Universidade Federal de Pernambuco

Graduação em Ciência da Computação

Centro de Informática
2017.1

Uma Avaliação de Desempenho de Técnicas de Identificação de Pontos de Acesso Não Autorizados

Trabalho de Graduação

Aluno: Caio César Leão Carneiro de Araújo (cclca@cin.ufpe.br)
Orientador: Paulo André da Silva Gonçalves (pasg@cin.ufpe.br)

Recife – Junho de 2017

Caio César Leão Carneiro de Araújo

Uma Avaliação de Desempenho de Técnicas de Identificação de Pontos de Acesso Não Autorizados

Trabalho de Graduação

Trabalho de Conclusão de Curso apresentado a Universidade Federal de Pernambuco - UFPE, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

Universidade Federal de Pernambuco

Orientador: Paulo André da Silva Gonçalves

Recife

2017

Caio César Leão Carneiro de Araújo

Uma Avaliação de Desempenho de Técnicas de Identificação de Pontos de Acesso Não Autorizados

Trabalho de Conclusão de Curso apresentado a Universidade Federal de Pernambuco - UFPE, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

Trabalho aprovado. Recife, 17 de abril de 2017.

Prof. Paulo André da Silva Gonçalves

Orientador

Prof. Carlos André Guimarães Ferraz

Avaliador

Recife

2017

Agradecimentos

Existem muitas pessoas a quem preciso agradecer e por motivos diferentes então peço desculpas caso essa seção fique um pouco longa.

Primeiramente gostaria de agradecer a minha mãe e irmãos que não só me deram a oportunidade de me dedicar completamente a Universidade mas também foram exemplos ímpares de com o trabalho duro e esforço, podemos alcançar qualquer objetivo independente de talento. Mostrando que quando queremos algo precisamos estar dispostos a fazer os sacrifícios necessários.

Também gostaria de agradecer a todos os amigos que fiz no Cin os quais sempre estavam dispostos a ajudar uns aos outros sem esperar nada em troca com o único objetivo de ajudar um amigo ou difundir o conhecimento, provando que não importa o quão diferente se é de alguém, basta ter uma coisa em comum para achar um amigo em qualquer pessoa. Como menção especial ficam as pessoas do grupo “Lionel Richie Go”, que estiveram comigo em boa parte da minha jornada e me ajudaram incontáveis vezes em minha vida acadêmica e pessoal.

Gostaria de agradecer também aos professores e monitores do Cin, os quais estavam sempre dispostos a repetir explicações incontáveis vezes, responder dúvidas durante a madrugada e finais de semana e que sempre se mostraram disponíveis e dispostos a ensinar e sempre responderam de forma positiva e com entusiasmo ao ver que um aluno estava disposto a ir além da sala para entender o conteúdo.

Resumo

O Número de usuários que utilizam a internet vem crescendo rapidamente, sendo registrados 3.739 bilhões de usuários em Março de 2017[3]. Junto com eles, cresce o número de pessoas utilizando a tecnologia Wi-Fi que permite a conexão com a internet sem fio. A internet se tornou algo essencial, tanto para a vida pessoal quanto para os negócios. A quantidade de operações envolvendo informações sensíveis que fazemos cresce a cada momento devido a facilidade oferecida e com isso, a segurança se torna mais importante. Infelizmente, os padrões definidos pelo protocolo de comunicação Wi-Fi, o IEEE 802.11 [1], demoram para se atualizar e não conseguem acompanhar as novas demandas que surgem em termos de segurança. Esse protocolo responsável pelas conexões sem fio ainda é vulnerável a diferentes tipos de ataques, entre eles o ataque de pontos de acesso falsos, ou como é mais comumente conhecido pela comunidade internacional, ataques de “Rogue Access Points”. São ataques relativamente simples de serem aplicados, precisando apenas replicar o Service Set Identifier (SSID), o Basic Service Set Identifier (BSSID) e o Medium Access Control (MAC) do ponto de acesso (AP) original. Fazendo isso, dispositivos tentando se conectar a internet utilizando o padrão de comunicação atual irão procurar pontos de acesso para estabelecer a conexão e irão se conectar com o ponto de acesso que possui as credenciais certas e com a maior força de sinal. Como o ponto de acesso do atacante copiou essas credenciais, o dispositivo não tem como saber que aquele ponto de acesso é falso, pois ele responde a todas as requisições feitas e possui as mesmas credenciais do ponto de acesso legítimo e contém as credenciais, copiadas pelo atacante, corretas. Este artigo irá utilizar uma técnica de identificação de pontos de acesso falso que utiliza o cálculo do desvio do relógio como uma credencial que é única para cada ponto de acesso e que não pode ser falsificada, irá analisar os resultados de testes feitos em vários AP’s diferentes e realizar um estudo do tempo necessário para cada etapa no processo de identificação do ponto de acesso.

Palavras-chave: Segurança, redes sem fio, desvio de relógio, detecção.

Abstract

The number of internet users is growing rapidly and with it, the number of people that also use Wifi and Wireless Connections. The internet has become essential to people's life, both for work and personal purposes. The amount of operations that involve sensitive information increases daily due to the commodity offered and with that, data security becomes even more relevant. Unfortunately, communications standards cannot keep up with the security needs as we do different transactions involving more and more sensitive data. The IEEE 802.11[1] protocol responsible for wireless communications is still vulnerable to the Rogue Access Point attack. This is a well known and relatively simple attack to perform. By copying the victim's Service Set Identifier (SSID), Basic Service Set Identifier (BSSID) and the Medium Access Control (MAC) it can make devices connect to the fake access point (AP) as these are the only credentials used by the device to check the AP's identity, if more than one AP is found with these credentials, it connects to the one that provides the stronger signal. Because the fake AP responds to every request as the real AP would, the device does not know it's connected to a fake AP and the users data is compromised. In this graduation thesis, we will apply a technique for identifying rogue AP's based on the AP Clock Skew and breakdown the time requirements necessary for each step of the identification process, as well as identify which process holds more influence over the total amount of time required.

Keywords: Access points, wireless network, network security, detection, clock-skew.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Fórmulas

1. Introdução

A crescente demanda para dispositivos móveis com uma conexão, sejam eles celulares, laptops ou até video-games somado ao desejo da ubiquidade faz com que a tecnologia wireless seja cada vez mais usada. O desejo pela ubiquidade e facilidade de uso elimina a necessidade de input humano para que os aparelhos se conectem a redes Wi-Fi conhecidas, redes essas que seguem o protocolo IEEE 802.11 de comunicação.

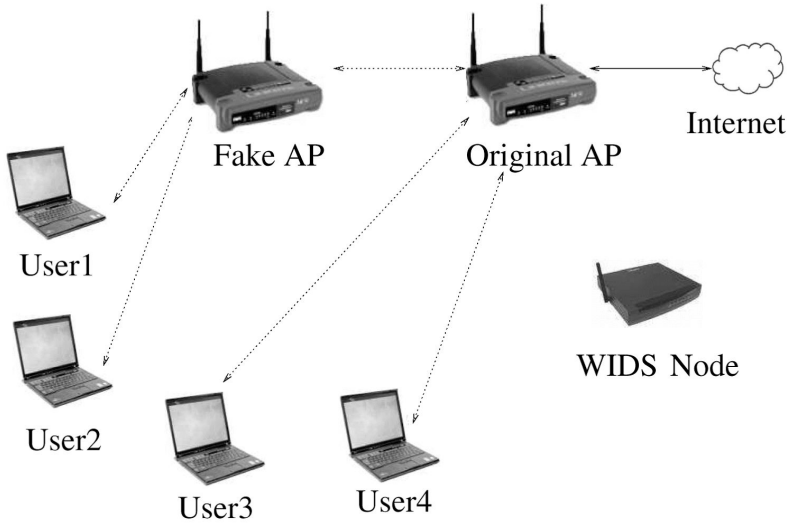
O protocolo usado infelizmente não acompanha as crescentes demandas de segurança atuais necessárias, o que se torna um problema de alta prioridade quando vemos a quantidade de operações realizadas envolvendo dados sensíveis. Dados esses que podem ser comprometidos por atacantes de várias formas, uma delas sendo os ataques de pontos de acesso falsos. Esses pontos de acesso falsos copiam as credenciais do ponto de acesso original como o SSID e o MAC e respondem todas as requisições feitas pelos dispositivos a ele conectados como o ponto de acesso original se portaria fazendo com que o dispositivo não saiba que está conectado a um AP falso. Uma vez que o dispositivo estiver conectado no AP falso, o atacante pode capturar os dados sensíveis, realizar um ataque de Denial of Service (DoS) impedindo que sua vítima tenha acesso a internet entre outros ataques.

