UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uma Abordagem de Segurança do Sistema Asterisk em Plataformas Embarcadas Usando o Protocolo SIP

Dissertação de Mestrado

Toniclay Andrade Nogueira

São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Toniclay Andrade Nogueira

Uma Abordagem de Segurança do Sistema Asterisk em Plataformas Embarcadas Usando o Protocolo SIP

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Admilson de Ribamar Lima Ribeiro
Coorientador(a): Edward David Moreno Ordonez

São Cristóvão – Sergipe

2018

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE

Nogueira, Toniclay Andrade
N778a Uma abordagem de segurança do Sistema Asterisk em plataformas embarcadas usando o protocolo SIP / Toniclay Andrade Nogueira ; orientador Admilson de Ribamar Lima Ribeiro. - São Cristóvão, 2018.
83 f. : il.

Dissertação (mestrado em Ciência da Computação) – Universidade Federal de Sergipe, 2018.

1. Computação. 2. Asterisk (Programa de computador). 3. Ciberterrorismo. 4. Rede de computador - Protocolos. I. Ribeiro, Admilson de Ribamar Lima orient. II. Título.

CDU 004.056

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do Curso de Mestrado em Ciência da Computação-UFS.
Candidato: TONICLAY ANDRADE NOGUEIRA

Em 30 dias do mês de novembro do ano de dois mil e dezoito, com início às 13h30min, realizou-se na Sala de Seminário do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Toniclay Andrade Nogueira**, que desenvolveu o trabalho intitulado: ***" Uma Abordagem de Segurança do Sistema Asterisk em Plataformas Embarcadas Usando o Protocolo SIP"***, sob a orientação do Prof. Dr. **Admilson de Ribamar Lima Ribeiro**. A Sessão foi presidida pelo Prof. Dr. **Admilson de Ribamar Lima Ribeiro** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **Edward David Moreno Ordonez** (PROCC/UFS); Prof. Dr. **Ricardo José Paiva de Britto Salgueiro** (DCOMP/UFS) e, em seguida, ao Prof. **José Augusto Andrade Filho** (**IFS**). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) aprovado "*(aprovado/reprovado)*" sem "*(com/sem)*" ressalvas. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária "*Prof. José Aloísio de Campos*", 30 de novembro de 2018.

Prof. Dr.Admilson de Ribamar Lima Ribeiro
(PROCC/UFS)
Presidente

Prof. Dr. Edward David Moreno Ordonez
(PROCC/UFS)
Examinador Interno

Prof. Dr. Ricardo José Paiva de Britto Salgueiro
(DCOMP/ UFS)
Examinador Interno

Prof. Dr. José Augusto Andrade Filho
(IFS)
Examinador Externo

Toniclay Andrade Nogueira
Candidato

Dedico essa dissertação de Mestrado primeiramente a Deus, por me dar força na hora em que mais precisei, e ao meu irmão Roberto Nogueira, "In Memoriam", que sempre me apoiou nesta caminhada. Ao meu pai Antônio Nogueira, minha mãe Gisélia Nogueira e minha irmã Hipácia Nogueira, pelo apoio familiar tão importante nos momentos difíceis. A minha querida e amada filha Giovanna Nogueira, por ver um grande futuro pela frente. Aos meus queridos sobrinhos Danilo, Luiza e Marianna, ao meu cunhado Charles e cunhada Tereza Nogueira. Aos professores Admilson de Ribamar Lima Ribeiro e Edward David Moreno pela amizade e paciência durante todo esse trajeto, e aos amigos que me deram o apoio necessário para chegar até aqui.

Resumo

A preocupação com a segurança nas redes *Internet Protocol* (IP) vem crescendo exponencialmente. Medidas legais, como penas severas para criminosos virtuais, já são uma realidade. Vários estudos estão sendo realizados com intuito de explorar os problemas de segurança relacionados à VoIP.

Por outro lado, dispositivos embarcados se mostram cada vez mais eficientes com sistemas complexos e que exigem um bom desempenho. O *software* livre voz sobre IP Asterisk tem como finalidade ser uma central telefônica, uma alternativa viável para ser utilizada em dispositivos embarcados sendo possível reduzir custos e maximizar resultados.

Esta dissertação de mestrado realiza uma abordagem de segurança Usando o protocolo SIP do Asterisk em plataformas embarcadas. Em paralelo, também objetiva monitorar o consumo de memória RAM, processamento e consumo de energia elétrica nos momentos de três ataques de segurança do tipo Autenticação, Man-in- the-middle e de Negação de serviço DoS.

Os resultados mostraram que o dispositivo Raspberry Pi 3 suporta de forma satisfatória os ataques de Autenticação e Man-in- the-middle , mas o sistema Asterisk, no ataque de Negação de serviço, não consegue suportar o ataque a partir de quinhentos mil pacotes enviados pelo atacante, ficando sem possibilidade de realizar chamadas tendo seu funcionamento totalmente neutralizado. Com relação ao consumo de energia notasse que o Raspberry Pi 3 em sua voltagem tende a ficar em um patamar médio de 5,19v e a Current variando entre -600,93mA a -832,41mA e o Power variando entre -3132,40mV a -4314,78mV tendo com parâmetro a quantidade de pacotes enviados pelo atacante de 0 a 25.000.000.

Palavras-chave: Ataque DoS, Asterisk, VoIP, Eficiência, Dispositivo Embarcados, protocolo.

Abstract

The concern for security in Internet Protocol (IP) networks has been growing exponentially. Legal measures, such as severe penalties for cybercriminals, are already a reality. Several studies are being conducted to explore the security issues related to VoIP.

On the other hand, embedded devices are increasingly efficient with complex and demanding systems. Free voice over IP software Asterisk aims to be a central telephone exchange, a viable alternative to be used in embedded devices being possible to reduce costs and maximize results.

This master's dissertation takes a security approach using Asterisk's SIP protocol on embedded platforms. In parallel, it also aims to monitor the consumption of RAM, processing and power consumption in the moments of three Authentication, Man-in-the-middle and DoS Denial of Service security attacks.

The results showed that the Raspberry Pi 3 device satisfactorily supports Authentication and Man-in-the-middle attacks, but the Asterisk system, in the Denial of Service attack, can not withstand the attack from five hundred thousand packets sent by the attacker, being unable to make calls having its operation totally neutralized. Regarding energy consumption, note that the Raspberry Pi 3 at its voltage tends to stay at an average level of 5.19v and Current ranging from -600.93mA to -832.41mA and Power ranging from -3132.40mV to -4314.78mV having parameterized the amount of packets sent by the attacker from 0 to 25,000,000.

Keywords: Attack DoS, Asterisk, VoIP, Efficiency, Embedded Device, protocol.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

APIs	Application Programming Interface
Arp	Address Resolution Protocol
CADC	Air Data Central Computer
DNS	Domain Name System
DoS	Denial Of Service
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPsec	Internet Protocol Security
PABX	Private Automatic Branch Exchange
PSTN	Public Switched Telephone Network
PBX	Private Branch Exchange
RAM	Random Access Memory
RSA	Rivest-Shamir-Adleman
RTP	Real Time Protocol
SIP	Session Initiation Protocol
SIPp	Self Invested Personal Pension
Snort	É um software livre de detecção de intrusão para rede
SQL	Structured Query Language
TCP	Protocolo de Controle de Transmissão
TLS	Transport Layer Security
UAC	User Agent Client
UDP	User Datagram Protocol
URA	Unidades de Respostas Audíveis

VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity

Sumário

1 Introdução

A tecnologia *Voice Over Internet Protocol* (VoIP) consiste na integração dos serviços das áreas de telecomunicações com os serviços de redes de computadores. Assim, torna-se possível a digitalização e codificação do sinal da voz, transformando-o a voz em pacotes de dados *Internet Protocol* (IP) para a realização de comunicação em uma rede que utilize os protocolos TCP/IP.

O VoIP existe desde 1995, apresentado pelo *software Internet Phone*, que foi desenvolvido pela empresa Vocaltech Communications. Porém, em 2003, o Skype possibilitou demonstrar ao mercado e aos consumidores a potencialidade dos aplicativos de telefonia IP (KUHN; WALSH; FRIES, 2005).

Esse novo conceito (VoIP) permite a redução dos custos de instalação, de manutenção e de gerência de redes paralelas, cada uma dedicada ao suporte de um único serviço. Ela possibilita a redução de custos, criando assim um novo conceito de telefonia, (SITOLINO, 1999) já que necessita de equipamentos, técnicas e de recursos humanos específicos (COLHER et al., 2005).

Os sistemas embarcados utilizam plataformas de *hardware*, uma vez que são dirigidos por *softwares* e diversas implementações de processadores que podem ser utilizados, o que implica uma forte redução de custos. Alguns problemas de confiabilidade são encontrados em dispositivos embarcados. Por exemplo: como não pode ser desligado com segurança para reparos, o sistema deve executar sempre, assim como modos de desempenho reduzidos não são admissíveis e o ambiente tende a apresentar perdas se for desligado (AKYILDIZ et al., 2002).

Quando se fala de VoIP, pode-se integrar vários blocos de IP diferentes a partir de fontes. Alguns IPs podem lidar com criptografia ou decodificação, enquanto outros blocos gerenciam as operações que se referem como IP de segurança *Session Initiation Protocol* (SIP). Todo IP deve atender a uma especificação de desempenho, mantendo-se dentro dos potenciais de área e tempo de colocação no mercado. O SIP também deve ser seguro sob vários modelos de ataque, uma vez que um usuário mal-intencionado poderá tentar extrair informações do SIP de várias maneiras.

Medidas legais, como penas severas para criminosos virtuais, já são uma realidade. Os administradores de redes mais do que nunca estão implantando soluções como detecção de intrusos, *firewalls* com filtros avançados, antivírus, chaves de criptografia, proxy, entre outros. Vários estudos estão sendo realizados com intuito de explorar os problemas de segurança relacionados a VoIP. O uso de protocolos de texto, a falta de autenticação e a complexidade da implantação de segurança end-to-end sólida são apenas alguns exemplos de como as redes VoIP são suscetíveis a diversos ataques.

O atacante pode examinar fisicamente o SIP, tentar obter informações seguras durante a operação, buscar informação do ambiente de desenvolvimento ou comunicação do designer SIP ou integrador. Por isso, ameaças são considerações de segurança que devem ser levadas em consideração quando trabalhando em um projeto de configuração de utilizando-se VoIP.

De acordo com (STAPKO, 2011), segurança de computadores consiste em proteger informações pessoais ou confidenciais e/ou recursos computacionais de indivíduos ou organizações que poderiam deliberadamente destruir ou se utilizar de tais informações para fins maliciosos. O chamado estado da arte em segurança na telefonia VoIP envolve a encriptação do áudio entre os dois pontos usados, interoperabilidade entre os fabricantes, servidores, criptografia indecifráveis e gerenciamento centralizado sem a necessidade de configuração (WILLIAM; STALLINGS, 2015).

Segundo (BARR; REILLY, 1999; CARRO; WAGNER, 2003), sistemas embarcados devem ser confiáveis, uma vez que falhas podem comprometer esta única função e por talvez ser difícil sua substituição remotamente.

A segurança em sistemas embarcados nem sempre foi levada em conta, uma vez que, inicialmente, a maioria deles operavam embutidos em sistemas sem conectividade com a Internet. Nesse sentido, pode-se justificar a importância da segurança da informação em novos dispositivos que estão sendo criados a partir de dispositivos embarcados, já que eles cada vez mais estão ficando presentes na vida do dia a dia. Com certeza, a comunicação entre dispositivos dará uma reviravolta na comunicação mundial, proporcionando eficiência nas comunicações VoIP.

1.1 Problemática e Hipótese

A segurança em dispositivos embarcados é uma preocupação real, tendo em vista que vários sistemas estão sendo criados a cada dia. A preocupação com a segurança é fundamental para que se possa ter a confiabilidade destes dispositivos. Segundo (JONES, 2007; MCGRAW, 2006), segurança de *software* é um tema cada vez mais relevante na medida em que muitos ataques veem explorando as vulnerabilidades deste *software*.

Para (ALHAZMI; MALAIYA; RAY, 2007), a segurança de *software* torna-se um tema central na segurança de sistemas computacionais como um todo. Já para (BARR; REILLY,

1999) (CARRO; WAGNER, 2003; MARWEDEL, 2011), os sistemas embarcados são sistemas especializados, diferentemente de um elemento computacional convencional. Isso, aliado ao fato de que eles são comumente inseridos em outros sistemas, faz com que sistemas embarcados tenham suas dimensões reduzidas. Tal redimensionamento, aliado à necessidade de redução de custos, por sua vez, torna os sistemas embarcados limitados de recursos computacionais (HAMACHER et al., 2012).

Tendo em vista a implantação do Asterisk em sistemas embarcados para reduzir despesa na área de telefonia com as características de uma central telefônica, o desafio está em garantir a segurança das vulnerabilidades encontradas. Faz-se necessário, então, prover a segurança desse dispositivo contra os ataques de invasores, analisando os dados do sistema para descobrir qual foi o tipo de ataque que ocorreu. Logo em seguida, executar ações que serão planejadas e baseadas no tipo de ataque, com intuito de mitigar os possíveis danos causados ao sistema embarcado.

1.2 Objetivos

O objetivo principal deste trabalho é realizar um estudo sobre segurança do sistemas Asterisk executando em plataformas embarcadas de baixo custo quando submetido a três ataques de segurança (ataques de Autenticação, Man-in-the-middle e Negação de Serviço), assim como analisar o consumo de energia, memória e uso de processamento para o sistema de comunicação de voz quando submetido a esses ataques utilizando uma placa Raspberry Pi 3.

1.3 Justificativa

O Asterisk uma solução VoIP híbrido de plataforma aberta e PABX com ótima relação custo beneficio.

Alguns dos benefícios que o Asterisk pode trazer às empresas:

Integrar empresas de forma a efetuar ligações custo zero.

Nos *Call centers* o Asterisk pode ser integrado a sistemas CRM, facilitando no fornecimento de informações a respeito de determinados tipos de clientes, fornecedores e etc...

Serviços de*voicemail,* fax e e-mail, tudo em uma única interface de gerencia web, ou seja, comunicação integrada.

Isso é só uma amostra das ferramentas que o Asterisk possui, pelo fato de ser uma plataforma livre, várias empresas estão visando desenvolver ferramentas personalizadas com o intuito de facilitar e gerenciar suas estruturas de comunicações, trazendo maior produtividade.

Ataque a servidores é um coisa rotineira, e para isto, temos sempre que monitorar e manter as políticas de segurança conforme cada serviço que roda em nossos servidores. Sempre se atualizando. Nos servidores VoIP, Asterisk, não é diferente.

A justificativa é baseada na crescente demanda em comunicação de voz e dados, bem como em tornar toda e qualquer comunicação confiável e segura, independente de ser dados ou voz, além do fato de ataques ocorrerem com frequência em redes de comunicação.

Este trabalho está pautado em realizar uma abordagem de segurança, analisar os riscos e vulnerabilidades em um dispositivo embarcado Raspberry Pi 3 com Asterisk e analisar o consumo de energia, memoria e CPU em relação aos ataques de Autenticação, Man-in-the-middle e Negação de Serviço em sistemas que usa o Asterisk.

1.4 Metodologia

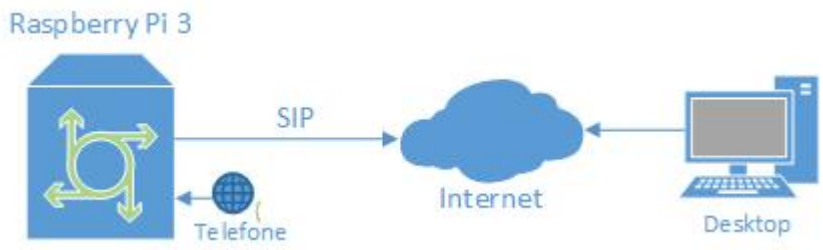
A partir de trabalhos obtidos em bases reconhecidas foi realizada uma revisão bibliográfica visando a identificar os principais ataques em dispositivos embarcados, bem como ao protocolo SIP, no intuito de verificar o quanto se tinha de informação sobre os ataques a dispositivos embarcados (ataque,SIP).

Ao aplicar esta metodologia, nota-se a importância de demonstrar como os ataques a dispositivos embarcados são realizados, assim como avaliar o estado de consumo de energia no dispositivo embarcado, consumo de CPU e memória. Dessa forma, faz-se necessário seguir uma sequência para que seja possível entender os ataques de Autenticação, Man-in-middle e Ataque Negação de Serviço DOS, conforme seguinte:

- Realizar levantamento dos ataques a sistemas de comunicação;
- Realizar levantamento das ferramentas e materiais necessários para a simulação dos ataques mais significativos em dispositivos embarcados;
- Realizar revisão de literatura do protocolo SIP;
- Analisar as vulnerabilidades do protocolo SIP;
- Analisar os ataques com relação ao consumo de energia, CPU e memória
- Escrever a dissertação e apresentar os resultados da análise dos ataques no dispositivo embarcado Raspberry Pi 3 com Asterisks.

O estudo se inicia com a demonstração dos ataques de Autenticação, Man-in-middle e Ataque Negação de Serviço DOS, bem como com a verificação do estado de consumo de energia no dispositivo embarcado, consumo de CPU e memória, utilizando os *software* Kali Linux e Zabbix. Foi utilizado o projeto de cenário de teste conforme Figura 1.

Figura 1 – Projeto de cenário teste



Fonte: Autoria própria.

Ao término da demonstração dos ataques e coleta das informações obtidas para o estado de consumo de energia, consumo de CPU e memória, No capitulo 6 teremos as conclusões dos resultados obtidos.

1.5 Organização da Dissertação

Para facilitar a navegação e melhor entendimento, este documento está estruturado em seis (6) capítulos, que são:

Na Introdução capitulo 1, apresenta o problema, justificativa e o que foi proposto nesta dissertação. No capítulo 2 da Fundamentação Teórica, são abordados os principais temas do trabalho e tecnologias para contextualizar o leitor, com o foco em VoIP, Sistemas Embarcados, Asterisk, protocolo SIP, os principais tipos de ataques ao protocolo SIP e segurança ao protocolo SIP. No capítulo 3 sobre os Trabalhos Correlatos, são apresentados os trabalhos relacionados ao problema descrito. No capítulo 4, são descritos o método de coleta dos dados de forma detalhada. No capítulo 5 de Experimento e Resultados, é apresentado como foram realizados os experimentos e seus resultados. Por fim, no capítulo 6, da conclusão, são abordadas as considerações dos resultados encontrados, bem como os trabalhos futuros a serem realizados tendo como base essa dissertação de mestrado.

2

Fundamentação Teórica

2.1 VoIP

A qualidade das redes baseadas no Internet Protocol, conhecida como redes IP, tornou possível a navegação de diferentes tipos de mídia (áudio, vídeo, voz e imagens) através de uma rede que foi projetada inicialmente para o tráfego de dados.

Segundo Raake (2006), Walker e Hicks (2004), VoIP é uma tecnologia que faz a transmissão de voz em uma rede de pacotes IP. O processo da transmissão consiste basicamente em transformar a voz analógica em digital, dividindo-a em vários pacotes e transportá-los sobre a rede IP. Após alcançar o destino, os pacotes são reorganizados e convertidos para o sistema analógico novamente. O processo está cada vez mais atual com *softwares* que possuem a tecnologia, como Facebook, Messenger, Skype, Viber e WhatsApp.

O Sistema VoIP existe desde o ano de 2015, através do *software Intenet Phone* que foi desenvolvido pela (VOLCATEC, 2016). Contudo foi com o surgimento do SKYPE, no ano de 2003, que os aplicativos de telefonia IP começaram a chamar atenção das empresas e de consumidores.

Segundo a Volcatec (2016), havia a necessidade de pessoas se comunicarem, já que, naquela época, houve uma imigração de judeus da antiga URSS para Israel. Eles eram pessoas de pouco poder aquisitivo, mas que necessitavam se comunicar com suas famílias na Rússia. Acontece que a telefonia normal tinha um custo elevado, então o mercado encontrou uma forma de viabilizar um novo negócio.

Na Figura 2, podemos observar o funcionamento de uma aplicação VoIP, na qual o áudio analógico é convertido em digital e agrupado em pacotes que são transmitidos para a rede IP através do protocolo *Real Time Protocol* (RTP). Após chegar ao receptor, os pacotes são organizados e depois reproduzidos.

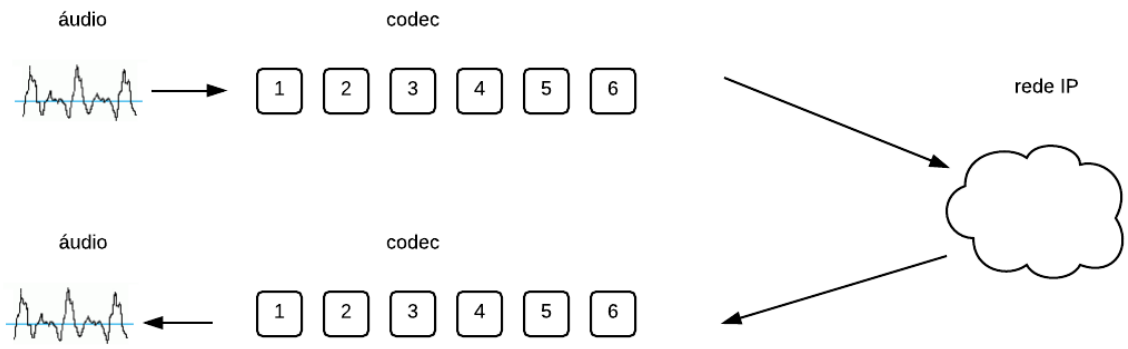


Figura 2 – Cenário do funcionamento ideal da aplicação VoIP.

Fonte: (RAFAEL SEIDI SHIGUEOKA, 2016)

Para podermos transmitir dados de voz são necessários componentes como codificador/decodificador, protocolos TCP/IP, VoIP, *Gateways* VoIP, Roteadores, Telefones IP e *Softphones*.

(KELLER, 2011), aponta as principais vantagens e desvantagens do VoIP. Para um melhor entendimento, vamos comentar algumas delas.

Vantagens:

- Diminuição dos custos das ligações;
- Diminuição dos custos dos equipamentos de rede;
- Infraestrutura Única;
- Facilidade de implantação devido à larga utilização do protocolo IP;
- Integração entre voz e dados e novas aplicações;
- Melhor aproveitamento da largura de banda;
- Mobilidade;
- Mercado, lucros e empregos.

Desvantagens:

- Falta de padronização de protocolos;
- Confiabilidade e disponibilidade da rede;

- Segurança;
- Qualidade de Voz.

Segundo (GRECCO, 2004), a principal função de um sistema de comunicação é permitir que uma mensagem seja gerada por uma fonte de informação e que possa ser entregue corretamente ao seu destino. Para que as funções sejam executadas de forma adequada, as camadas devem obedecer às regras conhecidas e que estejam de acordo com as máquinas que participam da rede.

Essas regras, que são conhecidas como protocolos, definem o modo de operação do sistema, estabelecendo procedimentos que devem ser tomados a cada linha e que determinam o que deve ser feito a cada momento.

No sistema de telefonia, em sua inicialização, faz-se necessário estabelecer, controlar e encerrar sessões entre usuários, o que chamamos de procedimento de inicialização. Um sistema convencional possui dois tipos de sinalização: o dentro da faixa e o fora da faixa.

O sistema dentro da faixa possui esse nome por se utilizar da mesma faixa de frequência do sinal de voz que é composto de um conjunto de tons audíveis. O sistema fora da faixa, que é conhecido por sinalização em canal comum, foi criado para aumentar a eficiência do sistema de telefonia (International Telecommunications Union, 1993) e entre suas funções estão estabelecer, configurar, monitorar, rotear e encerrar as chamadas da telefonia convenciona PSTN.

Os principais modelos de protocolos da rede IP são:

H.323 – Que foi desenvolvido pelo ITU-T em 1996, tendo resolvido diversos melhoramentos e revisões até o ano de 2006, e que possui como principais protocolos o H.225, Q.931, H.245, G.7xx, RTP, RTCP, T.12x, H.450, H.26x, H.246 e H.235.

RTP – Um protocolo que possui aspectos de Qualidade de Serviço (QoS), diferenciando-se dos outros tipos de tráfego da rede e estabelecendo características que devem ser suportadas por protocolos de transporte em tempo real.

RTCP – Um protocolo que possui um protocolo próprio de controle que se chama RTCP (RTP Control Protocol), responsável por cuidar da sincronização, resposta e interface com o usuário. Esse protocolo é baseado na transmissão periódica de pacotes de controle entre seus principais participantes de sessão, através do mesmo mecanismo utilizado para distribuição de pacotes de dados.

SCTP – Protocolo que está na camada de transporte do TCP/IP e que possui dois protocolos de transporte destinados a usos distintos. Ele permite que a flexibilidade de uma comunicação rápida e confiável recorra ao UDP. Esse é um protocolo novo que foi desenvolvido para superar as limitações impostas pelo TCP.

TCP/IP – Esse protocolo está atrelado ao desenvolvimento da Internet no ano de 1950.

Seu modelo de referência surgiu como uma descrição de um conjunto de protocolos que já era encontrado em operações práticas na ARPANET.

IP – Protocolo este que mantém a inter-rede unida. Tem a função de interligar redes transportando da melhor forma possível os datagramas através da rede. Um elemento na rede IP é identificado por um endereço IP único com 32 *bits*.

TCP – É um protocolo da camada de transporte que oferece um fluxo de bytes fim-a-fim com confiança de uma inter-rede não-confiável. O TCP associa cada fluxo de dados a um par de portas que forma uma conexão ponto a ponto entre máquinas de origem e destino.

UDP – É um protocolo de transporte não-confiável e sem conexão com o TCP/IP, capaz de oferecer um meio para as aplicações enviarem datagramas IP encapsulados sem a necessidade de estabelecer uma conexão (TANENBAUM, 2003).

SIP – É um protocolo de sinalização da camada de aplicação que é utilizado para iniciar, modificar e finalizar uma sessão interativa de multimídia entre usuários. Algumas comunidades na Internet consideram esse protocolo muito melhor do que o protocolo H.323, que é muito extenso, completo e inflexível.

2.2 Sistemas Embarcados

Alguns dados pesquisados em alta tecnologia mostram que mais de 90% dos micro-computadores que são fabricados no mundo são destinados a máquinas que não são de fato computadores, como por exemplo: celulares, automóveis, aparelhos de DVD, entre outros.

Segundo (REIS, 2004), o que vem a diferenciar o conjunto de dispositivos de um computador é o projeto baseado em um conjunto dedicado e especialista, constituído por *Hardware*, *Software* e Periféricos, ou seja, Sistemas embarcados.

Segundo (CUNHA, 2007), O termo embarcado significa que uma unidade de micro-processamento está encapsulada e a serviço de uma tarefa específica. “Colocar capacidade computacional dentro de um circuito integrado, equipamento ou sistema”.

Em 1970, o mundo ganhava mais um marco simbólico, o Air Data Central Computer (CADC), que foi o primeiro sistema baseado em microprocessadores e que tinha como função o controle de uma central de voo. Já nos anos 80, o mercado estava com circuitos que combinavam microprocessador, RAM e dispositivos de Entrada/Saída. Eles eram mais acessíveis, porém não muito flexíveis como os computadores convencionais.

Para (BALL, 2005), o sistema é classificado como embarcado quando ele é dedicado a uma única tarefa e interage continuamente com o ambiente à sua volta, por meio de atuadores e sensores. Na Figura 3, demonstramos um diagrama básico de um sistema embarcado dotado de um micro controlador e uma variável “ambiente” como temperatura e umidade de uma sala.

