UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RAFAEL MENEZES BARBOZA

MONITORAMENTO VOLTADO À CIBERSEGURANÇA EM SISTEMAS INDUSTRIAIS

CAMPO MOURÃO

2020

RAFAEL MENEZES BARBOZA

MONITORAMENTO VOLTADO À CIBERSEGURANÇA EM SISTEMAS INDUSTRIAIS

CYBERSECURITY MONITORING TO INDUSTRIAL SYSTEMS

Trabalho de Conclusão de Curso de graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Luiz Arthur Feitosa dos Santos

CAMPO MOURÃO

2020

RAFAEL MENEZES BARBOZA

MONITORAMENTO VOLTADO À CIBERSEGURANÇA EM SISTEMAS INDUSTRIAIS

Trabalho de Conclusão de Curso de graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 18/novembro/2020

Luiz Arthur Feitosa dos Santos
Doutorado
UTFPR

Aretha Barbosa Alencar
Doutorado
UTFPR

Rodrigo Campiolo
Doutorado
UTFPR

Lucas Nathan Barbosa de Oliveira
PTI

CAMPO MOURÃO

2020

RESUMO

Sistemas industriais desempenham papéis vitais nos processos de produção em indústrias e infraestruturas críticas. Falhas e ataques cibernéticos em tais sistemas podem ocasionar problemas econômicos e ambientais. Com a popularização da Internet e dos benefícios oferecidos por sistemas interconectados, os sistemas industriais que antes eram isolados, passam a participar da Internet, porém tendo que lidar com possíveis ataques cibernéticos. Desta forma, objetivou-se nesse trabalho o desenvolvimento de uma arquitetura de monitoramento de eventos de segurança cibernética para sistemas industriais. A arquitetura de monitoramento desenvolvida destaca-se por ser modular, escalável, distribuída e por facilitar o acoplamento de diferentes dispositivos e tecnologias do setor industrial e corporativo. A arquitetura de monitoramento coleta dados dos objetos monitorados como arquivos de *log*, tráfego de rede, tentativas de intrusão, e os armazena de forma a possibilitar à construção de visualizações gráficas e correlação dos dados. Visando avaliar a arquitetura proposta foi desenvolvido um protótipo da arquitetura de monitoramento utilizado para realização de experimentos, o que proporcionou a validação do uso da arquitetura no monitoramento e na construção de visualizações gráficas. Durante a avaliação da arquitetura constatou-se que as visualizações geradas apresentam dados relevantes dos objetos monitorados e podem ser úteis para que profissionais da área da segurança cibernética identifiquem visualmente anomalias e ciberataques. Sendo assim, verificou-se que a arquitetura de monitoramento proposta pode contribuir com a segurança cibernética em sistemas industriais, e assim, ajudar a mitigar danos causados por ciberataques e anomalias.

Palavras-chaves: Monitoramento. Cibersegurança. Sistemas Industriais.

ABSTRACT

Industrial systems are very important in industries and critical infrastructure. Failures and cyber attacks on such systems can cause economic and environmental problems. With the popularization of the Internet and the benefits offered by interconnected systems, industrial systems that were previously isolated, now participate in the Internet, but having to deal with possible cyber attacks. Thus, the objective of this work was to develop a cyber security event monitoring architecture for industrial systems. The monitoring architecture developed stands out for being modular, scalable, distributed and for facilitating the coupling of different devices and technologies from the industrial and corporate sectors. The monitoring architecture collects data from monitored objects such as log files, network traffic, intrusion attempts, and stores them in a way that allows the construction of graphical visualizations and data correlation. In order to evaluate the proposed architecture, a prototype of the monitoring architecture was used to carry out experiments, which provided the validation of the use of architecture in the monitoring and construction of graphical visualizations. During the architecture assessment, it was found that the generated visualizations present relevant data from the monitored objects and can be useful for cybersecurity professionals to visually identify anomalies and cyber attacks. Thus, it was found that the proposed monitoring architecture can contribute to cybersecurity in industrial systems, and thus, mitigate damage caused by cyber attacks and anomalies.

Keywords: Monitoring. Cybersecurity. Industrial Systems.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ACID: *Atomicity, Consistency, Isolation, Durability*. 24, 43
API: *Application Programming Interface.* 14

CLP: Controlador Lógico Programável. 10, 16–18, 22
CRUD: Criação, Recuperação, Atualização e Exclusão. 14

DDoS: *Distributed Denial of Service.* 16
DoS: *Denial of Service.* 16

HIDS: *Host Intrusion Detection System.* 13
HMI: *Human-Machine Interface.* 17
HTTP: *Hyper Text Transfer Protocol.* 14

IDS: *Intrusion Detection System.* 13, 24
IPS: *Intrusion Prevention System.* 24
IPv4: *Internet Protocol version 4.* 28
IPv6: *Internet Protocol version 6.* 28

JSON: *JavaScript Object Notation.* 14, 31

LAN: *Local Area Network*. 9

NIDS: *Network Intrusion Detection System.* 13

SCADA: Sistemas de Supervisão e Aquisição de Dados. 9, 16, 18, 22
SSH: *Secure Socket Shell.* 27

TCP: *Transmission Control Protocol.* 11
TCP/IP: *Transmission Control Protocol/Internet Protocol.* 9, 34
TI: Tecnologias da Informação. 22, 23
TO: Tecnologias Operacionais. 22, 23
TOR: *The Onion Router*. 17

UDP: *User Datagram Protocol.* 11
USB: *Universal Serial Bus.* 17

WAN: *Wide Area Network*. 9

SUMÁRIO

1 INTRODUÇÃO

Sistemas industriais atuam na automatização e supervisão de processos de produção e, estão presentes em indústrias e infraestruturas críticas, por exemplo, usinas de energia, sistemas de distribuição e indústrias automobilísticas. Inicialmente, os sistemas industriais estabeleciam uma abordagem de isolamento e obscuridade no uso de protocolos e tecnologias do meio industrial, justamente para dificultar a ação de *hackers* e ataques direcionados. Porém, com a modernização da indústria e com sistemas industriais conectados à Internet, as vulnerabilidades cibernéticas do setor industrial, tais como, falta de autenticação e criptografia de protocolos industriais mais antigos, foram expostas e tornaram-se mais suscetíveis a ataques cibernéticos (HADZIOSMANOVIC et al., 2012; Wang et al., 2019).

A infraestrutura industrial possui ampla diversidade de tecnologias e dispositivos, que nem sempre oferecem níveis de segurança aceitáveis, como autenticação e criptografia. Isso acontece devido ao fato de que essas tecnologias e dispositivos não foram desenvolvidos com os requisitos de segurança exigidos para os dias atuais. (HADZIOSMANOVIC et al., 2012; MICRO, 2019).

Em razão da importância que sistemas industriais desempenham no meio industrial, anomalias e ciberataques podem comprometer gravemente o processo de produção. Desta forma, o objetivo deste trabalho é desenvolver uma arquitetura para monitoramento de eventos em sistemas industriais que permita visualizar e identificar possíveis anomalias, principalmente relacionadas à segurança cibernética. A arquitetura de monitoramento em questão, captura e armazena informações como tráfego de rede, registro de *logs*, tentativas de intrusão, dos objetos monitorados e, disponibiliza tais informações para a construção de visualizações gráficas.

Foi realizado a implementação de um protótipo da arquitetura de monitoramento para prova de conceito e realização de experimentos. Os experimentos constataram efetividade no monitoramento de redes e *hosts*, viabilizando a construção de visualizações gráficas úteis para que profissionais da área de redes e segurança cibernética possam supervisionar e detectar anomalias/ciberataques nos objetos monitorado.

O tema deste trabalho surgiu de um programa de iniciação tecnológica de dois anos desenvolvido pela Fundação Parque Tecnológico Itaipu (PTI) em Foz do Iguaçu — PR, no qual os participantes são motivados a pesquisarem e desenvolverem tecnologias de segurança cibernética para sistemas industriais. No primeiro ano da iniciação tecnológica, foi desenvolvido outro artigo científico com estudos de ataques cibernéticos que afetavam a indústria. Tal pesquisa foi o primeiro passo para compreender a motivação e forma de atuação de ataques cibernéticos em sistemas industriais. Diante da bagagem de conhecimento construído no primeiro artigo, houve a oportunidade de contribuir com o desenvolvimento de uma ferramenta capaz de monitorar redes e *hosts* de sistemas de industriais e corporativos.

Sendo assim, o projeto de iniciação estendeu por mais um ano. No período de extensão, foi desenvolvida uma arquitetura de monitoramento de segurança cibernética para sistemas industriais, na qual faz-se uso exclusivo de software livre. Ao final do período de extensão um artigo científico foi escrito apresentando a arquitetura de monitoramento.

Desta forma, a presente monografia aborda a mesma arquitetura de monitoramento desenvolvida no período de iniciação tecnológica. Porém, diferentemente do artigo científico desenvolvido ao final da iniciação, a monografia aqui apresentada possibilita uma abordagem mais ampla e detalhada do escopo do problema e da arquitetura de monitoramento, em geral. Sendo assim, a contribuição efetiva deste trabalho para a segurança cibernética industrial são:

- Estudo de caso e levantamento dos problemas da indústria moderna;
- Desenvolvimento de uma arquitetura de monitoramento voltada à segurança cibernética em sistemas industriais;
- Escolha de softwares e implementação de um protótipo da arquitetura de monitoramento gratuita e *open source* para atuar com a coleta, tratamento, armazenamento e visualização dos dados do monitoramento;
- Implementação de regras de filtragem, normalização e enriquecimento de dados para redes e *hosts*;
- Implantação de métodos de autenticação e criptografia de dados na comunicação entre os segmentos do protótipo;
- Construção de visualizações gráficas úteis para o monitoramento e supervisão de redes e *hosts* dos setores corporativos e industriais.

Além deste capítulo introdutório, este trabalho dispõe de outros 4 capítulos. No Capítulo 2, é apresentado o referencial teórico abordando conceitos a respeito de tecnologias e dispositivos da infraestrutura industrial, monitoramento, segurança cibernética e ameaças. No Capítulo 3, é apresentada a arquitetura proposta para o monitoramento de sistemas industriais. O Capítulo 4, discute os resultados obtidos com experimentos em um protótipo da arquitetura de monitoramento. O Capítulo 5 apresenta as conclusões e trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos importantes para compreensão deste trabalho. A Seção 2.1 apresenta alguns softwares e dispositivos industriais utilizados normalmente na infraestrutura de controle industrial. A Seção 2.2 discute como é realizado o monitoramento de sistemas com ênfase no ambiente industrial. Na Seção 2.3 é abordado o funcionamento das ferramentas de armazenamento e visualização de dados que são utilizadas neste trabalho. A Seção 2.4 apresenta e discute algumas definições e termos recorrentes no escopo da segurança cibernética, bem como, alguns dos principais ataques e vulnerabilidades em ambientes industriais. Por fim, a Seção 2.5 apresenta trabalhos relacionados com esta monografia.

2.1 Infraestrutura de Controle Industrial

Indústrias e infraestruturas críticas por todo mundo utilizam tecnologias típicas de ambientes industriais. Estas tecnologias evoluíram durante os anos e assumiram grandes responsabilidades nos processos de produção. Esta seção é dedicada a introduzir alguns dos principais componentes envolvidos em infraestruturas industriais, bem como o seu envolvimento no processo de produção.

2.1.1 Sistemas de Supervisão e Aquisição de Dados

Os Sistemas de Supervisão e Aquisição de Dados (SCADA) são sistemas utilizados para monitorar e supervisionar dispositivos por meio de sensores e atuadores, permitindo a coleta de variáveis e execução de ações como diagnosticar erros em máquinas ou soar alarmes em tempo real. Esses dispositivos com seus sensores e atuadores devem se comunicar com o servidor central SCADA, por meio de protocolos de comunicação. Nas primeiras redes industriais, a comunicação era realizada em redes privadas do tipo *Local Area Network* (LAN) ou *Wide Area Network* (WAN) e utilizando protocolos proprietários (NCS, 2004; BOYER, 2004). Os sistemas SCADA atuais investem em protocolos abertos como *Transmission Control Protocol/Internet Protocol* (TCP/IP) e conexão com a Internet. Como consequência, surgem novas preocupações com a segurança cibernética, porque tais sistemas ao serem conectados à Internet podem expor vulnerabilidades viabilizando ciberataques (Lakhoua, 2018).

Os sistemas SCADA atuam na automação de processos e são amplamente utilizados em infraestruturas críticas como na produção de petróleo, usinas nucleares, hidrelétricas, abastecimento de água, que necessitam de monitoramento e ajustes a todo momento, reduzindo erros e perdas durante o processo (BOYER, 2004). Dado o valor das informações que tais sistemas transportam e interagem, eles são frequentemente alvo de ataques cibernéticos. O domínio e o controle destas informações críticas por *hackers* podem comprometer a saúde e a segurança das pessoas, danificar instalações industriais e gerar prejuízos econômicos (HADZIOSMANOVIC et al., 2012; Wang et al., 2019).

Um desafio considerável para a segurança é que os componentes utilizados em campo possuem recursos computacionais limitados ou são antigos demais para suportar algoritmos de criptografia de dados, logo raramente recebem atualizações de antivírus, *firewall* e de sistemas (MICRO, 2019). Além disso, quando o sistema industrial se torna operacional, os testes de segurança e análises de riscos são mais difíceis, pois podem gerar interrupções e instabilidade. O uso de simulações é uma forma de mitigar esses riscos realizando testes sem influenciar diretamente a plataforma operacional.

2.1.2 Sensores e Atuadores Industriais

Sensores são utilizados na automação de processos, realizando uma analogia, os sensores propiciam as máquinas os sentidos humanos. Os sensores podem ser definidos como dispositivos que, quando expostos a fenômenos físicos tais como temperatura, pressão, força, produzem saídas proporcionais. Atuadores são como os músculos e braços que aceitam comandos de controle como sinais elétricos e assim produzem mudanças no ambiente físico (BISHOP, 2007).

No contexto industrial, sensores e atuadores são responsáveis por obter dados referente ao processo de produção e atuar de alguma maneira sobre o processo. Sensores e atuadores são importantes para a automação pois permitem redução de custos de pessoal e aumento da qualidade na produção com velocidade e precisão (JUNIOR, 2003). Os sensores trazem ao dispositivo Controlador Lógico Programável (CLP), que será definido a seguir, variáveis de controle do processo e os atuadores realizam alguma ação no processo de produção.

2.1.3 Controlador Lógico Programável

Os CLP foram desenvolvidos para serem operados por engenheiros e profissionais com conhecimentos em programação. Estes dispositivos são utilizados em ambientes que necessitam automatizar processos de produção. O CLP pode ser definido como controlador baseado em microprocessador que utiliza memória programável para armazenar instruções e implementar funções tais como: lógica, contagem, sequenciamento, temporização e aritmética. Podendo assim ser utilizado para controlar equipamentos no ambiente de produção (BOLTON, 2009).

Os dispositivos CLP possuem como entrada, dados de sensores e atuadores presentes no processo de produção. Assim, os CLP podem aplicar a lógica de programação que receberam para realizar tarefas como: acionar ou desligar uma ventoinha de acordo com certa temperatura estipulada. Uma das principais vantagens dos CLP é que o mesmo controlador pode ser utilizado em uma ampla gama de sistemas de controle, apenas alterando a lógica para o novo escopo em que foi inserido.

