Implementação e configuração de um ambiente de monitoramento com honeypots através da interface gráfica integrada QueenVision

Giovani de Alencar Freitas[1], Carlos Eduardo Pagani[1]

[1] Campus Hortolândia - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – 13183-250 – Vila São Pedro - Hortolândia – São Paulo – São Paulo – Brasil.

giovani.alencar@aluno.ifsp.edu.br, pagani@ifsp.edu.br

Abstract. *In a world increasingly dependent on technological resources tends to impose increasingly complex relationships and policies on users, this human factor is constantly exploited by hackers. Honeypots have their origin based on the academic issue, on the idea of learning. This work is a case study of the implementation of honeypots in network environments to understand the technology and analyze data through the creation of a graphical interface module called QueenVision for online data visualization, to facilitate technology teaching. Using a Python language together with the WEB framework, created in order to be a teaching support tool, QueenVision is easily present, helps in the assimilation and understanding of the representative concepts.*

Resumo. *Em um mundo cada vez mais dependente de recursos tecnológicos tende a impor relações e políticas cada vez mais complexas aos usuários, esse fator humano é constantemente explorado por hackers. Os honeypots tem sua origem pautada na questão acadêmica, na ideia de apreender. Este trabalho é um estudo de caso da implementação de honeypots em ambientes de rede para compreensão da tecnologia e análise de dados através da criação de um módulo de interface gráfica intitulada como QueenVision para visualização on-line dos dados, para facilitar o ensino da tecnologia. Utilizando a linguagem Python junto com um módulo WEB, criado no intuito de ser uma ferramenta de apoio ao ensino, o QueenVision é facilmente customizável, ajudando na assimilação e compreensão dos conceitos envolvidos.*

1. Introdução

A informação tanto hoje quanto nos primórdios dos tempos é vista como um dos ativos mais valiosos na sociedade, porém sua importância é ligada direta e intrinsecamente no tempo; a velocidade de disponibilização de tal informação junto com sua análise se torna fundamental para sua viabilidade e uso com maior assertividade.

Devido a popularização de diversas ferramentas de rede, e a disponibilização deste conhecimento a nível global por meio da internet, além da demanda de tecnologia e infraestrutura gerada pelo mercado que consome cada vez mais esses tipos de serviços,

sistemas computacionais estão cada vez mais propensos a serem violados [Psafe, 2018].

Na última medição disponibilizada pela ITU Agência da ONU para Tecnologia da Informação e Comunicação Mundial de 2017 e 2018, os prejuízos advindos dos ataques cibernéticos no Brasil ultrapassaram US$ 20 bilhões [ITU, 20019].

Segundo Assunção (2008) honeypot é uma ferramenta ou sistema criado com objetivo de enganar um atacante e fazê-lo pensar que conseguiu invadir o sistema, quando na realidade, ele está em um ambiente simulado, tendo todos os seus passos vigiados, esse recurso vem ganhando cada vez mais notoriedade devido sua eficiência em obter informações e a ideia de interrupção de ataques e possíveis contra-ataques.

Esse trabalho tem o objetivo de implementar honetpots em ambientes controlados como suporte a segurança de redes, provendo em tempo real informação através de um módulo de interface que seja capaz de fornecer dados de maneira gráfica e simplificada.

O módulo de interface gráfica tem o intuito de ser uma aplicação inteiramente aberta a customização, com foco no ensino e no estudo, tanto da tecnologia dos honeypots quanto da exposição das informações coletadas.

A partir do pensamento de Travis Lowdermilk, compreendesse que apostar em elementos gráficos, intuitivos*, User Friendly* mudam de forma dramática a aceitação da tecnologia pelo mercado pois aumenta o interesse da camada de ensino, reduz o tempo gasto com implementação e aprendizado, facilitar o *onboarding* na ferramenta, acelera resultados obtidos em algumas atividades; é a essência da transformação digital [Travis Lowdermilk , 2013].

Por utilizar uma linguagem popular como *Python* e sua criação ser formulada utilizando a arquitetura MVT, o módulo de interface gráfica ajuda a intender e compreender as interações dos *honeypots* com a rede e seus *logs*, tornando o ensino mais visual e dinâmico. A informação considerada útil captada pelos honepots pode ser persistida para estudos posteriores, formulação de perfil de invasores, mineração de dados e aplicação de inteligência artificial na massa dos dados, dando a possibilidade de estar um passo a frente das eventualidades cibernéticas.

2. Referencial teórico

O referencial teórico foi estruturado de forma descritiva para que seja compreendida a linha temporal dos *honeypots* e a maturidade da tecnologia até o ponto da criação desse trabalho.

Honeypots são recursos computacionais dedicados a serem sondados, atacados ou comprometidos, num ambiente que permita o registro e controle dessas atividades SPIT ZNER(2002,p.23); Segundo Assunção (2008), *honeypot* é uma ferramenta ou sistema criado com objetivo de enganar um atacante e fazê-lo pensar que conseguiu invadir o sistema, quando na realidade, ele está em um ambiente simulado, tendo todos os seus passos vigiados, esse recurso vem ganhando cada vez mais notoriedade devido sua eficiência em obter informações e a ideia de interrupção de ataques e possíveis contra ataques.

Conforme as ideias de Marcos Pitanga(2005), Honeypot é uma ferramenta de estudo utilizada na Tecologia da Informação; tendo em mente que, métodos de prevenção habitualmente usados como anti-virus, *firewalls* e IDS, não devem ser deixados de lado.

Na literatura temos o Livro The Cuckoo's Egg" de Clifford Stoll como primeiro relato de sistemas computacionais específicos para coleta de informações de ataques cibernéticos, pois não há relatos de redes criadas para atrair ataques de hackers para estudos antes de 1990. Porém podemos estender a ideia de observar um inimigo para

obter conhecimento e assim prover proteção de territórios dos relatos de guerras ao longo dos tempos.

Uma guerra também é engodo. Quando você for capaz, finja incapacidade, quando você for bom deslocando unidades finja que é incapaz. Quando você estiver perto finja estar longe, quando você estiver longe, finja estar perto (TZU,2019.p.26). Esse era um dos pontos principais citados por Sun Tzu para sempre estar um passo à frente do inimigo, e essa é a ideia de um honeypot, conhecimento!

Ainda em seus primórdios tecnológicos, foram sendo criadas ferramentas para o desenvolvimento desta tecnologia de forma mais eficaz: *Deception Toolkit*(DTK) desenvolvida por Fred Cohen, foi a primeira ferramenta para esse uso que se tem conhecimento ganhando notoriedade em 1997. Logo depois foi reconhecido seu potencial de mercado e em 1998 a primeira ferramenta comercial, o CyberCop Sting, foi desenvolvida por Alfred Huger, para Windows NT [Grimes. 2011].

No final da década de 90 deu-se origem ao *Honeynet* Project, um grupo de pesquisa formado por 30 profissionais de segurança dedicado a pesquisar sobre a comunidade hacker para fins de conhecimento [Sptizner, 2011].

No Brasil, temos como referência de coleta de dados por honeypot/honeynets o CERT.BR mantida pelo NIC.BR, do Comitê Gestor da Internet no Brasil, é um dos maiores centros de tratamento de incidentes cibernéticos da América; conforme pode se observar na figura 1:

Figura 1 – Localidade dos CSIRTs (Computer Security Incident Response Teams)[1]

Através de seus *honeypots* a atuação d CERT.BR abrange qualquer rede brasileira conectada a internet, sendo o ponto central para atuações de grande porte em questão de coordenação e apoio a processos em todos os centros de informação do país assim como é demonstrado na figura 2.