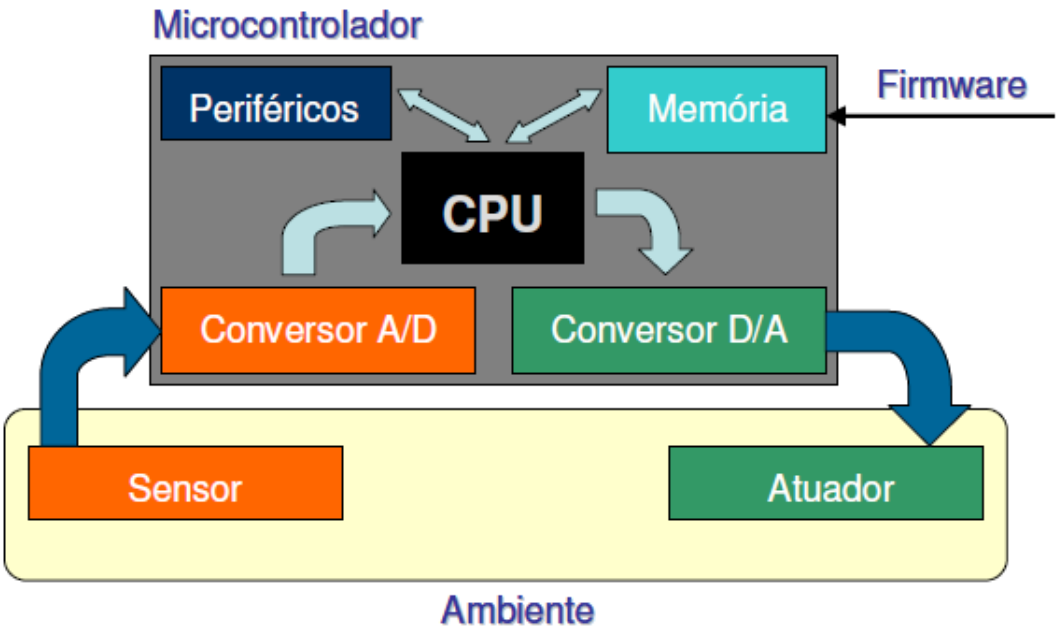


Figura 3 – Diagrama básico de um sistema embarcado dotado de um micro controlador monitorando o ambiente.

Fonte: (CHASE, 2007)

No artigo (SIQUEIRA et al., 2006), o autor comenta sobre o uso de sistemas embarcados em aplicações críticas. Aplicações críticas são aquela em que os riscos associados aos perigos envolvidos são considerados inaceitáveis e precisam ser tratados.

O sistema embarcado comumente é uma solução formada de microcontrolador e *software (firmware)* dedicados e específicos para desempenhar as funções operacionais de um equipamento para o qual foi projetado.

2.3 Raspberry Pi 3

Em 2006, Eben Upton, Rob Mullins, Jack Lang e Alan Mycroft resolveram criar um computador pequeno e acessível para crianças no laboratório da University of Cambridge, na Inglaterra. Eben Upton, diretor de Estudos em Ciência da Computação na universidade, havia observado que os alunos que se candidatavam a participar do laboratório de Ciências da Computação da Universidade não apresentavam as mesmas habilidades e domínio das máquinas que tinham os alunos da década de 1990.

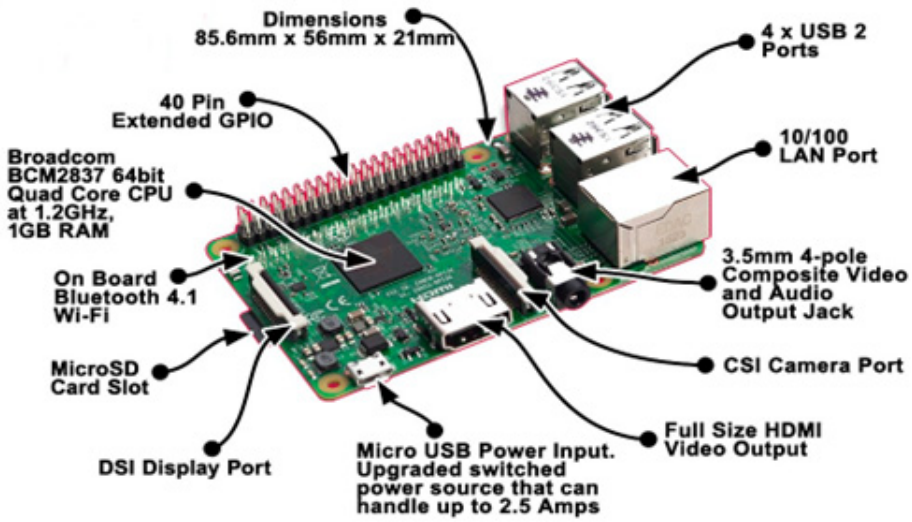
Naquela época, jovens de 17 anos que desejavam cursar essas disciplinas já chegavam à faculdade com conhecimento de linguagens de programação e do funcionamento do *hardware;* alguns até trabalhavam com a linguagem Assembly1.

O primeiro protótipo do pequeno computador surgiu na mesa da cozinha da casa de Eben. Ele e seus amigos começaram a soldar, em uma protoboard2 com um chip Atmel3 e alguns outros chips baratos de microcontroladores4 para monitorar um aparelho de TV. O projeto contava com apenas 512 K de memória RAM, atingindo poucos MIPS5 de processamento.

O Raspberry Pi 3 se trata de um dispositivo embarcado que possui processamento considerável, devido ao processador Broadcom BCM2837 de 64 *bits* e *clock* de 1.2GHz. Com *Wifi* e *Bluetooth* 4.1 integrados, ele evita que o usuário compre adaptadores adicionais, o que deixa as portas USB livres para serem utilizadas por outras aplicações.

A placa possui 1G de memória RAM, adaptador para cartão microSD e GPU Videocore IV 3D. É uma placa que possui compatibilidade com o modelo anterior, Raspberry Pi 2, não só em termos de aplicações como também em seu *layout*, já que os seus conectores foram mantidos na mesma posição, assim como o tamanho e a perfuração da placa. Com a Raspberry Pi 3, é possível executar diversas distribuições Linux como o Raspbian e Ubuntu, além do Windows 10 IoT. A Figura 4 ilustra a placa Raspberry Pi 3.

Figura 4 – Raspberry Pi 3.



Fonte: Adaptado de Zapals (2018).

2.4 Asterisk

O Asterisk, segundo (DASTERISK, 2016), é um *software* que emula funcionalidades de sistema de comunicação de voz. Ele foi criado e vem se aprimorando na mesma metodologia do Linux, por base de usuários em constante crescimento. O *software* foi desenvolvido inicialmente por Mark Spencer com o objetivo de criar uma PABX sobre IP que satisfaz as necessidades das empresas.O código do Asterisk é aberto, podendo ser manipulado por qualquer usuário, o

que possibilita uma infinidade de configurações e a realização de mudanças de forma rápida. O grande conjunto de opções de configurações e o código aberto permite um alto grau de adaptação às necessidades das empresas e usuários.

O Asterisk é utilizado em conjunto com o VoIP e, aliado a uma Internet rápida, permite uma conexão praticamente sem limites, possibilitando que empresas se comuniquem com seus escritórios ou funcionários em diversas partes do mundo com um custo baixo e com qualidade de serviço. Entre algumas funcionalidades que estão presentes em um sistema de comunicação, o Asterisk também suporta chamadas em espera, identificação do usuário e redirecionamento de chamadas. Podemos destacar outros recursos que não são oferecidos pelas operadoras, mas que o Asterisk oferece:

- Tronqueamento – Uma funcionalidade que permite acesso de vários usuários a um número irrestrito de linhas de comunicação. O tronqueamento permite ainda um compartilhamento tanto de acesso à rede telefônica pública quanto para o acesso a canais de comunicação privados;
- Distribuição de chamadas – para receber uma chamada, o Asterisk pode se utilizar de alguns atributos predefinidos e encaminhar as chamadas com mais rapidez ao seu destino, podendo ainda encaminhar uma chamada para um único usuário ou para um grupo de extensões que tocarão em uma ordem predefinida até que ela seja atendida.
- Gravação do histórico – Possibilita o armazenamento detalhado de cada chamada realizada com o mês, dia e horário da ligação, assim como o tempo da ligação, origem da ligação etc;
- Gravação de chamadas – É possível através do Asterisk a gravação de toda conversa tanto recebida como discada, possibilitando que uma empresa possa verificar o tipo de atendimento que seu funcionário está prestando aos seus usuários;
- *Interactive Voice Response* – Um recurso amplamente utilizado em call centers, o que possibilita robotizar o atendimento com voz, levando o usuário às informações ou ramais desejados;
- Correio de Voz – Podem ser realizadas configurações individualizadas. É possível ainda a notificação de novas mensagens no correio de voz via e-mail, podendo anexar sua própria mensagem de voz.

(GROSS, 2011), em seu livro "VoIP com Asterisk", coloca que o Asterisk, em sua arquitetura, foi desenvolvido para máxima flexibilidade. Suas APIs específicas são determinadas em volta de um avançado núcleo, que é um sistema de comunicação. Sendo assim, o núcleo faz a interação entres as APIs do sistema para que seja possível executar de forma simultânea e conectada às funções que se espera do *software.*

2.5 Protocolo *Session Initiation Protocol* (SIP)

O SIP foi desenvolvido a fim de facilitar a implementação dos aspectos básicos de uma sessão, que é um processo nada trivial. Hoje é utilizado em escala mundial e é também um forte "concorrente" do H.323. (BARBOSA, 2006) define SIP como um protocolo que sinaliza sessões cliente-servidor destacando presença e mobilidade, tendo como primitivas inicialização, modificação e finalização de sessões.

Segundo (DEFSIP, 2006), Juntamente com RTP (*Real-time Transport Protocol*) , RTSP (*Real Time Streaming Protocol*) e o SDP (*Session Description Protocol*), o SIP estabelece uma arquitetura multimídia completa, provendo serviços completos ao usuário. Por (GROSS, 2011), o protocolo SIP se parece com o protocolo HTTP, sendo também um protocolo que se baseia em texto e que funciona como cliente/servidor, implementando métodos de requisição e resposta na comunicação. Segundo (KELLER, 2011), o SIP é um protocolo de sinalização simples, modular e escalável de realizar chamadas de voz. Vale ressaltar ainda que esse protocolo é o único módulo projetado para interoperar bem com as aplicações da Internet.

O SIP deve proporcionar serviços de gerenciamento de participantes de uma seção. Segundo (CUERVO et al., 2000), por ter essa capacidade de trabalhar em conjunto com outros protocolos, ele permite que haja a integração com a telefonia pública, permitindo não só a ligação entre ramais IP, como também para telefones de rede pública. Os aspectos de segurança do SIP fornecem particularidades que incluem prevenção de negação do serviço, autenticação, integridade e serviços privados e encriptação. Na Figura 5, temos uma visão geral do protocolo SIP com o estabelecimento das sessões e uma arquitetura formada por agentes de usuários e servidores SIP.

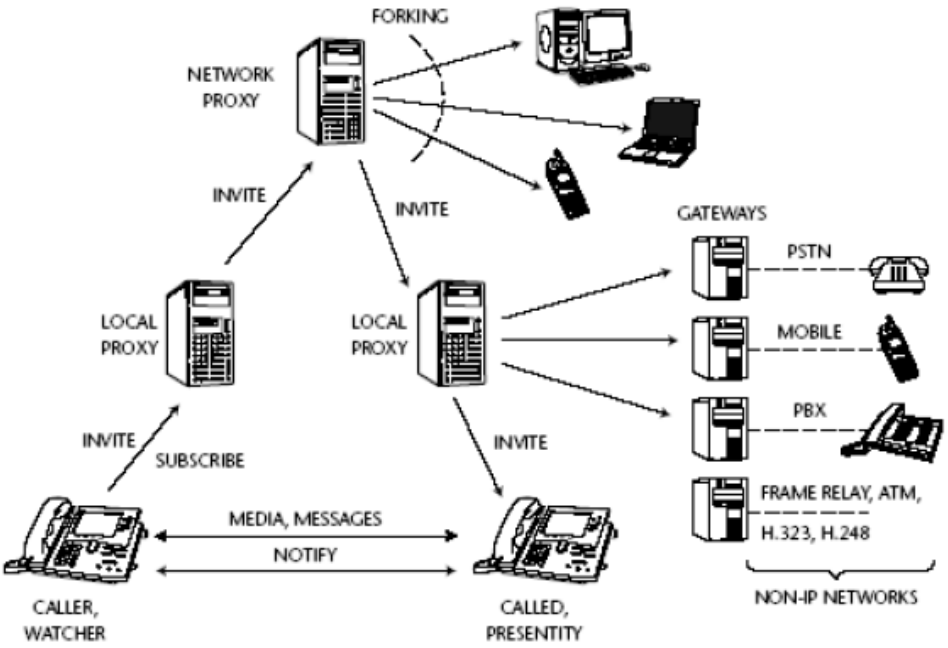


Figura 5 – Visão geral do SIP

Fonte: (SINNREICH, 2006)

Para (MINOLI, 2002), o SIP promete ser um protocolo das redes de comunicação convergentes. O seu desenvolvimento teve como foco os aspectos de intratabilidade com protocolos existentes da IETF (Internet Engineering Task Force), escalabilidade, simplicidade, rapidez, mobilidade e facilidade na implementação das características e serviços.

2.5.1 Tipos de ataques aos Protocolos SIP

2.5.1.1 Man-in-the-middle (Invasor no meio da negociação SIP)

Para esse ataque, o invasor pode utilizar duas técnicas: envenenamento da tabela ARP ou clonagem do DNS. Com qualquer uma delas, se consegue a permissão para estar entre o servidor SIP e o Agente Usuário. Com esse tipo de ataque, o intruso não precisa necessariamente conhecer usernames e passwords válidos; basta rotear o tráfego entre servidor e cliente e depois agir interceptando os pacotes, impedindo-os de chegar ao seu destino real, que é o servidor SIP. (NAKAMURA; GEUS, 2007)

2.5.1.2 Quebra de senha (Ataque por dicionário)

O protocolo SIP envia a sua senha de autenticação utilizando o algoritmo de desafio MD5. O invasor, por sua vez, pode capturar os pacotes na rede utilizando um programa de mercado comum, como por exemplo o Wireshark, para capturar dados como o usuário. Quando o ataque

for feito, ele já terá os dados necessários para efetuar a investida com sucesso já na primeira vez, eliminando as chances de medidas corretivas por parte do administrador da telefonia IP.

2.5.1.3 Ataques de dicionário na autenticação SIP

Segundo (THERMOS, 2007), esse ataque tem como objetivo obter credenciais de usuários válidos em um sistema de comunicação de telefonia SIP, utilizando-se de um ataque de força bruta, ou seja, enviando várias requisições de registro com identificação (IDs) e senhas a partir de um dicionário.

2.5.1.4 Negação de Serviço (Denial of Service)

Nos ataques conhecidos como negação de serviços, ou pelo acrônimo DoS (Deniel of Service), pode-se direcionar para camadas de infraestrutura em ambiente VoIP. Segundo (THERMOS, 2007)), os ataques DoS têm como principal objetivo provocar a interrupção do serviço alvo. Nesse caso, o ataque é direcionado tanto para o sistema operacional quanto para os serviços de rede. Essa é uma ameaça que gera muita preocupação para as empresas (THERMOS; ARI, 2008).

Os tipos de ataque de negação de serviço comum, conforme Figura 6, são os de inundação que consistem em enviar uma sobrecarga de mensagens para um único destino, provocando o mau funcionamento e o pacote deformado, conhecido como processo Fuzzing. Ele gera pacote deformado aleatoriamente, provocando um comprometimento do dispositivo alvo.

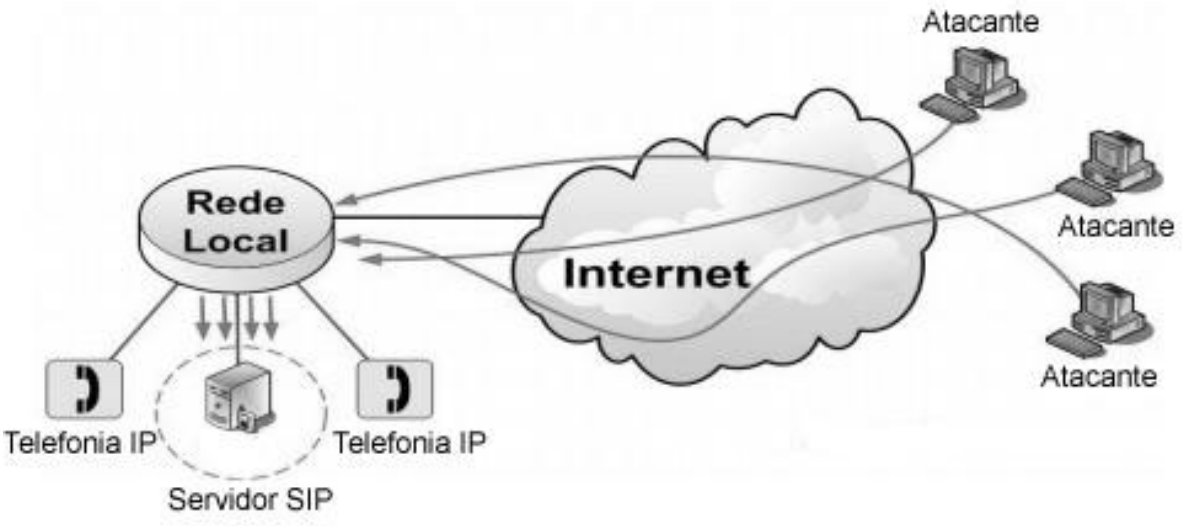


Figura 6 – Ataque de negação de serviço em servidor SIP

Fonte: (Brito S. H. B, 2011)

O ataque de negação de serviço a um sistema VoIP é fácil de ser realizado em redes que não são bem preparadas, pois, por ser um tipo de aplicação em tempo real, o VoIP é

particularmente sensível ao excesso de tráfego que pode ser gerado intencionalmente por um vírus, por exemplo (THERMOS; ARI, 2008).

O alvo de ataques de negação de serviço pode ser qualquer coisa no caminho da mensagem, incluindo as defesas de perímetro, o proxy SIP, ou o agente usuário (UA). O ataque também pode ser lançado a partir da rede PSTN *(Public Switched Telephone Network)* ou pode ser orientado para uma rede PSTN por trás de um proxy VoIP (THERMOS; ARI, 2008).

Qualquer interface de comunicação aberta pode ser inundada. Os melhores alvos para a inundação são as portas estáticas, como 5060 (TCP e UDP) de SIP e porta 1720 (TCP) para H.323/H.225 inicial de sinalização (THERMOS; ARI, 2008)

2.5.1.5 SIP Signalling Loop

Esse tipo de ataque, conforme Figura 7, segundo (THERMOS, 2007), afeta o sistema que não implementa mecanismo de detecção de looping. O ataque consiste em registrar dois usuários em um domínio SIP distinto, colocando dois valores no cabeçalho de contato, cada um apontando para um desses usuários em domínio contrário. Quando o SIP Proxy em um domínio recebe o INVITE para um desses usuários ele gera duas mensagens de INVITE sendo uma para cada usuário de outro domínio. No SIP Proxy do outro domínio por sua vez, ao receber esses dois INVITE's irá gerar quatro novas mensagens de INVITE para outro domínio. Sendo assim, o número de mensagens irá crescer em ordem de uma potência de base dois e rapidamente poderá comprometer o sistema SIP (THERMOS, 2007).

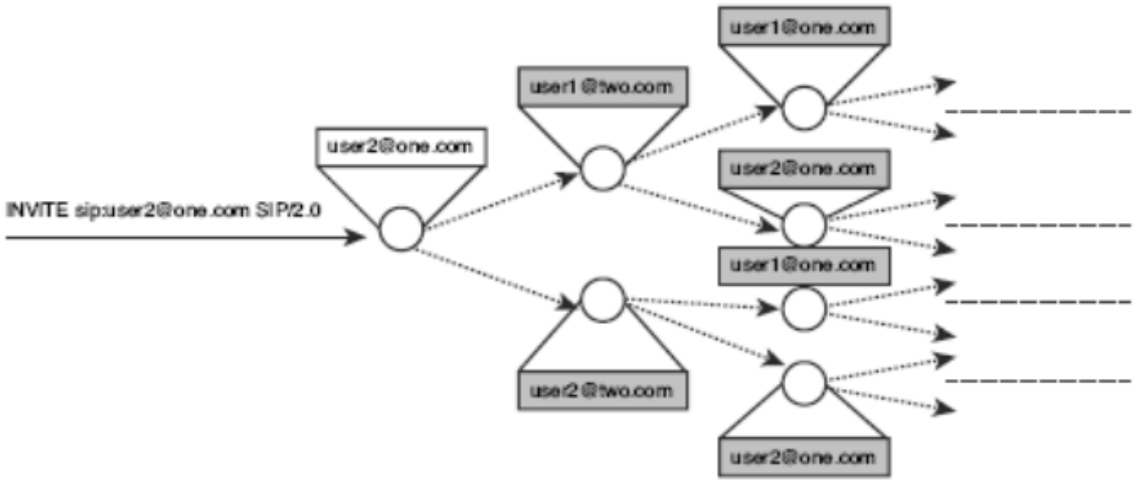


Figura 7 – SIP Signalling Loop

Fonte: (THERMOS, 2007)

2.5.1.6 Ataques Sequestro de chamadas

Para esse ataque, em (THERMOS, 2007), no cabeçalho de requisição chamado de Register, no sistema SIP, existe um registro com informações de contato que é usado pelo Prox do SIP para rotear ligações ao dispositivo do usuário, podendo ser realizado o ataque através da alteração das informações do endereço IP contidas no registro.

2.5.1.7 Ataques de dicionário na autenticação SIP

Segundo (THERMOS, 2007), esse ataque tem como objetivo obter credenciais de usuários válidos em um sistema de comunicação de telefonia SIP, utilizando-se de um ataque de força bruta, ou seja, enviando várias requisições de registro com identificação (Ids) e senhas a partir de um dicionário.

2.5.1.8 SIP Redirec

Para (BUTCHER; LI; GUO, 2007), o ataque emprega um servidor que recebe solicitações de um telefone ou Prox e retorna uma resposta de redirecionamento indicando onde o pedido deve ser repetido. Isso permite que o usuário tenha uma chamada onde o telefone toca diferente de onde está localizado, sendo que o chamador só marca um único número para chegar ao usuário. O atacante redireciona as chamadas da vítima para um número específico de sua escolha, sendo assim, ele pode receber chamadas que foram encaminhadas para o usuário atacado.

2.6 Segurança aos Protocolos SIP

Um sistema de comunicação configurado de maneira errada pode deixar brechas de segurança, proporcionando falhas nas configurações do plano de discagem, e, assim, liberando aplicativos para usuários internos e externos que não possuem autorização de acesso. Quando se desabilitam rotas que não são essenciais para o funcionamento de um sistema de comunicação, evitam-se esses e outros tipos de problemas cuidando da segurança, bem como tomando cuidados em configurações para não proporcionar esses tipos de invasões e emprego não autorizado dos sistemas de comunicação.

Segundo (GROSS, 2011), alguns cuidados devem ser tomados para evitar acessos indesejados, assim como ter um maior controle de quais rotas ou ramais podem ou não ser acessados, além do segmento de classes das extensões em diferentes contextos e trabalhar com as inclusões entre eles. Quando for utilizar URAs, tem que ter a certeza de que as ligações que entram por ela tenham seu acesso controlado e não se utilizem do sistema de comunicação. Devemos ter cuidado com o default do Asterisk, pois sempre que uma extensão não for encontrada, a mesma será direcionada para o contexto padrão.

Para (Brito S. H. B, 2011), o protocolo SIP deve oferecer confiabilidade de maneira que somente usuários autorizados possam ter acesso às informações que estão sendo transmitidas, pois o sigilo das ligações deve ser mantido. Outra característica é a integridade das seções do SIP, que deve garantir que as seções sejam mantidas até que uma das partes solicite formalmente a desconexão.

Segundo (YOSHIOKA, 2003), para que se possa prover segurança para a rede SIP, podemos utilizar o IPSec, S/MIME e TLS, pois o IPSec proporciona a capacidade de comunicação segura entre os pontos através do estabelecimento de uma rede virtual (VPN). O S/MIME concede a segurança de conteúdo, utilizando-se da criptografia do conteúdo das mensagens SIP, que, por sua vez, se utiliza da tecnologia RSA (Rivest-Shamir-Adleman), a qual é uma metodologia segura para emitir um e-mail, mas também para promover segurança ao SIP. O TLS (Trasnport Layer Securit) propõe uma camada segura de transporte que envolve o TCP.

2.7 Kali Linux

Segundo (HERTZOG;; O'GORMAN;; AHARONI, 2012), o projeto KaliLinux começou em julho de 2012, quando a Offensive Security decidiu substituir o projeto venerable black track linux, que foi mantido manualmente e poderia ser usado como Debian derivative3, com o intuito de concluir o trabalho de infra-estrutura e melhorar as técnicas de pacotes.

A decisão foi a de criar o Kali on top da distribuição Debia, porque ela é conhecida por sua qualidade, estabilidade e ampla seleção de *software* compatível. O primeiro lançamento (versão 1.0) aconteceu um ano depois, em março de 2013, e foi baseado no Debian 7 "Wheezy", a distribuição estável do Debian na época.

Nesse primeiro ano de desenvolvimento, foram empacotados centenas de aplicativos relacionados, assim como construída a infraestrutura. Nesta versão, o número de aplicativos foi significativo e uma lista de aplicativos foi cuidadosamente selecionada. Durante os dois anos após a versão 1.0, Kali lançou muitas atualizações incrementais, expandindo assim a gama de aplicações disponíveis e melhorando o suporte de *hardware,* graças às novas versões do kernel.

A distribuição do Kali Linux é baseada no teste do Debian 9. Portanto, a maioria dos pacotes disponíveis no Kali Linux tem a visão deste repositório do debian. Embora o kali Linux seja totalmente independente da infraestrutura e mantém a liberdade de mudanças.

A Figura 8 mostra a tela de inicialização do Kali Linux no Virtual Box.

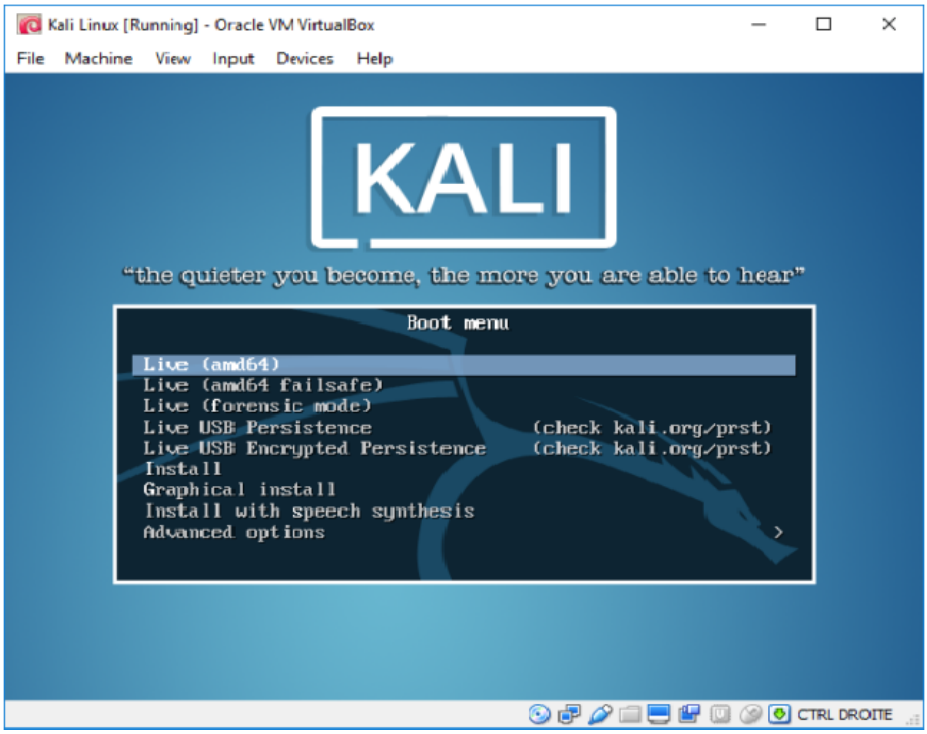


Figura 8 – Assistente de Máquina Virtual

Fonte: (HERTZOG;; O'GORMAN;; AHARONI, 2012)

2.8 Zabbix

O Zabbix foi elaborador por Alexei Vladishev, e atualmente ele é mantido e pela Zabbix SIA. O Zabbix é uma solução de nível enterprise, com código aberto e com suporte a monitoração da distribuída. Segundo (Dalle Vacche; Kewan Lee, 2015), o Zabbix surgiu em 2001 e desde o seu lançamento se distinguiu como uma solução de monitoramento poderosa e eficaz.az.