2.1.4 Protocolos de Comunicação Industrial

Protocolos de comunicação são normas e padrões estabelecidos entre as partes que desejam trocar informações, para a transferência e representação de dados por um canal de comunicação. Alguns

protocolos muito conhecidos como *User Datagram Protocol* (UDP) e *Transmission Control Protocol* (TCP) são exemplos dos principais protocolos operantes em redes de computadores (BRANQUINHO, 2014; TANENBAUM, 2010).

Quando relacionado a redes, protocolos de comunicação industriais possuem algumas características especiais que diferem de protocolos de redes de computadores convencionais. O tempo de envio e recebimento e a confiabilidade das informações transmitidas são fatores determinantes na comunicação entre dispositivos industriais, já que o nível de precisão e agilidade deve ser maximizado principalmente em operações críticas em processos de produção.

No início da utilização dos sistemas SCADA, poucos padrões e protocolos de comunicação existiam. Desta foram, diversos fabricantes de dispositivos industriais passaram a sentir a necessidade de protocolos industriais que suportem a comunicação entre os dispositivos industriais. Sendo assim, tais fabricantes passaram a implementar protocolos proprietários, o que obrigou a indústria a manter diversos protocolos atuantes nas redes industriais em caso de possuírem dispositivos de fabricantes diferentes.

O protocolo ModBus desenvolvido em 1979, define uma estrutura de mensagens para comunicações mestre-escravo entre dispositivos industriais e sistemas de supervisão. O ModBus é um exemplo de protocolo proprietário que sofreu mudanças e adaptações durante o tempo. A princípio o ModBus era um protocolo proprietário criado pela MODICON visando o uso em seus próprios dispositivos. Porém atualmente a MODICON autorizou o uso do ModBus por outros fabricantes passando a ser um protocolo aberto (ORGANIZATION, 2005).

Com a modernização dos sistemas industriais, novos equipamentos foram sendo implantados e almejados, fazendo com que os protocolos industriais passassem por padronizações por órgãos e comitês (por exemplo o Comitê de Transmissão e Distribuição e o Comitê de Subestações da IEEE Power & Energy Society), seguindo normas para melhor atender as necessidades de novos dispositivos (WAAGSNES.; ULLTVEIT-MOE., 2018).

2.2 Monitoramento

O monitoramento dos elementos envolvidos em sistemas, corporativos ou industriais, devem capturar os eventos que ocorrem no sistema ou rede e apresentá-los para o administrador do sistema de monitoramento. A captura dados pode ser realizada por meio de agentes diretamente instalados nos elementos a serem ser monitorados ou por analisadores de tráfego de rede, que analisam os pacotes de rede e as informações que ali trafegam, fruto da comunicação dos elementos monitorados. Tais dados quando correlacionados podem gerar informações úteis para detectar anomalias permitindo melhores perspectivas do real estado dos elementos monitorados.

Anomalias e ciberataques podem ocorrer em ambientes industriais independente da infraestrutura envolvida e, podem impactar seriamente o processo de produção. As anomalias são causadas por diferentes problemas e, são classificadas em duas categorias.

A primeira categoria de anomalia não possui presença de agentes maliciosos, que ocorre ocasionalmente devido falhas ou ações inesperadas. Um exemplo de anomalia sem a presença de agente malicioso pode acontecer quando algum dispositivo de uma rede monitorada começa a realizar *backups* inesperados, desencadeando um grande fluxo na rede. Já a segunda categoria, é anomalia causada por ataques de terceiros, que possuem o objetivo de comprometer a segurança (FARRAPOSO et al., 2007; KASPERSKY, 2018).

As anomalias com e sem presença de agentes maliciosos podem ser identificadas por meio de mecanismos de monitoramento, pois desencadeiam irregularidades no fluxo comum dos objetos monitorados. O monitoramento dos elementos no sistema permite identificar se algo está fora do padrão de funcionamento. Alguns mecanismos de monitoramento trazem interfaces de visualização dos eventos que ocorrem de modo a facilitar a identificação de comportamentos maliciosos dos objetos monitorados (WAZUH, 2019).

2.2.1 Sensores de Redes

Sensores de rede neste trabalho são ferramentas responsáveis por interceptar o tráfego de entrada e saída em redes corporativas e industriais. Sendo assim, ferramentas de análise de tráfego de rede, tais como, Tcpdump e Tshark, podem ser inclusas nesta categoria de sensores (TSHARK, 2017; GOYAL; GOYAL, 2017).

Tshark

Tshark é um software analisador de pacotes que permite ler ou capturar pacotes de redes em tempo real (TSHARK, 2017). O Tshark é similar ao software de análise e captura de pacotes Tcpdump, que trabalha capturando o tráfego de rede da interface de rede que foi lhe foi determinado (ASRODIA; PATEL, 2012).

A interface utilizada no Tshark é a linha de comando, podendo aceitar regras como, filtragem por protocolo, contagem, extração de estatísticas, e outras. Para os usuários que preferem utilizar interfaces gráficas, o software analisador de pacotes Wireshark oferece interface gráfica e desempenha as mesmas funções do Tshark (WIRESHARK, 2017).

O Tshark reconhece vários protocolos de comunicação, incluindo os protocolos industriais mais utilizados como o DNP3, ModBus e outros. A captura dos pacotes pode ser realizada em modo promíscuo, o que resulta na captura sem interferência, de qualquer pacote que esteja transitando na rede (GOYAL; GOYAL, 2017).

Os pacotes capturados podem ser apresentados na linha de comando ou gravados em arquivos de *log*. Gravar os pacotes em arquivos de *log* é uma opção interessante, porque permite que outros softwares possam utilizar estas informações para realizar operações mais complexas, como correlações, análises, filtros, monitoramento, detecção de anomalias e ciberataques.

2.2.2 Sensores de *Hosts*

Sensores de *hosts* neste trabalho são ferramentas responsáveis por coletar e ou analisar *logs*, atividades e métricas de sistemas. Sendo assim, ferramentas de detecção de intrusão podem ser incluso nesta categoria de sensores.

Sistema de Detecção de Intrusão

Intrusion Detection System (IDS) analisam atividade de dispositivos, sistemas ou redes, com objetivo de detectar possíveis sinais de invasões ou ataques. Os IDSs podem ser configurados para alertar comportamentos que são categorizados como maliciosos. Possuem opções para enviar e-mails, mensagens e avisos, ou até mesmo tomar alguma providência para prevenção de intrusões como negar tráfego de um dispositivo na rede, impedir acessos, entre outras (WAAGSNES.; ULLTVEIT-MOE., 2018; COLBERT, 2016).

Na detecção de invasões ou ataques, podem ser usadas duas abordagens: baseadas em assinaturas ou em comportamento (DEBAR; VIINIKKA, 2005). Em abordagens por assinatura, o IDS deve manter bases de dados com padrões de eventos (assinaturas) que caracterizam ameaças. Os eventos analisados são comparados com as assinaturas catalogadas a fim de detectar ataques conhecidos. Nas abordagens baseadas em comportamento, o IDS analisa desvios de padrões (anomalias) no comportamento rotineiro dos dispositivos ou redes, podendo detectar ataques não catalogados (desconhecidos).

Também é possível dividir IDS em duas categorias descritas a seguir:

- *Host Intrusion Detection System* (HIDS), na qual os sensores do sistema de detecção são instalados em dispositivos *(hosts)*, para monitorar informações especificas do sistema como chamadas de sistema, arquivos de *log* e sistemas de arquivos.
- *Network Intrusion Detection System* (NIDS), na qual monitora e analisa o tráfego no segmento da rede por meio de sensores ou analisadores de tráfego de rede *(Sniffers)*.

2.3 Ferramentas de Armazenamento e Visualização de Dados

Esta seção apresenta algumas das principais ferramentas de armazenamento e visualização gráfica, que vão compor a construção do núcleo do protótipo da arquitetura de monitoramento. É válido enfatizar que com intuito de tornar a construção do protótipo acessível e de baixo custo de implantação, todas as ferramentas e softwares escolhidos para compor o objetivo são de código aberto *(open source)*.

2.3.1 MongoDB

O MongoDB é um software de banco de dados de código aberto, alto desempenho e escalável. Por ser um banco de dados não relacional, os dados referentes a cada registro ficam armazenados juntos em documentos JSON ou BSON, não divididos em tabelas como é feito em bancos de dados relacionais (VOKOROKOS et al., 2016; MONGODB, 2019a). Por conta da utilização de objetos JSON

no armazenamento dos registros, manipulações como inserções, edições e outras ações, podem ser executadas facilmente, devido a estrutura de organização simples dos arquivos JSON.

2.3.2 ElasticSearch

O ElasticSearch é um software altamente escalável baseado no Apache Lucene e de código aberto, que permite armazenar, pesquisar e analisar grandes volumes de dados em tempo real. Geralmente o ElasticSearch é utilizado para alimentar aplicativos que necessitam realizar buscas rápidas e complexas em grande número de dados (ELASTIC, 2019b).

O ElasticSearch pode ser categorizado como banco de dados não relacional baseado em documentos. Ao enviar arquivos *JavaScript Object Notation* (JSON) para ElasticSearch são criados mapeamentos adequados para cada campo do arquivo, objetivando desempenho para buscas e inserções em larga escala (BAJER, 2017; VOKOROKOS et al., 2016). A comunicação com o ElasticSearch é realizada pela *Application Programming Interface* (API) Java RESTful, que permite executar operações de Criação, Recuperação, Atualização e Exclusão (CRUD), por meio de requisições *Hyper Text Transfer Protocol* (HTTP).

2.3.3 Logstash

O Logstash é um software de código aberto utilizado para coleta de dados. Tal software, pode unificar dinamicamente fontes de dados distintas, e aplicar filtros, normalizações, buscas e enriquecimento dos dados de entrada. O Logstash organiza o processamento dos dados em *threads* chamadas *pipelines*. Cada *pipeline* pode receber uma configuração específica de entrada, filtragem e saída (BAJER, 2017; ELASTIC, 2019d).

Os dados que entram no Logstash são enfileirados e enviados para filtragem e posteriormente para saídas. A política de enfileiramento persistente dos dados na entrada do Logstash permite que nenhum dado seja perdido em caso de falhas como, encerramento anormal do programa ou falta de processamento da máquina para lidar com grades volumes de dados simultâneos (MARQUARDT, 2019; ELASTIC, 2019d).

A ferramenta Logstash pode colaborar com os registros dos dados em ferramentas de armazenamento, visto que possui diversos *plugins* de conexão para softwares de banco de dados, tais como MongoDB e ElasticSearch.

2.3.4 Kibana

O Kibana é um software altamente escalável de código aberto que permite pesquisar, visualizar e interagir com os dados armazenados nos índices do ElasticSearch. Com o Kibana é possível construir e compartilhar rapidamente painéis dinâmicos com visualizações gráficas, tabelas e mapas, facilitando a compreensão e a interpretação de grandes volumes de dados (ELASTIC, 2019a).

2.4 Segurança Cibernética

A cada dia a segurança cibernética torna-se imprescindível em ambientes domésticos, corporativos e industriais. Tal importância decorre dos grandes danos que os ciberataques podem proporcionar, principalmente quando os alvos dos ataques são indústrias ou setores críticos (Transporte, Água, Energia e Telecomunicações), já que interferências causadas por ciberataques nestes ambientes industriais e críticos podem afetar muitas pessoas.

No entanto, a Internet traz muitas comodidades e benefícios, desta forma, torna-se desejável conectar ambientes domésticos, corporativos e industriais à Internet. Tal conexão abre novas portas para vetores de ataques e exploração de vulnerabilidades o que pode comprometer a segurança. Contudo, conectar tais sistemas na Internet é quase inevitável, sendo assim é necessário pensar em contramedidas para reduzir os riscos de ciberataques nestes ambientes.

Desta forma, ataques cibernéticos a sistemas e industriais de infraestruturas críticas podem afetar muitas pessoas e colocar vidas em risco. Nas subseções seguintes são apresentadas as definições de artefatos maliciosos que podem comprometer a segurança cibernética em qualquer ambiente.

2.4.1 Malware

De modo geral, o termo *malware* é qualquer tipo de *sofware* malicioso incluindo vírus, *ransomware*, *spyware*, *backdoor*, que são desenvolvidos e projetados para causar danos a algum sistema. Os danos variam de atividades maliciosas como roubo de dados, acesso não autorizado e exploração de vulnerabilidade (GRAVES, 2016; CERT.BR, 2017).

Alguns tipos de *malwares* são projetados para serem voltados a algum alvo bem definido, esses tipos de *malwares* viabilizam os chamados ataques direcionados. São exemplos de ataques direcionados, os *malwares* industriais, que foram desenvolvidos para atuarem na infraestrutura industrial e, em geral, não funcionam da mesma forma em ambientes fora do escopo da indústria.

Ransomware

Os *ransomwares* são códigos maliciosos que realizam criptografia de dados armazenados em equipamentos infectados, tornando-os indisponíveis. Em troca do resgate dos dados criptografados, o *ransomware* costuma exigir pagamento de uma certa quantia de dinheiro, geralmente em *bitcoins* (GRAVES, 2016; CERT.BR, 2017).

Os *ransomwares* costumam causar grandes danos, principalmente se os dados afetados não tiverem cópia de segurança *(backup)*. No contexto industrial a indisponibilidade destes dados e dos equipamentos afetados podem interferir em grandes atrasos e perdas no processo de produção.

Backdoor

Backdoors são softwares maliciosos com a finalidade de conceder aos *hackers* o acesso futuro ao sistema, mesmo se corrigida a vulnerabilidade original usada para atacar o sistema (GRAVES, 2016; CERT.BR, 2017). Em geral, *backdoors* podem ser inseridos em outros softwares maliciosos (Cavalo de Troia) ou não, podendo assim induzir a vítima a instalar softwares ilegítimos inconscientemente e ser infectadas pelo *backdoor*.

2.4.2 Negação de serviço (DoS e DDoS)

Negação de serviço, ou *Denial of Service* (DoS), é uma técnica em que o atacante utiliza de meios para afetar a disponibilidade de serviços, máquinas e redes. A técnica de negação de serviço, quando coordenada de forma distribuída, ou seja, quando vários computadores realizam o ataque simultaneamente, recebe o nome de negação de serviço distribuído, ou *Distributed Denial of Service* (DDoS) (GRAVES, 2016; CERT.BR, 2017).

O objetivo de ataques que afetam a disponibilidade não é invadir e roubar informações, mas sim esgotar recursos das vítimas. Ataques de negação de serviço podem ser realizados por meio de envio de grande quantidade de requisições para um serviço, geração de grande tráfego de dados para uma rede e pela exploração de vulnerabilidades existentes em programas (TANENBAUM, 2010).

2.4.3 Ameaças Cibernéticas Industriais

Ataques cibernéticos a sistemas industriais de infraestruturas críticas podem afetar muitas pessoas e colocar vidas em risco. Nesta subseção são apresentados alguns ataques e *malwares* que interferiram na indústria e sistemas críticos pelo mundo (GRAVES, 2016; CERT.BR, 2017).