[1] CERT.BR, Honeytarg.São Paulo: Comitê Gestor da Internet no Brasil, Disponível em: < https:// https://www.itu.int/pub/D-IND > Acesso em: 15 mar julho. 2021

#	City	Institutions	#	City	Institutions
01	São José dos Campos	INPE, CTA	14	Lins	---
02	Rio de Janeiro	CBPF, Eletrobras, Eletronuclear, Embratel, Furnas, Oi, RedeRio, VIVO	15	Passo Fundo	---
			16	Curitiba	PoP-PR
03	São Paulo	ANSP, CERT.br, LOCAWEB, PRODESP, TIVIT, UNESP, UOL, USP	17	Belém	---
			18	São Leopoldo	Unisinos
04	Campinas	ITAL, SEFAZ-SP, UNICAMP	19	Belo Horizonte	CSIRT PoP-MG, CEMIG
05	São José do Rio Preto	UNESP	20	Recife	Chesf, NLINK
06	Piracicaba	USP	21	Salvador	UFBA
07	Petrópolis	---	22	Vitória	PoP-ES
08	Brasília	CTIR Gov, Defesa, Eletronorte, UnB	23	Americana	---
09	Porto Alegre	CERT-RS, Commcorp, PROCERGS, TRI	24	Bebedouro	MD Brasil
10	Ribeirão Preto	USP	25	Porto Velho	PoP-RO
11	São Carlos	USP	26	Rio Claro	---
12	Florianópolis	POP-SC, UFSC DAS	27	Fortaleza	MORPHUS
13	Uberlândia	Algar Telecom	28	Natal	PoP-RN

Figura 2 – Cidades e instituições dos CSIRTs (Computer Security Incident Response Teams) [2]

Através dos dados coletados por *honeypots*, o CERT conscientiza as organizações e a população sobre os principais problemas e ameaças ocorridos e que estão em percurso analisando tendências e relações entre eventos e ocorrências, tendo como objetivo principal aumentar os níveis de segurança e de capacidade de reação e incidentes de rede no Brasil [Honeytarg, 2003] ; tendo como atribuições:

1. Estabelecer diretrizes estratégicas relacionadas ao uso e desenvolvimento da Internet no Brasil;

2. Promover estudos e recomendar procedimentos, normas e padrões técnicos e operacionais, para a segurança das redes e serviços de Internet, bem assim para a sua crescente e adequada utilização pela sociedade;

3. Ser representado nos fóruns técnicos nacionais e internacionais relativos à Internet.

De acordo com Sohal(2017) *Honeypots* são recursos computacionais muito diferentes de sistema de detecção e prevenção de intrusão (IDPS) e de firewalls, pois permite que o invasor interaja com uma versão diferente do sistema real.

Vale a pena salientar essa circunstancial diferença entre *honeypots* e um IDS, se for configurado um *honeypot*, deseja que ele seja atacado, já a configuração de um IDS (sistema de detecção de intrusão) tem o objetivo de saber quando e quem está tentando um acesso indevido a rede, sendo um processo normalmente automatizado. No entanto, configurar um *honeypot* é uma maneira de detectar invasões e coletar dados sobre comportamento do atacante; portanto, tecnicamente, esses dois não são exclusivos. A categorização dos *honeypots* ajuda-nos a compreender melhor a implementação dessa tecnologia e o seu uso prático. Os *honeypots* são categorizados pelo nível de interação que oferecem às ameaças e pela forma como respondem às ações dos invasores. Sendo

[2] CERT.BR, Honeytarg.São Paulo: Comitê Gestor da Internet no Brasil, Disponível em: < https:// https://www.itu.int/pub/D-IND > Acesso em: 15 mar julho. 2021

assim, podem ser divididos em *honeypots* de baixa, média e de alta interatividade [CERT.BR, 2018].

1. Baixa interatividade: emulam sistemas operacionais e serviços, interagindo com as solicitações do agente mal-intencionado a partir do envio de respostas falsas.

2. Média interatividade: emulam coleções de software para apresentar uma frente mais convincente para o atacante, mas protegendo o sistema operacional do *host*.

3. Alta interatividade: permitem maior interação de um atacante com sistemas operacionais, aplicações e serviços reais; é necessário mais comprometimento da equipe de segurança para com a proteção e o monitoramento constante da máquina usada como *honeypots*.

É na classificação e identificação de fluxos de dados que reside o objetivo e valor de um honeypot. Trata-se de um procedimento difícil e demorado, pois a quantidade de dados obtida é alta, sendo necessária uma filtragem de quais informações são consideradas úteis, o que exige conhecimento para distinguir sua utilidade. Uma técnica da Inteligência Artificial (IA), denominada Teoria dos Rough Sets, pode auxiliar na classificação desses fluxos de dados, para uma posterior identificação. Indo além dos *honeypots* temos *honeynets* e *honeyfarns, honeynet* são *honeypots* interligados a fins de simular uma rede completa de computadores/servidores, para estudos mais aprofundados de observação de atividades hackers e ameaças virtuais em geral, dentro da ideia de *honeynet* tambem existe a ideia de honeynet virtual que seria toda uma rede de honeynet dentro de apenas um servidor [Lance Spitzner,2002]. Conforme explica Nemati e Hamid(2007) *Honeynets* são ótimas soluções para ambientes de pequeno/médio porte já para ambientes mais robustos geograficamente distantes existe a solução conhecida como honeyfarm. *Honeyfarm* compreende a um aglomerado de *honeynets* correlacionados para diferentes fins. Dentro da temática dos *honeypots* sua comunicação pode ser implementada de diferentes formas: VLANs, túneis GRE , etc.

A essência do *honeypot* é ser realmente simples, descomplicado e prover informações de maneira rápida para um possível alinhamento da segurança da rede que habita, ele tem que ser constantemente revisto e redirecionado reconfigurado para poder manter sua eficiência.

> A principal vantagem de um *honeypot* é que ele não depende de assinaturas, regras ou algoritmos avançados, ele simplesmente captura os ataques que lhe são direcionados, facilitando a análise e correlação da pequena, mas valiosa, coleção de dados. Mas possui algumas desvantagens, sua capacidade limitada de visão, ou seja, ele só captura os dados direcionados a ele mesmo JABOUR(2003, p.8).

Existem uma grande gama de soluções rápidas para implementação de honeypot, essa facilidade acaba se tornando um grande problema, pois ferramentas de fácil uso costumam ser utilizadas com configurações padrões e/ou mau configuradas e isso acaba se tornando vetor para reais ataques, assim descaracterizando o uso de *honeypots*:

> Utilizar um *honeypot* é muito interessante do ponto de vista da facilidade de se detectar invasões e puder, assim, melhorar os sistemas existentes de detecção de intrusos. Mas é importante ressaltar que também é uma faca de dois gumes, caso o *honeypot* configurado esteja comprometido e o invasor consiga utilizá-lo para atacar outras redes, isso pode causar um grande problema. Portanto é importante pesar e balancear quando utilizar os serviços de baixa interatividade, que não oferecem praticamente risco algum, com aqueles de alta interatividade que, apesar de ajudarem a recolher mais informações interessantes, podem fazer o feitiço virar contra o feiticeiro. Outra questão muito comentada sobre os *honeypots* é em relação ao aspecto jurídico de sua utilização. Algumas pessoas alegam que um honeypot induz alguém a fazer algo errado e isso de forma alguma é verdade. Um pote de mel não está induzindo ninguém a realizar nada de errado, até porque muitas vezes ele é um computador como qualquer outro da rede, apenas a finalidade de colocá-lo ali é que foi diferente. O *honeypot* não está sendo exibido para ninguém, o invasor entrou porque quis. ASSUNÇÃO(2009, p. 27).

PROVOS(2007) ressalta que *HoneyD* é um dos principais softwares utilizados para criação de *honeypots*, de baixa interatividade, devido a seu amplo uso por instituições de ensino e organizações governamentais. Sua distribuição gratuita e seu código livre facilita essa interação com empresas e comunidades de tecnologia.

Dentre as ferramentas que podem ser utilizadas junto ao *HoneyD*, é interesante citar a *Honeyview*, aplicação criada por Niels Provos, é uma ferramenta de análise de arquivos de log para o *Honeyd*. Ele fornece uma visão geral gráfica dos dados coletados, mas também fornece uma saída textual detalhada para eventos. O *Honeyview* pode ser usado para determinar quais portas e endereços IP estavam mais ativos e também suporta plotagem de séries temporais [honeyd.org/tools, 2008]. Porém foi descontinuada e não sofre atualizão des de 2003[SourceForce, 2021].

3. Trabalhos correlatos

Em consideração aos trabalhos correlatos sera citado o trabalho de conclusão de curso de Ricardo de lima da Universidade Regional de Blumenau - Centro de Ciências Exatas e Naturais - Curso de Ciência da Computação, Trabalho esse intitulado como *Webservice honeypot* utilizando a biblioteca *Jhoney*, esse trabalho foca na importância dos *honeypots* em serviços *web* e na disponibilização de uma ferramenta para simular tais serviços para coletas de informações sobre acessos indevidos.