O Sistema de Detecção de Intrusão em redes sem fio [2](WIDS) é capaz de identificar alguns pontos de acesso falso, através do endereço MAC do mesmo. Essa prevenção pode ser facilmente burlada através de um Mac-spoofing, onde o atacante muda o endereço MAC de seu ponto de acesso para que esteja entre um dos endereços confiáveis. Em casos como esse, a necessidade de mais uma credencial se torna mais que evidente.

Existem várias metodologias que utilizam métodos tradicionais de criptografia para identificação de pontos de acesso falsos através de certificados e assinaturas digitais. Os métodos criptográficos tradicionais são capazes de identificar pontos de acesso não autorizados e pontos de acesso falsos através de assinaturas digitais[2], porém os métodos não criptográficos são apenas capazes de identificar os pontos

de acesso não autorizados. A metodologia estudada neste trabalho é um método não criptográfico que é capaz de identificar pontos de acessos não autorizados e pontos de acessos falsos através do desvio de relógio dos pontos de acesso. A metodologia não procura substituir os modelos criptográficos atuais, mas sim fornecer um nível de segurança maior para sistemas que não são capazes de usar os sistemas tradicionais criptográficos e ser utilizado em conjunto aos métodos tradicionais em sistemas capazes.

Figura 1 - Cenário com ponto de acesso falso.



Fonte: [2]

1.1. Objetivos

O objetivo deste trabalho é avaliar as soluções propostas por [2] para detecção de pontos de acesso falso, replicando a ferramenta e métodos utilizados, estudando o comportamento da mesma em ambientes distintos e gerando dados detalhados sobre o tempo necessário para a criação de uma identidade para o ponto de acesso baseado no seu desvio de relógio. Ao estudar esses dados teremos um entendimento melhor sobre as limitações dos métodos no quesito de tempo, assim como, um melhor conhecimento sobre o peso de cada etapa e o quão elas influenciam no tempo final.

Para replicar a ferramenta, foram necessárias modificações no driver da placa wireless usada para a captura de pacotes com o objetivo de se obter uma maior granularidade do campo TSF Timestamp dos pacotes Probes e Beacon. O driver modificado foi o driver backports[3] para versão o sistema operacional Linux Mint 17.03. O estudo possui quatro partes, a coleta de dados, a execução dos métodos, a geração de dados relacionados ao tempo e na análise dos dados gerados. Cada etapa será melhor explicada e detalhada no decorrer deste trabalho. A ferramenta foi replicada na Linguagem Java e pode ser encontrada em[4], utilizando o wireshark[5] como suporte para realizar a captura de pacotes.

1.2. Estrutura do Trabalho

Para facilitar o entendimento da proposta por [1] e como a mesma foi replicada, o trabalho foi dividido em sete capitulos.

O primeiro capítulo apresenta a motivação para o trabalho assim como o contexto no qual o mesmo se insere e explicitando os objetivos propostos.

O segundo capítulo é uma referência teórica com conceitos básicos e termos técnicos usados ao longo do trabalho a fim de familiarizar o leitor com os mesmos e fornecer um melhor entendimento sobre o trabalho.

O terceiro capítulo descreve o modelo de ameaça que os métodos utilizados no trabalho atacam

O quarto capítulo consiste da metodologia aplicada para estimação do desvio de relógio a partir dos quadros beacons, Softwares necessários, detalhamento da modificação do driver, algoritmos e modelos matemáticos utilizados.

O quinto capítulo descreve a implementação da ferramenta, suas fases e peculiaridades.

O sexto capítulo contém todos os resultados obtidos e analisados.

O sétimo capítulo é a conclusão do trabalho, com sugestões e direções para trabalhos futuros.

2. Conceitos Gerais

Neste capítulo serão descritos os termos utilizados ao longo do trabalho e conceitos básicos necessários para o entendimento do mesmo.

2.1. IEEE 802.11

Redes sem Fio IEEE 802.11 são mais comumente conhecidas como redes Wi-Fi. Define uma série de padrões de transmissão e codificação a serem seguidos para comunicação sem fio na camada física. Atualmente, é o padrão aceito pela comunidade, tendo a grande maioria de dispositivos móveis saindo da linha de produção já equipados com uma interface IEEE 802.11

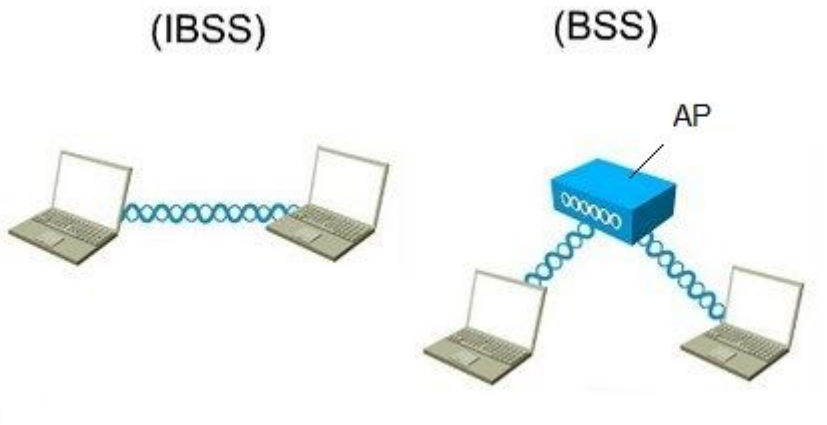
2.1.1. Arquitetura

São definidos pela IEEE 802 dois tipos de serviços, o *Basic Service Set* (BSS) e o *Extended Service Set* (ESS).

O BSS é formado por um conjunto básico de dispositivos capazes de fornecer serviços de comunicação que fazem parte dele. Comumente, entre esses dispositivos temos um *Ponto de Acesso* (Access Point, AP) e clientes opcionais. Existem BSSs que não tem um AP dentro do seu conjunto de dispositivos, redes

que se enquadram nessa descrição são classificadas como redes Ad-hoc ou uma Independent Basic Service Set (IBSS)[1] .

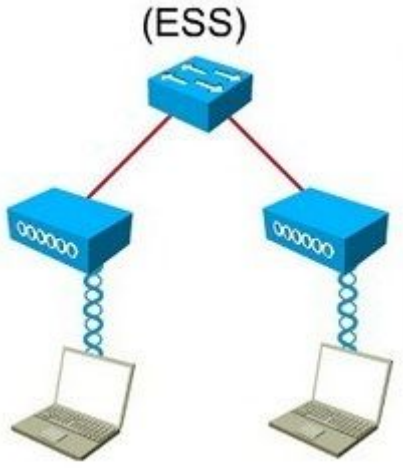
Figura 2 - Serviço IBSS e BSS.



Fonte: [7]

Quando nós fazemos a junção de dois ou mais BSSs através de uma infraestrutura de comunicação como hubs, roteadores e computadores, esses conjuntos passam a ser chamados de *Extended Service Set* (ESS)[1].

Figura 3 - Exemplo de serviço ESS



Fonte: [7]

2.1.2 Métodos de Associação

Como identificação básica dos BSSs temos o *Service Set Identifier (SSID). Esses SSIDs* são os nomes da rede que vemos quando quando escolhemos uma rede para realizarmos a conexão e é através deles que as estações identificam quais a *BSS*s elas irão se associar. Os *SSID*s podem ser customizados acessando o AP da rede e a varredura feita para identificar os *BSS*s ao alcance do dispositivos pode ser feita de forma ativa ou passiva.

