنامه‌ها ارائه می‌دهد. در اکثر موارد، چندین سازمان گرد هم می‌آیند و کانالی را تشکیل می‌دهند که در آن تراکنش‌ها با کد قرارداد هوشمند فراخوانی می‌شوند و مجوزها توسط مجموعه‌ای از سیاست‌ها تعیین می‌شوند که هنگام پیکربندی کانال با آن‌ها موافقت می‌شود. علاوه بر این، سیاست‌ها می‌توانند به مرور زمان منوط به توافق سازمان‌ها تغییر کنند.

در شکل (۲) ساختار شبکه تشخیص نفوذ مبتنی بر زنجیره بلوکی به‌صورت انتزاعی نشان داده شده است. این شبکه شامل ۳ سازمان (Org) است. سازمان شماره یک و دو دارای ۲ گره تشخیص نفوذ و سازمان شماره سه دارای ۳ گره تشخیص نفوذ است. این سازمان‌ها با اشکال ابری نشان داده شده‌اند. در هر کدام از این سازمان‌ها یک گره تأیید کننده (Trusted) وجود دارد که با اشکال شش ضلعی خط صاف مشخص شده‌اند. در هر سازمان باید حداقل یک گره‌های تأیید کننده وجود داشته باشد و امکان وجود چندین گره تأیید کننده نیز وجود دارد. همچنین بر روی این گره‌ها قرارداد هوشمند نصب شده است. در سازمان‌های شماره یک و دو، یک گره عمومی (Participant) نیز وجود دارد و در سازمان شماره سه، ۲ گره عمومی وجود دارد. گره‌های عمومی نیز با اشکال شش ضلعی و خطوط نقطه چین نشان داده شده‌اند. این گره‌ها به جز اعتبارسنجی و تأیید بلوک بقیه عملکردها گره تشخیص نفوذ را دارا هستند. تعداد این گره‌ها در هر سازمان می‌تواند از صفر تا N متغیر باشد.



شکل ۲. ساختار شبکه سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی

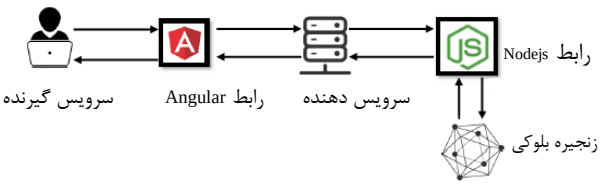
هر سازمانی در این شبکه دارای یک مرجع صدور مجوز[1] است که با مربع‌های کوچک (CA) مشخص شده‌اند. مرکز صدور گواهینامه در شبکه نقش اساسی دارند زیرا آن‌ها گواهینامه‌های X.509 را توزیع می‌کنند که می‌تواند برای شناسایی اجزای متعلق به یک سازمان استفاده شود. از گواهینامه‌های صادر شده توسط CA می‌توان برای امضای معاملات نیز استفاده کرد که نشان می‌دهد سازمانی نتیجه معامله را تأیید می‌کند. اجزای مختلف شبکه زنجیره بلوکی با استفاده از گواهینامه‌ها خود را به‌عنوان سازمانی خاص به یکدیگر معرفی می‌کنند. به همین دلیل است که معمولاً بیش از یک CA برای حمایت از شبکه زنجیره بلوکی وجود دارد. سازمان‌های مختلف اغلب از CAهای مختلف استفاده می‌کنند. ما می‌خواهیم از سه CA در کانال خود استفاده کنیم. در این شبکه همچنین ۵ گره سفارش دهنده[2] برای جمع آوری و پخش تراکنش‌ها و تولید و ثبت بلوک‌های جدید وجود دارد که با شش ضلعی در خارج از سازمان‌ها نشان داده شده‌اند. استفاده از ۵ گره سفارش دهنده برای افزایش تحمل خطا[3] در سامانه است.

۲-۴-۲. برنامه وب

در هایپرلجر فبریک برای ارتباط کاربر با زنجیره بلوکی باید از یک برنامه استفاده نمود. این برنامه برای انجام عملیات‌ها، ارسال و دریافت تراکنش و نتایج آن‌ها به‌کار می‌رود. اکثر این برنامه‌ها در محیط کنسول اجرا می‌شوند که با دستورات خط فرمان کنترل می‌گردند. استفاده از این نوع برنامه‌ها برای کاربران سختی‌های خاص خود را دارد. پیچیدگی و احتمال اشتباه کاربر در این محیط بسیار بالا است و درنهایت از ظاهر کاربرپسندی برخوردار نیست. با این حال استفاده از این برنامه‌ها ضروری و سریع است.