Stuxnet

Descoberto em 2010, o Stuxnet teve como alvo os sistemas SCADA. Acredita-se que foi desenvolvido com intuito de danificar centrífugas de enriquecimento de urânio nas usinas nucleares do Irã. O Stuxnet é um *malware* de ataque direcionado especificamente aos dispositivos CLP, que são amplamente utilizados nos processos de automação industrial (ZETTER, 2015).

O Stuxnet verifica se a máquina está conectada a modelos específicos de CLP, no caso os fabricados pela Siemens. Se forem identificados tais dispositivos, o Stuxnet altera a programação dos CLP, falsificando os sinais dos sensores e assim dando a impressão de normalidade (CHEREPANOV; LIPOVSKY, 2017; ZETTER, 2015). Com isso o *malware* pode manipular os equipamentos da forma como quiser, sem ser detectado pela unidade de controle ou pelos funcionários, pois a interface e o sistema SCADA sempre indicará valores estáveis de operação.

Caso a máquina infectada não possuir CLP conectados, o *malware* assume uma forma latente, ou seja, entra em inatividade, desativando temporariamente a atividade destrutiva, e assim

não causando danos à máquina vítima, porém continua se espalhando pela rede ou através de dispositivos *Universal Serial Bus* (USB).

O Stuxnet procura na máquina atacada softwares como Siemens Simatic WinCC Step7 o que indica que a máquina é um computador usado para controlar CLP industriais, também conhecida como estação de trabalho *Human-Machine Interface* (HMI).

Se determinados softwares que caracterizam estações de trabalho HMI forem encontrados, o *malware* analisa se há softwares de proteção como antivírus, para avaliar a melhor forma de implantar um arquivo .DLL falso na máquina, que mais tarde seria usada para desativar alarmes e interpretar relatórios de status que poderiam alertar os funcionários a respeito das atividades anormais. É importante ressaltar que o Stuxnet foi programado com um tempo de vida, de 21 a 90 dias. Este tempo é monitorado periodicamente e quando excedido o *malware* encerra sua atividade.

Industroyer

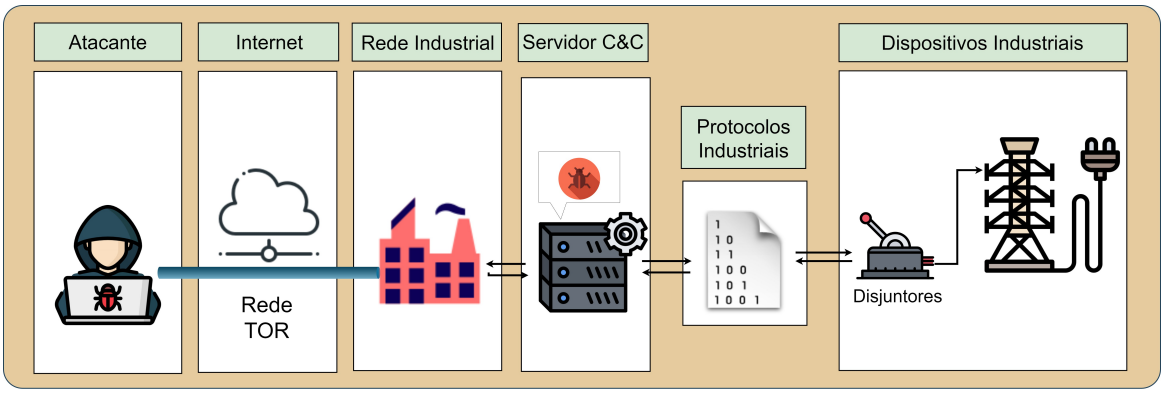
O Industroyer também conhecido como CrashOverride é um *malware* capaz de controlar interruptores e disjuntores de subestações de eletricidade, mas também pode ser modificado para controlar outros tipos de elementos em infraestruturas críticas. Em 2016, o Industroyer foi responsável por um ataque a rede elétrica da Ucrânia, que ficou desativada por uma hora (OSBORNE, 2018).

A comunicação do Industroyer com os disjuntores e controladores é através de protocolos de comunicação industriais usados mundialmente em infraestruturas críticas como no fornecimento de energia, água, gás entre outras. Estes protocolos de comunicação foram projetados há décadas quando não havia preocupação com segurança, já que estes protocolos ficavam isolados do mundo externo (OSBORNE, 2018). Portanto, o *malware* não precisou explorar vulnerabilidades de protocolo. apenas aprender como se comunicar já era o suficiente, pois estes protocolos não utilizavam nenhum tipo de segurança durante a comunicação.

O Industroyer é um *malware* modular para o sistema operacional Windows. O seu componente principal é um *backdoor* que gerencia todo o ataque instalando outros componentes, que se conectam com um servidor remoto para receber ou reportar informações. Outro módulo muito importante é a carga útil, que pode ser dividida em quatro componentes estrategicamente projetados para atender o maior número de protocolos de comunicação, independentemente do tipo de dispositivo, fornecedor ou arquivos de configuração.

A Figura 2.1 mostra como o *malware* Industroyer supostamente agiu contra a rede de distribuição de energia Ucraniana em 2016. Nota-se que o atacante utiliza a rede *The Onion Router* (TOR) para tornar anônima a conexão com o servidor industrial infectado. O Industroyer então envia pacotes de rede maliciosos nos protocolos citados anteriormente, com o objetivo de desativar os disjuntores da rede de distribuição e interromper o fornecimento de energia

Figura 2.1. Fluxo de ataque Industroyer ocorrido na Ucrânia em 2016



Fonte: Autoria própria

Vulnerabilidades em Protocolos Industriais

Sistemas industriais são compostos por vários componentes, tais como servidores SCADA, CLP, sensores e atuadores, que participam da automação, gerenciamento e supervisão do processo de produção. A comunicação dos componentes em sistemas industriais é realizada por meio de protocolos industriais. Até recentemente a indústria para garantir a segurança da infraestrutura de rede e tecnologias, estabelecia uma abordagem de isolamento e obscuridade do uso de protocolos e tecnologias proprietárias. Tal isolamento e obscuridade firmava certas barreiras de conhecimento, e dificultava a atuação de hackers no ambiente industrial (Cruz et al., 2015; Fan et al., 2015).

Com o advento da Internet e da indústria 4.0, surge cada vez mais a necessidade por parte das indústrias, de conectarem seus sistemas à Internet, devido aos diversos benefícios e facilidades que a Internet pode proporcionar no ambiente de produção, como exemplo, acesso remoto de servidores e dispositivos, conexão entre filiais e outras indústrias, compartilhamento de dados e outros. Porém conectar tais sistemas traz o malefício da perda de segurança por isolamento e obscuridade, e desta forma, podendo viabilizar ataques hackers externos direcionados.

Infelizmente a maioria das tecnologias e protocolos industriarias existentes foram projetadas com a confiabilidade em mente, relegando a segurança a um papel secundário. Sendo assim, protocolos de rede antigos porém, consolidados pela indústria, como exemplo, ModBus/TCP e DNP3, não implementam nenhuma camada de segurança de autenticação e criptografia de dados. Tais protocolos estão ultrapassados perante a evolução industrial e necessitam de mudanças, principalmente com relação à segurança, o que é um grande desafio, pois níveis de criptografia e autenticação podem interferir diretamente no desempenho da comunicação necessária para ambientes industriais e críticos (Fan et al., 2015).

Pela falta de segurança desses protocolos industriais, conectar sistemas industriais a Internet ainda é um grande desafio e pode expor muitas vulnerabilidades. Em ataques direcionados a indústrias, *hackers* poderiam capturar o tráfego de dados do que acontece na rede e roubar informações críticas e/ou enviar pacotes com instruções maliciosas que seriam recebidos pelos dispositivos industriais (HADZIOSMANOVIC et al., 2012).

2.5 Trabalhos Relacionados

Foram realizadas pesquisas de arquiteturas de monitoramento cibernético para sistemas industriais, semelhantes a proposta neste trabalho de monografia. Porém, a grande maioria dos trabalhos que relacionam monitoramento e sistemas industriais, aplicam o monitoramento do processo de produção e não a segurança cibernética especificamente.

Com ressalva, algumas empresas de segurança cibernética como por exemplo a TI Safe, oferecem serviços de gerenciamento, prevenção, detecção e resposta a incidentes em redes industriais (TI Safe, 2007). Porém tais serviços oferecidos são pagos e não possuem transparência das ferramentas e métodos de monitoramento utilizados. Desta forma, um dos destaques do presente trabalho, é a utilização de softwares *open source* e gratuitos para compor a solução de monitoramento, contribuindo com a redução dos custos de implantação no ambiente industrial.

Nesta seção são discutidos trabalhos relacionados ao tema desta monografia. A Seção 2.5.1, apresenta um artigo que expõe desafios e oportunidades na proteção de sistemas de controle industrial. Na seção 2.5.2, é abordado a plataforma Wazuh de monitoramento em *hosts*.

2.5.1 Desafios e oportunidades na proteção de sistemas de controle industrial

Hadziosmanovic et al. (2012) abordam dificuldades, oportunidades e esforços na construção de ferramentas e metodologias que ajudem a melhorar a segurança cibernética em sistemas industriais. Os autores argumentam que, infelizmente devido à natureza específica que assumem os sistemas industriais, as soluções comumente utilizadas em ambientes de TI, podem não funcionar com tanta eficácia em ambientes de produção. Isto acontece devido aos tipos de dispositivos, redes, protocolos de comunicação e dados que estes sistemas trabalham, que são específicos do ambiente industrial.

Desta forma, duas abordagens gerais baseadas em redes e *hosts* podem ser utilizadas, para melhorar a segurança em sistemas. A primeira abordagem analisa diretamente os dados do sistema, como tráfego de rede e *logs* de dispositivos, podendo assim criar mecanismos de alerta sobre possíveis assinaturas ou padrões de dados que caracterizem ciberataques. Um exemplo da primeira abordagem baseada em assinaturas, aconteceria se fosse analisado um pacote de rede, no qual sua carga útil incluísse comandos maliciosos já conhecidos, desta forma poderia soar algum alerta ou realizar o bloqueio do pacote na rede. A segunda abordagem é a análise baseada no comportamento do sistema, no qual poderiam ser descritas as operações comuns no sistema e assim, qualquer comportamento significativamente diferente das operações convencionais implicaria em uma potencial ameaça. Detecções de ameaças baseadas em comportamento ou anomalias poderiam ocorrer como exemplo, se em dado momento um sistema assumir uma atividade atípica não especificada previamente, como o aumento significativos no fluxo de dados de rede.

Como apresentado no artigo, ambas as abordagens possuem prós e contras. Abordagens baseadas em assinaturas são mais rápidas e precisas, porém podem detectar apenas ataques conhecidos, passando despercebidos, ataques que, os quais as assinaturas não foram previamente reconhecidas. No contexto de sistemas industriais o número de ciberataques conhecidos é relativamente pequeno, portanto, ainda existem poucas assinaturas reconhecidas o que implica na baixa eficácia de abordagem baseada em assinatura em sistemas industriais.

Em contra medida abordagens baseadas em anomalias ou comportamento, possuem grande potencial para detectar ataques desconhecidos. No contexto de sistemas industriais, por possuírem muitas vezes, endereços IP estáticos, números limitados de serviços, comportamentos repetitivos e padronizados, traçar limiares de comportamento em tais sistemas, pode ser facilitado e mais eficiente, na detecção de anomalias e ciberataques.

2.5.2 Wazuh

O Wazuh é uma plataforma gratuita e de código aberto que nasceu como uma derivação do OSSEC HIDS, e depois foi integrado ao Elastic Stack e OpenSCAP, evoluindo para uma solução mais abrangente (OSSEC, 2018; ELASTIC, 2019b). O Wazuh é usado para prevenção, detecção e resposta de ameaças sendo capaz de proteger ambientes locais, virtualizados, em contêineres e baseados em nuvem (WAZUH, 2019).

Os principais componentes da plataforma são agentes instalados e executando em cada *host* monitorado. Os agentes coletam informações realizando monitoramento da integridade de arquivos, leitura de mensagens de *log* do sistema e varredura das configurações dos sistemas. Os dados coletados são enviados para um servidor central da plataforma por meio de um canal de comunicação criptografado e autenticado.

No componente servidor, são realizadas decodificações e análises os dados recebidos dos agentes. Eventos maliciosos como exemplo, tentativas de intrusão, arquivos alterados, *malwares* e *rootkits*, podem ser categorizados em regras no servidor, permitindo que alertas sejam disparados caso algum evento corresponda às regras preestabelecidas.

O Wazuh é integrado ao ElasticStack (Elasticsearch, Kibana, Filebeat) para fornecer *feeds* de mensagens de *log* já decodificadas e tratadas pelo servidor da plataforma. Os dados são armazenados em índices no Elasticsearch e apresentados por meio de visualizações gráficas geradas pelo Kibana, viabilizando o monitoramento e gerenciamento dos dispositivos submetidos à plataforma.

A arquitetura de monitoramento apresentada no presente trabalho, possui muitos pontos semelhantes com plataforma Wazuh, como exemplo, a utilização do conjunto de ferramentas da ElasticStack e o monitoramento de *hosts* por meio do OSSEC HIDS. Porém, a plataforma Wazuh implementa no servidor central, métodos de detecção e análise de *malwares* e *rootkits*, que vão além do escopo definido da arquitetura proposta nesta monografia, o qual o objetivo primeiramente é construir uma base sólida para realizar o monitoramento no ambiente industrial.

Todavia, os pontos fortes da arquitetura proposta nesta monografia que diferem da plataforma Wazuh, é a possibilidade de acoplar agentes sensores e coletores para diversas tecnologias tais como, redes corporativas, redes industriais, softwares ou qualquer outro objeto que permita ser monitorado por algum tipo de sensor. A flexibilidade de permitir o monitoramento em diversos tipos de objetos faz com que a solução seja genérica e adaptável em qualquer ambiente.

A plataforma Wazuh se prova eficiente na prevenção, detecção e resposta de ameaças em *hosts* porém, até o momento não há nenhuma implementação de agentes para o monitoramento passivo de tráfego de rede ou de outras tocologias, o que poderia ser um diferencial interessante para segurança cibernética em sistemas corporativos e industriais.

3 METODOLOGIA

Neste capítulo é abordada uma arquitetura de monitoramento, que se destaca por ser modular e por viabilizar o monitoramento de dispositivos, bem como tecnologias industriais. A Seção 3.1 apresenta a arquitetura de monitoramento para sistemas industriais. Na Seção 3.2 é abordada a implementação de um protótipo da arquitetura de monitoramento. Por fim, a Seção 3.3, argumenta as considerações finais do deste capítulo.

Para contextualizar, a infraestrutura industrial pode ser separada em duas vertentes tecnológicas: Tecnologias da Informação (TI) e Tecnologias Operacionais (TO) (KASPERSKY, 2018; BAKUEI RYAN FLORES, 2018).

TI são sistemas/dispositivos responsáveis em trabalhar com os dados do contexto industrial, tais como: documentos importantes, segredos de negócio, dados de parceiros e funcionários. TO são os sistemas/dispositivos responsáveis em atuar e monitorar processos a fim de torná-los automatizados. Alguns exemplos desta vertente são sistemas SCADA, CLP, sensores e computadores envolvidos na produção.