2.1.2.1. Varredura Passiva

Nesse tipo de varredura, o dispositivo apenas recebe quadros do tipo Beacon os quais são enviados periodicamente, tipicamente de 10 a 100 quadros dependendo da configuração do ponto de acesso. Esses quadros são enviados em broadcast, sendo assim qualquer dispositivo que estiver ao alcance do AP e possuir uma interface IEEE 802 pode capturá-los e identificar quais BSSs estão em seu alcance. Com a lista formada, o usuário ou o próprio dispositivo pode selecionar um deles para enviar o pedido de associação.

2.1.2.2. Varredura Ativa

Nesse tipo de varredura o dispositivo ativamente envia em broadcast quadros de solicitação de investigação e todos os BSSs que recebem esse quadro respondem com seu SSID, a menos que explicitamente configurado para ignorar esse tipo de solicitação. Com as respostas, o dispositivo cria a lista de BSSs disponíveis.

2.1.3. Diferentes modos de operação de uma interface de rede

Interfaces de redes normalmente possuem três tipos de modos de interface

que podem assumir. Eles são os modos Passivo, Ativo e Estação.

O modo ativo, também conhecido como mestre ou AP, a interface de rede tem um comportamento igual ao de um AP comum, fornecendo seu próprio SSID e permitindo que outras estações se associem a ela.

O modo passivo, mais comumente conhecido como modo monitor, é quando a interface de rede tem a capacidade de capturar todos os pacotes que estão em seu alcance, independente de quem enviou ou para quem enviou. É nesse modo em que a captura dos pacotes beacons foram feitas para nosso estudo.

O modo estação é quando a rede troca pacotes com um AP a qual ela está associada.

2.1.4. Tipos de quadros

O padrão de comunicação IEEE 802.11 especifica três tipos de quadros para comunicação, sendo eles: quadros de dados, quadros de gerenciamento e quadros de controle.

Quadros de gerenciamento são responsáveis pelo controle de estabelecimento e manutenção da comunicação. Esses quadros possuem vários subtipos que ajudam na manutenção das Redes Locais Sem Fio (Wireless Local Area Networks, WLAN). Entre esses subtipos temos os quadros de associação, autenticação, dissociação, beacon entre outros que são enviados com alta frequência, independente da aplicação.

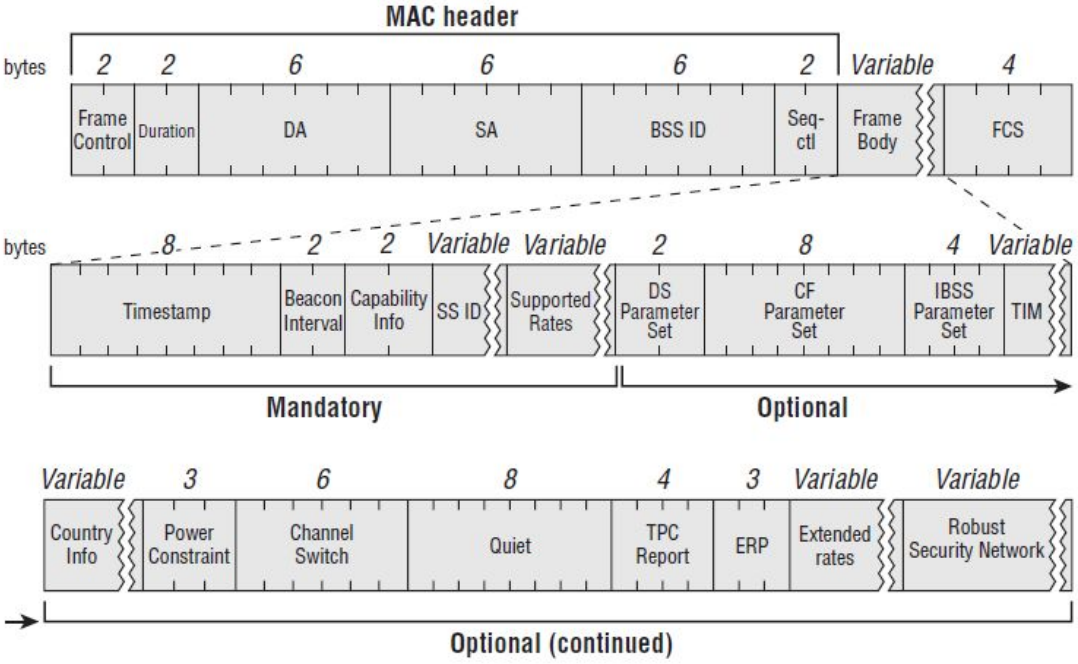
Quadros de dados são responsáveis por carregar protocolos e dados das camadas superiores, utilizados para o envio e recebimento de dados.

Quadros de controle são responsáveis por ajudar na comunicação entre estações. Entre os subtipos de quadros que são classificados como quadros de controle, estão o quadro de Request-To-Send (RTS), Quado de Clear-To-Send(CTS), Not Acknowledged (NACK), Acknowledgement (ACK), entre outros.

Nesse trabalho utilizamos de forma extensiva os quadros de gerenciamento. Em específico, os quadros beacons, pois são enviados com uma alta frequência,

100 por segundo em média, e nele podemos extrair a informação do *Timestamp*, o qual será explicado com mais detalhes na metodologia.

Figura 4 - Estrutura de um quadro Beacon



Fonte [8]

2.2. Cabeçalho Radiotap

É importante entender que o cabeçalho Radiotap não faz parte da estrutura de um quadro 802.11. É um container para metadata dos quadros, foi desenvolvido devido a ausência de containers dedicados para metadata. Por estar fora da estrutura do quadro 802.11 possui uma maior flexibilidade e é por isso que é capaz de fornecer dados adicionais sobre os quadros, os quais cabeçalhos como *Prism* ou *AVS* não são capazes, pois caso o fizessem iriam quebrar diversos parsers que esperam esses cabeçalhos contendo um número específico de bytes.[9]

2.3 Classificação de pontos de acesso

Pontos de acessos falsos são classificados por Kim[10] em duas categorias: *Rogue APs* e *Fake AP*s. Pontos de acesso classificados como *Fake APs* ou *APs Falsos* são aqueles instalados por atacantes com intenções maliciosas, sejam elas falsificar mensagens, obter informações ou outros tipos de ataques, normalmente utilizam a conexão sem fio para se infiltrar na rede alvo. São classificados como *Rogue AP*s pontos de acesso instalados sem intenção maliciosa, porém tiveram sua instalação feita sem a explícita permissão do administrador da rede. Eles são instalados comumente em busca de comodidade e normalmente utilizam ethernet para se conectarem a rede.

3. Modelo de Ameaça

O aumento da popularidade de redes locais sem fio (WLAN), baseadas nos padrões IEEE 802.11 tem visto um crescimento constante nos últimos anos. De forma similar, é possível ver um aumento na demanda de serviços ubíquos de qualidade como aplicações em internet das coisas. Isso significa que os padrões de comunicação estabelecidos pelo IEEE 802.11 serão cada vez mais usados e a demanda por um modelo capaz de identificar *Fake AP*s se torna mais evidente.

Um dos dois cenários mais comuns em que o AP Falso opera é quando apenas o AP Falso está ativo. Isso pode ocorrer por conta de um problema no AP Original onde o mesmo foi desativado, talvez pelo simples fato que o usuário saiu do alcance do AP Original e está em uma área que apenas o AP Falso cobre ou até mesmo podendo ter o realizando um ataque de negação de serviço no AP Original, fazendo com que apenas o falso responda.

O segundo cenário mais comum é quando temos dois AP’s ativos ao mesmo tempo. Com os mecanismos de seleção atual na escolha de AP, o usuário de conectará com o AP que oferecer o sinal de maior força e tiver as credenciais corretas.

É importante lembrar que a falsificação dessas credenciais podem ser feitas

facilmente, onde o AP Falso pode adotar as configurações de BSSID, SSID e endereço MAC do AP Autorizado e essas são as únicas credenciais atualmente usadas para a identificação de pontos de acesso.

4. Metodologia

Nesta seção iremos descrever as dependências de softwares e hardwares assim como algoritmos e modelos matemáticos usados no trabalho.

