Universidade Estadual de Campinas

Faculdade de Tecnologia

Atividades dinâmicas para o ensino de segurança da informação em dispositivos para Internet das Coisas

Projeto de Iniciação Tecnológica submetido ao
Programa de Bolsas de Iniciação Tecnológica – PIBIT

Beneficiária: Larissa Benevides Vieira

Pesquisador Responsável: Prof. André Leon Sampaio Gradvohl, Dr.

Limeira, 27 de janeiro de 2020

Informações gerais sobre o projeto

- Título do projeto:

  Atividades dinâmicas para o ensino de segurança da informação em dispositivos para Internet das Coisas

- Título do projeto em inglês:

  Dynamic activities for the teaching information security in the Internet of Things devices

- Nome do pesquisador responsável:

  Prof. André Leon Sampaio Gradvohl, Dr.

- Nome da beneficiária:

  Larissa Benevides Vieira

- Instituição sede do projeto:

  Faculdade de Tecnologia da Universidade Estadual de Campinas

- Grupo de pesquisas ao qual o projeto se vincula:

  ***High Performance Intelligent Decision Systems* – HighPIDS**

  `https://highpids.ft.unicamp.br`

- Período de vigência:

  01/agosto/2019 a 06/fevereiro/2020

Resumo

Nos últimos tempos, com o aumento da velocidade de transmissão nas redes e com pequenos dispositivos com um razoável poder de processamento, a Internet das Coisas tem se tornado cada vez mais viável e despertando mais interesse do mercado. Entretanto, a corrida por lançamentos de novos produtos têm feito a indústria negligenciar as questões de segurança. Por outro lado, a academia não tem conseguido formar recursos humanos suficientes para suprir a demanda de profissionais capazes de lidar adequadamente com as questões de segurança da informação. Considerando esse cenário, este projeto de iniciação tecnológica, como resultado, elaborou e documentou algumas atividades para tornar mais interessante o ensino de segurança da informação em dispositivos para a Internet das Coisas. A ideia de documentar essas atividades é despertar o interesse dos estudantes ao mesmo tempo que os capacita para atuar em duas áreas cuja demanda de profissionais tem aumentado: segurança da informação e Internet das Coisas. Além disso, é importante ressaltar que este projeto foi encerrado na metade do seu desenvolvimento, em função de intercâmbio acadêmico no exterior.

Abstract

Recently, with the speedup in networks and with small devices with reasonable processing power, the Internet of Things has become increasingly feasible and arousing more interest in the market. However, the race for new products has made the industry neglect security issues. On the other hand, academy has not been able to train enough human resources to meet the demand of professionals capable of dealing adequately with information security issues. Considering this scenario, this technology initiation project, as a result, has elaborated and documented some activities to make the teaching of information security on IoT devices more interesting. The idea of documenting these activities is to arouse student's interest while enabling them to work in two areas where professional demand has increased: information security and internet of things. In addition, it is important to note that this project was closed in mid-development due to academic exchange abroad.

Sumário

1 Introdução

Na medida em que alguns dispositivos físicos começaram a ganhar algum poder computacional, logo vislumbrou-se a possibilidade de interconectá-los a uma rede de comunicação para que esses dispositivos pudessem não apenas enviar dados a um servidor central, mas também serem controlados à distância. A parir dessa possibilidade de conexão entre os diversos dispositivos, a quantidade de aplicações desenvolvidas tem crescido cada vez mais.

Hoje em dia, a chamada Internet das Coisas (*Internet of Things* – IoT) é um conceito bastante discutido tanto no mercado como na academia. De acordo com Minerva, Biru e Rotondi (2015), há várias definições para IoT, inclusive definições de vários institutos dedicados à especificação de padrões de comunicações. Entre esses institutos estão o *International Telecommunication Union* (ITU), o *Institute of Electrical and Electronics Engineers* (IEEE), a *Internet Engineering Task Force* (IETF) e o *World Wide Web Consortium* (W3C) entre outros. Contudo, em um esforço para chegar a uma definição simples e que contenha as visões dos principais institutos de padronização, o texto propõe uma definição que inclui conceitos como conexão à rede, programabilidade ou possibilidade de ser programado a partir de uma linguagem de programação, capacidade de interoperabilidade, capacidade de sensoriamento ou atuação, autoconfiguração e autoidentificação, entre outros.

Assim, adaptando a definição de Minerva, Biru e Rotondi (2015), consideramos que a IoT é uma rede que conecta "coisas" (objetos) unicamente identificáveis à Internet. Esses objetos têm capacidades de sensoriamento ou atuação e podem ser programados. Dadas essas características primordiais, as informações sobre os objetos podem ser coletadas e o estado desses objetos pode ser alterado de qualquer lugar, a qualquer momento, por qualquer outro objeto com as mesmas características primordiais. Em poucas palavras, pode-se afirmar que IoT é a interconexão via Internet de dispositivos computacionais incorporados em objetos do cotidiano, permitindo que eles enviem e recebam dados (RAWES, 2019).

Considerando essa definição, destaca-se que há um potencial significativo de ameaças à segurança e à privacidade na IoT. Afinal, com as capacidades de programação, conexão à Internet e sensoriamento e ativação, os dados processados e transmitidos pela rede podem ser interceptados, adulterados ou até negados, comprometendo os principais pilares da segurança: confiabilidade, integridade, disponibilidade (JOSE; VIJYALAKSHMI, 2018).

Entretanto, apesar da necessidade de se observar aspectos de segurança na IoT, percebe-se que o mercado está mais interessado no lançamento mais breve de novos dispositivos e

aplicações do que na qualidade desses equipamentos no que tange à segurança (DABBAGH; RAYES, 2017). Por outro lado, há falta de recursos humanos devidamente capacitados para o desenvolvimento de aplicações para IoT que considerem as questões de segurança desde o projeto.

