# Análise de vulnerabilidades na implementação de dispositivo de Internet das Coisas como fechadura eletrônica

**Sander M. Migotto**[1], **Gleizer B. Voss**[1]

[1]Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Instituto Federal de Educação, Ciência e Tecnologia Farroupilha – (IFFar)
CEP 97.420-000 – São Vicente do Sul – RS – Brazil

sander.mm97@gmail.com, gleizer.voss@iffarroupilha.edu.br

***Abstract.*** *The popularization of IoT increased the number of Internet-connected devices. Thus, the security-related challenges have also increased. In this work, an environment was simulated with an electronic lock using Arduino and an RFID reader for access control. The purpose was to test the information security of the devices. As a contribution, it was possible to analyze which types of attacks the system is most susceptible to failures.*

***Resumo.*** *Com a popularização da IoT aumentam os desafios relacionados à segurança. Neste trabalho foi simulado um ambiente com uma fechadura eletrônica utilizando um microcontrolador Arduino e um leitor RFID para controle de acesso. Foram realizados testes de segurança que permitiram analisar a quais tipos de ataque o sistema é mais suscetível a falhas, demonstrando que a sua instalação padrão é insegura e a necessidade de usar rede segregada.*

## 1. Introdução

O termo *Internet of Things* (IoT), ou Internet das Coisas, surgiu em 1999, apresentado por Kevin Ashton do MIT. Segundo ele, Internet das Coisas trata de uma revolução tecnológica que tem por finalidade conectar os objetos de uso diário à Internet [Ashton et al. 2009]. A ampla abrangência que a IoT terá em diversos aspectos da vida comum da sociedade é um dos fatores fundamentais que contribuem para seu crescimento, desde questões simples do cotidiano até ambientes empresariais em várias áreas como o transporte inteligente de pessoas e bens, a logística, automação da produção industrial, a gestão de negócios e processos [Atzori et al. 2010].

No contexto de IoT, é crescente a probabilidade de novas aplicações, porém, novos desafios também surgem ao conectar à Internet objetos com limitações de memória, processamento, energia, comunicação [Loureiro et al. 2003] e de segurança [Sicari et al. 2015], [Atzori et al. 2010], [Weber 2010]. No que diz respeito à segurança da informação, a proteção da informação relacionada a diversos tipos de ameaças, de modo a garantir a continuidade do negócio, minimizando riscos, maximizando o retorno sobre o investimento e as oportunidades comerciais [ABNT 2005]. Ao discutir sobre segurança da informação, a norma NBR ISO/IEC 17799 refere-se à preservação da confidencialidade, da integridade e da disponibilidade da informação. Adicionalmente, outras propriedades, tais como autenticidade, responsabilidade, não repúdio e confiabilidade, podem também estar envolvidas [ABNT 2006].

Os ataques relacionados a segurança da informação são um sério transtorno para as organizações, pois essa área tem de lidar com a proteção dos dados imprescindíveis de uma empresa, tais como dados de clientes, relatórios, entre outros. Sendo assim proteger-se completamente é praticamente impossível, porém, a tomada de medidas corretas e um correto planejamento auxilia a precaver diversos problemas [Stefanini 2019]. Neste trabalho são abordados dois tipos de ataque: negação de serviço e interceptação de tráfego (sniffing).

De acordo com [CERT.BR 2017] o ataque de negação de serviço, do termo inglês *Denial of Service* (DoS), é realizado a partir de um computador com o objetivo de interromper algum serviço, computador ou rede conectada à Internet. Possui como variação a técnica de Negação de serviço distribuído (DDoS, da sigla em inglês para *Distributed Denial of Service*) que utiliza um conjunto de computadores para executar o ataque. Tais técnicas não visam coletar informações e nem invadir sistemas, mas sim consumir todos os recursos e tornar o alvo indisponível. No momento em que esse ataque ocorre, os usuários que utilizam os serviços ou recursos impactados pelo ataque ficam impedidos de usá-los.

Por sua vez a interceptação de tráfego (*Sniffing*), segundo [CERT.BR 2017], tem como objetivo examinar os dados que trafegam em uma rede de computadores, através de programas específicos chamados de *sniffers*. Tal técnica pode ser empregada de maneira legítima, em que administradores de redes utilizam para analisar desempenho, detectar problemas e monitorar atividades maliciosas relacionadas às redes ou computadores; ou maliciosa, em que os atacantes a utilizam para capturar informações de risco, como senhas, conteúdos confidenciais ou números de cartões de crédito que estejam trafegando em conexões inseguras, isto é, sem criptografia.

