https://eventos.utfpr.edu.br//sicite/sicite2020

Segurança da informação na coleta de dados científicos

Information security in the collection of scientific data

Elimar Sanches Kauffmann
elimarkauffmann@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Winderson Eugenio dos Santos
winderson@professores.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

RESUMO

O presente trabalho apresenta as técnicas utilizadas para a garantia de segurança de informação dos dados gerados pelas estações solarimétricas e fotovoltaicas instaladas em seis Campi da UTFPR, em que o Laboratório de Energia Solar (LABENS) é o responsável. Esses dados são acessados através de um sistema web, desenvolvido para monitoramento e gerenciamento das estações, no qual professores e pesquisadores possuem acesso. E para consultas aos dados de irradiação solar e metereológicos, foi desenvolvida uma Interface de Programação de Aplicação (API), para permitir acesso em aplicações externas, tais como planilhas eletrônicas, que também faz uso de medidas de segurança. Todas essas medidas visam a integridade do Banco de Dados, do Servidor e das informações que eles possuem.

PALAVRAS-CHAVE: Proteção de dados. Segurança de Sistemas. Interface de Programação de Aplicativos. Armazenamento de dados.

ABSTRACT

The present work presents the techniques used to guarantee information security of the data generated by the solarimetric and photovoltaic stations installed in six Campuses of UTFPR, in which the Solar Energy Laboratory (LABENS) is responsible. These data are accessed through a web system, developed for monitoring and managing stations, to which teachers and researchers have access. And for queries on solar radiation and weather data, an Application Programming Interface (API), to allow access to external applications, such as spreadsheets, was developed, which also makes use of security measures. All these measures aim at the integrity of the Database, the Server and the information they have.

KEYWORDS: Data Protection. Cybersecurity. Application Programming Interface. Data warehousing.

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

INTRODUÇÃO

A lei no 9.991/2000, determina que as empresas concessionárias, permissionárias e autorizadas do setor de energia elétrica apliquem anualmente um percentual de sua Receita Operacional Liquida (ROL) em pesquisa e desenvolvimento (P&D) (BRASIL, 2000). Sendo a UTFPR uma das contempladas com essa modalidade de investimento, através do financiamento da COPEL-Distribuição (COPEL, 2017), instalou Estações Solarimétricas e Fotovoltaicas em seis Campi. As quais geram um volume de aquisição de dados próximo a 24 milhões de registros mensais.

E é neste contexto que este trabalho se insere, pois para o monitoramento e gerenciamento dessas estações, foi desenvolvido um sistema web, que possibilita o acompanhamento da geração de energia, enquanto compara com informações climatológicas. E toda essa gestão faz uso de medidas de segurança, as quais serão exploradas neste trabalho.