Assim, tendo em vista a demanda do mercado por dispositivos mais seguros conectados à Internet e a necessidade da formação de recursos humanos para lidar com as questões de segurança computacional para dispositivos na IoT, este projeto propôs um estudo sobre dinâmicas para o ensino de segurança da informação em dispositivos na IoT. A ideia era desenvolver uma metodologia de ensino que incluiria práticas específicas para ilustrar e evidenciar problemas de segurança em dispositivos conectados à IoT. No entanto, por causa do encerramento precoce do projeto, em razão de um intercâmbio acadêmico no exterior, somente os experimentos práticos foram realizados, não havendo tempo hábil para estudar uma metodologia de ensino específica para este projeto. Outrossim, também se propôs realizar estimativas de prazos e custos para se implementar as práticas, porém não houve tempo hábil para a realização da mesma.

Os experimentos práticos propostos por esse projeto poderão ser incorporados posteriormente em disciplinas regulares de cursos de graduação na área de Informática (e. g. Ciência da Computação, Engenharia de Computação, Sistemas de Informação, Tecnologia em Análise e Desenvolvimento de Sistemas) ou em cursos de extensão específicos para complementação da formação profissional nos níveis técnico ou superior.

Por fim, a expectativa é que os resultados parciais desse projeto deem subsídios para a tomada de decisão sobre a incorporação das práticas implementadas. Com isso, docentes poderão incorporar essas práticas em disciplinas como Sistemas Operacionais, Redes de Computadores, Segurança Computacional ou outras disciplinas similares.

1.1 Objetivos

A pesquisa deste projeto visava investigar e propor uma metodologia de ensino e uma série de dinâmicas práticas para ensino dos conceitos de segurança da informação em dispositivos para IoT. No entanto, como resultado final, devido ao encerramento antecipado do projeto, somente as dinâmicas práticas foram realizadas.

Para atingir esse objetivo principal, os seguintes objetivos específicos foram alcançados:

- Fazer um levantamento dos principais conceitos de segurança da informação e IoT que são abordados nos currículos de referência da Sociedade Brasileira de Computa-

ção (SBC), da *Association for Computing Machinery* (ACM) e *Institute of Electrical and Electronics Engineers* (IEEE).

- Arquitetar práticas de ensino em laboratório com dispositivos móveis ou portáteis ligados à Internet.

2 Cronograma de atividades

O trabalho se iniciou com um levantamento bibliográfico sobre outros trabalhos similares, bem como a análise dos currículos de referência da SBC, ACM e IEEE, com vistas à identificação de habilidades e competências necessárias para as atividades. Em paralelo, ainda nesta primeira fase, que sucedeu-se entre agosto e novembro, foi realizada a preparação de um laboratório, com a instalação dos equipamentos e softwares necessários para testar as práticas levantadas a partir do levantamento bibliográfico.

Na segunda fase, que ocorreu de dezembro a janeiro, elaborou-se e refinou-se toda a documentação necessária para implementar e replicar as práticas contidas neste relatório.

Na terceira fase, que ocorreria a partir de fevereiro, apesar de não ter havido tempo hábil para a a realização da tarefa, foi proposto um estudo acerca de uma metodologia de ensino para que as práticas fossem aplicadas aos estudantes, além de um levantamento sobre custos e prazos para implementá-las.

3 Materiais e Métodos

Para a realização deste projeto, o primeiro passo realizado foi um levantamento bibliográfico em uma das principais bases da área: O IEEE Xplore Digital Library (IEEE, 2020). Nessa base de pesquisa foram inseridas palavras-chaves para que a busca por artigos com trabalhos semelhantes fossem encontrados. As palavras-chaves utilizadas foram: “Education”, “Hacking”, “IoT”, “Information Security” e “Attacks”. Embora outras bases de dados tenham sido consultadas, não houve artigos científicos suficientes que servissem como base para agregar na efetivação dos experimentos práticos, pois ao todo foram encontrados aproximadamente 44 artigos relacionados ao tema, mas somente 7 artigos foram utilizados para auxiliar neste projeto.

Além disso, vale ressaltar que, na maioria das bases de dados consultadas, notou-se uma falta de artigos científicos com conteúdos que informassem passo-a-passo como um experi-

mento prático deveria ser realizado. Em outras palavras, os experimentos práticos relacionados à segurança da informação em dispositivos de IoT constantes em outros artigos, em sua maioria, não dispunham de forma mais detalhada de instruções de configuração e comandos para que os experimentos fossem implementados sem que ocorresse algum problema durante a sua realização. Isso fez com que houvesse dificuldades de implementação e uma necessidade de se consultar outras fontes (e. g. sites, vídeos) para o teste de alguns experimentos.

Após o levantamento dos principais conceitos necessários para o ensino de segurança da informação na IoT, o próximo passo foi testar os experimentos práticos para ilustrar esses conceitos. Para este relatório, são usadas as palavras "prática" e "atividade" de forma indistinta.

Foram testadas e implementadas três atividades práticas que correlacionaram segurança da informação com dispositivos de IoT. Para que isso ocorresse, as atividades se dispuseram dos equipamentos e softwares apresentados na Tabela 3.1. Os experimentos são apresentados a seguir.

Tabela 3.1: Configurações gerais dos computadores utilizados.

Componente	**Máquina alvo**	**Máquina de ataque**
Sistema Operacional	Kali Linux 2019.1	Kali Linux
Kernel	x86_64 Linux 4.19.0-kali3-amd64	x86_64 Linux 4.19.0-kali5-amd64
Disco	10G / 291G (4%)	27G / 185G (15%)
CPU	Intel Core2 Duo E7600 @2x2.843GHz	Intel Core i5-7200U @4x3.1GHz
Endereço de rede	Controller (rev 10)	192.168.1.104
Memória	2916MiB / 3934MiB	RAM: 3053MiB / 3825MiB

Experimento 1 – Ataque de negação de serviço a sistema de IoT.
Equipamentos:

- Dois Computadores com Sistema Operacional Kali Linux. Um computador denominado PC (Kali atacante), e o outro como PC (Kali alvo).
- Roteador: wireless-g broadband router wrt54g.
- Arduino uno.
- Ethernet shield 2.

