UNIVERSIDADE DO SUL DE SANTA CATARINA

BRUNO BRESCIANI DE SOUSA

RAUL ESPÍNDOLA

VULNERABILIDADES VOIP:
UM ESTUDO DE CASO SOBRE ALGUMAS VULNERABILIDADES EXISTENTES NO PROTOCOLO SIP

Palhoça

2011

BRUNO BRESCIANI DE SOUSA
RAUL ESPÍNDOLA

VULNERABILIDADES VOIP:
UM ESTUDO DE CASO SOBRE ALGUMAS VULNERABILIDADES EXISTENTES NO PROTOCOLO SIP

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica - Telemática da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Engenheiro Eletricista.

Orientador: Djan Rosário

Palhoça
2011

BRUNO BRESCIANI DE SOUSA

RAUL ESPINDOLA

VULNERABILIDADES VOIP:
UM ESTUDO DE CASO SOBRE ALGUMAS VULNERABILIDADES EXISTENTES NO PROTOCOLO SIP

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Engenheiro Eletricista e aprovado em sua forma final pelo Curso de Engenharia Elétrica – Telemática da Universidade do Sul de Santa Catarina.

Palhoça, 09 de dezembro de 2011.

_____

Professor e orientador Djan Rosário, Eng.
Universidade do Sul de Santa Catarina

_____

Prof. Job Medeiros de Souza, Eng.
Universidade do Sul de Santa Catarina

_____

André Vescovi, Eng .
Alcatel Lucent.

Aos nossos queridos pais, por todo amor, carinho, compreensão, fé e estimulo.

AGRADECIMENTOS

Primeiramente, gostaríamos de agradecer aos nossos pais, avós e demais familiares por terem dado todo o suporte necessário durante esta trajetória. Com o auxílio e palavras de conforto nos momentos difíceis, eles tornaram essa caminhada menos árdua e mais gratificante.

Às nossas namoradas, pelo amor, carinho e compreensão sempre que foi necessário abdicar dos momentos juntos e focar na realização deste trabalho.

Aos nossos colegas de turma, pelo companheirismo apresentado nas horas de estudo e pelos momentos de descontração nas horas apropriadas.

Ao nosso orientador, Prof. Djan Rosário, pela confiança, paciência, dedicação e orientação.

À Prof. Sheila Santisi Travessa, pelo suporte e dedicação oferecido em todo o período de desenvolvimento deste trabalho.

Aos membros da banca examinadora, por aceitarem o convite.

“Temos o destino que merecemos. O nosso destino está de acordo com nossos méritos.” (Albert Einstein).

RESUMO

O VoIP é uma tecnologia cada vez mais difundida e utilizada nos dias atuais. Nas organizações já virou uma tendência e vem aumentando o seu crescimento entre usuários com acesso à Internet. Devido a sua ascensão, inúmeras evoluções são realizadas para melhorar a comunicação, destacando-se a criação do protocolo SIP (*Session Initial Protocol*) capaz de iniciar, alterar e finalizar sessões multimídia através de mensagens transmitidas em formato texto pela rede. Apesar das vantagens oferecidas por esta tecnologia, ela trouxe algumas preocupações, principalmente relacionadas à integridade das informações disponibilizadas no protocolo SIP. Ataques que se utilizam das vulnerabilidades existentes neste protocolo podem gerar diversos danos, e trazer prejuízos a quem utiliza este tipo de comunicação. Neste sentido, o presente trabalho teve como objetivo estudar e analisar algumas das vulnerabilidades presentes nos sistemas VoIP, focando apenas nos ataques ao protocolo SIP. Esta análise foi realizada através de um cenário montado para simular três vulnerabilidades estudadas no decorrer do desenvolvimento, são elas: roubo de registro, ataque de *flood* e mensagens maliciosas. Como resultado dessas simulações, ficou visível que o SIP é um protocolo bastante frágil quando está desprotegido, e essas fragilidades podem causar graves danos na estrutura em que ele fornece comunicação. Conclui-se então que, plataformas VoIP que utilizam o SIP para transporte das informações, devem ser implantadas levando em consideração uma maior proteção deste protocolo, evitando assim, que usuários maliciosos venham a desfrutar das informações do mesmo.

Palavras-chave: SIP, vulnerabilidades, ataque, segurança e VoIP.

ABSTRACT

The VoIP technology is an increasingly widespread and used today. In organizations has become a trend and has increased its growth among users with Internet access. Due to its rise, many changes are made to improve communication, especially the creation of SIP (Session Initial Protocol) able to initiate, modify and terminate multimedia sessions by means of messages transmitted in plain text over the network. Despite the advantages offered by this technology, it brought some concerns, mainly related to the integrity of the information provided in the SIP protocol. Attacks that use vulnerabilities in this protocol may generate various damages, losses and bring those who use this type of communication. In this sense, the present work was to study and analyze some of these vulnerabilities in VoIP systems, focusing only on attacks on the SIP protocol. This analysis was performed using a scenario set up to simulate three vulnerabilities studied during the development, they are: theft record, flood attack and malicious messages. As a result of these simulations, it became apparent that the SIP protocol is a very fragile when it is unprotected, and these weaknesses can cause serious damage to the structure in that it provides communication. It is concluded that VoIP platforms using SIP for transport of information, should be implemented taking into account an added protection for this protocol, thus preventing malicious users will enjoy the same information.

Keywords: SIP, vulnerabilities, attack, security e VoIP.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS

ARP – Address Resolution Protocol
DOS – Denial Of Service
IETF – Internet Engineering Task Force
IP – Internet protocol
ITU – International Telecommunication Union
LAN – Local Area Network
MAN – Metropolitan Area Network
OSI – Open Systems Interconnection
RTCP – RealTime Transport Control Protocol
RTP – RealTime Transport Protocol
SDP – Session Description Protocol
SIP – Session Inition Protocol
TLS – Transport Layer Security
TCP – Transmission Control Protocol
UA – User Agent
UAC – User Agent Client
UAS – User Agent Server
UDP – User Datagram Protocol
URI – Uniform Request Indentifier
URL – Uniform Resource Locator
VOIP – Voice Over IP
WAN – Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

Acompanhando a evolução e analisando o cenário atual das telecomunicações, o VOIP se tornou a grande tendência para aqueles que possuem acesso a uma arquitetura capaz de realizar comunicação de voz, vídeo e dados, utilizando uma rede comutada por pacotes. O desejo de economia, interoperabilidade e serviços agregados veem trazendo cada vez mais adeptos a esta tecnologia. Ao encontro deste crescimento, surge a preocupação dos provedores em oferecer aos usuários uma solução que agregue qualidade e confiabilidade a custos reduzidos, procurando minimizar ao máximo a quantidade de vulnerabilidades presentes na arquitetura e que podem ser exploradas por aqueles que desejam fazer uso ilícito deste tipo de solução.

Estudos sobre as vulnerabilidades dos componentes que formam a arquitetura VoIP são essenciais para garantir um sistema confiável e com qualidade. O SIP, protocolo responsável por estabelecer a comunicação entre os agentes da estrutura é o ponto mais susceptível aos ataques realizados e pode causar graves danos ao funcionamento apropriado desta tecnologia. Devido a sua importância, fragilidade e por transportar informações importantes para o contexto das transações VoIP, tem-se como foco deste trabalho de conclusão de curso as vulnerabilidades oferecidas pelo protocolo SIP e quais possíveis danos elas podem causar.

A fundamentação teórica julgada necessária para compreensão do trabalho é tratada inicialmente, conceitos básicos sobre SIP, redes de computadores, segurança e ataques são requisitos para compreensão dos estudos de caso. No desenvolvimento, estudos de caso com base em uma estrutura VoIP e ferramentas de simulação e manipulação do SIP são utilizados para demonstrar e analisar algumas das principais vulnerabilidades do protocolo como roubo de registro, manipulações indesejadas e ataques de *flood*. Por fim, são

apresentados os resultados e conclusões finais sobre as vulnerabilidades existentes no SIP bem como as formas de proteção deste protocolo.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Expor, testar e analisar os resultados dos estudos de caso desenvolvidos para atacar as vulnerabilidades existentes no protocolo SIP.

1.1.2 Objetivo Específico

- Descrever a função do SIP no VoIP;
- apresentar cenários e componentes que englobam a arquitetura SIP;
- estudar a estrutura do protocolo, tipos de mensagens e formas de transporte;
- apresentar algumas vulnerabilidades e ataques utilizando o protocolo SIP;
- analisar os resultados do estudo de caso desenvolvido durante a execução do trabalho.

2 REVISÃO BIBLIOGRÁFICA

VOIP é um termo utilizado para caracterizar um serviço de transmissão da voz utilizando o protocolo IP através de uma rede comutada por pacotes. Este procedimento consiste em digitalizar a voz em pacotes de dados para trafegarem pela rede IP até chegar ao seu destino, onde serão novamente convertidos para o formato original. A grande vantagem da tecnologia VOIP é a possibilidade da redução dos custos de utilização dos serviços de telefonia comum, principalmente em ambientes corporativos, pois utilizam a própria estrutura de rede já instalada para realizar a comunicação. À medida que esta tecnologia foi crescendo até se tornar uma tendência mundial, uma série de padrões, protocolos e aplicações foram desenvolvidas para compor a arquitetura e melhorar cada vez mais o nível da comunicação. Este capítulo visa apresentar os principais elementos que compõem a arquitetura VOIP.

2.1 EVOLUÇÃO DO SISTEMA TELEFÔNICO

Segundo Colcher et AL (2005), desde 1844 quando Samuel Morse enviou a primeira mensagem usando seu sistema de telegrafia, os sistemas de telecomunicações evoluiram gradativamente. Em 1875 durante um projeto relacionado ao sistema de telegrafia, Alexander Graham Bell percebeu que a variação da densidade de ar junto ao transmissor produzia uma corrente elétrica de mesma intensidade que por sua vez, produzia um som totalmente diferente do inesperado. Em 14 de fevereiro de 1876, um ano após seus experimentos, Grahan Bell patentiou sua criação com o nome de telefone.

Com o crescimento da demanda por serviços de telefonia, ficou inviável manter o sistema criado por Bell, então, as linhas diretas e dedicadas foram substituidas pelas primeiras

centrais telefônicas onde o circuito estabelecido entre os interlocutores era feito manualmente. Em 1891, Almon Strowger inventou a primeira central automática de chaveamento de circuitos.

Até a década de 1950, o sistema telefônico era totalmente baseado em tecnologia analógica, mas com a invensão do transistor, a produção dos circuitos integrados e a evolução dos sistemas computacionais proporcionaram a criação das primeiras centrais telefônicas digitais, que ofereceram uma série de vantagens em termos de operação, manutenção e provisão dos serviços de telefonia.

Com a era digital, os sistemas telefônicos tiveram seu crescimento atrelado com o crescimento das tecnologias computacionais, a integração delas oferecia aos usuários produtos cada vez mais sofisticados como os telefones sem fio e celulares, até quem em 1995 surge em Israel uma tecnologia que permite transmitir voz utilizando recursos multimídia de um computador pessoal através uma rede de dados, o VoIP.

A criação foi revolucionária, o fato de utilizar as infra-estruturas de redes de dados criou um conceito novo de telefonia, denominado Telefonia IP. As redes de voz, vídeo e dados que antes eram separadas agora são integradas em uma única estrutura, administração e gerenciamento, reduzindo custos e oferecendo uma série de facilidades que não são encontradas na telefonia convencional.

Para poder compreender o VoIP é necessário conhecer alguns conceitos importantes sobre redes de computadores pois é ela que fornece toda a estrutura necessária para prover esse tipo inovador de comunicação.

2.2 REDES DE COMPUTADORES

Para Colcher e outros (2005, p. 21) “rede de comunicação é formada por um conjunto de módulos processadores capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação”.

Um sistema de comunicação é o agrupamento e interligação de vários módulos processadores através de enlaces físicos (meios de transmissão), e de um conjunto de regras com o fim de organizar a comunicação definidos como protocolos.

Na grande maioria, as informações que trafegam pelas redes de computadores são fragmentadas, sendo cada fragmento chamado de “pacote”, utilizado para facilitar o compartilhamento de recursos e para a detecção de erros na informação transmitida. Cada pacote deve conter no seu conjunto de dados referências ao endereçamento, permitindo chegar ao seu destino e serem recompostos. Existe para cada aplicação um conjunto de regras (protocolo) que identifica o momento de enviar o pacote, quais devem ser enviados, reenviados e como organizá-los para que a informação seja reconstruída.

2.2.1 Categorias de Rede

Existem três categorias de redes utilizadas para realizar a comunicação de dados, cada uma caracterizada pelo contexto onde ela está sendo inserida, são elas: redes locais, redes metropolitanas e redes geograficamente distribuídas.

2.2.1.1 Redes Locais (LAN)

As redes locais, comumente chamadas LANs *(Local Area Network*), são redes privadas contidas em um espaço limitado com até alguns poucos quilômetros de extensão. Elas são normalmente usadas para conectar computadores pessoais, estações de trabalho, escritórios, empresas, permitindo o compartilhamento de recursos e a troca de informações.

