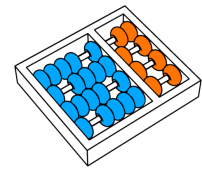


Alexandre Or Cansian Baruque

Visualização de dados de honeypots

CAMPINAS
2015



Universidade Estadual de Campinas
Instituto de Computação

Alexandre Or Cansian Baruque

Visualização de dados de honeypots

Orientador(a): **Prof. Dr. Paulo Licio de Geus**

Co- **Prof. Dr. André Abed Grégio**

Orientador(a):

Relatório para a disciplina MC040 - "Estágio em
Inicição Científica" para o curso de Graduação em Engenharia de
Computação do Instituto de Computação da Universidade Estadual de
Campinas .

Assinatura do Orientador(a)

CAMPINAS

2015

Resumo

O projeto tem como objetivo a implantação de um sistema para o tratamento e visualização de dados coletados por meio de sensores participantes de uma infra-estrutura de coleta de programas maliciosos.

Esse sistema foi desenvolvido fazendo o uso de tecnologias Web para o desenvolvimento de um serviço de visualização dos dados que correlaciona a posição geográfica da origem do potencial ataque com os registros desses ataques e suas respectivas datas e horas das coletas.

O uso funcionalidades integradas no sistema proposto pelo presente projeto permitirá que um analista consiga identificar os padrões e as tendências das atividades maliciosas registradas, tornando possível conclusões acerca dos dados observados.

Sumário

Resumo	iii
1 Introdução	1
2 Atividades desenvolvidas	2
3 Metodologia	3
4 Resultados	6
5 Conclusões	8
6 Conhecimentos adquiridos	9
6.1 Aplicabilidade	9
6.2 Incorporação de novas técnicas	9
6.3 Geração de produtos e processos	9
6.4 Contribuição da participação no projeto para sua formação	10
Referências Bibliográficas	11

Capítulo 1

Introdução

Os programas maliciosos (*malwares*) representam uma das maiores ameaças aos usuários e aos sistemas conectados à Internet.

A facilidade da geração de novos exemplares a partir de um malware já existente, as chamadas variantes, dificulta a atuação dos mecanismos de defesa, tanto na criação de vacinas quanto no processamento das centenas de milhares de variantes em operação. Por exemplo, no caso de antivírus com detecção baseada em assinaturas é geralmente necessário a atuação de um analista humano que descubra um modo de detecção para um dado malware, e então, produza a assinatura em questão. Para tanto, o analista também precisa ter em mãos a variante ainda não detectada.

A fim de obter programas maliciosos e as suas variantes, necessita-se de uma infraestrutura de coleta distribuída. A motivação principal para essa infraestrutura é a de obter e armazenar exemplares dos malwares em circulação no Brasil para análises posteriores melhorando as atuações dos mecanismos de defesa.

Uma vez tendo acesso aos exemplares de malwares armazenados, pretende-se obter informações acerca dos tipos de malwares que atacam as redes brasileiras que possuem sensores de coleta (honeypots)[1].

As informações obtidas desses honeypots de coleta incluem tráfego de rede, registros de auditoria (*logs*) das vulnerabilidades exploradas e tipos de ataques utilizados no comprometimento dos serviços/sistemas e do binário que representa o programa malicioso. De posse de tais informações é possível representá-las por meio de técnicas de visualização e disponibilizá-las por meio de uma arquitetura Web, permitindo a observação imediata de tendências ataques de uma maneira local (em cada rede) e global (conjunto das redes com sensores), além de geração de estatísticas.

Capítulo 2

Atividades desenvolvidas

Para alcançar os objetivos propostos no projeto, as seguintes atividades foram desenvolvidas:

1. Estudo de conceitos sobre visualização de dados através do uso de plataformas de desenvolvimento Web, como Django e Javascript. Em específico, foi estudado extensivamente o uso de bibliotecas para D3js[2] (Data Driven Documents) e Jvectormap.
2. Definição da arquitetura a ser adotada para a implantação da infra-estrutura básica utilizada pelo servidor Web utilizado para a visualização dos dados e em outros sistemas necessários para a administração dos serviços.
3. Instalação das ferramentas e outros programas (e suas dependências) necessários para a implementação de um protótipo em uma máquina desktop provida pelo coorientador.