Podemos o artigo científico dos autores Martim d'orey Posser de Andrade Carbone e Paulo Lúicio de Geus intitulado como: Um mecanismo para coleta automatizada de evidências digitais em *honeypots* de alta interatividade, artigo esse publicado em 2004 na Scientia – Revista de Computação da UNISINOS, esse artigo evidencia a metodologia para criação de um *honeypot* de alta interatividade, levando em consideração processos de automação da coleta de evidencias computacionais geradas pelo *honeypot*.

4. Objetivo geral

Implementação de *honeypots* como ferramenta de seguraça em ambientes controlados e criação de um modulo para rapida e simplificada visualização e análise dos dados úteis coletados via *Honeypot* visando facilitar o estudo e o ensino.

3.1 Objetivos específicos

Compreende aos resultados que se pretender alcançar apartir do desdobramento do objetivo geral desse trabalho.

1. Compreender a tecnologia dos *honeypots* e sua aplicação e benefícios;

2. Simular um ataque de exploração para verificar o comportamento do sistema nesse ambiente;

3. Criar a arquitetura da interface gráfica de maneira aberta e modular para facilitar a aplicação e reprodução em ambientes de ensino.

5. Ferramentas e métodos

Para a realização desse trabalho, seguindo um modelo de pesquisa exploratória, foi realizado um estudo sobre a implementação de *honeypots* para coleta de dados.
Esse capítulo apresenta as ferramentas(softwares e hardwares) utilizados para montar o ambiente para implementação da coleta de dados.

Hardware

Foi utilizado um equipamento do tipo Servidor *PowerEdge* T140, BCC, com processador Intel *Xeon* E-2124 de 3,3GHz, cache de 8MB, 4 núcleos / 4 segmentos, com turbo (71W), e 16GB UDIMM DDR4 de 2666 MT/s, 1TB SATA cabeado, 6 Gbps, 7200 RPM e 3,5".

Software

Este tópico referencia os principais softwares, ferramentas e sistemas operacionais utilizados nessa demosntração. Conforme apresenta a figura 3 podemos compreender a relação entre os *hosts* no ambiente.

Sistema hospedeiro

Como sistema hospedeiro do ambiente foi utilizado o Archlinux versão 2019.11.01-64bits, por se tratar de uma versão estável e com comunidade ativa de profissionais e colaboradores do Linux quando se trata de servidores, vale a pena ressaltar que o ponto central da versão é justamente a estabilidade e eficiência, que foram de grande importância para a virtualização dos outros sistemas em questão.

Virtualizador e gerenciador de ambientes

Para essa implementação foi escolhido *VirtualBox* versão 6.0.14-1, por ser simples e de fácil configuração, além de ser desenvolvido pela Sun Micro Systems uma empresa que é considerada referência quando o assunto é virtualização.

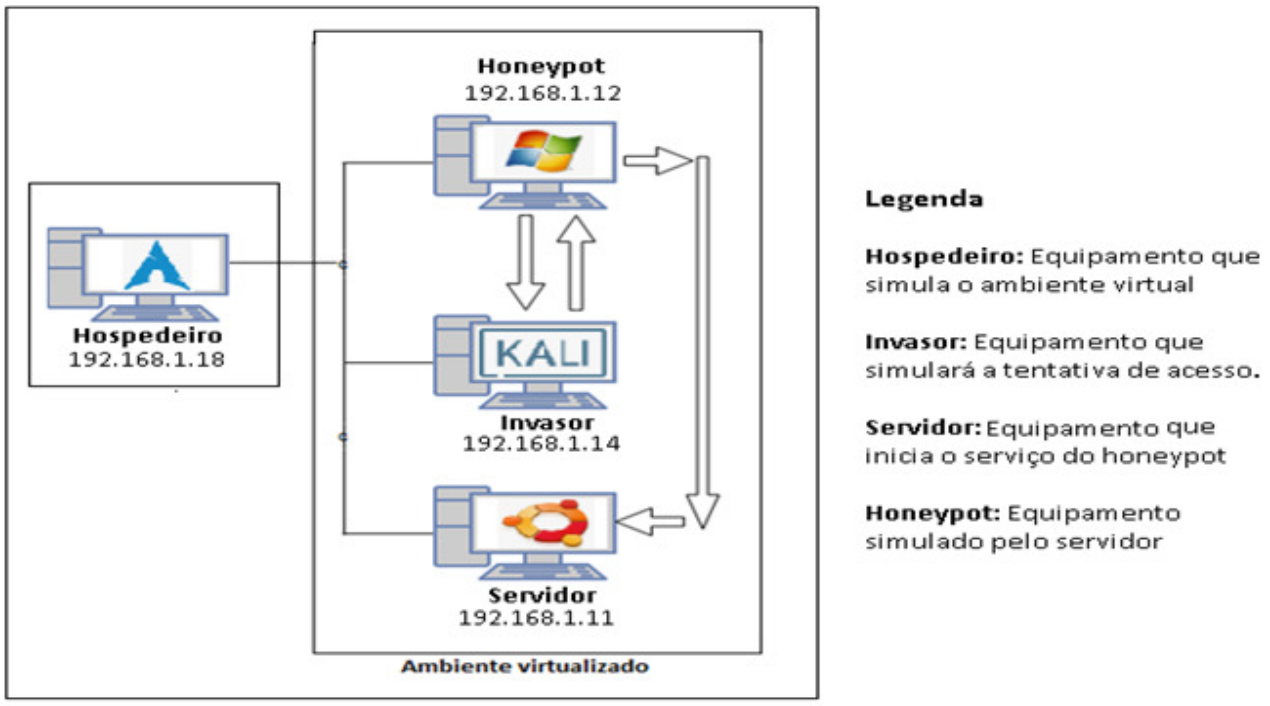


Figura 3 – Ambiente de teste

Sistema operacional hospedeiro do Honeypot

O sistema operacional escolhido para a implementação do serviço de honeypot foi Ubuntu 18.04- LTS64bits, por boa parte do material consultado envolvendo *honeypots* ser escrito com base nesse sistema.

Software de Honeypot

Como Serviço de honeypot escolhemos a ferramenta HoneyD que é distribuída gratuitamente possuindo código aberto, consideramos o aspecto de que é a ferramenta utilizada pelo Cert.br para suas coletas de dados.

Sistema operacional instigador dos serviços

O sistema operacional Linux Kali 2019.3- 64bits foi escolhido tendo como requisito a eficiência, a consideração da comunidade para os fins desejados, essa versão já é contemplada com a maioria das ferramentas necessárias para implementação dos testes.

Nmap

Foi utilizado o Nmap 7.80-1 por se tratar de uma ferramenta robusta e muito utilizada entre profissionais para escanear portas em hosts, considerando que ela vem instalada por padrão no Linux Kali desde suas primeiras versões. Neste trabalho ele foi usado para scanear a rede virtualizada e identificar o potencial alvo.

MetaSploit

MetaSploit 4.16.0 é um software construído como um projeto de segurança da informação, uma das principais ferramentas de código aberto a serem consideradas para testes de penetração e contemplação de vulnerabilidades, também vem como padrão em todas as distribuições do Linux Kali. Sendo a ferramenta escolhida para a tentativa de acesso via *exploit*.

6. Implementação

O sistema operacional Arch Linux possui a função de hospedeiro das máquinas virtuais do ambiente dos experimentos, na virtualização do sistema Ubuntu que tem como objetivo ser o servidor de *honeypots*, a configuração da conexão da rede está definida como *Bridged Adapter* para que qualquer equipamento da rede interna possa interagir com esta estação.

Vale salientar a importância da configuração do honeyd2.conf que simulará o equipamento com brechas na segurança, neste caso foi escolhido um sistema já defasado como Windows XP SP1 para estimular ainda mais os possíveis atacantes, pois é um sistema que vem sendo constantemente utilizado por sistemas legados e não possui mais suporte do fabricante, e por ser o único sistema operacional nessas condições que ainda consta na lista dos 10 mais utilizados de 2020(w3schools, 2020).

Para outras situações vale a apena mencionar que nativamente o *HoneyD* pode simular 986 tipos de sistemas operacionais, mainframes, roteadores e etc presentes no arquivo de configuração “nmap.assoc”. Ainda é possível configurar outros sistemas através da edição do mesmo arquivo; foi definido as portas abertas para possíveis ataques nos serviços disponíveis, assim como demonstra a figura 4.

