Atraindo, conhecendo e repudiando atacantes VoIP

Daniel Bauermann[1,2], **Giovani Luís Emmert**[1]

[1]Faculdade IENH (IENH)
Rua Frederico Mentz, 526 – 93.525-360S – Novo Hamburgo – RS – Brasil

daniel.b@ienh.com.br, giovaniemm@hotmail.com

[2]Universidade Feevale
ERS-239, 2755 – 93.525-075 – Novo Hamburgo – RS – Brasil

danielbauermann@feevale.br

Abstract. *The expansion of the use of voice over IP systems has attracted a growing number of people who try to find vulnerabilities that may cause harm to organizations. To understand how these people exploit the systems, it is crucial know their way of acting. For this, different works uses honeypot technology with the objective of investigating the caracteristics of many kinds of possible attacks. However, few works present a system that can repulse attackers interested in exploiting vulnerable systems, especially for VoIP services. In this work, we propose a framework capable of compiling attack data to increase the security of an IP telephony network.*

Resumo. *A expansão do uso de sistemas de voz sobre IP tem atraído um número cada vez maior de indivíduos que buscam vulnerabilidades que podem causar prejuízos às organizações. Para entender de que forma estes indivíduos exploram os sistemas, é fundamental conhecer seu modo de agir. Para isto, diferentes trabalhos fazem uso da tecnologia de honeypots com o objetivo de investigar as características dos diversos ataques possíveis. Entretanto, poucos trabalhos apresentam um sistema capaz de repelir malfeitores interessados em explorar sistemas vulneráveis, principalmente para serviços VoIP. Neste trabalho propomos um arcabouço capaz compilar dados de ataques buscando incrementar a segurança de uma rede de telefonia IP.*

1. Introdução

O uso de sistemas de telefonia Voice over Internet Protocol (VoIP) há muito deixou de ser uma promessa e uma tendência de crescimento para se tornar uma realidade para muitas empresas e organizações. A popularização destes sistemas tem atraído atacantes interessados em usufruir de sistemas vulneráveis ou expostos. A evidência deste crescimento bem como do interesse neste tipo de sistemas foi observada pelo CERT.br[1], originando um estudo de aprofundamento do interesse destes atacantes [Steding-Jessen 2016], no qual se registrou um elevado número de abusos em servidores Session Initiation Protocol (SIP).

[1]O CERT.br é o Grupo de Resposta a Incidentes de Segurança para a Internet brasileira, mantido pelo NIC.br, do Comitê Gestor da Internet no Brasil. É responsável por tratar incidentes de segurança em computadores que envolvam redes conectadas à Internet brasileira. Fonte: http://www.cert.br/sobre/

Este cenário tem motivado diversos pesquisadores a investigar as fragilidades e possíveis proteções para sistemas de telefonia VoIP, gerando obras extensas sobre o assunto, como por exemplo, [Endler e Collier 2007]. Em Nassar [Nassar et al. 2011] podemos encontrar diferentes possibilidades de ataques, além disso, o autor destaca que tais possibilidades exigem diferentes tratamentos de defesa.

Dentre as tecnologias para aprofundamento do conhecimento dos métodos e das técnicas de um atacante, o uso de *honeypots* e *honeynets* tem se apresentado como uma alternativa bastante eficaz, uma vez que simulam ambientes reais e armazenam diversos dados sobre os atacantes, permitindo um estudo detalhado de um ambiente atacado.

O conhecimento aprofundado sobre ataques SIP permite criar medidas para mitigar ataques. Porém, quando existe a necessidade de expor sistemas de telefonia VoIP em uma rede pública, ainda que se implementem tais medidas, os riscos aumentam, já que a central de telefonia se torna exposta a todos. Neste cenário, o uso de *blacklists* com endereços de atacantes é uma técnica que aumenta significativamente a segurança de um sistema de telefonia. No entanto, a atualização de seus registros bem como seus critérios de inclusão nem sempre são claros. Além disso, não se tem conhecimento de listas que representem o cenário nacional.

Diferente de outros trabalhos nos quais o uso de *honeynets* é um meio para levantar dados de atacantes e analisar seu comportamento, no presente trabalho, o uso desta técnica é o meio para criar uma *blacklist* em tempo real para proteger um sistema de telefonia exposto em uma rede pública.

Assim, o objetivo do presente trabalho é desenvolver uma estrutura de proteção para um sistema de telefonia VoIP, fazendo uso de dados coletados através de uma *honeynet*, a fim de incrementar o nível de segurança de perímetro existente em um ambiente real de telefonia.