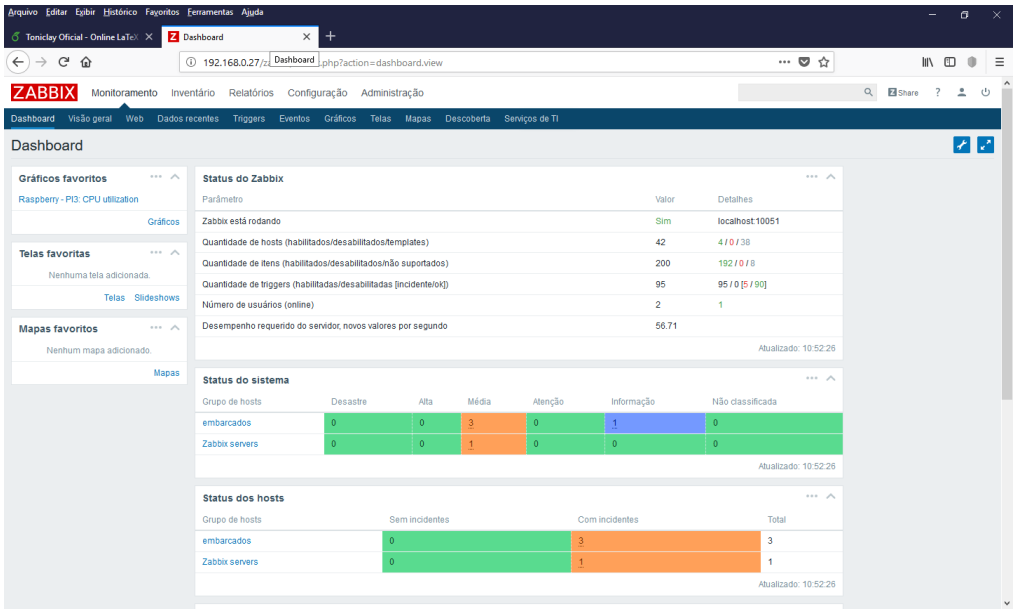
O Zabbix é um *software* que monitora vários procedimentos da rede, dos servidores e de seus serviços. Utiliza-se de um mecanismo flexível de notificação que permite configurar alertas através de e-mail em seus eventos, alertando assim seu administrador. As notificações permitem que rapidamente os problemas no ambiente sejam resolvidos. O Zabbix também oferece recursos de relatórios e visualização de dados armazenados. Isso faz com que o Zabbix seja uma ferramenta de planejamento de capacidade.

Os relatórios e estatísticas do Zabbix, e seus parâmetros de configuração, estão sempre disponíveis em interface web. O uso desta interface web garante que se possa avaliar o estado de sua rede e de seus servidores a partir de qualquer local. Quando corretamente configurado o Zabbix desempenhar um papel importante na infraestrutura de monitoramento de TI. Estas características se aplicam as empresas de pequeno e grande porte.

Zabbix é um *software* consolidado como ferramenta de monitoramento em redes de

computadores, servidores e serviços. O mesmo possui o intuito de monitorar a integridade, disponibilidade, experiência de usuário e qualidade de serviços. A figura 9 demonstra a interface do *software* Zabbix.

Figura 9 – *Software* Zabbix.



Fonte: Autoria própria.

3

Trabalhos Correlatos

Os trabalhos abordados nesta pesquisa contêm um grande número de referências. Dessa forma, utilizou-se a Base do IEEE com estudos realizados entre 2012 e 2016, com foco nas palavras-chaves (VoIP, Asterik, SIP, Segurança).

3.1 Sistema de Comunicação IP

O artigo (LOMOTEY; DETERS, 2014) mostra que os sistemas de comunicação IP têm sido alvo de ataques como roubo de chamada e ataques a servidores, o que possibilita acesso aos dados dos usuários. Sendo assim, o autor propôs uma solução para prevenir o acesso de atacantes ao sistema de telefonia construído em Asterisk. Em seu experimento, não se utilizou uma plataforma completa para o Asterisk, pois ele propôs um middleware baseado em nuvem, camada esta que mantém a parte mais sensível da chamada de informações.

O Asterisk foi utilizado para as discagens, chamadas, roteamento e recebimento das chamadas. O middleware utilizou-se do padrão REST para interação com o Asterisk. O sistema comunicação IP é uma tecnologia adotada na maioria das empresas para gerir a telefonia, permitindo uma comunicação intraoffice, que é a comunicação com entidades empresariais externas. Neste trabalho, os autores propuseram um sistema comunicação IP distribuído e baseado na tecnologia Asterisk para auxiliar na marcação de clientes com monitoramento mínimo dos empregados no call center. Foram utilizadas as ferramentas Fail2Ban e Snort como medidas para verificar as limitações de ataque, pois a escalabilidade do sistema Asterisk vem sendo posta em questão quando está sob ataque.

O ataque DoS na primeira experiência foi avaliado por três meses em cenário real, onde foram identificados cerca de 25.241 ataques com o intuito de inundar os servidores Asterisk, tornado-os inacessíveis.

Foram detectados 39.689 ataques de identidade falsa os quais envolveram a emissão

de credenciais falsas na tentativa de entrar no sistema e realizar milhares de chamadas quando esse tipo de ataque era realizado em empresas. Tendo em vista os ataques mencionados, os autores propuseram a camada de middleware para coordenar todas as atividades do sistema, armazenamento dos dados em SQL e empacotamento de marshaling no middleware para prevenir o roubo de informações.

Os autores concluíram que o experimento teve sucesso nos ataques Denial of Service (DoS), que são ataques de identidades falsas e ataques de sondagem. Foi avaliado ainda o desempenho do sistema contra inundações, que mostrou um aumento de alto desempenho. Este trabalho ainda sugere como estudo futuro a exploração da expansão no discador preditivo onde se misturam a discagem preditiva e a discagem automática.

3.2 Análise de Segurança VoIP

Em (REHMAN; ABBASI, 2014)), o termo VoIP é utilizado para a comunicação que fornece dados de voz e multimídia utilizando-se da Internet que, devido à sua popularidade, tornou-se alvo de diversos ataques.

No artigo em questão, o autor analisou a segurança na arquitetura VoIP no sistema de comunicação de voz sobre IP Asterisk. Percebendo que a maioria dos ataques estavam relacionados à fragilidade do protocolo SIP, foram detectados ataques de espionagem, modificação e interrupção involuntária.

No estudo, foi proposta, para resolução do problema apresentado, a necessidade de o protocolo SIP de fornecer um mecanismo de autenticação eficiente e seguro, garantindo assim uma maior proteção aos ataques.

Foi sugerido ainda atribuir um token criptográfico que autenticaria os usuários, possibilitando a identificação do utilizador e proporcionando uma maior segurança. Assim não existiria a necessidade do usuário de colocar a senha para utilizar outros serviços disponíveis.

3.3 Detecção de intrusão VoIP com Snort

No artigo de (ČÍŽ et al., 2012), os autores descrevem alguns tipos de ataque em tráfego de VoIP e apresentam formas de proteção contra eles.

Em seu experimento, foi proposto um modelo focado em ataque DoS com o objetivo de causar um mau funcionamento no Asterisk. Foi utilizado o SIPp, ferramenta usada para verificar a funcionalidade do sistema de detecção e causar anomalias em ataques de negação de serviço, bem como a ferramenta de *software* Snort, usada para a detecção de ataque em rede livre, e de sistemas de prevenção capazes de realizar análise do tráfego e log de pacotes em redes IP utilizadas.

O artigo foi organizado pelas descrições dos tipos de ameaças em VoIP, uma proposta de modelo de proteção IDS com experiência, finalizando com explicação dos resultados encontrados. O tráfego foi controlado através do intercâmbio do Asterisk, criando regras definidas focadas em negação de serviços. O objetivo do trabalho futuro é encontrar outras variantes de regras de negação de serviço e avaliar a sua eficácia, eventualmente, para se concentrar em outro tipo de ataque.

3.4 Ataque de Negação de Serviço ao protocolo SIP

O artigo de (BANSAL; PAIS, 2015), apresenta-se o protocolo SIP como sendo o mais popular usado em protocolo VoIP e propõe um esquema de mitigação para SIP em sistemas VoIP para protegê-lo de inundações de ataques DoS.

Os autores criaram um protótipo para criar inundação de ataques DoS baseado em um servidor SIP para avaliar o desempenho do sistema proposto, no qual realizaram um total de 167 chamadas, significando que 167 canais ficaram disponíveis no servidor SIP no esquema de mitigação. A ferramenta SIPp foi executada em 10 terminais, onde cada um emitiu 1000 mensagens CONVIDADOS e o número de terminais foi aumentando um por um, ao tempo em que foram enviadas 1000 * 10 mensagens INVITE, tendo como resultado canais ocupados no servidor Asterisk .

Sendo assim, antes de implementar o esquema de mitigação, apenas uma atacante poderia envolver todos os canais SIP enviando apenas 200 INVITE mensagens no servidor SIP. Após a implantação do esquema proposto, foi identificado um usuário atacante que começou a descartar todas as mensagens INVITE recebidas, bem como parou de enviar mensagens BYE para o atacante desprender todos os canais que foram ocupados por ele inicialmente.

Os autores concluíram que o plano de mitigação proposto atingiu o seu objetivo de proteger o servidor SIP de ficar caindo e fornecendo o serviço a todos os novos usuários legítimos durante um ataque de negação de serviço (DoS).

3.5 Considerações sobre os Trabalhos Correlatos

A tabela 1 ilustra a comparação entre os trabalhos correlatos, demostrando a utilização de *hardware, software,* protocolo, desempenho e se é um experimento. Podemos verificar na tabela 1 (BANSAL; PAIS, 2015), que é realizada uma avaliação de um ataque de negação de serviço em um computador com protocolo SIP. Trata-se de caso semelhante ao nosso, com a diferença de que utilizamos um Raspberry Pi 3 em um ataque de negação de serviço e testamos o Raspberry Pi 3 em mais dois ataques, além de coletar os resultados de consumo de energia bem como o comportamento da memória e CPU.

Podemos observar que dois estudos se utilizam do protocolo IP e dois do protocolo SIP, sendo os dois trabalhos IP: (LOMOTEY; DETERS, 2014) e (ČÍŽ et al., 2012) verificam o desempenho contra inundações de DoS e a análise do tráfego com os logs de pacotes; (REHMAN; ABBASI, 2014) e (BANSAL; PAIS, 2015) analisam a eficiência da segurança e avaliam um esquema de mitigação para SIP em sistemas VoIP com o intuito de protegê-los de inundações dos ataques DoS.

Tabela 1 – Comparação entre os trabalhos correlatos.

Autores	Tema	Hard	Soft	Prot	Desemp	Exper
(LOMOTEY; DETERS, 2014)	Sistema de comunicação IP	X	Asterik Fail2 Ban2 Snort	IP	Contra inundações DoS	X
(REHMAN; ABBASI, 2014)	Analise de Segurança VoIP	X	Asterik	SiP	Eficiência de Segurança	X
(ČÍŽ et al., 2012)	Detecção de intrusão VoIP com Snort	X	Asterik, Snort	IP	Analise de trafego e log de pacotes	X
(BANSAL; PAIS, 2015)	Ataque de negação de serviço ao protocolo SiP	X	Asterik	SiP	Avaliar	X
Esta Dissertação	Uma Abordagem de Segurança do Sistema Asterisk em Plataformas Embarcadas usando o Protocolo SIP	X	Asterik Zabbix Wireshark x-lite VM Virtual Box	SiP	Analise de ataque	X

Fonte: Autoria própria.

4

Cenário de Testes - Iniciando os Ataques

Este capítulo apresenta a implementação do experimento, que consiste em: realizar a montagem do cenário de teste com o dispositivo embarcado Raspberry Pi 3; efetuar a abordagem dos *softwares* necessários no dispositivo para a elaboração do experimento de invasão, seguindo a logica de primeiramente levantar a topologia da rede do servidor a ser at acado; e montar os três tipos de ataque na seguinte ordem :

- Ataque de Autenticação;
- Ataque Man-in-middle;
- Ataque Negação de Serviço DOS.

Nos ataques são realizados os monitoramentos do consumo do processamento, memória e energia, tendo como objetivo avaliar o dispositivo embarcado Raspberry Pi 3 e o *software* Asterisks.

4.1 Elaboração do Cenário de Testes

Ao realizar o cenário de teste foi necessário instalar um sistema operacional no dispositivo embarcado, utilizando Raspbian no Raspberry Pi 3, sistema operacional este baseado no GNU Linux Debian 8.

Em sequência, ocorreu a instalação do *software* de comunicação por voz sobre IP Asterisk. Por último foi instalado um dissipador de calor, assim como um *cooler*, a fim de refrigerar os dispositivos. Isso porque houve um elevado número de ocorrências das mensagens de alarme referindo-se à alta temperatura no dispositivo.

Foi necessária a instalação de duas máquina virtuais utilizando o Oracle VM VirtualBox: uma com o Kali Linux, para realizar o ataque, e outra com *software* de monitoramento Zabbix, para capturar o processamento e memória do Raspberry Pi 3 na hora de realização dos ataques.

A Tabela 2 ilustra os *softwares* utilizados no experimento.

Tabela 2 – *Softwares* utilizados no experimento.

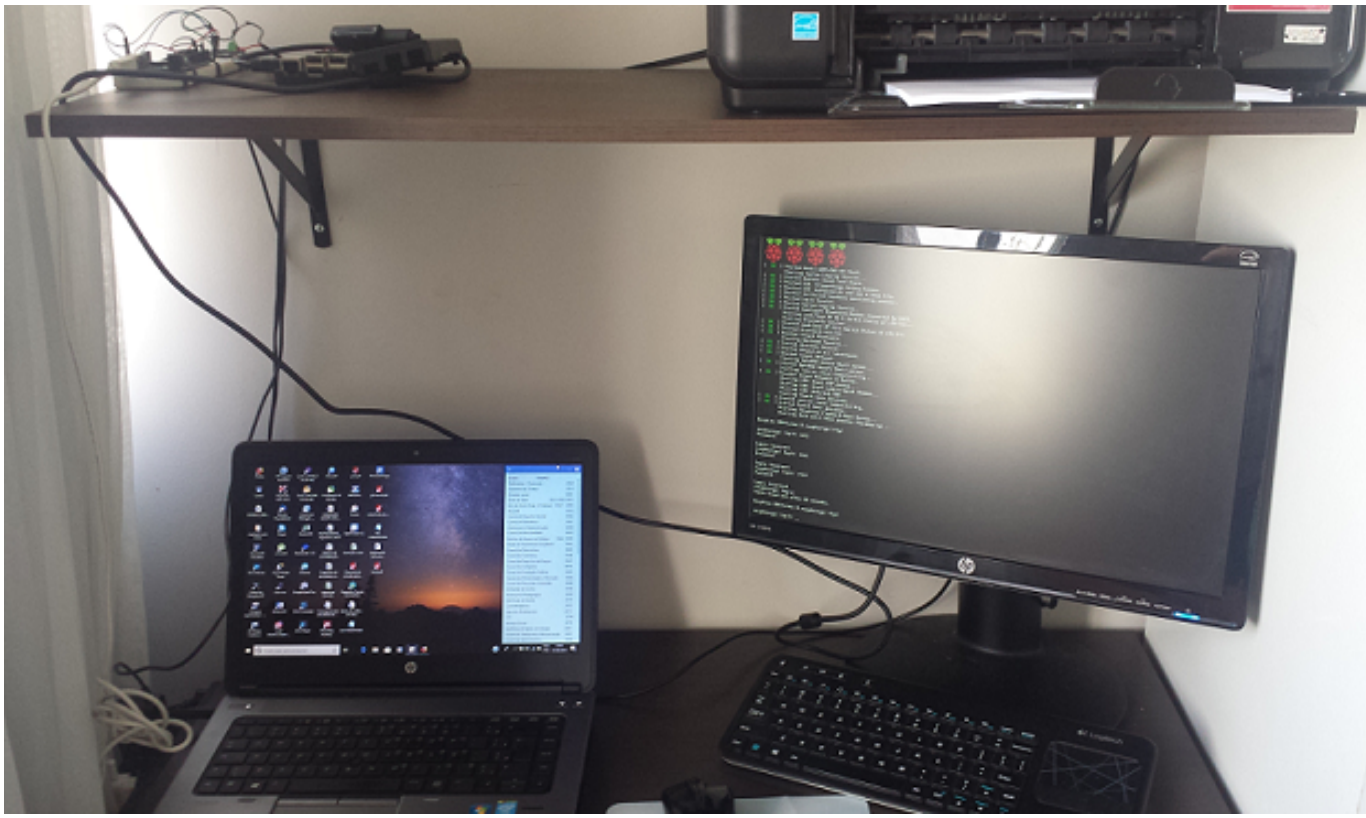
Dispositivos	***Notebook***	**VM VirtualBox**
Raspberry Pi 3 Asterisk 13	Wireshark Zoiper X-Lite VM Virtual Box	Asterisk 13 Zabbix Kali Linux

Fonte: Autoria Própria

A intenção é verificar o comportamento do Raspberry Pi 3, juntamente ao *software* de comunicação voz sobre IP Asterisk, bem como verificar o status do processamento, memoria e consumo de energia durante os ataques propostos, utilizando o *software* Zabbix; acompanhamento do consumo de energia com o circuito INA219.

A Figura 10 ilustra o cenário real em que os testes foram realizados.

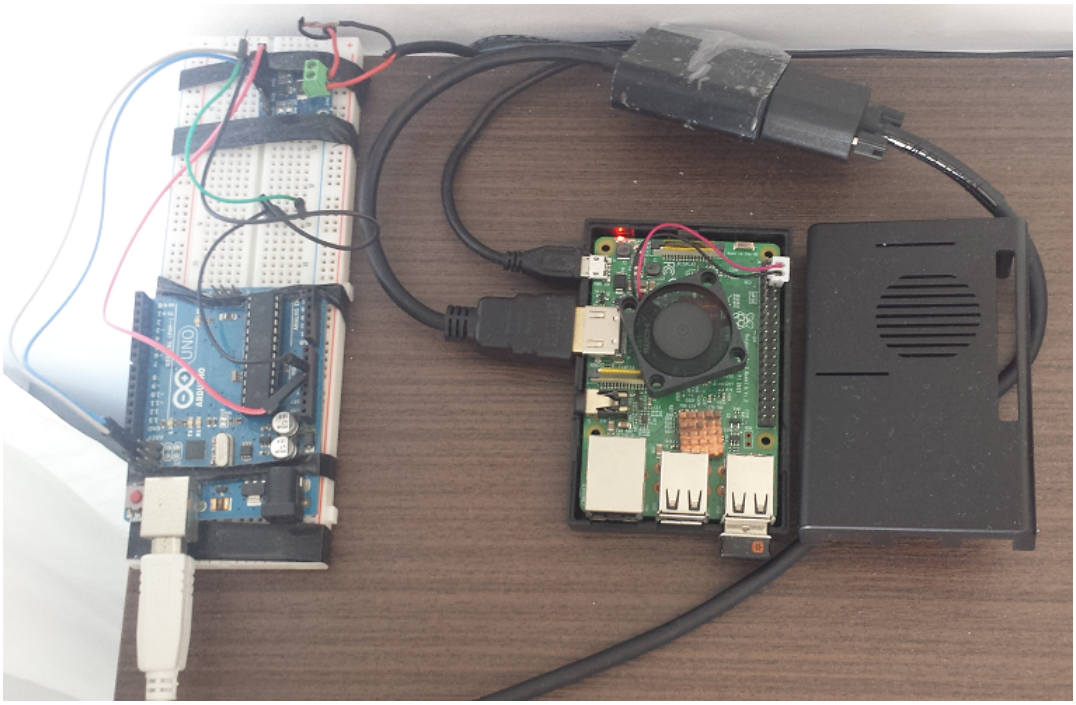
Figura 10 – Cenário real de testes.



Fonte: Autoria própria.

A Figura 11 mostra o Raspberry Pi 3 e o circuito INA219 medidor de energia.

Figura 11 – Raspberry Pi 3 e o circuito INA219 medidor de energia.



Fonte: Autoria própria.

4.2 Iniciando os ataques

Para inciar os ataques com o Kaki LInux no Raspberry Pi 3, juntamente ao Astrerisk, foi necessário verificar inicialmente a topologia da rede na qual realizamos o ataque. Para isso, foi preciso:

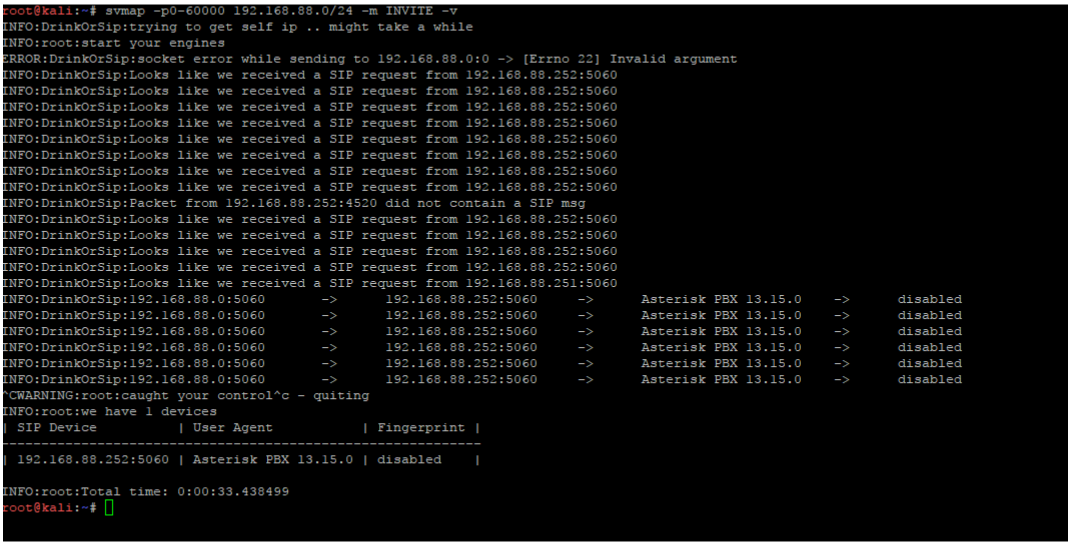
- Verificar a faixa de Ip;
- Versão da aplicação;
- Extensões.

Sendo assim, o atacante faz uma varredura dos ip's e portas em uma rede com o comando no kalli Linux, para fazer a varredura conforme Figura 12:

Para realizarmos a varredura da rede a ser atacada, usamos o comando no Kali Linux abaixo.

```
# svmap -p0-60000 192.168.0.0/24 -m INVITE -v
```

Figura 12 – Varredura de rede através do comando smvmap



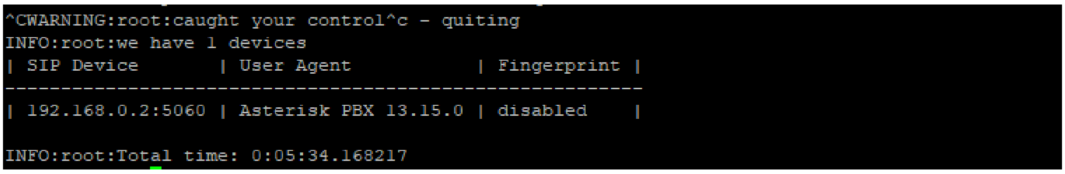
```
root@kali:~# svmap -p0-60000 192.168.88.0/24 -m INVITE -v
INFO:DrinkOrSip:trying to get self ip .. might take a while
INFO:root:start your engines
ERROR:DrinkOrSip:socket error while sending to 192.168.88.0:0 -> [Errno 22] Invalid argument
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Packet from 192.168.88.252:4520 did not contain a SIP msg
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.252:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.88.251:5060
INFO:DrinkOrSip:192.168.88.0:5060     ->     192.168.88.252:5060     ->     Asterisk PBX 13.15.0     ->     disabled
INFO:DrinkOrSip:192.168.88.0:5060     ->     192.168.88.252:5060     ->     Asterisk PBX 13.15.0     ->     disabled
INFO:DrinkOrSip:192.168.88.0:5060     ->     192.168.88.252:5060     ->     Asterisk PBX 13.15.0     ->     disabled
INFO:DrinkOrSip:192.168.88.0:5060     ->     192.168.88.252:5060     ->     Asterisk PBX 13.15.0     ->     disabled
INFO:DrinkOrSip:192.168.88.0:5060     ->     192.168.88.252:5060     ->     Asterisk PBX 13.15.0     ->     disabled
INFO:DrinkOrSip:192.168.88.0:5060     ->     192.168.88.252:5060     ->     Asterisk PBX 13.15.0     ->     disabled
^CWARNING:root:caught your control^c - quiting
INFO:root:we have 1 devices
| SIP Device          | User Agent           | Fingerprint |
-----------------------------------------------------------
| 192.168.88.252:5060 | Asterisk PBX 13.15.0 | disabled    |

INFO:root:Total time: 0:00:33.438499
root@kali:~#
```

Fonte: Autoria própria.

Tendo como resultado o IP e a versão da aplicação conforme Figura 13

Figura 13 – Resultado da Varredura de rede através do comando smvmap



```
^CWARNING:root:caught your control^c - quiting
INFO:root:we have 1 devices
| SIP Device        | User Agent           | Fingerprint |
---------------------------------------------------------
| 192.168.0.2:5060 | Asterisk PBX 13.15.0 | disabled    |

INFO:root:Total time: 0:05:34.168217
```

Fonte: Autoria própria.

Para que pudéssemos identificar uma extensão no Kali Linux, usamos o comando abaixo:

```
# svwar  192.168.0.2 -force
```

Na Figura 14 mostra a existência de um ramal com a extensão 100.

Figura 14 – O atacante identifica uma extensão, a extensão 100.

```
root@kali:~# svwar  192.168.0.2 --force
WARNING:TakeASip:Bad user = SIP/2.0 401  - svwar will probably not work!
WARNING:TakeASip:We got an unknown response
ERROR:TakeASip:Response: 'SIP/2.0 401 Unauthorized\r\nVia: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK-1041488787;received=192.168.0.4;rport=5060\r\nFrom: "100"<sip:100@1
92.168.0.2>;tag=3130300133373637313838363537\r\nTo: "100"<sip:100@192.168.0.2>;tag=as01a34bca\r\nCall-ID: 1924117149\r\nCSeq: 1 REGISTER\r\nServer: Asterisk PBX 13.15.0
\r\nAllow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE\r\nSupported: replaces, timer\r\nWWW-Authenticate: Digest algorithm=MD5,
realm="asterisk", nonce="09548e43"\r\nContent-Length: 0\r\n\r\n'
WARNING:root:found nothing
root@kali:~#
```

Fonte: Autoria própria.

Concluindo esta primeira etapa, podemos agora realizar os ataques.

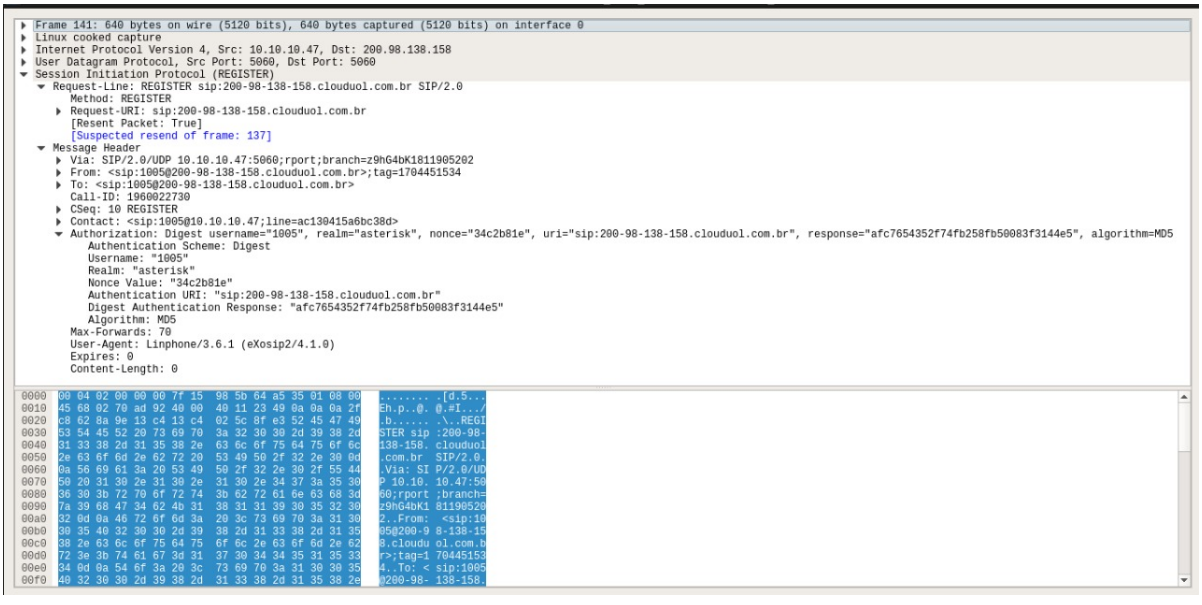
5

Experimento Dos Ataques

5.1 Ataque de Autenticação

O protocolo de iniciação de sessão (IETF RFC 3261) é um padrão amplamente utilizado em comunicações VoIP para configurar e desativar chamadas SIP. A Figura 15 representa uma mensagem SIP que foi trocada durante a realização do teste.