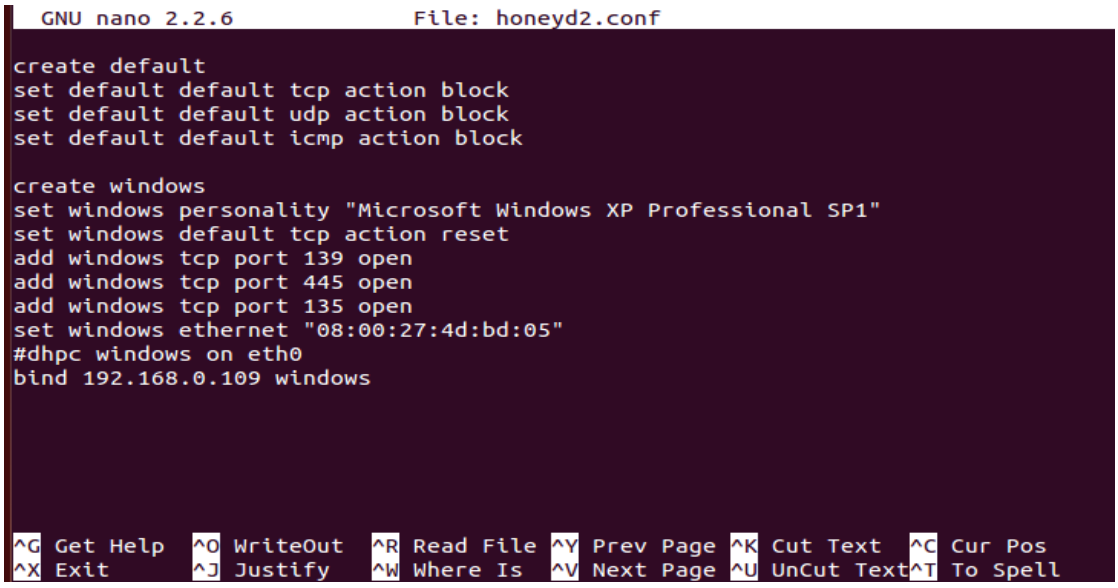


Figura 4 - Script de configuração da estação simulada

Após o termino das configurações utilizamos o comando"sudo honeyd -d -f honeyd2.conf -l /tmp/log/honeyd2" para iniciar o Honeypot, onde o parâmetro -d tem a função de executar o *honeypot* em primeiro plano para que assim o processo oferte todas as informações disponíveis do *honeypot* dando assim uma impressão maior de ser um sistema real para o atacante, o parâmetro "-f" especifica que um arquivo será utilizado para levantar o serviço, e "-l" cria o *log* das atividades do *honeypot* em questão.

A se tratar da virtualização do sistema Linux Kali para tentativa de invasão, foi utilizado também configurações de rede da VM como *Bridged Adapter* para os mesmos fins já relatados, iniciamos primeiramente o Nmap como *scanner* a rede utilizando a faixa de IP da rede local como alvo. Com o resultado do comando "sudo nmap 192.168.1.0/24" identificamos facilmente o potencial alvo, o *honeypot*. Foi facilmente identificado na rede o *honeypot*, conforme podemos ver na imagem 5, e por suas configurações se torna um potencial alvo na rede.

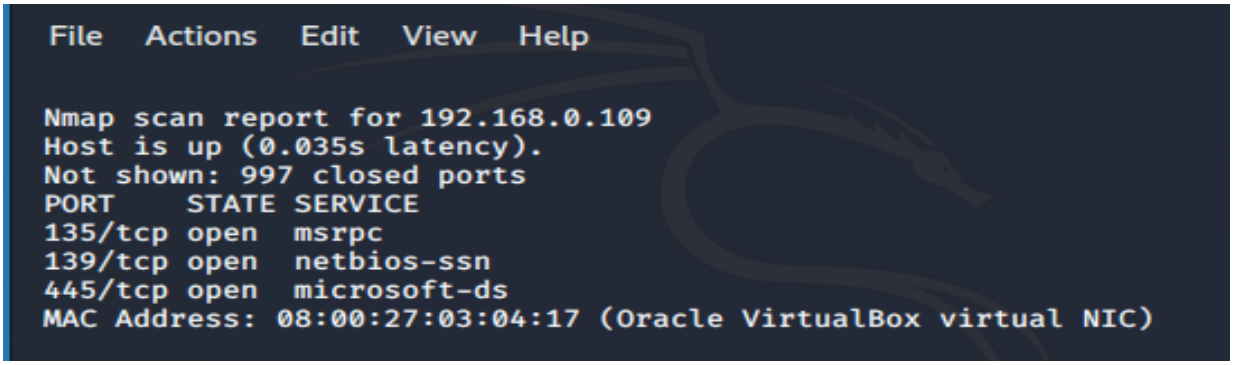


Figura 5 – Potencial alvo exposto pelo NMAP

É evidente a disparidade na questão da segurança entre as plataformas onde a oportunidade de exploração se faz. Como já foi realizado a prospecção da informação é iniciado o MetaSploit com o comando "msconsole" dentro do console é escolhido o *exploit* exploit/windows/dcerpc/ms03_026_dcom através do comando "use *exploit/windows/* dcerpc/ms03_026_dcom", pois ele se utiliza de uma brecha do protocolo MSRPC atuando na porta 135, foi iniciado e definido os dados para a *payload* conforme a figura 6, com os respectivos comandos:

```
set rhost 192.168.0.109 // IP do host alvo

set payload windows/meterpreter/reverse_tcp // especifica payload

set LHOST 192.168.1.105

show option // verifica os parâmetros passados pelo console

exploit // executa o exploit
```

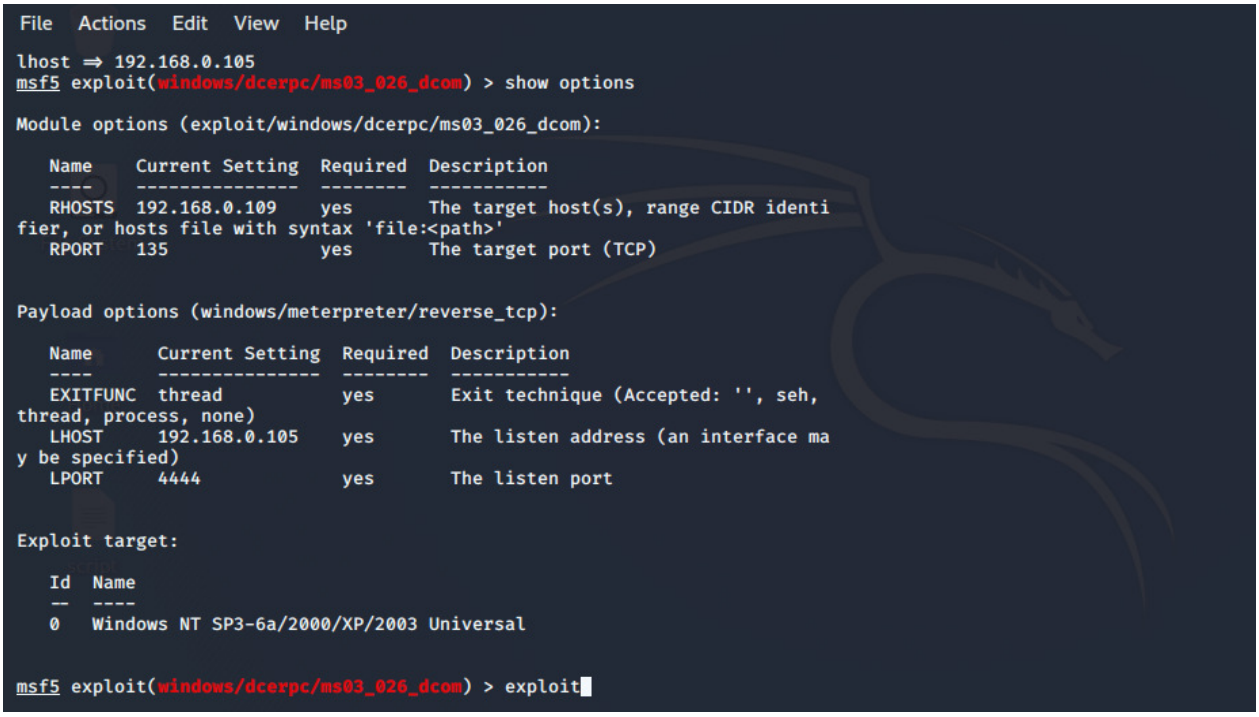


Figura 6 – Cabeçalho da payload

Foi executado o teste de penetração, o *exploit* conecta, mas não inicia o interpretador para comandos remotos, conforme mostra a figura 7.