O restante deste trabalho está organizado como segue. A seção 2 apresenta um visão geral de trabalhos relacionados. Uma breve descrição de VoIP, bem como os ataques possíveis ao sistema são apresentados na seção 3. Na seção 4 descreve o cenário de implementação deste trabalho e descreve a arquitetura criada no decorrer deste estudo para incrementar o nível de segurança da rede onde este trabalho foi desenvolvido. Os resultados encontrados e análises são detalhados na seção 5. A seção 6 encerra com conclusões e perspectivas de trabalhos futuros.

2. Trabalhos relacionados

A questão de segurança em VoIP tem estimulado muitas investigações sobre o assunto, gerando um série de trabalhos que classificam os ataques a sistemas de telefonia conforme seus objetivos ([Thermos e Takanen 2008] dedica um capítulo inteiro sobre o assunto). O trabalho de [Hoffstadt et al. 2012] serviu de base para classificação de ataques SIP usados na presente pesquisa.

Apesar destas e outras tantas fontes contribuírem para o entendimento e classificação de agressores a sistemas VoIP, para Gruber [Gruber et al. 2011] existia a necessidade de entender o comportamento dos atacantes, buscando padrões e assinaturas de ataques. Isto gerou um extenso trabalho investigativo [Gruber et al. 2015] no qual se validou comportamentos semelhantes, mesmo em redes diferentes.

Ainda que Gruber [Gruber et al. 2015] considere que o comportamento de atacantes em diferentes redes seja semelhante, buscou-se para esta pesquisa dados que apontassem o comportamento dos ataques em redes nacionais. Embora o trabalho desenvolvido pelo CERT.br no projeto honeyTARG [CERT.br 2016] apresente dados interessantes sobre o ambiente nacional, não existem dados específicos sobre ataques VoIP em sua coleção. Através da análise de seus dados relativos às tentativas de exploração de portas User Datagram Protocol (UDP), fica explícito que a porta 5060 (padrão do protocolo SIP) lidera as tentativas dos atacantes que objetivam serviços executando em UDP. No entanto, não existem outros detalhes sobre estes atacantes disponibilizados de forma pública. Assim sendo, o presente trabalho faz uso de *honeynets* para coletar tais informações dos agressores.

Assim como em [Nassar et al. 2011] e [Gruber et al. 2015], Carmo [Carmo 2011] também faz uso de *honeynets* para coletar informações de atacantes. Diferentemente destes e outros trabalhos cujo objetivo é compreender padrões de ataques, o presente trabalho tem como objetivo repelir atacantes que tenham intencionado explorar a rede na qual a *honeynet* implementada foi instalada.

Para o controle de *spammers* é comum a adoção de *blacklists* que impendem o recebimento de conteúdo indesejado oriundo de fontes duvidosas. A consulta de uma lista pública onde constam registrados alguns destes endereços Internet Protocols (IPs) ajuda a minimizar o recebimento deste tipo de conteúdo. Há muito tempo é possível consultar diferentes fontes que mantêm este tipo de lista com apoio da comunidade da internet, como por exemplo, [Spa 2016] e [bar 2016], entre outras.

Ainda que existam *blacklists* específicas para VoIP, que auxiliam na construção de uma proteção para centrais telefônicas expostas em redes públicas, seus registros nem sempre atendem aos objetivos desta pesquisa. [Sco 2016], por exemplo, oferece uma lista aberta de IPs que podem ser usados para bloqueio. Mas sua lista não possuía nenhum registro do Brasil até a conclusão deste trabalho, ao contrário do que encontramos em nosso cenário desenvolvido. Já a lista disponibilizada em [Net 2016] oferece abertamente apenas um número limitado de registros, sendo que a lista completa é disponibilizada apenas para clientes da empresa mantenedora da lista. Além disso, não constam detalhamentos dos métodos utilizados para criação desta lista.

Alinhado com os objetivos de Nassar [Nassar et al. 2011], o presente trabalho também se buscou criar estratégias de defesa sem complicação e sem isolamento, a fim de defender um sistema de telefonia IP exposto em uma rede pública.

3. Voice over Internet Protocol

Assim como na telefonia tradicional, também na telefonia digital, ao se tentar conectar com alguém, é necessário ser informado se a pessoa de destino está com o telefone ocupado ou se está sendo chamado. Da mesma forma, a pessoa que está recebendo a chamada deve ser notificada. Este procedimento é conhecido como sinalização [Callado et al. 2007].

Existem diferentes protocolos para o processo de sinalização, sendo os mais comuns: H.323, SIP, Inter-Asterisk eXchange v2 (IAX2), Skinny Client Control Protocol (SCCP), Media Gateway Control Protocol (MGCP) e H.248 ou MEGACO. Ainda que o

H.323 tenha sido um dos primeiros protocolos apresentado e utilizado pela indústria, sua complexidade deu espaço para o avanço do protocolo SIP, sendo este um dos mais populares. Assim sendo, na próxima seção é apresentado um detalhamento deste protocolo.