Ao longo de todas as atividades desenvolvidas foi também feita a documentação do projeto, assim como outros relatórios e atividades requeridas (demonstração, levantamento de dados, etc).

Capítulo 3

Metodologia

No início do projeto foi necessário um estudo preliminar que permitiu obter conhecimentos para tomada de decisões acerca da arquitetura inicial, tais como escolha de ferramentas e plataformas de desenvolvimento a serem utilizadas. Para tanto, observaram-se as ferramentas e os sistemas para visualização de dados de segurança já existentes, entre os quais as disponíveis pelos Projetos Honeynet[6], Norse[9] e DioneaFR[3], visto que os objetivos e as funcionalidades providas por essas ferramentas possuem componentes similares aos propostos no presente projeto. Entretanto, tais ferramentas/sistemas são limitados em certos aspectos, por exemplo: tipo e abrangência de dados, necessidade de compartilhamento de informações remotamente, entre outros, o que gera a necessidade de desenvolvimento de um sistema interno, de acesso controlado e extensível para os requisitos específicos dos dados coletados. Após essa análise inicial e tendo definido as ferramentas a serem utilizadas, o enfoque do estudo foi em adquirir proficiência nas plataformas de desenvolvimento e ferramentas escolhidas para uso no projeto: Django, JQuery, Jvectormap, D3js, GeoIP, iNotify.

Para o controle de versão dos componentes de código produzidos ao longo do projeto foi utilizado o sistema *git*. Tais códigos são armazenados em um repositório privado criado no GitHub através de uma conta de estudante. O repositório consiste de dois ramos: o ramo mestre, no qual se mantém a versão mais atual do projeto com suas funcionalidades mais recentes, e o ramo demonstração, onde fica a última versão estável com o objetivo de ser utilizada para demonstrações funcionais do protótipo do sistema.

O framework Django[4] em linguagem Python foi escolhido para constituir a infraestrutura principal do projeto, tendo como função prover serviços de administração no back-end, assim como servir o código em JavaScript a ser executado pelo cliente em um navegador Web. O Django também permite um fácil gerenciamento da base de dados (sqlite3) através da criação de procedimentos para administração. Foram desenvolvidas rotinas em shell script para consolidar os dados coletados na base de dados por meio dos procedimentos mencionados.