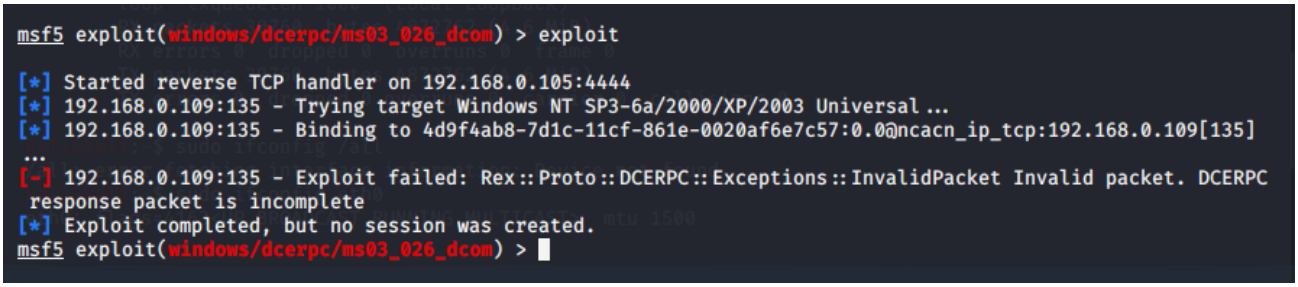


Figura 7 – Tentativa de penetração no honeypot

No sistema operacional Ubuntu se acessar o arquivo /tmp/log/honeyd2 existem informações realmente úteis sobre o ataque, tais como: nome do host, IP, horário e data do ataque, quantidade de pacotes, tipo de ataque. Esse é o intuito do experimento, mostrar um ambiente com uma aplicação de honeypot em execução.

7. Desenvolvimento

Intitulado "*QueenVision*" o módulo de interface gráfica criado com a IDE *Pycharm* para visualização rápida e simplificada dos dados coletados pelo servidor de *honeypot.* Foi utilizado a linguagem *Python* para o desenvolvimento da mesma junto ao *framework Django* levando em consideração a arquitetura MVT. Na figura 8 pode ser acompanhado o fluxo das requisições dentro dessa metodologia.

Figura 8 – Interações da aplicação no modelo MVT[3]

A figura 9 mostra as principais classes da aplicação junto a árvore de diretórios do projeto.

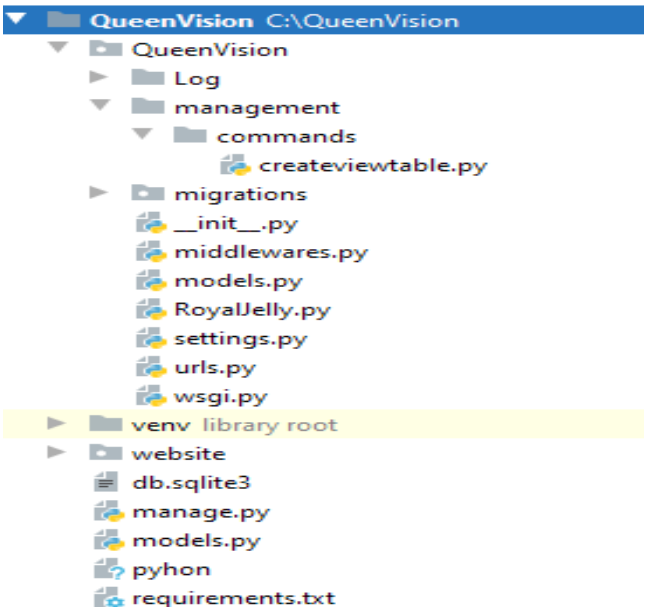


Figura 9 – Arquitetura da aplicação

RoyalJelly.py: classe responsável pelo acesso ao arquivo de *log* e por separar as informações uteis contidas no arquivo de log para persistência no banco de dados. Pode ser executada separadamente do resto da aplicação para análises posteriores ou dentro do código junto a *view.*

createviewtable.py : classe responsável por gerar *views* no banco de dados a partir de anotações dentro do *model*.py, assim podemos organizar a criação de novos gráficos

[3] DEV.TO, Django Tutorial - MVT Architecture, Custom Commands, Disponível em: <https://dev.to/sm0ke/django-tutorial-mvt-architecture-custom-commands-19nb > Acesso em: 17 mar julho. 2021

utilizando as *views,* tornando mais rápido e fácil o processo de customização da aplicação.

models.py: modelo de dados, contém os campos a serem persistidos, conforme pode ser visto na figura 10.

A pasta venv fornece a aplicação um modulo para execução de um ambiente virtual contendo seus próprios conjunto de pacotes, facilitando a execução da aplicação por terceiros.

A pasta *Website* contém os *templates* da aplicação e arquivos para *layout*.

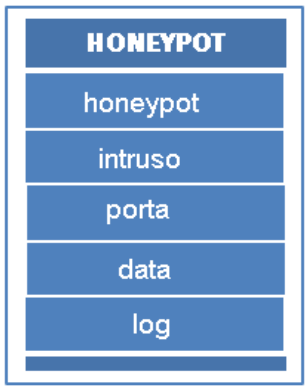


Figura 10 – Modelo de dados

Os dados que são coletados pelo *honeypot* são enviados a um servidor de dados onde através da classe RoyalJelly.py conforme é mostrado na figura 11, são tratados e escritos em um único banco de dados para persistirem. O servidor de dados foi mapeado como unidade de acesso no computador, pois a interface foi executada localmente.

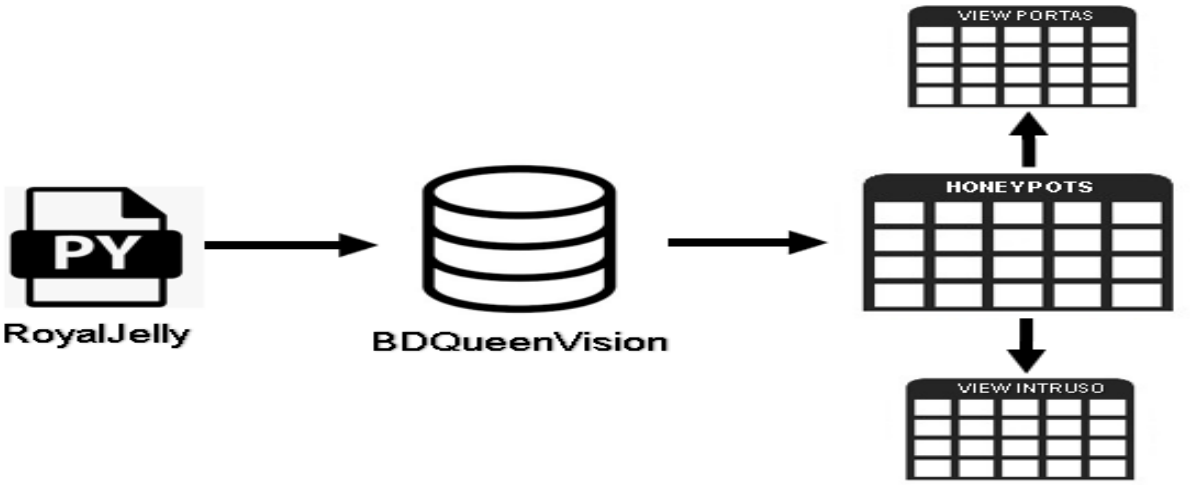


Figura 11 – Fluxo de persistência dos dados

A partir da execução da aplicação e fornecido as credenciais de acesso temos a tela principal conforme apresentado na figura 12. Essa tela apresenta as funcionalidades do *Queenvision* de maneira objetiva e de fácil visualização.