3.1. Session Initiation Protocol

O SIP é um protocolo da camada de aplicação baseado em texto (semelhante ao protocolo Hypertext Transfer Protocol (HTTP)) desenvolvido para estabelecer, manter e terminar ligações telefônicas, videoconferência e outras aplicações multimídia [Callado et al. 2007]. A arquitetura SIP é formada por dois componentes, o agente de usuário e o servidor de rede.

O agente de usuário é responsável por iniciar e terminar as requisições SIP. Já o servidor de rede, nesta arquitetura, se divide em outros três tipos. O (1) servidor de registro, responsável por receber as requisições dos agentes de usuários e manter um banco de dados com informações dos usuários. O (2) servidor *proxy*, responsável por receber pedidos e atendê-los diretamente ou encaminhá-los para outro servidor, trazendo mais desempenho no processo de encaminhamento. E o (3) servidor de redirecionamento, responsável por receber requisições e informar ao cliente o endereço do próximo servidor. Assim, o cliente contata diretamente o próximo servidor [Ghafarian et al. 2016]. No contexto deste trabalho são utilizados apenas servidores de registro.

Os agentes se comunicam com servidores através de transações (ou pedidos). Cada transação gera uma ou mais respostas. A figura 1 ilustra a troca de mensagens em uma comunicação SIP, já considerando que os agentes dos usuários Marcos e João estejam devidamente registrados em um servidor de registro. Marcos inicia a comunicação enviando uma mensagem de requisição para o servidor SIP, solicitando contato com João. O servidor verifica em sua base se João existe e, existindo, avisa a Marcos que é possível comunicar-se com seu destino, informando-lhe o caminho até João. Após estabelecida a sessão, há o tráfego de pacotes de voz (sessão de mídia), que são transportados sob o protocolo Realtime Transport Protocol (RTP). Por fim, a comunicação se encerra através das mensagens representadas pela transação 3 na figura.

Conforme a RFC 3261 [RFC 2016] existem seis tipos de mensagens SIP: `INVITE`, `ACK`, `BYE`, `CANCEL`, `REGISTER` e `OPTIONS`. As mensagens ainda são classificadas em mensagens de requisição e de resposta. Uma transação é formada por, pelo menos, uma mensagem de requisição e uma mensagem de resposta.

3.2. Ataques SIP

Hoffstadt [Hoffstadt et al. 2012], através de experimentos com *honeypots*, caracterizou ataques SIPs em quatro estágios distintos, todos executados através de mensagens do protocolo SIP, conforme descritos a seguir:

3.2.1. Varredura de servidores e dispositivos SIP

Assim como em outros tipos de ataques, a primeira fase é a descoberta de alvos. Isto é feito através de varreduras de redes, buscando informações e portas abertas.

O fato de um dispositivo SIP responder a pacotes `OPTIONS` permite ao atacante usar um "`ping`" para um único IP, ou até mesmo para uma rede inteira, com pacotes

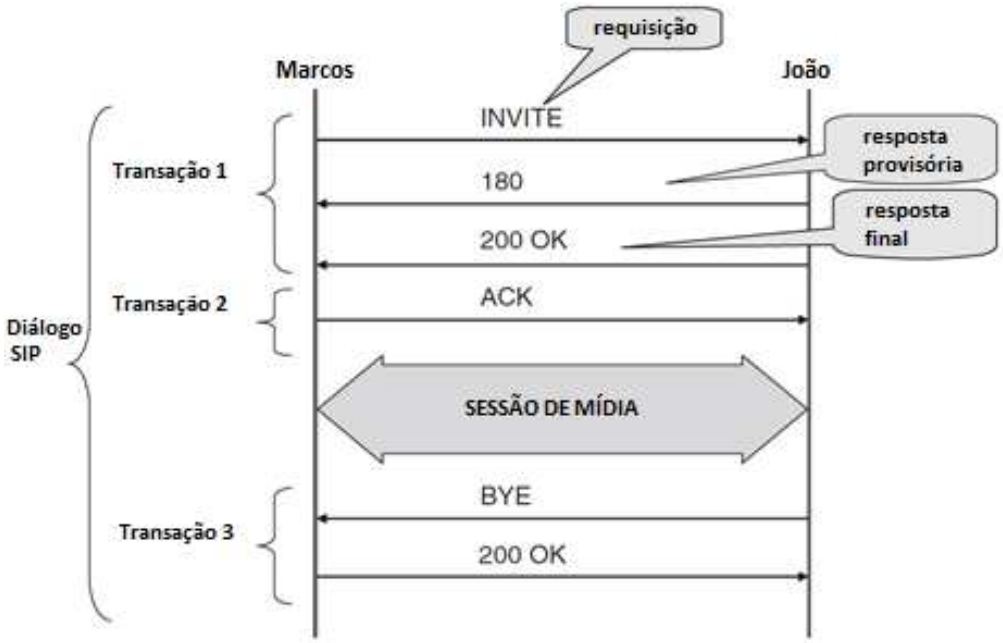


Figura 1. Fluxo de comunicação SIP

`OPTIONS` buscando identificar dispositivos SIP. Em caso de agentes que não implementem o protocolo SIP padrão e só respondam pacotes `OPTIONS` de fontes conhecidas, ainda assim uma varredura é possível. Neste caso, o atacante pode utilizar solicitações do tipo `REGISTER`.