Software:

- ping (VITUX, 2020).

- Hping3 (KALI LINUX PENETRATION TESTING TOOLS, 2020b).

Experimento 2 – Ataque a Raspberry Pi usando credenciais padrão.
Equipamentos:

- Um Computador com Sistema Operacional Kali Linux.
- Raspberry Pi 3 Modelo B+.
- Roteador: wireless-g broadband router wrt54g.

Softwares:

- Rpi-hunter (BUSESCANFLY, 2020).
- Nmap (KALI LINUX PENETRATION TESTING TOOLS, 2020c).

Experimento 3 – Interceptação de dispositivo Bluetooth.
Equipamentos:

- Um Computador com Sistema Operacional Kali Linux.
- Celular iPhone 8 Plus.

Software:

- Bettercap (KALI LINUX PENETRATION TESTING TOOLS, 2020a).

É importante ressaltar que o sistema operacional Kali Linux serviu como base geral para todos experimentos pois, por ser uma distribuição Linux derivada do Debian, esta é projetada para análise digital forense e testes de penetração, contendo softwares previamente instalados (OFFENSIVE SECURITY, 2019). Sendo assim, é uma distribuição bastante adequada para identificar e explorar problemas de segurança computacional. Portanto, foi bastante útil no teste e desenvolvimento das práticas citadas neste artigo.

As placas Arduino uno e o Raspberry Pi modelo B+, por sua vez, são um ecossistema de *hardware* e *software open-source*. Em outras palavras, a primeira é uma placa microcontroladora que permite a inclusão de outros dispositivos (e. g. sensores, dispositivos de entrada e saída, placas adaptadoras para que o Arduíno possa se conectar a uma rede local ou internet – Ethernet Shield) e pode ser programável (ARDUINO, 2019). A segunda, por sua vez, é um computador de baixo custo, do tamanho de um cartão de crédito, geralmente com o sistema

operacional Linux, capaz de realizar atividades que um computador de mesa faria, desde navegar pela internet, até executar vários programas de uma vez (RASPBERRY PI FOUNDATION, 2020).

Normalmente, uma placa Arduino é composta por um controlador, algumas linhas de entrada e saída digital e analógica, além de uma interface serial ou USB, para interligar-se a um computador hospedeiro, que é usado para programá-la e interagir com a placa em tempo real. Enquanto o Raspberry Pi, além de conter características similares ao Arduino, se assemelha muito mais com um computador, sendo possível realizar experimentos mais complexos e elaborados. Essas características são bem interessantes para dispositivos na IoT.

A utilização do roteador também foi primordial porque, a partir do mesmo, criou-se uma rede local, com o propósito de criar um tráfego de rede segmentado para utilizar nos testes e exercícios deste projeto, evitando assim, acesso indevido a sistemas acadêmicos ou administrativos.

Por fim, utilizou-se um telefone celular com um protocolo de comunicação sem-fio (*wireless*) chamado *Bluetooth*. Visto que algumas aplicações IoT estão cada vez mais utilizando tecnologias de comunicação que habilitam diversas formas de interação entre objetos, o uso do *Bluetooth* nos chamados *smartphones* podem controlar e enviar dados a objetos IoT, fazendo com que este tipo de protocolo esteja vulnerável a interceptações de dados e ataques externos que podem comprometer tanto a integridade do aparelho, quanto do dispositivo IoT conectado ao mesmo.

No que se refere às práticas, para cada uma delas foram documentadas os conceitos e conhecimentos prévios necessários para sua realização, além de conter, de modo claro e preciso, todos os comandos que deveram ser executados para que o experimento possa ser concluído com êxito pelo estudante.

Por fim, as práticas desenvolvidas e testadas estão devidamente documentadas, de maneira que sua adoção, em qualquer disciplina, possa ser feita de forma simples e sem muito esforço por parte do docente ou até mesmo de um instrutor (auxiliar do docente). Também considerou-se a utilização de materiais de custo razoável, como é o caso do Arduino e do Raspberry Pi e a utilização de um Sistema Operacional de código aberto, devido ao uso de seus programas de forma gratuita.

4 Resultados

Foram desenvolvidas e documentadas três práticas com temas distintos, relacionadas a IoT. A documentação para cada uma dessas práticas (experimentos) foi divididas em três partes:

(i) Preparação prévia para a realização do experimento.

(ii) Realização do Experimento.

(iii) Estratégias de Mitigação.

4.1 Experimento 1 – Ataque de negação de serviço a sistema de IoT.

O ataque *Denial of Service* (DoS), tenta impedir ou restringir o uso normal da rede ou da administração da rede, com ou sem fio, de modo que a máquina alvo, receba solicitações supérfluas com o intuito de sobrecarregar a rede e impedir, por exemplo acesso a um *site* ou serviço web (RIBEIRO, 2018). Esse tipo de ataque compromete tanto as redes sem fio, quanto as redes IoT (KALITA; KAR, 2009).

Neste experimento, o objetivo principal é simular um ataque DoS a uma placa Arduino uno conectada a uma rede local. Para isso foi utilizado uma das mais poderosas ferramentas do Kali linux, o Hping3. Essa ferramenta é capaz de enviar arquivos entre um canal coberto, além de suportar os protocolos TCP, UDP, ICMP (KALI LINUX PENETRATION TESTING TOOLS, 2020b). O Hping3 é considerada uma importante ferramenta para criação de pacotes, além de conter muitos outros recursos úteis (WONDERHOWTO COMPANY, 2013).

Ademais, para complementação, foi realizado um segundo experimento utilizando o mesmo ataque em um projeto de Trabalho de Conclusão de Curso, atualmente sendo realizado na Faculdade de Tecnologia da UNICAMP como proposta de mestrado pelo aluno Vinícius Gustavo de Jesus F. Lemes . O projeto consiste em utilizar um microcontrolador Arduino e sensores para o monitoramento de uma plantação real por parte de um usuário (LEMES, 2018). Este experimento foi baseado no artigo de Cui et al. (2018).