در این پروژه برای راحتی و افزایش سادگی سامانه ارتباطی از یک برنامه وب استفاده شده است. این برنامه جدا از سامانه زنجیره بلوکی نوشته شده است و واسط ارتباطی بین کاربر و سامانه زنجیره بلوکی هایپرلجر فبریک است. با توجه به اینکه در برنامه‌های وب سرعت کاهش پیدا می‌کند، اما به دلیل وجود برنامه وب در سامانه داخلی، سرعت برنامه وب سامانه تشخیص نفوذ تغییر قابل توجهی نسبت به برنامه کنسول نداشته است. در این برنامه تمامی عملیات‌هایی که می‌توان در تعامل با زنجیره بلوکی انجام داد طراحی و تعبیه شده است. سادگی و ظاهر کاربرپسند به کاربران کمک می‌کند تا در انجام عملیات‌های خود دقت و ظرافت کافی را داشته باشند و از رخداد خطا و اشتباه کاربران تا حد زیادی جلوگیری به عمل آید.

در شکل (۳) شماتیک کلی از نحوه ارتباط کاربر با زنجیره بلوکی نشان داده شده است. این برنامه از دو قسمت سرویس دهنده و سرویس گیرنده تشکیل شده است. سمت سرویس دهنده که به‌عنوان رابط به زنجیره بلوکی وصل می‌شود، درخواست‌های کاربر را که از سمت سرویس گیرنده می‌آید، به زنجیره بلوکی ارسال می‌کند. بعد از انجام عملیات‌های مورد نظر کاربر، نتایج از طریق سرویس دهنده به سرویس گیرنده انتقال پیدا می‌کند و کاربر از نتایج مطلع می‌گردد. ارتباط سرویس دهنده با زنجیره بلوکی با استفاده از زبان برنامه‌نویسی NodeJS صورت می‌گیرد. همچنین برای ارتباط بین سرویس دهنده و سرویس گیرنده از زبان برنامه‌نویسی Angular استفاده شده است. برای ارتباط کاربر با سرویس گیرنده نیز از زبان‌های برنامه‌نویسی سمت وب مانند Typescript، Html و CSS استفاده شده است.



شکل ۳. شماتیک کلی از نحوه ارتباط کاربر با زنجیره بلوکی با استفاده از برنامه وب

۲-۵. چالش‌های پیاده‌سازی

با وجود فناوری زنجیره بلوکی و هایپرلجر فبریک چالش‌هایی در پیاده‌سازی سامانه تشخیص نفوذ غیر متمرکز وجود داشت که با تلاش‌های فراوان رفع و پیاده‌سازی کاملی از آن صورت گرفت.

- قرارداد هوشمند سامانه تشخیص نفوذ: قرارداد هوشمند موجود در هایپرلجر فبریک را می‌توان در ۴ زبان برنامه‌نویسی Nodejs، Java، TypeScript و Golang پیاده‌سازی نمود. در این پیاده‌سازی دو چالش عمده وجود داشت. هر نرم‌افزار تشخیص

[1] Certificate Authority
[2] Ordered Node
[3] Fault Tolerance

نفوذ دارای یک الگوی خاص از داده ساختار قوانین است که تنها با آن داده ساختار کار می‌کند [۱۷-۲۰]. با توجه به اینکه سامانه تشخیص نفوذ ما از سه نرم‌افزار تشخیص نفوذ اسنورت، سوریکاتا و ساگان پشتیبانی می‌کند طراحی یک داده ساختار مشترک برای آن‌ها که هم بتوان قوانین هر کدام را به زنجیره بلوکی اضافه نمود و هم بتوان از آن قوانین در نرم‌افزارها استفاده نمود، چالش بزرگی محسوب می‌شد. این چالش با ایجاد داده ساختار مناسب در پیاده‌سازی قرارداد هوشمند رفع گردید. همچنین برای پیاده‌سازی عملیات درج قوانین انبوه، روش درج قوانین تکی باعث افزایش زیاد سربار می‌شود. با این حال با ایجاد برخی تغییرات این مشکل نیز رفع گردید. داده-های خام قوانین در فرمت JSON به قرارداد هوشمند ارسال می‌شود و در آنجا تجزیه می‌شود. پس از آن داده‌ها در چند تراکنش به گره‌های سفارش دهنده ارسال می‌گردد. در این حالت سربار به‌طور محسوسی کاهش پیدا می‌کند.