Segundo Tanenbaum (2003) as LANs tem um tamanho restrito, o que significa que o pior tempo de transmissão e limitado e conhecido com antecedência. O conhecimento desse limite permite a utilização de determinados tipos de projetos que em outras circunstâncias não seriam possíveis, além de simplificar o gerenciamento da rede.

A tecnologia de transmissão das LANs quase sempre consiste em cabos metálicos que são responsáveis também pela interligação dos elementos da rede. Normalmente as taxas de transmissão são de 100Mbps superiores, gerando um atraso na transmissão quase desprezível e tornando a comunicação rápida e eficiente.

2.2.1.2 Redes Metropolitanas (MAN)

Metropolitan Area Network (MAN), são redes de caráter metropolitano ligando computadores e utilizadores numa área geográfica compreendida entre a LAN e a WAN na qual esta última comentaremos a seguir. Uma MAN normalmente resulta da interligação de várias LANs, cobrindo uma área geográfica de média dimensão, tipicamente um campus ou uma cidade com domínio privado ou público.

2.2.1.3 Redes Geograficamente Distribuídas (WAN)

Wide Area Network (WAN), é uma rede de telecomunicações que está dispersa por uma grande área geográfica, distingue-se da LAN e MAN pelo seu porte e estrutura de telecomunicações e meios de transmissões. As WAN normalmente são de caráter público, geridas por uma operadora de telecomunicações.

Segundo ED Tittel, (2002) as redes WANs são projetadas para serem alugadas por circuitos, na qual o cliente paga mensalmente pelo circuito, e uma taxa de uso baseada na quantidade de tráfego enviado sobre a rede da operadora.

2.2.2 O modelo OSI

Segundo José Maurício Santos Pinheiro (2004):

> A arquitetura de uma rede é formada por camadas (ou níveis), interfaces e protocolos. As camadas são processos, implementados por hardware ou software, que se comunicam com o processo correspondente na outra máquina. Cada camada oferece um conjunto de serviços ao nível superior, usando funções realizadas no próprio nível e serviços disponíveis nos níveis inferiores.

Com o objetivo de padronizar a comunicação entre máquinas heterogêneas de diversos fabricantes, na década de 1980 a Organização Internacional de Normatização ISO elaborou padrões de comunicações a nível mundial para sistemas abertos visando prover comunicação independente da tecnologia empregada e das distâncias envolvidas para a comunicação. O modelo criado foi denominado OSI (*Open Systems Interconnection*) servindo como base de implementação para vários tipos de rede existente. Sua arquitetura é baseada em camadas onde os dados são escalonados, não podendo passar diretamente para a camada superior antes de passar pelas inferiores.

Segundo Tanenbaum (2003, p.45):

> O modelo OSI propriamente dito não é uma arquitetura de rede, pois não especifica os serviços e os protocolos exatos que devem ser usados em cada camada. Ele apenas informa o que cada camada deve fazer. No entanto, a ISO também produziu padrões para todas as camadas, embora esses padrões não façam parte do próprio modelo de referencia. Cada um foi publicado como um padrão internacional distinto.

O modelo OSI foi dividido em sete camadas, na qual as três camadas inferiores são responsáveis por aspectos relacionados a transmissão, as camadas centrais são relacionadas a comunicação propriamente dita e as três camadas superiores são responsáveis pela apresentação a nível de usuário. São elas:

- Camada Física: A camada física trata diretamente com os bits que trafegam por um canal de comunicação. Os protocolos deste nível são os que realizam a codificação/decodificação do bits em sinais elétricos lançados no meio de transmissão.
- Camada de enlace de dados: A função da camada de enlace de dados é garantir que o canal de comunicação seja entregue à camada de rede sem erros de transmissão.
- Camada de Rede: A camada de rede é responsável por controlar as operações de rede propriamente dita. Suas principais funções são a de roteamento de pacotes entre a origem e o destino.
- Camada de Transporte: Segundo Tanembaum (2003) “camada de transporte é uma verdadeira camada fim a fim, que liga a origem ao destino como um programa da máquina de origem mantendo uma conversação com um programa semelhante instalado na máquina de destino, utilizando os cabeçalhos de mensagens e as mensagens de controle”.

- Camada de Sessão: A camada de sessão tem como principal função administrar e sincronizar diálogos entre os usuários de diferentes máquinas que estabeleçam sessões entre eles.
- Camada de Apresentação: Segundo Tanembaum (2003) “a função da camada de apresentação é assegurar que a informação seja transmitida de tal forma que possa ser entendida e usada pelo receptor, dessa forma, este nível pode modificar a sintaxe da mensagem, mas preservando sua semântica além de realizar a tradução entre formatos distintos em cada ponta da comunicação”.
- Camada de Aplicação: A camada de aplicação é a camada que possui o maior número de protocolos existentes, devido ao fato de estar mais perto do usuário e os usuários possuírem necessidades diferentes.

2.2.3 O modelo TCP/IP

Segundo Rubens Prates (2000, p.05):

> O desenvolvimento do protocolo TCP/IP começou em 1969, com o projeto ARPANET, da Agência de Projetos de Pesquisas Avançadas do Departamento de Defesa dos EUA (*Department of Defense Advanced Research Projects Agency* - DARPA). O objetivo desse projeto foi desenvolver uma rede que interligasse os computadores do governo americano, de diferentes fabricantes e utilizando diferentes sistemas operacionais. Essa rede deveria ser descentralizada e mesmo que um dos computadores dessa rede fosse destruído num eventual ataque militar, os demais continuariam a funcionar normalmente, graças a um mecanismo de rotas alternativas. Algum tempo depois desse início com finalidade militar, a *National Science Foundation* criou uma rede semelhante para interconectar instituições de pesquisa e universidades, utilizando os mesmos protocolos da rede ARPANET. Desses projetos surgiu o protocolo TCP/IP, que serviu como alicerce para a construção da rede que hoje conhecemos como Internet. A partir de 1993 a Internet ficou disponível para uso comercial e se popularizou de tal forma, que hoje a maioria de nós a utiliza com familiaridade.

O modelo TCP/IP é distribuído em quatro camadas, física, rede, transporte e aplicação. Essas camadas tomam como referência o modelo OSI e cada uma delas suporta um conjunto de protocolos como ilustrado na figura a seguir.

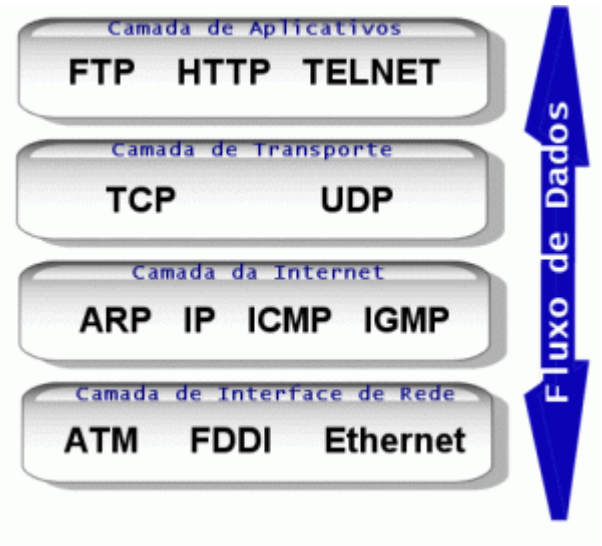

Figura 1 – Modelo TCP/IP

2.2.3.1 Protocolo Internet - IP

O protocolo Internet (IP) é definido na camada de redes do modelo OSI. É responsável pelo endereçamento dos pacotes de informação dos dispositivos de origem e destino, além do roteamento das redes compreendidas entre os mesmos. O IP é normalmente composto de 4 octetos, reservando uma parte para rede e outra para os dispositivos (hosts). Dependendo da forma como é feito a divisão, pode-se gerar redes de classe A, B e C. pelo tipo de classe definido pelo primeiro octeto do IP, e pela subrede que é definida pelo número de máscara.

Este protocolo, pode definir o melhor caminho a ser usado para a transmissão, utilizando uma tabela de roteamento mantida e atualizada pelos roteadores no trajeto da

comunicação. O protocolo IP recebe os dados fragmentados da camada de transporte onde os conjuntos de dados passam a ser chamados de datagramas. Estes datagramas são então codificados e enviados à camada física para então ser encaminhados ao meio físico propriamente dito.

2.2.3.2 *Address Resolution Protocol* - ARP

O ARP é o protocolo que caracteriza e relação entre o endereço IP e o endereço *Ethernet* (MAC). De maneira simplificada, podemos considerar o protocolo ARP como sendo um broadcast no segmento de rede, perguntando qual é o endereço MAC do dispositivo que tem um determinado endereço IP.

Segundo Tanenbaum 2003, para que os dados cheguem aos hosts é necessário um outro tipo de endereço além do IP que é o endereço *Media Access Control* - MAC ou *Ethernet*.

A troca de dados entre dispositivos IP é efetuada através do endereço, conhecido como endereço Físico do dispositivo. Na construção do datagrama, a aplicação sabe os endereços MAC e IP da origem por ser o mesmo dispositivo onde a mesma se encontra, e somente o endereço IP do destino para onde será transmitido os dados. Para descobrir o endereço MAC do destino o protocolo ARP envia um *broadcast* a todos os dispositivos (*hosts*) que se encontram no mesmo segmento de rede, perguntando ao dono do IP (endereço de destino) o seu endereço MAC. Assim que o destino recebe a mensagem de *broadcast* envia uma mensagem com o seu endereço físico. Como se trata de um *broadcast* todos os hosts do mesmo segmento de rede que se encontra recebe essa informação e armazena em uma tabela temporária para que em uma nova transmissão não seja necessário o envio de um novo

broadcast. No momento em que o roteador do segmento de rede possui a informação do IP/MAC do destino, ele encaminha os dados destinados a ele.

Possuindo IP e o MAC pode-se então estabelecer uma comunicação ponto-a-ponto dentro do mesmo segmento de rede. Dependendo da topologia VoIP empregada pode existir a troca de mensagens de mídia entre clientes SIP sem que haja a necessidade de passar pelo SIP *Proxy*.

2.2.3.4 *Transmition Control Protocol* - TCP

Segundo Craig Hunt (2002, p.1):

> O TCP é responsável pela segurança na recepção dos dados ao destino alem de definir todo o processo de início de conexão e multiplexação de múltiplos protocolos da camada de aplicação em uma única conexão, minimizando a conexão múltipla de aplicações com transmissões direcionadas ao mesmo destino.

O protocolo TCP é orientado a conexão, isso significa que para ter um controle sobre os pacotes enviados e certificar que a entrega foi realizada com sucesso, o TCP especifica três fases durante uma conexão: Estabelecimento da conexão, transferência dos dados e término da conexão.

2.2.3.5 *User Datagram Protocol* - UDP

Diferente do TCP na qual existe a garantia e a confirmação de que os dados serão recebidos corretamente no destino, existem situações em que esse comportamento não é necessário, é o que acontece com o protocolo RTP que realiza o tráfego de mídia pela rede. Para esse tipo de aplicação o TCP é substituído pelo UDP, um protocolo não orientado a

conexão, que não necessita estabelecer uma conexão entre origem e destino antes de enviar os dados.

Segundo Tanenbaum (2003, p.400):

> Vale a pena mencionar explicitamente algumas ações que o UDP não realiza. Ele não realiza controle de fluxo, controle de erros ou retransmissão apos a recepção de um segmento incorreto. Tudo isso cabe aos processos do.

O funcionamento do protocolo UDP é muito simples: ele empacota os dados e os envia para camada de rede para que o protocolo IP prossiga ao envio dos dados. Estes pacotes, apesar de serem numerados antes de serem enviados, não sofrem nenhuma verificação de chegada ao destino, podendo chegar desordenado ou mesmo nem chegar.

Pelo fato de o protocolo UDP ser um protocolo considerado muito rápido é amplamente empregado em soluções VoIP, principalmente nos dados referentes aos pacotes de mídia trocado entre os clientes SIP, na qual o ideal é que os pacotes cheguem ao destino com o menor atraso possível.

2.3 SESSION INITIATION PROTOCOL (SIP)

O SIP é um protocolo desenvolvido pelo IETF (*Internet Engineering Task Force*) com a finalidade de estabelecer, modificar e finalizar as transações existentes no VoIP. Por se tratar de um protocolo baseado em texto, o ele oferece flexibilidade, simplicidade e rapidez na entrega dos dados, fazendo dele o protocolo mais popular e utilizado pelas aplicações presentes no VOIP.

Conforme COLCHER, Sérgio et. al (2005, p. 189):

> O principal objetivo do IETF ao definir o protocolo SIP (*Session Initiation Protocol*) como um de seus padrões foi contemplar a criação e o gerenciamento de sessões para troca de fluxos multimídia entre aplicações. O SIP atua desse modo, como um protocolo de sinalização em nível de aplicação. Ele negocia os termos e as condições de uma sessão, definindo, por exemplo, os tipos de mídia e padrões de codificação utilizados na sessão, além de auxiliar na localização dos participantes da mesma.