(i) Preparação prévia para a realização do experimento

1. Mapeie os IP's dos dispositivos

    - Primeiro, deve se descobrir os endereços IP locais dos dois computadores, deixando claro que um será utilizado para realizar o ataque e o outro para ser o alvo. Esses computadores são denominados de PC (Kali atacante) e PC (Kali alvo), respectivamente.

Importante ressaltar que, o endereço IP local se refere ao endereço do computador dentro de uma rede privada (e. g. uma rede *wireless*). Para descobrir o IP de cada máquina, basta digitar no terminal respectivo o seguindo comando:

```
$ ifconfig
```

- Para descobrir o IP do roteador, é necessário que os computadores estejam conectados a rede do roteador. Essa conexão pode ser realizada via cabo ethernet ou via wifi. Agora, basta digitar o seguinte comando no terminal de qualquer um dos dois computadores:

  ```
  $ route
  ```

- Para utilizar o Arduino é necessário configurar corretamente o Ethernet Shield com um endereço IP válido de acordo com o endereço da rede local utilizada. Nesta prática, como o endereço IP do roteador é 192.168.1.1, então o endereço IP do Arduino poderia ser 192.168.1.110. Todos os endereços IP utilizados estão na Tabela 4.1.
- Para facilitar a configuração, foi fornecido adicionalmente um código para configurar o IP do Arduino, que pode ser acessado neste site: `https://www.filipeflop.com/blog/tutorial-ethernet-shield-w5100`.

Tabela 4.1: Endereços IP dos dispositivos utilizados

Nome do Equipamento	**Endereço IP**
PC (Kali atacante)	192.168.1.100
Roteador	192.168.1.1
PC (Kali alvo)	192.168.1.104
Arduino	192.168.1.110

2. Conecte um cabo de rede no roteador e na Ethernet Shield do Arduino.
3. Conecte o PC(Kali alvo) ao roteador.
4. Teste a conexão entre o PC(Kali alvo) e o Arduino. Use o comando ping no PC(Kali alvo) para testar a conexão. Digite no terminal:

   ```
   $ ping 192.168.1.110
   ```

5. Conecte o PC (Kali atacante) a mesma rede.

(ii) Realização do Experimento

1. Descubra qual porta está aberta para que possa ser feito o ataque. Para que isso possa ser feito, digite o comando a seguir no terminal do PC (Kali atacante). As informações sobre os parâmetros do comando `hping3` estão na Tabela 4.2.

```
$ hping3 --scan 0-500 -S 192.168.1.110
```

2. Digite o comando a seguir no terminal do PC (Kali atacante) para realizar o ataque no Arduino uno.

```
$ hping3 -c 10000 -d 1200 -S -w 64 -p 80 --flood  \
        --rand-source 192.168.1.110
```

Tabela 4.2: Descrição dos parâmetros utilizados no comando `hping3`.

Parâmetro	**Descrição**
`scan`	Ativação do modo *scanner*.
`0-500`	Significa que será escaneado as portas de 0 até 500.
`-c 10000`	Número de pacotes enviados.
`-d 1200`	Tamanho do pacote enviado em bytes.
`-S`	Enviando pacotes SYN.
`-w 64`	TCP *window size*.
`-p 80`	Porta de destino (Neste caso a porta 80 estava livre).
`--flood`	Mandar pacotes o mais rápido possível.
`--rand-source`	Utiliza IP randômico para não descobrirem quem está enviando os pacotes.
`192.168.1.110`	Endereço IP do dispositivo IoT que estamos atacando, que neste caso seria o IP do Arduino.

Enquanto o ataque está sendo realizado, consegue-se perceber que o tempo de resposta entre o Arduino e o PC (Kali Alvo) consequentemente aumenta, ou seja, as funções do Arduino ficam comprometidas. Para que isso fosse visualizado mais claramente, foi criado um servidor web no Arduino a partir deste site: `https://portal.vidadesilicio.com.br/shield-ethernet-w5100-servidor-web`, para que uma página web fosse fornecida, e assim, pudemos perceber que a página web não pôde ser mais ser acessada, dada a sobrecarga durante o ataque.

A sobrecarga ocorreu pois, houve um número exorbitante de envio de pacotes rapidamente, comprometendo assim o acesso ao Arduino. O mesmo ocorreu quando se tentou realizar o experimento no projeto de TCC do Lemes (2018), fazendo com que as informações que estavam sendo coletadas pelo sensores ficassem inacessíveis pelo usuário. Isto poderia acarretar problemas, pois como o usuário necessita analisar os dados transmitidos pelos sensores, quando este não está acessível, os dados recebidos pelo servidor ficam inexatos e sem confiabilidade.

(iii) Estratégias de Mitigação.

Para impedir ataques DoS, é melhor para a rede rejeitar requisições de *ping broadcast* para alcançar a rede de "fora" ou configurar um firewall para rejeitar todas as solicitações de pacotes (JAVED et al., 2018). Visto que o *firewall* pode ser uma das formas de evitar ataque DoS vindos da rede, tentamos utilizar os mecanismos do próprio roteador em questão para impedir ataques `hping3`, mas as configurações do *firewall* são muito limitadas e por este motivo, será proposta a instalação de um novo *firmware* chamado DD-WRT (BRAINSLAYER, 2020).

O DD-WRT é um *firmware* baseado em Linux, que adiciona diversas funcionalidades e oferece muitos recursos avançados não encontrados nos *firmwares* do roteador utilizado neste experimento (MAURICIO, 2020). É importante ressaltar que o método de *firewall* não é tão atual e totalmente eficiente, porém é de fácil acesso e configuração aos estudantes e pode ser melhor explorado.