Figura 15 – Mensagem SIP trocada.



Fonte: Autoria própria.

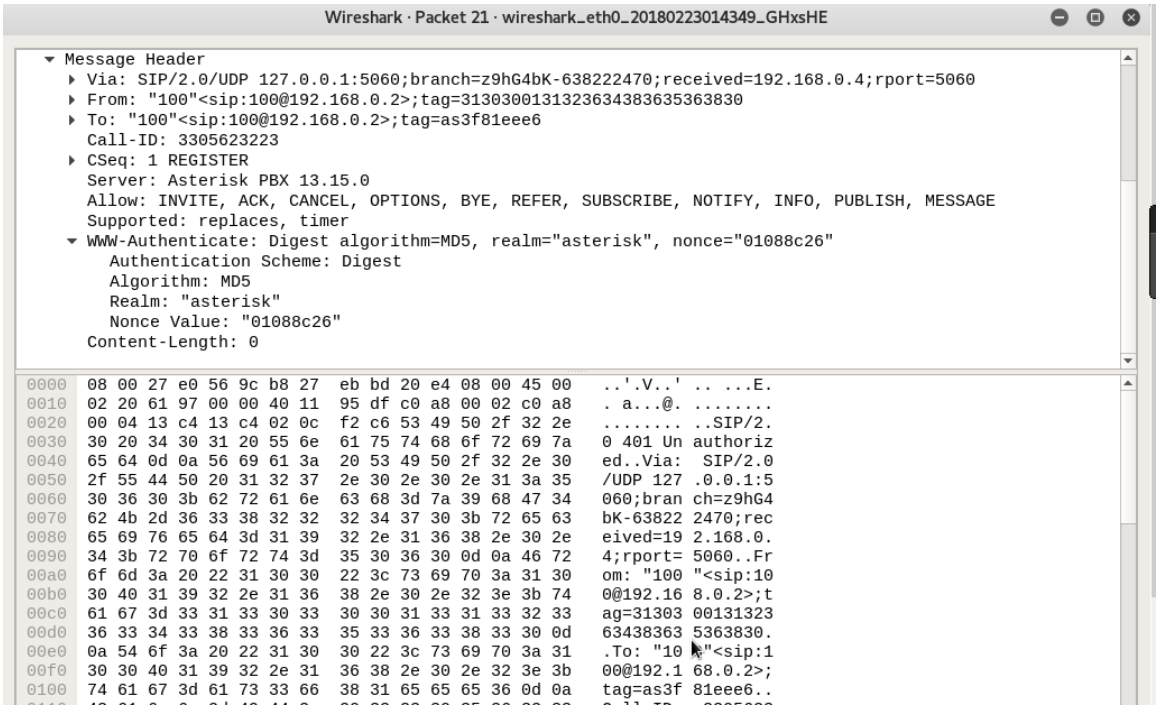
O dispositivo do usuário (chamado de *User Agent* na terminologia SIP) é registrado no servidor de registros responsável por manter um banco de dados de registros de todos os assinantes. No caso deste teste foi utilizado o Asterisk para servidor de registro e servidor proxy.

O registro do usuário no VoIP é necessário porque fornece os meios para localizar e contactar uma parte remota.

Para um usuário se registrar, é feita uma solicitação utilizando um pacote SIP chamado REGISTER para o servidor de registro. Esse pacote carrega informações como nome de usuário, domínio ou IP de origem, domínio ou IP de destino, tempo de expiração do registro e algumas outras informações padrões do pacote SIP.

Ao receber a solicitação, o servidor de registro responde para este usuário um pacote SIP chamado 401 Unauthorized. Esse pacote foi baseado na autenticação em HTTP utilizando Digest (WWW-Authenticate), conforme Figura 16.

Figura 16 – Resposta com informações do registro.



Fonte: Autoria própria.

A autenticação *Digest* é um esquema de desafio/resposta que substitui a autenticação básica. O servidor de registro, dentro do pacote SIP 401 Unauthorized, envia uma cadeia de caracteres dos dados aleatórios chamados nonce ao usuário como um desafio.

O usuário responde com um *hash* que inclui o nome de usuário, senha e nonce entre informações adicionais. A complexidade que apresenta este *exchange* e o *hash* de dados, ou seja, utilização de senhas mais fortes, torna mais difícil roubar e reutilizar as credenciais do usuário com esse esquema de autenticação. Quando o usuário quer entrar em contato com outro usuário, ele enviará uma solicitação INVITE para o servidor proxy.

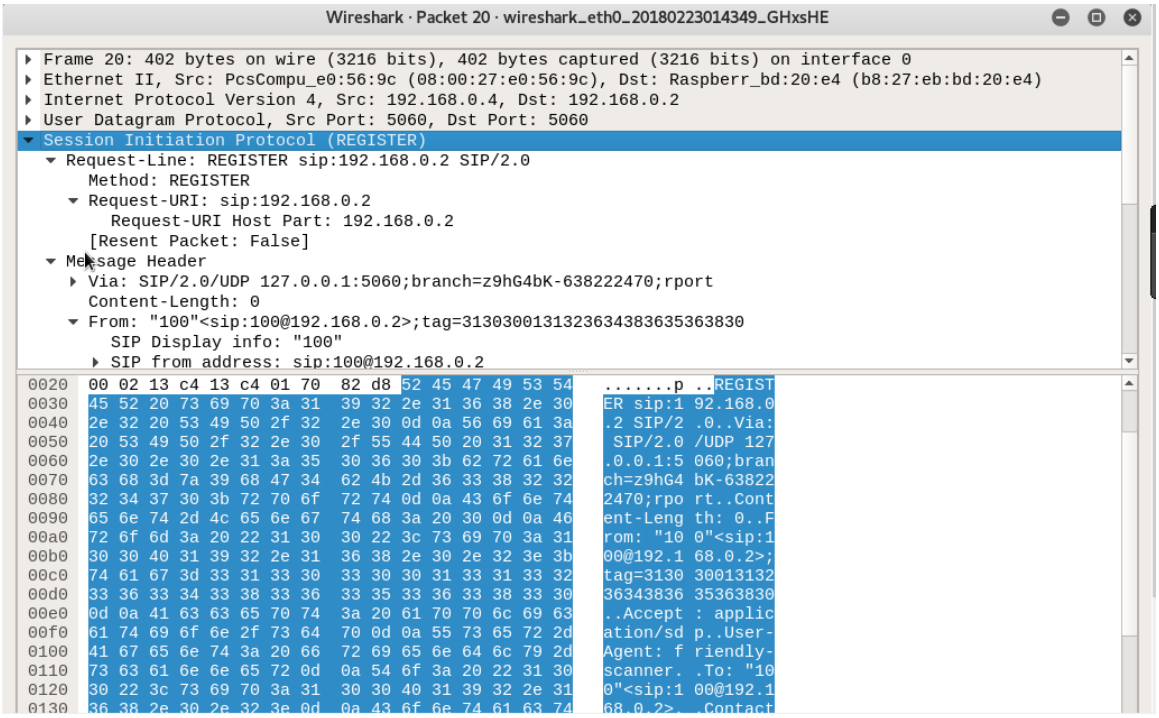
Servidores proxy são responsáveis em rotear mensagens SIP e localizar assinantes. Quando o servidor proxy recebe uma solicitação INVITE, ele tenta localizar a parte chamada

para retransmitir o progresso ao chamador executando várias etapas, como pesquisas de DNS e o roteamento de diversas mensagens SIP (provisórias e informativas).

Uma das mensagens roteadas para chegar até o usuário destino é a mensagem SIP 401 Unauthorized, o usuário só irá conseguir contactar o outro usuário quando responder o desafio com a *hash* correta. Com estes parâmetros, o invasor pode criar algorítimos de descoberta e quebra dessa cadeia de caracteres enviada pelo servidor de registro.

Entre algumas das aplicações utilizadas, está o ataque bruto que usa John The Ripper, dicionário ataque ou Quebrando a Resposta Digest. Utilizando o John the ripper, pode-se quebrar a senha e autenticar a extensão. A utilização de uma quebra de senha é um processo demorado e geralmente utiliza força bruta.

Figura 17 – Envio de pacote de REGISTER.



Fonte: Autoria própria.

No nosso experimento com o ataque de Autenticação, conforme cenário Figura 10, obtivemos exito tendo em vista nenhuma implementação de segurança dentro ou fora do Raspberry Pi 3, também não identificamos nenhum problema junto ao *Software* Asterisk.

5.2 Ataque Man-in-the-middle

Espionagem em VoIP é um pouco diferente da escuta tradicional em redes de dados, mas o conceito geral permanece o mesmo. Escutas em VoIP exigem a interceptação da sinalização e dos fluxos de mídia associados de uma conversa. As mensagens de sinalização usam protocolos

de rede separados (por exemplo, UDP ou TCP) e portas da própria mídia. Os fluxos de mídia geralmente são transportados por UDP usando o protocolo RTP (Real Time Protocol).

O ataque do tipo ARP-Spoofing ou envenenamento ARP é o meio mais eficiente de executar o ataque conhecido por Man-In-The-Middle e obter informações e fluxos de mídia em uma ligação VoIP. O *Address Resolution Protocol* (ARP) é um protocolo para mapear um endereço IP do endereço de uma máquina física (MAC) que é reconhecida na rede local.

Por exemplo, um IP versão 4 (IPv4), o tipo de IP mais comumente usado hoje em dia, tem 32 *bits* de tamanho. Em uma rede local Ethernet, entretanto, os endereços de dispositivos conectados possuem 48 *bits* de tamanho. Para aumentar a eficiência da rede e não engargalar a conexão realizando o broadcast do ARP, cada computador mantém uma tabela de endereços IP e endereços Ethernet na memória.

Isto é chamado de cache ARP. Antes de enviar um broadcast para toda a rede, o computador transmissor verificará se a informação existe em seu cache ARP. Se existir, ele completará os dados Ethernet sem enviar um broadcast ARP, evitando de engargalar a conexão. Cada entrada dura normalmente 20 minutos (mas depende do sistema operacional).

A RFC 1122 especifica que é possível configurar o valor do tempo de expiração do cache ARP no host. Para examinar o cache em um computador com Windows, UNIX ou Linux, digite "arp -a"no console ou prompt de comando. O ARP provê as regras do protocolo realizando esta correlação e possibilitando a conversão de endereços em ambas as direções.

5.2.1 Como a tabela ARP funciona?

Quando um pacote destinado a uma máquina de uma rede local particular chega no *gateway*, o *gateway* solicita ao programa ARP que encontre um host físico ou endereço MAC que esteja de acordo com o endereço IP. O programa ARP olha no ARP cache e, se encontra o endereço, retorna o mesmo e assim o pacote pode ser convertido ao formato e tamanho corretos, sendo enviado à máquina, Se nenhuma entrada é encontrada para o endereço IP, o ARP faz um broadcast de um pacote de requisição especial a todas as máquinas na rede para ver se uma das máquinas sabe qual delas tem o IP associado.

Se uma máquina reconhecer o endereço IP como o seu, ela retorna uma resposta indicando o fato. Assim, o ARP atualiza seu cache para futura referência e então envia o pacote de dados para o endereço MAC que respondeu. O ARP Spoofing é um tipo de ataque no qual uma falsa resposta ARP é enviada a uma requisição ARP original. Enviando uma resposta falsa, o roteador pode ser convencido a enviar dados destinados ao computador 1 para o computador 2, e o computador por último redireciona os dados para o computador 1. Se o envenenamento ocorrer, o computador 1 não tem ideia do redirecionamento das informações.

A atualização do cache do computador alvo (computador 1) com uma entrada falsa é chamado de envenenamento. Uma terceira pessoa está inserida entre o caminho de comunicação

dos dois. Não há qualquer interrupção do tráfego de ambos os computadores, pois a terceira pessoa redireciona os pacotes de dados ao computador destino.

5.2.2 Realizando o Ataque

O Monitoramento de tráfego de VoIP pode permitir que um invasor capture pedidos SIP, dados RTP, captura de autenticação SIP e escutas de telefonemas. Para este ataque, o invasor utiliza o ataque chamado Man-in-the-middle (homem no meio) que exigem os seguintes passos:

- Envenenamento ARP / spoofing (arpspoof);
- sniffing tráfego (wireshark).

Comandos utilizado no kali linux para fazer o man in the middle:

Ativar o forward

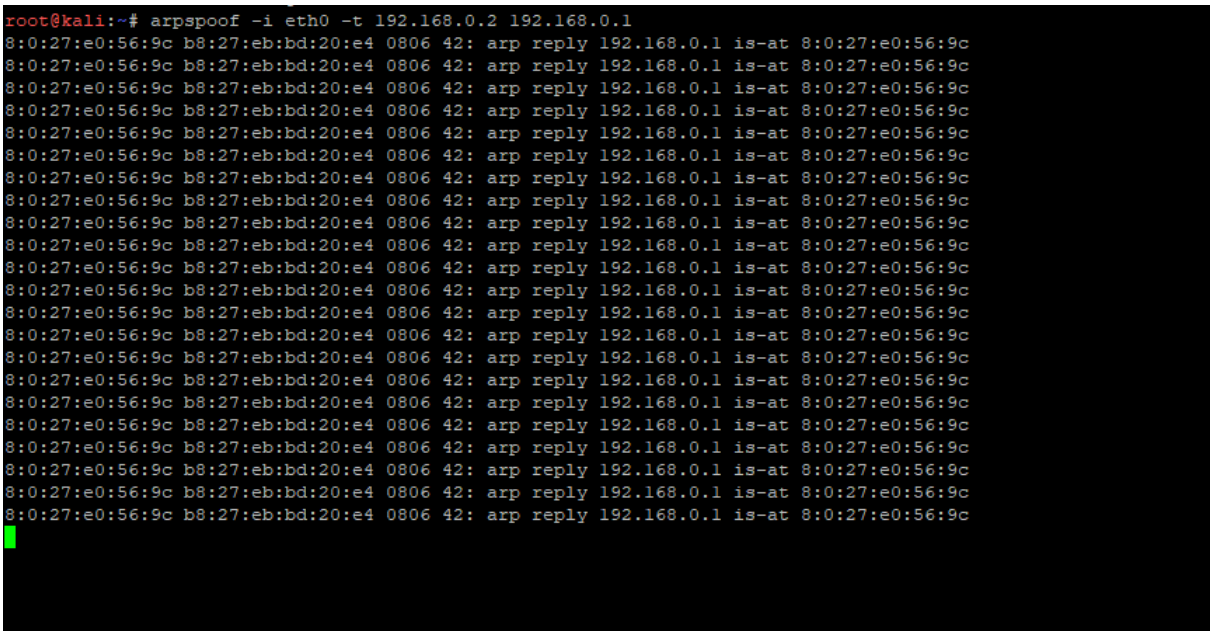
```
# echo ‘‘1? > /proc/sys/net/ipv4/ip_forward
 ou  # sysctl -w net.ipv4.ip_forward=1
```

Envenenar a tabela arp

```
# arpspoof -i eth0 -t 192.168.0.2 192.168.0.1
```

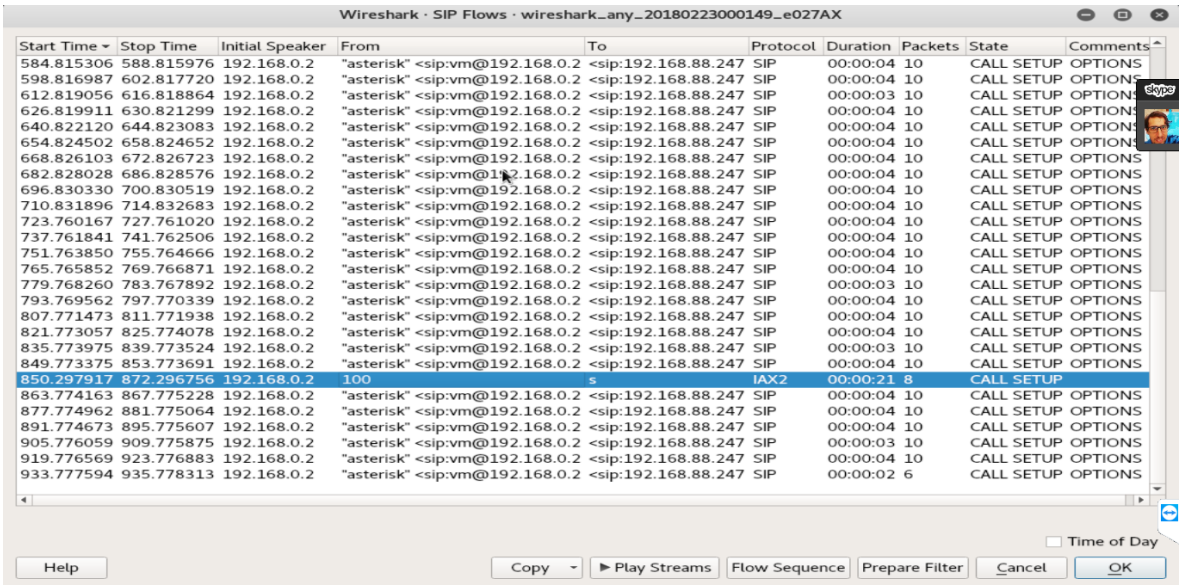
Na Figura 18 temos o resultado do comando que realiza o envenenamento da tabela ARP e com o Wireshark no kali linux o atacante consegue capturar os pacotes com a extensão 100, conforme a Figura 19.

Figura 18 – Ativação do arpspoof.



Fonte: Autoria própria.

Figura 19 – Captura de pacotes com wireshark.



Fonte: Autoria própria.

No experimento com o ataque de Ataque Man-in-the-middle, conforme Figura 18, obtivemos êxito, tendo em vista que não houve nenhuma implementação de segurança dentro ou fora do Raspberry Pi 3. Também não identificamos nenhum problema junto ao *Software* Asterisk.

5.3 Ataque Negação de Serviço DoS

O objetivo de qualquer ataque DoS é sobrecarregar o sistema com tantas solicitações ao ponto de ele ser forçado a encerrar. Os ataques DoS de telefonia são uma subcategoria em que esses tipos de ataques são nivelados em sistemas VoIP. Infelizmente, este tipo de ataque saltou para a frente das preocupações nos boletins de segurança como resultado de seu uso contra hospitais e linhas telefônicas 9-1-1.

Em outro desenvolvimento deprimente, alguns atacantes DoS exigem um resgate para deter o ataque. Muito parecido com o ransomware, com a ajuda de criptomoedas e spoofing de identificador de chamadas, é incrivelmente difícil identificar os invasores.

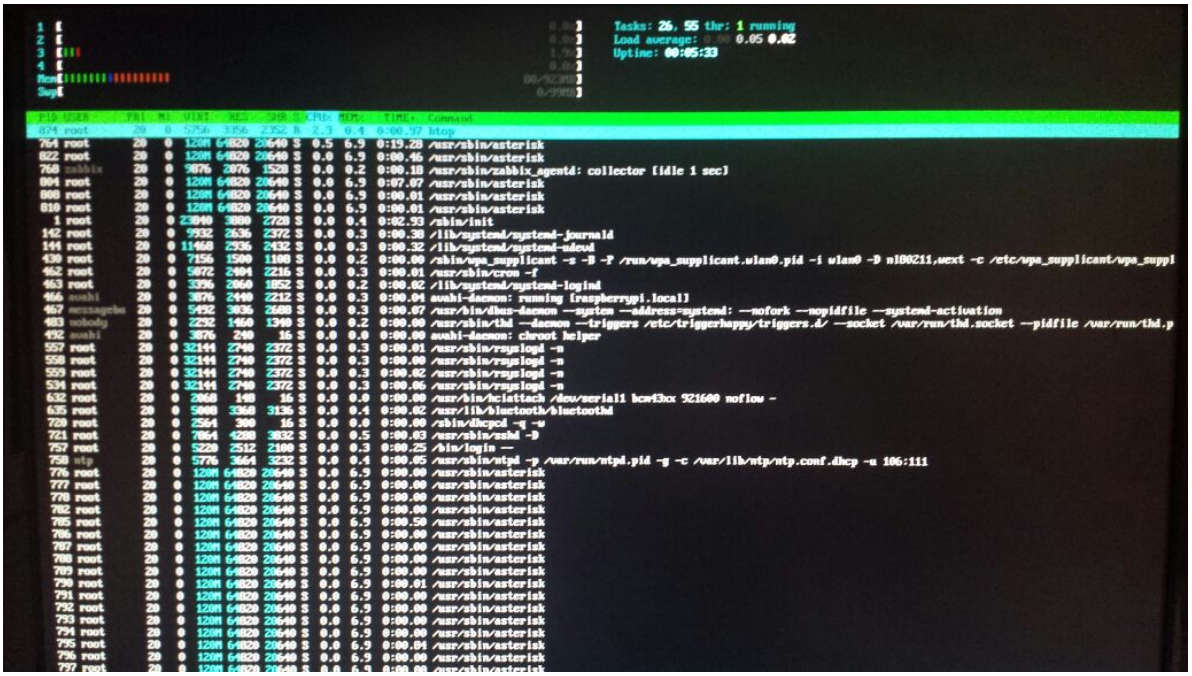
Os ataques TDoS geralmente empregam menos recursos do que os ataques DoS projetados para prejudicar os sistemas de TI, como redes, servidores e software.

Em sua forma mais básica, tudo o que um ataque TDoS exige é um discador telefônico automatizado que chame um número de telefone de destino e desligue – repetidamente. Esse conceito muito simples pode impedir qualquer outra pessoa de passar pela linha.

Com este ataque, o invasor inunda o aplicativo VoIP com inúmeras requisições e dados, o que pode levar ao travamento do PBX, impossibilitando o tráfego na rede.

A Figura 20 nos mostra o estado do Raspberry Pi 3 antes do ataque, apresentando uma livre utilização de recursos, as 4 CPU's livres e utilizando 92 MB de memória RAM.

Figura 20 – Estado do Raspberry Pi 3 antes do ataque.



Fonte: Autoria própria.

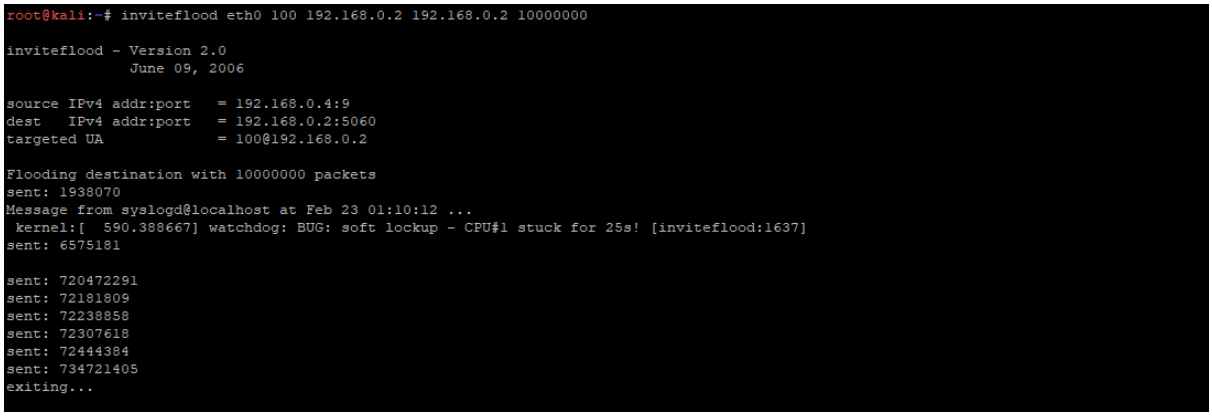
Primeiramente, vamos utilizar o kali linux para fazer o ataque de negação de serviço, conforme é demonstrado abaixo na Figura 21.

```
# inviteflood eth0 100 192.168.0.2 192.168.0.2 100000
```

Neste comando, temos o seguinte:

- inviteflood : comando
- eth0 : placa do usuário
- 100 : usuário
- 192.168.0.2 : IPdoPABX
- 192.168.0.2 : IPdoPABX
- 100000 : QuantidadePacotes

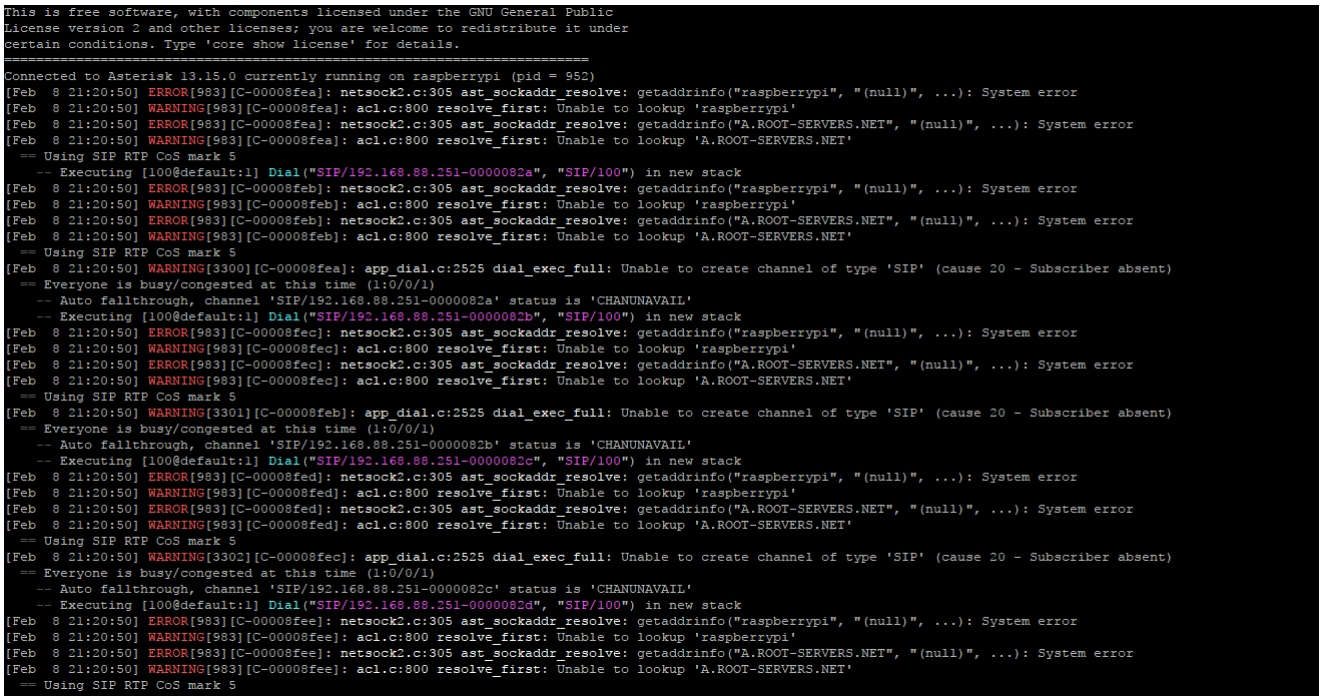
Figura 21 – comando inviteflood.



Fonte: Autoria própria.

No Asterisk do Raspberry Pi 3, o mesmo recebe os pacotes do atacante e começa a afetar a comunicação das linhas telefônicas ligadas ao Asterisk Figura 22. O Raspberry Pi 3, mesmo com um número grande de pacotes, ainda continua rodando, mas o *software* Asterisk, à medida que vai aumentado a quantidade de pacotes, começa a ser afetado.

Figura 22 – Resultado do ataque de DoS.



Fonte: Autoria própria.

No ataque de Negação De Serviço, que tem como característica o envio de pacotes, realizamos vários ataques com diferentes quantitativos de pacotes - conforme Tabela 3 para verificar o comportamento do Raspberry Pi 3 e do *software* Asterisk.

Tabela 3 – Análise de Ataque DoS por quantidade de pacotes.