2.3.1 Componentes da arquitetura

Por se tratar de um padrão, o SIP descreve quais são os componentes necessários em sua arquitetura para que o mesmo posssa prover a comunicação de voz pela rede comutada por pacotes. São eles:

- *SIP User Agent* (UA): Conhecido como o ponto inicial ou final da arquitetura. Um User Agent (UA) interage com o usuário de forma a fazer requisições, agindo como cliente (UAC), ou responsável por responder as requisições, agindo como servidor (UAS). Desta forma, o UA tem um papel de cliente / servidor na arquitetura SIP e pode ser representado na forma de um telefone IP ou um *softphone* rodando em um PC com suporte multimídia.
- Servidor SIP *Proxy* (SIP *Proxy Server*) : Simplificando, um servidor SIP *proxy* atua como um intermediário, fazendo (cliente) ou recebendo (servidor) requisições e as repassando para seu destino final, ou até mesmo para outros servidores. Segundo Dorgham Sisalem et al (2009) normalmente, as tarefas executadas por um servidor *proxy* na recepção de um pedido incluem: Autenticação de uma requisição, decisões de roteamento, decisões de autorização, modificação de pedidos conforme a definição do protocolo SIP.
- Servidor de Redirecionamento (*Redirect Server*): Basicamente, sua função é prover informações de roteamento e redirecionar as requisições para um próximo destino. Segundo Dorgham Sisalem et al (2009, p.43), “ele responde a solicitação de entrada com uma sugestão para o UAC tentar novamente um outro destino(s)”. Essa função pode ser muito útil

nos casos onde um usuário deixa as instalações da sua empresa e deseja continuar recebendo as ligações em sua nova localidade. Outra aplicação para o servidor de redirecionamento é no caso da necessidade de balanceamento de carga.

- Servidor de Registro (*Registrar Server*): É a entidade responsável pelo armazenamento e localização lógica dos UAC. Através de uma mensagem *REGISTER* (falaremos sobre ela com mais detalhes posteriormente), os servidores de registro recebem constantemente as atualizações sobre a localização de cada usuário registrado nele, além de efetuar a resolução de nomes.

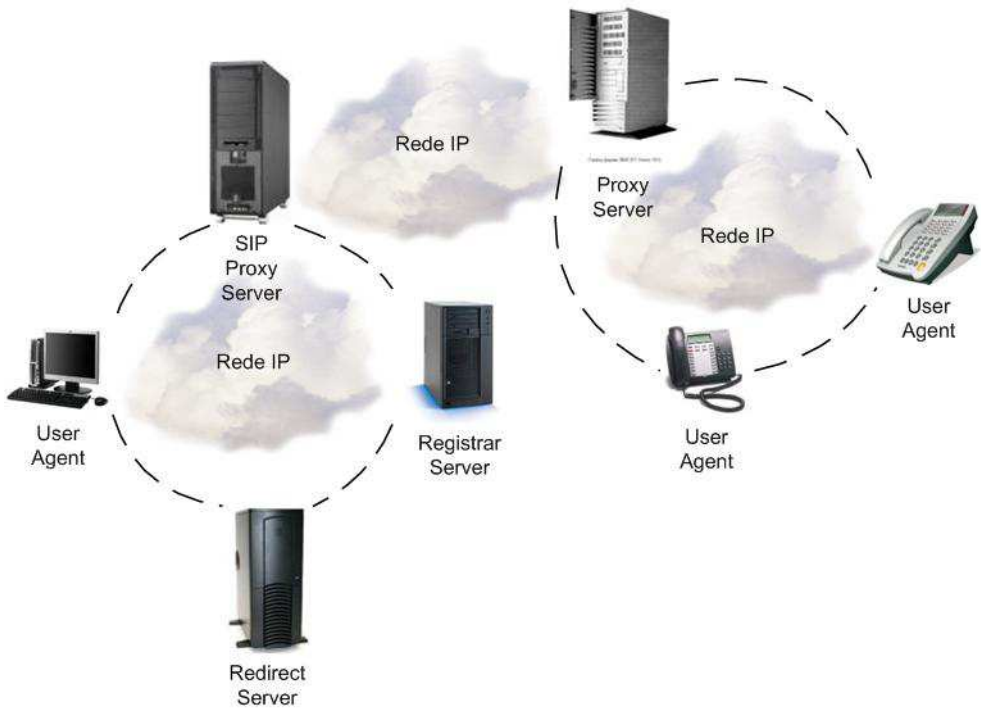

Figura 2 – Componentes do SIP
Fonte: Rezende (2006)

2.3.2 Mensagens SIP

Para estabelecer uma sessão SIP é necessário que haja a interação entre os componentes da arquitetura, para tornar isso possível, foi criado um padrão de mensagens SIP que possibilitam essa comunicação. Dependendo da função exercida, as mensagens podem ser classificadas como requisição ou resposta, a primeira é responsável por realizar requisições que indicam uma ação, abaixo temos uma tabela com o grupo de mensagens de requisição e qual a sua respectiva funcionalidade.

Método	Funcionalidade
INVITE	Convida pessoas para participar de uma sessão
ACK	Confirma o recebimento de uma resposta final para um INVITE
BYE	Solicita o término de uma sessão
CANCEL	Solicita que uma sessão prévia seja cancelada, diferente do BYE
REGISTER	Registra a informação de contato de um indivíduo
OPTIONS	Consulta servidores com respeito a suas capacidades
INFO	Carrega informações de controle geradas durante a sessão
MESSAGE	Permite a transferência de mensagens instantâneas
NOTIFY	Permite a notificação de eventos específicos
PRACK	Confirma a recepção de uma mensagem de resposta informativa
PUBLISH	Publica o estado de um evento
REFER	Solicita que o receptor faça contato com um terceiro participante
SUBSCRIBE	Permite se subscrever para um estado particular de um recurso
UPDATE	Permite a atualização dos parâmetros de uma sessão

Figura 3 – Mensagens de Requisição
Fonte: Rezende (2006)

O grupo de mensagens classificada como de resposta é dividido por classes, as mensagems são separadas dependendo do tipo de informação se deseja transmitir, sempre em resposta a uma mensagem de requisição.

Classe	Descrição	Ação
1xx	Informativo	Indica o status da chamada antes que esta se complete
2xx	Sucesso	A requisição foi recebida com sucesso
3xx	Redirecionamento	O servidor retornou possíveis localidades. O cliente deve reenviar a requisição para outros servidores
4xx	Erro no Cliente	A requisição falhou devido a um erro no cliente
5xx	Falha no Servidor	A requisição falhou devido a um erro no servidor
6xx	Falha Global	A requisição falhou. A requisição não deve ser enviada a este ou outros servidores

Figura 4 – Classificação das mensagens de resposta
Fonte: Rezende (2006)

A próxima tabela lista grande parte das mensagens de resposta definidas na RFC 3261, documento que descreve os padrões dos protocolos presentes na Internet.

100 Trying	380 Alternative Service	410 Gone	482 Loop Detected	501 Not Implemented
180 Ringing	400 Bad Request	413 Request Entity Too Large	483 Too Many Hops	502 Bad Gateway
181 Call Is Being Forwarded	401 Unauthorized	414 Request-URI Too Long	484 Address Incomplete	503 Service Unavailable
182 Queued	402 Payment Required	415 Unsupported Media Type	485 Ambiguous	504 Server Time-out
183 Session Progress	403 Forbidden	416 Unsupported URI Scheme	486 Busy Here	505 Version Not Supported
200 OK	404 Not Found	420 Bad Extension	487 Request Terminated	513 Message Too Large
300 Multiple Choices	405 Method Not Allowed	421 Extension Required	488 Not Acceptable Here	600 Busy Everywhere
301 Moved Permanently	406 Not Acceptable	423 Interval Too Brief	491 Request Pending	603 Decline
302 Moved Temporarily	407 Proxy Authentication Required	480 Temporarily Unavailable	493 Undecipherable	604 Does Not Exist Anywhere
305 Use Proxy	408 Request Timeout	481 Call/Transaction Does Not Exist	500 Server Internal Error	606 Not Acceptable

Figura 5 – Mensagens de Resposta
Fonte: Rezende (2006)

Por ser um protocolo baseado em texto, no SIP toda mensagem é formada por um cabeçalho acompanhado do corpo da mensagem. O cabeçalho por sua vez é composto de uma seqüência estruturada de campos denominados *header fields*, contendo informações sobre o contexto da sessão, como por exemplo, quem a originou (*From*) e qual seu destino (*To*). Dentre os diversos campos especificados, alguns são de presença obrigatória e outros não, abaixo apresentaremos alguns dos mais importantes *header fields* e quais informações contidas nele.

- *Via*: Este campo contém informações da versão SIP, o protocolo da camada de rede, o endereço IP por onde passou a requisição. Ele é usado para gravar a rota de um pedido, através dele, é possível ter uma noção do caminho percorrido pela mensagem até chegar a seu destino final, pois a cada componente da arquitetura SIP que ela passa, é acrescentado um

cabeçalho *Via*, contendo as informações deste componente nos modos do padrão descrito acima.

- *Call-ID*: É o identificador único de uma chamada, formado por um número randômico. Para cada nova chamada, um novo *Call-Id* deve ser gerado.
- *From*: Identifica o transmissor de determinada requisição e formado pelo nome do usuário ou *anonymous* mais o endereço do originador da chamada. Presente tanto em requisições quanto respostas SIP.
- *To*: Este por sua vez indica o destino da chamada, tendo o mesmo formato do cabeçalho *From* assim como sua presença também é obrigatória em mensagens de requisição e respostas.
- *Cseq*: Contém um número sem sinal incrementado a cada requisição feita. Este cabeçalho é utilizado para diferenciar requisições novas de retransmissões ou para identificar de qual requisição determinada resposta pertence.
- *Contact*: O campo *contact* é formado pelo nome (opcional), endereço IP e a porta que enviaram a mensagem, indicando para onde ele deseja receber futuras mensagens SIP intencionadas a ele.
- *Content-Type*: Informa o tipo de conteúdo que está presente no corpo da mensagem.
- *Content-Length*: Informa o número de octetos (tamanho) do corpo da mensagem.

Abaixo é listado a tabela com alguns campos de cabeçalho definidos na RFC 3261.

Accept	Content-Encoding	Min-Expires	Route
Accept-Encoding	Content-Language	MIME-Version	Server
Accept-Language	Content-Length	Organization	Subject
Alert-Info	Content-Type	Priority	Supported
Allow	CSeq	Proxy-Authenticate	Timestamp
Authentication-Info	Date	Proxy-Authorization	To
Authorization	Error-Info	Proxy-Require	Unsupported
Call-ID	Expires	Record-Route	User-Agent
Call-Info	From	Reply-To	Via
Contact	In-Reply-To	Require	Warning
Content-Disposition	Max-Forwards	Retry-After	WWW-Authenticate

Figura 6 – Tipos de cabeçalho
Fonte: Rezende (2006)

Como mencionado anteriormente, além do cabeçalho, uma mensagem SIP é composta também pelo chamado corpo da mensagem, e segundo COLCHER, Sérgio et al (2005,p.196):

> O transporte de informações relevantes para o contexto de algumas operações é feito no corpo da mensagem. Tanto requisições quanto respostas podem transportar informações no corpo da mensagem. A interpretação dessas informações depende da operação à qual uma mensagem se refere, podendo, inclusive, não estar presente.

Outra informação muito importante transportada pelo corpo da mensagem SIP é o *Session Description Protocol* (SDP). Segundo Peter Thermos e Ari Takanem (2008, p.48):

> O *Session Description Protocol* (SDP) é definido pelo IETF na RFC 2327. O objetivo do SDP é comunicar recursos de mídia e propriedades desejadas entre as partes comunicantes. SDP pode ser usado em conexão com diferentes protocolos de sinalização, *gateway* de mídia e protocolos de controle. A descrição da sessão SDP é representada em uma lista de variáveis e seus parâmetros.

No processo de estabelecimento de uma sessão SIP, é necessário que haja uma negociação de mídia e uma série de outras informações para que a transição dela possa ser realizada com sucesso, portanto, fica a cargo do SDP fazer este trabalho. A figura a seguir indica os parâmetros responsáveis por compor do cabeçalho SDP e seus respectivos significados.