4.2 Experimento 2 – Ataque a Raspberry Pi usando credenciais padrão

Ao configurar um Raspberry Pi, é fácil ignorar a alteração de senha padrão. Como muitos dispositivos de IoT, o sistema operacional Raspbian – padrão do Raspberry Pi – é instalado com uma senha padrão amplamente conhecida, deixando o dispositivo vulnerável ao acesso remoto. Levando isto em consideração, o propósito deste experimento é demonstrar o quanto dispositivos IoT com senhas padrão podem ser consideradas um risco tanto para o próprio dispositivo, quanto para a rede conectada a ele (GURUNATH et al., 2018).

Utilizando a ferramenta rpi-hunter, foi possível descobrir, acessar e enviar *payloads* – um código malicioso que executa uma ação destrutiva no sistema alvo, fornecendo acesso privile-

giado e permissões (e. g. criar um usuário, iniciar ou migrar um processo e até mesmo apagar arquivos) – para o Raspberry Pi com credenciais padrão conectado a mesma rede local utilizada pelo atacante. Utilizou-se também o Nmap, uma importante ferramenta do Kali que é útil na descoberta de portas abertas na rede (KALI LINUX PENETRATION TESTING TOOLS, 2020c). Este experimento foi baseado no site (WONDERHOWTO COMPANY, 2020).

(i) Preparação prévia para a realização do experimento

1. É necessário que se tenha um Raspberry Pi 3 Modelo B+ ou modelo Zero W, que esteja rodando o S.O. Raspbian ou Debian.
    - No nosso experimento utilizamos o Raspberry Pi 3 modelo B+ rodando o Raspbian.
    - Pode-se baixar o S.O. Raspian no site `https://www.raspberrypi.org/downloads/raspbian`
    - Conecte o Raspberry pi ao roteador usando um cabo Ethernet ou ao Wi-Fi.
    - Conecte o computador utilizado a mesma rede que o Raspberry Pi está conectado.
    - É necessário que se tenha um computador com Python para executar o rpi-hunter. Para isso, pode-se instalar o Python diretamente do site `https://www.python.org/downloads`.
2. Para executar o rpi-hunter, é necessário instalar algumas bibliotecas. Para esse fim, digite os seguintes comandos no terminal:

    ```
    $ sudo pip install -U argparse termcolor
    ```

    ```
    $ sudo apt -y install arp-scan tshark sshpass
    ```

    - Assim que as bibliotecas forem instaladas, pode-se instalar o rpi-hunter a partir do repositório do GitHub `https://github.com/BusesCanFly/rpi-hunter`
    - Para clonar o repositório no computador, digite no terminal:

        ```
        $ git clone https://github.com/BusesCanFly/rpi-hunter
        ```

    - Após clonar o repositório, deve se entrar na pasta do rpi-hunter, onde se encontrará baixado o arquivo *rpi-hunter.py* que estará pronto para ser executado. Para isso, digite no terminal:

        ```
        $ cd rpi-hunter
        ```

3. Neste momento, como o Raspberry Pi já está conectado a rede, deve verificar se o *secure shell* (SSH) está habilitado. Então, para esta verificação, execute o seguinte comando no terminal do Raspberry Pi:

   ```
   $ raspi-conf
   ```

   - Selecione "Opções de interface" e ative o acesso remoto da linha de comando para o Pi usando SSH.
   - Depois de habilitar o SSH, é necessário salvar suas opções e reiniciar o dispositivo.
   - Quando o Pi for reiniciado, digite o seguinte comando no terminal para descobrir o número IP do dispositivo:

     ```
     $ ifconfig
     ```

     Saber o número IP do dispositivo será importante para realizarmos o ataque.
   - Após descobrir o IP, execute o seguinte comando no computador para ter certeza que o SSH está sendo executado no Raspberry Pi:

     ```
     $ sudo nmap -p 22 192.168.1.101
     ```

     192.168.1.101 : Número IP do Raspberry Pi

     22 : Porta do SSH

     Se o Nmap scan indicar que a porta está "aberta", então o SSH está sendo executado no Raspberry Pi.

4. Por fim, antes de executarmos o ataque, é necessário transformar o arquivo *rpi-hunter.py* executável, e para isto, digite o seguinte comando no terminal:

   ```
   $ chmod +x rpi-hunter.py
   ```

(ii) Realização do Experimento

1. Neste instante, podemos processar o programa e observar vários comandos que podemos utilizar dependendo do nosso propósito em relação ao Raspberry Pi. Digite este código no terminal para visualizar as diversas funções que o rpi-hunter possui:

   ```
   $ sudo python rpi-hunter.py -h
   ```

2. No entanto, para esse experimento, iremos nos focar somente nos *payloads*. Para listar os *payloads* digite no terminal:

```
$ sudo python rpi-hunter.py --list
```

3. Pode-se observar que existem algumas opções disponíveis de *payloads*. Você pode tanto utilizar um *payload* contido na lista, como pode criar o seu proprio *payload* e enviá-lo. Neste caso, usaremos os dois exemplos.

4. Depois de verificar a lista de *payloads* disponíveis, iremos utilizar um comando para escanear a rede a procura de Raspberry Pi's conectados a ela, e enviar um *payload* para qualquer Raspberry Pi que possua credenciais padrão. Para isso digite no terminal:

```
$ sudo python rpi-hunter.py --payload whoami
```

Este *payload* irá retornar a palavra "pi" em resposta ao uso do comando *whoami*, se tudo ocorrer de acordo, ele mostrará também o número IP do Raspberry Pi encontrado, assim como pode ser observado na Figura 4.1.

Figura 4.1: Utilização do *payload whoami*

5. A seguir, vamos utilizar um dos comandos listados na seção de *payloads*. O *payload* utilizado será o *warning*. Quando executado, o script irá conectar ao Pi por SSH usando credenciais padrão, e então a mensagem "Change your password" aparecerá na linha de comando do terminal do Raspberry Pi.