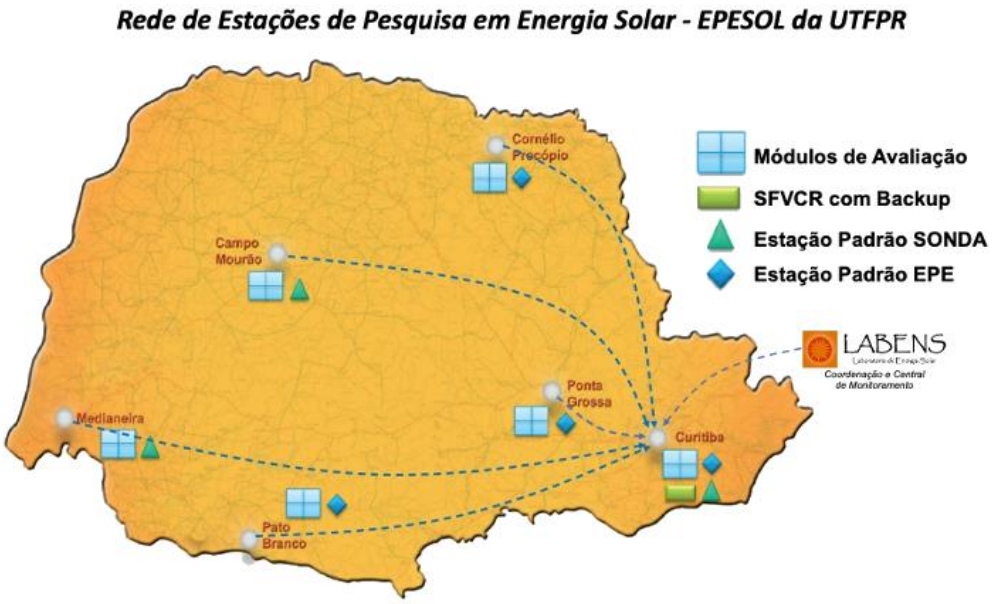
METODOLOGIA

Os princípios metodológicos visam empregar técnicas e mecanismos de segurança, para evitar que pessoas com má intenção, ou até mesmo acidentalmente, pudessem, por exemplo, acessar dados de usuários e parceiros, apagar ou modificar os dados coletados, danificar equipamentos através do envio de comandos pela *Application Programming Interface* (API), fazer inúmeras requisições ao servidor, até o ponto de não suportar a sobrecarga e ficar fora do ar; dentre outras possíveis situações de ataques cibernéticos.

DISPOSIÇÃO FÍSICA E LÓGICA

É um sistema web para comunicação, armazenamento e recuperação de dados oriundos de seis plantas fotovoltaicas, geograficamente distribuídas em diferentes cidades do estado do Paraná, conforme a Figura 1. Também prove meios para a gestão de recursos e ativos do sistema, o qual é composto por unidades de geração de energia e ao menos uma estação metereológica em cada planta.

Figura 1 – Mapa EPESOLs

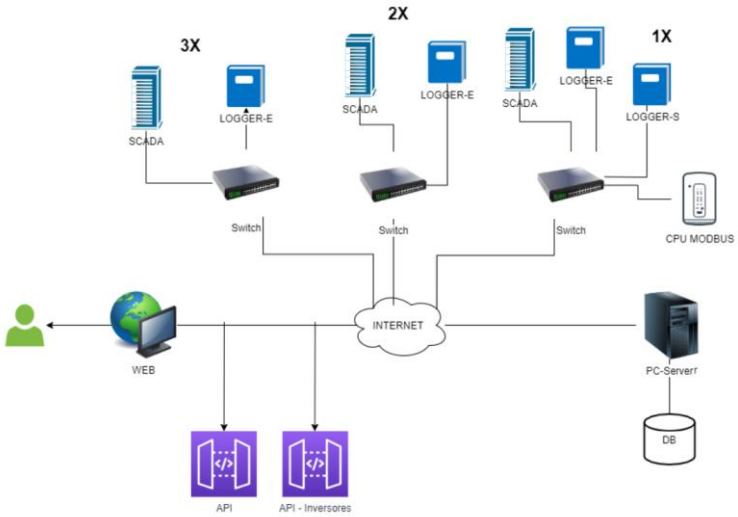


Fonte: LABENS (2019).

Os dados coletados são produzidos de forma síncrona, a uma taxa de 1 payload por minuto, e são armazenados em banco de dados (BD).

O diagrama físico do sistema, ilustrado na Figura 2 compreende: A) um computador servidor instalado no campus centro-Curitiba da UTFPR para hospedagem das aplicações; B) seis plantas micro-geradoras de energia elétrica fotovoltaica, nas cidades de Curitiba, Ponta Grossa, Pato Branco, Medianeira, Campo Mourão e Cornélio Procópio; cada planta contendo dois tipos de CPU remota: i) computador de automação (SCADA) programado em linguagem IoT Node-Red e ii) Datalogger (CR1000x) programado em linguagem proprietária Campbell Scientific. C) um Raspberry Pi 3, programado em python, instalado no campus neoville-Curitiba da UTFPR. Todas as CPUs estão conectadas a internet e atualmente enviando dados para um servidor FTP.

Figura 2 – Diagrama Físico



Fonte: Autoria própria (2020).

CYBERSECURITY E INFORMATION SECURITY

"Information Security" é um termo maior, que preocupa-se com a confidencialidade, integridade e disponibilidade dos dados (OLCOTT, 2019). O termo "Cybersecurity" ou "Cyberspace security" é referido pela ISO/IEC 27032 de 2012, como proteção de privacidade, integridade, e acessibilidade de informações no ciberespaço. Sendo, o ciberespaço, reconhecido como uma interação de pessoas, software e serviços tecnológicos mundiais (ISO/IEC 27032, 2012). Ou seja, o primeiro engloba tanto o que é digital, quanto o que não é, já o segundo, que é o nosso escopo, foca somente no digital. É importante identificar onde estão os dados mais críticos, que devem ser protegidos, e quais os riscos de serem comprometidos ou atacados, assim como a tecnologia que será usada para a proteção deles (OLCOTT, 2019).