As vertentes TI e TO possuem propósitos diferentes e por esta razão, as estratégias de segurança não podem ser as mesmas para ambas (HADZIOSMANOVIC et al., 2012). Sistemas de TI priorizam a confidencialidade, integridade e disponibilidade. Já os sistemas TO precisam priorizar a disponibilidade e autenticidade. Tais prioridades estão relacionadas ao objetivo e tipo de informação que estes sistemas lidam (Wang et al., 2019; KASPERSKY, 2018).

Diante da vasta heterogeneidade das tecnologias e dispositivos em ambientes de TI e TO, foi desenvolvida uma arquitetura de monitoramento flexível que possa trabalhar com a diversidade de tecnologias e necessidades presentes nos ambientes de TI e TO. A solução de monitoramento tem como base uma sequência bem definida de tarefas que podem ser brevemente ilustradas pela Figura 3.1, no qual apresenta as quatro principais tarefas base da solução de monitoramento (Coletar, Tratar, Armazenar e Apresentar). Desta forma, as etapas do monitoramento apresentadas neste trabalho consiste em:

Figura 3.1. Principais etapas do monitoramento

Fonte: Autoria própria

- Coletar informações relevantes dos objetos em monitoramento;
- Tratar as informações coletadas realizando correlações, decodificações e enriquecimento dos dados;

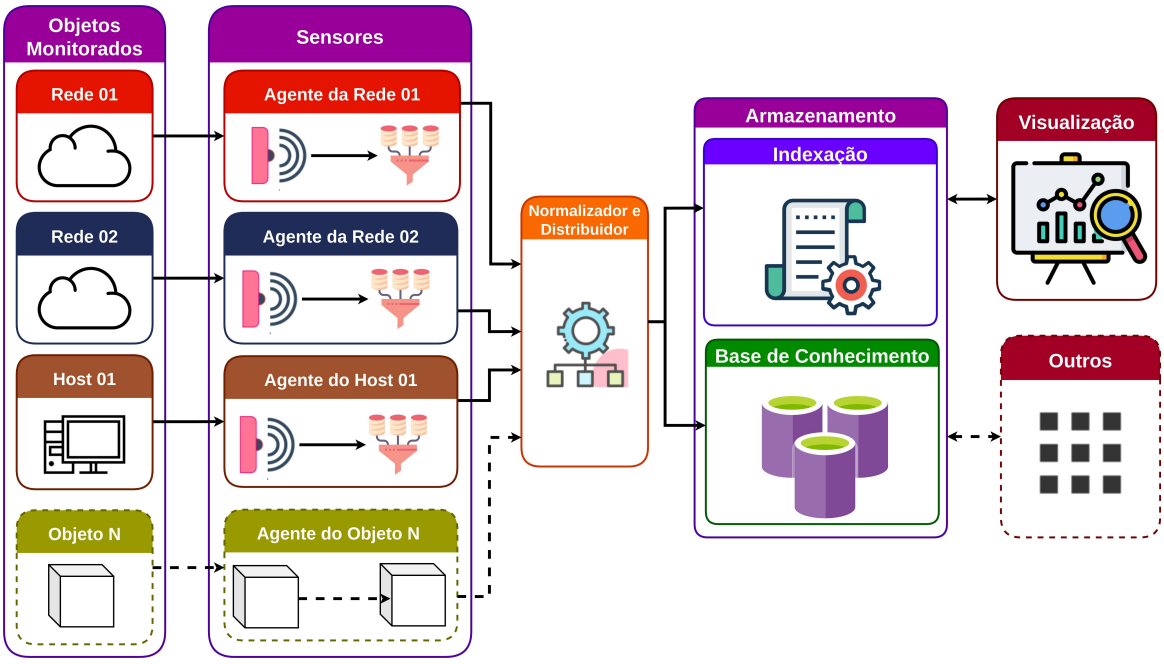
- Armazenar os dados em estruturas de armazenamento que assegurem segurança e velocidade de acesso para que tais dados possam ser posteriormente visualizados;
- Apresentar graficamente os dados coletados para o usuário final, de preferência de uma forma que propicie uma fácil interpretação dos dados coletados.

A arquitetura de monitoramento e seu protótipo desenvolvido neste trabalho são apresentadas em mais detalhes nas Seções 3.1 e 3.2.

3.1 Arquitetura de Monitoramento

Com o objetivo de prover monitoramento em ambientes industriais foi elaborada uma arquitetura de monitoramento, apresentada na Figura 3.2, que possa trabalhar com as tecnologias e necessidades presentes nos ambientes de TI e TO. A arquitetura é dividida em seis segmentos e, funciona semelhante a um *pipeline* na qual o resultado ou saída de cada segmento atua como entrada para o próximo segmento da arquitetura (AHLAWAT, 2017). Sendo assim, os próximos parágrafos apresentam em detalhes cada um dos segmentos da arquitetura.

Figura 3.2. Arquitetura de monitoramento genérica



Fonte: Autoria própria

O segmento Objetos Monitorados representa o conjunto de possíveis redes e *hosts*, softwares e tecnologias existentes nos ambientes de TI e TO. As redes podem ter protocolos industriais como ModBus, DNP3 ou protocolos convencionais, tais como TCP/IP. Os *hosts* podem ser computadores, servidores, estações de trabalho do ambiente de TI ou TO.

O segmento Sensores simboliza o conjunto de tecnologias responsáveis por capturar dados relevantes dos objetos monitorados, e transformar em tais dados em objetos estruturados ao estilo

chave e valor organizados em blocos de informações. Dentro do segmento Sensores encontram-se subsegmentos agentes para cada um dos objetos monitorados. Os subsegmentos agentes são compostos por sensores e coletores que vão capturar dados relevantes dos objetos monitorados, como exemplo, tráfego da rede, *logs* de sistemas, dentre outros.

No segmento Normalizador e Distribuidor são aplicadas padronizações nos campos dos objetos estruturados para que possam ser relacionados futuramente. A normalização neste ponto padroniza os dados coletados, permitindo maior flexibilidade no acoplamento de novos sensores, pois independentemente dos tipos de dados trabalhados pelos sensores, todos serão normalizados para o padrão compatível com a arquitetura. Após o processo de normalização, o próprio segmento Normalizador e Distribuidor é responsável por armazenar tais dados nos sub-segmentos Base de Conhecimento e Indexação.

O segmento Normalizador e Distribuidor é muito importante na arquitetura, tendo em vista que ele recebe todo o fluxo dos sensores e coletores e necessita pré-processar e armazenar um grande volume de dados. Sendo assim é essencial que o segmento Normalizador e Distribuidor possa ser escalável e possua maneiras de gerenciar entradas e saídas.

Com os dados armazenados na Base de Conhecimento torna-se possível realizar consultas, relacionar dados e desenvolver visualizações gráficas. Porém, buscas realizadas diretamente na Base de Conhecimento (banco de dados) podem ser lentas, devido à natureza de armazenamento robusta da base, na qual é priorizado as propriedades de transações de *Atomicity, Consistency, Isolation, Durability* (ACID), garantindo a segurança e validade dos dados apesar de erros, falhas de energia e outros contratempos (VOKOROKOS et al., 2016).

Por conta disto, foi implementado o subsegmento Indexador, que permite manipulações (inserções, edições e outras ações) dos dados mais rapidamente devido a forma de armazenagem dos dados em índices. Sendo assim, o conteúdo da Base de Conhecimento é replicado para o Indexador que organiza os dados de forma a otimizar buscas e manipulações, agilizando a construção de visualizações gráficas com os dados dos objetos monitorados (GROUP, 2019).

O segmento Visualização representa softwares e tecnologias utilizadas para criar visualizações gráficas. No segmento Visualização é possível monitorar o que ocorre nos objetos monitorados do primeiro segmento da arquitetura. As visualizações gráficas podem ser interpretadas por profissionais da área de segurança cibernética, propiciando o monitoramento de ambientes de TI e TO, e a identificação de possíveis anomalias e ciberataques que venham a ocorrer nos objetos monitorados.

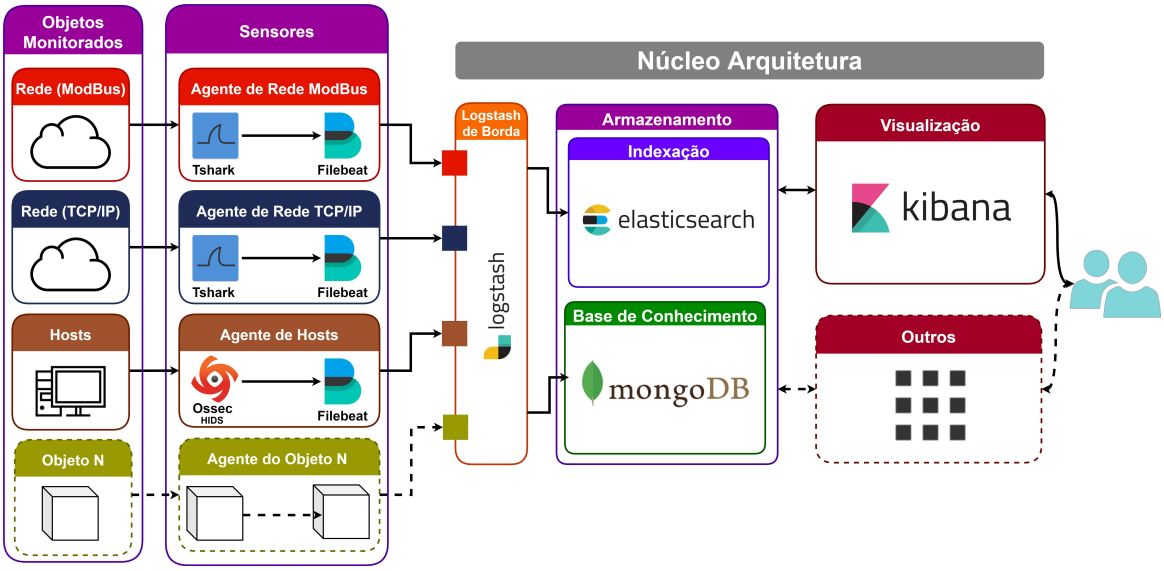
Na arquitetura há ainda o segmento nominado como Outros. Este segmento está diretamente conectado ao segmento de Armazenamento e representa qualquer outro tipo de software ou ferramenta que possa ser alimentada pelos dados enriquecidos e armazenados. IDS, *Intrusion Prevention System* (IPS) e outras ferramentas de análise, são exemplos de softwares que poderiam ser acoplados ao segmento de Armazenamento da arquitetura. O benefício de acoplar outros segmentos ao Armazenamento é poder utilizar dos dados já coletados e armazenados pela arquitetura e realizar analises destes dados externamente.

A utilização de uma arquitetura de monitoramento ao invés de softwares de monitoramento isolados oferece benefícios, por exemplo, a possibilidade de trabalhar com diversas redes e *hosts* e tecnologias existentes no ambiente industrial, podendo correlacionar informações e trazer ao sistema de monitoramento várias perspectivas dos objetos monitorados. Na próxima Seção 3.2, é apresentado um protótipo da arquitetura de monitoramento proposta.

3.2 Protótipo da Arquitetura de Monitoramento

Esta seção apresenta um protótipo da arquitetura de monitoramento abordada Seção 3.1. O protótipo representado na Figura 3.3 faz uso de softwares e ferramentas, para construção de um ambiente capaz de validar o uso da arquitetura no monitoramento de redes, *hosts* e outras tecnologias.

Figura 3.3. Protótipo de arquitetura de monitoramento com tecnologias



Fonte: Autoria própria

Para capturar o tráfego das redes foi utilizado como sensor o software analisador de tráfego rede Tshark, compatível com a maioria de protocolos de rede existentes, inclusive protocolos de redes industriais (TSHARK, 2017). Na coleta de dados nos *hosts* são utilizados agentes OSSEC - HIDS, que funcionam como sistema de detecção de intrusão, podendo coletar dos *hosts*, comportamentos potencialmente maliciosos (OSSEC, 2018). Ambos os sensores contribuem para colher e registrar as informações dos objetos a serem monitorados. Nos coletores é utilizado o software FileBeat da família Elastic (ELASTIC, 2019c). O FileBeat monitora e gerencia os arquivos de *log* gerados pelos sensores e, realiza o envio dos dados para o próximo segmento da arquitetura. Tanto os sensores quanto os coletores podem ser ajustados ou alterados para atender as necessidades de monitoramento, por exemplo, se houver a necessidade de monitorar um equipamento (N) muito específico o qual, não é compatível com nenhum dos sensores e coletores já existentes, outros sensores e coletores poderiam ser acoplados para atender tais necessidades.

O segmento Logstash de Borda utiliza o software Logstash e, trabalha como uma barreira de entrada entre os elementos a serem monitorados e o núcleo da arquitetura. Em tal segmento são implementadas portas de comunicação diferentes para cada categoria de objetos monitorados. Desta forma, os coletores podem organizar e direcionar corretamente o fluxo de dados entre as portas disponibilizadas. Para dados de tráfego de rede a porta disponibilizada é a TCP/5000 e para dados de *log* de dispositivos *hosts* é utilizado a porta TCP/5001.

No segmento Logstash de Borda também são aplicadas padronizações nos campos dos objetos estruturados para que estes campos possam ser relacionados futuramente. Um exemplo de normalização aplicada neste segmento acontece nos campos de endereços IP, no qual o campo é convertido para o tipo *string*. Na subseção 3.2.1, será abordado em mais detalhes o funcionamento do Logstash de Borda, quando submetido à um fluxo de dados.

Para a Base de Conhecimento a tecnologia MongoDB é utilizada para o armazenamento, porque oferece simplicidade no registro de objetos JSON e por armazenar todas as informações de cada objeto em uma única instância no banco de dados, facilitando o mapeamento/organização dos objetos (MONGODB, 2019a).

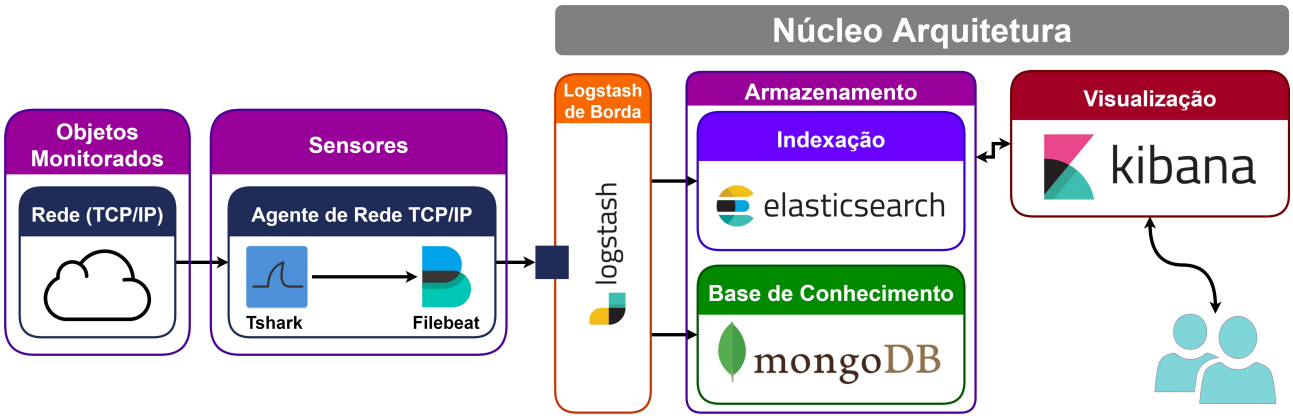
Na Indexação e Visualização são usados os softwares ElasticSearch e Kibana respectivamente. Ambos os softwares pertencem à mesma organização, portanto, oferecem ótima compatibilidade (ELASTIC, 2019b; ELASTIC, 2019a).