4.1. Dependências

Para replicar corretamente os experimentos, foram necessários os seguintes softwares e hardwares:

- Sistema operacional Linux, distribuição Mint, versão 17.03 Rosa, edição Cinnamon 64 bits.[11]
- Versão do Kernel: 3.19.0-32-generic.
- Adaptador Wireless contendo um chipset Atheros. Foram utilizados dois na fase da coleta. O Adaptador Wireless TP-Link, Modelo TL- WN722N e o adaptador Killer E2200 Game Networking.
- Driver backports modificado, versão modificada foi a 3.18[12]
  OBS: A versão do Kernel precisa ser superior ou igual a versão do backports.
- O pacote aircrack, que contém o programa airmon-ng, utilizado para colocar a interface de rede em modo monitor, utilizado na fase de coleta. Versão utilizada: 1.2
- O pacote Wireshark, utilizado em conjunto com o airmon-ng na fase de coleta para captura de quadros Beacon.
- Eclipse Neon, compilador e gerenciador de projetos em Java, linguagem utilizada na implementação dos métodos matemáticos.[13]
- JFreeChart, biblioteca Java open source utilizada para criação de gráficos.[14]

4.2 Modificação do Driver

Para implementar corretamente a ferramenta, é necessário realizar uma modificação no driver na interface de rede do dispositivo que está rodando o programa. Como cada fabricante possui um driver diferente, a proposta estudada utiliza placas contendo chipset Atheros e utilizamos duas interfaces contendo esse mesmo chipset para realizarmos nossos experimentos.

A Atheros fornece drivers para seus produtos de código aberto, o driver Atheros utilizado foi o Ath9k[15], devido a facilidade proporcionada pela biblioteca compartilhada que executa o protocolo 802.11.

O Principal objetivo da modificação do driver é aumentar a granularidade do campo *TSF Timestamp*, encontrado no cabeçalho de metadados Radiotap. A maior granularidade proporcionará uma estimativa mais confiável para o cálculo de desvio de relógio.

A modificação foi realizada apenas em um arquivo, o "*net/mac80211/r.c"*. Ele é responsável pelo processamento de pacotes recebidos pelo protocolo 802.11. As modificações foram:

Adicionar as seguintes variável no método ***ieee80211_add_rx_radiotap_header(),*** que é iniciada na **linha 132:**

```
struct timeval timevalue;
unsigned long long int timeDay;
```

timevalue é usada para armazenar o retorno do método "*gettimeofday(),* o qual retorna um *struct timeval*, composto por *tv_sec* e *tv_usec* que são o tempo em segundos e seu complemento em microsegundos desde do *Epoch* (1 de Janeiro de 1970, 00:00:00), padrão utilizado para geração de *timestamps*, e o *timeDay* é a variável que iremos armazenar o valor total em microsegundos contido no timevalue.

Substituir o código atual da **linha 186**:

put_unaligned_le64(ieee80211_calculate_rx_timestamp(local,status,mpdulen,0),pos);

Pelos seguintes comandos:

Comando 1: do_gettimeofday(&timevalue);

Comando 2: *timeDay = ((unsigned long long) timevalue.tv_sec) * 1000000 + ((unsigned long long) timevalue.tv_usec);*

Comando 3: *put_unaligned_le64(timeDay, pos);*

O Primeiro comando é a chamada do sistema operacional usada na proposta de [1]. O segundo comando faz uma conversão da chamada de sistema, a qual retorna um *struct timeval* e o converte para um inteiro de 64 bits e o terceiro comando adiciona no campo *Time Synchronization Function Timer (TSF Timestamp)* do cabeçalho *radiotap,* o valor já convertido.

Essas modificações apenas são aplicadas na interface quando a mesma se encontra no modo monitor, não afetando assim a performance da interface de rede quando a mesma se encontra em qualquer outro modo.

4.3. Conjunto de Dados

O conjunto de dados é criado a partir de informações obtida em quadros Beacon capturados com a interface em modo monitor. Dentro dos quadros Beacon, temos uma estrutura de 8 bytes representando o *timestamp* conforme a figura 3. Esse é o *timestamp* colocado no pacote pelo AP, no momento em que o envio do mesmo é iniciado e se encontra na base 16.

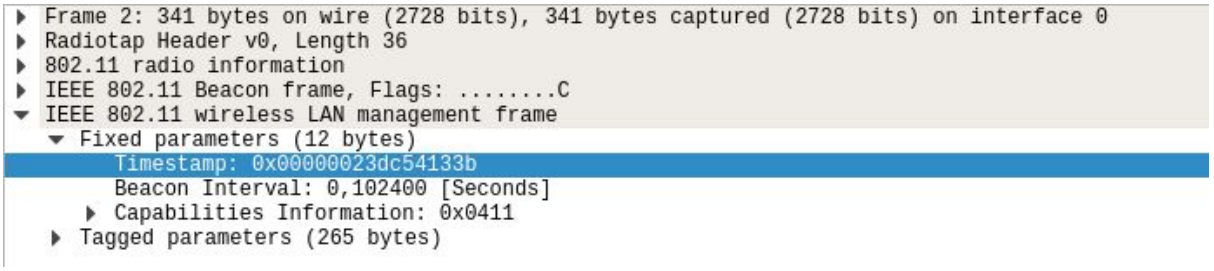


Figura 5 - Campo timestamp destacado

O segundo campo que nós pegamos para construir o nosso conjunto de dados é o *TSF Timestamp*, encontrado no cabeçalho radiotap. Esse *TSF Timestamp* é colocado no cabeçalho quando o pacote é capturado pela interface de rede e é o valor com maior granularidade que obtemos através da modificação do driver da interface de rede e se encontra no formato de um inteiro na base 10, com precisão de microsegundos.

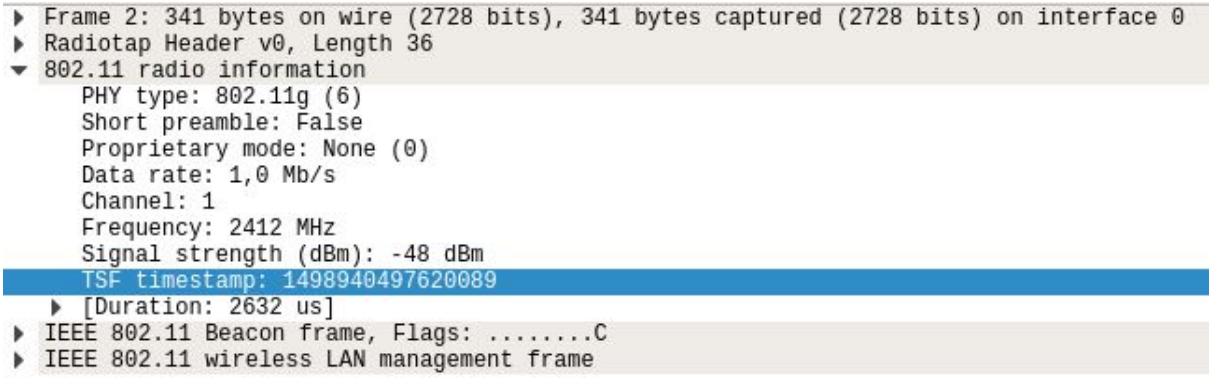


Figura 6 - Campo TSF Timestamp destacado

Nós utilizamos essas duas medidas para estimarmos o valor de desvio de relógio de um AP como proposto por [1].

O relógio de um AP é formado por um **Oscilador** e um **Contador**. O **Oscilador** é controlado por um cristal e oscila em frequências fixas. Essa frequência depende primordialmente do tipo de cristal e o ângulo em qual o mesmo foi cortado. Mesmo cristais do mesmo tipo e de mesmo corte possuem frequências levemente diferentes devido a limitações no processo de corte. O **Contador** por sua vez mantém o registro do número de oscilações do **Oscilador**.