Mediante o exposto, este trabalho tem como objetivo principal implementar e realizar testes de vulnerabilidades em um dispositivo IoT simulando uma fechadura eletrônica. A justificativa para o desenvolvimento do trabalho foi de tentar identificar e testar vulnerabilidades no dispositivo de Iot e na sua comunicação, de forma a tornar o acesso aos laboratórios de informática do Centro de Informática Educativa e Tecnológica (CIET) no Instituto Federal Farroupilha – Campus São Vicente do Sul (IFFAR – SVS) mais simples e seguro, uma vez que só poderão acessar os laboratórios usuários cadastrados. Esse sistema é parte inicial de um projeto de uma sala de aula inteligente, a qual possuirá diversos sensores em seu interior, como de presença e de luminosidade, entre outros. A ideia é elaborar um ambiente autônomo e protegido.

## 2. Ambiente de testes

O ambiente (Figura 1) foi projetado de acordo com o que pretende-se implementar no Campus, com o dispositivo IoT (Arduino Mega 2560) comunicando-se via Ethernet com um servidor (computador *desktop*) para verificação dos dados de acesso, para posteriormente aplicar os testes de intrusão. Esse ambiente possibilita que o usuário, por meio de uma tag RFID, cartão ou chaveiro, com sua ID previamente cadastrada no banco de dados do sistema (MySQL [MySQL 2020]), possa acessar uma determinada fechadura. Com o sistema em funcionamento foram realizados alguns testes de intrusão utilizando um *notebook* e um computador *desktop* ambos com o sistema operacional Kali Linux [Kali 2020]. Essa estrutura foi montada para verificar se o ambiente estava resguardado

contra certos tipos de ameaças à segurança da informação. Contudo, intencionalmente, foram realizadas apenas configurações básicas padrão, sem configurações adicionais de segurança, como a utilização de bibliotecas de criptografia.

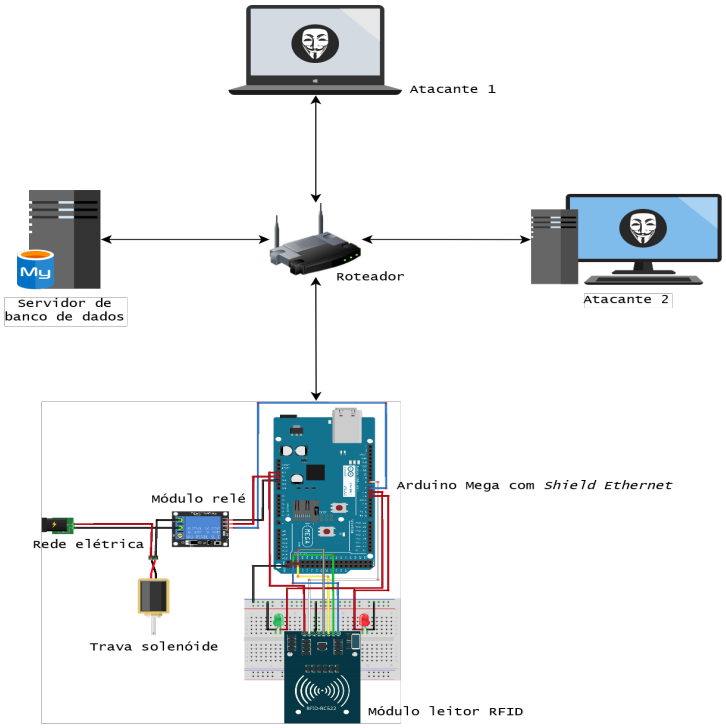


**Figura 1. Visão geral do ambiente**

## 2.1. Testes de intrusão

A realização dos testes de intrusão ocorreu de duas formas distintas: coleta de informações/análise de vulnerabilidades e testes de *stress*. Para a coleta e análise foram utilizadas três ferramentas, Nmap [Nmap 2020], tcpdump [tcpdump 2020] e Wireshark [Wireshark 2020]. Cada ferramenta desempenhou uma função específica: o Nmap realizou uma varredura na rede com o objetivo de descobrir os endereços IP do objeto IoT e do servidor; o tcpdump realizou a captura do tráfego de comunicação entre o dispositivo IoT e o servidor, na tentativa de obter algum dado sensível, como o ID da *tag* de acesso do usuário; já o Wireshark foi utilizado para analisar as informações obtidas pelo tcpdump. Os testes de *stress* ocorreram no dispositivo IoT e no servidor a fim de tornar o serviço indisponível, para tal, foi utilizada a ferramenta T50. Projetada para executar testes de *stress*, a T50 é uma ferramenta de injeção de pacotes multiprotocolo, capaz de realizar o ataque em até 15 protocolos diferentes (e.g., ICMP, TCP, UDP, GRE, IPSec, RSVP, RIP, EIGRP e OSPF). Possui alta performance, sendo capaz de atingir por volta de 1 milhão de pacotes por segundo. É capaz de simular ataques DoS e DDoS para, por exemplo, autenticar regras de firewall e políticas de Intrusion Detection System (IDS) e Intrusion Prevention System (IPS) [T50 2020].