As tecnologias de armazenamento trabalham com dados replicados e estabelecem segurança devido a redundância. As tecnologias MongoDB e ElasticSearch são complementares e juntas oferecem o armazenamento consistente e velocidade na busca de dados.

3.2.1 Fluxo de Dados

Para exemplificar o fluxo de dados do protótipo da Arquitetura de Monitoramento foi elaborada a Figura 3.4, que apresenta o fluxo ocorrido no monitoramento de redes TCP/IP.

Figura 3.4. Fluxo de monitoramento para a rede TCP/IP



Fonte: Autoria própria

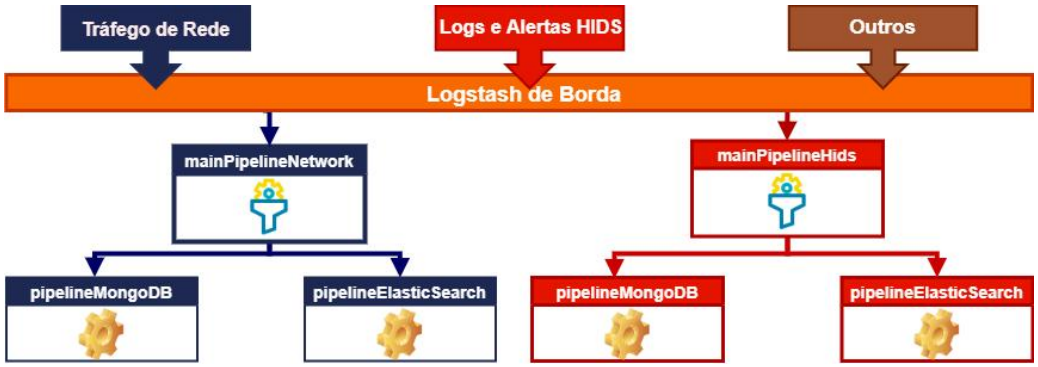
Inicialmente, os pacotes de rede são capturados pelo sensor (Tshark) e coletados pelo coletor (FileBeat) que controla os dados que precisam ser enviados para o segmento Logstash de Borda. A entrada dos dados coletados da rede acontece através da porta TCP/5000 no Logstash de Borda. A divisão em fluxos de entradas no Logstash de Borda permite que os dados possam ser encaminhados para *pipelines* específicos de acordo com cada tipo de entrada.

Parte significante das contribuições deste trabalho, foram destinadas a construção de *pipelines* para pré-processar, filtrar e enriquecer dados coletados de hosts, redes TCP/IP e ModBus. O processamento no Logstash de Borda é realizado por meio de arquivos de configuração de processamento do Logstash, conhecidos como *pipelines*. Cada *pipeline* possui três estágios que são: Entradas, Filtros e Saídas. As entradas recebem os dados, os filtros os modificam e as saídas enviam tais dados processados para outro lugar.

Na Figura 3.5 o fluxo de entrada de tráfego de rede (azul) é direcionado para o *pipeline* `mainPipelineNetwork` no qual são realizadas filtragens, normalizações e enriquecimento dos dados de tráfego de rede. Para a rede em questão (TCP/IP), são realizadas normalizações de campos que possuem endereços IP, portas, números para os respectivos tipos de dado. As normalizações fazem diferença para elaborar visualizações gráficas que utilizam correlações e comparações porque, com as normalizações é possível realizar padronizações, por exemplo, o campo *timestamp* que indica um instante único no tempo, é sempre traduzido para o tipo *Unix Timestamp*.

Correlações de dados podem acontecer, por exemplo, se no monitoramento de rede for detectado o crescimento anormal de tráfego *Secure Socket Shell* (SSH). É possível verificar quais são os dispositivos que estão se comunicando por meio de SSH e assim averiguar nos dados registrados, se houve alguma tentativa de intrusão para estes dispositivos. Desta forma, ataques podem ser identificados, favorecendo a mitigação de possíveis ataques.

Figura 3.5. Fluxos de entradas e direcionamentos no Logstash de Borda



Fonte: Autoria própria

O Código 1 mostra um trecho do *pipeline* `mainPipelineNetwork`. Tal trecho é responsável pelo enriquecimento de dados com coordenadas geográficas dos endereços IP nas origens dos pacotes de rede. O código ao identificar a presença do campo IP na origem do pacote (`ip_ip_src`), faz uso da API *online* "geoip" do próprio Logstash, para encontrar dados de geolocalização do endereço, se for encontrado, um novo campo é adicionado a estrutura do dado contendo informações sobre a geolocalização como latitude, longitude, país e outras.

Se não for identificado uma geolocalização do endereço através da API "geoip", a arquitetura busca o endereço IP em um arquivo de dicionário preestabelecido. Outras regiões do `mainPipelineNetwork`, lidam com endereços de entrada e saída *Internet Protocol version 4* (IPv4) e *Internet Protocol version 6* (IPv6), e apresentam estrutura bem semelhante ao Código 1.

Os arquivos de dicionários são coleções de itens com elementos estruturados em chave e valor. Os dicionários podem ser construídos pelo administrador da arquitetura e, conter informações úteis para o enriquecimento de dados, por exemplo, coordenadas geográficas de endereços IP locais, informações mais refinadas do posicionamento de dispositivos físicos dentro de uma indústria e códigos de funções para determinados protocolos. Alguns dicionários esclarecidos na subseção 3.2.3, foram implementados no protótipo da arquitetura para contribuir com o enriquecimento das informações coletadas. Tais dicionários estão disponíveis publicamente no repositório deste trabalho <https://github.com/rmmenezes/prototipo-arq-mononitoramento>https://github.com/rmmenezes/prototipo-arq-mononitoramento (BARBOZA, 2019).

Código 1 Enriquecimento de dados com endereço físico

```
# Adiciona cordenadas geográficas (IP) IPV4
if [packet][layers][ip] {

    # GeoIP para endereços de Origem
    geoip {
        source => "[packet][layers][ip][ip_ip_src]"
        target => "[geo_location][source]"
    }

    # Caso não encontrado localização, verificar em um dicionário pre-definido
    if ![geo_location][source][location] {
        translate {
            exact => true
            regex => true
            override => true
            field => "[packet][layers][ip][ip_ip_src]"
            destination => "geo_point_source"
            dictionary_path => "../pathExemple/databaseGeoIPLocal.yml"
            fallback => "null"
            refresh_behaviour => "replace"
        }
        json {
            source => "geo_point_source"
            target => "[geo_location][source]"
            add_field => { "[geo_location][source][ip_ip_src]" =>
            "%{[packet][layers][ip][ip_ip_src]}" }
            remove_field => [ "geo_point_source" ]
        }
        mutate {
            remove_tag => [ "_geoip_lookup_failure" ]
        }
    }
}
```

Para o armazenamento no ElasticSearch alguns campos desnecessários para as visualizações podem ser removidos, a fim de organizar melhor os dados e trazer mais desempenho aos índices do ElasticSearch, porém o administrador da arquitetura pode incluir ou não determinados campos aos índices manualmente, caso necessário. Para o armazenamento no MongoDB, é recomendado registrar os dados com poucas ou nenhuma remoção e alteração de campos. Tal recomendação tem objetivo de manter o MongoDB com os dados íntegros assim como foram coletados e desta forma, facilitar que posteriormente outras tecnologias e ferramentas de análise possam utilizar tais dados armazenados no MongoDB da forma mais natural possível.

O Kibana viabiliza a construção de telas *dashboards*, que são agrupamentos de visualizações gráficas. Com o Kibana inserido na arquitetura de monitoramento, profissionais da área de segurança cibernética podem desenvolver suas próprias telas com as visualizações mais relevantes para o contexto de monitoramento de anomalias e ciberataques. Para construir visualizações no Kibana são realizadas buscas no Indexador, dos campos e valores que serão representados ou correlacionados pelas visualizações gráficas. Após a construção das *dashboards* com as visualizações, o usuário Kibana pode optar por, atualizar automaticamente as visualizações gráficas com os dados mais recentes no ElasticSearch, ou definir um intervalo de tempo para que o painel atualize as informações nas visualizações.

A Tabela 3.1 apresenta as portas de comunicação abertas para conexões na arquitetura. Nota-se que o Logstash de Borda possui as portas 5000/TCP e 5001/TCP disponíveis para que coletores possam conectar e enviar dados. A porta 5000/TCP recebe conexão dos coletores de rede, enquanto a porta 5001/TCP recebe conexões dos coletores de *hosts*. O Logstash de borda estabelece conexão com o ElasticSearch e MongoDB, pelas portas 9600/TCP e 27017/TCP respectivamente. A interface Kibana responsável por apresentar os dados graficamente, conecta ao ElasticSearch por meio da porta 9600/TC

Tabela 3.1. Portas e serviços de conexão

Componente	Porta	Protocol	Proposito
Logstash de Borda	5000	TCP	Entrada de dados de rede dos sensores
	5001	TCP	Entrada de dados de host dos sensores
ElasticSearch	9600	TCP	REST API do ElasitcSearch
MongoDB	27017	TCP	REST API do MongoDB
Kibana	5601	TCP	Interface Web Kibana

Fonte: Autoria própria

3.2.2 Segurança no Protótipo da Arquitetura de Monitoramento

O protótipo da arquitetura de monitoramento opera com dados sensíveis, arquivos de *log*, tráfego de rede e métricas de dispositivos. Então, medidas de segurança devem ser implementadas para garantir a confidencialidade e integridade dos dados armazenados ou em trânsito (CERT.BR, 2003).

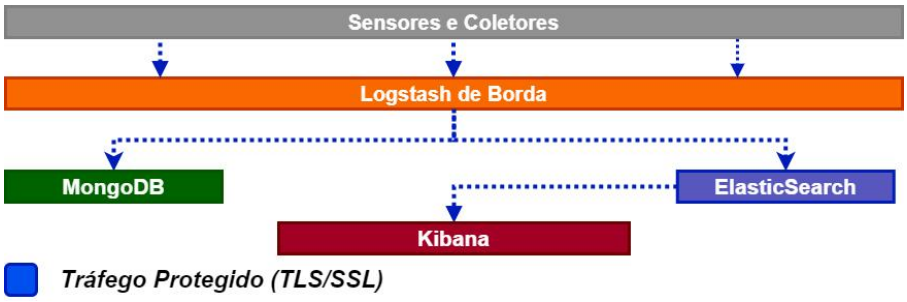
Foram implementados no núcleo do protótipo, métodos de autenticação por *login* e senha com níveis programáveis de privilégio de acesso. Desta forma, diversos usuários podem ser criados por um administrador, e posteriormente tais usuários podem visualizar diferentes perspectivas dos dados apresentados no indexador e na visualização, impedindo que usuários não autorizados vejam, alterem ou danifiquem dados do protótipo da arquitetura.

Em conjunto com a autenticação, foi implementado em cada segmento comunicação por *Transport Layer Security* e *Secure Sockets Layer* (TLS/SSL), que são protocolos de criptografia que garantem a confidencialidade e integridade das conexões, além de garantir autenticação quando certificados digitais são apresentados pelo cliente e/ou servidor (SCHäFFER, 2016).

A Figura 3.6 ilustra em azul pontilhado o fluxo de dados protegido com os protocolos criptográfico (TLS/SSL), presente em todas as vias de comunicação entre os segmentos. Nos segmentos MongoDB, ElasticSearch e Kibana foram definidas credenciais (*login* e senha) diferentes para cada um que deseja-se comunicar com tais segmentos.

Um exemplo de fluxo de comunicação autenticado e seguro, acontece no Logstash de Borda que recebe credenciais de acesso dos segmentos MongoDB e ElasticSearch, e envia os dados para tais segmentos utilizando protocolos de criptografia (TLS/SSL).

Figura 3.6. Fluxo de comunicação segura entre os segmentos



Fonte: Autoria própria

A adição desses mecanismos de segurança apresentados proporciona maior confiabilidade e responsabilidade com os dados no protótipo, outros métodos, como exemplo, sistemas de *firewall*, também podem ser anexados aos métodos já existentes reforçando a segurança. Se posteriormente houver a necessidade de adicionar novos segmentos ou aplicações ao núcleo, estes devem aderir aos padrões de segurança já existentes, o que significa, utilizar credenciais de acesso e canais de comunicação protegidos para realizar comunicações.

3.2.3 Implantação do Ambiente

O protótipo da arquitetura de monitoramento por ser modular, permite a troca ou acoplamento de novos segmentos. Desta forma, a solução de monitoramento pode ser enquadrada em diferentes ambientes de produção, viabilizando a escalabilidade e a distribuição dos segmentos do protótipo da arquitetura em diversos servidores.

A acoplagem de módulos deve seguir o padrão de entrada e saída definidos na arquitetura, que são objetos JSON. Um exemplo de acoplagem de um novo módulo pode acontecer se houvesse a necessidade de monitorar um novo segmento de rede industrial isolado das demais redes. Neste caso seria necessário implantar sensores e coletores compatíveis com o novo segmento de rede, e após isso disponibilizar uma porta de comunicação no arquivo de configuração do Logstash de Borda, para que a nova rede possa ser monitorada.

Neste trabalho foi desenvolvido um protótipo para testes dos segmentos da arquitetura, podendo viabilizar a realização dos experimentos em uma única máquina física. Os segmentos do núcleo da arquitetura foram implementados em contêineres Docker isolados. Foi utilizado o Docker Compose para organizar o ambiente de testes e facilitar alterações em arquivos de configurações de cada serviço.

O núcleo da arquitetura e seus segmentos, por trabalharem com grandes volumes de dados, quando executados simultaneamente, podem exigir muito das configurações de *hardware* do dispositivo no qual está sendo executado a arquitetura. Portanto, é desejável que em ambiente de produção, os segmentos possam executar em servidores ou máquinas isoladas, e que tais dispositivos possuam *hardwares* compatíveis para suportar as atividades de cada segmento.

Para melhorar o desempenho no armazenamento e indexação dos dados coletados pelos sensores, foi realizado o pré-processamento dos dados antes do armazenamento, esta tarefa é realizada no segmento de Logstash de Borda, e possibilita enriquecer o dado acrescentando novos campos ou remover dados desnecessários para o monitoramento como exemplo, informações repetidas e campos de controle gerados automaticamente pelas ferramentas de coleta.

No contexto de enriquecimento dos dados, o administrador da arquitetura pode construir diversos dicionários locais ou fazer uso de APIs para obter mais informações dos campos nos dados. Neste trabalho, utilizamos a API *online* do Logstash "geoip" para obter dados de geolocalização de endereços IP, mas também foram implantados alguns dicionários locais os quais podem ser modificados e incorporadas novas informações.

Quando a API *online* do Logstash "geoip" não obtêm resultados sobre o endereço, o mesmo é submetido a buscas no dicionário local de endereços IP representado pelo trecho de Código 2. Tal dicionário pode ser muito útil, por exemplo, se o administrador da rede precisar gerenciar a geolocalização de dispositivos de endereços locais dentro de um galpão industrial, ou até mesmo acrescentar informações ao dado de acordo com setores de atuação.