3.2.2. Varredura de ramais

Uma vez identificado um alvo, o atacante passa a executar uma varredura mais aprofundada, buscando ramais no servidor. Isto é feito através de tentativas sistemáticas de registros de ramais sem senhas. Um ramal consiste em uma sequência de dígitos e/ou série de letras. Se um ramal existir, o servidor normalmente responde com uma mensagem `401 UNAUTHORIZED` (não autorizado) ou `403 FORBIDDEN` (senha errada). Se o ramal não existir, uma mensagem `404 NOT FOUND` (não encontrado) é retornada. O resultado desta fase é uma lista completa de ramais existentes.

3.2.3. Registro de ramais

Para registrar um ramal, o atacante tenta adivinhar a senha. Para isto são enviadas múltiplas mensagens `REGISTER` com diferentes senhas para um determinado ramal. Se uma senha é adivinhada, a informação é salva para um posterior registro.

3.2.4. Fraude

A fraude acontece sempre que uma pessoa gerar custos usando um ramal ”sequestrado”. O atacante pode, por exemplo, fazer chamadas internacionais, usando as funcionalidades do VoIP. Outro benefício de um ramal sequestrado está no fato do atacante permanecer anônimo.

Em termos de mensagens SIP, o atacante primeiro envia uma mensagem `REGISTER` com a senha correta. Após a mensagem `200 OK` do servidor, o invasor pode iniciar chamadas usando mensagens `INVITE`.

4. Implementação

Esta seção descreve como foi implementada a *honeynet* usada para coleta de dados de atacantes que alvejavam a rede em que este trabalho foi exposto. Na sequência é descrito o arcabouço desenvolvido para compilar o conjunto de dados coletados e banir IPs mal intencionados que tentaram explorar os serviços propositalmente expostos.

4.1. *Honeynet*

O software de código aberto utilizado neste trabalho foi o Dionaea[2], um *honeypot* de baixa interação [Weissheimer Júnior 2008] com capacidade de emular serviços de telefonia VoIP, além de possuir módulos para outros serviços.

Foram utilizados dois servidores virtuais rodando Ubuntu Linux 12.04 LTS 32 *bits*[3] nos quais o Dionaea foi instalado. Cada servidor possui 4 GB de memória RAM, um processador de 2 núcleos com *clock* de 1,2 GHz e 100 GB de espaço em disco.

Ainda que o uso de *honeypots* se apresente como uma alternativa excelente para coletar dados e formas de ataque, conforme [Safarik et al. 2012], um atacante tende a perder o interesse por um alvo ao perceber que está atacando um ambiente simulado. Por isto, diferentes trabalhos fazem uso de *honeynets* a fim de maximizar a atração de malfeitores para sua rede.

Assim sendo, objetivando maximizar a atração de malfeitores à rede na qual este trabalho foi desenvolvido, cada *honeypot* foi instalado em um segmento de rede, sendo o primeiro servidor disponibilizado com um IP real da faixa 200.x.y.z e o segundo na faixa 177.x.y.z.

4.2. Arquitetura do *Honey To Block*

Uma vez que a implementação da *honeynet* foi superada, se fez necessário o desenvolvimento de um sistema capaz de concentrar os dados de forma única. Para isto, foi criado em Python[4] um software denominado *Honey To Block*, com o objetivo de receber os dados dos *honeypots*, agrupá-los de forma única e disponibilizá-los para a consulta de um *firewall*. A figura 2 apresenta a arquitetura criada.

O `H2B-server` é o servidor responsável por atender as requisições dos demais componentes deste arcabouço. Atualmente seu papel se concentra em duas atividades

[2]https://github.com/rep/dionaea
[3]https://www.ubuntu.com/
[4]https://www.python.org/

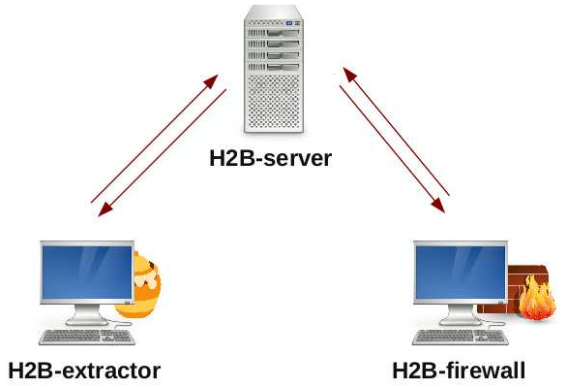


Figura 2. Arquitetura Honey to Block

principais, que são: (1) receber a lista de novos atacantes coletadas em cada *honeypot*, organizando-os de forma única; e (2) informar ao *firewall* quais são os novos atacantes registrados. Estas interações consideram o tempo entre um contato e outro a fim de minimizar o volume de tráfego entre clientes e servidor.