   Para direcionar o *payload* para o Pi correto, será colocado o comando `-r` e o número IP do Raspberry Pi utilizado. Digite o seguinte comando no terminal:

   ```
   $ sudo python rpi-hunter.py -r 192.168.1.101 --payload warning
   ```

6. A criação de *payloads* é inovadora, pois estimula o estudante a elaborar seus próprios comandos e utilizá-los. Agora para a criação do nosso próprio *payload*, é necessário colocar o comando criado entre aspas depois do comando `--payload`. Com o intuito de reiniciar o aparelho, será enviado o comando `sudo reboot` como nosso *payload*. Digite o seguinte comando no terminal:

   ```
   $ sudo python rpi-hunter.py -r 192.168.1.101 --payload "sudo reboot"
   ```

   Conseguiu-se observar que, dispositivos com credenciais padrão podem ser vulneráveis e podem ter sua integridade questionável dependendo dos *payloads* recebidos.

(iii) Estratégias de Mitigação.

Para se prevenir desde tipo de ataque, é necessário sempre alterar as credenciais padrão do dispositivo (senha e login), isso serve tanto para o Raspberry Pi, quanto para qualquer outro dispositivo e, se não for possível sua alteração, não exponha o dispositivo diretamente a internet. É importante também desativar o SSH quando não precisar utilizá-lo, pois com o SSH habilitado a probabilidade de seu dispositivo ser invadido e controlado por terceiros é muito maior.

4.3 Experimento 3 – Interceptação de dispositivo *Blueto-oth*

Existem protocolos e tecnologias de comunicação que possuem curto alcance, e são comumente usados nas aplicações de IoT, e um deles é o *Bluetooth* (RIBEIRO, 2018). Este experimento foi baseado neste site (WONDERHOWTO COMPANY, 2019)

Por este motivo, para este experimento, utilizou-se a ferramenta Bettercap. A mesma é utilizada para escanear e interceptar dispositivos *Bluetooth*. Com esta ferramenta, foi possível

obter informações específicas sobre o dispositivo *Bluetooth* a fim de expor suas vulnerabilidades.
(i) Preparação prévia para a realização do experimento

1. É necessário somente instalar a ferramenta Bettercap. Para isso, como estamos utilizando o Kali, basta digitar no terminal:

   ```
   $ apt-get install bettercap
   ```

(ii) Realização do Experimento

1. Para iniciar o Bettercap, basta digitar no terminal:

   ```
   $ sudo bettercap
   ```

2. Após inicializá-lo, deve-se habilitar um módulo de descoberta de dispositivos *bluetooth*. Para isso, digite no terminal o seguinte comando e pressione enter, dessa maneira, aparecerão os dispositivos próximos de modo incensante.

   ```
   $ ble.recon on
   ```

3. Depois de alguns segundos, alguns dispositivos foram encontrados. Como é apenas um experimento simulado, utilizou-se um celular iPhone com o *bluetooth* habilitado para facilitar a análise de vulnerabilidade. Qualquer *smartphone* com acesso a *Bluetooth* poderia ter sido usado.

   Para listar os dispositivos encontrados e seus respectivos endereços MAC e outras informações, digite no terminal:

   ```
   $ ble.show
   ```

4. Por fim, a partir do scanner anterior, conseguimos o endereço MAC do dispositivo que queremos explorar, por isso agora, podemos realizar um escaneamento mais detalhado ao dispositivo. Para isso, digite juntamente com o endereço MAC do dispositivo:

   ```
   $ ble.enum 56:73:e6:ea:ce:c5
   ```

Informações mais detalhadas sobre o dispositivo apareceram na tela, como pode ser observado na Figura 4.2. Essas informações são relevantes, pois a partir delas, é possível explorar vulnerabilidades que o dispositivo *Bluetooth* pode conter, um dos exemplo seria a possibilidade de gravar outros tipos de dados em um campo gravável no próprio dispositivo, utilizando o

comando `ble.write`, para que a identificação do dispositivo possa ser feita, mesmo se o endereço MAC do mesmo for mudado.

Handles	Service > Characteristics	Properties	Data
0001 -> 0005	Generic Access (1800)		
0003	Device Name (2a00)	READ	iPhone
0005	Appearance (2a01)	READ	Generic Phone
0006 -> 0009	Generic Attribute (1801)		
0008	Service Changed (2a05)	INDICATE	
000a -> 000e	Apple Continuity Service (d0611e78bbb44591a5f8487910ae4366)		
000c	8667556c9a374c9184ed54ee27d90049	WRITE, NOTIFY, X	
000f -> 0013	9fa480e0496745429390d343dc5d04ae		
0011	af0badb15b9943cd917aa77bc549e3cc	WRITE, NOTIFY, X	
0014 -> 0017	Battery Service (180f)		
0016	Battery Level (2a19)	READ, NOTIFY	insufficient authentication
0018 -> 001d	Current Time Service (1805)		
001a	Current Time (2a2b)	READ, NOTIFY	insufficient authentication
001d	Local Time Information (2a0f)	READ	insufficient authentication
001e -> 0022	Device Information (180a)		
0020	Manufacturer Name String (2a29)	READ	Apple Inc.
0022	Model Number String (2a24)	READ	iPhone10,2
0023 -> 002c	Apple Notification Center Service (7905f431b5ce4e99a40f4b1e122d00d0)		
0025	69d1d8f345e149a898219bbdfdaad9d9	WRITE, X	
0028	9fbf120d630142d98c5825e699a21dbd	NOTIFY	
002b	22eac6e924d64bb5be44b36ace7c7bfb	NOTIFY	
002d -> 0038	Apple Media Service (89d3502b0f36433a8ef4c502ad55f8dc)		
002f	9b3c81d857b14a8ab8df0e56f7ca51c2	WRITE, NOTIFY, X	
0033	2f7cabce808d411f9a0cbb92ba96c102	WRITE, NOTIFY, X	
0037	c6b2f38c23ab46d8a6aba3a870bbd5d7	READ, WRITE, X	insufficient authentication

Figura 4.2: Informações sobre o celular iPhone com o uso da ferramenta Bettercap

(iii) Estratégias de Mitigação.