Agora vamos assumir que nosso dispositivo *fingerprinter* (Um nó WIDS) tenha recebido n quadros beacon de um ponto de acesso em particular. Vamos

definir como T_i o *timestamp* do i^{th} quadro beacon convertido para a base 10 e t_i como o tempo de chegada do quadro no dispositivo, em microsegundos (Valor encontrado no campo *TSF Timestamp* do wireshark). Iremos definir S_i como o tamanho do quadro e R_i a taxa que o mesmo é enviado. Com essas informações definidas, o tempo de acordo com o relógio do AP que o dispositivo fingerprinter recebe o i^{th} quadro beacon é definido como $T_i + S_i / R_i$. Definimos nosso offset aproximado do i^{th} quadro beacon como θ_i e a diferença de tempo entre o primeiro e o i^{th} quadros recebidos pelo dispositivo *fingerprinter* de acordo com seu próprio relógio seja x_i. Então ficamos com:

$$x_i = t_i - t_1 \qquad \text{(Fórmula 1)}$$

$$\theta_i = \left(\left(T_i + \frac{S_i}{R_i}\right) - \left(T_1 + \frac{S_1}{R_1}\right)\right) - (t_i - t_1) \qquad \text{(Fórmula 2)}$$

Os quadros Beacon, na grande maioria dos casos são enviados à uma taxa constante e tamanho fixo. Logo, podemos assumir que $\frac{Si}{Ri} = \frac{S1}{R1}$,o que resulta em:

$$\theta_i = (T_i - T_1) - (t_i - t_1) \qquad \text{(Fórmula 2.1)}$$

Nessas circunstancias, se o desvio de relógio de um AP em particular permanecer constante, o gráfico feito através dos pontos ($xi, \theta i$), nós iremos obter um padrão aproximadamente linear. E o desvio de relógio seria representado pelo coeficiente angular desse padrão linear.

Iremos então referenciar o conjunto de dados compostos pelos pontos $\{(x1, \theta1), (x2, \theta2), \ldots, (xn, \theta n)\}$ como o conjunto de dados offset de um dado AP.

4.4. Método da Programação Linear (MPL)

O MPL é uma solução aplicada quando precisamos maximizar/minimizar uma

função objetivo enquanto obedecemos todas as restrições lineares do problema. Iremos aplicar o método para estimarmos o desvio de relógio de um AP a partir do conjunto de dados *offset*.

Seja o nosso conjunto de dados *offset* do formato:

$$\{(x1,\ \theta1),\ (x2,\ \theta2),\ \ldots,\ (xn,\ \theta n)\}$$

O MPL encontra uma linha descrita como $y = \delta x + \phi$, que representa o limite superior do conjunto de pontos de nossos dados, tal que δ representa a inclinação da reta e ϕ o ponto em que ela cruza o eixo y. A estimativa do desvio de relógio do AP será dado por a, tal que a seguinte função objetivo seja minimizada

$$\frac{1}{n}\sum_{i=1}^{n}(\delta.x_i + \phi - \theta_i) \qquad \text{(Fórmula 3)}$$

e as seguintes restrições sejam obedecidas:

$$\delta.x_i + \phi \geq \theta_i, \forall i = 1,2,\ldots,n. \qquad \text{(Fórmula 4)}$$

Para encontrarmos essa reta, o algoritmo Simplex foi utilizado. Esse algoritmo consiste em verificar todas as retas que podem ser formadas utilizando todas as possíveis combinações de 2 pontos dos nossos dados. Para cada reta formada, nós verificamos se o conjunto de restrições é obedecido, e verificamos o valor da função objetivo. Na primeira iteração, como não temos valor da função objetivo, a primeira reta que cumpre as restrições é tida como resposta temporária, até que encontremos uma reta que cumpra as restrições e tenha um valor da função objetivo menor que a resposta temporária. Fazendo isso para todas combinações de pontos, garantimos a reta com os valores que tornam o valor da função objetivo o menor possível e cumprem as restrições obedecidas.

O MPL minimiza quaisquer atrasos inesperados devido a sua alta tolerância a outliers. O lado positivo disto é que, mesmo em casos onde temos um AP enviando quadros beacon, a estimativa ficará próxima da real, o lado negativo é que se o

atacante for qualificado o suficiente, ele pode adquirir um AP com desvio de relógio similar e inserir uma quantidade pequena de pacotes Beacon no conjunto de dados, fazendo com que o MPL trate esses pacotes do AP falso como outliers e não seja capaz de identificar a presença de um AP Falso. Por conta disso, usaremos o Método dos Mínimos Quadrados, o qual é extremamente sensível a outliers para ajudar na identificação de APs falsos em conjunto com o MPL.

4.5. Método dos Mínimos Quadrados (MMQ)

Utilizamos o método dos mínimos quadrados (MMQ) quando queremos ajustar uma curva a um conjunto de dados. Ele é aplicável em nosso contexto pois, apesar do comportamento linear, o desvio de relógio não é constante mas existe dentro de um intervalo constante.

Usaremos o MMQ para o mesmo fim que usamos o MPL, estimar o valor do desvio de relógio de um dado AP a partir de um conjunto de dados *offset* capturados do mesmo. Dado nosso conjunto de dados *offset* do tipo:

$$\{(x1,\ \theta1),\ (x2,\ \theta2),\ \ldots,\ (xn,\ \theta n)\}$$

O método encontra uma reta definida por $y = \delta x + \phi$, onde a é a inclinação da reta e b é o ponto em que a reta cruza o eixo y, de tal forma que:

$$\sum_{i=1}^{n}(\theta_i - (\delta.x_i + \phi))^2$$

(Fórmula 5)

permaneça mínimo. O valor estimado do desvio de relógio será dado pela inclinação da reta δ que minimizar a função. Com o conjunto de dados *offset* definido, podemos encontrar a inclinação da reta, assim como o ponto de intercessão com o eixo Y através da seguinte forma:

$$\delta = \frac{\sum x\theta - \frac{\sum x \sum \theta}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}}$$

(Fórmula 6)

$$\phi = \frac{\sum \theta - (\delta . \sum x)}{n}$$

(Fórmula 7)

Já podemos ver que para o cálculo do MMQ precisamos apenas varrer nosso conjunto de dados uma vez para estimarmos o valor do desvio de relógio, já que não precisamos calcular o ϕ, o que torna o algoritmo muito mais rápido que o MPL.

Devido a baixa tolerância a outliers do MMQ ele é capaz de identificar possíveis pontos de acesso falsos, mesmo que o atacante consiga um AP com desvio de relógio parecido e coloque um baixo número de pacotes. O lado negativo dessa característica é que outliers são relativamente comuns em ambientes de transmissão sem fio devido ao grande número de pacotes sendo transmitidos em uma mesma faixa de frequência. Podemos então utilizar o MMQ junto com o MPL para estimarmos o desvio de relógio de um AP de forma confiável e que sejamos capazes de identificar a existência de APs falsos.

5. Implementação

A implementação da ferramenta usada para captura, armazenamento e cálculo das estimativas do desvio de relógio foram todas feitas utilizando um único Laptop - um MSI GT70 PC2 Dominator rodando Linux Mint 17.3 Rosa. Utilizamos dois dispositivos wireless na captura, a interface de rede do próprio laptop - Uma Killer E2200 Game Networking - e um adaptador wireless TP-Link Modelo TL-WN722N. Ambos possuem chipsets Atheros e a modificação no driver foi a mesma para ambas as interfaces e dão suporte ao modo monitor, necessário para realizarmos a captura de pacotes.

A geração correta de resultados utilizando essa metodologia está diretamente ligada com a corretude da implementação das etapas do projeto, as quais serão descritas a seguir.

5.1 Coleta

Durante a fase de coleta, a interface de rede usada para a captura é colocada em modo monitor utilizando o pacote airmon-ng para que possamos realizar a captura de quadros beacons. Para garantir a inexistência de APs falsos, os métodos de identificação propostos por [1] foram usados e os resultados determinaram que não haviam APs falsos agindo no alcance dos experimentos.

Os quadros beacons foram então capturados pelas interface de redes modificada. Gerando seis conjunto de dados para três diferentes APs, cada AP contendo dois conjuntos de dados gerados a partir da captura realizada por diferentes interfaces de rede. Com a interface de rede modificada, o campo *TSF Timestamp* do cabeçalho *radiotap* possui a granularidade necessária em microsegundos.

5.2. Separação

Após determinar a inexistência de APs falsos e de capturar quadros suficientes de cada rede para realizarmos os experimentos necessários nós dividimos o conjunto de dados baseado baseado no endereço MAC do AP a qual eles pertencem. Fizemos a separação assim pois, apesar do endereço MAC ser facilmente falsificado verificamos anteriormente a inexistência de APs falsos atuantes no alcance do dispositivo usado para captura dos quadros.