Neste contexto, os dados de interesse são: dados gerados pelas estações - explicados no capítulo de Disposição Física e Lógica; registros de log; aqueles inseridos por usuários a título de gestão (cadastros de manutenção, usuários, dispositivos, grandezas, variáveis, contatos, entre outros); chaves de acesso para consultas via API; cadastro de comandos de fator de potência para os inversores das estações; e permissões de usuários;

Sistemas que não possuem quaisquer medidas de segurança, são facilmente comprometidos quando atacados. Um sistema web que não tem protocolo de segurança, ou seja, criptografia, está vulnerável a interceptação de dados entre cliente e servidor (ALVES, 2014). Um sistema que não possui autenticação para logar, permite que qualquer usuário tenha acesso as informações dele. Além disso, se existem níveis diferentes de permissões que cada um deve ter, é necessário que sejam criados perfis de usuários diferentes. Acesso direto ao banco de dados para coleta de dados, pode também ser perigoso, pois quem acessa diretamente, tem acesso às tabelas e pode fazer praticamente qualquer coisa com os dados existentes. Também é interessante ter um mediador entre o BD e o cliente para a implementação das próprias regras de negócio. Neste projeto, um dos acessos mais críticos, é o de envio de comandos para os inversores, onde a configuração de um fator errado de potência neles, pode causar problemas. E essa configuração errada pode acontecer por ao menos três situações: A) Envio acidental de valores incorretos; B) Valor incorreto inserido propositalmente; C) Falha de sistema que gerou valores errados; Erros humanos podem acontecer, e visando a integridade dos dados é importante exigir confirmação do usuário, quanto a alguma modificação importante (Guerra, 2020). Se possível e necessário, até mesmo o uso de um segundo fator de confirmação. Em questão de disponibilidade de dados, e também de segurança, é importante que a aplicação e o banco de dados estejam em servidores distintos. Em especial porque caso ocorra de o servidor de aplicação ficar indisponível, o de dados não necessariamente ficará.

Tendo a descrição dos dados de interesse e dos possíveis riscos de segurança, pode se descrever as tecnologias/abordagens idealizadas visando a sua proteção. São elas: HTTPS (*Hyper Text Transfer Protocol Secure*); confirmação de comandos de *create, update* e *delete*; máscaras nos campos (validação de C.P.F para o login e inserção de dados); quatro níveis de usuários: Admin, Manager, Guest e User; virtualização de servidores (apesar de ser um único servidor físico, o servidor de aplicação e de dados estão em máquinas virtuais diferentes); envio de email solicitando confirmação para mudanças específicas; Interface de programação de

aplicação; NGINX como web server (gateway de API); chaves de acesso para as APIs (JSON Web Token). Este trabalho dará maior ênfase na parte da API, por ser mais complexa e de maior importância para o modelo de negócio do LABENS.

INTERFACE DE PROGRAMAÇÃO DE APLICAÇÕES

Neste trabalho, toma-se como base a definição apresentada por Rodrigues (2017), onde dois ou mais sistemas de informação, independendo da plataforma e do tipo de acesso, faz uso de uma estrutura formal de regras e protocolos para ter a interoperabilidade de conjunto de dados entre eles. Ou seja, API é a estrutura que possibilita a comunicação de diferentes sistemas.

Existem tipos de APIs que seguem o estilo de arquiterura REST (Representational State Transfer) e as que são baseadas em SOAP (Simple Object Access Protocol). A primeira é mais indicada para acesso a dados, quanto que a segunda é mais indicada para executar operações por meio de um conjunto mais padronizado de mensagens. REST aproveita a exposição da URL (Uniform Resource Locator) e segue padrões como HTTP, JSON (JavaScript Object Notation) e XML (Extensible Markup Language) (Durães). Essas padronizações foram importantes na escolha do padrão REST.

Trabalhando com REST, é possível usar o padrão JSON Web Tokens (JWT), o qual define uma maneira compacta e independente para transmitir informações com segurança entre aplicações, como um objeto JSON. As informações são assinadas digitalmente (o que possibilita verificação e confiança). Os JWTs podem ser criptografados, mas o foco é nos tokens (JWT).