Um dos métodos mais eficazes para proteger o dispositivo é desligando o *Bluetooth*. Para situações como em IoT, que o uso do *Bluetooth* é em sua maioria necessário, sugerisse que apenas ative-o quando for realmente utilizá-lo (DENIS; ZENA; HAYAJNEH, 2016). Isso evita que ocorra exposição desnecessária a ameaças e ataques *Bluetooth*.

5 Conclusões

Neste projeto de iniciação tecnológica, foi proposta uma documentação detalhada de práticas que englobassem a segurança da informação em Internet das Coisas, tema este amplamente importante, visto que há objetos inteligentes com acesso a internet que estão sendo fabricados sem a devida atenção em relação a segurança destes dispositivos. Com o intuito de empregar essas práticas a estudantes, foram utilizados dispositivos acessíveis no que diz respeito a custos para adquiri-los, como o Arduino e o Raspberry Pi e incentivou-se a utilização de softwares

livres, como o Kali Linux, a fim de familiarizar o estudante no uso de algumas ferramentas disponíveis de maneira gratuita.

Além disso, foi levado em consideração ser o mais amplo possível em relação as práticas que envolvessem ataques atuais distintos a IoT, ou seja, abordou-se o ataque do tipo Negação de Serviço, um problema que pode prejudicar o acesso a dispositivos IoT, visto que é necessário que estes dispositivos estejam sempre disponíveis ao utilizador, ademais demonstrou-se a importância de não manter nos dispositivos credenciais padrão de senha e login porque isso pode deixar o dispositivo vulnerável a ataques, e por fim mostrou-se a importância de não manter o protocolo de comunicação *bluetooth* sempre habilitado porque pode deixá-lo inseguro e acessível a hackers.

Outrossim, a documentação dos experimentos realizados foram organizadas por temas e sub-temas com o propósito de facilitar o entendimento e a implementação do passo-a-passo a serem realizados pelos alunos e o docente.

Por fim, espera-se que os resultados obtidos possam ser implementados a fim de agregar conhecimento aos alunos e incentivá-los a explorar a área de segurança da informação relacionada a Internet das Coisas.

6 Trabalhos Futuros

Como o projeto teve que ser encerrado precocemente, não houve tempo hábil para a a implementação de outras práticas e metodologias anteriormente propostas. Além disso, algumas outras práticas não puderam ser implementadas por conta de limitações de hardware, de software e até mesmo de documentação explicativa. Assim, como proposta de trabalhos futuros são indicadas algumas práticas que podem ser evoluídas, testadas ou implementadas.

A primeira delas envolve a realização das práticas que não puderam ser realizadas por conta de limitações de hardware. Houve, se apoiando neste site (SPACEHUHN, 2018), a possibilidade de usar um Arduino com um controlador Ethernet, com o intuito de executar um ataque de falsificação ARP para bloquear a comunicação de todos os dispositivos conectados à rede local. Entretanto, como era necessário utilizar uma placa Ethernet específica, não foi possível realizar o experimento.

Outro experimento que teve limitação de instrumentos para sua realização foi a tentativa de clonagem de um cartão com tecnologia RFID, pois não havia um cartão com um UID que permitia sua alteração (HUMAN, 2018). Além disso, tentou-se outro experimento com o mesmo tema, mas não houve possibilidade para sua implementação, por não conseguirmos um PN532 e um conversor USB-UART (LANDAIS, 2019).

Visto que, um experimento desta natureza seria extremamente importante, em razão do uso frequente da tecnologia RFID para controle e segurança de acesso de pessoas a um estabelecimento, em transportes públicos, para gerência e armazenamento de estoque de uma empresa, para rastreamento de animais e até para controle da distância percorrida por atletas em eventos esportivos. Assim, o aumento dos níveis de segurança para esta tecnologia é imprescindível (DE SENNA; SOARES, 2020).

No que tange à limitação relacionada a documentação e ao software, houve tentativas de utilizar a ferramenta Bluesnarfer, disponível no Kali Linux para proceder um ataque a um dispositivo `Bluetooth`. Porém, não se obteve êxito, mesmo seguindo exatamente o passo a passo proposto no trabalho de Denis, Zena e Hayajneh (2016).

Em relação aos experimentos que foram implementados neste artigo, no primeiro experimento, em estratégias de mitigação, poderia ser implementado e documentado como seria o experimento se o *firmware* DD-WRT baseado em Linux fosse implantado no roteador.

Já no segundo experimento, para que o estudante possa explorar a sua criatividade, seria interessante que fosse feita a criação de outros comandos para serem utilizados no *payload*.

No terceiro e último experimento, seria pertinente se fosse utilizado as informações levantadas acerca do *smartphone* utilizado para realizar um ataque específico a ele.

Por fim, uma metodologia de ensino deve ser pensada para que as práticas possam ser aplicadas de modo a se preocupar com o conhecimento adquirido pelo estudante e poderiam ser realizadas estimativas de prazos e custos para se implementar essas práticas.

7 Agradecimentos

Primeiramente, agradeço ao Professor André Leon Sampaio Gradvohl por toda a orientação e apoio para realizar este projeto.

Ao Vinicius Gustavo de Jesus F. Lemes, estudante de mestrado da Faculdade de Tecnologia da Unicamp, por ter disponibilizado seu tempo e conhecimento me auxiliando em algumas práticas.

Ao Laboratório de Telecomunicações, citando os membros da equipe, o João Francisco Vianna e o Alexandre Pereira Silva, e o Laboratório LAPET, citando os membros da equipe Vinicius Gustavo de Jesus F. Lemes e Luiz A. Fabri Junior , por terem disponibilizado os equipamentos necessários para realizar os experimentos citados neste projeto de iniciação tecnológica.

Referências bibliográficas

ARDUINO. **Arduino**. 2019. Disponível em: <https://www.arduino.cc>. Acesso em: 17 abr. 2019.

BRAINSLAYER. **What Is DD-WRT?** 2020. Disponível em: <https://wiki.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F>. Acesso em: 7 jan. 2020.