Já o `H2B-extactor` tem o papel de extrair o endereço IP dos atacantes que tentaram explorar o módulo VoIP do *honeypot*, informando-os ao servidor. Isto é feito através da análise dos *logs* do Dionaea em ciclos de tempo. Entre cada execução são extraídos somente os *logs* registrados naquele intervalo de tempo. Assim, podem existir registros duplicados de IPs, como o caso de um atacante que está tentando fazer uma nova exploração. Caso isto ocorra, o servidor será o responsável por manter a unicidade destes registro, embora mantenha também armazenada a informação de todas as notificações do *honeypot*, permitindo análises futuras.

Por fim, o `H2B-firewall` é o cliente que está rodando em uma estrutura de *firewall* com a responsabilidade de proteger um sistema de telefonia IP. Ele tem o papel de consultar o servidor (`H2B-server`) para buscar uma lista de quais atacantes serão banidos da rede, evitando qualquer tentativa de interação destes com o sistema de telefonia. Esta consulta ocorre em períodos de tempo determinados e, também, busca somente os novos malfeitores registrados desde a última consulta. Uma vez obtida a lista de quais atacantes deverão ser banidos, o `H2B-firewall` aplica as regras sobre o filtro de pacotes em execução, carregando somente as regras necessárias para tal negação, não alterando as demais regras criadas no ambiente.

5. Resultados

Desde que foram implementados os *honeypots*, juntos, já somam 15.114 horas de execução, sendo 8.193 horas no primeiro *honeypot* e 6.921 horas no segundo. Neste período foram registrados 2.121.660 ataques em diferentes serviços. Especificamente para o serviço de VoIP foram contabilizados 308.631 tentativas de exploração do serviço, até o fechamento deste artigo.

Para validar os dados coletados em nossos *honeypots*, utilizamos as estatísticas [CERT.br 2017] disponibilizadas pelo projeto honeyTARG [CERT.br 2016], coordenado pelo CERT.br. Estas estatísticas são disponibilizadas diariamente, sendo os dados apresentados neste trabalho referente ao dia 09/07/2017.

Na tabela 1 podemos observar que o principal protocolo atacado nos serviços disponibilizados pelo honeyTARG é o protocolo Transmission Control Protocol (TCP). A diferença entre o primeiro protocolo (TCP) e o segundo (UDP), em volume de dados trafegados, é significativa. O mesmo comportamento foi encontrado em nossos *honeypots*.

Tabela 1. Protocolos atacados em 09/05/2017 no honeyTARG.

#	Protocol	Total		Max	Avg
01	TCP	1.23 GB	85.45 %	108.63 KB/s	14.28 KB/s
02	UDP	209.20 MB	14.49 %	18.35 KB/s	2.42 KB/s
03	ICMP	929.58 KB	0.06 %	121.49 B/s	10.76 B/s
04	Others	0.00 B	0.00 %	0.00 B/s	0.00 B/s

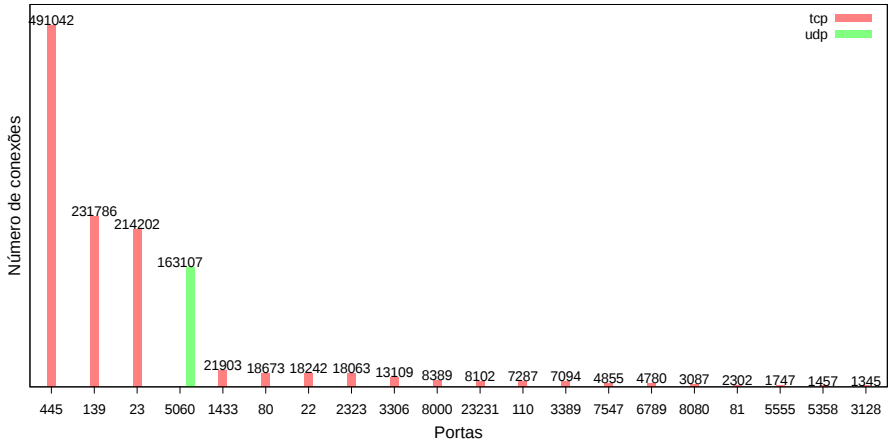
A figura 3 apresenta o número de requisições registradas nos *honeypots* expostos em nosso trabalho. Cada gráfico nesta figura representa o comportamento coletado em um *honeypot* exposto. No eixo *x* do gráfico são apresentados os números das portas atacadas, sendo portas TCP representadas em vermelho e portas UDP representadas em verde. O eixo *y* representa o número total de conexões recebidas em cada porta. Para facilitar o entendimento, o número de conexões por porta é apresentado no topo de cada barra. São apresentadas apenas as 20 primeiras portas atacadas, ordenadas de forma decrescente, para facilitar a visualização.