RESULTADOS E DISCUSSÕES

O sistema web faz uso de *Hyper Text Transfer Protocol Secure*, garantindo um tráfego de dados mais seguro na rede.

Para ações de modificação dos dados (tanto criação, quanto atualização e exclusão), é necessária confirmação pelo usuário.

Nos campos de preenchimento, são usadas máscaras, como no caso de inserção do Cadastro de Pessoa Física (para realizar login), inserção de valores para configurar o fator de potência dos inversores, e outras telas de gestão dos dados.

Existem quatro níveis de usuários, com diferentes níveis de permissões: Admin (maior nível de permissão), Manager, Guest e User (menor nível de permissão). Pode-se verificar o número de permissões que cada perfil possui na figura 3. Onde Admin possui 61, sendo capaz de realizar todas as ações de CRUD (*Create, Read, Update, Delete*) no sistema. Enquanto Guest só pode ler (*Read)*.

Figura 3 – Tabela de permissões de usuários

Perfil	Permissões
Admin	61
Manager	41
Guest	16
User	9

Fonte: Autoria própria (2020).

Foi virtualizado e encapsulado em máquinas virtuais diferentes, a aplicação e a base de dados. Assim, caso ocorra problema com o servidor de aplicação, ainda será possível ter acesso ao banco de dados.

Quando uma mudança de fator de potência é feita no sistema, automaticamente um e-mail é disparado para o responsável cadastrado, para que confirme, ou não, se realmente pode ser feita esta mudança. Assim, cadastrando um responsável técnico para essa validação, é possível garantir que valores errados ou indesejados não tenham sido configurados.

Dentre as tecnologias utilizadas, uma que merece destaque neste projeto é a de API. Para fazer uso dela, neste contexto (dado o modelo de negócio), necessita a criação de *tokens*. As quais podem ser criadas através de uma interface de criação de chaves de acesso no próprio sistema web. Para cada requisição são guardados os seguintes atributos de log: o "timestamp" da requisição, qual foi a "chave" que requisitou, qual o "user-agent" do usuário, qual o "ip" do requerente, e qual o "status" da requisição (se obteve sucesso, se o limite de consultas por hora foi alcançado ou se a chave de acesso está expirada).

A chave de acesso é uma string criptografada com 128 caracteres, neste caso. No seu uso, são registradas informações de quem está usando a API (*IP* e *User Agent*). Limita a permissão, onde cada chave de acesso tem um limite de requisições que pode ser feita (para não sobrecarregar o servidor com loop de consultas desnecessárias), limite o qual é configurado por quem gera essa chave de acesso. Na própria chave de acesso, pode-se vincular o formato de dados que deseja de retorno (Comma-separated values ou JSON). Essa escolha pode ser muito útil, pois em algumas aplicações, normalmente a que são feitas por desenvolvedores, será usado o JSON. Já em casos de uso de estudantes/pesquisadores, o CSV (Comma-separated values) é mais prático, podendo, inclusive, recuperar os dados diretamente nas planinhas eletrônicas, somente usando uma formula da planilha e passando a URL (já com a chave de acesso) como parametro (Figura 4). Esse controle mais eficiente e seguro, quanto ao uso da API, é facilidado pelo Gateway de API NGINX (NGINX, c2020).

Figura 4 – Exemplo de consulta de dados através da API, usando a planilha do Google

3	https://dados.labens.ct.utfpr.edu.br/api/My45NzIuMTY5LTk4IiwicHJ2IjoiNTg			
4				
5	id_variavel	timestamp	valor	flag
6	glo_std-so01CM	8/26/2020 14:30:00	1.444717	0
7	glo_min-so01CM	8/26/2020 14:30:00	718.8898	0
8	glo_max-so01CM	8/26/2020 14:30:00	723.5204	0
9	dif_avg-so01CM	8/26/2020 14:30:00	95.9282	0
10	dif_std-so01CM	8/26/2020 14:30:00	0.0670754	0
11	dif_min-so01CM	8/26/2020 14:30:00	95.83139	0
12	dif_max-so01CM	8/26/2020 14:30:00	96.08446	0
13	dir_avg-so01CM	8/26/2020 14:30:00	780.837	0
14	dir_std-so01CM	8/26/2020 14:30:00	1.441931	0
15	dir_min-so01CM	8/26/2020 14:30:00	777.9679	0
16	dir_max-so01CM	8/26/2020 14:30:00	782.3997	0

Fonte: Autoria própria (2020).