Código 2 Dicionário local de endereços IP

```
...
// IPV4 Classe C
"(^192\.168\.)" : {"timezone":"America/Sao_Paulo",
                   "continent_code":"BR",
                   "country_name":"Brazil",
                   "country_code2":"BR",
                   "country_code3":"BR",
                   "region_name":"Paraná",
                   "latitude":-24.043740,"longitude":-52.378071,
                   "location":{"lat":-24.043740,"lon":-52.378071}
},
// LocalHost
"(^127\.)|" : {"timezone":"America/Sao_Paulo",
                 "continent_code":"BR",
                 "country_name":"Brazil",
                 "country_code2":"BR",
                 "country_code3":"BR",
                 "region_name":"Paraná",
                 "latitude":-25.462632,"longitude":-54.583335,
                 "location":{"lat":-25.462632,"lon":-54.583335}
},
...
```

O Código 3 apresenta um trecho do dicionário de portas do protocolo TCP. Através deste dicionário a arquitetura pode trazer mais informações a respeito dos serviços e protocolo dos pacotes de rede TCP. Para pacotes UDP, um dicionário muito semelhante ao Código 3 também foi implementado, possibilitando o mesmo enriquecimento dos dados para a categoria de pacotes de rede UDP.

Código 3 Dicionário de serviços TCP

```
...
"22": {
    "service_name": "ssh",
    "transport_protocol": "tcp",
    "description_service": "The Secure Shell (SSH) Protocol"},
"23": {
    "service_name": "telnet",
    "transport_protocol": "tcp",
    "description_service": "Telnet"},
"8080": {
    "service_name": "http-alt",
    "transport_protocol": "tcp",
    "description_service": "HTTP Alternate (see port 80)"},
...
```

No Código 4, é representado um dicionário de códigos de serviços do protocolo ModBus. Tais códigos de serviços são utilizados na comunicação entre os dispositivos industriais e, representam as funções requisitadas pelo pacote de rede ModBus (leitura, escrita, etc). O Dicionário de códigos de serviços ModBus foi implementado na arquitetura e, auxilia no enriquecimento e, na tradução de códigos numéricos para o nome da função por escrito, ajudando na construção de visualizações gráficas mais simples de interpretar.

Código 4 Dicionário de códigos de função ModBus

```
...
"2": {"function_name": "Read Discrete Input"},
"3": {"function_name": "Read Holding Registers"},
"4": {"function_name": "Read Input Registers"},
"5": {"function_name": "Write Single Coil"},
...
```

3.3 Considerações do Capítulo

Neste capítulo foi desenvolvido uma arquitetura de monitoramento, que se destaca por ser modular e por suportar o acoplamento de diferentes dispositivos e tecnologias a serem monitoradas. Um protótipo da arquitetura foi desenvolvido utilizando softwares livres, sinalizando que tal arquitetura pode ser implementada no meio de produção com baixo custo agregado.

A arquitetura de monitoramento armazena os dados coletados pelos sensores em duas bases, sendo uma das bases um Indexador que organiza os dados armazenados em índices de forma a otimizar buscas e manipulações dos dados, consequentemente contribuindo com a velocidade na construção de visualizações gráficas.

As visualizações gráficas quando analisadas por profissionais da área da segurança cibernética ou redes de computadores viabilizam a análise de comportamento dos objetos monitorados, podendo resultar na detecção manual/visual de anomalias e ciberataques.

4 EXPERIMENTOS E RESULTADOS

Neste capítulo são descritos os experimentos com o objetivo de efetuar uma prova de conceito da efetividade no funcionamento da arquitetura de monitoramento. Na Seção 4.1, são abordados os experimentos realizados para o monitoramento de *hosts*, rede TCP/IP e rede ModBus. Também, são discutidos alguns detalhes da implementação da arquitetura em módulos virtuais e a obtenção dos dados para alimentar os componentes da arquitetura. Por conseguinte, a Seção 4.2 apresenta os resultados dos experimentos e a utilização do Kibana para visualizações gráficas.

4.1 Experimentos

Para realizarmos experimentos e comprovar o funcionamento da arquitetura bem como sua efetividade no monitoramento de redes e *hosts*, implementamos os módulos do protótipo da arquitetura apresentados na seção 3.2, em um computador Notebook Dell, Intel i7-7500U 2.70GHz, 8GB de memória RAM, com sistema operacional Linux Ubuntu 18.04.3 LTS x64.

A implementação da arquitetura em um único computador é possível em razão de que cada segmento da arquitetura utiliza máquinas virtuais (contêineres Docker) para simular e suprir as os recursos e configuração dos softwares alocados em cada segmento. Porém, é altamente recomendável que em modo de produção, cada segmento da arquitetura seja implementado em máquinas diferentes e distribuídas, contribuindo com o desempenho e segurança das informações.

O objetivo do experimento é realizar o monitoramento de duas redes e *hosts*:

(i) Rede TCP/IP tradicional representando o setor corporativo de uma indústria no qual podemos encontrar diversos tipos de protocolos e diversidade de máquinas, bem como acessos externos;
(ii) Rede industrial, com protocolo de rede ModBus, na qual podemos encontrar o tráfego de comunicação existente entre os dispositivos industriais da linha de produção;
(iii) *Hosts*, com sistemas operacionais Windows e Linux.

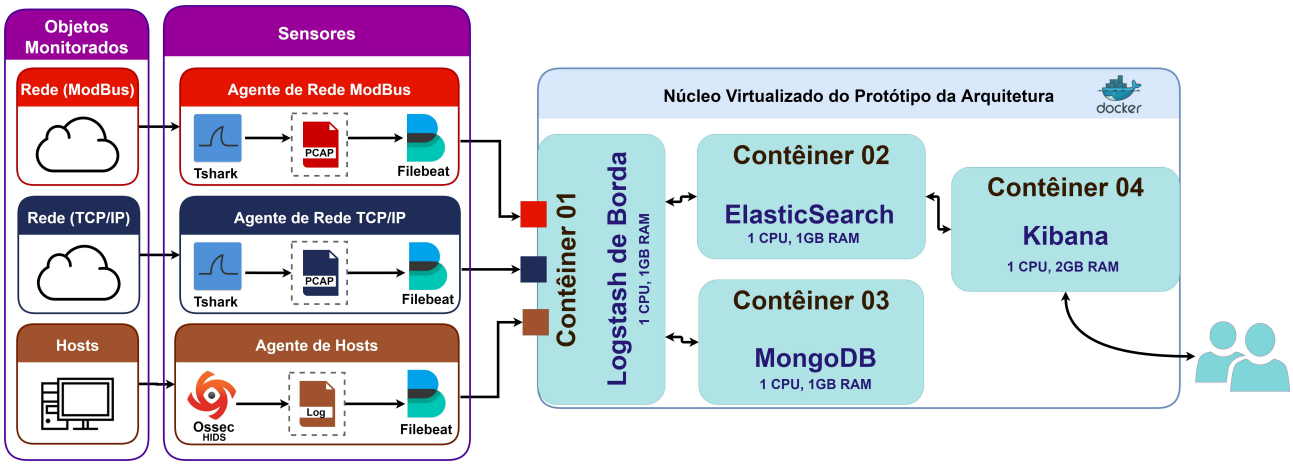
Um dos principais problemas na realização deste experimento, é a falta dos equipamentos industriais e redes para realizar a coleta de informações pelos sensores da arquitetura. A maneiras que encontramos para enfrentar esta dificuldade foi utilizar arquivos de saída dos sensores disponíveis na Internet. Desta forma, é possível simular o arquivo de saída dos sensores, e assim prosseguir normalmente com o fluxo normal da arquitetura (BARBOZA, 2019).

A Figura 4.1 ilustra o ambiente de experimentos. É possível observar que foi realizado o monitoramento de duas redes e *hosts*, no segmento Sensor e Coletor está presente o arquivo de saída de cada sensor, que foi previamente capturado e disponibilizado publicamente na Internet no repositório deste projeto no GitHub (BARBOZA, 2019). Os arquivos de saída de cada sensor são armazenados em diretórios monitorados pelo Coletor Filebeat de cada segmento. O Filebeat observa alterações e acréscimos de dados nos arquivos de saída dos sensores, e realiza o controle dos dados

que já foram enviados na arquitetura e os que necessitam ser enviados. Os dados, portanto, seguem o fluxo da arquitetura até que possam ser visualizados pelo usuário final.

Como mostrado na Figura 4.1, o ambiente de experimentos possui em seu núcleo quatro contêineres virtuais Docker. O primeiro contêiner dispõem do software Logstash e atua como uma barreira de entrada dos dados coletados e enviados dos objetos monitorados. O segundo e o terceiro contêiner são reesposáveis pelo armazenamento dos dados e possuem instalados os softwares ElasticSearch e MongoDB respectivamente. O quarto e último contêiner possui o software Kibana responsável pela visualização dos dados armazenados.

Figura 4.1. Ambiente de experimentos



Fonte: Autoria própria

Os resultados do experimento no monitoramento das redes e *hosts* comprovam a efetividade do monitoramento de tais objetos. Para cada um dos objetos monitorados foi possível elaborar visualizações gráficas com os dados coletados e tratados pela arquitetura. Foram disponibilizas no repositório deste trabalho, versões interativas das visualizações gráficas elaboradas e apresentadas na Seção 4.2 (BARBOZA, 2019).

4.2 Resultados

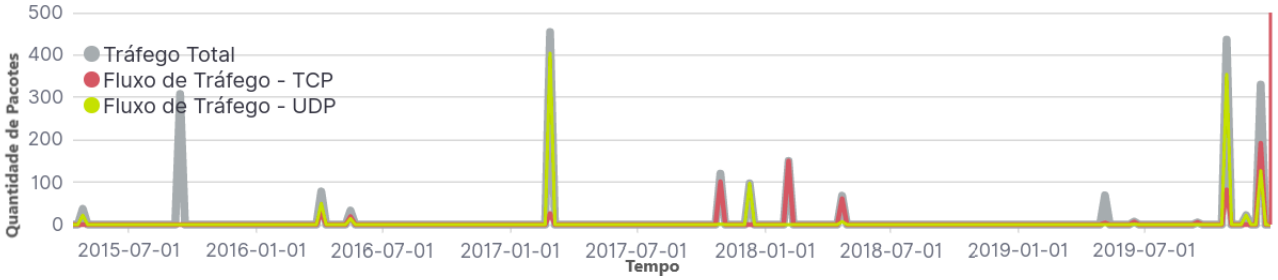
A seção de resultados foi dividida em subseções contendo os resultados e visualizações gráficas de cada objeto monitorado e uma breve discussão sobre a interpretação das visualizações para identificar possíveis ataques e anomalias. A subseção 4.2.1, traz os resultados do monitoramento de uma rede TCP/IP, comumente encontrada em ambientes corporativos. Na subseção 4.2.2, apresentamos os resultados de monitoramento de uma rede tipicamente industrial, a qual faz uso do protocolo ModBus. Por fim, a subseção 4.2.3, exibe os resultados do monitoramento de diversos *hosts*, com sistemas operacionais Windows e Linux.

4.2.1 Monitoramento em rede TCP/IP

No experimento, para o monitoramento de rede TCP/IP foi utilizado um arquivo de tráfego de rede previamente capturado, contendo pacotes de rede com diversos protocolos de comunicação. Os próximos parágrafos trazem algumas das visualizações geradas pela arquitetura no monitoramento da rede TCP/IP convencional.

A Figura 4.2 exibe uma visualização de séries temporais, na qual é relacionado o tempo pela quantidade de tráfego de rede TCP e UDP. A distribuição da quantidade de pacotes pelo tempo pode contribuir para identificar anomalias de desvios comportamentais. Por exemplo, ataques cibernéticos que geram grandes fluxos de pacotes podem ser identificados em tal visualização.

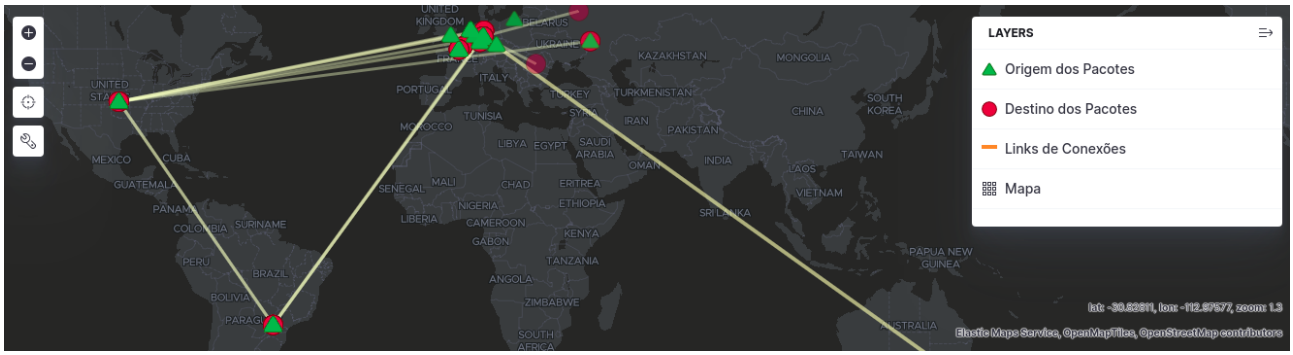
Figura 4.2. Fluxo de rede temporal



Fonte: Autoria própria

A visualização por meio de mapas(Figura 4.3) permite visualizar conexões e acessos entre os endereços e dispositivos espalhados pelo mundo. Um exemplo de anomalia poderia ser identificada se houvesse aumento significativo de conexões com endereços externos desconhecidos. Uma análise mais detalhada poderia ser realizada para identificar quais máquinas estão participando do tráfego e o motivo das conexões.

Figura 4.3. Mapa global de conexões

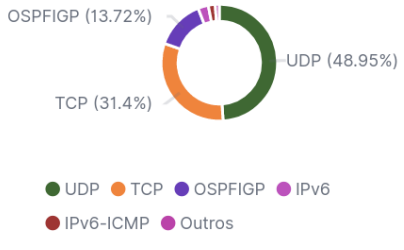


Fonte: Autoria própria

A Figura 4.4 apresenta duas visualizações relacionadas aos protocolos de comunicação e serviços mais frequentes na rede. O gráfico de fatias ilustra proporcionalmente, alguns dos protocolos

de transporte mais utilizados na rede. A visualização em tabela apresenta com mais detalhes a correlação entre os protocolos de comunicação e portas de serviço. No gráfico de fatias nota-se que a maior parte do tráfego de rede são pacotes UDP e, na tabela, alguns dos serviços e portas de comunicação que utilizam o protocolo de transporte UDP. Como exemplo, o serviço de "domain" (DNS) que faz uso da porta UDP/53 para traduzir/resolver domínios e IP.