Parâmetros SDP	Descrição
v=0	Numero da versão
o=Anderson 2890844526 2890844526 IN IP4 laboratorio.faculdade.com.br	Nome do originador
s=Phone Call	Assunto
c=IN IP4 100.101.102.103	Conexão
t=0 0	Tempo
m=audio 49170 RTP/AVP 0	Mídia
a=rtpmap:0 PCMU/8000	Atributos

Figura 7 – Cabeçalho SDP
Fonte: Rezende (2006)

Ordem específica das linhas da descrição da Sessão:
v – versão do protocolos o – originador ou criador da sessão e identificador da sessão s – nome da sessão i – informação sobre sessão u – URI da sessão e – endereço de e-mail p – número do telefone c – informação sobre conexão b – informação sobre largura de banda z – ajustes de time zone k – chave de encriptação a – zero ou mais linhas de atributo da sessão
Ordem específica das linhas da descrição do horário:
t – horário que a sessão esta ativa r – zero ou mais horários repetidos
Ordem específica das linhas da descrição da mídia:
m – nome da mídia e endereço de transporte i – título da mídia c – informação sobre a conexão b – informação sobre a largura de banda k – chave de encriptação a – zero ou mais linhas de atributo da mídia

Figura 8 – Descrição do cabeçalho SDP
Fonte: Rezende (2006)

2.3.3 Cenários e troca de mensagens

Como mencionado anteriormente, para que exista iteração entre os UAs presentes na arquitetura VoIP é necessário que os mesmos troquem informações durante algum tipo de transação. Essa troca de informação é feita através de mensagens SIP que trafegam na rede IP formando cenários específicos para cada tipo de requisição e a resposta retornada. Devido a grande quantidade de cenários possíveis apenas dois serão apresentados, sua escolha deve-se ao fato da importância no contexto de uma comunicação de voz entre dois usuários utilizando uma rede comutada por pacotes. Primeiramente é descrito o cenário relacionado ao registro de um usuário em um servidor SIP *Proxy*.

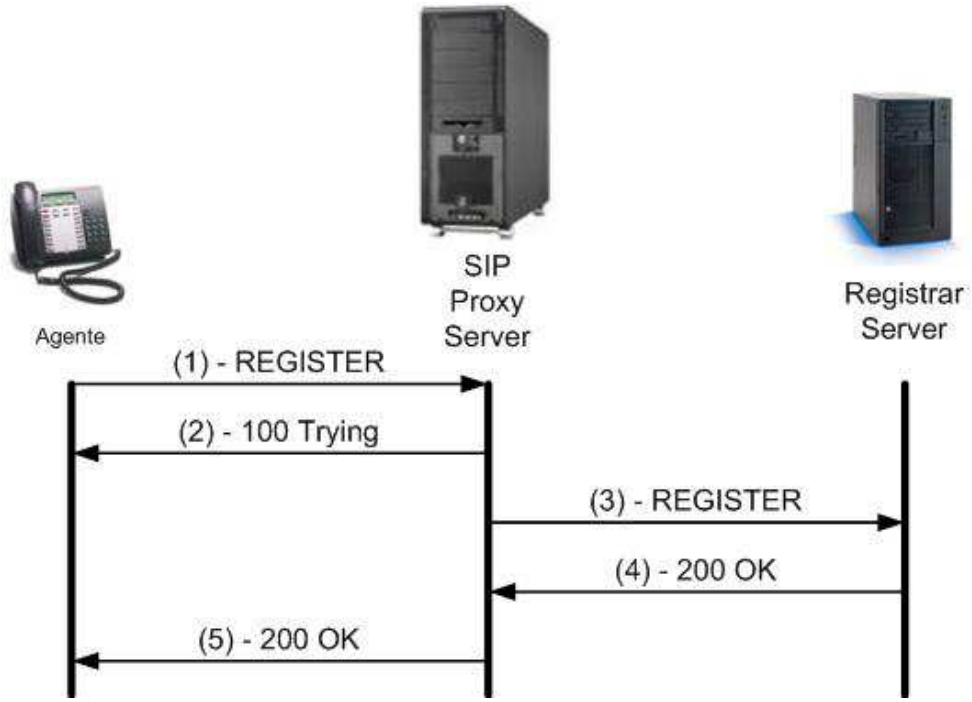

Figura 9 – Fluxo de mensagens para registro
Fonte: Rezende (2006)

Segundo RFC3665 (http://tools.ietf.org/rfc/rfc3665.txt), para realizar um registro, o UA envia uma mensagem *REGISTER* ao servidor *proxy*, que responde com um *100 Trying*, indicando que a requisição foi recebida e está sendo processada, evitando assim a retransmissão de mensagens *REGISTER* por parte do UA. O servidor *proxy* encaminha a mensagem recebida para o servidor de registro (lembrando que eles podem estar presentes na mesma máquina) que fará o armazenamento da localização do usuário. Finalizado com

sucesso, o mesmo envia uma resposta *200 OK* para o servidor *proxy* que reencaminha esta mensagem para o UA, notificando que o registro foi efetuado. Estando registrado o UA se torna apto a receber chamadas, mesmo que o usuário chamador tenha apenas a localização (IP) do servidor de registro.

O próximo cenário descreve a troca de mensagens necessárias para iniciar e finalizar uma sessão multimídia entre dois UAs.

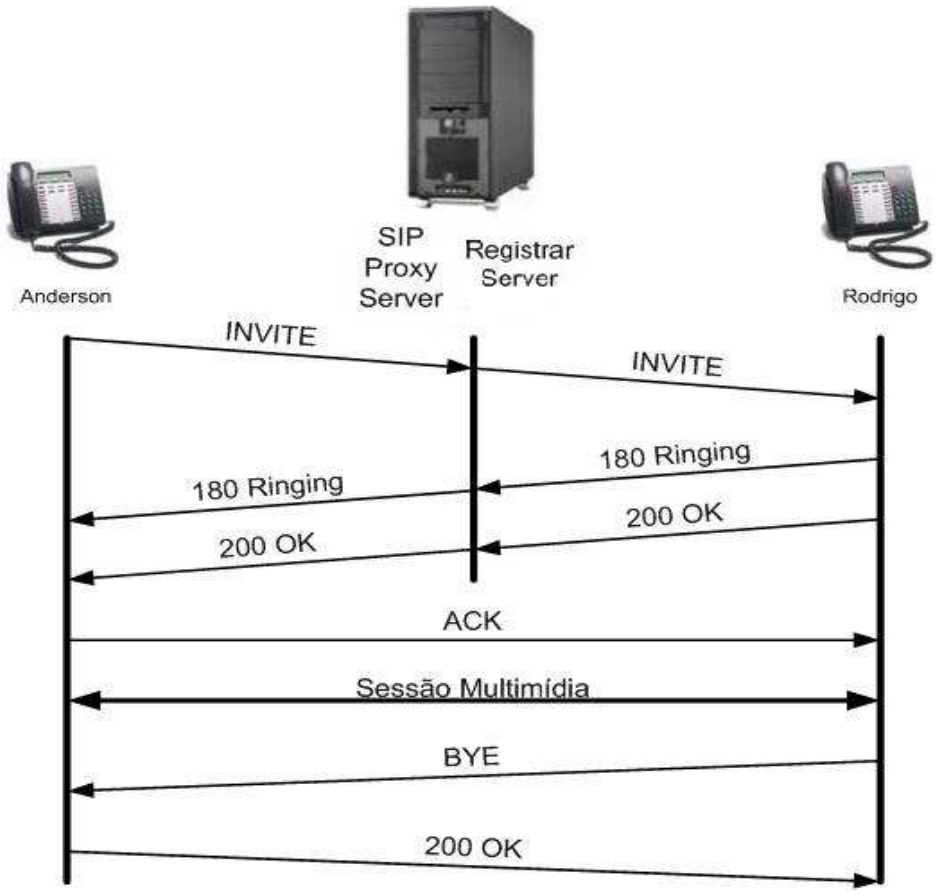

Figura 10 – Fluxo de mensagens para estabelecer uma chamada
Fonte: Rezende (2006)

Como ilustrado na figura, o servidor *proxy* e de registro se encontram na mesma máquina, assim, a troca de mensagem entre eles será feita internamente. Partindo do pressuposto de que o usuário chamado esteja registrado no servidor, e que o usuário chamador conheça o IP do servidor SIP (*proxy* e registro).

Segundo RFC3665 (http://tools.ietf.org/rfc/rfc3665.txt), primeiramente o UA chamador envia uma mensagem de requisição *INVITE* para o servidor SIP informado que deseja estabelecer comunicação com um determinado usuário. Ao receber essa requisição, o

servidor SIP certifica-se de que o usuário destino se encontra registrado em sua plataforma, em caso afirmativo, verifica a localização (IP) e encaminha a mensagem para o UA chamado. Quando receber o pedido de início de sessão multimídia, o UA chamado responde a requisição com uma mensagem *180 Ringing* indicando o recebimento do *INVITE* e informando que está a espera do atendimento da chamada por parte do seu usuário, para então estabelecer a conexão entre as partes envolvidas. Até que a chamada seja atendida, o servidor SIP funciona apenas como *proxy*, reencaminhando aos locais corretos.

No momento do atendimento, o UA chamado envia uma mensagem *200 OK* confirmando a ação, agora ambos os UAs já conhecem as respectivas localidades de seu parceiro de comunicação, excluindo então o servidor SIP de participar das próximas trocas de informações (este comportamento pode variar dependendo da configuração da estrutura VoIP). Para informar que já está ciente do atendimento da chamada, o UA chamador envia um *ACK*. Com todos os parâmetros necessários já trocados, os UA iniciam a sessão multimídia responsável por transportar a voz através da rede. Para finalizar a chamada, o usuário chamado envia uma mensagem *BYE*, que é confirmada pelo usuário chamado através de um *200 OK* finalizando a chamada.

2.3.4 *Real Time Transport Protocol* (RTP)

O objetivo maior do VoIP é possibilitar que duas ou mais pessoas consigam se comunicar por uma rede de dados através de voz e/ou vídeo. Para estabelecer e negociar todos os parâmetros necessários na comunicação utiliza-se o protocolo SIP como apresentado, porém, o transporte da sessão multimídia será responsabilidade de um outro tipo de protocolo, denominado RTP.

Conforme Antônio Tadeu A. Gomes et. al (2005, p. 140):

> O RTP é um protocolo que oferece funções de transporte de rede fim a fim direcionadas para aplicações que transmitem fluxos de dados em tempo real, como áudio, vídeo e dados de simulações, através de serviços de rede unicast e multicast.

Por se tratar de um protocolo que transporta dados de tempo real, é desejado que seja eliminado qualquer fator que implique no atraso da entrega das mídias, por este motivo o

RTP foi desenvolvido para ser um protocolo simples e que geralmente é transportado pelo protocolo UDP, otimizando assim a entrega dos pacotes na rede.

2.4 SEGURANÇA

Por natureza, o VOIP já surgiu rodeado pelos diversos tipos de ataque existentes na rede IP, devido ao seu crescimento, os ataques começaram a ser realizados diretamente ao protocolo responsável pela troca de informações desta tecnologia, neste caso em específico o SIP. Diantes deste novo panorama criou-se a necessidade de desenvolver ferramentas que fossem capazes de impedir ou dificultar estes ataques, consequentemente, os estudos e o desenvolvimento de métodos para garantir a segurança das redes VOIP foram crescendo junto com a tecnologia. Para compreender melhor os métodos de segurança que serão apresentados a seguir, são necessários cinco princípios básicos de segurança listados abaixo, para que se possa obter o efeito desejado da solução:

- Confiabilidade: Impede que pessoas não autorizadas tenham acesso ao conteúdo da mensagem, garantindo que apenas a origem e o destino tenham conhecimento.
- Integridade: Serviço que garante que a mensagem não sofreu nenhum tipo de alteração ao longo do caminho de transmissão.
- Disponibilidade: As informações e serviços devem estar disponíveis para os autorizados.
- Autenticação: Pode-se aplicar tanto a uma entidade ou mensagem. No primeiro caso é a garantia do emissor sobre a identidade da outra parte envolvida (autenticação de identidade), já no segundo, permite que o destinatário tenha condições de verificar se a mensagem recebida foi realmente originada por quem alega ter produzido.
- Não-repúdio: Mecanismo de impede um remetente de negar posse de uma mensagem produzida e enviada por ele a um destinatário.

2.4.1 Criptografia

A criptografia é uma técnica empregada em diversos métodos de segurança, transformando textos originais ou texto claro, em uma informação transformada, chamada de texto cifrado, permitindo que somente um legítimo destinatário a decifre e compreenda-a. Podemos distinguir os métodos de criptografia entre.

- Criptografia de chave simétrica: Utiliza uma única chave que tem a função tanto de cifrar como de decifrar as mensagens. Pela sua característica, os algoritmos simétricos exigem que a chave seja mantida secreta pelos dois participantes da comunicação, necessitando de um canal seguro para a transmissão das chaves.
- Criptografia de chave assimétrica: Este tipo de criptografia utiliza chaves distintas, uma para cifrar e outra para decifrar as mensagens. A chave utilizada para cifrar uma mensagem é denominada chave pública, já a chave que decifra as mensagens é conhecida como chave privada.

Na maioria dos casos, os algoritmos simétricos são mais eficientes que os assimétricos, porém, possuem o problema de compartilhamento das chaves. Geralmente é feito a utilização destes dois algoritmos em conjunto a fim de eliminar o problema de compartilhar o segredo da chave, mas tudo vai depender da política de segurança adotada pelos gerenciadores da arquitetura.

2.4.2 Assinatura Digital

Assinatura digital pode ser entendido com um processo de assinatura eletrônica baseado no sistema criptográfico, utilizado para autenticação de informação digital utilizada como prova inegável de que uma mensagem veio do emissor, fornecendo ao receptor os requisitos de autenticidade, integridade e não repúdio.