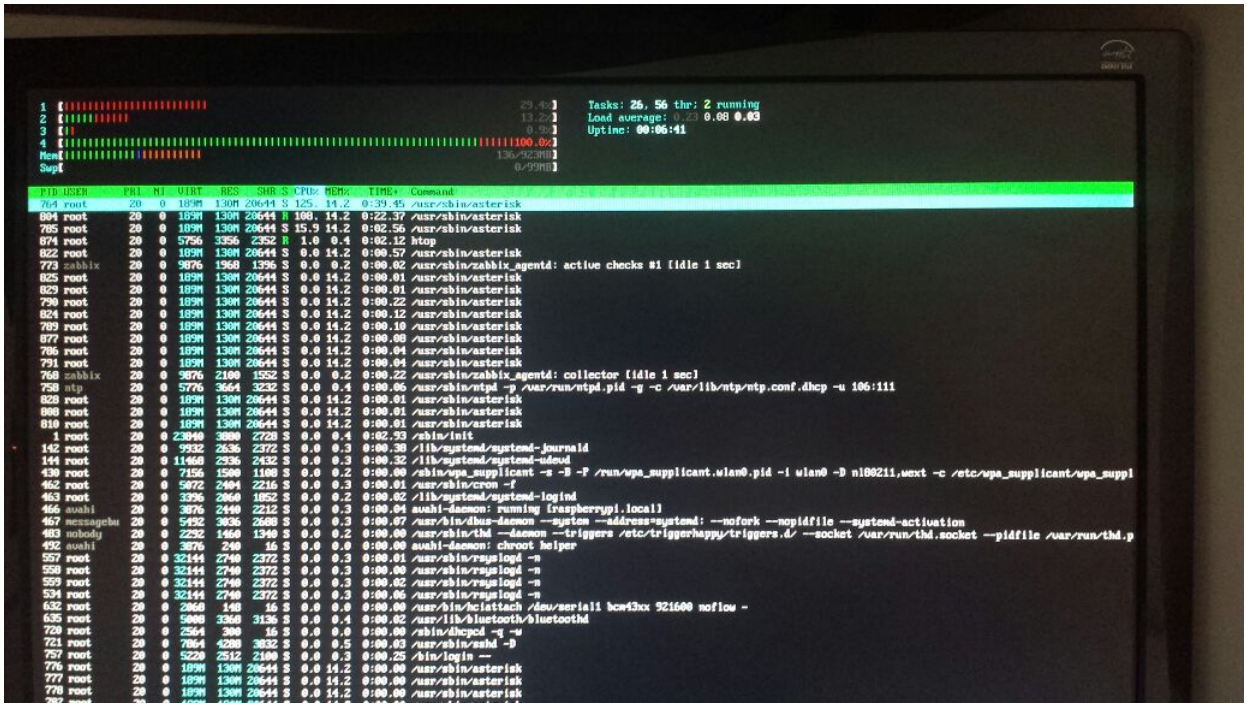
Quant De Pacotes	Status Raspberry Pi	Status Asterisk	Status Da Chamada
00000	Excelente	Excelente	Sem Falha
10.000	Excelente	Excelente	Sem Falha
50.000	Excelente	Excelente	Sem Falha
75.000	Excelente	Excelente	Sem Falha
100.000	Excelente	Excelente	Sem Falha
150.000	Excelente	Bom	Com falhas
250.000	Excelente	Ruim	Com Muitas falhas
500.000	Excelente	Não funciona	Não funciona
1.000.000	Ótimo	Não funciona	Não funciona
4.000.000	Ótimo	Não funciona	Não funciona

Fonte: Autoria própria.

Na realização do ataque de negação de Serviço, conseguimos monitorar o resultado do ataque no Raspberry Pi 3 com o Asterisk com a quantidade 1.000.000 pacotes. Foi obtido o resultado de uso de 100 por cento de uma das cpu´s e de toda a memória RAM conforme a Figura 23. Mesmo assim o Raspberry Pi 3 continuou funcionando normalmente. À medida que fomos aumentando a quantidade de pacotes, o funcionamento do Raspberry Pi 3 continuou mostrando ser muito bom.

O Asterisk começou a apresentar problemas a partir de 150.000 pacotes, tendo como consequência falha nas ligações, e a partir de 250.000 pacotes já ficava impossível entender as ligações, ultrapassando essa quantidade de pacotes o Asterisk já não funcionava. O Raspberry Pi 3 se mostrou muito eficiente até a quantidade de 4.000.000 de pacotes até onde realizamos o experimento.

Figura 23 – Uso da Memória e CPU.

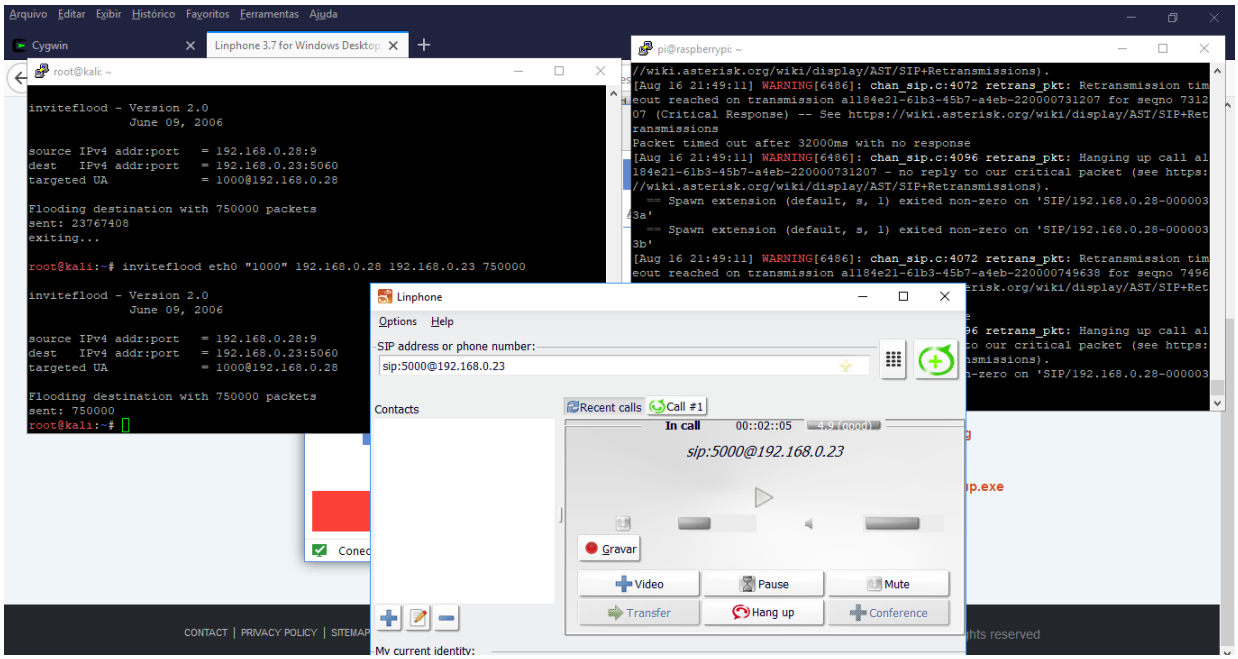


Fonte: Autoria própria.

5.4 Eficiência do processador e Memória nos ataques usando o Zabbix

Nesta seção, vamos mostrar o consumo do processador e da memória na realização dos ataques no Raspberry Pi 3 com o Asterisk em funcionamento com realizações de chamada conforme a Figura 24 . O Raspberry Pi 3, possui um total de 1GB RAM e um processador Broadcom BCM2837 de 64 *bits* e *clock* de 1.2GHz.

Figura 24 – Cenário para coleta de eficiência do Processador e Memória

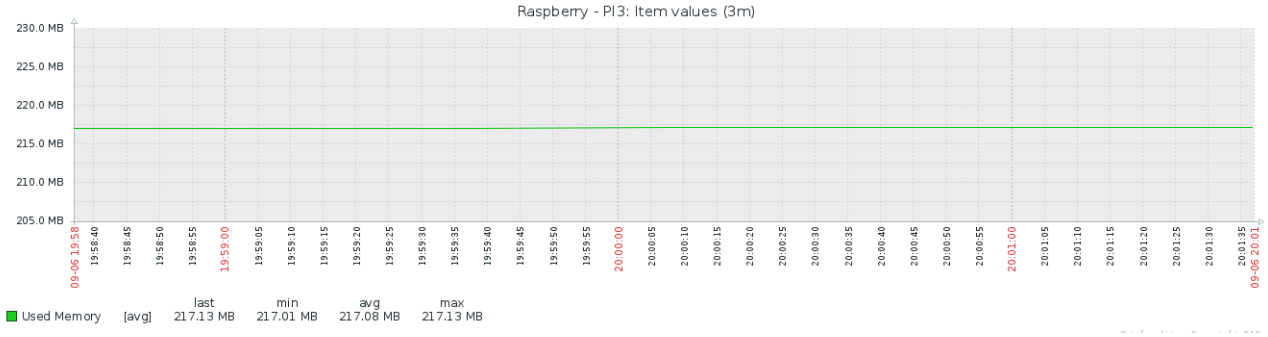


Fonte: Autoria própria.

Neste cenário, conforme Figura 24, utilizamos o Linphone para realizar chamadas durante os ataques juntamente com a maquina virtual para gerar o ataques no Raspberry Pi 3 com o Asterisk. As Figura 25 e Figura 26 no horário 19:58:00 até 20:00p:00, mostram a CPU e a memória antes de qualquer ataque.

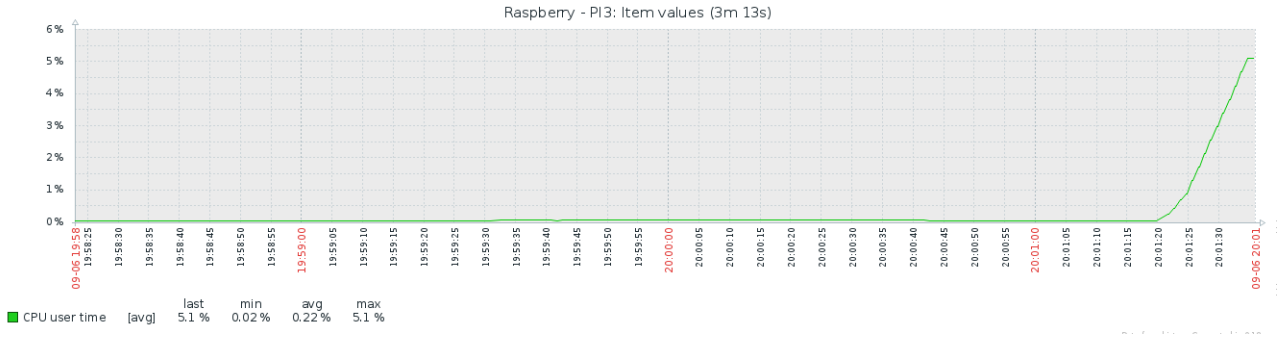
No Ataque de Autenticação que estava sendo realizado exatamente às 20:00:10 e no Ataque Man-in-the-middle que ocorreu exatamente no horário de 20:01:00, não foi possível perceber alteração na memória nem no processador de modo a prejudicar o Raspberry Pi 3. No Asterisk também não houve perda em nenhum momento conforme Figura 25 e Figura 26. Como consequência, as ligações continuaram normalmente.

Figura 25 – Eficiência da Memoria



Fonte: Autoria própria.

Figura 26 – Eficiência do Processador



Fonte: Autoria própria.

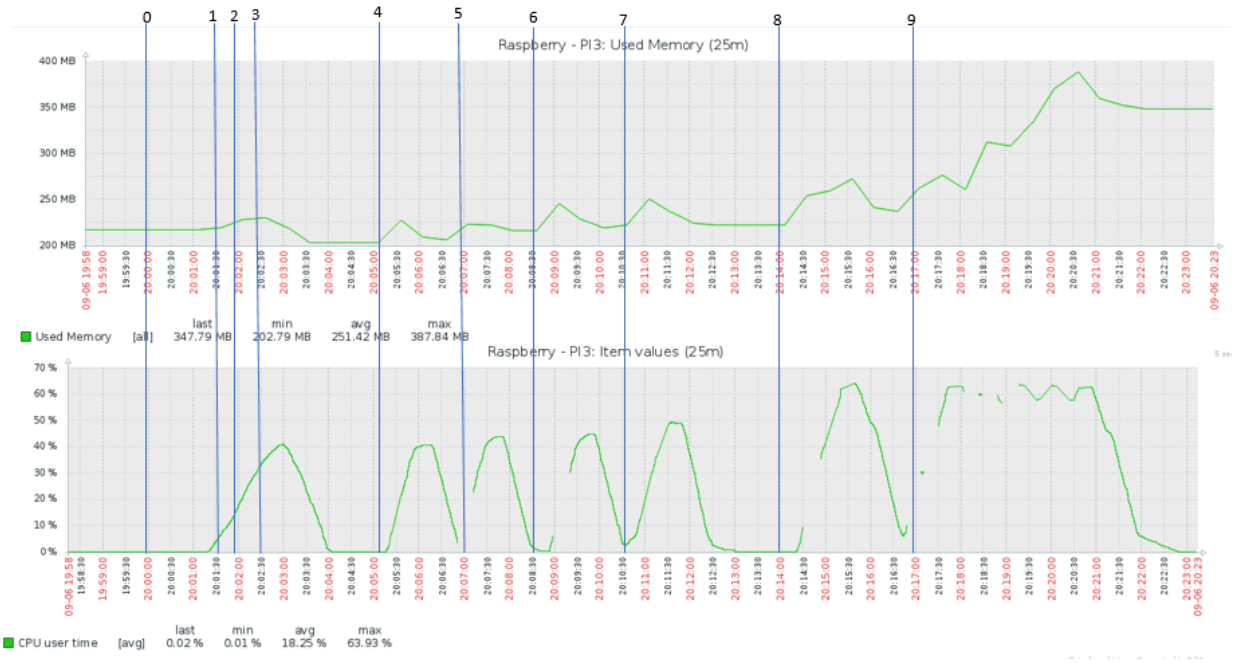
No Ataque Negação de Serviço DoS, realizamos a coleta das informações conforme Tabela 3 para que pudéssemos observar o que acontece em cada quantidade de pacote enviados ao Raspberry Pi 3 com Asterisk.

Na figura Figura 27, podemos observar a evolução do ataque de negação de serviço a começar pelo ponto "0", Figura 27, este sendo o ponto em que o Raspberry Pi 3 com AsterisK ainda não sofreu ataque de negação de Serviço DoS ou seja não foi enviada nenhuma quantidade de pacotes, o que nos mostra que a memoria se encontra entre 200 MB e 250 MB e a CPU entre O e 5 % não demonstrando nem modificação no Raspberry Pi 3, AsterisK e nem problemas nas ligações.

Nos pontos 1 (10.000 pacotes), 2 (50.000 pacotes) e 3 (75.000 pacotes), podemos observar que a memória começa a subir com tendência a chegar ao ponto de 250 MB e a CPU sai de 0% a 40%. Mesmo com essa subida, nem Raspberry Pi 3 com AsterisK são afetados e continuam com seu funcionamento sem nenhuma alteração e as ligações continuam sem nenhum problema.

No ponto 4 (100.000 pacotes), podemos observar que a memoria fica entre os "O MB "e 225 MB descendo sua velocidade aos poucos. Já a CPU chega a ter o seu uso em 40 % e permanecendo com os mesmos por alguns ms (milésimos de segundos) conforme a Figura 27. Com essa quantidade de pacotes também não foi percebida nenhuma alteração no Raspberry Pi 3, AsterisK e nenhum problema nas ligações.

Figura 27 – consumo de Memoria e CPU em Ataque DoS



Fonte: Autoria própria.

No ponto 5 (150.000 pacotes, podemos observar que a memoria fica entre "O MB "e 225 MB descendo sua velocidade aos poucos. Já a CPU chega a ultrapassar o uso de 40 % e permanece com os mesmos por alguns ms (milésimos de segundos) conforme a Figura 27.

Com essa quantidade de pacotes, também não foi percebida nenhuma alteração no Raspberry Pi 3, mas no AsterisK as ligações começaram a ter interferências dificultando a escuta nos contatos provenientes dessa quantidade de pacotes. Perceba que na CPU - Figura 27 existe uma quebra na linha mostrando claramente o momento da dificuldade de transmissão.

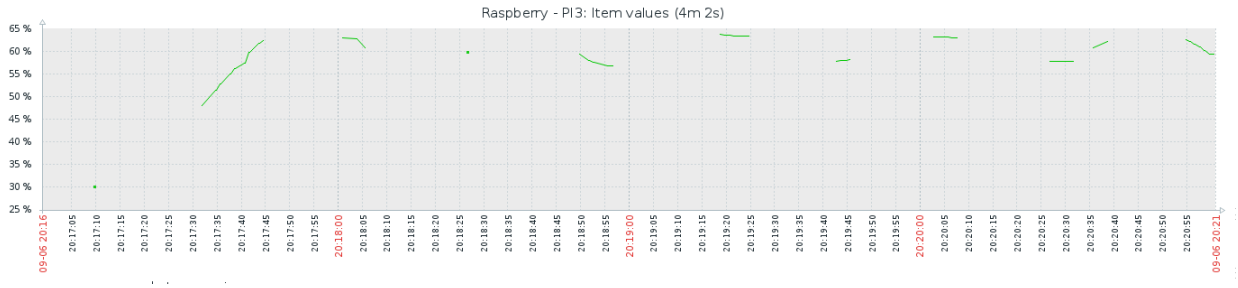
No ponto 6 (250.000 pacotes), foi observado que a memória fica entre "O MB "e chega muito próximo de 225 MB, descendo sua velocidade aos poucos. Já a CPU chega a ultrapassar o uso de 40 % e permanece com os mesmos por alguns ms (milésimos de segundos) conforme a Figura 27.

Com essa quantidade de pacotes, também não foi percebida nenhuma alteração no Raspberry Pi 3, mas no AsterisK foi observado que as ligações começaram a ter muitas interferências, dificultando profundamente a escuta proveniente da quantidade de pacotes. Note que na CPU - Figura 27 existe uma quebra na linha mostrando claramente o momento da dificuldade de transmissão.

Nos pontos 7 (500.000 pacotes), 8 (1.000.000 pacotes) e 9 (4.000.000 pacotes), houve uso da memoria entre 250 MB a próximo de 400 MB, mostrando a força do ataque de negação de serviço DoS. Já a CPU fica oscilando entre 0 % e 65 % aproximadamente. Do ponto 7 e 8 houve uma perda de performance no Raspberry Pi 3. No AsterisK, foi observado que o sistema parou.

Consequentemente, as ligações e os ramais foram desligados em decorrência da quantidade de pacotes. Perceba mais claramente na Figura 28 que existem várias quebras de linha ao serem enviados 4.000.000 de pacotes ao Raspberry Pi 3 com Asterisk.

Figura 28 – 4.000.000 pacotes em CPU em Ataque DoS



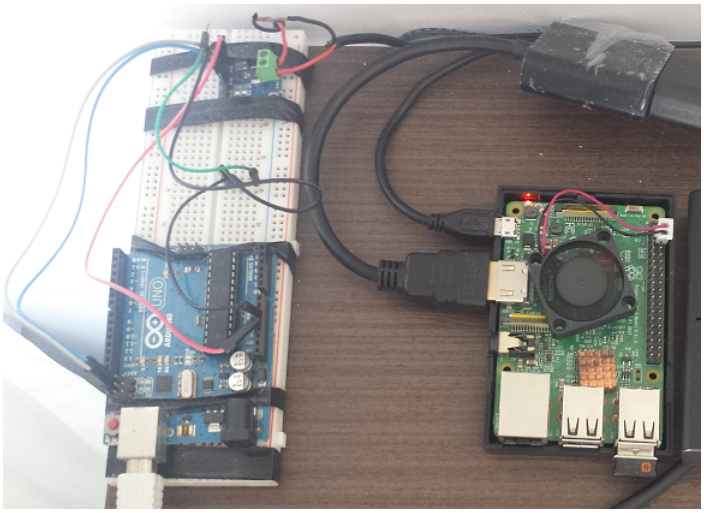
Fonte: Autoria própria.

5.5 Consumo de Energia nos ataques usando Zabbix

O objetivo deste experimento é realizar uma análise de eficiência energética através da medição de corrente e de tensão elétrica no dispositivo embarcado Raspberry Pi 3 com Asterisk no momento de ligações com os ataques de Autenticação, ataque Man-in-the-middle e ataque de negação de serviço DoS.

Para isso utilizou-se um protótipo abordado por Maia (2017), o qual realiza uma medição física. O mesmo é constituído por um dispositivo embarcado Arduino Uno, que realiza a comunicação com a placa de monitoramento de tensão e corrente Adafruit. Essa, por outro lado, utiliza um sensor de corrente e tensão INA219, desenvolvido pela empresa Texas Instruments conforme Figura 29.

Figura 29 – Dispositivo embarcado Arduino Uno com Raspberry Pi 3 com Asterisk.



Fonte: Autoria própria.

A Tabela 4 mostra a coleta da eficiência energética antes de qualquer ataque no Raspberry Pi 3 com Asterisk.

Tabela 4 – Coleta da eficiência energética inicial.

Estado Inicial de Energia		
Voltage (V) Média	**Current (mA) Média**	**Power (mW) Média**
5,21	-600,93	-3132,40
Desvio Padrão		
0,01	5,67	29,08

Fonte: Autoria própria.

No Ataque de Autenticação e no Ataque Man-in-the-middle, não foi percebido um consumo muito diferenciado do estado normal do Raspberry Pi 3 com Asterisk. A Tabela 5 mostra que não houve nada significativo que comprometesse o o funcionamento do Raspberry Pi 3 com o Asterisk.

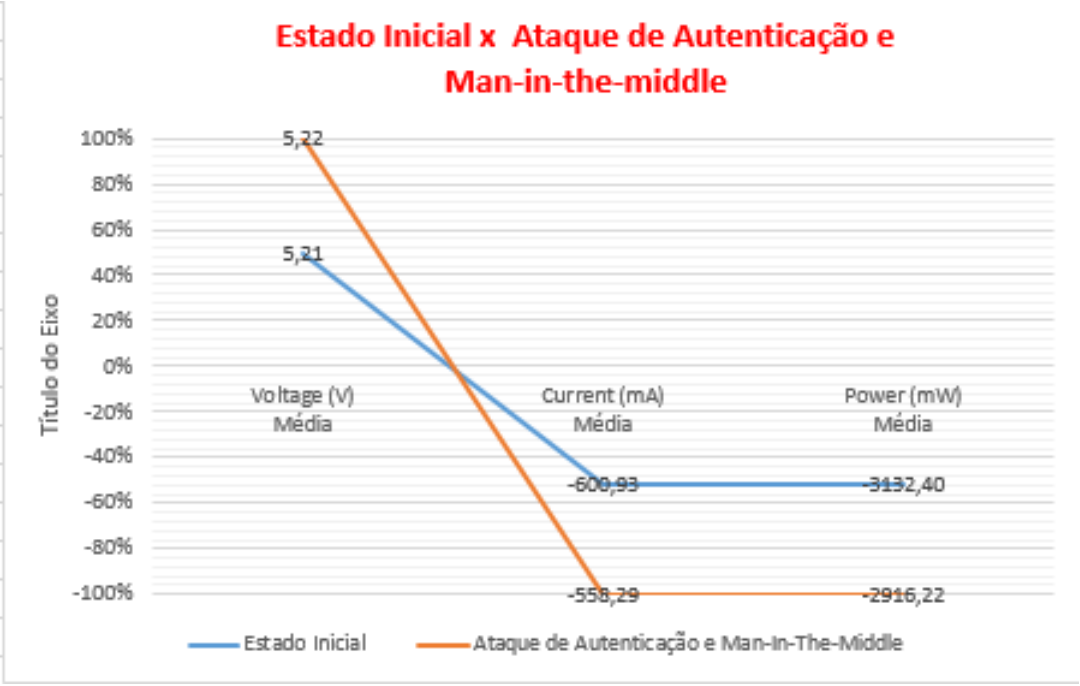
Tabela 5 – Coleta de eficiência energética nos Ataque de Autenticação e no Ataque Man-in-themiddle

Estado de Energia Ataque de Autenticação e Man-in-the-middle		
Voltage (V) Média	Current (mA) Média	Power (mW) Média
5,22	-558,29	-2916,22
Desvio Padrão		
0,00	5,84	30,68

Fonte: Autoria própria.

Na Figura 30 fica bem evidente que não existe uma diferença significativa entre o estado inicial e os ataques Ataque de Autenticação e no Ataque Man-in-the-middle.

Figura 30 – coleta de eficiência energética inicial x coleta de eficiência energética Ataque de Autenticação e no Ataque Man-in-the-middle.



Fonte: Autoria própria.

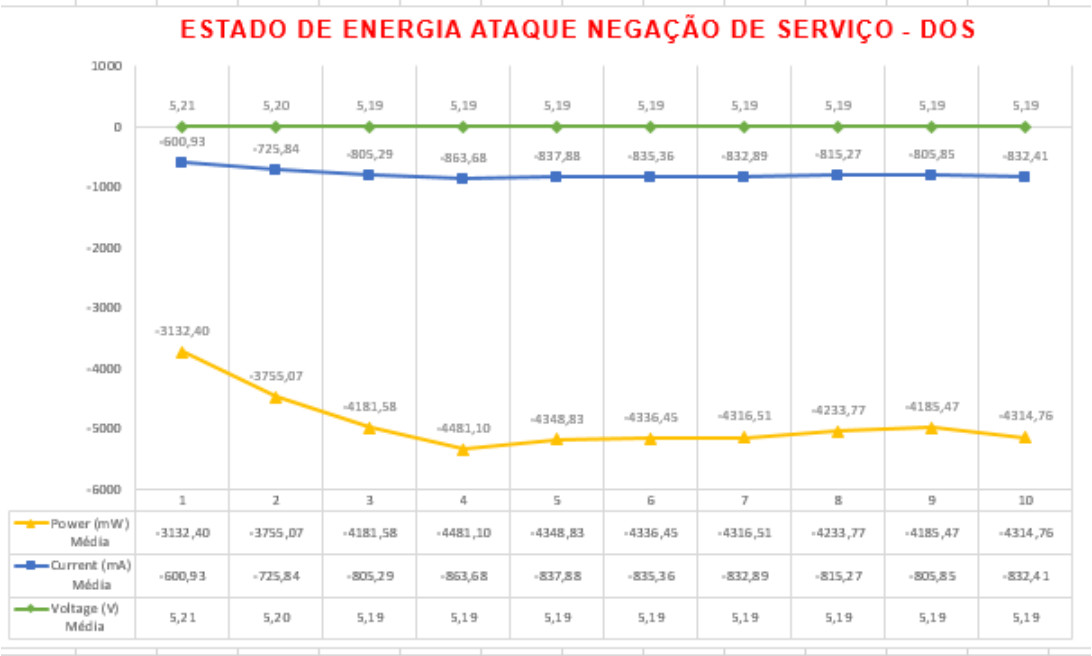
No Ataque Negação de Serviço DoS, utilizamos a medição pela quantidade de pacotes enviados para o Raspberry Pi 3 com Asterisk conforme Tabela 6 e Figura 31. No ataque de negação serviço, podemos observar que quanto mais pacotes são enviados para o Raspberry Pi 3 com Asterisk a voltagem tende a ficar em um único patamar e a current e o power, a variar.

Tabela 6 – Coleta da eficiência energética no Ataque de Negação de Serviço - DoS.

Estado de Energia Ataque Negação de Serviços DoS			
Quant Pacotes	Voltage (V) Média	Current (mA) Média	Power (mW) Média
0	5,21	-600,93	-3132,40
10.000	5,20	-725,84	-3755,07
50.000	5,19	-805,29	-4181,58
100.000	5,19	-863,68	-4481,10
500.000	5,19	-837,88	-4348,83
750.000	5,19	-835,36	-4336,45
1.000.000	5,19	-832,89	-4316,51
1.500.000	5,19	-815,27	-4233,77
10.000.000	5,19	-805,85	-4185,47
25.000.000	5,19	-832,41	-4314,76

Fonte: Autoria própria.

Figura 31 – Eficiência energética no Ataque de Negação de Serviço - DoS.



Fonte: Autoria própria.

6
Conclusão

A segurança em Sistemas Embarcados nem sempre foi levada em conta uma vez que, inicialmente, a maioria deles operavam embutidos em sistemas sem conectividade exterior, como a internet. No entanto novas aplicações que utilizam o conceito de Sistemas Embarcados são dispositivos que precisam se interconectarem à Web via protocolos Internet e diversas conexões sem fio como WiFi, 3G/GPRS e mesmo a Ethernet com fio.