- پیاده‌سازی شبکه: برای پیاده‌سازی همه سازمان‌ها و گره‌ها در یک رایانه و ایجاد یک شبکه محلی ابزارهای خیلی کمی وجود دارد. یک بستر مناسب برای ایجاد چنین شبکه‌ای داکر است. داکر یک پلتفرم برای اجرای چند برنامه بر روی یک ماشین است. همچنین وجود تعداد گره‌های نامتقارن در سازمان‌های شماره یک، دو و سه نیز یکی از مسائل چالش برانگیز است.
- برنامه وب: وجود یک برنامه برای ارتباط بین کاربر و زنجیره بلوکی لازم و ضروری است. این برنامه به‌طور معمول به‌صورت کنسولی و دستورات خط فرمان است که با سختی‌هایی برای کاربران ایجاد می‌کند. به همین جهت یک برنامه وب که ارتباط بین کاربر و زنجیره بلوکی را برقرار می‌کند در پیاده‌سازی سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی در دستور کار قرار گرفت. این برنامه برای پیاده‌سازی نیز دارای چالش‌هایی بود که درنهایت با تحقیقات فراوان این کار به سرانجام رسید. چالش‌هایی نظیر ارتباط سرویس دهنده و سرویس گیرنده، دسترسی برنامه وب به داده‌های محلی و امنیت برنامه وب مورد توجه قرار گرفته و رفع گردیده است.

۳. نتایج و بحث

در این بخش نتایج اجرای سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی در یک رایانه تشریح شده و ارزیابی انجام شده از سرعت و تأخیر نصب و اجرای سامانه شرح داده شده است. برای نصب و راه‌اندازی سامانه تشخیص نفوذ از یک رایانه با پردازنده Core i7 دو هسته‌ای و فرکانس ۲/۴ گیگاهرتز، حافظه کش L3 با ظرفیت ۴ مگابایت، حافظه RAM ۸ گیگابایتی و کارت گرافیکی Nvidia GeForce 940m استفاده شده است. در این رایانه از سامانه عامل لینوکس نسخه Ubuntu 20.04 بهره گرفته شده است. به دلیل استفاده از داکر به‌عنوان بستر نصب و اجرای سامانه تشخیص نفوذ، استفاده از پردازنده‌های چند هسته‌ای و چند نخی در افزایش راندمان و کاهش تأخیرها می‌تواند کمک بزرگی باشد. همچنین به دلیل حجم بالای ماژول‌های هایپرلجر فبریک استفاده از یک RAM با ظرفیت بالا برای نصب و اجرای شبکه با چندین گره ضروری است. [۱۹و ۱۸].

نتیجه ارزیابی برای دو مرحله جداگانه، محاسبه شده است. مرحله یک برای نصب و راه‌اندازی سامانه تشخیص نفوذ، شبکه هایپرلجر فبریک و برنامه وب است که در جدول (۱) نشان داده شده است. این سامانه برای سه سازمان (Org) طراحی شده است که در سازمان‌های شماره یک و دو تعداد ۲ گره و در سازمان شماره سه تعداد ۳ گره برای تشخیص نفوذ درنظر گرفته شده است. با افزایش/کاهش تعداد سازمان‌ها و گره‌ها مدت زمان نصب و راه-اندازی سامانه نیز افزایش/کاهش خواهد یافت. مرحله دوم، اجرا و آزمایش این سامانه برای تمام عملیات‌های ممکن که در قرارداد هوشمند سامانه تعریف شده است، است. جدول (۲) ارزیابی مربوط به مراحل آزمایش و اجرا را نشان می‌دهد. عملیات‌های آزمایش و اجرا به دو قسمت تولید بلوک جدید و عدم تولید بلوک جدید تقسیم می‌شوند. زمان اجرای عملیات‌هایی که منجر به تولید بلوک می‌شوند در مقیاس ثانیه است. با این حال عملیات‌هایی مثل پرس‌وجو که تغییری در زنجیره بلوکی ایجاد نمی‌کنند، در مقیاس میلی‌ثانیه اجرا و نتایج را به کاربر برمی‌گردانند.