Segundo Sisalem (2009, p.13), a assinatura digital permite um remetente assinar uma mensagem usando sua chave privada de modo que qualquer receptor que contém a chave pública do remetente pode verificar a assinatura digital, em caso de sucesso, conclui-se que a mensagem é genuína.

2.4.3 Certificados Digitais e Autoridades Certificadoras

Além dos conceitos de criptografia e assinatura digital, existem outros mecanismos para auxiliar alguns métodos segurança VOIP, conhecidos como certificados digitais e autoridades certificadoras.

Segundo Joel Guilherme (http://www.sbis.org.br/Criptografia.doc):

> Um certificado digital nada mais é que um documento (eletrônico) contendo a chave pública de um usuário (ou processo) e os dados de identificação do mesmo. Este documento deve ser assinado por uma autoridade confiável, a Autoridade Certificadora, atestando sua integridade e origem. Usualmente, certificados digitais são utilizados para garantir a integridade e origem de chaves públicas depositadas em bases de dados de acesso público.

2.4.4 *HTTP Digest Authentication*

A autenticação *Digest* é um mecanismo de segurança simples baseado em *hashes* criptográficas que evitam a transferência de senha do usuário sem nenhum tipo de proteção. Para autenticar, um servidor gera um desafio *digest* que consiste em um grupo de parâmetros descritos abaixo:

- *realm*: Parâmetro obrigatório e precisa estar presente em todos os desafios. Seu objetivo é identificar credenciais dentro de uma mensagem SIP e normalmente é definido com o nome do servidor *proxy* no qual ele é responsável.
- *nonce*: É uma *string* dados gerada toda vez que um desafio é lançado pelo servidor. Os dados geralmente incluem carimbo de tempo e uma frase

secreta garantindo que após algum tempo ela irá expirar limitando a vida útil de resposta do usuário.

- *opaque*: Este parâmetro contém uma *string* de dados opaco passado ao usuário permitindo que os servidores não mantenham estado.
- *algotithm*: Algoritmo usado para calcular *hashes*. Geralmente é utilizado o MD5 ou SHA-1.
- qop: Indica quais os tipos de proteção são suportados pelo servidor .

Ao receber um pedido de autenticação do servidor com um desafio *digest*, o usuário gera uma resposta similar ao desafio.

- *uri*: Este parâmetro contém a URI de destino final da mensagem .
- *qop*: O nível de proteção escolhida.
- *nc*: Contagem de *nonce* possui um valor hexadecimal que permite o servidor detectar repetições de requisições.
- *cnonce*: *String* opaca enviada pelo cliente para evitar ataque, fornecer autenticação e proteção de integridade da mensagem.
- response: Uma *string* computada pelo UA para provar que o usuário conhece uma senha.

Apesar de garantir a identidade do usuário, este método de segurança pode ser vulnerável a ataques de dicionário, que consistem em cifrar as palavras de um dicionário e fazer a comparação com arquivos de senhas de usuário. Outro ponto vulnerável é que *o HTTP Digest Authentication* não protege a integridade das mensagens SIP.

2.4.5 TLS (*Transport Layer Security*)

O TLS é um mecanismo de segurança projetado para o transporte seguro do TCP através de mensagens criptografadas. No SIP, seu uso deve ser feito durante todo o caminho que a mensagem irá percorrer e necessita de uma estrutura de chave assimétrica para cifrar e decifras as mesmas.

Segundo Peter Thermos e Ari Takanem (2008), o protocolo é composto de duas camadas: o protocolo TLS *Record* e o protocolo *Handshake*. O TLS *Record* tem a finalidade de manter a conexão segura entre os pontos finais, já o TLS *Handshake* é responsável pela autenticação mútua entre cliente/servidor e por negociar as propriedades de criptografia (por exemplo, os algorítimos de criptografia e chaves) da respectiva sessão. A RFC do SIP recomenda o uso do TLS para fornecer proteção necessária contra ataques de interceptação, mensagens de falsificação, repetição de mensagens e assim por diante.

Abaixo se encontra os pontos fortes e fracos relacionados ao TLS.

- Pontos Fortes: Suporta autenticação mútua de certificados, fornece confidencialidade e integridade das mensagens, pode proteger a negociação de chaves criptográficas, utilizado amplamente em aplicações da Internet e baixo impacto no desempenho.
- Pontos Fracos: Requer infraestrutura de chaves públicas para impor autenticação mútua na camada SSL, requer o encerramento e a criação de uma nova sessão para cada salto, não pode ser utilizado em estruturas com transporte UDP e é sensível a ataques de DoS.

2.4.6 *Secure* MIME (S/MIME)

S/MIME (*Secure / Multipurpose Internet Mail Extensions*) é um mecanismo de segurança desenvolvido para fornecer confidencialidade, integridade e autenticação para protocolos como SMTP e SIP. O objetivo do S/MIME é a interoperabilidade de mensagens seguras, obtida pela criptografia e assinatura digital.

Segundo Peter Thermos e Ari Takanem (2008), o S/MIME pode ser usado para proteger cabeçalhos de uma mensagem SIP, exceto o cabeçalho Via. Ao contrário do TLS que abrange toda a mensagem, o S/MIME fornece uma maior flexibilidade permitindo proteger seletivamente porções da mensagem SIP, além de poder ser utilizado para transporte sobre o protocolo UDP e TCP.

Para concluir sobre o S/MIME, apresentamos abaixo os pontos fortes e fracos deste mecanismo de segurança.

- Pontos Fortes: Pode ser usado com UDP e TCP, flexibilidade, ao proteger partes da mensagem SIP e fornece confidencialidade de ponta-a-ponta, integridade, autenticação e não repúdio.
- Pontos Fracos: Requer mais esforço para sua implantação devido à complexidade e requisitos de infra-estrutura, difícil integração com as atuais infra-estruturas e escalabilidade questionável.

2.4.7 IP *Security* (IPsec)

IPsec é um mecanismo de propósito geral que pode ser usado para proteger mensagem SIP na camada de rede usando UDP ou TCP como transporte. Este método opera sobre a camada de rede (camada 3) do modelo OSI e pode oferecer confidencialidade, integridade e autenticação para sinalização e mensagens multimídia através de túneis seguros entre os pontos finais da comunicação. O IPSec combina diversas tecnologias para prover uma melhor segurança, como mecanismos de troca de chave, criptografia de chave pública, algoritmos de criptografia para grandes volumes de dados.

Segundo Peter Thermos e Ari Takanem (2008), os pontos fracos e fortes do IPSec são:

- Pontos Fortes: Protocolo de segurança comprovado e amplamente utilizado, opera na camada de rede para suportar UDP, TCP, SIP e RTP, oferece proteção contra ataque de espionagem, DoS, entre outros e fornece confidencialidade, integridade, autenticação e não repúdio.
- Pontos Fracos: Infra-estrutura complexa para implementar, componentes intermediários devem ser confiáveis e não é bem dimensionado para grandes redes e aplicações distribuídas.

2.4.8 *Security* RTP (SRTP)

O SRTP é um padrão criado pelo IETF para garantir a confidencialidade e a integridade do áudio transportado pelo RTP. Com o uso do SRTP, mesmo que o tráfego de áudio consiga ser capturado por um terceiro, através de ataques na sinalização, será impossível remontar o áudio sem a chave de criptografia. O uso SRTP para proteção de mídia e de um protocolo chamado Mikey (*Multimedia Internet Keying*) definido na RFC3830 para troca de chaves, gera uma combinação de mecanismos de segurança adequada para aplicações multimídia, incluindo VOIP, vídeo e conferência.

Segundo Peter Thermos e Ari Takanem (2008), os pontos fortes e fracos do RTP são:

- Pontos Fortes: Fornece proteção contra ataque ao RTP e RTPC, suporte que permite a recepção de pacotes fora de ordem e minimiza o consumo de recursos computacionais para a geração de chaves criptográficas.
- Pontos Fracos: Não é possível manter de ponta a ponta, a integridade de mensagens e autenticação quando o fluxo de mídia é enviado de uma rede IP para uma rede SS7 (PSTN). Impactos de processamento e no consumo de recursos em grandes grupos *multicast*, isso não é desejável para dispositivos móveis com recursos computacionais limitados.

2.5 VULNERABILIDADES DOS SISTEMAS VOIP BASEADOS EM SIP

Como o VOIP usufrui da estrutura de transporte dos dados da rede, ele está sujeito a todo o tipo de ataque nela existente. Facilitado pela estrutura textual do SIP, todas as informações transportadas pelas mensagens para o estabelecimento de uma sessão podem ser facilmente capturadas, caso não haja mecanismos que assegurem a integridade destes dados, oferecendo oportunidades para ataques de falsificação, sequestro e adulteração de mensagens.

Segundo RFC3067 (http://www.ietf.org/rfc/rfc3067.txt), a vulnerabilidade é definida como uma falha ou fraqueza na concepção de um sistema, aplicação ou funcionamento de gestão que poderiam ser exploradas para violar as politicas de segurança do

sistema. Já os ataques, segundo Thermos e Takanen (2008), é uma tentativa de impactos ativos de infra-estrutura e operações realizados por um agente de ameaça, que para ser bem sucedido depende da vulnerabilidade do sistema e das contramedidas existentes. Um ataque pode ser ativo, resultando na alteração de dados, ou passivo, resultando na liberação de dados.

2.5.1 *Registration Hijacking*

Segundo Anderson Rodrigues Souza Rezende (2006), o ataque de *hijacking* ou sequestro é uma forma de se obter o controle de uma conexão iniciada por um UA legítimo, impedindo o mesmo de fazer uso do sistema e tomando seu lugar. O *Registration Hijacking* (Sequestro do Registro) ocorre quando um atacante falsifica um UA válido para um SIP *Register*, alterando a inscrição legítima para seu próprio endereço. Assim todas as chamadas são enviadas para o UA registrado pelo atacante.

2.5.2 *Session Tear Down*

Session tear down ocorre quando o atacante observa a sinalização da chamada, e então envia um falso SIP *BYE* ou um *re-INVITE* para os participantes. A maioria dos UAs não trabalha com uma forma de autenticação forte, o que permite ao atacante enviar o SIP *BYE* para encerrar a chamada, ou um *re-INVITE* para alterá-la.

Este tipo de ataque derruba as chamadas ou altera parâmetros importantes dela, sem o consentimento dos usuários que estabeleceram essa sessão. O Desligamento abrupto e as alterações nos parâmetros de envio dos pacotes de mídia são causas comuns deste ataque.

2.5.3 *Proxy Impersonation*

Segundo Anderson Rodrigues Souza Rezende (2006), o ataque de *Proxy* Impersonation, ocorre quando um atacante engana um dos SIP UAs ou *proxy* do sistema de comunicação com um falso *proxy*. Se o atacante obtiver sucesso no ataque, este terá acesso à todas as mensagens SIP, e também ao controle total da chamada.

Se passando por um falso *proxy*, toda troca de mensagens SIP entre os componentes da plataforma é destinada quem realizou o ataque, permitindo manipular essa comunicação conforme desejado. Redirecionamentos, desligamentos, negação de serviço e uso ilícito de serviços da plataforma são alguns dos efeitos colaterais causados pelo *Proxy Impersonation*.

2.5.4 *Denial of Service* (DoS)

Denial of Service (DoS), ou Negação de serviço, é ataque que visa deixar os recursos do sistema indisponíveis para seu utilizadores e consiste em enviar uma quantidade de mensagens mais do que um determinado alvo pode suportar. Pode ocorrer por quaisquer meios descritos nos outros ataques, ou por um ataque de DoS especifico. Por ser raramente usada uma autenticação forte, componentes SIP podem processar mensagens de atacantes. O DoS pode ser especialmente prejudicial se o alvo for um recurso chave do sistema como o *Media Gateway* ou um SIP o *proxy* por exemplo.

A finalidade maior deste ataque é derrubar os serviços oferecidos pelo dispositivo atacado, no caso de um servidor SIP com funções de *Proxy* e Registro por exemplo, os usuários da plataforma ficariam impossibilitados de realizar registros ou estabelecer algum tipo de comunicação que dependesse do SIP *Proxy* para ser estabelecida.

2.5.5 *Eavesdropping*

Segundo Anderson Rodrigues Souza Rezende (2006), o *Eavesdropping* em redes VoIP requer a interceptação da sinalização e do respectivo fluxo de mídia da conversação. As mensagens de sinalização utilizam protocolos como UDP e TCP e o fluxo da mídia tipicamente são transportadas sobre UDP utilizando o protocolo RTP. Com o uso de ferramentas como *sniffing* de rede, é possível capturar a sinalização e a mídia transportada obtendo informações completas sobre determinados usuários.

O *eavesdropping* é muito útil quando os pacotes SIP e RTP trafegam na rede sem nenhum método de proteção das informações. Ele é extremamente necessário quando se deseja saber informações sobre usuários e servidores que compõem rede VoIP, ou capturar pacotes RTP com o áudio das chamadas estabelecidas.

Neste trabalho, foi utilizado o *eavesdropping* apenas para capturar as mensagens SIP, o ataque ao RTP não faz parte nos objetivos propostos.