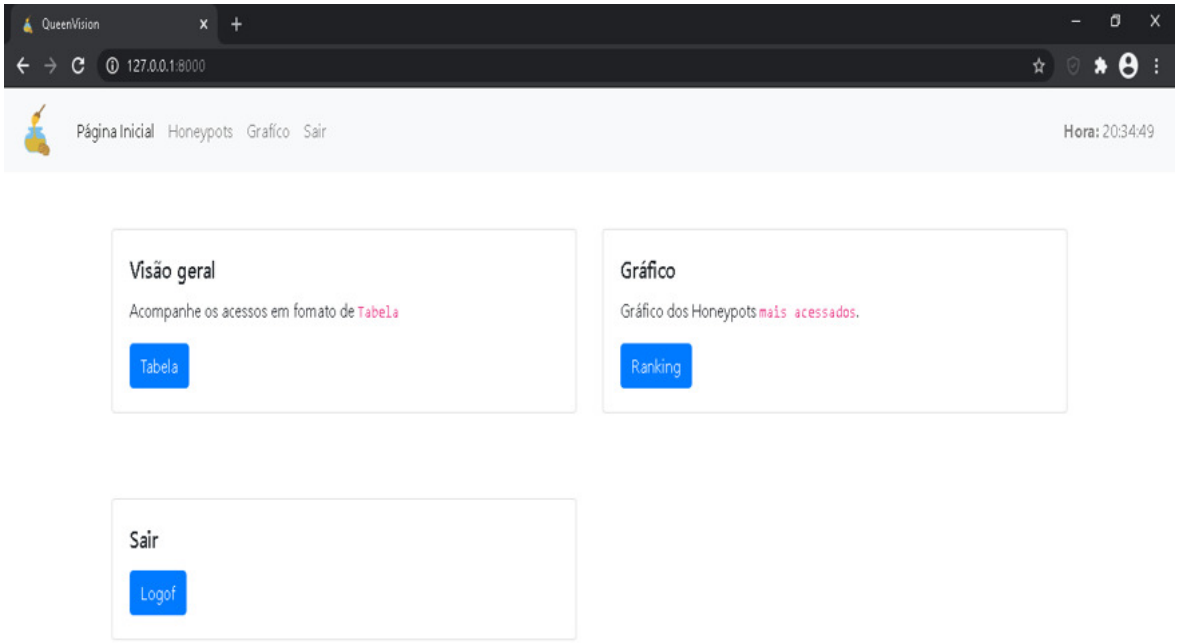
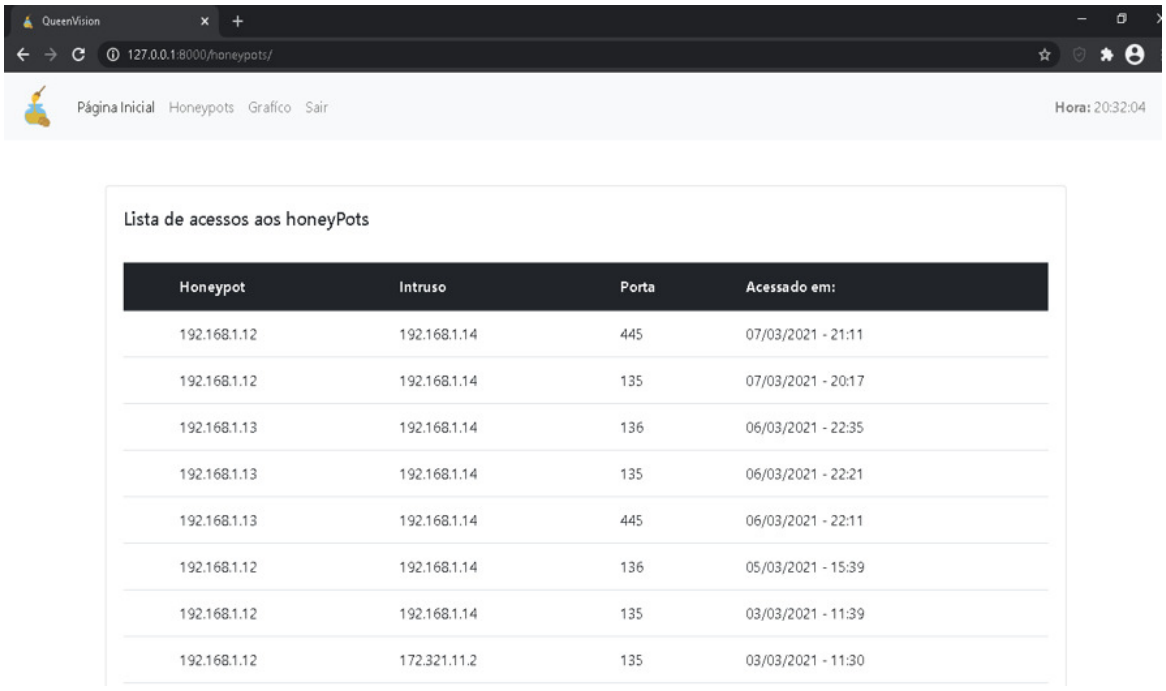


Figura 12 – Tela principal da aplicação

Quando clicado na opção “Tabela” a aplicação mostra todas as interações entre os invasores e os *honeypots,* a porta que foi utilizada para a tentativa de acesso, a data e a hora que as interações aconteceram de modo tabular, conforme apresenta a figura 13.



Honeypot	Intruso	Porta	Acessado em:
192.168.1.12	192.168.1.14	445	07/03/2021 - 21:11
192.168.1.12	192.168.1.14	135	07/03/2021 - 20:17
192.168.1.13	192.168.1.14	136	06/03/2021 - 22:35
192.168.1.13	192.168.1.14	135	06/03/2021 - 22:21
192.168.1.13	192.168.1.14	445	06/03/2021 - 22:11
192.168.1.12	192.168.1.14	136	05/03/2021 - 15:39
192.168.1.12	192.168.1.14	135	03/03/2021 - 11:39
192.168.1.12	172.321.11.2	135	03/03/2021 - 11:30

Figura 13 – Tela de visualização dos dados

Foi utilizado a biblioteca gráfica *Chart.JS* para a geração dos gráficos por ser uma biblioteca simples de utilizar e com uma comunidade muito ativa, levando em consideração a qualidade dos gráficos e seu dinamismo.

O gráfico mostrado na figura 14, mostrar a parte que representa por cada *honeypot*

sobre o montante dos acessos indevidos registados.

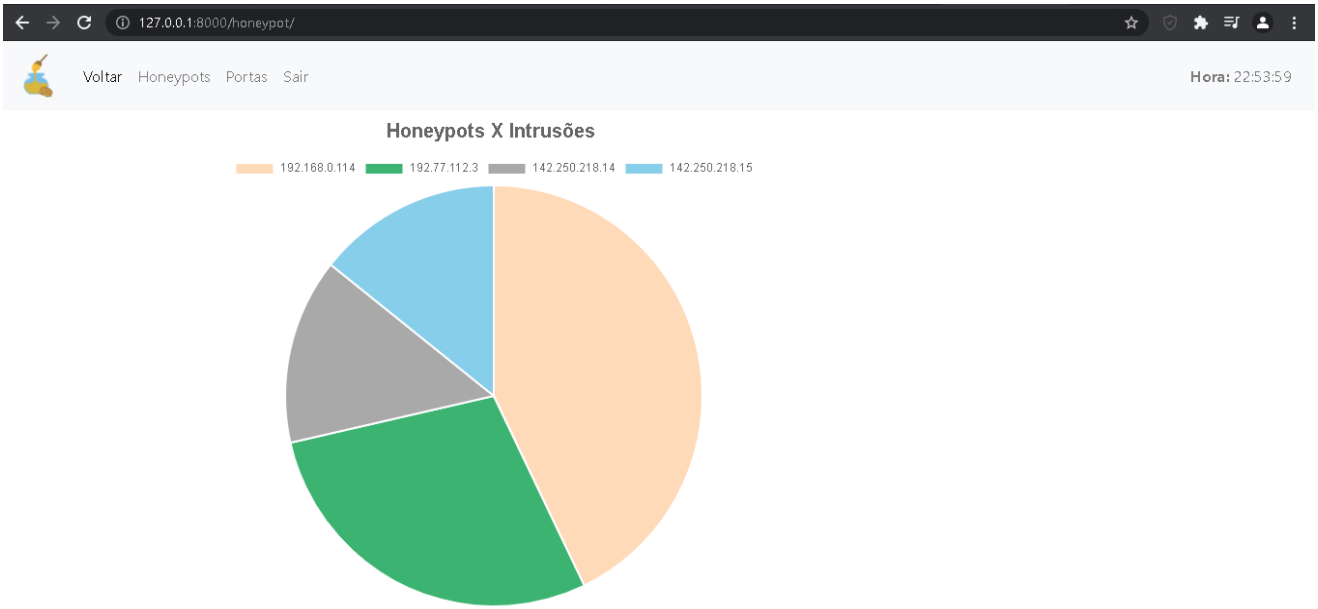


Figura 14 – Gráfico de honeypots mais acessados

O gráfico da figura15, mostra a quantidade de tentativas indevidas de acesso por porta.