Conforme foi visto nesta dissertação, as vulnerabilidades e ameaças estão por toda a parte, presentes em todos os elementos de infraestrutura que compõem a arquitetura VoIP. Seja um *hardware,* um *software*, um protocolo de comunicação ou mesmo os próprios usuários, todos esses elementos possuem vulnerabilidade que pode ser explorada. Um usuário desatento que acaba fornecendo informações para um invasor, seja um equipamento mal configurado ou sem atualizações, seja falta de conhecimento sobre os riscos dos protocolos e tecnologias utilizados para a implementação do VoIP, todas essas variáveis influenciam na segurança e privacidade das redes VoIP. Aliado ao fato de que aplicações para Sistemas Embarcados são geralmente desenvolvidas em C. A opção por está, se dar por sua eficiência, ou seja, aplicações escritas em C são usualmente mais rápidas e com isso mais adequadas a sistemas com pouco recursos como Sistemas Embarcados. Apesar disso, tal eficiência tem preço. Quando comparada a outras linguagens de programação, C não implementa alguns mecanismos de segurança, o que deixa suas aplicações mais vulneráveis que as demais. Desta forma nesta dissertação, foi realizada uma abordagem de segurança em VoIP usando Asterisk e protocolo SIP em Plataforma Embarcada e uma análise de desempenho e eficiência energética no dispositivo embarcado Raspberry Pi 3 com Asterisk.

Nos Ataque de Autenticação e no Ataque Man-in-the-middle, ficou claro que o Raspberry Pi 3 com Asterisk não faz nenhuma interferência de funcionamento tanto no sistema embarcado como no Asterisk. Já no Ataque de Negação de serviço DoS, o Raspberry Pi 3 se mostrou muito eficiente no ataque, não mostrando perda em seu desempenho nas quantidades de pacotes

enviados neste trabalho.

Conforme Tabela 6 e Figura 31, o Asterisk por sua vez demonstrou que a quantidade de pacotes enviados para o Raspberry Pi 3 influencia no sistema ao ponto de parar todas as chamadas simultâneas, inviabilizando o uso do Raspberry Pi 3 com Asterisk.

Com relação ao consumo de energia nota-se que o Raspberry Pi 3 em sua voltagem tende a ficar em um patamar médio de 5,19v e a Current variando entre -600,93mA a -832,41mA e o Power variando entre -3132,40mV a -4314,78mV tendo com parâmetro a quantidade de pacotes enviados pelo atacante de 0 a 25.000.000.

Raspberry Pi 3 com Asterisk mostra que o dispositivo é muito eficiente, mas o Asterisk não. Sendo assim existe a necessidade do dispositivo ter um sistema de segurança embutido que venha a garantir a segurança no Raspberry Pi 3 com Asterisk. Isso pode prejudicar o desempenho, tendo em vista que vai necessitar do uso de mais memória e processador.

6.1 Trabalhos Futuros

Uma abordagem de segurança em VoIP usando Asterisk e protocolo SIP foi realizada neste trabalho com o intuito de alertar a comunidade para melhorar a segurança nos dispositivos embarcados, pois novos dispositivos irão surgir com capacidades muito maiores do que o estudado. Dessa forma, ampliam-se as possibilidades de estender este trabalho.

Como trabalhos futuros, propõe-se a realização de mais 3 ataques aos dispositivos e verificar o seu comportamento: pode ser realizada uma comparação de ataques em outros dispositivos com o uso do *software* Asterisk e também serem verificados os ataques utilizando outro protocolo como, por exemplo, o protocolo H.232.

6.2 Artigos Publicados

Apêndice A - A Security Approach using SIP Protocol in Imbedded Systems. Nogueira, T., Menezes, A., Ribeiro, A. and Ordonez, E. DOI: 10.5220/0006354403520355 In Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017), pages 352-355 ISBN: 978-989-758-246-2 Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

Apêndice B - An Approach to the Performance and Efficiency Power Analysis on Embedded Devices Using Asterisk. Journal of Computer Science. © 2018 Adauto Cavalcante Menezes, Toniclay Andrade Nogueira, Edward David Moreno Ordonez and Admilson de Ribamar Lima Ribeiro. This open access article is distributed under a Creative Commons Attribution (CC-BY) 3.0 license. Adauto Cavalcante Menezes et al. / Journal of Computer Science 2018, 14 (7): 1038.1052 DOI: 10.3844/jcssp.2018.1038.1052

Referências

AKYILDIZ, I. et al. Wireless sensor networks: a survey. *Computer Networks*, v. 38, n. 4, p. 393–422, 2002. ISSN 13891286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128601003024>. Citado na página 13.

ALHAZMI, O. H.; MALAIYA, Y. K.; RAY, I. Measuring, analyzing and predicting security vulnerabilities in software systems. *Computers and Security*, v. 26, n. 3, p. 219–228, 2007. ISSN 01674048. Citado na página 14.

BALL, S. R. *Embedded Microprocessor Systems: Real World Design*. 3. ed. EUA: [s.n.], 2005. 1–28 p. ISSN <null>. ISBN 9780750675345. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780750675345500230>. Citado na página 21.

BANSAL, A.; PAIS, A. R. Mitigation of Flooding Based Denial of Service Attack against Session Initiation Protocol Based VoIP System. In: *2015 IEEE International Conference on Computational Intelligence & Communication Technology*. IEEE, 2015. p. 391–396. ISBN 978-1-4799-6023-1. Disponível em: <http://ieeexplore.ieee.org/document/7078732/>. Citado 2 vezes nas páginas 35 e 36.

BARBOSA, C. S. VoWLAN : Voz sobre IP em Redes Locais Sem Fio. *Network*, Centro Federal de Educação Tecnológica de Goias, Goiânia, n. Ii, p. 1–17, 2006. Citado na página 25.

BARR, M.; REILLY, P. O. *Programming Embedded Systems in C and C++*. [S.l.: s.n.], 1999. 1–187 p. ISBN 1565923545. Citado 2 vezes nas páginas 14 e 15.

Brito S. H. B. *Aspectos de Segurança e Sigilo em Comunicações VoIP*. São Paulo: [s.n.], 2011. 13 p. Citado 2 vezes nas páginas 27 e 30.

BUTCHER, D.; LI, X.; GUO, J. Security challenge and defense in VoIP infrastructures. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, v. 37, n. 6, p. 1152–1162, 2007. ISSN 10946977. Citado na página 29.

CARRO, L.; WAGNER, F. R. Sistemas Computacionais Embarcados. *XXII Jornadas de Atualização em Informática - JAI*, p. Capítulo 2, 2003. Citado 2 vezes nas páginas 14 e 15.

CHASE, O. Sistemas Embarcados. *SBAJovem 2010*, p. 7, 2007. Citado na página 22.

ČÍŽ, P. et al. VoIP Intrusion Detection System with Snort. *ELMAR, 2012 Proceedings*, n. September, p. 137–140, 2012. ISSN 13342630. Citado 2 vezes nas páginas 34 e 36.

COLHER, S. et al. *VoIP: Voz sobre IP*. 3. ed. Rio de Janeiro: [s.n.], 2005. 288 p. ISBN 9788535217878. Citado na página 13.

CUERVO, F. et al. *Megaco Protocol Version 1.0 RFC 3015 (Proposed Standrd). Obsoleted by RFC 3525.* [S.l.: s.n.], 2000. Citado na página 25.

CUNHA, A. Sistemas Embarcados. *Revista Saber Eletrônica 414*, Revista Saber Eletrônica 414, 2007. Citado na página 21.

Dalle Vacche, A.; Kewan Lee, S. *Zabbix Network Monitoring Essentials*. Birmingham: [s.n.], 2015. 178 p. ISBN 978-1784399764. Citado na página 31.

DASTERISK. *Definição de asterisk*. 2016. Disponível em: <http://www.asterisk.org>. Acesso em: 23 maio 2018. Citado na página 23.

DEFSIP. *Definindo o que é um protocolo de sinalização*. 2006. 01 p. Disponível em: <https://www.gta.ufrj.br/grad/06{_}1/sip/Definindooqueumprotocolodesinalizao.h>. Acesso em: 20 dez 2017. Citado na página 25.

GRECCO, F. Revista de TI. p. http://www.timaster.com.br/revista/artigos/main_ar, 2004. Disponível em: <http://www.timaster.com.br/revista/artigos/main{_}artigo.asp?codigo=>. Citado na página 20.

GROSS, F. D. *VoIP com Asterisk*. 1. ed. São PauloSão Paulo: Linux New Media do Brasil Editora Ltda. Coleção Academy, 2011. Citado 3 vezes nas páginas 24, 25 e 29.

HAMACHER, C. et al. *Computer Organization and Embedded Systems*. 6. ed. [S.l.]: Mc Graw-Hill, 2012. ISBN 978-0073380650. Citado na página 15.

HERTZOG;, R.; O'GORMAN;, J.; AHARONI, M. Kali Linux Revealed Mastering the PenetrationTesting Distribution. *European Journal of Organic Chemistry*, v. 2012, n. 14, p. 2756–2765, may 2012. Disponível em: <http://doi.wiley.com/10.1002/ejoc.201200111>. Citado 2 vezes nas páginas 30 e 31.

International Telecommunications Union. Specifications of Signalling System No. 7. v. 1, 1993. Citado na página 20.

JONES, J. R. Estimating software vulnerabilities. *IEEE Security and Privacy*, v. 5, n. 4, p. 28–32, 2007. ISSN 15407993. Citado na página 14.

KELLER, A. Asterisk na prática. v. 2, p. 336, 2011. Citado 2 vezes nas páginas 19 e 25.

KUHN, D. R.; WALSH, T. J.; FRIES, S. Security Considerations for Voice Over IP Systems Recommendations of the National Institute of Standards and Technology Voice Over IP Systems. *Nist Special Publication*, 2005. Citado na página 13.

LOMOTEY, R. K.; DETERS, R. Intrusion Prevention in Asterisk-Based Telephony System. In: *2014 IEEE International Conference on Mobile Services*. IEEE, 2014. p. 116–123. ISBN 978-1-4799-5060-7. Disponível em: <http://ieeexplore.ieee.org/document/6924302/>. Citado 2 vezes nas páginas 33 e 36.

MAIA, W. P. *PROJETO, IMPLEMENTAÇÃO E DESEMPENHO DOS ALGORITMOS CRIPTOGRÁFICOS AES, PRESENT E CLEFIA EM FPGA*. Aracaju: Universidade Federal de Sergipe, 2017. Disponível em: <https://ri.ufs.br/bitstream/riufs/5029/1/WILLIAM{_}PEDROSA{_}MA>. Citado na página 54.

MARWEDEL, P. *Embedded System Design*. [s.n.], 2011. 258 p. ISBN 9780387300870. Disponível em: <http://books.google.com.br/books?id=DZEoizXV1swC>. Citado na página 15.

MCGRAW, G. *Software security: building security in*. [S.l.]: Addison-Wesley Professional., 2006. Citado na página 14.

MINOLI, D. *Delivering Voice Over IP Networks*. 2. ed. Indiana: [s.n.], 2002. Citado na página 26.

NAKAMURA, E. T.; GEUS, P. L. de. *Segurança de rede em ambientes corporativos*. [S.l.]: Novatec, 2007. 482 p. ISBN 9788575221365 8575221361. Citado na página 26.

RAAKE, A. *Speech quality of VoIP: assessment and prediction*. [S.l.: s.n.], 2006. 336 p. ISBN 978-0-470-03060-8. Citado na página 18.

RAFAEL SEIDI SHIGUEOKA. *ANÁLISE COMPARATIVA DE TÉCNICAS PARA OCULTAMENTO DE PERDAS DE PACOTES EM APLICAÇÕES DO TIPO VOZ SOBRE IP (VOIP)*. 40 p. Tese (Doutorado) — Universidade Estadual de Londrina, 2016. Disponível em: <http://www.uel.br/cce/dc/wp-content/uploads/VersaoPreliminarTCC-RafaelShigueoka.pdf>. Citado na página 19.

REHMAN, U. U.; ABBASI, A. G. Security analysis of VoIP architecture for identifying SIP vulnerabilities. In: *2014 International Conference on Emerging Technologies (ICET)*. Islamabad, Pakistan: IEEE, 2014. p. 87–93. ISBN 978-1-4799-6089-7. Disponível em: <http://ieeexplore.ieee.org/document/7021022/>. Citado 2 vezes nas páginas 34 e 36.

REIS, C. *Sistemas Operacionais para Sistemas Embarcados"*. [S.l.]: ED-UFBA; BRASIL, 2004. Citado na página 21.

SINNREICH, H. *Internet communications using SIP: Delivering VoIP and multimedia services with session*. Indianapolis: Wiley Publishing, 2006. Citado na página 26.

SIQUEIRA, F. T. et al. Desenvolvimento de Sistemas Embarcados para Aplicacões Críticas. 2006. Citado na página 22.

SITOLINO, C. L. *Voz sobre IP – Um estudo experimental*. 1999. http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana p. Disponível em: <http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana99/sitolino/sitolino.html>. Acesso em: 16 ago 2018. Citado na página 13.

STAPKO, T. *Practical Embedded Security: Building Secure Resource-Constrained Systems*. [S.l.]: Embedded technology series. Elsevier Science., 2011. Citado na página 14.

TANENBAUM, A. S. *Redes de Computadores*. Rio de Janeiro: [s.n.], 2003. 946 p. ISBN 8535211853. Citado na página 21.

THERMOS, P. *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures*. boston. [S.l.]: Pearson Education, Inc., 2007. ISBN 978-0321437341. Citado 3 vezes nas páginas 27, 28 e 29.

THERMOS, P.; ARI, T. *VOIP Network: Theats, Vulnerabilities and Countermeasures*. Boston, MA, USA: [s.n.], 2008. Citado 2 vezes nas páginas 27 e 28.

VOLCATEC. *VOLCATEC*. 2016. http://www.vocaltec.com p. Disponível em: <http://www.vocaltec.com>. Acesso em: 13 jul 2017. Citado na página 18.

WALKER, J. Q.; HICKS, J. T. *Taking Charge of Your VoIP Project*. 1. ed. [S.l.: s.n.], 2004. 312 p. ISBN 10: 1-58720-092-9. Citado na página 18.

WILLIAM; STALLINGS. *Criptografia e Segurança de redes, PRINCIPIOS E PRÁTICAS*. 6. ed. [S.l.]: Pearson Education, Inc., 2015. 557 p. ISBN 9788543014500. Citado na página 14.

YOSHIOKA, S. *Aspectos de Segurança para Telefonia IP utilizando o Protocolo SIP*. Campinas: UNICAMP, 2003. 73 p. Citado na página 30.

ZAPALS. *Raspberry Pi 3 Model B Motherboard*. 2018. Disponível em: <https://www.zapals.com/raspberry-pi-3-model-b-motherboard-on-board-wi-fi-bluetooth-development-board-rs-original-uk-version.html>. Acesso em: 18 jan. 2018. Citado na página 23.

Apêndices

APÊNDICE A – WEBIST 2017 - B3

A Security Approach using SIP Protocol in Imbedded Systems

Toniclay Andrade Nogueira, Adauto Cavalcante Menezes,
Admilson De Ribamar Lima Ribeiro and Edward David Moreno Ordonez
Computing Department, Segipe Federal University, Aracaju, Brazil

Keywords: VoIP, Asterisk, Embedded Systems, SIP, Safety.

Abstract: Voice over IP communication will dominate the world. However, given the growing demand for voice and data communication to make any and all communication reliable and secure, several attacks occur frequently in communication networks, so this work is based on verifying security, analyzing risks, vulnerabilities, such as verifying the attacks and proposing a security measure for voice over IP communication on embedded devices.

1 INTRODUCTION

The Voice Over Internet Protocol (VoIP) technology consists in the integration of the services in the telecommunication areas and the network services provided by computers. In this way, it enables the digitization and encoding of the voice signal and transforms it into data packages for communication in a network using UDP protocols.

In this context, the VoIP concept allows cost reduction in installations, maintenance and management of parallel networks. With this, a new concept of telephony is created (Sitolino 1999). However, it will be necessary equipment, techniques and specific human resources (Silva, 2016).

Stapko (2007) understands as information security the protection of personal or confidential information, as well as the computational resources of individuals or organizations. Without information security, malicious individuals can destroy or use such information for malicious purposes. The state-of-the-art security in VoIP telephony involves audio encryption between the two distinct points as well as interoperability between communication server manufacturers through indecipherable encryption and centralized management (Stallings, 2008).

According to Barr (1999), Car and Wagner (2003) and Marwedel (2011), embedded systems must be reliable, since failures can compromise their functionality and make system recovery unfeasible. Embedded systems use hardware platforms, which are driven by softwares. Several implementations of processors can be used, which implies a great reduction of costs.

Some reliability issues are found on embedded devices, as they cannot be safely shut down for repairs, the system must run continuously.

As its operating mode has reduced performance, the environment tends to fail if it is turned off (Akyildiz, 2002). Security in embedded systems was not always considered, since most of these systems were initially operated without Internet connectivity.

Information security and new embedded device paradigms are increasingly present in our lives. However, the communication between devices will have a great impact on global communication, which will increase the efficiency and security of VoIP communication.

This article is organized as it follows: section 2 is composed of theoretical grounding presentation, section 3 presents related works, section 4 has a description of the proposal, and section 5 show the expected contribution of the research.

2 THEORETICAL FOUNDATION

2.1 VoIP

According to Raake (2007) and Walker, (2004), VoIP is a technology that performs voice communication over an IP network.

The communication process consists of transforming the analog voice into digital, through the

Nogueira, T., Menezes, A., Ribeiro, A. and Ordonez, E.
A Security Approach using SIP Protocol in Imbedded Systems.
DOI: 10.5220/0006354403520355
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 352-355
ISBN: 978-989-758-246-2

fragmentation of the package and transport over the IP network. The process is becoming more modern, it is possible to mention some softwares that work with this technology, among them, Facebook, Messenger, Skype, Viber and WhatsApp.

In image 01 we can observe the operation of a VoIP application where the analog audio is converted into digital and grouped into packages that are transmitted to the IP network through the Real Time Protocol (RTP) protocol, after arriving at the receiver the packages are organized and then reproduced.

Figure 1: Scenario of the ideal operation of the VoIP application, Shigueoka, 2016.

2.2 Embedded Systems

Some data researched in high technology shows that more than 90% of microcomputers manufactured in the world are intended for machines that are not called computers, such as cell phones, automobiles, DVD players, among others.

According to Reis (2004), what comes to differentiate the set of devices from a computer is the project based on a dedicated set and specialist, consisting of Hardware, Software and Peripherals, i.e., embedded system.

For Ball (2005), the system is classified as embedded when it is dedicated to a single task and continuously interacts with the environment around it, by the use of actuators and sensors.

In their article, Siqueira, Menegotto, Weber, César Netto and Wagner (2006) comment on the use of embedded systems in critical applications, which comprises as applications in which the risks associated with the hazards involved are considered unacceptable and need to be handled.

The embedded system is commonly a solution formed from dedicated and specific microcontroller and software to performing the operational functions of equipment for which it was designed.

2.3 Session Initiation Protocol (SIP)

SIP has been developed to facilitate the implementation of the basic aspects of a session, which is a non-trivial process. Today it is used worldwide and it is also a strong “competitor” of H.323. Barbosa (2006) defines SIP as a protocol that signals client-server sessions, and that stands out for its simplicity and mobility; it has a primitives the initialization, modification and termination of sessions in a VoIP communication.

According to Defsip, together with Real-time Transport Protocol (RTP), Real Time Streaming Protocol (RTSP), Session Description Protocol (SDP), SIP establishes a complete multimedia architecture, providing complete services to the user.

SIP also provides participant management services in a multimedia session.

According to Cuervo (2000), due to the ability of working in conjunction with other protocols, it allows integration with public telephony, allowing not only the connection between IP extensions, but also for public network telephones.

2.4 Types of Attacks to SIP Protocol

2.4.1 Main-in-the-middle

For this attack, the attacker can use two techniques: ARP table poisoning, or DNS cloning. With either of these, permission is granted to be between the SIP server and the User Agent. With this type of attack, the intruder does not necessarily know valid usernames and passwords; they can simply route traffic between the server the and client and act intercepting the packages, preventing them from reaching their real destination, which is the SIP server (Nakamura, Emílio, Geus and Lício, 2007).

2.4.2 Subsection Titles

According to Thermos (2007), this attack aims to obtain credentials from valid users in a SIP telephony communication system using a brute-force attack, which is, sending multiple ID requests and passwords to from a dictionary.

2.4.3 Denial of Service

In attacks known as Denial of Service, it is possible to layers of infrastructure in VoIP environment. According to Thermos (2007) the DoS attacks have as main objective to cause the interruption of the target service. In this case, the attack is directed to both the operating system and also to the network services.

2.4.4 SIP Redirect

For Butcher, Li and Guo (2007), the attack employs a server that receives requests from a telephone or proxy and returns a redirect response, which indicates

where the request is to be repeated, thus enabling users to have a call redirected to another location rather than where they are located. However, the caller normally dials only the number to reach the user.

The attacker redirects the victim's calls to a specific number, so the attacker starts receiving the calls that were forwarded to the attacked user.

3 RELATED WORK

The work covered in this article contains a large amount of research. Thus, the IEEE Base has been used with works from the year 2012 to 2016.

3.1 The Communication System

In their article, Lomotey and Deters (2014) show that IP communication systems have been the target of attacks, such as call theft, attacks on servers, which allows access to users' data. Thus, the author proposed a solution to prevent the intrusion of attackers in the communication system built in VoIP Asterisk.

In his experiment, a complete platform for Asterisk was not used because he proposed a cloud-based middleware, which layer maintains the most sensitive part of the information call.

3.2 VoIP Security Analyses

Rehman and Abbasi (2014) analyzed security in the VoIP architecture with the Asterisk voice over IP communication system. It has been noted that most of the attacks are related to the fragility of the SIP protocol, espionage attacks, modification and involuntary interruption were detected.

The authors have proposed as solution of the presented problem, an efficient and secure mechanism of authentication for the protocol SIP, with this, it is possible to assure greater protection to the attacks.

It was suggested to assign a cryptographic token that would authenticate the users allowing their identification and providing greater security as well as there would be the need for the user to enter a password to use other available services.

3.3 VoIP Intrusion Detection with Snort

Číž, Lábaj, Podhradský and Londák (2012), have proposed in their experiment a model focused on DoS attack in order to cause a malfunction in Asterisk voice over IP communication software. The authors used the SIP tool, in order to verify the functionality of the detection system and cause anomalies in denial of service attacks, the Snort software tool was also used to detect open network attacks, capable of performing analysis of the Traffic and packet logging on IP networks.

4 DESCRIPTION OF THE PROPOSAL

The present proposal aims at creating a defense method for the IP asterisk over SIP protocol, in order to use embedded devices (Raspberry Pi3, Banana Pi M3 and Orange Pi Plus 2) as shown in image 2. The method will be based on the main attacks that occur in embedded systems, contemplated by authors in related works in diverse bases.

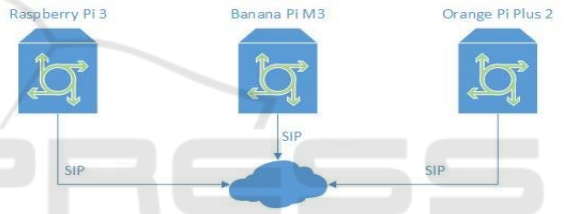


Figure 2: Architecture of the scenario.

With the result of the main attacks, simulations will be made in the device in order to propose security methods.

It will be necessary to study the SIP protocol to verify the vulnerabilities in order to apply the best configuration and defense methods to ensure the security of the device.

In order to achieve the objectives of this research, it will be necessary to elaborate a scenario that makes possible to carry out all the experiments as close as possible to a real production environment, so the scenario should include three low cost embedded devices already configured with the system, which must be directly connected to the Internet.

5 EXPECTED CONTRIBUTION

This proposal presents as main contribution the elaboration of a security method for a VoIP communication central in an embedded device using the Asterisk system.

With the development of this proposal we intend

to obtain the following contributions: to survey the main techniques used to attack the communication systems, to survey the tools and materials necessary to simulate the most significant attacks on embedded devices; to perform a literature review of the SIP protocol, to analyze the vulnerabilities of the SIP protocol, to propose a defense for these attacks, to write the dissertation and present the results of the security analysis on the devices shipped with Asterisks.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks, 38(4):393–422.*

Ball, Stuart - ”Embedded Microproceessor Systems: Real Woard Desing”, *3º edition,* Editora:Mcpros, EUA, 2005.

Barbosa, Camila Soares. *Voz sobre IP em Redes Locais sem Fio*. CEFET – Centro Federal de Educação Tecnológica de Goiás. Goiânia, (2006).

Barr, M. (1999). *Programming embedded systems in C and C++*. O'Reilly.

Carro, L. and Wagner, F. R. (2003). Sistemas computacionais embarcados. *Jornadas de atualização em informática. Campinas:* UNICAMP.

Číž, P., Lábaj, O., Podhradský, P., & Londák, J. (2012). VoIP Intrusion Detection System with Snort. *In ELMAR, 2012 Proceedings (pp. 137-140).* IEEE.

Cuervo, F., Greene, N., Rayhan, C., Rosen, B., and Segers, J. (2000). *Megaco Protocol Version 1.0 RFC 3015 (Proposed Standrd)*. Obsoleted by RFC 3525.

D. Butcher, X. Li, and J. Guo. Security challenge and defense in VoIP infrastructures. *IEEE Transactions on Systems, Man, and Cybernetics, Part C:* Applications and Reviews, 37(6):1152–1162, November 2007.

DEFSIP. *Definindo o que é um protocolo de sinalização*. Disponível em http://www.gta.ufrj.br/grad/06_1/SIP/Definindooqueumprotocolodesinalizao.html, acessado em 27 nov 2016.

Lomotey, R. K. and DETERS, R. (2014). Intrusion Prevention in Asterisk-Based Telephony System. In 2014 *IEEE International Conference on Mobile Services, pages 116–123*.IEEE.

Marwedel, P. (2011). Embedded system design. Springer.

Nakamura, Emílio T.; *GEUS, Paulo Lício de. Segurança de rede em ambientes corporativos.* São Paulo: Novatec Editora, 2007.

Raake, *A. Speech quality of VoIP: assessment and prediction. [S.l.]:* John Wiley & Sons, 2007.

Rehman, U. U. and Abbasi, A. G. (2014). Security analysis of VoIP architecture for identifying SIP vulnerabilities. *In 2014 International Conference on Emerging Technologies (ICET), number i, pages 87–93*. IEEE.

Reis, Claiton – *"Sistemas Operacionais para Sistemas Embarcados",* Tutorial, Editora: EDUFBA, BRASIL, 2004.

Silva, Adailton. Qualidade de Serviço em VoIP – *Rede Nacional de Ensino e Pesquisa.* Maio/2000 – Disponível em: http://www.rnp.br/newsgen - Acessado em 18/11/2016.

Siqueira, Tórgan Flores de ; Menegotto, C. C. ; Weber, T. S. ; César Netto, João ; Wagner, F. R. . Desenvolvimento de Sistemas Embarcados para Aplicações Críticas. In: *Escola Regional de Redes de Computadores, 2006, Passo Fundo*. Escola Regional de Redes de Computadores. Porto Alegre : Sociedade Brasileira de Computação, 2006. v. 1. p. 1-10.

Sitolino, Cláudio Luiz., Voz sobre IP – *Um estudo experimental 1999.* http://www.inf.ufrgs.br/pos/SemanaAcademica/Semana99/sitolino/sitolino.html. Acessado 16/08/2016.

Stallings, W. (2008). *Criptografia e segurança de redes: princípios e práticas.* PRENTICE HALL BRASIL.

Stapko, T. (2011). Practical Embedded Security: Building Secure Resource-Constrained Systems. *Embedded technology series.* Elsevier Science.

Thermos, P.; Takanen, A. *Securing VoIP networks: threats, vulnerabilities, countermeasures.* Boston: Pearson Education, 2007.

Walker, J. Q.; HICKS, J. T. *Taking charge of your VoIP project. [S.l.]:* Cisco Press, 2004.

APÊNDICE B – Journal of Computer Science 2018 - B1

Journal of Computer Science

Original Research Paper

An Approach to the Performance and Efficiency Power Analysis on Embedded Devices Using Asterisk

Adauto Cavalcante Menezes, Toniclay Andrade Nogueira, Edward David Moreno Ordonez and Admilson de Ribamar Lima Ribeiro

Department of Computing, UFS - Segipe Federal University, Aracaju, Brazil

Article history
Received: 21-05-2018
Revised: 02-06-2018
Accepted: 01-08-2018

Corresponding Author:
Adauto Cavalcante Menezes
Department of Computing, UFS - Segipe Federal University, Aracaju, Brazil
Email: adauto.cavalcant@gmail.com

Abstract: Voice over IP (VoIP) communication will dominate the computing world for years to come. In order to perform VoIP communication, it is necessary to encode and decode the voice. This process consumes the main computational resources, as an example, it is possible to mention the processor and memory. The telecommunication industries provide equipment with high purchasing prices, which makes the access to this technology still very restricted. Embedded devices are purposely constructed for certain applications, they execute systems with high criticality complexity. Asterisk is a free software for voice over IP communication and its main function is to implement the functions of a telephone exchange. These technologies promise to reduce costs and maximize results. This work describes a performance analysis on three modern embedded devices (Raspberry Pi 3, Orange Pi Plus 2 and Banana Pi M3) using the Asterisk voice over IP communication system. The performance analysis consists of evaluating the jitter, delay and bandwidth, as well as the number of concurrent calls supported in each device with SIP and lAX2 protocols with CODEC's G.711a, G.711u, Gsm, Speex, Ilbc, G.722 and in parallel, monitor the RAM memory consumption, processing and energy. The results show that the Raspberry Pi 3 and the Banana Pi M3 devices support in a satisfactory manner a high number of simultaneous calls with moderate memory, processing and energy consumption. However, the Orange Pi Plus 2 device showed high processing consumption.