Uma arquitetura REST (Representational State Transfer) foi implantada fazendo

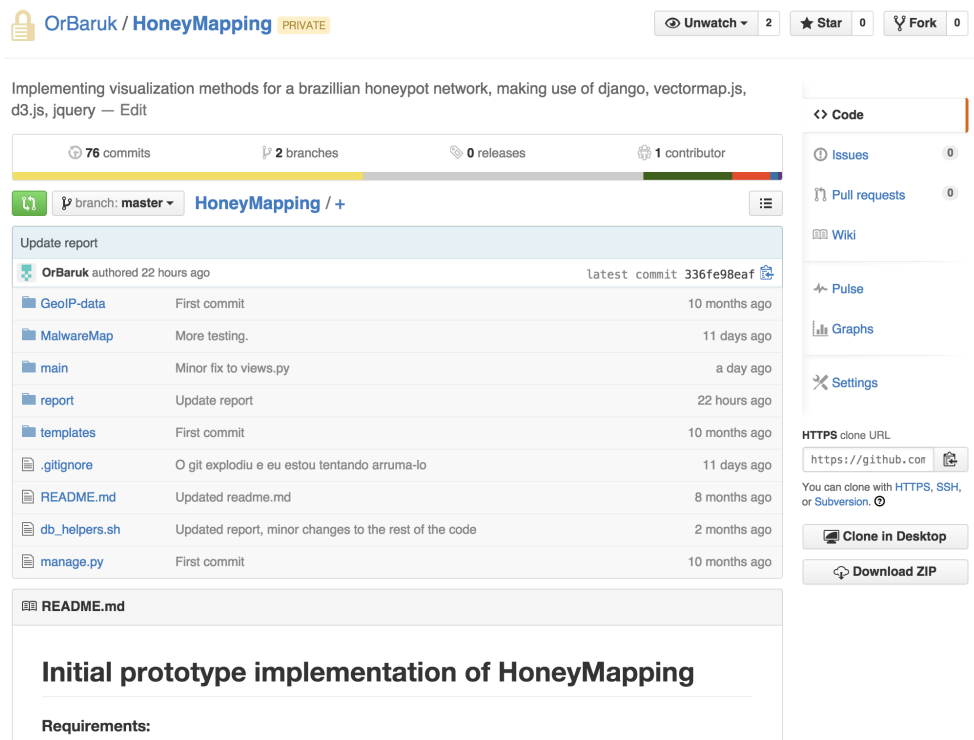


Figura 3.1: Repositório criado no github para controle de versão do código gerado.

uso de funcionalidades do Django, implicando que o acoplamento entre o cliente e o servidor não seja tão rígido, permitindo assim que modificações sejam feitas em ambos os lados sem muito impacto ao restante do sistema. O sistema implementado sobre tal arquitetura realiza consultas na base de dados e serializa os dados para serem encaminhados ao cliente executando código JavaScript no formato JSON.

Para solucionar o problema de relacionar o endereço IP de origem do ataque com sua posição geográfica no mundo real, utilizou-se a ferramenta GeoIP[5], que permite baixar uma base relacional de IPs e latitude/longitude. Com isso, é possível traduzir a maioria dos endereços para localizações no mapa, obtendo também o país e a cidade de origem.

Após obter as localizações no mundo real a partir dos IPs os dados foram inseridos em uma base de dados em sqlite3 utilizando ferramentas do Django para o gerenciamento, o modelo de tabelas utilizado pela base de dados pode ser visualizado na tabela 3.1.

A visualização dos dados pelo cliente foi realizada pelo uso de bibliotecas de código aberto baseados em JavaScript, sendo elas: jVectorMap[8], que permite a fácil criação e modificação de mapas interativos; D3js para a geração de gráficos contendo estatísticas gerais sobre os dados coletados; jQuery[7], que permite a criação de elementos de interface para prover interação com o usuário.

Um protótipo do sistema foi instalado em uma máquina provida pelo coorientador, a qual já continha dados de segurança coletados em um outro projeto (infraestrutura

Tabela 3.1: Exemplo de um log de entrada e o resultado obtido ao inserir as informações na base de dados.

-rw—— - 1 pakkun pakkun 1779 Oct 06 2014 ftpd-6752-115.29.224.200-kGBDlq

Locations				
ip(<i>PK</i>)	latitude	longitude	cityName	countryCode
115.29.224.200	39.93	116.39	Beijing	CN

Sources				
id(<i>PK</i>)	location_id(<i>FK</i>)	port	collector	protocol
6752	115.29.224.200	21	acme	ftpd

Attacks		
key(<i>PK</i>)	dateTime	source_id(<i>FK</i>)
kGBDlq	2014-10-06-18:34:57	6752

distribuída para coleta e análise de dados de honeypots). Nesta máquina, os procedimentos previamente instalados fazem o polling dos dados coletados por sensores distribuídos e os agrega em uma estrutura no sistema de arquivos. Com todos os sistemas em funcionamento, o serviço Web que consome e processa tais dados atualiza sua informações de forma autonoma, provendo a visualização desejada.

Capítulo 4

Resultados

O resultado final foi um protótipo de um sistema cuja ferramenta principal permite a visualização de dados de potenciais ataques a protocolos de rede e aplicações de sistema, bem como de servidores comprometidos realizando o provimento de códigos maliciosos para sensores distribuídos. A visualização desses dados é feita em um mapa, no qual são correlacionadas a origem do ataque (coordenadas geográficas do endereço IP obtido) com a frequência das ocorrências do ataque em questão (raio do círculo impresso em sua coordenada específica).

O mapa permite observar os dados sobre os ataques em uma dimensão temporal, atualizando as informações de acordo com o horário no qual tais dados foram coletados pelos sensores. Dessa forma, pode-se observar as tendências dos tipos de ataques no decorrer do dia, analisando-se os protocolos, serviços e locais de origem. Desenvolveu-se uma imagem em máquina virtual para VirtualBox. A configuração desta máquina virtual já está corretamente aplicada com todos os requisitos para o funcionamento adequado do servidor Web utilizado no provimento da visualização de dados formatados de acordo com os requisitos do projeto. O desenvolvimento da máquina virtual citada tem como objetivo armazenar a ferramenta de uma forma portátil para ser utilizada em demonstrações.