2.5.6 *Flooding attack*

Tipo de ataque onde o objetivo do atacante é consumir os recursos da rede e do sistema gerando um DoS, assim, serviços usuais da rede ficarão indisponíveis. Como o VOIP depende da disponibilidade da infraestrutura de dados, ele também ficará indisponível. Vale lembrar que existem ataques de *flooding* realizados diretamente ao protocolo SIP.

2.5.7 *Message Tampering*

O ataque de *Message Tampering* ou adulteração de mensagens ocorre quando o atacante intercepta e modifica pacotes trocados entre componentes SIP. Este ataque pode acontecer através de *Resgistration Hijacking*, *Proxy Impersonation*, ou ataque a um

componete SIP com direitos para processar mensagens, tais como *proxy*, media *gateway* ou *firewall.*

Este tipo de ataque é focado em fazer alterações nas mensagens SIP de acordo com o tipo de dano que deseja causar no dispositivo alvo. Existem vários tipos de ataques que podem ser realizados adulterando uma mensagem SIP, dentre eles, podemos citar uma requisição *NOTIFY* que realiza o *reboot* automático de telefones IP, sem que o usuário desse dispositivo tenha permitido essa ação.

3 DESENVOLVIMENTO

O fato do SIP ser um protocolo texto facilita a interceptação dos dados nele transmitidos caso nenhuma política de segurança seja tomada para impedir este propósito. A captura pode ser feita através de ferramentas que filtram os pacotes que trafegam na rede as quais são encontradas facilmente na internet e se utilizada para fins maliciosos podem revelar informações importantes a quem deseja se aproveitar das fragilidades existentes no protocolo.

Qualquer pessoa com conhecimento médio ou avançado no conteúdo apresentado na fundamentação teórica deste trabalho, pode ser considerada apta a realizar ataques que danifiquem ou comprometam uma estrutura VoIP, principalmente se ela foi criada sem a preocupação com a visibilidade e integridade dos dados.

3.1 MONTAGEM DO CENÁRIO PARA O ESTUDO DE CASO

Pensando em simular algumas das vulnerabilidades do protocolo SIP e analisar qual impacto que elas podem oferecer, montou-se um cenário com os componentes

necessários para trafegar voz e vídeo através de uma rede de comunicação de dados. Segue abaixo figura ilustrativa.

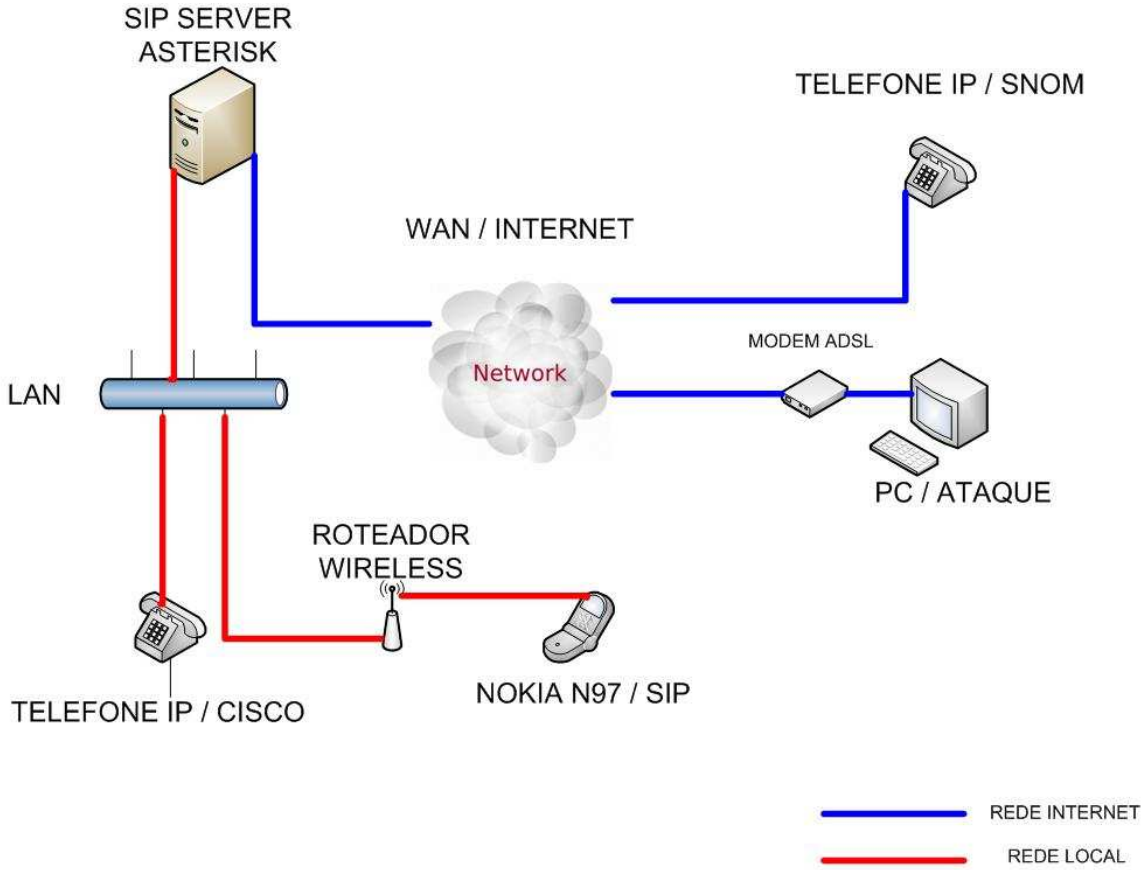

Figura 11 – Cenário do estudo de caso

Na figura temos como componente central da arquitetura o servidor SIP, ele exerce funções de registro, encaminhamento e redirecionamento das mensagens a ele destinadas. No mercado é possível encontrar uma variedade de *softwares* que possuem as características de funcionamento descritas acima, para este trabalho de conclusão de curso foi decidido utilizar o *Asterisk*. Sua escolha deve-se ao fato do *Asterisk* ser um *software* livre mundialmente conhecido e utilizado, seu desenvolvimento é bem consolidado, possui boa documentação e oferece todos os requisitos necessários para a realização do estudo de caso proposto.

Para simular os usuários conectados a rede, utilizamos quatro tipos diferentes de *User-Agent*. O primeiro é o *softphone* X-Lite, desenvolvido para computadores pessoais com suporte a multimídia, não se trata de um *software* livre mas está disponível gratuitamente na página do fabricante CounterPath. O segundo UA é um SIP *Client* rodando na plataforma

Symbian para celulares Nokia, uma espécie de *softphone* nativo para celulares dessa marca. Por fim temos dois telefones IP de fabricantes diferentes, o primeiro é fabricado pela Cisco e o segundo pela Snom.

O último componente dessa arquitetura é o PC que realizará os ataques, nessa máquina foram compilados duas ferramentas, SIPp e siprtp, desenvolvidas especificamente para simulação de testes em redes VoIP que utilizam o protocolo SIP para comunicação. Apesar de não terem sido desenvolvidas para realizar ataques, elas podem exercer tal função e foram utilizadas para este propósito. Segue descrição mais detalhada destas ferramentas.

3.1.1 SIPp

O SIPp é um *software* livre desenvolvido pela empresa HP com finalidade de testar, simular e gerar tráfego de mensagens SIP. Esta ferramenta permite ao usuário montar qualquer tipo de cenário, através de arquivos com formato XML, simulando sessões existentes em uma arquitetura VoIP real. Para a execução destes cenários o SIPp fornece uma gama de configurações ao usuário, como por exemplo, a quantidade de vezes que a rotina será executada, o intervalo de execução, o destino de entrega dos pacotes SIP, entre outras configurações que são descritas com mais detalhes no site ou no help do programa.

Para utilizar o SIPp no estudo de caso, foi necessário fazer o *download* do código fonte no site , instalar o compilador C++, a biblioteca ncurses e OpenSSL na máquina que realizou os ataques. Para compilar, ou seja, criar um executável a partir do código fonte do programa, foi necessário executar o comando "*make ossl*" no diretório principal onde o código fonte foi armazenado. O parâmetro "*ossl*" indica ao compilador que o SIPp deve ser compilado com suporte a autenticação pois é necessário para rodar cenários de registro de usuário.

3.1.2 siprtp

O siprtp é um *open source* desenvolvido para estabelecer chamadas utilizando o protocolo SIP, e gerar tráfego RTP entre elas. Esta ferramenta pode ser executada em dois modos de operação, cliente ou servidor, permitindo ao usuário determinar a quantidade e duração das chamadas que se deseja criar ou receber. Enquanto as chamadas estiverem estabelecidas e gerando tráfego RTP, o siprtp disponibiliza ao usuário um menu com estatísticas de tempo, *delay*, pacotes enviados e recebidos, *payload, jitter* e porcentagem de perda de pacotes do total de chamadas ou de uma em específico. Essa característica, permite aos usuários obter informações do comportamento relevantes sobre os testes com essa ferramenta e analisar o impacto delas na rede VoIP.

Para utilizar o siprtp no estudo de caso, além de fazer o *download* do código fonte no site foi necessário realizar algumas alterações para que o programa executasse de acordo com as necessidades do estudo de caso. Por *default*, a ferramenta suporta gerar/receber apenas 32 chamadas simultâneas, como o intuito do ataque que utilizou essa ferramenta é gerar uma quantidade grande de chamadas para o servidor SIP, foi acessado o arquivo config.h, localizado em pjlib/include/pj/, e alterado o valor do define PJ_IOQUEUE_MAX_HANDLE para 1024. Essa primeira alteração permite ao siprtp gerar/receber até 512 chamadas simultâneas.

Outra alteração realizada em código, foi a inclusão da identificação do usuário no cabeçalho *from* da mensagem de *INVITE* enviada para o servidor SIP. Por *default*, a URI que compões o *from* não especifica a identificação do usuário que deseja estabelecer a comunicação, esta característica não é suportada pelo Asterisk, que nega a requisição com uma mensagem “404 – User Not Found”. Para incluir uma identificação válida no cabeçalho *from*, foi necessário acessar o arquivo siprtp.c, e inserir a identificação na linha 985 que monta a URI do cabeçalho *from* para inserir na mensagem de INVITE.

Feito as alterações descritas acima, bastou executar o comando “./config & make & make install” no diretório raiz do código fonte para que o programa fosse compilado.

3.2 ATAQUE DE ROUBO DE REGISTRO

O primeiro ataque realizado no cenário do estudo de caso foi baseado no roubo de registro dos usuários legítimos da plataforma, sua escolha deve-se ao fato de que possuir um registro roubado impede o proprietário de usufruir dos serviços a ele ofertados passando a ser utilizado por um usuário ilícito, além de facilitar outros tipos de ataque que possam danificar gravemente qualquer componente da estrutura. Para realizar o roubo de registro é necessário primeiramente obter o IP e porta do servidor SIP, além da conta e senha do respectivo usuário. Uma mensagem SIP trafegando abertamente na rede pode ser facilmente capturada e fornece algumas informações importantes para um ataque, como a conta do usuário e o IP e porta do servidor SIP. A senha vai geralmente "escondida" no cabeçalho *Authorization* utilizando a política de criptografia definida pelo servidor durante a negociação, porém, existem várias técnicas de ataque para obtê-la e não estão relacionadas com vulnerabilidades do protocolo SIP especificamente. Por não ser o foco deste trabalho, considerou-se que a senha da conta atacada já foi capturada utilizando algum outro método de ataque para este propósito.