Observa-se que as portas TCP atacadas nos alvos expostos no presente trabalho são as mesmas encontradas nas estatísticas do projeto honeyTARG [CERT.br 2017] (exceto a porta 110 que não possui registros no *honeypot 2*, dentre as 20 primeiras portas apresentadas). Ainda que a distribuição do volume de ataques destas portas dentre nossos *honeypots* divirjam do projeto honeyTARG, pode-se perceber que os serviços buscados pelos atacantes são os mesmos. O volume e distribuição de ataques varia conforme dias analisados no honeyTARG.

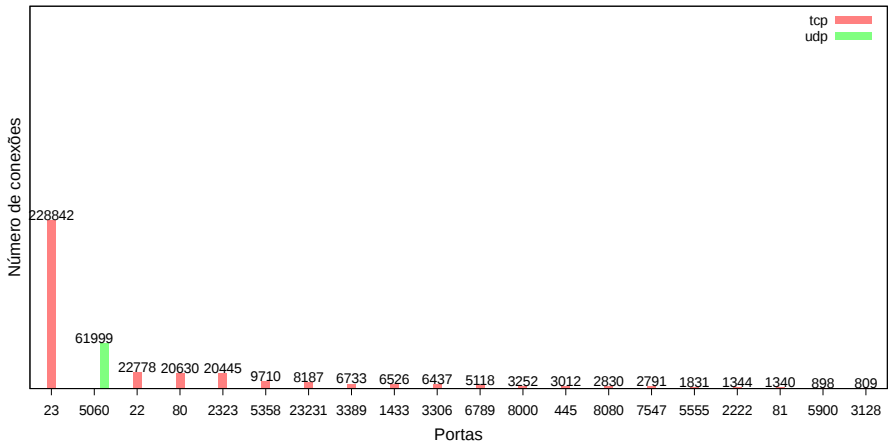
Já para as portas UDP, encontramos registros significativos em nossos *honeypots* apenas para a porta 5060, sendo inexpressivos os números de ataques registrados nas demais portas UDP de nossos alvos, em relação ao total de ataques registrados. Este comportamento é o mesmo observado no dia 09/07/2017 das estatísticas do projeto honeyTARG, onde o tráfego SIP representa quase 98% do total de tráfego UDP. Considerando que o objetivo do `H2B-extactor` é coletar especificamente os dados dos ataques VoIP, os demais serviços UDP eventualmente explorados pelos atacantes não são significativos em nosso contexto.

O segundo *honeypot* acumula um número menor de ataques em relação ao primeiro, conforme podemos observar na figura 3. Pode-se observar que os serviços atacados nos dois alvos expostos são um pouco diferentes. Enquanto que a porta 5060 UDP é o segundo serviço mais atacado no *honeypot* 2, demonstrado na figura 3(b), este mesmo serviço é apenas o quarto mais desejado no *honeypot* 1, conforme a figura 3(a).

A figura 4 apresenta em cada um de seus eixos o total acumulados de IPs por número de ataques. Na primeira barra da esquerda, por exemplo, podemos observar que 1.566 endereços IPs diferentes geraram um único ataque em nossos *honeypots*. No outro extremo do figura, o eixo da direita representa um único atacante com 176 tentativas catalogadas em nosso banco de dados.



(a) Honeypot 1



(b) Honeypot 2

Figura 3. Número de ataques por portas.

Destacamos a representatividade de atacantes com um baixo número de entradas em nossas observações. De um total de 3.109 endereços IPs coletados, aproximadamente 84%, ou 2.624 endereços, fizeram até 5 ataques apenas. Este baixo número de tentativas de ataques não despertaria suspeita sobre estes atacantes e os mesmos possivelmente não seriam denunciados para uma *blacklist*, por exemplo.

Para comprovar a eficácia dos bloqueios de atacantes registrados na *honeynet* e barrados em nosso servidor *firewall* que protege nosso ambiente de produção real, geramos *logs* para cada tentativa de exploração oriunda de algum dos endereços IPs catalogados previamente.

O volume destes *logs* é bastante grande, pois além de nossos registros, são gravadas outras informações características do ambiente e necessárias para o dia a dia das operações dos serviços que servem como barreira à rede de produção. Por isto, não é mantido um histórico extenso destes registros. Analisamos dados provenientes de 28 dias de proteção, compreendidos entre os dias 29/01/2017 e 26/02/2017.