جدول ۱. نتایج ارزیابی از زمان نصب و راه‌اندازی سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی

عملیات‌ها	حداقل زمان	حداکثر زمان
راه‌اندازی شبکه هایپرلجر فبریک (ثانیه)	۵۲	۶۰
نصب قرارداد هوشمند (ثانیه)	۱۰	۱۵
نمونه گیری از قرارداد هوشمند (ثانیه)	۵۰	۸۰
ایجاد Wallet و کاربر (ثانیه)	۱	۲
نصب سرویس دهنده (ثانیه)	۳	۵
نصب سرویس گیرنده (ثانیه)	۳	۸
مجموع (ثانیه)	۱۱۹	۱۷۰

جدول ۲. نتایج ارزیابی از زمان اجرا سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی

عملیات‌ها	حداقل زمان	حداکثر زمان
افزودن قانون تکی (ثانیه)	۲	۱۵
افزودن قانون انبوه (ثانیه)	۵	۱۵
به‌روزرسانی صاحب قانون (ثانیه)	۲	۱۵
حذف قانون (ثانیه)	۲	۱۵
پرس‌وجوی قانون تکی (میلی‌ثانیه)	۵۰۰	۸۰۰
پرس‌وجوی همه قوانین (میلی‌ثانیه)	۵۰۰	۸۰۰

تمام زمان‌های مربوط به اجرا از میانگین چند بار اجرای سامانه تشخیص نفوذ به‌دست آمده است. با این حال در برخی از موارد که گره‌ها با اختلال مواجه می‌شوند این زمان افزایش می‌یابد. دلیل افزایش این حالت از اجرا این است که سایر گره‌های برخط در شبکه زنجیره بلوکی تا زمان اتمام مهلت زمانی برای گره‌هایی که اختلال دارند منتظر می‌مانند. بعد از اتمام این مهلت، پروتکل و الگوریتم خود را ادامه داده و کار را بدون آن‌ها به سرانجام می‌رسانند. این مهلت زمانی برابر با ۱۲۰۰۰۱ میلی ثانیه معادل ۱۲۰ ثانیه است.

۴. نتیجه‌گیری

سامانه‌های تشخیص نفوذ همکارانه برای شناسایی حملات سایبری پیچیده و فزاینده پیشنهاد شده است. غلبه بر مسئله اعتماد و به اشتراک‌گذاری قوانین نرم‌افزارهای تشخیص نفوذ همچنان یک چالش به حساب می‌آید. در این پروژه، یک معماری غیر متمرکز را پیشنهاد داده شد که از قابلیت تغییر ناپذیری رکوردهای فناوری زنجیره بلوکی و خواص ضد دستکاری داده‌های ذخیره شده در دفترکل توزیع شده استفاده می‌کند. جزئیات مربوط به گردش کارهای مختلف را برای شبکه همکارانه خود ارائه داده شد، از جمله نحوه افزودن قوانین یا به‌روزرسانی مجموعه قوانین. نمونه اولیه این پروژه را با استفاده از چارچوب هایپرلجر فبریک پیاده‌سازی شد و با استفاده از یک معیار موجود ارزیابی شد. نتایج اولیه امیدوار کننده به نظر می‌رسد، از جمله اجرای بیش از ۱۰۰۰ تراکنش در زنجیره بلوکی، تعداد تأخیر برای یک شبکه کوچک نظیر به نظیر و محدود، قابل قبول بود. سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی از یک رابط کاربری برنامه وب برای ارتباط کاربر با زنجیره بلوکی استفاده نموده است. این برنامه وب از سادگی و زیبایی خاصی برخوردار است که باعث کاهش پیچیدگی‌های ارتباط با زنجیره بلوکی شده است. همچنین احتمال رخداد خطاهای سهوی کاربر تا حدودی جلوگیری به عمل آمده است. در این سازوکار از کاهش سرعت اجرای هایپرلجر فبریک و ثبت تراکنش‌ها جلوگیری شده است و راندمان بالای رابط کاربری باعث بهتر شدن سامانه تشخیص نفوذ شده است.