Tendo em mãos todos os dados necessários para realizar o ataque, utilizou-se a ferramenta SIPp para gerar o pedido de desregistro do usuário legítmo através de um cenário desenvolvido para este propósito. Os dados gerados ao término da execução do programa são mostrados na figura abaixo.

```
------------------------------ Scenario Screen -------- [1-9]: Change Screen --
  Call-rate(length)   Port   Total-time  Total-calls  Remote-host
  10.0(0 ms)/1.000s   9878       0.20 s            1  10.0.10.12:5060(UDP)

  Call limit reached (-m 1), 0.000 s period  0 ms scheduler resolution
  0 calls (limit 30)                    Peak was 1 calls, after 0 s
  0 Running, 3 Paused, 0 Woken up
  0 dead call msg (discarded)           0 out-of-call msg (discarded)
  1 open sockets

                              Messages  Retrans   Timeout   Unexpected-Msg
    REGISTER ---------->      1         0         0
         200 <----------      0         0         0         0
         401 <----------      1         0         0         0
    REGISTER ---------->      1         0         0
         200 <----------      1         0         0         0
------------------------------ Test Terminated --------------------------------
```

Figura 12 – Estatísticas do fluxo de mensagens do ataque de roubo de registro

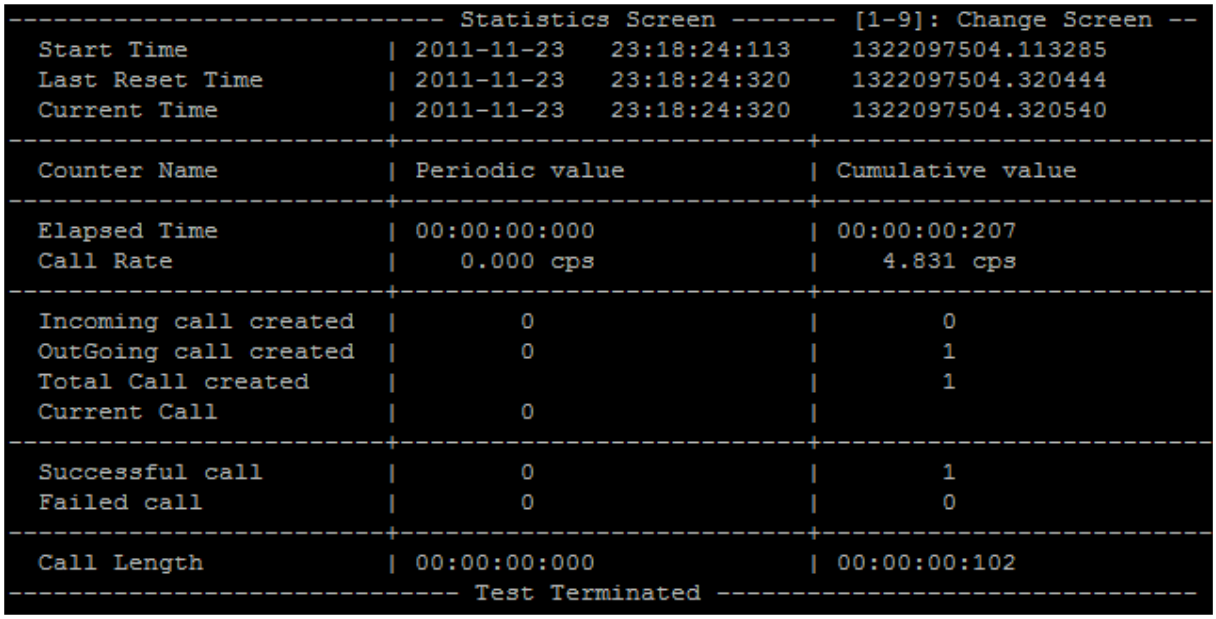

Figura 13 – Estatísticas gerais do ataque de roubo de registro

Pelas informações disponibilizados pelo SIPp, é possível perceber que o desregistro foi realizado com sucesso. Primeiramente, a ferramenta lista todos os dados relacionados ao cenário que foi executado e a troca de mensagens SIP com o *Asterisk*, depois, é mostrado as estatísticas finais da comunicação informando o sucesso da chamada.

A única diferença do processo de registro e desregistro de um usuário é o valor do campo Expires da mensagem *REGISTER*, todo o fluxo de mensagens trocadas com o servidor SIP é o mesmo para ambos. Quando o valor do *Expires* for igual a zero, indica que a requisição é de desregistro, quando for maior que zero, indica o tempo que o usuário vai permanecer registrado. Para comprovar a validade da mensagem pelo SIPp, foi capturado todos os pacotes SIP que trafegavam pela interface de rede da máquina de ataque no momento em que a ferramenta foi executada, retirado apenas a mensagem de desregistro como ilustrado na figura abaixo.

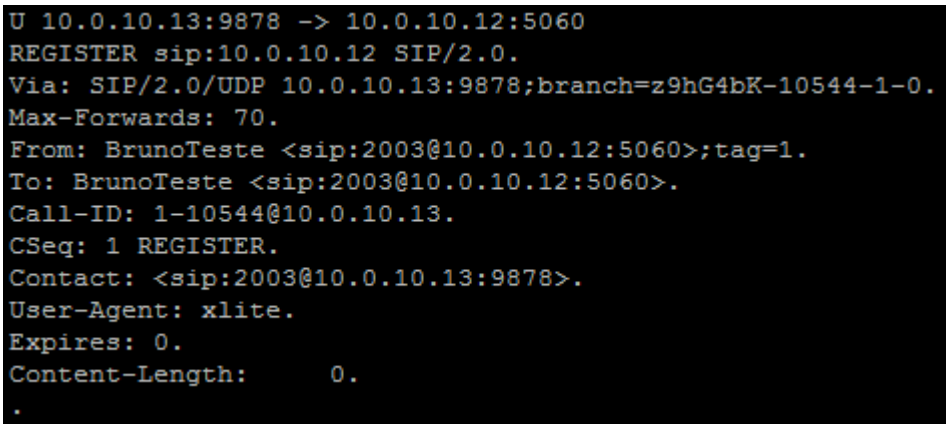

Figura 14 – Mensagem de registro

Logo após a confirmação do desregistro, o SIPp foi executado novamente para gerar um pedido de registro utilizando o cenário desenvolvido para este propósito, ele informa a nova localidade do usuário desta conta, neste caso, o IP e porta da máquina que realiza o ataque. Como o fluxo de mensagens trocadas pelo SIPp para realizar ambos os ataques é igual, o retorno das estatísticas da ferramenta ao fim de sua execução também foi igual, mudando apenas o valor do cabeçalho *Expire* da mensagem *REGISTER*, como capturado e mostrado a seguir.

```
U 10.0.10.13:9878 -> 10.0.10.12:5060
REGISTER sip:10.0.10.12 SIP/2.0.
Via: SIP/2.0/UDP 10.0.10.13:9878;branch=z9hG4bK-10506-1-0.
Max-Forwards: 70.
From: BrunoTeste <sip:2003@10.0.10.12:5060>;tag=1.
To: BrunoTeste <sip:2003@10.0.10.12:5060>.
Call-ID: 1-10506@10.0.10.13.
CSeq: 1 REGISTER.
Contact: <sip:2003@10.0.10.13:9878>.
User-Agent: xlite.
Expires: 3600.
Content-Length:     0.
.
```

Figura 15 – Mensagens de desregistro

Finalizado com sucesso o ataque de roubo de registro, o usuário ilícito tem agora todas as permissões de um usuário legítimo para utilizar os serviços oferecidos pela plataforma VoIP do estudo de caso.

3.3 ATAQUE DE *FLOOD*

O segundo ataque ao cenário montado para o estudo de caso foi explorado com a ajuda do roubo de registro. Aproveitando-se da posse de um registro ilícito na plataforma, a ferramenta siprtp foi utilizada para gerar uma grande quantidade de chamadas simultâneas ao servidor SIP, elas serão re-encaminhadas para a máquina de ataque onde existe outra instância do siprtp especificada para receber essas chamadas. A intenção deste ataque é fazer com que toda a troca de sinalização SIP e tráfego de pacotes RTP trafeguem pelo servidor, consumindo os recursos de rede e o processamento da máquina onde ele está rodando.

Na primeira execução do ataque, decidiu-se gerar 70 chamadas simultâneas que ficaram estabelecidas durante 320 segundos até sua finalização. Segue abaixo as estatísticas geradas pelo programa ao término da execução.

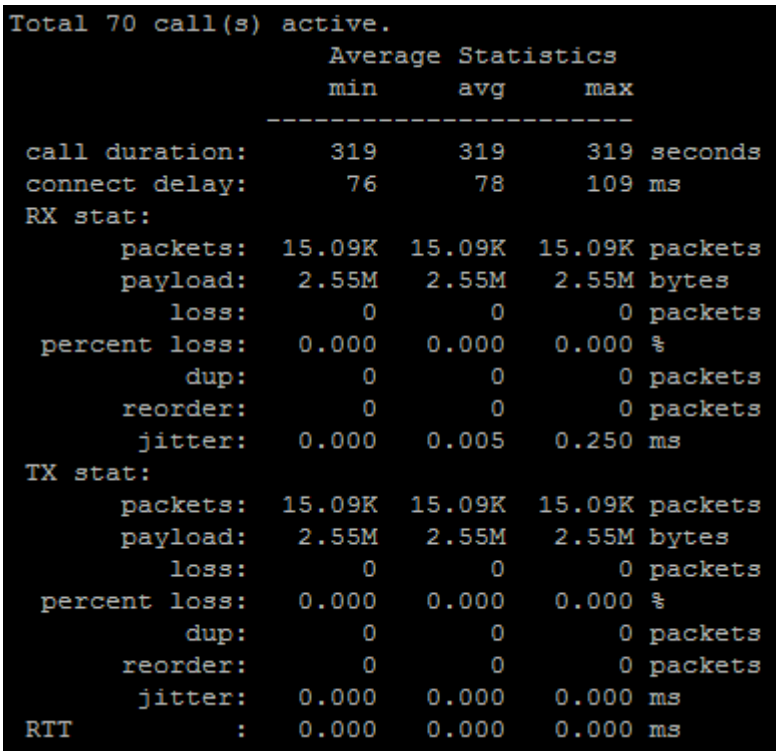

Figura 16 – Estatísticas do siprtp para o ataque de flood com 70 chamadas

Enquanto as 70 chamadas estavam estabelecidas, foi analisado o consumo de CPU da máquina onde o servidor SIP estava rodando através do comando “top”, as seguintes informações foram retornadas.

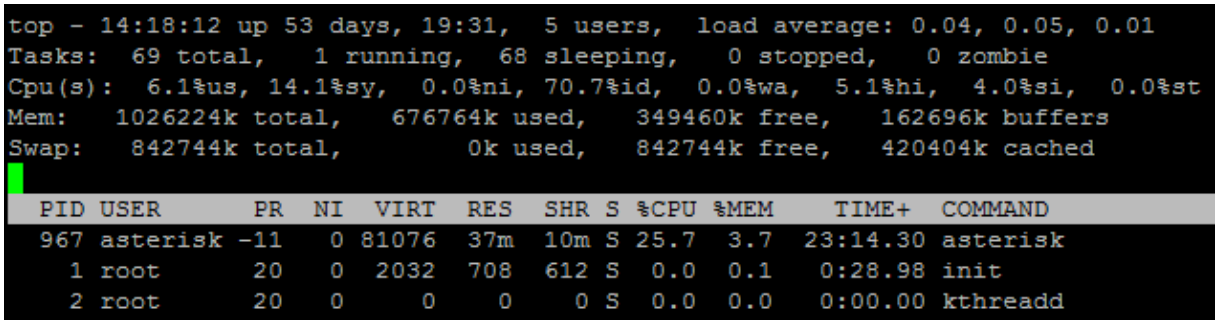

Figura 17 – Estatísticas do comando top para ataque de flood com 70 chamadas

Fazendo uma análise das estatísticas fornecidas pelo siprtp e pelo comando top, pode-se perceber que todas as 70 chamadas foram estabelecidas com sucesso, não apresentaram perda de pacotes RTP e apesar do processo *Asterisk* consumir cerca de 25% de

CPU do servidor, não afetaram as características da estrutura e o funcionamento do serviço VoIP.

Percebendo o fracasso do primeiro ataque, foi decido aumentar a quantidade de chamadas simultâneas para 150. Ao final dos 320 segundos em que elas estiveram estabelecidas, o siprtp retornou as seguintes estatísticas.

```
Total 150 call(s) active.
                    Average Statistics
                    min     avg     max
                 ----------------------
 call duration:     319     320     320 seconds
 connect delay:      17     114    1017 ms
 RX stat:
       packets:  15.09K  15.09K  16.00K packets
       payload:   2.55M   2.55M   2.56M bytes
          loss:       0       0       0 packets
  percent loss:   0.000   0.000   0.000 %
           dup:       0       0       0 packets
       reorder:       0       0       0 packets
        jitter:   0.000   0.285   2.125 ms
 TX stat:
       packets:  15.09K  15.09K  16.00K packets
       payload:   2.55M   2.56M   2.56M bytes
          loss:       0       0       0 packets
  percent loss:   0.000   0.000   0.000 %
           dup:       0       0       0 packets
       reorder:       0       0       0 packets
        jitter:   0.000   0.000   0.000 ms
 RTT           :   0.000   0.000   0.000 ms
```

Figura 18 – Estatísticas do siprtp para ataque de flood com 150 chamadas

No comando “top”, foi obtido as seguintes informações.

```
top - 05:33:14 up 54 days, 10:46,  5 users,  load average: 0.00, 0.02, 0.00
Tasks:  68 total,   1 running,  67 sleeping,   0 stopped,   0 zombie
Cpu(s):  6.9%us, 26.7%sy,  0.0%ni, 45.5%id,  0.0%wa, 12.9%hi,  7.9%si,  0.0%st
Mem:   1026224k total,   670584k used,   355640k free,   162704k buffers
Swap:   842744k total,        0k used,   842744k free,   420776k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3420 root      20   0 90608  31m  9m S 60.9  3.1   4:23.46 asterisk
    1 root      20   0  2032  708  612 S  0.0  0.1   0:29.27 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
```

Figura 19 – Estatísticas do comando top para ataque de flood com 150 chamadas

Com o final de execução do segundo ataque, nenhum efeito colateral de grande importância foi percebido na plataforma. Todas as 150 chamadas foram estabelecidas com

sucesso, o tráfego de pacotes RTP passando pelo servidor foi realizado sem perdas, e o aumento do consumo de CPU devido ao processamento da sinalização e do áudio não foi suficiente para causar algum tipo de negação de serviço.