5.3. Análise

Após a coleta e separação de dados em vários arquivos, sendo eles

classificados por AP e interface utilizada para captura, nós começamos o processo de análise. Rodando os métodos MPL e MMQ para estimativa de desvio de relógio proposto por Jana[2] e gerando dados sobre o tempo que cada processo necessário levou para ser concluído, assim como, dados mais detalhados sobre os processos que mais possuem peso sob o tempo total necessário para a criação da identidade digital do ponto de acesso identificando em qual partes do processo devemos focar estudos e propostas para poder diminuir o tempo necessário para geração de uma identidade virtual do AP, aumentando assim a comodidade e permitindo que a implementação dessa técnica não tenha um impacto forte na ubiquidade dos produtos futuros enquanto fornece um nível a mais de segurança.

6. Resultados

O primeiro passo para garantir que a análise temporal do processo é válida é garantir que o processo está funcionando corretamente. Então testes foram feitos para determinar se a estimação do desvio de clock estavam corretas. Os resultados foram inicialmente comparados com a implementação de [16], quando os resultados deram os mesmos para o mesmo conjunto de dados, devido a natureza determinística dos algoritmos, pudemos concluir que os resultados estavam corretos. Tanto para o MMQ quanto para o MPL. Dentre os testes realizados, pegamos os resultados de estimativa mais próximos, podemos ver que os métodos, tanto MMQ quanto MPL consegue corretamente diferenciar os APs. Sabendo que os métodos estão funcionando corretamente e classificando APs corretamente, como vemos na figura 7 onde temos no eixo X o número do dataset usado e no eixo Y o valor de desvio de relógio estimado em partes por milhão (ppm), podemos então começar a análise temporal dos processos.

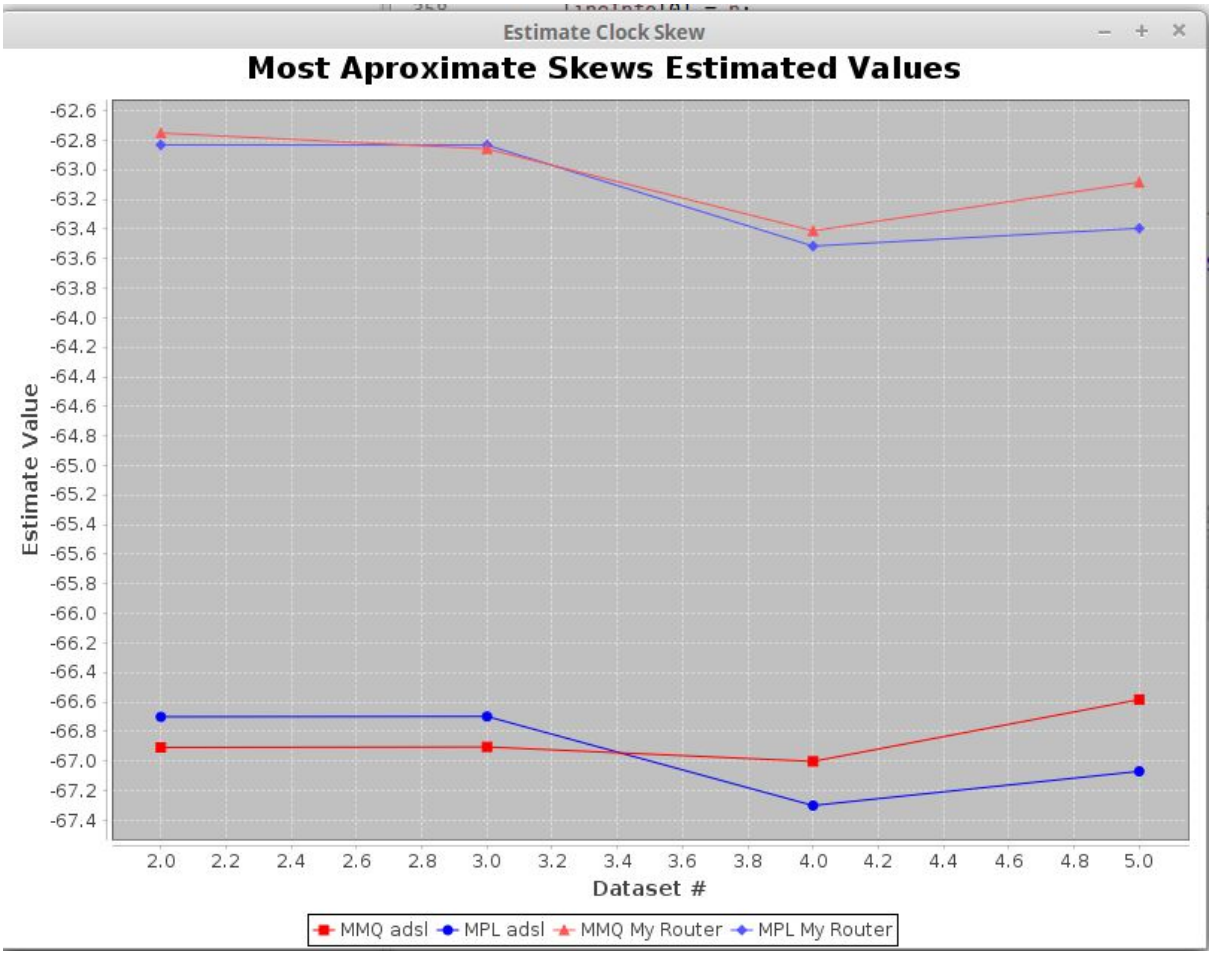


Figura 7 - Comparação entre desvio de relógio de 2 AP's usando MMQ e MPL

Na tabela 1 podemos ver cada etapa do processo de identificação de APs e o tempo em % que ela consumiu.

	Tempo em % levado em cada etapa			
# Pacotes Examinados	Captura de Pacotes	Gerenciamento de dados	MPL	MMQ
100	99,90%	0,08%	0,02%	0,00%
200	99,91%	0,06%	0,03%	0,00%
300	99,92%	0,05%	0,03%	0,00%
400	99,90%	0,06%	0,05%	0,00%
500	99,89%	0,05%	0,05%	0,00%
600	99,90%	0,04%	0,06%	0,00%
1000	99,87%	0,04%	0,09%	0,00%

Tabela 1 - Percentagem do tempo total de cada parte do processo

Podemos ver que a captura de pacotes é responsável quase que

completamente pelo tempo necessário. Se torna fácil de entender o porque isso acontece quando nós lembramos que o intervalo médio entre envios de beacons é 100 milisegundos, tendo em média um envio de 100 beacons por segundo no máximo. Para verificar a normalidade no intervalo do recebimento de Beacon Frames, usando o mesmo conjunto de dados pudemos criar a tabela 2 que confirma que o intervalo se mantém próximo.

# Pacotes Examinados	Tempo Médio entre Pacotes	Tempo total:
100	104.45 ms	10479.50 ms
200	103,91 ms	24809,45 ms
300	117,42 ms	35.253,73 ms
400	114,95 ms	46.023,70 ms
500	114,90 ms	57.502,28 ms
600	115,20 ms	69.181,69 ms
1000	111,62 ms	111.748,54 ms

Tabela 2 - Tempo total do processo e tempo médio entre quadros Beacon

Como a etapa de captura de pacotes representa a parte mais significante dos processos necessários para a geração de uma identidade virtual para os APs baseada no desvio de relógio e não temos direto controle sobre ela, iremos isolar ela para que possamos analisar as outras etapas de forma significativa. Mesmo capturando os pacotes com o maior número de envios por segundo, o tempo necessário para a captura continuou sendo o mais elevado por uma grande margem.

Para gerar os dados mostrados a seguir, a execução do método foi feita 10 vezes no mesmo dispositivo e nas mesmas condições, usando como valores as média aritmética entre as 10 execuções. Dessa vez, para termos um melhor entendimento no que acontece nas outras etapas não estaremos contabilizando o tempo necessário para captura de pacotes e as percentagens mostradas são relativas ao tempo levado para a execução do processo sem contar o tempo para captura dos quadros.