A Figura 4.1 representa um mapa em um dado instante de tempo com dados de potenciais ataques apresentados de acordo com a localização geográfica do endereço IP obtido, bem como outras informações (vide quadro escuro na América do Norte), tais como, a cidade estimada por meio das coordenadas obtidas do IP, o tipo do ataque (serviço alvo) e a porta de rede vulnerável que recebeu o ataque. Logo abaixo do mapa, pode-se notar uma barra de rolagem, a qual movimenta-se de forma automática para mostrar as tendências de ataque ao longo do dia e, mais abaixo, um gráfico que representa os protocolos monitorados e a frequência das conexões recebidas por estes protocolos no instante de tempo em questão.

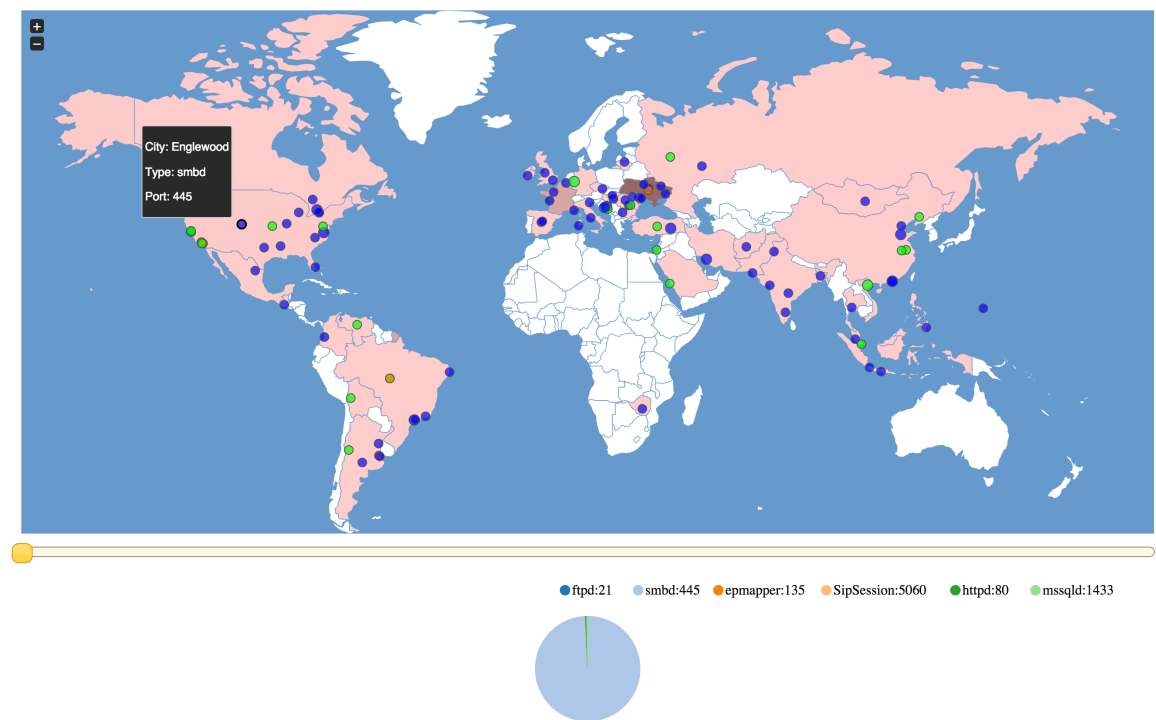


Figura 4.1: Mapa mostrado pela interface web retratando os locais de origem dos ataques.

Capítulo 5

Conclusões

Conforme apresentado na seção anterior, foram obtidos resultados concretos no projeto durante o período da bolsa. Entretanto, existem atividades importantes a serem feitas para os futuros trimestres, como previsto no plano de trabalho, as quais irão permitir a obtenção de informações de mais qualidade do ponto de vista de segurança. As principais propostas dessas atividades consistem do desenvolvimento de procedimentos para o tratamento dos dados obtidos, agregação de outros tipos de dados (provenientes de outros sensores) e a correlação de dados de diferentes endereços de origem e fontes visando a descoberta de ataques em associação.