## 3. Desenvolvimento

O ambiente foi montado e foram realizadas as devidas conexões entre os dispositivos conforme mostrado anteriormente na Figura 1. Para verificar a comunicação entre todos

os dispositivos e medir o tempo de resposta, foi disparado o comando PING para cada um dos dispositivos da rede. Após instalado e configurado o banco de dados no servidor, foi realizada a leitura das *tags* RFID, e caso já estivesse cadastrada no banco de dados um *led* verde seria ligado simulando assim a abertura da fechadura, e caso contrário um *led* vermelho seria acionado. Todos esses testes de conectividade e leitura obtiveram sucesso.

Após os testes iniciais, foi realizada a varredura na rede tanto no meio cabeado quanto *wireless*. Essa varredura conseguiu obter os endereços IP, bem como portas abertas não restringindo-se apenas nos alvos (servidor e Arduino) como também no próprio roteador, independente do meio de comunicação, demonstrando uma vulnerabilidade na rede, já que a mesma não está segmentada, ou seja, todas as redes fazem parte da mesma VLAN permitindo a descoberta dos *hosts*.

Posterior a varredura da rede, foram realizados os testes de negação de serviço os quais foram divididos em dois cenários principais, sendo eles: **(i) Ataques durante a leitura das *tags*:** nesse modo os ataques foram lançados ao Arduino e ao servidor de banco de dados durante a leitura das *tags*, ou seja, simulando a ocorrência de um ataque durante o acesso à fechadura; e **(ii) Ataques durante a inicialização do Arduino:** Nesse modo, os ataques foram lançados ao Arduino e ao servidor de banco de dados durante a inicialização do Arduino, ou seja, no momento em que o dispositivo tenta conectar-se ao banco de dados e carregar suas configurações.

Dentro desses dois cenários principais, foram criados diversos outros cenários com diferentes opções de uso da ferramenta, além da opção de inundação (*flood)* que está presente em todos os cenários, como modo estendido de performance e indicação de uma porta específica. Além dessas variações, foram realizados testes utilizando apenas um dos atacantes bem como a alteração do estado da proteção do roteador. De todas as variações de cenário, apenas duas obtiveram sucesso, sendo durante a leitura das *tags* em que o alvo era o Arduino e um dos atacantes utilizava a opção de performance estendida. Durante esse ataque o Arduino ficou incapaz de realizar novas leituras bem como consultas no banco de dados, consequentemente foi necessário, além de interromper o ataque, reiniciá-lo. Todos os ataques direcionados ao servidor de banco de dados não obtiveram sucesso.

Na Tabela 1 são elencados os cenários, onde é possível visualizar o alvo, o comando utilizado para o ataque, o número de atacantes, o estado da proteção do roteador e também a efetividade do ataque, isto é, se o mesmo indisponibilizou o serviço ou não.

**Tabela 1. Cenários de ataque durante a leitura das *tags***

| Cenários | Alvo | Nº atacantes | Comando | Proteção roteador | Efetividade |
|---|---|---|---|---|---|
| Cenário 1 | Servidor | 1 | –dport 3306 | Ativada | Não |
| Cenário 2 | Servidor | 1 | –dport 3306 | Desativada | Não |
| Cenário 3 | Servidor | 1 | –turbo –dport 3306 | Ativada | Não |
| Cenário 4 | Servidor | 1 | –turbo –dport 3306 | Desativada | Não |
| Cenário 5 | Servidor | 2 | A1: –turbo –dport 3306<br>A2: –dport 3306 | Ativada | Não[1] |
| Cenário 6 | Servidor | 2 | A1: –turbo –dport 3306<br>A2: –dport 3306 | Desativada | Não |
| Cenário 7 | Arduino | 1 | t50 –flood | Ativada | Não |
| Cenário 8 | Arduino | 1 | t50 –flood | Desativada | Não |
| Cenário 9 | Arduino | 1 | –turbo | Ativada | Não |
| Cenário 10 | Arduino | 1 | –turbo | Desativada | Não |
| Cenário 11 | Arduino | 2 | A1: –turbo<br>A2: –flood | Ativada | Sim |
| Cenário 12 | Arduino | 2 | A1: –turbo<br>A2: –flood | Desativada | Sim |

[1]No **Cenário 5** apesar do serviço não ficar indisponível, houve um *delay* em torno de 10 segundos para efetivar a leitura da *tag* e ativação da relé.

Fonte: Elaborado pelos autores.