Figura 4.4. Protocolos de comunicação



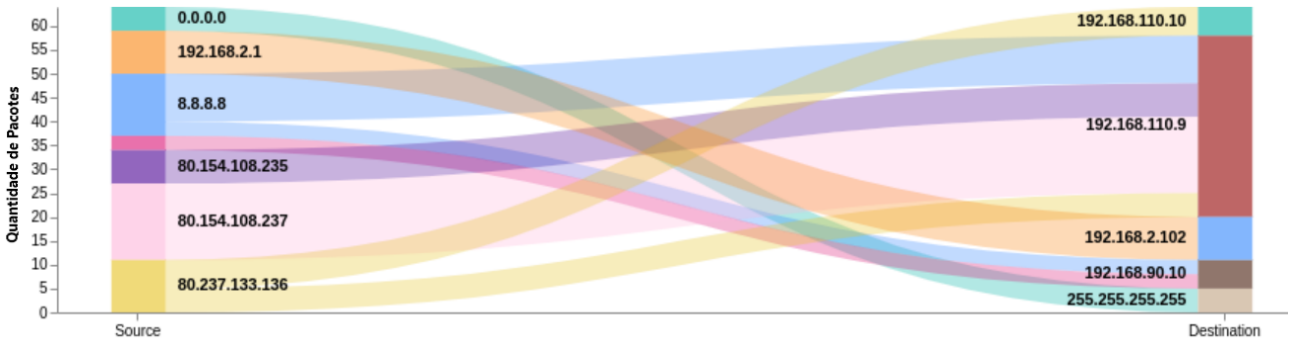
Porta / Nome do Serviço	Protocolo de Transporte	Descrição	Qtd
53 domain	udp	Domain Name Server	248
43 nicname	tcp	Who Is	192
2029 hsrpv6	udp	Hot Standby Router Protocol IPv6	156
1985 hsrp	udp	Hot Standby Router Protocol	156
179 bgp	tcp	Border Gateway Protocol	130
53 domain	tcp	Domain Name Server	110

Fonte: Autoria própria

O diagrama de Sankey, Figura 4.5, ilustra uma estrutura de rede, no qual a largura das vias que conectam os eixos, são proporcionais a quantidade de fluxo. A visualização representa os principais fluxos de comunicação de endereços IP existentes em redes. A coluna a direita, rotulada como *Source*, corresponde aos endereços IP de origem dos pacotes e a coluna à esquerda, rotulada como *Destination*, corresponde aos endereços IP de destino dos pacotes de rede.

Ao aplicar o diagrama de Sankey, nota-se a formação de fluxos entre as duas colunas, o que representa a comunicação existente entre os endereços IP de origem e destino. Quanto maior a largura do caminho de ligação, maior a quantidade de pacotes trocados. Várias características podem ser extraídas da visualização, como exemplo, endereços que geram mais fluxos de rede, comunicações de origem e destino, quantidade de endereços que se conectam a um endereço IP específico, dentre outras.

Figura 4.5. Fluxos de rede representados pelo diagrama de Sankey



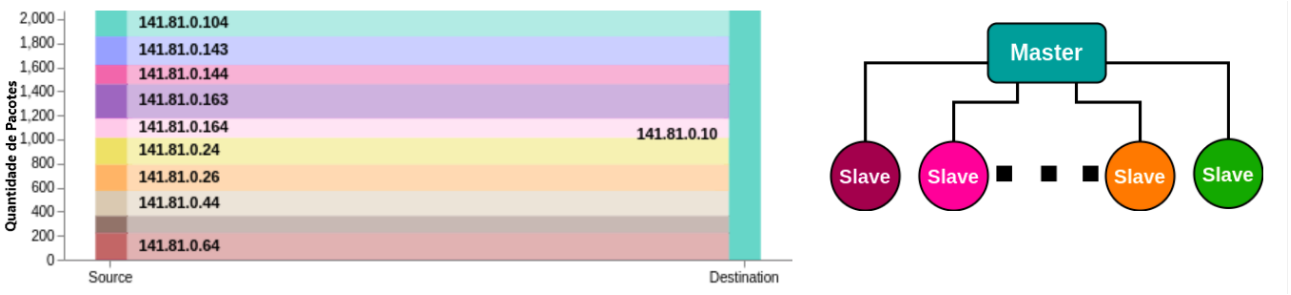
Fonte: Autoria própria

4.2.2 Monitoramento em rede ModBus/TCP

No experimento, foi utilizado um tráfego previamente coletado e armazenado de rede industrial ModBus/TCP para simular o trabalho realizado pelos sensores de rede. Como resultado, algumas visualizações gráficas foram desenvolvidas, tais visualizações podem auxiliar profissionais da área de redes e segurança cibernética no monitoramento e gerenciamento da rede industrial ModBus/TCP.

O modelo de comunicação do protocolo ModBus é do tipo *master-slave* (mestre-escravo), no qual somente o dispositivo *master* pode realizar as requisições de dados aos dispositivos *slaves*. A visualização da Figura 4.6 possibilita observar o fluxo de comunicação existente entre os dispositivos da rede industrial ModBus. Assim, podemos deduzir que no conjunto de dados utilizado, o endereço IP "141.81.0.10" é o dispositivo *master*, e que recebe todo fluxo de dados dos dispositivos *slaves*. Uma possível anomalia poderia ser identificado caso algum dispositivo *slave* passa-se a responder a um servidor *master* malicioso.

Figura 4.6. Fluxo de comunicação ModBus *master-slave*

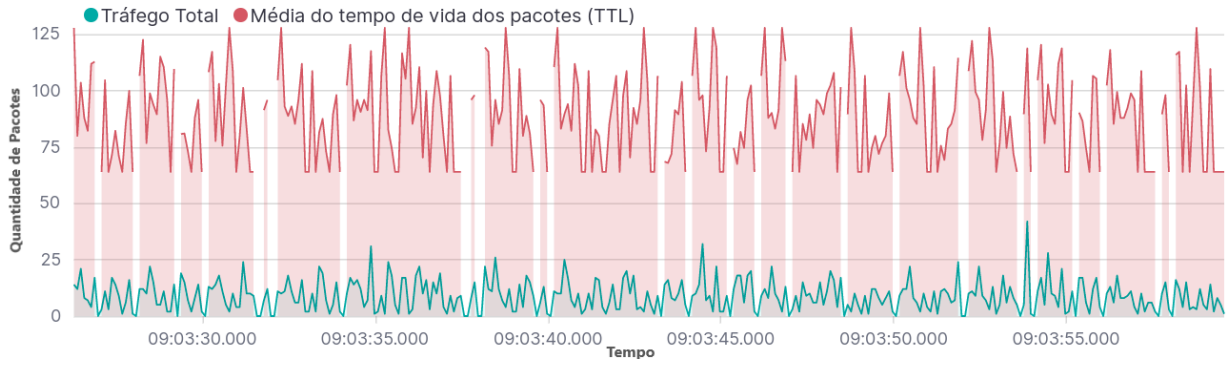


Fonte: Autoria própria

Na Figura 4.7 é exposto uma visualização de séries temporais relacionando o fluxo de tráfego de rede ModBus com a média do tempo de vida dos pacotes (TTL). O TTL representa o número de saltos entre máquinas e roteadores que os pacotes podem realizar antes de serem descartados. O valor TTL dos pacotes é decrementado na medida em que tais pacotes passam pelos os dispositivos da rede (Han et al., 2012). Os dispositivos que recebem pacotes com TTL zero devem informar para a origem do pacote que a houve a expiração, e que tal pacote será descartado.

Com a visualização da Figura 4.7 é possível identificar ataques DDoS de expiração de TTL, que afeta a disponibilidade de dispositivos da rede. No ataque, agentes DDoS enviam pacotes de rede no qual o valor TTL se torna zero nas vítimas (*hosts*, roteadores e *switch*). Isto faz com que as vítimas consumam todo poder de processamento em responder as origens de cada um dos pacotes descartados, podendo tornar as vítimas indisponíveis durante o ataque (Han et al., 2012).

Figura 4.7. Fluxo de rede ModBus relacionado a média dos valores TTL dos pacotes



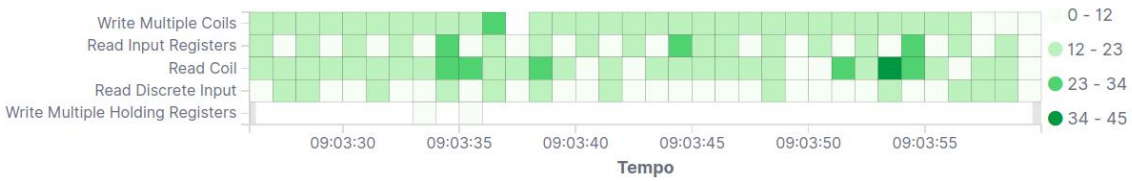
Fonte: Autoria própria

No protocolo ModBus, as operações de leitura, escrita e outras são categorizadas através de códigos de função. Desta forma, o dispositivo mestre envia comandos numéricos aos dispositivos escravos para representar qual operação deve ser realizada. Para melhor interpretação, os códigos de função foram categorizados e traduzidos pela arquitetura para a descrição verbal da operação.

As operações realizadas pelo mestre podem ser monitoradas através da Figura 4.8. Conhecida como mapa de calor *(Heatmap)*, a visualização faz uso da intensidade de cor para representar o número de ocorrências. No eixo Y da visualização estão os códigos dos comandos traduzidos e no eixo X o domínio do tempo. A legenda da visualização apresenta o fator de escala para cada intensidade de cor, quanto mais escuro maior a demanda de solicitações do comando naquele instante.

Redes industriais costumam ter padrões bem definidos no tempo de leitura dos registradores e realização de operações. A visualização de mapa de calor pode contribuir para extrair o comportamento padrão executado nos dispositivos, desta forma, colaborar na identificação visual de comportamentos anômalos.

Figura 4.8. Mapa de calor para comandos da rede ModBus



Fonte: Autoria própria

4.2.3 Monitoramento em *Hosts*

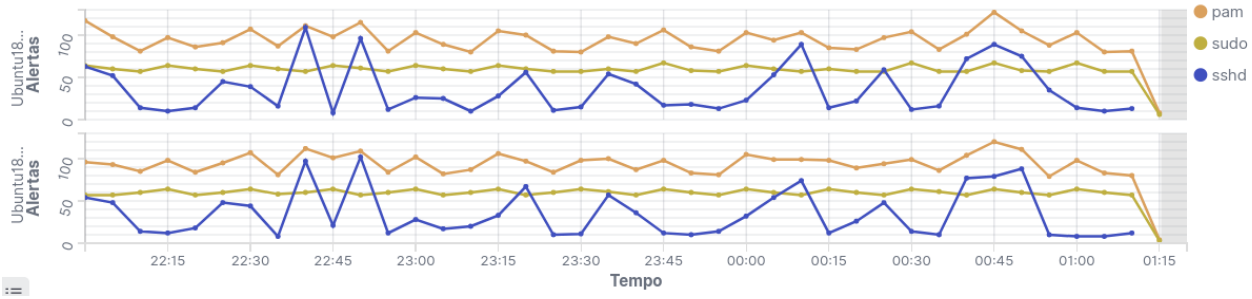
A ferramenta OSSEC monitora o comportamento de cada dispositivo em que foi instalado e reporta por meio de alertas os comportamentos considerados relevantes, suspeitos e maliciosos. Para o monitoramento de *hosts*, foram utilizados arquivos *log* de diversos *hosts*, contendo alertas

OSSEC. Algumas visualizações foram desenvolvidas a fim de monitorar o comportamento dos dispositivos/*hosts* monitorados.

A Figura 4.9 apresenta os dois dispositivos que mais geraram alertas OSSEC durante o monitoramento e os serviços críticos do sistema operacional, que foram utilizados ou requisitados nas ocorrências dos alertas.

Com ajuda do gráfico da Figura 4.9, é possível correlacionar os serviços para obter características enriquecidas dos *hosts*, por exemplo, o "sshd" representado pela linha azul remete ao serviço do servidor de conexões SSH, o serviço "pam" *(Pluggable Authentication Modules)* em laranja, é responsável por realizar a autenticação de usuários nos ambientes Linux/Unix. Percebe-se que ambos serviços são requisitados no mesmo momento na linha temporal (23:00, 00:00 e 00:45), este comportamento é coerente e, ocorre devido o serviço "sshd" fazer uso do serviço "pam" na autenticação de usuário via SSH.

Figura 4.9. Serviços críticos requisitados na ocorrência dos alertas

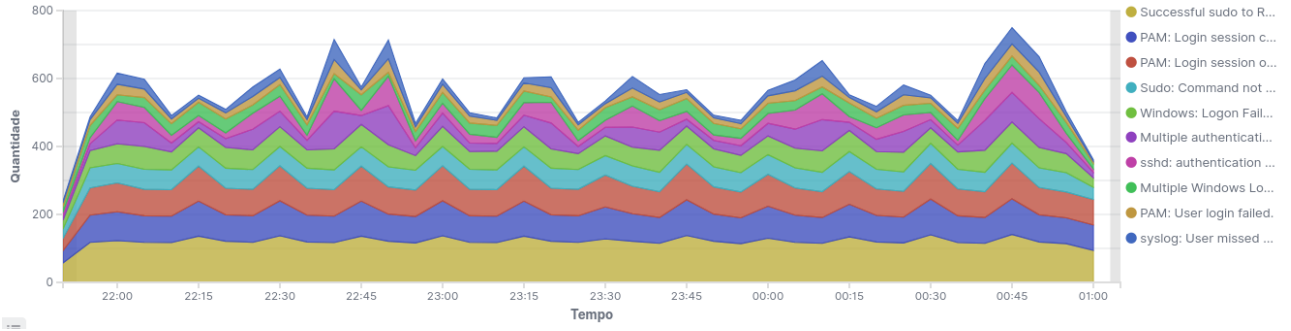


Fonte: Autoria própria

A visualização na Figura 4.10 remete as regras em que os alertas capturados foram classificados pelo OSSEC no momento da ocorrência. A legenda da visualização mapeia cada regra por uma cor única, a largura de cada fatia é referente a quantidade de alertas de tal categoria naquele instante.

Monitorar o fluxo das categorias de alertas no tempo de ocorrência pode ajudar a identificar visualmente ataques com acessos suspeitos por SSH, manipulação de arquivos protegidos e autenticação por ataque de força bruta *(Brute Force)*. Desta forma, se por exemplo, houvesse um ataque de dicionário para descobrir as credenciais de acesso de um dispositivo monitorado, a visualização representaria picos com alertas relacionados a suscetíveis falhas de autenticação.

Figura 4.10. Alertas OSSEC no tempo da ocorrência

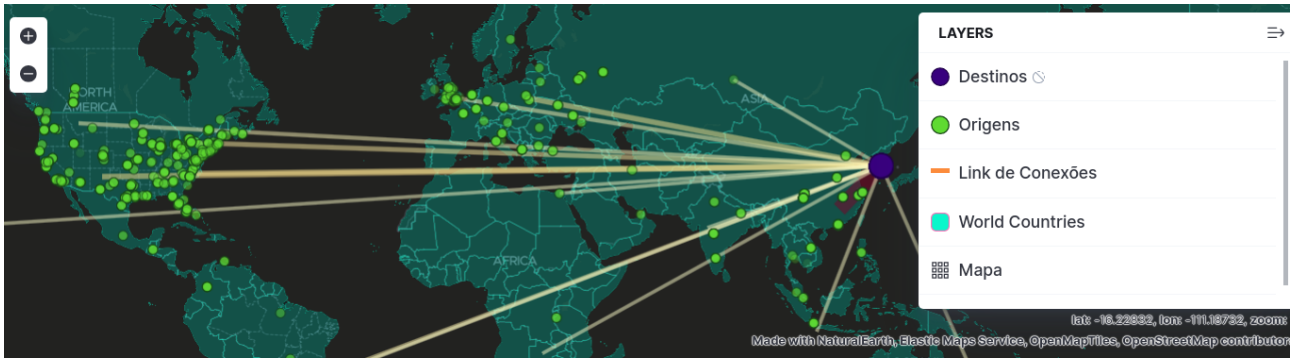


Fonte: Autoria própria

Em situações no qual o alerta gerado pelo OSSEC está relacionado a acessos externos, assim como conexões HTTP, SSH e FTP, é interessante conhecer a localização física dos dispositivos envolvidos. Sendo assim a visualização da Figura 4.11 revela a localização e o *link* de conexões entre endereços de origem e destino.