Keywords: VoIP, Asterisk, Embedded Devices, Performance Analysis, Embedded Systems

Introduction

The term Voice over Internet Protocol (VoIP) is conceptualized as the voice communication in networks that use the Internet Protocol (IP), which was developed with the emergence of IP Telephony, which consists of the provision of telephony services using the IP network for the establishment of calls and voice communication (Bernal, 2007). In the middle of 1990, the definition of VoIP was consolidated, when emerged the Internet Phone from VocalTec Communications, the first commercial software that enabled the communication of voice over IP, but with poor communication quality (Colcher *et al*., 2005).

For Androulidakis (2016), private telephone exchanges serve to communication between internal telephones and communication with the public telephone network. There are IP communication systems or also called Private Branch Exchange (PBX) IP and Time Division Multiplexing (TDM) communication systems or conventional TDM PBXs. Their software can be offered through proprietary or open source. Communication with an external medium depends on the interface for connecting analog or digital lines, usually provided by telecommunication operators; and the communication with internal telephones depends exclusively on the central office itself. Initially, the main advantage of the use of PBX communication systems was the cost of calling internal lines, as there is an internal switching of circuits, which makes the call free.

Also according to Androulidakis (2016), voice communication systems have gained popularity and now have functionalities, services that were not available from telecom operators such as hunt groups, call

forwarding and dial-by-extension. According to Sulkin (2002), the trend is to migrate to the IP telephony.

In 1999, the software for communication of the voice over IP Asterisk emerged. According to Bryant *et al.* (2013), Asterisk is a free software that performs all the functions of a conventional telephone exchange, developed by Digium. Currently, it receives several contributions from developers around the world, as it is a promising area of application that is constantly in development.

The telecommunication industries provide equipment with high values and mostly proprietary equipment. This makes it difficult to access VoIP technology.

Thus, it is necessary to provide a mechanism to reduce the expenses in the area of telephony, preferably with characteristics that are similar to that of a conventional telephone exchange; it is also necessary to expand and encourage the knowledge of students and researchers in the area of information technology and telecommunications. In this context, the importance of efficiency and cost reduction of the new generation of revolutionary personalized systems for voice over IP communication is justified. Thus, it is reinforced the idea that any quality advance in this technology can propitiate a considerable increase in the quality of the voice communication, both in the academic environment, as in the telecommunications industry, thus, we have the motivation for the aim that is to accomplish this work.

From Asterisk, it is possible to implement a voice communication system on embedded, low-cost devices that provides the necessary mechanisms of a conventional telephone exchange. For this, a performance analysis should be run on three modern embedded devices (Raspberry Pi 3, Orange Pi Plus 2 and Banana Pi M3), using the Asterisk voice over IP communication system. For this, we must perform an analysis of the behavior of the IP phone calls on the devices through research jitter, delay and bandwidth, as well as measuring the number of concurrent calls supported on each device with the SIP and IAX2 protocols with different CODECs and in parallel monitoring the RAM, processing and power consumption. At the end, we would be able see which implementation has performed best to support the Asterisk voice over IP communication system.

Related Work

In this section, all works the classified as significant and related to the present study are discussed. A systematic analysis was carried out in order to find relevant works in bases considered important in the area of computing (IEEE Xplore, Science Direct and the Brazilian Digital Library of Computation). The amount of work that Asterisk addresses in embedded devices is greatly reduced. In this way, the work on the performance analysis with the use of the SIP and IAX2 protocol was also researched, thus, it was possible to know the measurement techniques currently used.

In a paper entitled ”Performance Analysis of VoIP Services over WiFi-based systems”, the authors Villac´ıs *et al.* (2013) presented an Alix hardware performance analysis for usage in VoIP systems under Wireless Fidelity (Wifi) networks, a server with Embedded Asterisk software is adopted. This analysis consists of tests of the Central Processing Unit (CPU) and RAM memory with a fixed number of simultaneous calls with the SIP and IAX2 protocols and with the CODECs GSM, G.711 u/a, SPEEX and G.726. Still in Villac´ıs *et al.* (2013), the authors could perform an energy efficiency analysis at the time of a high number of simultaneous calls, as well as conduct simultaneous calls behavior tests in an interconnection between Asterisk servers, this way making a more robust research.

Edan *et al.* (2016) present in their article a performance evaluation and Quality of Service (QoS) multimedia transmission (voice and video), using the SIP and IAX2 protocols based on an Asterisk server. The quality of the service evaluated used some parameters of the Qos, such as bandwidth, jitter and delay, in order to investigate the performance of different CODECs of voice and video. In the work of Edan *et al.* (2016), the authors were able to conduct concurrent call behavior tests with both softphones and servers. They were also able to design and develop an application for video transmission with support for the IAX2 protocol, in order to allow the completion of one of the tests listed in this work.

In the article ”Implementation and Evaluation of Open Source Unified Communications for SMBs”, Tesfamicael *et al.* (2014) presented the implementation and evaluation of a unified communication system composed of instant messaging and sharing (voice and video), voice messaging, VoIP communication and mobility. The evaluation covered only quantitative measurements of instantaneously supported telephone calls. However, Tesfamicael *et al.* (2014) started their work with a unified communication system approach, nevertheless, in their experiments they performed only voice and video communication tests, thus, they did not demonstrate the efficiency of the system when they were used with several modules of the proposed unified communication system. Tesfamicael *et al.* (2014) also carried out a work with great contribution, as the accomplishment of these same ones for the industry, researchers and academic community, however, there is still much to be done; experiments in embedded hardware devices, as well as the measurement of energy efficiency at times of high system consumption, are not addressed in this research.

Abid *et al.* (2012), in their article entitled ”Embedded Implementation of an IP-PBX/VoIP Gateway”, propose the idea of designing and implementing an embedded PBX-IP gateway, which uses low cost and open source solutions. The system integrates the FPGA hardware and incorporates the software into an external memory. Their experiment consists of an ML501 FPGA hardware, with embedded asterisk software and two pcx86 computers, one connected to a serial port and the other connected to

the ethernet network. Abid *et al.* (2012) have performed only one test of Asterisk software support in the embedded FPGA hardware . The authors performed only one network connectivity test. However, they did not perform tests to gauge the number of concurrent calls supported, measurement of power consumption, processing and memory.

The studies presented above are of relevant content, as they address an investigative practice for the concept of performance evaluation of a VoIP communication system that uses the SIP and IAX protocols.

Table 1 shows the hardwares, softwares, protocols and CODECs used in the experiments of the related works, as well as in our work. Table 2 shows a comparison between the activities carried out in our work and those presented in the related works. It is observed that ours is more complete. In the following sections, the implementation details of the performance analysis and energy efficiency in embedded devices using Asterisk, as well as the prototyping and the measurements taken to extract the results will be described.

It is worth noticing the use of different versions of the Linux operating system, as well as the use of the Asterisk software in embedded devices and in a different hardware. In addition, all the works performed experiments in real environment, which makes the research more productive. Tesfamicael *et al.* (2014) and Villac´ıs *et al.* (2013) addressed the use of the SIPP tool, which makes it possible to make several simultaneous calls. The SIP protocol, predominant in the three studies, was used more frequently, as well as CODEC G.711.

A good research work requires parameters for validating the results. However, all the above mencioned authors did not address any metrics for validation, which leads us to believe in the occurrence of possible errors in the results demonstrated.

Methodology and Approach

When defining a performance evaluation methodology, care must be taken in order not to make common mistakes, such as lack of objectives, tendentious proposals, incorrect methods of evaluation, among others. To avoid such errors, the ideal is to adopt a systematic approach such as the one proposed by Jain (1991), which was applied in this work. To employ this methodology, it is necessary to follow a sequence of steps.

The first step is the definition of the objectives and of the system. The second step is the preparation of the list of expected services and results. The third step is the selection of metrics, which establish the criteria for the performance comparison. The fourth step is to compile the parameter list, in fact, it is the list of parameters that affects the performance. The fifth step is about the choice of factors for studying, these factors are parameters that will suffer variations during the research. The sixth step is the selection of the evaluation technique; there are three techniques, which are simulation, analytical modeling and measurement. As a seventh step, there is the load choice , which consists of a list of service requests to the system. It is important to portray the current use.

Table 1: Data of Related Works

Authors	Hardware	Software	Protocol	Codec
Villac´ıs *et al.* (2013)	PC X86 i7 8gb RAM Switch Iphone	AsteriskNow X-lite Zoiper Wireshark	SIP IAX2	G.711 u G711a Gsm G.722 Speex H.263 H.264 H.261 H.263p
Edan *et al.* (2016)	PC x86 i7 3.40Ghz 8GB RAM	CentOS Ubuntu FreePBX Sipp Openfire	SIP	G.711 u G.711a G.729 Gsm
Tesfamicael *et al.* (2014)	Alix 2d2	Voyage Asterisk Sipp	SIP IAX2	G.711 u G.711a G.722 Gsm Speex G.726
Abid *et al.* (2012)	ML501 FPGA	Linux Asterisk	-	-
This work	Raspberry Pi 3, Banana Pi M3, Orange Pi Plus 2, Mikrotik 951g.	Raspbian Armbian Wireshark Zabbix Zoiper X-lite	SIP IAX2	G.711a G.711u G.722 Gsm Ilbc Speex

Table 2: Activities carried out

Qualities	Villac´ıs *et al.* (2013)	Edan *et al.* (2016)	Tesfamicael *et al.* (2014)	Abid *et al.* (2012)	This work
Embedded device	x	-	-	x	x
Simultaneous calls sip	x	-	x	-	x
Simultaneous calls iax2	x	-	-	-	x
Memory Consumption	x	-	-	-	x
CPU consumption	x	-	x	-	x
Energy consumption	-	-	-	-	x
Validation metric	-	-	-	-	x

Then the eighth step is the planning of the experiments. As a ninth step, we have to analyze and interpret the data, in this step we must use adequate statistical techniques in order to consolidate the results obtained, in order to allow conclusions about the performance of the system. Already tenth and final step is the presentation of the results, in this step we must pay attention to the final presentation of the evaluation.

Application of the Methodology

When applying this methodology, it is possible to notice its importance, given the organized form in which the work was conducted. Initially, it is necessary to define the objectives, then the scope of the system, the services offered, as well as the evaluation technique, which are shown below.

Goals:

- Carrying out a performance analysis on three low-cost embedded devices.
- Determining relevant factors in the performance of these equipments.

System:

- The system corresponds to a voice communication software over IP called Asterisk, it interacts with the medium through the reception and realization of telephone calls through the IP Protocol.

Service:

- Voice over IP communication.

Assessment Technique:

- Measurement, therefore, it is a useful technique for analyzing the performance of computer systems.

The activity was divided into five stages:

- Step 1 - Design and test scenario
- Step 2 - Specification of metrics
- Step 3 - Definition of parameters, factors and load
- Step 4 - Planning and conducting the experiments
- Step 5 - Statistical analysis of the results obtained

The next subsection describes the first three steps, while the fourth and fifth are discussed in the Experiments and Results section.

Design and Test Scenario

To carry out the prototyping, it is necessary to assume the existence of a computational model identical to the real production environment. In the literature, good descriptions of the performance analysis in voice over IP communication systems have been found, for example, the work of Villac´ıs *et al*. (2013) and Edan *et al*. (2016). The first work presents a performance analysis of the Alix 2D2 hardware for use in VoIP systems. The second one presents a performance assessment and Quality of Service (QoS) for multimedia transmission (voice and video).

Based on these works and technical specifications of commercially available equipment, the present work was carried out with three modern embedded devices and with the proposal of complementing the research already done, in this way, it effectively contributes to the telecommunications industry, for small and large companies, as well as for the academic area, therefore, extending the conduction of new researches.

This work analyses the possibility of verifying the use of modern embedded devices as servers of a voice over IP communication system called Asterisk using the SIP and IAX2 protocols and with the following CODECs : G.711a (Alaw), G.711u (Ulaw), Gsm, Speex, Ilbc and G.722, no other CODECs were used because of the limitations found in the work performed, such as free softphones used only to support these CODECs. The embedded devices discussed in this work have an average price of 50 dollars, thus, it is possible to consider as a low cost solution.

The performance analysis consists in verifying and evaluating the jitter, delay and bandwidth, as well as the number of concurrent calls supported in each embedded device with the protocols and CODECs mentioned above , and in parallel to carry out the monitoring of the of RAM, processing and energy consumption. For this, the tool used for traffic analysis was the wireshark, a software specialized in traffic analysis in IP networks, widely used by researchers in the academic environment, for example, by Edan *et al*., 2016.

In order to perform this work, it was necessary to elaborate a test scenario, so the scenario included three modern and low cost embedded devices (Raspberry Pi 3, Banana Pi M3 and Orange Pi Plus 2), a Mikrotik RouterBoard 951g that made the switch , a Core i7 8GB RAM 500GB HD laptop. Figure 1 illustrates the architecture of the scenario designed to perform the experiments.

The Mikrotik RouterBoard that performs the switch is responsible for interconnecting all the equipment in the network. Each embedded device supports the Asterisk voice over IP communication system. The laptop has been allocated to perform the data collection with the Wireshark software.

The devices were submitted to moments of increased system consumption. For this, a virtual machine was installed on the laptop, which made it possible to make several calls in order to reach the maximum level of calls supported on each device.

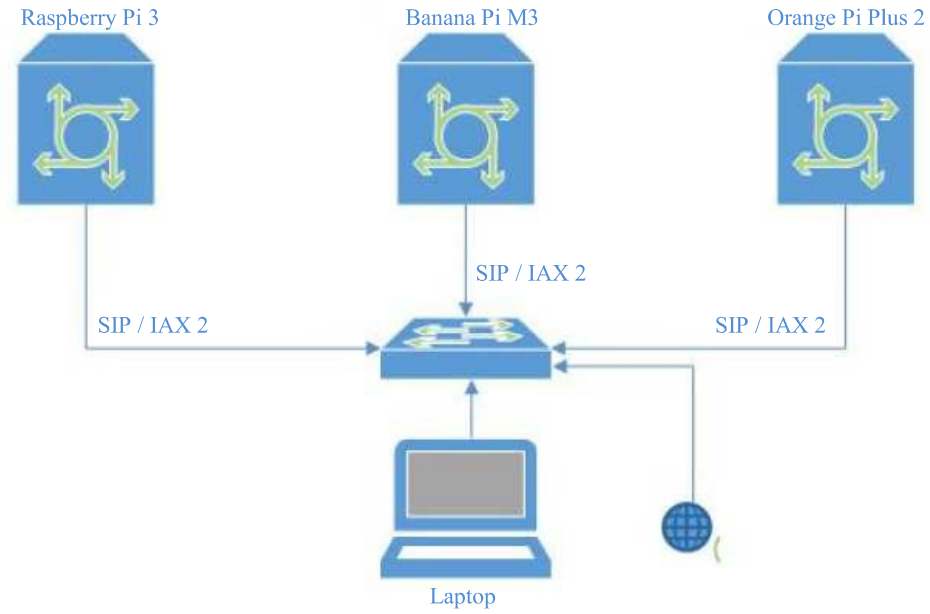


Fig. 1: Test scenario

Table 3: Embedded Devices

Manufacturer	Raspberry Pi	Banana Pi	Orange Pi	Odroid
Model	3 Model B	M3	Plus 2	XU4
CPU Cores	4	8	4	4+4
CPU Freq.	1.2 GHz	2 GHz	1.5 GHz	2.1 GHz 1.5 GHz
Memory	1 GB DDR2	2 GB DDR3 MicroSD	2 GB DDR3 MicroSD	2 GB DDR3
Storage	MicroSD	USB Sata 2.0	USB Sata 2.0	MicroSD eMMC
Linux	Yes	Yes	Yes	Yes
Price	$40	$59,99	$59,99	$59

Currently, there is a great offer of embedded devices in the market, also called Single-Board Computers (SBC), with the most diverse configurations, for example, it is possible to mention the Raspberry Pi Zero, Raspberry 2 Model B, Raspberry Pi 3 Model B, among others. Table 3 illustrates four models of the last generation embedded devices, their configurations and price.

According to Digium (2017), it is not easy to size a minimum hardware for the installation of the voice over IP communication system Asterisk, however, for this difficult question, there is no precise answer. It is recommended to always use a hardware slightly beyond the needs, it is also suggested that if the system needs to support more than 20 simultaneous calls, a dedicated server should be used. Thus, for the accomplishment of this work, three devices of low cost and with good configurations were chosen, which are: Raspberry Pi 3, Banana Pi M3 and Orange Pi Plus 2.

Experiments and Results

This section presents the implementation of the experiment, which consists of: assembling the test scenario with the embedded devices individually; accomplishment of the software approach in the devices for the elaboration of the experiment; proceeding with the process of collecting the jitter, delay and bandwidth with the Wireshark software; implementing a code in a Shell Script in order to generate loads of SIP and IAX calls with CODEC's G.711u (Ulaw), G.711a (Alaw); Gsm, Speex, Ilbc and G.722; running high call flows and in parallel to carry out the monitoring of the consumption of processing; RAM and energy; and finally, to gauge the results.

Figure 2 illustrates the current testing scenario with the Banana Pi M3 device, INA219 power meter circuit, two monitors to debug the occurrence of possible errors in Asterisk, the MikroTik switch, Yalink t19p IP phone, as well as the laptop to generate loads and gauge the processing and memory consumption.

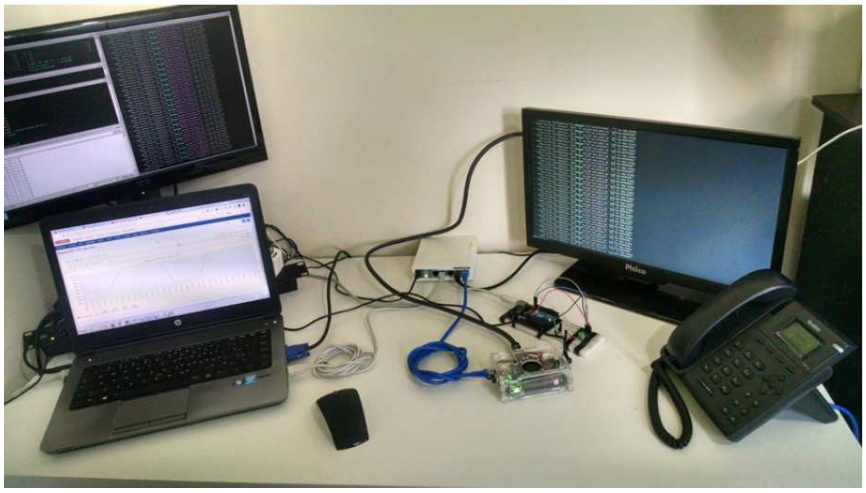


Fig. 2: Eperiment running

The tests were performed in order to reach the maximum number of calls supported with no occurrence of errors. Thus, several attempts were made until the expected result was found. Thereafter, 10 replicates were performed to validate the occurrence of no errors. For it is a static result, there is no variation of the mean and thus a greater number of repetitions can be done with it.

The collection was performed with the SIP protocol and IAX2 and with the CODECs supported by the free softphones addressed (Zoiper and X-lite). However, 33 collections (6 SIP and 5 IAX2 in each embedded device) were performed, each of the calls lasting approximately 2 minutes, so it was possible to obtain a good quantitative of SIP and IAX2 packets to carry out the analysis, approximately 5,000 datagrams IP.

The Wireshark tool has filters that automatically perform jitter, bandwidth and delay analysis. Though, the mean was used as a parameter for the measurements. Notwithstanding, the mean bandwidth and delay were calculated manually by exporting the captured data to LibreOffice Calc, since Wireshark does not report the average, only the maximum value reached.

Thus, the calls were made between the softphones and a self-service set up in the Asterisk of the embedded device.

The tests were performed with the SIP and IAX2 protocols, as well as with the CODEC's mentioned above, in order to achieve the objective of this work with the measurements.

In order to carry out the generation of simultaneous SIP call load, Asterisk was used in a virtual machine installed on the laptop and a dialer was implemented in Shell Script. For calls with the IAX2 protocol, it was held the exchange of the channel on the dialer, from SIP to IAX2. The dialer code is shown in Fig. 3, as well as the comments of each command executed. Figure 4 shows the flow chart of the code, which makes it is possible to better understand its operation.

Thus, two files were generated to perform a high number of simultaneous calls, it is called CallFilePlay.sh and CallFileCodec.sh. The first is used for normal calls, the second for calls with transcoding. The only difference between the two files is that in CallFileCodec.sh the extension number on the dialer is different. In this way, these files were inserted in the Asterisk/etc/Asterisk directory on the virtual machine dialer. Then, it was possible to start the tests.

The callfile works as a script to generate functionality in Asterisk, a user or application writes a calling file to the/var/spool/asterisk/outgoing/directory, and Asterisk processes immediately.

It was also necessary to configure an audio file called test.gsm, with 3 minutes, and to include it in Asterisk's "sounds/var/lib/asterisk/sounds" directory, whose purpose is for Asterisk to receive calls from the load generator, forward them to self-service and then play the audio. In this way, the media flow limited time occurs, enough to run the concurrent call test.

In order to start the tests, the virtual machine was used on the laptop to run the dialer load generator, and in parallel the monitoring of the RAM memory and processing consumption was carried out with the Zabbix software; the monitoring of the energy consumption was carried out with the INA219 circuit, as well as the verification of errors occurring in the processing of calls, through the debug in Asterisk.

A Yalink T19p E2 IP telephone was also used in order to validate the quality of the call at a high load consumption. However, it was not possible to validate the quality with the IAX2 protocol and neither with the Speex and Gsm CODEC's with the SIP protocol, due to the license limitations of the telephone set.

As the tests were carried out, the CODECs were modified, as well as the protocol. The experiments of the simultaneos calls were divided into 4 stages, which will be described in next subsections, as well as their results.

```
#!/bin/bash
aux1=$1; 'Variable defined the number of calls'
aux2=$2; 'Variable defined the number of repetitions'
CONTADOR=0 'Counter for number of calls'
CONTADOR2=0 'Counter for number of repetitions'
echo "Enviando $1 chamadas $2 vezes" 'Print the number of calls and replays'
while [ $CONTADOR2 -lt $aux2 ]; do 'Command to execute the number of repetitions'
echo "$CONTADOR2 x"
while [ $CONTADOR -lt $aux1 ]; do 'Command to execute the number of calls'
echo "Channel: IAX2/vm/5000" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Asterisk channel to send the call'
echo "CallerID: vm" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Set Callerid'
echo "MaxRetries: 0" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Maximum number of dial attempts (in case of failure)'
echo "RetryTime: 60" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Number of seconds to wait until the next dialing attempt (in case of failure)'
echo "WaitTime: 30" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'Number of seconds the system waits for the call to be answered'
echo "Context: default" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'When the call is answered, the call flow enters this Asterisk context
(/etc/asterisk/extensions.conf)'
echo "Extension: 3000" >>
/etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call 'When the call is answered, the call flow enters the Asterisk context above and
 in that extension (/etc/asterisk/extensions.conf)'
'The above commands create the test_file_repeat_number_call_number.call'
/var/spool/asterisk/outgoing/
'The command below moves to Asterisk to process'
mv /etc/asterisk/callfiles/teste$CONTADOR_$CONTADOR2.call
/var/spool/asterisk/outgoing/
let CONTADOR=CONTADOR+1; 'Adding the number of calls counter'
done
sleep 10s; 'Wait time to execute the next replay'
CONTADOR=0
let CONTADOR2=CONTADOR2+1; 'Adding the number of repetitions counter'
done 'Finish shell script'
```

Fig. 3: Dialer code

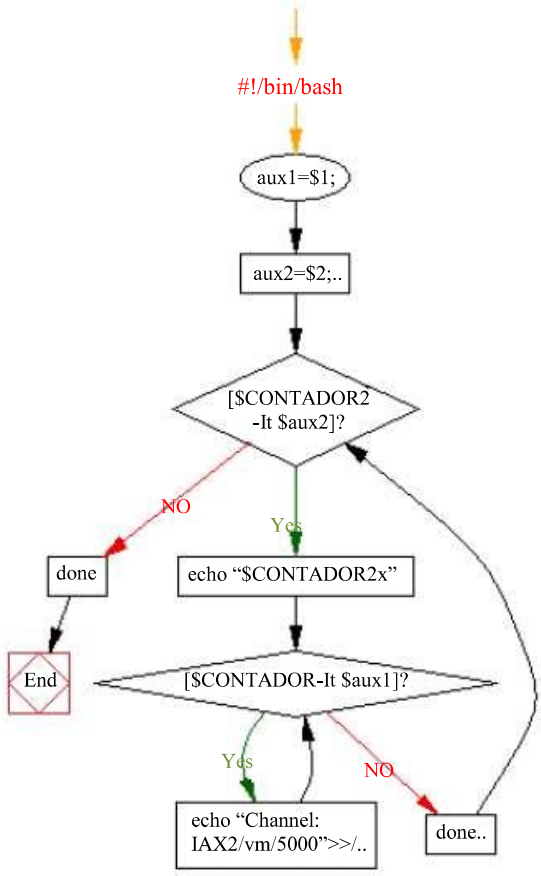


Fig. 4: Code Flowchart

The following subsections demonstrate the results of the jitter, delay and bandwidth collected with the SIP and IAX2 protocols on the embedded devices Raspberry Pi 3, Banana Pi M3 and Orange Pi Plus 2, with mean, standard deviation and confidence interval of 95%, as well as it holds brief discussions.

Bandwidth

Tables 4 and 5 illustrate the bandwidth of the collected packets. It is possible to observe that the CODECs G.711a, G.711u, and G.722 have higher bandwidth rates and identical values, since the CODECs Gsm, Speex, and Ilbc stand out because they have a lower bandwidth, both with SIP protocol and with IAX2. The results found on the three embedded devices are similar. However, the distinction is clear in the approach of the SIP and IAX2 protocols, since it is observed that the IAX2 protocol consumes a smaller band in front of the SIP, as shown in Fig. 5. It was not possible to collect the CODEC G.722 with the protocol IAX2 for not finding a free softphone with CODEC support. Moreover, it is possible to state that the standard deviations and the confidence interval are minimal due to the high number of samples to carry out the approach.

In order to facilitate the identification of the devices in Fig. 5, a code predecessor to the CODECs name has been inserted, so that, (R.) for the Raspberry Pi device 3, (B.) for the Banana Pi device M3 and (O.) , respectively, for the Orange Pi Plus 2 device.