# Pacotes Examinados	Gerenciamento de dados	MPL	MMQ
100	85.64%	14.27%	0.09%
200	73.27%	26.64%	0.09%
300	59.88%	39.92%	0.2%
400	52.88%	47.01%	0.11%
500	69.94%	29.98%	0.08%
600	26.63%	73.23%	0.14%
1000	12.49%	87.45%	0.06%

Tabela 3 - Percentagem relativa de tempo utilizada em cada parte do processo

Podemos ver na tabela 3 que a fatia necessária de processamento do MPL cresce à medida que o número de pacotes aumenta. Esse comportamento acontece pois o algoritmo Simplex tem um custo exponencial. Ainda é uma questão aberta se existe uma variação algoritmo do método de programação linear que é não-exponencial ou polinomial. É importante notar o outlier na fatia necessária para o MPL quando executamos a rotina para 500 pacotes, onde vemos um aumento de crescimento no gerenciamento de dados e uma queda no valor necessário para o MPL. Esse é um comportamento que é repetido, pois como dito anteriormente, a geração dos valores foi feita calculando a média aritmética dos resultados de 10 iterações. Devido ao comportamento não esperado do outlier, o experimento foi realizado mais cinco vezes e os resultados apresentaram o mesmo comportamento. Podemos também analisar o crescimento da função nos gráficos nas figuras 8 e 9, onde a área verde representa o tempo usado para leitura de arquivos, a tempo vermelho para o MPL e a azul (Quase invisível) o tempo para o MMQ.

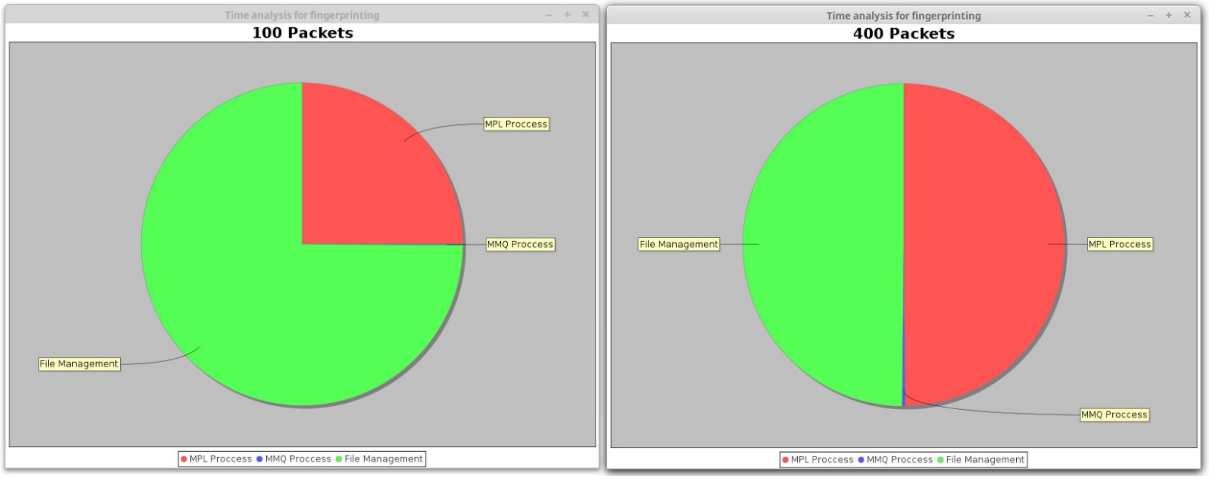


Figura 8 - Gráficos em % do tempo relativo para 100 e 400 pacotes

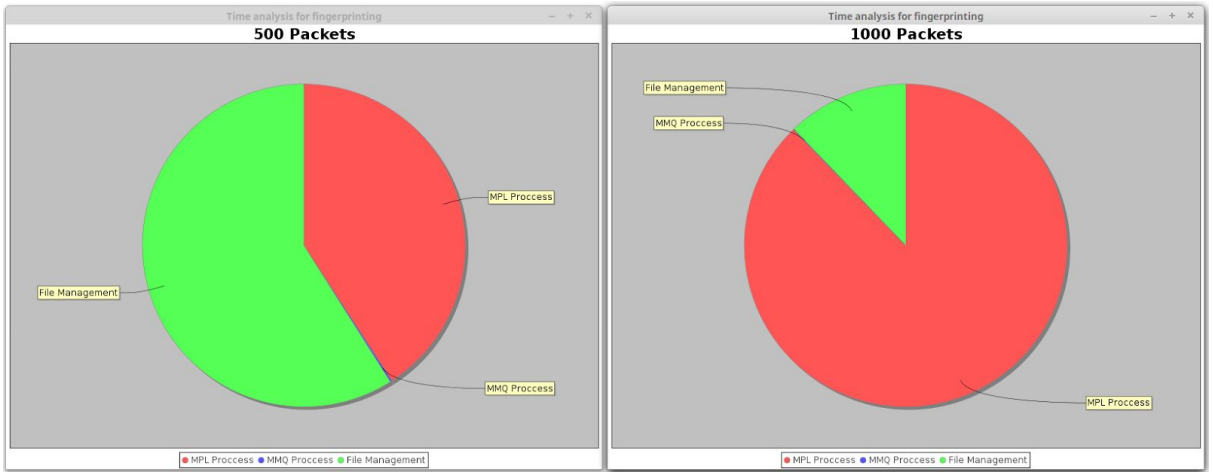


Figura 9 - Gráficos em % do tempo relativo para 500 e 1000 pacotes

O método MMQ por sua vez permanece com uma percentagem baixa não importando o número de pacotes necessários devido ao custo linear do algoritmo utilizado.

Como foi visto anteriormente, o tempo de captura de quadros é a fonte majoritária de tempo, então determinando a quantidade mínima de quadros que precisamos capturar podemos diminuir de forma significativa o tempo necessário para a criação da identidade digital baseada no desvio de relógio. Com poucos pacotes também iremos garantir que o crescimento exponencial do algoritmo Simplex utilizado pelo método estimador MLP também não saia de controle. Nas tabelas 4 e 5, podemos analisar as estimativas realizadas com diferentes número de pacotes para diferentes AP's

	Netcore T Dataset #1			
# Pacotes Examinados	MPL	MMQ	Diferença	Tempo Total
10	-3,90	-104,28	100,38	1.026.28 ms
30	-68,35	-85,80	17,45	3.074,17 ms
50	-74,07	-68,00	-6,07	5.122,35 ms
100	-69,30	-68,14	-1,16	11.678,39 ms
200	-68,11	-68,55	0,44	22.126,93 ms
300	-67,83	-67,54	-0,29	32.879,64 ms
500	-68,35	-67,51	-0,84	53.683,24 ms
1000	-67,46	-67,41	-0,05	108.314,82 ms

Tabela 4 - Valores estimados para o conjunto de dados 1 do AP Netcore

	Netcore T Dataset #2			
# Pacotes Examinados	MPL	MMQ	Diferença	Tempo Total
10	-59,98	-134,83	74,85	1.026,08 ms
30	-59,73	-67,37	7,64	3.382,47 ms
50	-63,47	-67,67	4,20	5.839,21 ms
100	-66,77	-65,84	-0,93	12.702,49 ms
200	-67,45	-66,71	-0,74	27.969,14 ms
300	-66,89	-66,87	-0,02	44.772,79 ms
500	-66,69	-66,80	0,11	70.893,40 ms
1000	-66,91	-66,86	-0,05	133.604,77 ms

Tabela 5 - Valores estimados para o conjunto de dados 2 do AP Netcore

Analisando as tabelas 4 e 5 podemos chegar a conclusão de que precisamos de 100 a 200 quadros, dependendo do conjunto de dados para termos uma convergência dentro de um intervalo suficientemente confiável, resultando em um intervalo médio de 20 a 30 segundos em captura de quadros. Porém o comportamento dos estimadores podem mudar de acordo com o modelo do ponto de acesso, assim como a proximidade do dispositivo fingerprinter entre outros fatores. Então foram capturados e analisados os dados para um segundo AP com modelo diferente dos dados originais.