Por fim, foi realizada a captura de pacotes durante uma tentativa de acesso a fechadura, ou seja, no momento em que uma *tag* RFID é lida. Após essa captura, o arquivo gerado foi analisado e verificado que o Arduino por si só não implementa nenhum tipo de criptografia, possibilitando ver as informações em texto puro (embora, vale ressaltar, que podem ser utilizadas bibliotecas que permitem a encriptação). Sendo assim, um invasor de posse dessas informações poderia facilmente clonar o ID da *tag* ou então caso a captura fosse realizada no momento em que o Arduino realiza a conexão com a base de dados seria possível obter o nome de usuário e senha de um usuário do próprio banco de dados.

## 4. Conclusão

Neste trabalho foi simulado um ambiente educacional com uma fechadura eletrônica utilizando a tecnologia RFID controlada por Arduino que realiza o registro dos acessos em um banco de dados MySQL, e posteriormente foram aplicados testes à segurança da informação desse sistema, sendo eles, testes de *stress* (negação de serviço) e captura/análise de tráfego utilizando as ferramentas t50, tcpdump e Wireshark.

A captura e análise dos pacotes demonstrou que por padrão, o Arduino não implementa criptografia, dessa maneira, os dados enviados são vistos em texto puro, o que pode ocasionar, nesse escopo, a clonagem de uma *tag* RFID e sua utilização para acesso indevido às fechaduras.

Visando corrigir a vulnerabilidade da confidencialidade dos dados é possível modificar o método de verificação de usuário para liberar o acesso utilizando a tecnologia

JSON, em que o Arduino envia uma solicitação para um *web service* e este responde com os dados solicitados

Com relação a descoberta dos dispositivos na rede, uma alternativa é a criação de uma VLAN somente para esses, com o intuito de implementar também outros recursos de segurança, como um *firewall*, protegendo a disponibilidade do sistema contra ataques de negação de serviço.

Uma limitação do trabalho foi a incapacidade de utilizar a ferramenta tcpdump diretamente no roteador, mesmo alterando o seu *firmware* diversas vezes para tentar solucionar esse problema. Sendo assim, a captura dos pacotes foi realizada diretamente no servidor de banco de dados. Ainda, o fato de não ter focado especificamente no Arduino, ou seja, realizando ataques específicos ao banco de dados, pode ser considerada outra limitação. Embora, justifique-se pelo ambiente simulado representar uma particularidade do campus, o que não se justificaria em um ambiente doméstico, por exemplo.

Como sugestão para trabalhos futuros tem-se a possibilidade de desenvolver um sistema web para o gerenciamento dos usuários, facilitando a interação com o sistema. Também é possível modificar o método de verificação de usuário para liberar o acesso utilizando a tecnologia JavaScript Object Notation (JSON), em que o Arduino envia uma solicitação para um web service e este responde com os dados solicitados. Desse modo, a vulnerabilidade dos dados em texto puro pode ser corrigida.

## Referências


ABNT (2005). *Tecnologia da informação – Técnicas de segurança – Código de prática para a gestão de segurança da informação*.

ABNT (2006). *ABNT. NBR ISO/IEC 17799 - Tecnologia da informação: código de prática para a gestão da segurança da informação*.

Ashton, K. et al. (2009). That 'internet of things' thing. *RFID journal*, 22(7):97–114.

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

CERT.BR (2017). Ataques na internet. Disponível em: https://cartilha.cert.br/ataques/. Acesso: Maio de 2019.

Kali (2020). Kali linux. Disponível em: ¡https://www.kali.org/¿. Acesso: setembro de 2020.

Loureiro, A., Nogueira, J. M., Ruiz, L., Ruiz, R., Aparecida, Mini, F., Nakamura, E., and Figueiredo, C. (2003). Redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores*, 21.

MySQL (2020). Mysql. Disponível em: ¡https://www.mysql.com/¿. Acesso: setembro de 2020.

Nmap (2020). Nmap. Disponível em: ¡https://nmap.org/¿. Acesso: setembro de 2020.

Sicari, S., Rizzardi, A., Grieco, L. A., and Coen-Porisini, A. (2015). Security, privacy and trust in internet of things: The road ahead. *Computer networks*, 76:146–164.

Stefanini (2019). Ataques à segurança da informação: conheça as principais ameaças. Disponível em: https://stefanini.com/pt-br/trends/artigos/ameacas-a-seguranca-da-informacao/. Acesso: Maio de 2019.

T50 (2020). T50. Disponível em: ¡https://gitlab.com/fredericopissarra/t50¿. Acesso: setembro de 2020.

tcpdump (2020). tcpdump. Disponível em: ¡https://tcpdump.org/¿. Acesso: setembro de 2020.

Weber, R. H. (2010). Internet of things–new security and privacy challenges. *Computer law & security review*, 26(1):23–30.

Wireshark (2020). Wireshark. Disponível em: ¡https://www.wireshark.org/¿. Acesso: setembro de 2018.