Neste período, de um total de 343.173 ataques de IP catalogados pelo servidor de telefonia em produção e banidos com o apoio das informações da *honeynet*, 76.319

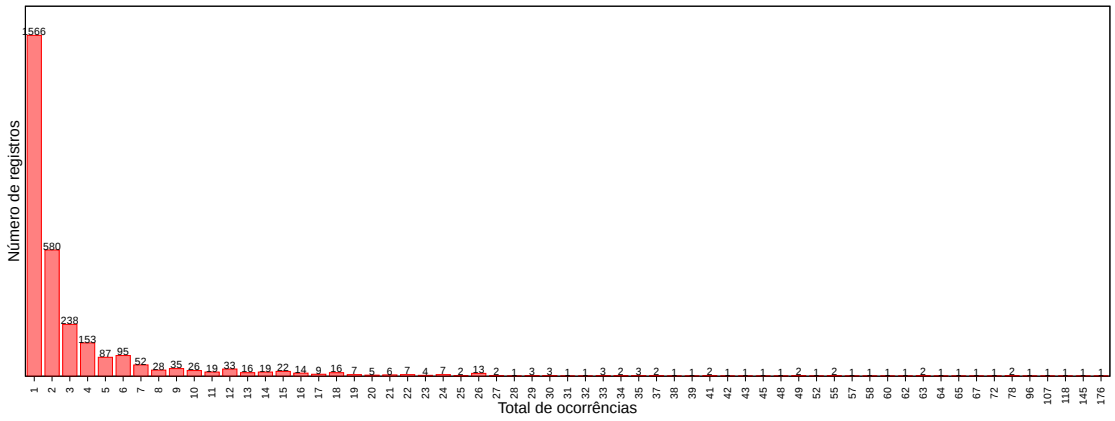


Figura 4. Quantidade de IPs por número de ataques

registros tentaram explorar especificamente a porta 5060 disponibilizada em nosso ambiente de produção, ou seja, cerca 22% dos ataques tinham como objetivo a exploração dos serviços de VoIP. Vale ressaltar que embora nosso objetivo tenha sido proteger especificamente o serviço de telefonia, com o banimento destes atacantes em nossa rede, conseguimos ainda proteger outros serviços que os atacantes buscavam, ou seja, 267.034 tentativas de ataques não eram direcionadas especificamente para VoIP.

Destacamos na figura 5 os 20 IPs com maior número de ataques bloqueados. A primeira barra da esquerda apresenta o IP 51.15.143.238 com 67.485 tentativas de ataque no período analisado. Chama a atenção que os 5 IPs com maior número de tentativas, juntos, somam 305.066, ou seja, quase 89% dos ataques barrados;

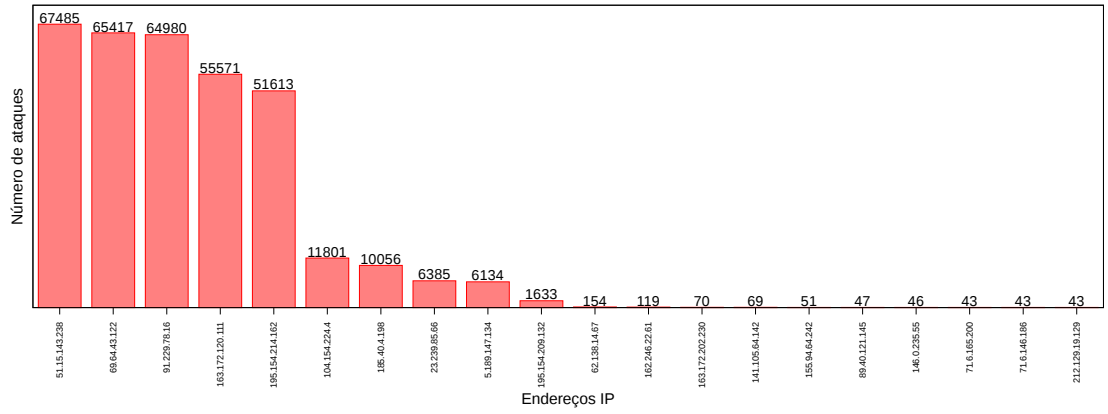


Figura 5. Quantidade de ataques barrados por endereços IPs

No período que os *logs* foram analisados, contávamos com 2.425 IPs únicos em nossa base de dados. Estes atacantes tentaram explorar os serviços de VoIP, ”ofertados” em nossos *honeypots*, desde o início de nosso projeto. Desta base, no período de 28 dias analisados nos *logs*, 204 endereços únicos foram barrados, graças à coleta e filtragem prévia destes atacantes em nossa *honeynet*. Ou seja, 8% dos atacantes que tentaram explorar os serviços VoIP dos *honeypots*, também tentaram explorar os serviços de um servidor de telefonia de produção, expostos na Internet. Ainda assim, 343.173 tentativas de ataques que foram bloqueadas.