CONCLUSÕES

As medidas de segurança adotadas, ainda estão em fase de implementação, devendo passar por inúmeros testes nos próximos meses. Espera-se que essas técnicas e metodologias descritas no artigo possam trazer maior confiança quanto a proteção de disponibilidade do servidor, integridade dos equipamentos e a proteção dos dados científicos coletados.

AGRADECIMENTOS

Os autores agradecem a UTFPR pelo apoio e infraestrutura disponibilizada para o desenvolvimento desta pesquisa e a COPEL-Distribuição pelo apoio e financiamento dos recursos para realização deste projeto de P&D "ANEEL PD 2866-0464/2017 - Metodologia Para Análise, Monitoramento e Gerenciamento da GD por Fontes Incentivadas".

REFERÊNCIAS

ALVES, P. **O que é HTTPS e como ele pode proteger a sua navegação na internet.** 2014. Disponível em: <https://www.techtudo.com.br/noticias/noticia/2014/02/o-que-e-https-e-como-ele-pode-proteger-sua-navegacao-na-internet.html>. Acesso em: 28 de ago. 2020.

BRASIL. LEI No 9.991, DE 24 DE JULHO DE 2000. Dispõe sobre realização de investimentos em pesquisa e desenvolvimento e em eficiência energética por parte das empresas concessionárias, permissionárias e autorizadas do setor de energia elétrica, e dá outras providências. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l9991.htm#:~:text=LEI%20No%209.991%2C%20DE%2024%20DE%20JULHO%20DE%202000.&text=Disp%C3%B5e%20sobre%20realiza%C3%A7%C3%A3o%20de%20investimentos,el%C3%A9trica%2C%20e%20d%C3%A1%20outras%20provid%C3%AAncias>. Acesso em: 06 de out. 2020.

COPEL. 2017. RESULTADO PROJETOS SELECIONADOS – CHAMADA PÚBLICA VPDE 001/2017. [S.l.], 2017.

DURÃES, G. REST X SOAP: QUAL O MELHOR TIPO DE INTEGRAÇÃO?. Disponível em: <https://blog.tecnospeed.com.br/rest-x-soap/#:~:text=A%20API%20REST%20n%C3%A3o%20possui,baseada%20em%20HTTP%20e%20XML%3B&text=REST%20%C3%A9%20compat%C3%ADvel%20com%20JavaScript%20e%20tamb%C3%A9m%20pode%20ser%20implementado%20facilmente>. Acesso em: 31 de ago. 2020.

GUERRA, B. O que é integridade de dados e por que ela é importante. 2020. Disponível em: <https://blog.in1.com.br/o-que-%C3%A9-integridade-de-dados-e-por-que-ela-%C3%A9-importante>. Acesso em: 31 de ago. 2020.

INTERNATIONAL STANDARD ORGANIZATION. ISO/IEC 27032 - Information technology - Security Techniques - Guidelines for cybersecurity. 2012.

JWT. Introduction to JSON Web Tokens. Disponível em: <https://jwt.io/introduction/>. Acesso em: 31 de ago. 2020.

LABENS. Projeto P&D ANEEL/COPEL Distribuição – PD 2866-0464/2017 – Metodologia para Análise, Monitoramento e Gerenciamento da Geração Distribuída por Fontes Incentivadas. 2019. Disponível em: <https://labens.ct.utfpr.edu.br/projetos/projeto-pd-aneel-copel-distribuicao-pd-2866-0464-2017-metodologia-para-analise-monitoramento-e-gerenciamento-da-geracao-distribuida-por-fontes-incentivadas/>. Acesso em: 22 de ago. 2020.

NGINX. c2020. Ensinando a configurar o NGINX (API Gateway). Disponível em: <https://docs.nginx.com/nginx/admin-guide/web-server/app-gateway-uwsgi-django/>. Acesso em: 31 de ago. 2020.

OLCOTT, J. **Cybersecurity Vs. Information Security: Is There A Difference?**. 2019. Disponível em: <https://www.bitsight.com/blog/cybersecurity-vs-information-security>. Acesso em: 22 de ago. 2020.

RODRIGUES, F. A. Coleta de dados em redes sociais: privacidade de dados pessoais no acesso via Aplplication Programming Interface. 2017. Dissertação (Doutorado em Ciência da Informação) - Universidade Estadual Paulista, Marília, 2017.