برنامه کاری آینده ما شامل استفاده از نمونه اولیه پروژه در یک شبکه واقعی، آزمایش با ترافیک شبکه حمله شبیه‌سازی شده و ارزیابی عملکرد با تعداد گره‌های IDS زیاد در محل است. استفاده از گره‌های زیاد برای ایجاد فضای واقعی از شبکه‌های بزرگ در جهت آزمایش و ارزیابی کارایی و تأخیر در ثبت تراکنش‌ها لازم و ضروری است. همچنین از برنامه‌های آتی می‌توان به افزایش تعداد نرم‌افزارهای تشخیص نفوذ پشتیبانی شده توسط سامانه اشاره نمود. با افزایش پایداری نسخه ۲ هایپرلجر فبریک نیز می‌توان در آینده از نسخه جدید آن در سامانه تشخیص نفوذ مبتنی بر زنجیره بلوکی استفاده نمود و آن را ارتقاء داد.

[1] [1] Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. “Blockchain ”; BIS 2017, 59, 183-187.

[2] [2] Gupta, S. “Blockchain”; IBM Online (http://www. IBM. COM), 2017.

[3] [3] Zheng, Z.; Xie, S.; Dai, H. N.; Chen, X.; Wang, H. “Blockchain Challenges and Opportunities: A Survey”; Int. J. Web. Grid. Serv. 2018, 14, 352-375.

[4] [4] Nakamoto, S. “Bitcoin P2P E-Cash Paper”; The Cryptography Mailing List, 2008.

[5] [5] Franco, P. “Understanding Bitcoin”; Wiley, 2014.

[6] [6] Hopwood, D.; Bowe, S.; Hornby, T.; Wilcox, N. “Zcash Protocol Specification”; GitHub: San Francisco, CA, USA, 2016.

[7] [7] Wood, G. “Ethereum: A Secure Decentralised Generalised Transaction Ledger”; Ethereum Project Yellow Paper, 2014, 151, 1-32.

[8] [8] Cachin, C. “Architecture of the Hyperledger Blockchain Fabric”; DCCL 2016, 310, 4.

[9] [9] Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Yellick, J. ”Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”; Proc. Thirteenth EuroSy .Conf. 2018, 1-15.

[10] [10] Olson, K.; Bowman, M.; Mitchell, J.; Amundson, S.; Middleton, D.; Montgomery, C. “Sawtooth: An Introduction”; The Linux Foundation, 2018.

[11] [11] Dhillon, V.; Metcalf, D.; Hooper, M. “The Hyperledger Project”; Blockchain Enabled Applications, Apress, Berkeley, CA 2017, 139-149.

[12] [12] Ashoor, A. S.; Gore, S. “Importance of Intrusion Detection System (IDS)”; J. Sci. Eng. Res. 2011, 2, 1–4.

[13] [13] Sabahi, F.; Movaghar, A.; “Intrusion Detection: A Survey”; Third Int. Conf. Commun. Syst. Netw. 2008, 23–26.

[14] [14] Caswell, B.; Beale, J. “Snort 2.1 Intrusion Detection”; Elsevier, 2004.

[15] [15] Arboleda, A. F.; Bedón, C. E. “SnortTM Diagrams for Developers”; Universidad Del Cauca-Colombia, 2005.

[16] [16] Ujjan, R. M. A.; Pervez, Z.; Dahal, K. “Snort Based Collaborative Intrusion Detection System Using Blockchain in SDN”; 13th Int. Con. Softw. Knowl. Inf. Manag. Appl. 2019, 1-8.

[17] [17] Alexopoulos, N.; Vasilomanolakis, E.; Ivánkó, N. R.; Mühlhäuser, M. “Towards Blockchain-Based Collaborative Intrusion Detection Systems”; Int. Con. Criti. Inf. Infrastruct. Secur. 2017, 107-118.

[18] [18] Li, W.; Wang, Y.; Li, J.; Au, M. H. “Toward a Blockchain-Based Framework for Challenge-Based Collaborative Intrusion Detection”; Int. J. Inf. Secur. 2020, 1-13.

[19] [19] Putra, G. D.; Dedeoglu, V.; Kanhere, S. S.; Jurdak, R. “Towards Scalable and Trustworthy Decentralized Collaborative Intrusion Detection System for IoT”; IEEE/ACM Fifth Int. Con. Internet-of-Things Des. and Implement 2020, 256-257.

[20] [20] Meng, W.; Tischhauser, E. W.; Wang, Q.; Wang, Y.; Han, J. “When Intrusion Detection Meets Blockchain Technology: A Review”; IEEE Access 2018, 6, 10179-10188.