BUSESCANFLY. **Rpi-hunter**. 2020. Disponível em: <https://github.com/BusesCanFly/rpi-hunter>. Acesso em: 7 jan. 2020.

CUI, Y.; LIU, Q.; ZHENG, K.; HUANG, X. Evaluation of Several Denial of Service Attack Methods for IoT System. **International Conference on Information Technology in Medicine and Education**, p. 794–798, 2018. ISSN 2474-3828. DOI: 10.1109/ITME.2018.00179.

DABBAGH, M.; RAYES, A. Internet of Things Security and Privacy. In: ANAIS do Internet of Things From Hype to Reality. Cham: Springer International Publishing, 2017. p. 195–223. ISBN 978-3-319-44858-9. DOI: 10.1007/978-3-319-44860-2_8.

DE SENNA, C. C. L.; SOARES, P. I. E. **Estudo de aplicações RFID na plataforma de IoT**. Trabalho de Conclusão de Curso – Universidade Federal Fluminense, Niterói, 2017. Disponível em: <https://app.uff.br/riuff/bitstream/1/5535/1/TCC_Caio_Pedro_entrega.pdf>. Acesso em: 6 jan. 2020.

DENIS, M.; ZENA, C.; HAYAJNEH, T. Penetration testing: Concepts, attack methods, and defense strategies. **2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)**, IEEE, 2016. DOI: 10.1109/LISAT.2016.7494156.

GURUNATH, R.; AGARWAL, M.; NANDI, A.; SAMANTA, D. An Overview: Security Issue in IoT Network. **Proceedings of the Second International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)**, p. 104–107, 2018. DOI: 10.1109.

HUMAN, I.-I. **Clone your RFID badge - Gain entry in style!** 2018. Disponível em: <https://www.youtube.com/watch?v=fyr4sYbB-Mo>. Acesso em: 4 jan. 2020.

IEEE. **IEEE Xplore Digital Library**. 2020. Disponível em: <https://ieeexplore.ieee.org/Xplore/home.jsp>. Acesso em: 4 jan. 2020.

JAVED, Y.; KHAN, A. S.; QAHAR, A.; ABDULLAH, J. Preventing DoS Attacks in IoT Using AES. v. 9, n. 3-11, 2018. ISSN 2289-8131.

JOSE, D. V.; VIJYALAKSHMI, A. An Overview of Security in Internet of Things. **Procedia Computer Science**, Elsevier B.V., v. 143, p. 744–748, 2018. ISSN 1877-0509. DOI: 10.1016/j.procs.2018.10.439.

KALI LINUX PENETRATION TESTING TOOLS. **Bettercap**. 2020. Disponível em: <https://tools.kali.org/sniffingspoofing/bettercap>. Acesso em: 7 jan. 2020.

_____. **Hping3**. 2020. Disponível em: <https://tools.kali.org/information-gathering/hping3>. Acesso em: 4 jan. 2020.

_____. **Nmap**. 2020. Disponível em: <https://tools.kali.org/information-gathering/nmap>. Acesso em: 4 jan. 2020.

KALITA, H. K.; KAR, A. Wireless sensor network security analysis. **International Journal of Next-Generation Networks (IJNGN)**, v. 1, n. 1, 2009.

LANDAIS, B. **How to clone a Mifare tag with a PN532**. 2019. Disponível em: <https://blandais.github.io/mifare/en>. Acesso em: 10 ago. 2019.

LEMES, V. d. J. F. Sistemas de monitoramento, controle e automação para irrigação com o uso do microcontrolador arduino, 2018.

MAURICIO. **Firmware DD-WRT no roteador Linksys WRT54G v8**. 2020. Disponível em: <https://mauricionh.wordpress.com/2017/11/30/firmware-dd-wrt-no-roteador-linksys-wrt54g-v8/>. Acesso em: 7 jan. 2020.

MINERVA, R.; BIRU, A.; ROTONDI, D. **Towards a definition of the Internet of Things (IoT)**. Torino, mai. 2015. p. 86.

OFFENSIVE SECURITY. **Kali Linux Penetration Testing Tools**. 2019. Disponível em: <http://www.who.int/mediacentre/factsheets/fs313/es/>. Acesso em: 17 abr. 2019.

RASPBERRY PI FOUNDATION. **Raspberry Pi**. 2020. Disponível em: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>. Acesso em: 4 jan. 2020.

RAWES, E. **Unsure about just what the Internet of Things is? Here's a breakdown.** 2019. Disponível em: <https://www.digitaltrends.com/home/what-is-the-internet-of-things>. Acesso em: 21 abr. 2019.

RIBEIRO, R. M. O. Segurança em IoT: Simulação de ataque em uma rede RPL utilizando Contiki. **Universidade Federal de Uberlância**, p. 30–38, 2018.

SPACEHUHN. **Arduino ARP-Spoof.** 2018. Disponível em: <https://github.com/spacehuhn/ArduinoARPspoof>. Acesso em: 4 jan. 2020.

VITUX. **ping**. 2020. Disponível em: <https://vitux.com/linux-ping-command/>. Acesso em: 7 jan. 2020.

WONDERHOWTO COMPANY. **Discover and Attack Raspberry Pis Using Default Credentials with Rpi-hunter**. 2020. Disponível em: <https://null-byte.wonderhowto.com/how-to/discover-attack-raspberry-pis-using-default-credentials-with-rpi-hunter-0193855/>. Acesso em: 4 jan. 2020.

_____. **How to Conduct Active Reconnaissance on Your Target with hping3**. 2013. Disponível em: <https://null-byte.wonderhowto.com/how-to/hack-like-pro-conduct-active-reconnaissance-your-target-with-hping3-0148092/>. Acesso em: 4 jan. 2020.

_____. **Target Bluetooth Devices with Bettercap**. 2019. Disponível em: <https://null-byte.wonderhowto.com/how-to/target-bluetooth-devices-with-bettercap-0194421/>. Acesso em: 4 jan. 2020.