Observa-se, no mapa da Figura 4.11, que grande parte das conexões partem de regiões Norte Americanas com destino a Coreia do Sul. Se for constatado que tais conexões fazem parte de ataques em massa a redes ou dispositivos, medidas paliativas poderiam ser estabelecidas como exemplo, regras de firewall para bloquear qualquer conexão vinda da região Norte Americana.

Figura 4.11. Mapa global de conexões externas em *logs* OSSEC



Fonte: Autoria própria

4.3 Considerações do Capítulo

Neste Capítulo foram descritos os experimentos para avaliar o funcionamento da arquitetura de monitoramento. Detectou-se um gargalo na arquitetura, presente nos segmentos Base de Conhecimento e Indexador. Ao requisitar o armazenamento de grandes quantidades de objetos JSON no banco de dados MongoDB, ocorre lentidão e perda de desempenho, o mesmo acontece ao realizar a indexação dos dados no ElasticSearch.

Como relatado em MongoDB (2019b) e Elastic (2019e), a perda de desempenho no armazenamento e na indexação pode estar relacionada as velocidades de processamento da máquina e taxa de armazenamento dos discos. Desta forma, algumas soluções propostas são a utilização de hardwares mais modernos e potentes, uso de *threads* para armazenamento em lotes com menor volume de dados e o uso da computação distribuída, dividindo os segmentos do protótipo da arquitetura em máquinas diferentes. Em contrapartida, os experimentos proporcionaram a validação da arquitetura e protótipo no monitoramento de *hosts*, redes TCP/IP e ModBus.

Por meio do software Kibana foi possível construir diversas visualizações gráficas a respeito de dados relevantes coletados dos objetos monitorados. Para cada visualização gráfica foi realizada uma breve análise da utilidade da mesma no monitoramento, gerenciamento e supervisão dos objetos monitorados. Com o Kibana, diversas visualizações podem ser desenvolvidas e analisadas pelos olhares técnicos de profissionais das áreas de segurança cibernética e rede de computadores. Portanto, tais visualizações podem contribuir na identificação de ataques e anomalias nos objetos monitorados.

5 CONCLUSÕES

Neste trabalho foi desenvolvida uma arquitetura de monitoramento e um protótipo da arquitetura proposta, utilizando softwares reais e consolidados por profissionais de segurança cibernética e *Big Data.* Com o protótipo, foram realizados experimentos para constatar a efetividade do monitoramento de redes e *hosts* em ambientes industriais. Os resultados dos experimentos apresentam a efetividade do protótipo da arquitetura de monitoramento, possibilitando a construção de visualizações gráficas com os dados dos objetos monitorados.

A solução de monitoramento abordada neste trabalho, divide as tarefas de monitoramento em Coletar, Tratar, Armazenar e Apresentar, tais etapas foram utilizadas na construção da arquitetura de monitoramento e protótipo, dividido os segmentos de forma semelhante a organização das principais tarefas da solução de monitoramento.

A arquitetura de monitoramento realiza a coleta e tratamento de informações dos objetos monitorados. Tais informações são armazenadas em dois segmentos de armazenamento distintos que juntos trazem os benefícios de velocidade em busca de informações e segurança no armazenamento baseado em ACID.

Os dados armazenados na arquitetura podem ser consultados a qualquer momento ou servirem como entrada para ferramentas de análise e visualização externa. Desta forma, a arquitetura conta com o segmento de visualização que está conectado ao segmento de armazenamento em índices.

No segmento de visualização, profissionais da área de segurança e redes podem consultar, filtrar e correlacionar dados dos objetos em monitoramento, desta forma auxiliando no desenvolvimento visualizações gráficas para supervisionar redes e *hosts* do ambiente industrial e corporativo.

5.1 Trabalhos Futuros

A segurança cibernética assim como outras áreas da computação, estão sempre em mudança e passando por atualizações para adequar novos cenários e perspectivas. A arquitetura de monitoramento para sistemas industriais, apresentada neste trabalho, por ser projetada de forma modular, facilita que novas funcionalidades possam ser projetadas e acopladas.

Como trabalhos futuros, poderiam ser desenvolvidos agentes de monitoramento contendo sensores e coletores específicos para monitorar outros tipos de tecnologias do setor industrial, tais como softwares industriais, *firmware* de roteadores, DNS e dispositivos de *firewall.*

Com os recursos de monitoramento em *firmwares*, por exemplo, os administradores da arquitetura podem projetar visualizações e identificar alterações de configuração de equipamentos, mesmo que tais configurações sejam realizadas de forma manual, por operários no ambiente de produção.

Segmentos referente a detecção e análise automatizada de eventos maliciosos, também poderiam ser acoplados na arquitetura para consumir os dados armazenados na base de conhecimento, e apresentar os resultados e alertas na interface de visualização Kibana ou em interfaces externas. Neste caso, IDS e IPS poderiam ser opções viáveis para detecção de eventos por assinatura e comportamento, podendo até ser programados para atuar diretamente nos objetos monitorados, realizando por exemplo, bloqueio de tráfego de rede ou interceptando serviços maliciosos em *hosts*.

A respeito do desempenho da arquitetura de monitoramento em ambiente de computação distribuída. Testes de *benchmark* poderiam ser realizados para entender melhor as limitações da arquitetura e identificar pontos de estresse, e que necessitem de otimizações.

Mesmo diante dos trabalhos futuros que ainda podem ser realizados, o monitoramento empregado pela arquitetura pode ser utilizado por profissionais de segurança cibernética no gerenciamento e supervisão de redes e *hosts* em ambientes industriais.

REFERÊNCIAS

AHLAWAT, Abhishek. ***What is Pipelining?*** 2017. Disponível em: https://www.studytonight.com/computer-architecture/pipelining, Acessado em: 17/11/2019.

ASRODIA, Pallavi; PATEL, Hemlata. ***Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis***. 2012.

BAJER, Marcin. **Building an IoT Data Hub with Elasticsearch, Logstash and Kibana**. In: *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. [S.l.]: IEEE, 2017.

BAKUEI RYAN FLORES, Vladimir Kropotov Matsukawa. ***Securing Smart Factories Matsukawa Bakuei, Threats to Manufacturing Environments in the Era of Industry 4.0***. 2018. Acessado em: 14/11/2019.

BARBOZA, Rafael Menezes. ***Repositório do protótipo da arquitetura de monitoramento***. 2019. Disponível em: https://github.com/rmmenezes/prototipo-arq-mononitoramento, Acessado em: 26/11/2019.

BISHOP, Robert H. ***Mechatronic Systems, Sensors, and Actuators***. [S.l.]: CRC Press, 2007.

BOLTON, W. **Counters**. In: *Programmable Logic Controllers*. [S.l.]: Elsevier, 2009. p. 239–259.

BOYER, Stuart A. ***SCADA: Supervisory Control and Data Acquisition***. 3. ed. [S.l.]: ISA-The Instrumentation, Systems, and Automation Society, 2004, 2004. ISBN 9781556178771.

BRANQUINHO, Marcelo Ayres. ***Segurança de Automação: Industrial e Scada***. 1. ed. [S.l.]: Elsevier, 2014. ISBN 8535277331.

CERT.BR. ***Práticas de Segurança para Administradores de Redes Internet***. 2003. Disponível em: https://www.cert.br/docs/seg-adm-redes/, Acessado em: 22/04/2020.

CERT.BR. ***Cartilha de Segurança para Internet***. 2017. Disponível em: https://cartilha.cert.br, Acessado em: 20/10/2019.

CHEREPANOV, Anton; LIPOVSKY, Robert. ***Industroyer: Biggest threat to industrial control systems since Stuxnet***. 2017. Acessado em: 21/09/2019.

COLBERT, Edward. ***Cyber-security of SCADA and other industrial control systems***. 1st. ed. Switzerland: Springer, 2016. ISBN 9783319321233.

Cruz, T.; Barrigas, J.; Proença, J.; Graziano, A.; Panzieri, S.; Lev, L.; Simões, P. **Improving network security monitoring for industrial control systems**. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. Ottawa, Canada: Institute of Electrical and Electronics Engineers, 2015. v. 1, p. 878–881.

DEBAR, Hervé; VIINIKKA, Jouni. **Intrusion Detection: Introduction to Intrusion Detection and Security Information Management**. In: *Foundations of Security Analysis and Design III*. [S.l.]: Springer Berlin Heidelberg, 2005. p. 207–236.

ELASTIC. ***A sua janela para o Elastic Stack***. 2019. Disponível em: https://www.elastic.co/guide/en/kibana/7.4/introduction.html, Acessado em: 21/10/2019.

ELASTIC. ***Elasticsearch Reference***. 2019. Disponível em: https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html, Acessado em: 21/10/2019.

ELASTIC. ***Filebeat Overview***. 2019. Disponível em: https://www.elastic.co/guide/en/beats/filebeat/master/filebeat-overview.html, Acessado em: 19/02/2020.

ELASTIC. ***Logstash Introduction***. 2019. Disponível em: https://www.elastic.co/guide/en/logstash/current/introduction.html, Acessado em: 19/02/2020.

ELASTIC. ***Tune for indexing speed: Elasticsearch Reference [7.4]***. 2019. Acessado em: 24/11/2019.

Fan, X.; Fan, K.; Wang, Y.; Zhou, R. **Overview of cyber-security of industrial control system**. In: *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. Shanghai, China: Institute of Electrical and Electronics Engineers, 2015. p. 1–7.

FARRAPOSO, S.; OWEZARSKI, P.; MONTEIRO, E. **A Multi-Scale Tomographic Algorithm for Detecting and Classifying Traffic Anomalies**. In: *2007 IEEE International Conference on Communications*. [S.l.]: IEEE, 2007.

GOYAL, Piyush; GOYAL, Anurag. **Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark**. In: *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*. [S.l.]: IEEE, 2017.

GRAVES, Kimberly. ***CEH Certified Ethical Hacker Study Guide***. 3. ed. Switzerland: Sybex, 2016. ISBN 0470525207.

GROUP, Carnegie Mellon Database. ***Database of Databases: Elasticsearch***. 2019. Disponível em: https://dbdb.io/db/elasticsearch, Acessado em: 12/09/2020.

HADZIOSMANOVIC, Dina; BOLZONI, Damiano; ETALLE, Sandro; HARTEL, Pieter. **Challenges and opportunities in securing industrial control systems**. In: *2012 Complexity in Engineering (COMPENG). Proceedings*. [S.l.]: IEEE, 2012.

Han, Y.; Ko, N.; Kim, M.; Park, H. **Vulnerability of small networks for the TTL expiry DDoS attack**. In: *2012 Computing, Communications and Applications Conference*. Hong Kong, China: Institute of Electrical and Electronics Engineers, 2012. v. 1, p. 147–149.

JUNIOR, Christiano Vasconcelos das Chagas Raphaela Galhardo Fernandes Antônio Pereira de Araújo. **Uma Rápida Análise Sobre Automação Industrial**. *Redes para Automação Industrial*, 2003.

KASPERSKY. ***Kaspersky Industrial CyberSecurity: visão geral das soluções em 2018***. 2018. Disponível em: https://www.aquarius.com.br/wp-content/uploads/2018/08/KICS_overview.pdf, Acessado em: 28/10/2019.

Lakhoua, M. N. **Cyber Security of SCADA Network in Thermal Power Plants**. In: *2018 International Conference on Smart Communications and Networking (SmartNets)*. [S.l.: s.n.], 2018. p. 1–4.

MARQUARDT, Alex. ***Using parallel Logstash pipelines to improve persistent queue throughput***. 2019. Disponível em: https://www.elastic.co/pt/blog/using-parallel-logstash-pipelines-to-improve-persistent-queue-performance, Acessado em: 10/10/2020.

MICRO, Trend. ***Security in the Era of Industry 4.0: Dealing With Threats to Smart Manufacturing Environments***. 2019.

MONGODB. ***Welcome to the MongoDB Docs***. 2019. Disponível em: https://docs.mongodb.com/, Acessado em: 14/11/2019.

MONGODB, Inc. ***Write Operation Performance***. 2019. Acessado em: 24/11/2019.

NCS, NATIONAL COMMUNICATIONS SYSTEM. ***Supervisory Control and Data Acquisition (SCADA) Systems***. [S.l.]: Technical Information Bulletin 04-1, 2004.

ORGANIZATION, Inc. Modbus. ***About The Modbus Organization***. 2005. Disponível em: https://modbus.org/faq.php, Acessado em: 23/11/2020.

OSBORNE, Charlie. ***Industroyer: An in-depth look at the culprit behind Ukraine's power grid blackout***. 2018. Acessado em: 26/09/2019.

OSSEC. ***OSSEC Documentation***. 2018. Disponível em: https://www.ossec.net/docs/manual/index.html, Acessado em: 13/11/2019.

SCHäFFER, Leonardo. ***Uma infraestrutura baseada em certificados digitais para efetuar autenticação de cliente***. 2016. Disponível em: https://www.linux.ime.usp.br/ schaffer/-mac0499/monografia.pdf, Acessado em: 20/04/2020.

TANENBAUM, Andrew S. ***Computer Networks (5th Edition)***. [S.l.]: Pearson, 2010. ISBN 0132126958.

TI Safe. ***Soluções de Governança e Monitoramento***. 2007. Disponível em: https://tisafe.com/index.php/pt-br/solucoes/governanca-e-monitoramento, Acessado em: 24/11/2020.

TSHARK. ***Tshark - The Wireshark Network Analyzer***. 2017. Disponível em: https://www.wireshark.org/docs/man-pages/tshark.html, Acessado em: 10/10/2019.

VOKOROKOS, Liberios; UCHNAR, Matus; LESCISIN, Lubor. **Performance optimization of applications based on non-relational databases**. In: *2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)*. [S.l.]: IEEE, 2016.

WAAGSNES., Henrik; ULLTVEIT-MOE., Nils. **Intrusion Detection System Test Framework for SCADA Systems**. In: INSTICC. *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. New York, NY, USA: SciTePress, 2018. v. 1, p. 275–285. ISBN 978-989-758-282-0.

Wang, F.; Qi, W.; Qian, T. **A Dynamic Cybersecurity Protection Method based on Software-defined Networking for Industrial Control Systems**. In: *2019 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2019. p. 1831–1834.

WAZUH. ***Documentation Getting started***. 2019. Disponível em: https://documentation.wazuh.com, Acessado em: 01/05/2020.

WIRESHARK. ***Tshark - The Wireshark Network Analyzer***. 2017. Disponível em: https://www.wireshark.org/docs/wsug_html_chunked/, Acessado em: 05/10/2020.

ZETTER, Kim. ***Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon***. [S.l.]: Broadway Books, 2015. ISBN 9780770436193.