	Tp-Link Dataset #1			
# Pacotes Examinados	MPL	MMQ	Diferença	Tempo Total
10	-14,64	-38,48	23,84	1.140,54 ms
30	-66,18	-67,74	1,56	3.385,64 ms
50	-67,08	-64,10	-2,98	5.742,15 ms
100	-64,62	-61,31	-3,31	14.352,29 ms
200	-63,66	-62,62	-1,04	28.501,19 ms
300	-62,83	-62,84	0,01	44.349,40 ms
500	-62,81	-62,72	-0,09	68938,93 ms
1000	-62,75	-62,69	-0,06	127633,44 ms

Tabela 6 - Valores estimados para o conjunto de dados 1 do AP Tp-Link

	Tp-Link Dataset #2			
# Pacotes Examinados	MPL	MMQ	Diferença	Tempo Total
10	-44,64	-41,57	-3,07	1.337,24 ms
30	-58,94	-61,94	3,00	3.790,94 ms
50	-61,84	-65,49	3,65	5.947,49 ms
100	-64,18	-64,89	0,71	11.780,71 ms
200	-63,37	-63,04	-0,33	23.258,23 ms
300	-62,99	-63,07	0,08	34.941,52 ms
500	-62,81	-62,93	0,12	62.715,69 ms
1000	-63,05	-62,97	-0,08	126.326,67 ms

Tabela 7 - Valores estimados para o conjunto de dados 2 do AP Tp-Link

Analisando a tabela 6, fomos capazes de encontrar um valor convergente apenas com 300 quadros e no na tabela 7 conseguimos encontrar uma diferença entre estimativas boa apenas com 100 quadros. O que sugere que o número de quadros pode não ser a única métrica usada para determinar o número mínimo de quadros necessários para a convergência entre estimadores.
Também realizamos um experimento onde o tempo médio de envio de pacotes foi reduzido em 10%, 30% e 50% alterando valores no conjunto de dados. Como esperado, o ganho na questão de tempo é linear, onde a redução de tempo necessária para a captura de pacotes total é igual a redução relativa entre a média do intervalo de pacotes.

7. Conclusão e Trabalhos Futuros

Devido ao crescimento de usuários, quantidade de operações envolvendo informações sensíveis, sua natureza de difusão e a facilidade de realizar ataques às redes sem fio tem se tornado um alvo cada vez mais frequente de ataques. Uma das diferentes formas de realizar esse ataque é através da instalação do ponto de acesso falso que se comportam da mesma maneira que os originais e possuem as credenciais necessárias para enganar o dispositivo que está tentando se conectar.

Essa vulnerabilidade gerou métodos que são capazes de criar uma

identidade virtual para APs baseado no desvio de relógio, métodos comprovadamente robustos e eficientes na identificação. Sabendo que o método é capaz de estimar corretamente o desvio de relógio e é capaz de identificar a presença de APs falsos, uma análise temporal sobre a execução do método se mostra o passo natural a ser feito devido a demanda constante por ubiquidade e pela disposição de grande parte dos usuários em abrir mão de segurança pela ubiquidade, nos mostrando os pontos críticos que mais influenciam no tempo necessário para a execução do mesmo.

Para realizar o estudo tivemos que replicar a ferramenta proposta por Jana[2] e a implementação foi feita em quatro fases, contendo as heurísticas, modelos matemáticos e algoritmos usados que suportam os resultados. Com os dados cuidadosamente gerados, pudemos identificar os processos que mais consomem tempo dentro de todos necessários para a criação da identidade digital baseada no desvio de relógio. Como esperado o processo de captura de pacotes é o que consome a fatia majoritária do tempo onde temos uma diminuição linear de tempo quando diminuímos o intervalo no envio entre pacotes. Observamos também que apesar da grandeza exponencial do algoritmo Simplex utilizado como um dos estimadores do desvio de relógio o tempo consumido por ele não sai de controle devido ao número relativamente pequeno de quadros necessários para que consigamos estimar o desvio de relógio de forma a termos uma alta confiabilidade. O MMQ por sua vez se mostrou extremamente eficiente em relação ao tempo necessário para sua execução representando sempre menos que 1% do tempo total necessário, mesmo quando o tempo para captura de pacotes não foi levado em consideração. Operação de leitura de dados dos pacotes capturados consomem uma parte significativa do tempo, o que era de se esperar por se tratar de um processo que envolve operações de I/O.

Como trabalhos futuros, podemos buscar uma heurística para o MPL capaz de identificar o valor estimado sem ter que percorrer todas as possíveis combinações de dois pontos para formação da reta. Como para a metodologia conseguir identificar APs falsos o cálculo do MMQ é necessário podemos talvez utilizar a taxa de convergência dos valores estimados dos métodos como métrica assim implementação do método em outras linguagens e sistemas operacionais

para a verificação de desempenho relativo.

8. Referências

[1] Kurose, James F; Ross, Keith W., Redes de computadores e a Internet: uma abordagem top-down, 5. ed., São Paulo : Addison Wedsadsadsasley, 2010.

[2] Jana, S.; Kasera, S.K., On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews, Mobile Computing, IEEE Transactions on, vol.9, no.3, pp.449,462, March 2010.

[3] **Backports**. Disponível em:
< https://backports.wiki.kernel.org/index.php/Main_Page >. Acesso em 30 de Junho de 2017.

[4] Caio César, TG disponível em < https://github.com/caiocleao/tgcclca > Acessado em 30 de Junho de 2017.

[5] **Wireshark**, disponível em < https://www.wireshark.org/ > Acessado em 30 de Junho de 2017.

[6] **Airmon-ng**, disponível em < https://www.aircrack-ng.org/ > Acessado em 30 de Junho de 2017

[7] **Learn Cisco**, disponível em
< http://www.learncisco.net/courses/icnd-1/wireless-lans/implementing-a-wlan.html > Acessado em 30 de Junho de 2017.

[8] **802.11 Mgmt: Beacon Frame**, disponível em
< https://mrncciew.com/2014/10/08/802-11-mgmt-beacon-frame/ >

[9] **Radiotap header**, disponível em < http://www.radiotap.org/ >, Acessado em 30 de Junho de 2017.

[10] Taebeom Kim; Haemin Park; Hyunchul Jung; Heejo Lee,. Online Detection of Fake Access Points Using Received Signal Strengths, Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th , pp.1,5, 6-9 May 2012

[11] **Linux Mint 17.3 Rosa, Cinnamon**, disponível em
< https://www.linuxmint.com/edition.php?id=204 >, acessado em 1 de Julho de 2017.

[12] **Backports**, disponível em
< https://backports.wiki.kernel.org/index.php/Main_Page >, acessado em 1 de Julho

de 2017.
[13] **Eclipse Neon**, disponível em < http://www.eclipse.org/neon/ >, acessado em 1 de julho de 2017.
[14] **JFreeChart**, disponível em < http://www.jfree.org/jfreechart/ >, acessado em 1 de julho de 2017.
[15] **Linux Wireless**, About Ath9k. Disponível em < https://wireless.wiki.kernel.org/en/users/Drivers/ath9k >. Acesso em 9 de Março de 2016.
[16] Taebeom Kim; Haemin Park; Hyunchul Jung; Heejo Lee,. Online Detection of Fake Access Points Using Received Signal Strengths, Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th , pp.1,5, 6-9 May 2012
[17] Loureiro, A. A. F.; Oliveira, R. A. R.; Moura, T. R. de; Júnior, W. R. P.; Oliveira, L. B. R. de; Moreira, R. A.; Siqueira, R. G.; Rocha, B. P. S.; Ruiz, L. B. Computação Ubíqua Ciente de Contexto: Desafios e Tendências. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2009 - Minicurso, 2009. p.99–149.
[18] T. Kohno, A. Broido and K.C. Claffy, Remote Physical Device Fingerprinting, IEEE Trans. Dependable Secure Computing, vol. 2, no. 2, pp. 93-108, Apr.-June 2005.
[19] HiFiDuino. Oscillator. Disponível em< https://hifiduino.files.wordpress.com>. Acesso em 15 de Maio de 2016.
[20] IEEE Guide for Measurement of Environmental Sensitivities of Standard Frequency Generators, IEEE Standards Coordinating Committee 27-SCC27- on Time and Frequency, 1995.