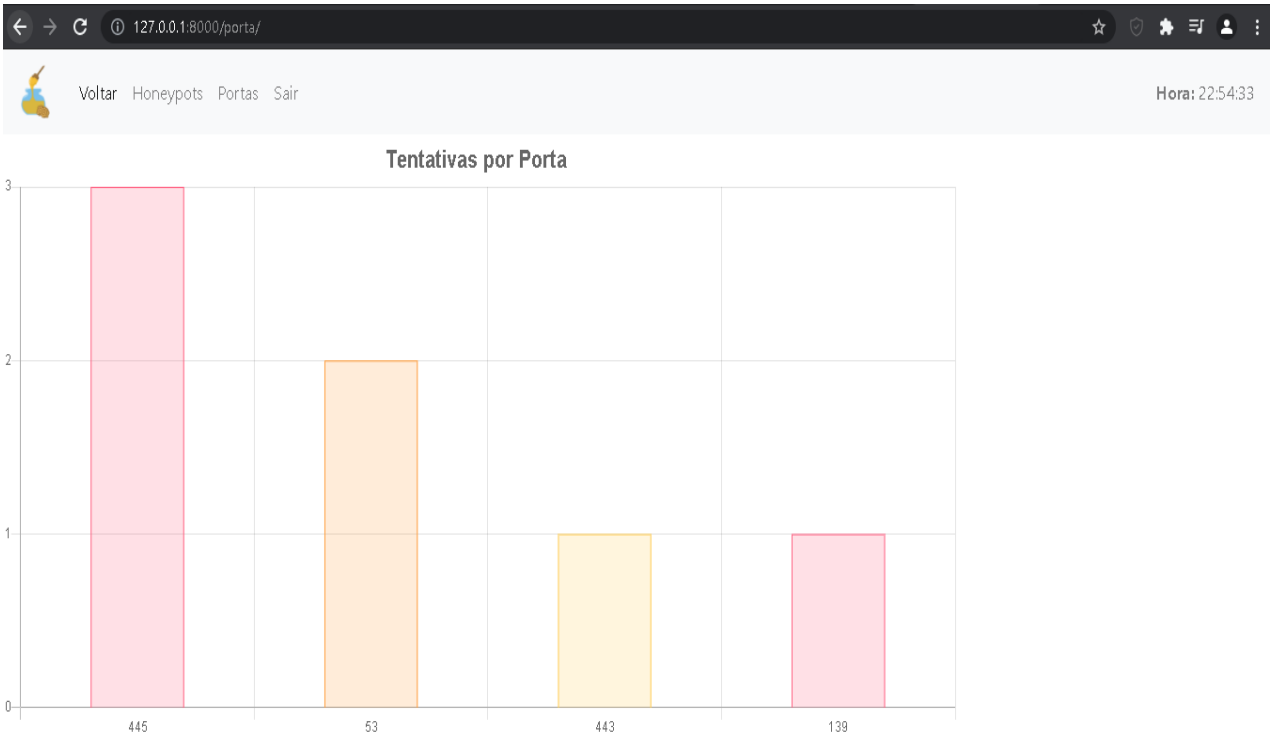


Figura 15 – Gráfico que representa as portas mais acessadas

8. Análise dos resutados

A partir do referencial teórico é compreendido o potencial dos *honeypots*, que historicamente vem sendo usados para entender as deficiências dos cenários onde são

implementados, tratasse de uma realidade já concretizada a capacidade que essa tecnologia tem de demonstrar as interações indevidas com *hosts*.

Ao estabelecer um cenário comum para implementar o *honeypot*, existem questões de compatibilidade das plataformas de software, pois diversas bibliotecas devem ser instaladas separadamente em uma ordem específica para que possam ser usadas e acessadas conforme a arquitetura da aplicação, a respeito do hardware o *HoneyD* não exige hardwares muito robustos para funcionar com sua completa gama de possibilidades.

Por ser uma ferramenta que funciona em tempo real, ou seja, conforme as interações na rede e como próprio *Honeypot* e sem interface gráfica para uma análise dos dados é necessário habilitar as opções para que sejam salvas todas as interações com o *Honeypot* para que assim se possa fazer uma análise posterior dos dados. Porém, esse tipo de análise não se faz de forma rápida ou interativa, pois o arquivo de *log* gerado é comumente extenso. Assim como comentado, existem ferramentas gráficas para a análise, porém não são comumente usadas, pois não sofrem atualização, podendo ser potencialmente instáveis ou inseguras, e não oferecem uma análise posterior viável e sua arquitetura não é modular tornando a customização e a atualização por terceiros mais complicada, ou em algum caso inviável. Sem um modulo gráfico para análise em tempo real ou para análise e/ou apresentação dos potenciais riscos, a aplicação fica potencialmente distante do interesse acadêmico, pois não acaba alcançando o engajamento na camada discente, tornando seu estudo algo mais específico.

O honeypot após configuração e implementação entregou os resultados prometidos, interagindo como um host qualquer na rede, o NMAP assim como mostrado nesse trabalho considerou *honeypot* como parte comum da rede, e apos a prospecção, a interação do *Honeypot* com o *MetaSplot* ocorreu assim como se esperava, sendo mostrada como tentativa de interação no log do *HoneyPot* junto ao seu IP e porta relacionada.

O modulo de interface QueenVision mostrou-se uma ferramenta interessante na visualização gráfica das informações em tempo real, sua implementação pode ser realizada de diversas formas, tanto como ferramenta de análise pós incidente *offline* ou on-line, além do modulo intitulado *RoyaJelly* pode ser facilmente customizado para a apresentação dos dados da forma que o administrador melhor desejar. Por ser escrito com *Django* utiiza-se o modelo MVT, possibilitando a interação de modelo com o banco de dados a partir disso podemos criar *views* para criação dos gráficos de modo mais facil e eficiente a partir da biblioteca manager.

O modulo gráfico pode ser usada para tornar o ensino e a aprendizagem de *honeypots* mais atraente. A informação sendo disponibilizada de forma rápida e simples com gráficos e tabelas ajuda na questão da tomada de decisão, e na área de segurança da informação tempo é uma razão que pode determinar a eficiência ou o fracasso de uma tentativa de penetração de uma determinada rede.

9. Conclusão

Através dos estudos e dos testes realizados, a ideia de auxiliar a proteção de redes com *honeypots* mostra-se viável, pois os custos são relativamente baixos se utilizados com softwares livres. Como citado; há níveis de interação e ação dos *honeypots*, pois podem ser usados exclusivamente para coleta de dados ou/e manutenção da segurança promovendo ataques automatizados a invasores da rede, em hipótese nenhuma esse trabalho sugere a implementação de *honeypots* a fim de serem a defesa principal de uma rede ou afirmar a eficiência, pois denotaria estudos mais detalhados e continuidade

desses estudos pela comunidade acadêmica e a implementação em ambientes reais para fins de homologações e comparações entre a academia e a comunidade, mas mostra sim uma possibilidade de incrementação para o sistema de segurança da rede.

O exemplo aqui demonstrado foi apenas para a compreensão do funcionamento dessa ferramenta, pois em um ambiente real seria implementado diversos *hosts* em redes destinadas a realmente prover serviços.

Os dados colhidos por *honeypot* podem persistir em uma base interconectada de comunidades corporativas, científicas, etc; para gerar políticas de segurança com compreensão mais regionalizadas para as instituições envolvidas, podem acarretar mineração desses aglomerados de dados para aumentar a eficiência de tais políticas.

Essa arquitetura distribuída de *honeypots* é usada principalmente para monitoria de atividades em grandes centrais de dados, aplicando-se principalmente na esfera de organizações governamentais criadas especificadamente para isso. O intuito desse trabalho é mostra o potencial dos *honeypots* como forma de tornar a disponibilização de serviços mais eficiente e seguros. Pelo fato da maioria dos *honeypots* serem de código aberto são extremamente customizáveis podendo compreender inúmeras necessidades dos ambientes em que poderão ser implantados.

A interface gráfica para visualização simplifica a administração dos *honeypots* tornando a tomada de decisão mais veloz por disponibilizar a informação já tratada e de forma gráfica. Sendo um ambiente limpo focado na disponibilidade acaba se tornando muito efetivo para a administração da informação dos *honeypots*; podendo ser ponte para continuidade do trabalho pela comunidade de desenvolvimento.