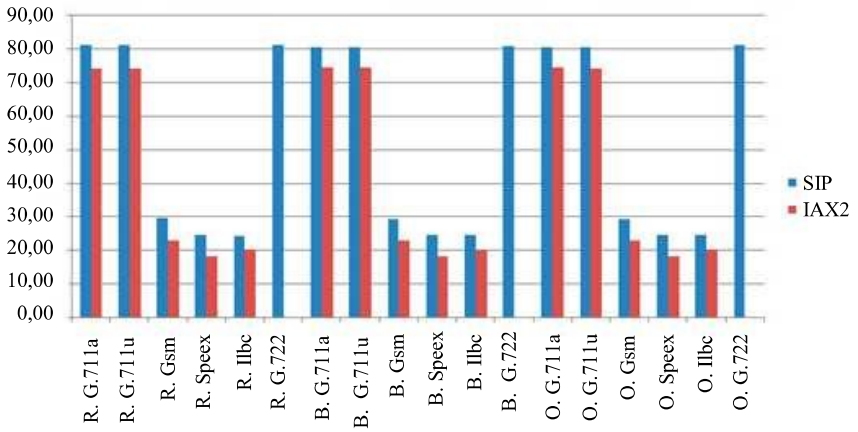


Fig. 5: Bandwidth SIP e IAX2. Unit kilobits per second

Table 4: Bandwidth SIP. Unit kilobits per second (Kbps)

	Raspberry Pi 3			
Codecs	Average	Standard Deviation	Inferior Limit	Upper Limit
G.711a	81.13	0.76	81.08	81.18
G.711u	81.11	0.77	81.06	81.15
Gsm	29.57	0.28	29.56	29.58
Speex	24.52	0.20	24.51	24.52
Ilbc	24.36	0.38	24.35	24.38
G.722	81.09	0.75	81.06	81.11
Banana Pi M3				
G.711a	80.50	0.73	80.48	80.53
G.711u	80.48	0.72	80.46	80.50
Gsm	29.38	0.27	29.37	29.39
Speex	24.58	0.24	24.58	24.59
Ilbc	24.49	0.00	24.49	24.49
G.722	80.94	0.79	80.92	80.97
Orange Pi Plus 2				
G.711a	80.51	0.74	80.49	80.53
G.711u	80.52	0.74	80.50	80.54
Gsm	29.40	0.27	29.39	29.41
Speex	24.58	0.23	24.57	24.58
Ilbc	24.49	0.00	24.49	24.49
G.722	81.06	0.76	81.04	81.08

Table 5: Bandwidth IAX2. Unit kilobits per second

	Raspberry Pi 3			
Codecs	Average	Standard deviation	Inferior limit	Upper limit
G.711a	74.21	0.72	74.19	74.23
G.711u	74.20	0.72	74.18	74.22
Gsm	22.99	0.22	22.98	23.00
Speex	18.14	0.17	18.13	18.14
Ilbc	20.13	0.02	20.13	20.13
Banana Pi M3				
G.711a	74.37	0.78	74.35	74.40
G.711u	74.33	0.78	74.31	74.36
Gsm	23.03	0.23	23.03	23.04
Speex	18.18	0.18	18.17	18.18
Ilbc	20.03	0.32	20.02	20.04
Orange Pi Plus 2				
G.711a	74.57	0.70	74.55	74.60
G.711u	74.25	0.73	74.23	74.28
Gsm	23.01	0.23	23.00	23.02
Speex	18.16	0.18	18.15	18.16
Ilbc	20.13	0.19	20.12	20.13

Delay

Tables 6 and 7 demonstrate the delay of the collected packets. It is possible to observe that the Delay with the SIP protocol is in accordance with the standard Rfc1890 (2017), which establishes that the default delay of the RTP packet should be 20ms. However, there is an exception to the Ilbc CODEC, which stands out with a value that is 50% higher than the others, as shown in Fig. 6. This is due to its coding algorithm, which performs a high compression of the audio and consequently increases the transmission delay.

The delay with the IAX2 protocol tends to zero with all the CODECs addressed in this research, which validates its proposal as a new protocol standard. According to Rfc5456 (2016) it is an open protocol that carries the transport of signaling and the media. In addition, the IAX2 protocol also proposes to eliminate any transmission delays. It is also observed that the standard deviations and the confidence interval are minimal due to the high number of samples to carry out the approach.

Jitter

Tables 8 and 9 illustrate the jitter of the collected packets. It is possible to observe that, with both the SIP protocol and the IAX2 protocol, the jitter tends to zero, this is because both protocols have as strategy to keep the frames in a buffer, in order to allow the slower frames to arrive in time to be played in the correct sequence. The higher the amount of jitter, the greater the number of frames in the buffer in order to minimize the jitter in VoIP calls.

SIP Calls

Table 10 illustrates the data collected using the SIP protocol. It is possible to observe that the GSM CODEC supported the largest number of calls in the Raspberry Pi 3 and Banana Pi M3 devices. However, there is a high consumption of RAM. Nevertheless, given the RAM capacity of the Raspberry Pi 3 device of 1GB RAM and Banana Pi M3 of 2GB, this result becomes insignificant.

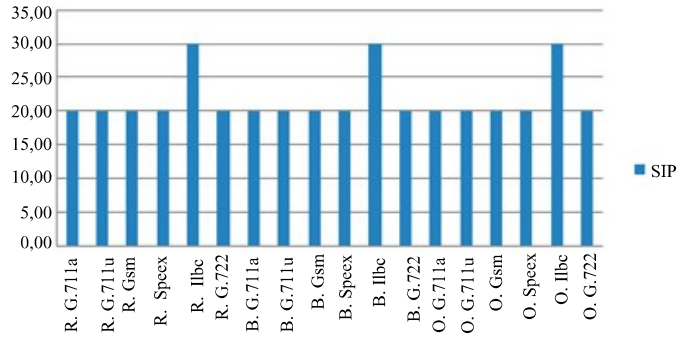


Fig. 6: Delay SIP. Unit milliseconds (ms)

Table 6: Delay SIP. Unit milliseconds (ms)

	Raspberry Pi 3			
Codecs	Average	Standard deviation	Inferior limit	Upper limit
G.711a	20.02	0.30	20.00	20.04
G.711u	20.02	0.30	20.00	20.04
Gsm	20.01	0.12	20.01	20.02
Speex	20.02	0.12	20.01	20.03
Ilbc	29.92	0.25	29.91	29.93
G.722	20.00	0.30	19.99	20.01
Banana Pi M3				
G.711a	20.00	0.06	20.00	20.01
G.711u	20.00	0.05	20.00	20.01
Gsm	20.00	0.06	20.00	20.01
Speex	20.00	0.12	20.00	20.01
Ilbc	30.01	0.41	29.99	30.02
G.722	20.00	0.27	20.00	20.01
Orange Pi Plus 2				
G.711a	20.00	0.06	20.00	20.01
G.711u	20.00	0.07	20.00	20.01
Gsm	20.00	0.11	20.00	20.01
Speex	20.00	0.12	20.00	20.01
Ilbc	30.01	0.17	30.00	30.01
G.722	20.00	0.10	20.00	20.01

Table 7: Delay IAX2. Unit milliseconds (ms)

	Raspberry Pi 3			
Codecs	Average	Standard Deviation	Low Limit	Upper Limit
G.711a	0.02	0.00	0.02	0.02
G.711u	0.02	0.00	0.02	0.02
Gsm	0.01	0.02	0.02	0.02
Speex	0.02	0.00	0.02	0.02
Ilbc	0.03	0.00	0.03	0.03
Banana Pi M3				
G.711a	0.02	0.01	0.02	0.02
G.711u	0.02	0.01	0.02	0.02
Gsm	0.02	0.01	0.02	0.02
Speex	0.02	0.01	0.02	0.02
Ilbc	0.03	0.01	0.03	0.03
Orange Pi Plus 2				
G.711a	0.02	0.00	0.02	0.02
G.711u	0.02	0.00	0.02	0.02
Gsm	0.02	0.00	0.02	0.02
Speex	0.02	0.00	0.02	0.02
Ilbc	0.03	0.00	0.03	0.03

Table 8: Jitter SIP. Unit milliseconds (ms)

	Raspberry Pi 3			
Codecs	Average	Standard Deviation	Low Limit	Upper Limit
G.711a	0.15	0.09	0.15	0.15
G.711u	0.15	0.09	0.15	0.15
Gsm	0.10	0.23	0.09	0.10
Speex	0.21	0.15	0.21	0.21
Ilbc	0.24	0.15	0.23	0.25
G.722	0.15	0.10	0.14	0.15
Banana Pi M3				
G.711a	0.03	0.01	0.03	0.03
G.711u	0.03	0.01	0.03	0.03
Gsm	0.03	0.01	0.03	0.03
Speex	0.06	0.02	0.06	0.06
Ilbc	0.24	0.08	0.24	0.25
G.722	0.15	0.09	0.14	0.15
Orange Pi Plus 2				
G.711a	0.04	0.01	0.04	0.04
G.711u	0.04	0.01	0.04	0.04
Gsm	0.06	0.03	0.06	0.06
Speex	0.06	0.02	0.06	0.06
Ilbc	0.09	0.04	0.10	0.10
G.722	0.06	0.03	0.06	0.06

Table 9: Jitter IAX2. Unit milliseconds (ms)

	Raspberry Pi 3			
Codecs	Average	Standard Deviation	Low Limit	Upper Limit
G.711a	0.02	0.19	0.01	0.02
G.711u	0.02	0.19	0.01	0.02
Gsm	0.00	0.00	0.00	0.00
Speex	0.02	0.19	0.01	0.02
Ilbc	0.02	0.19	0.01	0.03
Banana Pi M3				
G.711a	0.02	0.19	0.01	0.03
G.711u	0.02	0.19	0.01	0.03
Gsm	0.02	0.19	0.01	0.02
Speex	0.02	0.19	0.01	0.02
Ilbc	0.03	0.19	0.01	0.02
Orange Pi Plus 2				
G.711a	0.02	0.19	0.01	0.02
G.711u	0.02	0.19	0.01	0.02
Gsm	0.02	0.19	0.01	0.02
Speex	0.02	0.19	0.01	0.02
Ilbc	0.02	0.19	0.01	0.02

Table 10: Data collected using the SIP protocol.

	Raspberry Pi 3 SIP						
Codecs	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	257.00	251.00	291.00	211.00	211.00	241.00
Memory (MB)	48.48	95.01	93.92	121.13	102.00	94.66	125.75
Processing (%)	0.05	64.36	70.55	55.72	100.00	100.00	78.89
Current (energy)	390.76	486.91	691.31	644.37	838.33	848.5	726.43
Banana Pi M3 SIP							
Simultaneous calls	0.00	261.00	251.00	305.00	229.00	221.00	249.00
Memory (MB)	89.04	149.62	139.13	180.09	142.13	142.68	145.57
Processing (%)	0.51	69.05	72.43	63.89	100.00	100.00	73.16
Current (energy)	686.11	890.39	950.45	934.21	1038.25	988.64	905.00
Orange Pi Plus 2 SIP							
Simultaneous calls	0.00	247.00	261.00	251.00	241.00	241.00	251.00
Memory (MB)	81.97	113.77	125.50	113.51	122.32	124.46	117.83
Processing (%)	0.56	98.97	99.82	99.48	98.67	98.53	99.34
Current (energy)	805.33	1133.69	1138.51	1119.26	1130.21	1135.64	1137.43

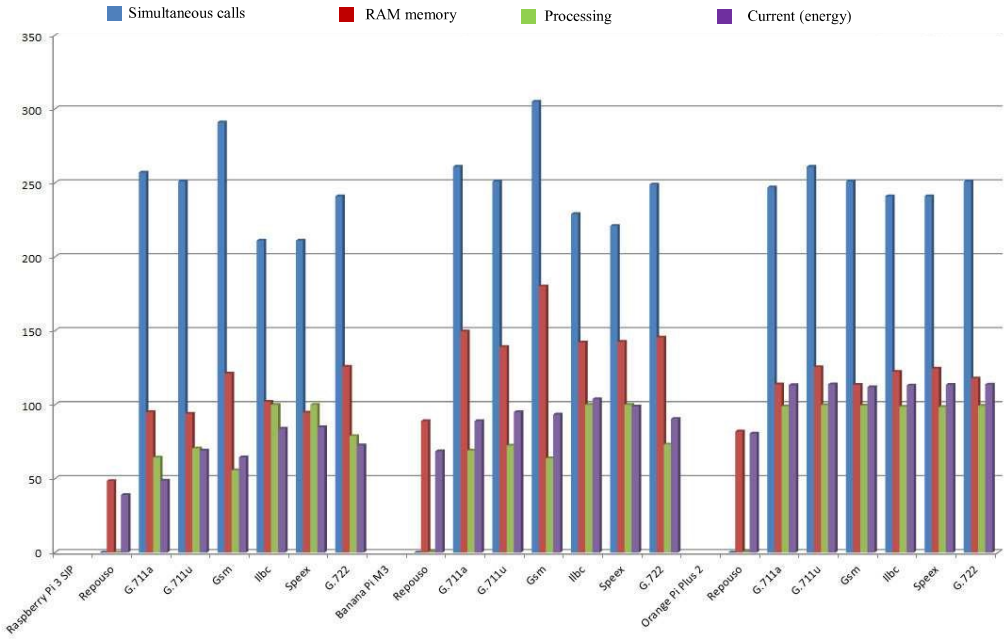


Fig. 7: SIP comparative analysis

Though the GSM CODEC has the lowest processing. The CODEC G.711a stands out because it is the CODEC that consumes less energy and supports a significant number of calls because it is an embedded device.

On the other hand, the Orange Pi Plus 2 device got a high processing in all the tests. Given this information, it is possible to state that this device is not ideal for use with Asterisk.

Figure 7 illustrates the behavior of the calls made on the 3 embedded devices, which allows better evaluation and comparison of the data. However, it is possible to observe that the Orange Pi Plus 2 device has identical results to all the CODECs addressed in this research, which reinforces the thesis that this device is not adequate to use with the Asterisk software.

SIP Calls with Transcoding

Table 11 shows the results collected on the calls made with the SIP protocol and with transcoding. It is posible to observe that with transcoding the performance of the devices is greatly reduced. Even so, the processing of the Orange Pi Plus 2 device remains high. This, once again, reinforces the claim that such a device is not suitable for Asterisk use. In the other devices, it is observed that the G.711a and G.711u CODECs have a higher number of concurrent calls supported and lower processing, memory and power consumption. In addition, the Raspberry Pi 3 device stands out with a considerable consumption of processing with CODECs G.711a and G.711u, as shown in Fig. 8.

Table 11: SIP with Transcoding

	Raspberry Pi 3						
Codecs	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	119.00	121.00	111.00	87.00	101.00	91.00
Memory (MB)	48.48	87.17	95.98	99.47	148.30	247.16	97.49
Processing (%)	0.05	3.97	41.87	68.83	100.00	100.00	93.70
Current (energy)	390.76	438.85	592.64	592.04	821.49	577.56	654.27
Banana Pi M3							
Simultaneous calls	0.00	131.00	129.00	131.00	109.00	111.00	129.00
Memory (MB)	89.04	136.14	115.63	122.30	121.01	120.95	126.86
Processing (%)	0.51	54.57	41.13	75.87	86.11	93.23	92.48
Current (energy)	686.11	718.81	844.7	934.56	977.82	987.33	975.25
Orange Pi Plus 2							
Simultaneous calls	0.00	101	97.00	95.00	87.00	95.00	91.00
Memory (MB)	81.97	132.74	120.92	119.70	120.95	124.95	122.67
Processing (%)	0.56	100.00	99.99	99.99	99.98	99.99	99.99
Current (energy)	805.33	1154.52	1150.26	1145.42	1155.57	1159.00	1157.58

Table 12: Data collected with the IAX2 protocol

	Raspberry Pi 3 IAX2						
Codecs	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	181.00	163.00	171.00	85.00	75.00	165.00
Memory (MB)	48.48	89.54	83.66	78.16	78.99	78.38	82.49
Processing (%)	0.05	52.28	50.67	86.41	95.21	86.64	72.23
Current (energy)	39.07	67.28	75.12	58.58	73.81	76.87	71.73
Banana Pi M3 IAX2							
Simultaneous calls	0.00	201.00	191.00	201.00	105.00	105.00	181.00
Memory (MB)	89.04	107.09	107.62	108.77	101.94	103.04	106.08
Processing (%)	0.51	59.82	61.47	47.97	68.06	78.69	53.46
Current (energy)	68.61	83.39	85.96	82.92	92.7	95.29	86.69
Orange Pi Plus 2 IAX2							
Simultaneous calls	0.00	201.00	189.00	201.00	81.00	45.00	91.00
Memory (MB)	81.97	107.23	105.25	101.09	97.29	91.97	102.00
Processing (%)	0.56	90.30	89.77	69.78	94.00	75.27	94.00
Current (energy)	80.53	110.87	110.31	106.29	116.04	95.65	93.49

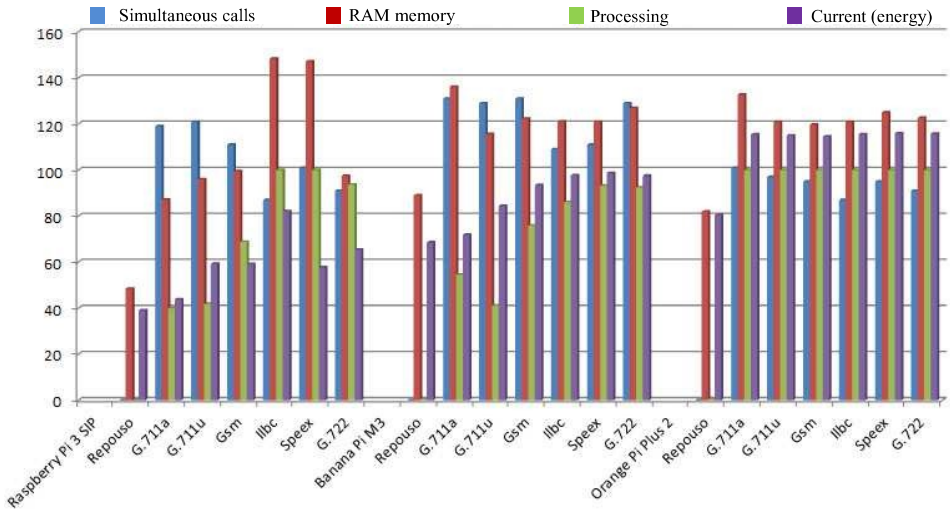


Fig. 8: SIP with transcoding comparative

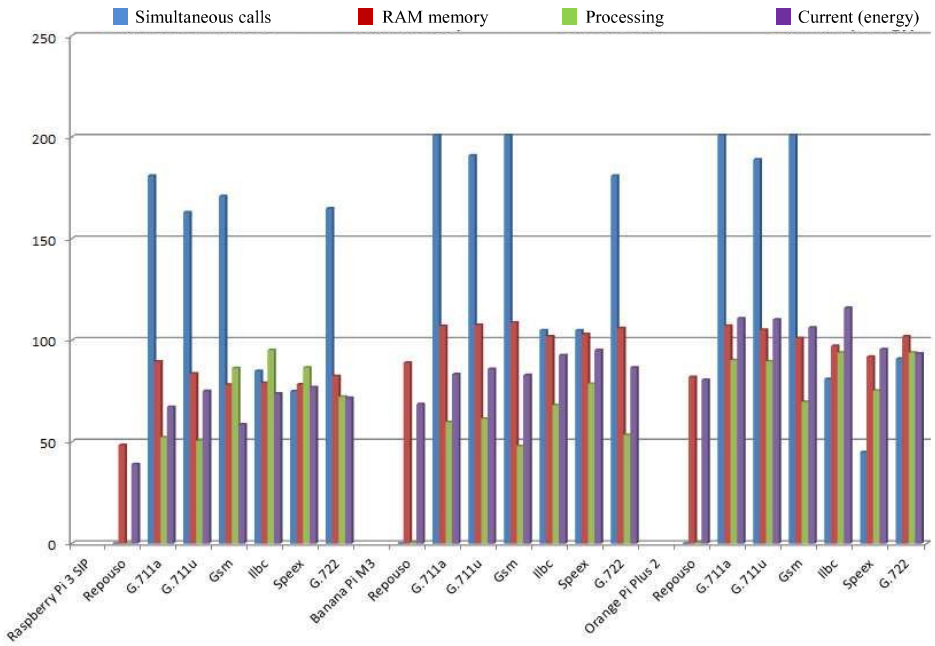


Fig. 9: IAX2 Comparative analysis

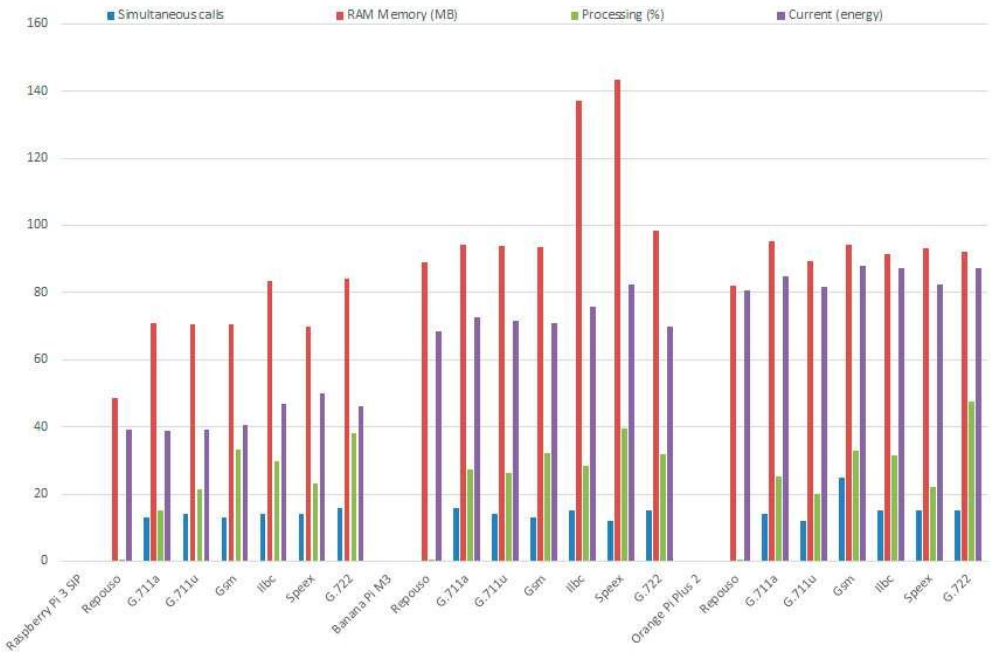


Fig. 10: IAX2 with Transconding Analysis

IAX2 Calls

This subsection addresses the data collected using the IAX2 protocol. They are shown in Table 12. It is possible to note that the G711.a CODEC stands out with low memory, processing and power consumption, and a high number of simultaneous calls on the Raspberry Pi 3 and Banana Pi M3 devices. The Orange Pi Plus 2 was highlighted by the GSM CODEC due to the high number of simultaneous calls, low processing and power consumption, and moderate memory consumption. However, if compared to the other devices addressed in this research, this is the one that has inferior performance as shown in Fig. 9.

Table 13: Data collected with the IAX2 protocol with Transcoding

	Raspberry Pi 3						
Codec's	Repose	Alaw	Ulaw	Gsm	Ilbc	Speex	G.722
Simultaneous calls	0.00	13.00	14.00	13.00	14.00	14.00	16.00
Memory (MB)	48.48	70.81	70.43	70.43	83.48	69.95	84.02
Processing (%)	0.05	14.98	21.46	33.35	29.92	23.30	37.99
Current (energy)	390.76	389.51	392.62	403.92	467.55	498.12	463.00
Banana Pi M3							
Simultaneous calls	0.00	16.00	14.00	13.00	15.00	12.00	15.00
Memory (MB)	89.04	94.18	93.79	93.66	137.07	143.37	98.33
Processing (%)	0.51	27.34	26.44	32.18	28.42	39.57	31.74
Current (energy)	686.11	725.73	716.77	708.26	758.22	823.97	697.62
Orange Pi Plus 2							
Simultaneous calls	0.00	14.00	12.00	25.00	15.00	15.00	15.00
Memory (MB)	81.97	95.12	89.29	94.15	91.29	93.15	92.26
Processing (%)	0.56	25.32	20.06	32.92	31.63	22.05	47.52
Current (energy)	805.33	846.61	817.15	879.85	874.27	825.30	873.12

IAX2 Calls with Transcoding

Table 13 shows the data collected using the IAX2 protocol with transcoding. Immediately, it is possible to observe that there is a low number of concurrent calls supported. Given this context, it is possible to state that the embedded devices addressed in this research do not support the process of transcoding CODECs with the IAX2 protocol. Only in this test the Orange Pi Plus 2 device did not report high processing consumption, as shown in Fig. 10.

Conclusion and Future Work

In this work, an approach was performed to analyze the performance and efficiency energy in three state-of-art embedded devices using the Asterisk software of voice over IP, which measured the jitter, delay and bandwidth with SIP and IAX2 protocols with CODEC's G.711a (Alaw), G.711u (Ulaw), G.722, Ilbc, Speex and Gsm.

The measurements were performed in order to compare the three embedded devices with the use of Asterisk. However, the results showed great similarity in the data, both with the SIP protocol and with the IAX2 protocol, this in the network requirement.

We also verified the number of concurrent calls supported in each device with the SIP and IAX2 protocols, both in normal calls and in transcoded calls and, in parallel, the analysis of the RAM memory, processing and energy consumption was performed.

The prototyping was performed to compare the three embedded devices using the Asterisk. The results were surprising. The Raspberry Pi 3 and Banana Pi M3 devices satisfactorily support a high number of simultaneous calls with moderate memory, processing and power consumption through CODEC G.711a and G.711u. However, the Orange Pi Plus 2 device showed a high processing power. Thus, it is possible assert that this device is not suitable for use with Asterisk.

All the 3 devices showed stability throughout the research. Not occurring unintentional restart of the equipment, even during high loads.

The performance of the 3 embedded devices discussed in this work were evaluated with the purpose of finding the best device to support the communication system by the Asterisk voice over IP. Nevertheless, new embedded devices, as well as new technologies, will emerge. In this way, the possibility of extending this work is certain.

As future work, it is proposed to carry out experiments with CODEC Opus, since it was not possible to perform the compilation on the embedded devices due to incompatibility. It is also possible to carry out the same approach with the IAX2 protocol, using encryption. We can to propose a comparison of the behavior of the IAX2 protocol with transcoding on computing platforms with x86 architecture. It should be performed the same bandwidth, jitter and delay tests with other tools available in the market, specific to VoIP packet analysis, such as NetQuality Voip and SIP Tester.

Acknowledgement

This research work is possible through the and support of all, including family, future wife, friends and especially in recognition of gratitude for my Professors, Dr. Admilson de Ribamar Lima Ribeiro and Dr.Edward David Moreno Ordonez, thank you for all the support and encouragement.

Author's Contributions

Adauto Cavalcante Menezes: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Toniclay Andrade Nogueira: The author organized the study, participated in all experiments, coordinated the data-analysis.

Edward David Moreno Ordonez: The author designed the research plan, gave final approval of the version to be submitted, as well as supervised and coordinated the entire work.

Admilson de Ribamar Lima Ribeiro: The author designed the research plan, gave final approval of the version to be submitted, as well as supervised and coordinated the entire work.

Ethics

The authors confirm that they abide to all ethical protocols and procedures while preparing this manuscript.

References

Abid, F., N. Izeboudjen, M. Bakiri, S. Titri and F. Louiz, 2012. Embedded implementation of an IP-PBX /VoIP gateway. Proceedings of the 24th International Conference on Microelectronics, (ICM' 12), Algeria, pp: 5-8.

Androulidakis, I.I., 2016. VoIP and PBX Security and Forensics.

Bernal, P.S.M., 2007. Voz sobre protocolo IP: A nova realidade da telefonia. Sa˜o Paulo.

Bryant, R., L. Madsen and J.V. Meggelen, 2013. Asterisk: The Definitive Guide. Sebastopol, 4th Edn., O'Reilly Media, Inc., Sebastopol, ISBN-10: 1449332455, pp: 846.

Colcher, S., A.T.A. Gomes, A.O. da Silva, G.L.d.S. Filho and L.F.G. Soares, 2005. VoIP: voz sobre IP. Rio de Janeiro.

Digium, 2017. Dimensioning, [accessed 2017 Dez 10]. Available: http://www.digium.com/blog/2012/09/25/asteriskdimensioning-what-server-do-i-need.

Edan, N. M., A. Al-Sherbaz, S. Turner and S. Ajit, 2016. Performance evaluation of QoS using SIP and IAX2 VVoIP protocols with CODECS. Proceedings of the SAI Computing Conference, 13-15 Jul., IEEE Xplore press, London, UK, pp: 631-636. DOI: 10.1109/SAI.2016.7556048

Jain, R., 1991. The Art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and modeling. New York.

Rfc1890, 2017. RTP Profile for audio and video conferences with minimal control. RFC.

Rfc5456, 2016. Rfc5456.

Sulkin, A., 2002. PBS systems for Ip telephony: Migrating enterprise communications. Sykesville.

Tesfamicael, A.D., V. Liu, W. Caelli and J. Zureo, 2014. Implementation and evaluation of open source unified communications for SMBs. Proceedings of the International Conference on Computational Intelligence and Communication Networks, Jul. 14-16, IEEE Xplore press, Bhopal, India, pp: 1243-1248. DOI: 10.1109/CICN.2014.260

Villac´ıs, D., F.R. Acosta and R.A. Lara Cueva, 2013. Performance analysis of VoIP services over WiFi-based systems. Proceedings of the IEEE Colombian Conference on Communications and Computing, May 22-24, IEEE Xplore press, Medellin, Colombia, pp: 1-6. DOI: 10.1109/ColComCon.2013.6564813