6. Conclusões e trabalhos futuros

A popularidade do serviço de telefonia IP tem atraído atacantes que buscam explorar os recursos disponibilizados por esta tecnologia em benefício próprio, causando prejuízos para as organizações que proveem este serviço.

Diferentes pesquisas tem sido desenvolvidas para criar mecanismos de defesas contra atacantes de VoIP. Dentre estes mecanismos, o uso de *honeypots* e *honeynets* tem se apresentando como uma alternativa para conhecer detalhes dos atacantes. Neste trabalho utilizamos os dados coletados de *honeypots* para incrementar o nível de proteção de uma estrutura de *firewall* que protege um sistema de telefonia exposto de forma pública.

Através da implementação de um arcabouço de proteção, conseguimos atrair, conhecer e, posteriormente, repudiar atacantes que tentaram explorar serviços VoIP em nosso ambiente. O registro e a análise de *logs* mostrou-nos a eficiência do mecanismo de proteção criado.

Embora nosso objetivo de proteger o sistema de telefonia exposto publicamente tenha sido plenamente atingido, surpreendeu-nos descobrir que o repúdio dos possíveis atacantes VoIP de nossa rede também contribuiu para a proteção de outros serviços pelos quais os atacantes também buscavam explorar. Por isto, podemos afirmar, que tivemos benefícios adicionais em nossos mecanismos de proteção de rede.

Como trabalhos futuros, pretendemos investigar o motivo do baixo número de ataques por IP registrado em nossos servidores. Será perda de interesse como apontado por [Safarik et al. 2012] ou será o nível de interatividade do *honeypots* expostos?

Referências

(2016). Barracuda reputation block list (brbl). http://www.barracudacentral.org/rbl.

(2016). Protect you asterisk voip server from hackers and voip hijackers. http://www.networksystemssolutions.eu/voipblocklist.php.

(2016). Sip: Session initiation protocol. https://www.ietf.org/rfc/rfc3261.txt.

(2016). The spamhaus project. https://www.spamhaus.org/.

(2016). Voip blacklist. http://www.voipbl.org/.

Callado, A., Fernandes, G., Silva, A., Barbosa, R., Sadok, D., e Kelner, J. (2007). Construção de redes de voz sobre ip. 25^o *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 11–58.

Carmo, Rodrigo do e Nassar, M. e. F. O. (2011). Artemisa: an open-source honeypot back-end to support security in voip domains. *IFIP/IEEE International Symposium on Integrated Network Management*, páginas 361–368.

CERT.br (2016). Projeto honeypots distribuídos. http://honeytarg.cert.br/honeypots/index-po.html.

CERT.br (2017). Distributed honeypots project. https://honeytarg.cert.br/stats/flows/current/.

Endler, D. e Collier, M. (2007). *Hacking Exposed VoIP*. McGraw-Hill, New York.

Ghafarian, A., Seno, S. A. H., e Dehghani, M. (2016). An empirical study of security of voip system. *SAI Computing Conference*, páginas 1031–1036.

Gruber, M., Fankhauser, F., Taber, S., Schanes, C., e Grechenig, T. (2011). Security status of voip based on the observation of real-world attacks on a honeynet. *The Third IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT)*, páginas 1041–1047.

Gruber, M., Hoffstadt, D., Aziz, A., Fankhauser, F., Schanes, C., Rathgeb, E., e Grechenig, T. (2015). Global voip security threats - large scale validation based on independent honeynets. *IFIP Networking Conference (IFIP Networking), 2015*, páginas 1–9.

Hoffstadt, D., Marold, A., e Rathgeb, E. (2012). Analysis of sip-based threats using a voip honeynet system. *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, páginas 541–548.

Nassar, M., State, R., e Festor, O. (2011). Voip honeypot architecture. *2011 10th IFIP/IEEE International Symposium on Integrated Network Management*, páginas 109–118.

Safarik, J., Voznak, M., Rezac, F., e Macura, L. (2012). Malicious traffic monitoring and its evaluation in voip infrastructure. *Telecommunications and Signal Processing*, páginas 259–262.

Steding-Jessen, Klaus e Ceron, J. a. M. e. H. C. (2016). Anatomia de ataques a servidores sip. http://www.cert.br/docs/palestras/certbr-ctir2013-1.pdf.

Thermos, P. e Takanen, A. (2008). *Securing VoIP Networks*, capítulo 3, páginas 53–125. Addison-Wesley.

Weissheimer Júnior, Carlos Alfredo e Bastos, E. L. (2008). Honeynet - estudo teórico e experimentação. *Seminário de Pós-Graduação*.