O módulo gráfico torna o aprendizado mais atrativo fazendo uma divulgação orgânica da tecnologia em camadas acadêmicas engajando adeptos a segurança e a programação. Podendo ser usado tanto na estrutura apresentada quanto *off-line* para análise pós incidente, ou em aulas mais dinâmicas como ferramenta de ensino.

10. Trabalhos futuros

Ao decorrer da criação desse trabalho foi vista diversas possibilidades, e a própria continuidade desse trabalho é fomentar essas possibilidades.

Tendo isso em mente, a ideia desse trabalho não apenas a disponibilizá-lo em plataformas de compartiolhamento de projetos/codigos, mas sim, procurar parcerias e vínculos com outras comunidades, por exemplo a do próprio CERT.BR e do HoneyD, no site do HoneyD temos a área de indicações de ferramentas a serem utilizadas, pretendo disponibilizar a interface gráfica e os scripts como ferramentas oficiais do projeto HoneyD.

Deixar em aberto a oportunidade para melhoria da interface de visualização como proposta para TCCs de outros alunos no campus IFSP Hortolândia ou para qualquer um da comunidade acadêmica; pretendo fazer vídeos explicando o projeto e como implementá-lo e disponibilizá-los em plataformas de estudo on-line (por exemplo Udemy), e realizar um projeto interno para implementar o mesmo na rede do IFSP – Hortolândia com as devidas aprovações e proposta para expansão para os outros campus do instituto. Vejo a semana da Ciência e tecnologia como uma grande possibilidade para apresentar o projeto para a comunidade e disseminar o interesse da continuidade da ideia.

11. Referências

ERICKSON, Jon. Haching: The Art of Exploitation. 2 ed. San Francisco: No Starch Press, 2008.

IMONIANA, Joshua Onome. Auditoria de Sistemas.3 ed. São Paulo: Gen/Atlas, 2016.

SOHAL, Amandeep.et al. A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. Punjab: Guru Nanak Dev University, p.2,12 jan 2017.

HONEYNET PROJECT. The Honeynet Project, Know Your Enemy. Diponivél: em< http://www.project.honeynet.org/book/> Acessado em: ago 2019.

ULBRICH, Henrique Cesar. Universidade Hacker: Exercícios Práticos para Desvendar os segredos do submundo hacker.1 ed. São Paulo: Digerati, 2007

ROSINI, Alessandro Marco; PALMISANO, Ângelo. Administração de sistemas de informação e a gestão do conhecimento. São Paulo: Thomson Pioneira, 2002.

CERT. Honeypots. Disponível em: <https://honeytarg.cert.br/honeypots>Acesso em: 25 abri. 2019.

NORTON REPORT. Norton LifeLock Cyber Safety Insights: Report Global Media Deck. Disponível em:<https://now.symassets.com/content/dam/content/pt-br/collaterals/datasheets/norton-cyber-security-insights- report2016.pdf > Acesso em: 20 dez. 2018

ITU. ITU Statistics and Indicators. Disponível em: < https:// https://www.itu.int/pub/D-IND > Acesso em: 04 mar dez. 2020

PITANGA, Marcos; MARCELO, Antonio. Honeypots: A arte de iludir hackers. São Paulo: Brasport 2003.

MEMÓRIA, Felipe. Design para a Internet: Projetando a Experiência Perfeita. Editora Campus, 2006.

SPITZNER, Lance. Honeypots: Tracking Hackers. Boston, Addison Wesley, 2002.

JOHNSON, S. Cultura da Interface: Como o computador transforma nossa maneira de criar e comunicar. Rio de Janeiro: Jorge Zahar, 2001.

LOWDERMILK, T. Designer Centrado no Usuário: Um Guia para desenvolvimento de Aplicativos Amigavéis. São Paulo: Novatec, 2019.

ASSUNCAO, Flavio. Honeypots e honeynets Aprenda a detectar e enganar invasores. São Paulo: Visual Boks 2009.

PROVOS, Niels; HOLZ, Torsten. Honeypots: From botnet tracking to intrucion detection. Boston: Person 2007.

CAMPOS, André. Sistema de segurança da informação – controlando riscos. Florianópolis: Visual Books, 2007.

DUARTE, Otto Carlos Muniz Bandeira; JABOUR, Eugênia Cristina Müller Giancoli. Honeynets: invasores, ferramentas, técnicas e táticas. Rio de Janeiro: UFRJ, 2003. Disponível em: . Acesso em: 17 jan. 2019.

MCCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. Hackers expostos: segredos e soluções para a segurança de redes. São Paulo : Makron Books, 2000.

SPITZNER, Lance. Honeypots: Tracking hackers. Boston: Addison Wesley, 2002.

CANALTECH. DDOS gera prejuizo no setor financeiro . Disponível em: <https://canaltech.com.br/seguranca/ataques-ddos-geram-preocupacao-no-setor-financeiro-156861/>Acesso em: 03 mar. 2020.

HUGE NETWORKS. Honeypots alternativa contra ataques de negação de serviço . Disponível em: <https://www.linkedin.com/company/huge-networks/>Acesso em: 03 jan. 2020.

HUGE NETWORKS. Honeypots. Disponível em: < https://www.huge-networks.com/>Acesso em: 12 fev. 2020.

CERT. Gráfico de inicidentes. Disponível em: < https://www.cert.br/stats/incidentes/2018-jan-dec/tipos-ataque.html/>Acesso em: 17 fev. 2020.

SOURCEFORGE. Honeyview. Disponível em: < http://honeyview.sourceforge.net/>Acesso em: 17 abr. 2020.

HONEYD. Tools. Disponível em: <http://www.honeyd.org/tools.php/>Acesso em: 17 abr. 2020.

IMPERVA. Relatorio de segurança. Disponível em: <http://www.aplidigital.com.br/solucoes/imperva/>Acesso em: 17 mai. 2020.

PYTHONWIKI. HoneyPython. Disponível em: <https://wiki.python.org.br/HoneyPython/>Acesso em: 18 abr. 2020.

VIVAOLINUX. NMAP – A nalise de rede. Disponível em: <https://www.vivaolinux.com.br/artigo/Nmap-30-Exemplos-para-Analises-de-Redes-e-Portas/>Acesso em: 18 abr. 2020.

HACKERTARGET . NMAP – Utilizando Scripts recon. Disponível em: < https://hackertarget.com/7-nmap-nse-scripts-recon/>Acesso em: 20 abr. 2020.

NMAP . Scripts – SMB protocol. Disponível em: < https://nmap.org/nsedoc/scripts/smb-

os-discovery.html/>Acesso em: 25 abr. 2020.

TERMINALROOT. Escaneando redes. Disponível em: <https://terminalroot.com.br/2018/07/escaneado-redes-com-nmap.html/>Acesso em: 27 abr. 2020.

VIVAOLINUX. NMAP – Deteccao de Sistema Operacional com NMAP. Disponível em: <https://www.vivaolinux.com.br/dica/Deteccao-de-Sistema-Operacional-com-NMAP/>Acesso em: 29 abr. 2020.

VIVAOLINUX. Teste de Intrusao com Metasploit. Disponível em: < https://www.vivaolinux.com.br/artigo/Teste-de-Intrusao-com-Metasploit/>Acesso em: 03 Mai. 2020.

METASPLOIT. Running a Vulnerability Scan. Disponível em: <https://metasploit.help.rapid7.com/docs#section-running-a-vulnerability-scan/>Acesso em: 06 Mai. 2020.

W3SCHOOLS. OS Platform Statistics. Disponível em: <https://www.w3schools.com/browsers/browsers_os.asp/>Acesso em: 16 Fev. 2021.

QASTACK. Serviços na porta 135. Disponível em: <https://qastack.com.br/superuser/860583/why-is-port-135-so-overused/>Acesso em: 16 Fev. 2021.

Documento Digitalizado Público

Artigo de Trabalho de Conclusão do Curso

Assunto: Artigo de Trabalho de Conclusão do Curso
Assinado por: Carlos Pagani
Tipo do Documento: Outro
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Carlos Eduardo Pagani**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 19/08/2021 19:09:43.

Este documento foi armazenado no SUAP em 19/08/2021. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifsp.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 748456
Código de Autenticação: ff037e0120