Com mais uma tentativa sem sucesso, o terceiro ataque foi realizado com 250 chamadas simultâneas, e desta vez os resultados obtidos ao fim dos 320 segundos foram totalmente diferentes. Segue abaixo estatísticas do siprtp e do comando top.

```
Total 219 call(s) active.
                    Average Statistics
                    min     avg     max
                  ----------------------
 call duration:     289     317     320 seconds
 connect delay:      16    1963   28003 ms
 RX stat:
       packets:  11.08K  15.07K  16.00K packets
       payload:   1.89M   2.52M   2.56M bytes
          loss:       0      75    2916 packets
  percent loss:   0.000   0.474  15.414 %
           dup:       0       0       0 packets
       reorder:       0       0       0 packets
        jitter:   0.000   3.424 2209.000 ms
 TX stat:
       packets:  15.08K  15.09K  16.00K packets
       payload:   2.53M   2.55M   2.56M bytes
          loss:       0       0       0 packets
  percent loss:   0.000   0.000   0.000 %
           dup:       0       0       0 packets
       reorder:       0       0       0 packets
        jitter:   0.000   0.000   0.000 ms
 RTT           :   0.000   0.000   0.000 ms
```

Figura 20 – Estatísticas do siprtp para ataque de flood com 250 chamadas

```
top - 06:17:53 up 54 days, 11:30,  5 users,  load average: 13.08, 3.80, 1.56
Tasks:  68 total,   1 running,  67 sleeping,   0 stopped,   0 zombie
Cpu(s): 16.6%us, 39.0%sy,  0.0%ni,  0.0%id,  0.0%wa, 26.6%hi, 17.8%si,  0.0%st
Mem:   1026224k total,   681768k used,   344456k free,   162704k buffers
Swap:   842744k total,        0k used,   842744k free,   421964k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3420 root      20   0  114m  39m  10m S 99.9  3.9  19:04.48 asterisk
    1 root      20   0  2032  708  612 S  0.0  0.1   0:29.29 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
```

Figura 21 – Estatísticas do comando top para ataque de flood com 250 chamadas

A primeira grande mudança percebida foi que das 250 chamadas realizadas 219 foram estabelecidas, portanto, 31 delas não foram completadas pelo servidor SIP. Outro dado reportado foi a perda de pacotes RTP, cerca de 15% deles não chegaram ao destino final.

Além dos problemas apresentados pelo siprtp, uma importante informação foi relatada na execução do comando top, o processo *Asterisk* começou a consumir 99% da capacidade reservada para ele no processamento da CPU, esse consumo fez com que toda a estrutura do servidor ficasse comprometida. Para certificar o comportamento inadequado da plataforma, foi realizado algumas chamadas entre os usuários legítimos registrados nela, o resultado foi falhas no estabelecimento das sessões ou grande perdas e atrasos na reprodução de áudio entre as pontas.

O ataque de *flood* não chegou a derrubar o servidor SIP da plataforma, mas tornou inadequado os serviços oferecidos por ele, sendo assim, partiu-se para o último ataque realizado ao cenário do estudo de caso e descrito a seguir.

3.4 ATAQUE DE MENSAGEM MALICIOSA

O terceiro e último ataque foi executado através de mensagens maliciosas enviadas aos *user-agents* registrados na plataforma, o intuito é realizar um *reboot* dos dispositivos de comunicação com o usuário através de uma requisição *NOTIFY*. Para realizar este ataque utilizou-se mais uma vez a ferramenta SIPp e um cenário novo, desenvolvido para o envio de mensagem *NOTIFY* com o valor do cabeçalho *Event* igual a *check-sync*, se o receptor da mensagem aceitar esse evento um 200 OK é retornado e o *reboot* é realizado, caso o evento não seja suportado, uma mensagem de erro é retornada.

O primeiro dispositivo a ser atacado foi o telefone IP da Cisco, inicialmente o SIPp foi configurado para executar apenas uma vez o cenário para o destino desejado. Ao final da execução obteve-se as seguintes estatísticas da ferramenta.

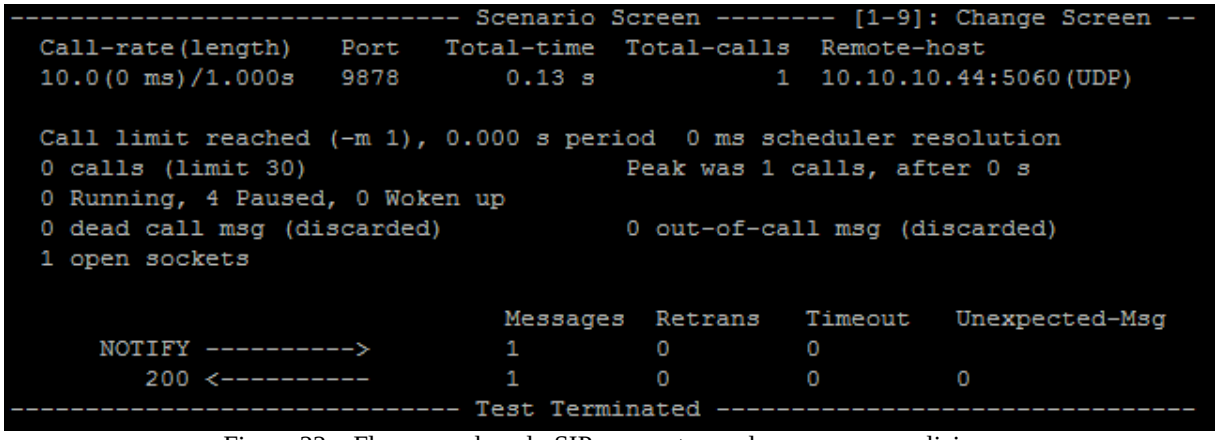

Figura 22 – Fluxo gerado pelo SIPp para ataque de mensagem maliciosa

De acordo com o resultado, é possível concluir que o ataque foi realizado com sucesso, ao receber o *NOTIFY*, o telefone IP da Cisco reconheceu o evento e realizou o *reboot* automaticamente. Segue abaixo mensagem *NOTIFY* enviada para o telefone via ferramenta SIPp.

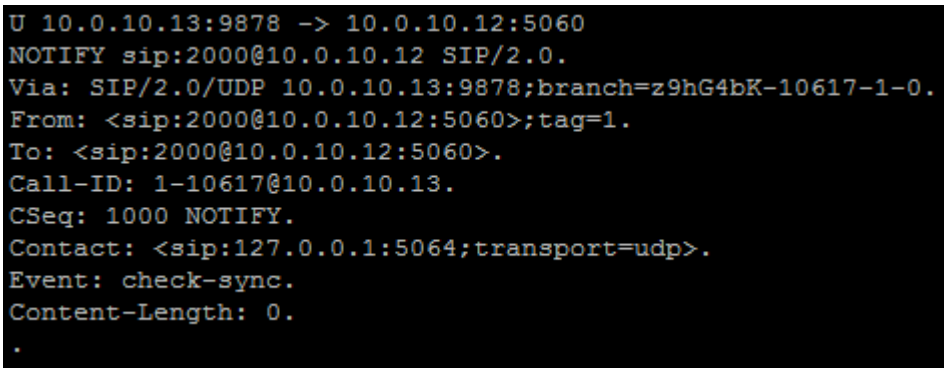

Figura 23 – Mensagem *Notify* com evento de *reboot*

O mesmo procedimento de ataque descrito acima foi realizado para o telefone IP do fabricante Snom, ao final da execução o resultado obtido também foi igual. Ao receber a mensagem de *NOTYFY* enviada pelo SIPp, o telefone retornou um 200 OK e automaticamente realizou seu *reboot*.

Afim de saber o comportamento do *softphone* X-lite e do SIP *Client* do Nokia N97, foi enviado a mensagem de *NOTIFY* para esses usuários, no momento em que receberam a requisição, ambos retornaram mensagens de erro informando que o evento não pôde ser realizado.

Como apenas os telefones IP da estrutura realizavam o *reboot* especificado na mensagem maliciosa, programou-se o SIPp para enviar o *NOTIFY* para os dois dispositivos a cada 2 minutos. Este tipo de execução da ferramenta fez com que os usuários registrados na estrutura através dos telefones IP não conseguissem utilizar o sistema, já que sua interface de comunicação com a plataforma ficava indisponível a todo momento.

Visto com sucesso a execução do último ataque, foi finalizado a parte de desenvolvimento e parti-se agora para a conclusão dos testes de vulnerabilidades do SIP, executados na plataforma montada para o estudo de caso.

CONCLUSÃO

O SIP é o protocolo mais utilizado para estabelecer a comunicação entre os elementos de uma rede VoIP, por isto, tornou-se um importante alvo para aqueles que desejam interferir no comportamento das estruturas montadas para este tipo de comunicação. O fato de ser baseado em mensagens de texto que transportam informações importantes sobre os dispositivos envolvidos nas sessões, já faz do SIP um protocolo bastante vulnerável caso nenhum método de segurança seja adotado para garantir a integridade e esconder o conteúdo que nele é transportado. Por esses motivos, a realização deste trabalho foi essencial para compreender e comprovar todo o mal causado por ataques que utilizam as vulnerabilidades do protocolo SIP.

Todo conteúdo apresentado foi importante para compreensão geral do tema. Através dos conceitos de redes de computadores foi possível planejar, definir e montar uma a estrutura que ofereceu comunicação de dados entre usuários conectados a ela. A escolha dos elementos da arquitetura, dos ataques e das ferramentas utilizadas foram avaliadas com base no que foi tratado sobre SIP e suas vulnerabilidades. Apesar de não implementar nenhum método de proteção do protocolo, os conceitos de segurança serviram para despertar o interesse em trabalhos futuros, com foco nas formas de combater as fragilidades expostas e testadas neste trabalho.

Por fim, os resultados obtidos com a realização do estudo de caso confirmam que os objetivos de expor, simular e analisar algumas das vulnerabilidades do protocolo SIP foram alcançados. Com um simples ataque de roubo de registro, todos os atributos delegados a um usuário legítimo foram transferidos para um usuário malicioso, que pôde usufruir livremente dos recursos VoIP disponibilizados na plataforma. O ataque de *flood* mostrou que com 250 chamadas simultâneas é possível comprometer todo o processamento de sinalização SIP e pacotes RTP destinados ao servidor SIP, o alto consumo de CPU requerido para estabelecer essa quantidade de requisições comprometeu drasticamente o desempenho da plataforma, apresentando danos de negação de serviço e reprodução de áudio nas pontas.

A execução do último ataque possibilitou que uma simples mensagem de notificação SIP realizasse um *reboot* automático nos dois Telefones IP registrados no servidor. A forma como foi executada a ferramenta de ataque agravou ainda mais o nível de danos do ataque, o *reboot* era realizado a cada dois minutos e impossibilitou o usuário de utilizar seu equipamento VoIP. Essas características mostram a importância de se preocupar

não apenas com a segurança dos componentes centrais desta arquitetura, mas também com os dispositivos que fazem a interface de comunicação com o usuário.

REFERÊNCIAS

COLCHER, Sérgio et al. **VOIP, Voz sobre IP**. Rio de Janeiro, RJ: Campus, 2005.

ENDLER, David e COLLIER, Mark. **Hacking Exposed VoIP**: Voice Over IP Security Secrets & Solutions. USA: McGraw-Hill, 2007.

GORALSKI, Walter J. e KOLON, Mattew C. **IP Telephony**. USA: McGraw-Hill, 2000.

HERSENT, Oliver, GUIDE, David e PETITI, Jean-Pierre. **Telefonia IP**: Comunicação multimídia baseada em pacotes. São Paulo, SP: Addison-Wesley, 2002.

REZENDE, Anderson Rodrigues Souza. **Análise das Vulnerabilidades e Ataques ao Protocolo SIP,** 2006. Trabalho de conclusão de curso – Sistemas de Informação, UNIMINAS, Uberlândia – MG

SISALEM, Dorgham et al. **SIP Security**. Reino Unido: Wiley, 2009.

TANENBAUM, Andrew S. **Redes de Computadores, Quarta Edição**. Rio de Janeiro, RJ: Campus, 2003.

PALMA, Luciano e PRATES, Rubens. **TCP/IP**: Guia de Consulta Rápida. São Paulo, SP: Novatec, 2000.

ED, Tittel. **Redes de Computadores**. Porto Alegre, RS: Bookman, 2002.

HUNT, Craig. **TCP/IP Network Administration, Third Edition**. USA: O'Reilly, 2002.

THERMOS, Peter e TAKANEM Ari. **VOIP Network**: Theats, Vulnerabilities and Countermeasures. Boston, USA: Addison-Wesley, 2008

O Modelo OSI, José Maurício Santos Pinheiro, Novembro de 2004, <http://www.projetoderedes.com.br/artigos/artigo_modelo_osi.php> Acesso em: 12 out. 2011.

Criptografia, Chaves Públicas e Assinatura Digital para Leigos, Joel Guilherme, <http://www.sbis.org.br/Criptografia.doc> Acesso em: 12 out. 2011.

<http://www.ietf.org/rfc/rfc3067.txt> Acesso em: 28 out. 2011.

<http://tools.ietf.org/rfc/rfc3665.txt> Acesso em: 01 dez 2011.