Além disso, há espaço para melhorias no que se refere a usabilidade da ferramenta, entre as quais incluem-se: uma melhor escolha de cores para a representação dos dados no mapa, melhor disposição do gráfico de protocolos, apresentação de mais informações sobre os potenciais ataques e uma interface mais avançada para a manipulação da dimensão temporal dos dados, permitindo ao analista definir intervalos de tempo para apresentar os dados com o objetivo de identificar padrões com maior facilidade.

Por último, pretende-se buscar uma melhoria na performance do sistema em relação ao tempo de resposta entre a busca dos dados no banco de dados e a renderização dos pontos no mapa. Isto é possível por meio do uso de cache e outras técnicas de otimização disponíveis na plataforma Django, mas tais funcionalidades estavam além do escopo do protótipo desenvolvido até o momento.

Capítulo 6

Conhecimentos adquiridos

6.1 Aplicabilidade

Os conhecimentos adquiridos durante esse projeto possuem uma ampla área de aplicação, sendo esta qualquer área que necessite uma forma visual de interpretar dados. Além disso, a experiência que obtive com a criação de serviços Web e utilização/programação em plataformas e arquiteturas diferentes não possui nenhuma restrição quanto a área de aplicação, podendo ser útil tanto em pesquisas futuras quanto em trabalhos de desenvolvimento.

6.2 Incorporação de novas técnicas

Neste projeto adquiri intimidade com o estilo de programação de APIs no padrão REST, que permite o desenvolvimento de aplicativos Web fracamente acoplados, possibilitando assim a criação de serviços Web escaláveis e capazes de se adaptar a mudança de requisitos de forma flexível. Além disso, familiarizei-me com a plataforma Django, que tem sido muito utilizada em diversos meios, do acadêmico ao industrial, para criação de sistemas Web com diversas características integradas e possibilidade de criação de módulos em Python.

6.3 Geração de produtos e processos

Ao longo do projeto desenvolvi um cliente em JavaScript que permite a visualização de dados não específicos em um mapa de forma genérica, podendo assim ser reutilizado para futuros projetos envolvendo representação de dados nos quais sua posição geográfica possui importância.

Adquiri conhecimento para sobre o processo de geração de imagens em Virtualbox, sendo útil para encapsular sistemas em um ambiente virtual controlado, tornando-o assim portátil. Este processo de criação de imagens executáveis (appliances) serve

para a preparação de demonstrações de qualquer tipo, tornando- as mais confiáveis ao custo de perda de performance, porém permitindo a disponibilização e utilização de maneira facilitada.

6.4 Contribuição da participação no projeto para sua formação

Participar no projeto contribuiu para o meu conhecimento de ferramentas para rápida prototipagem de aplicações e implantação de serviços Web. Tais conhecimentos não são restritos a área de pesquisa, mas sim também para o mercado de trabalho, onde no modelo atual de startups, o tempo para a execução de um projeto é capaz de determinar o futuro da empresa. Aprendi também a estruturar relatórios técnicos de maneira adequada de forma a expor ideias e mostrar resultados.

Referências Bibliográficas

- [1] CERT.br distributed honeypots project. <http://honeytarg.cert.br/honeypots/>. Acessado: outubro/2013.
- [2] D3js. <http://d3js.org/>. Acessado: junho/2015.
- [3] DionaeeFR. <https://rubenespadas.github.io/DionaeeFR/>. Acessado: junho/2015.
- [4] Django. <https://www.djangoproject.com/>. Acessado: junho/2015.
- [5] GeoIP maxmind. <https://dev.maxmind.com/geoip/>. Acessado: junho/2015.
- [6] Honeynet. <http://map.honeynet.org/>. Acessado: junho/2015.
- [7] jQuery. <https://jquery.com/>. Acessado: junho/2015.
- [8] jVectorMap. <http://jvectormap.com/>. Acessado: junho/2015.
- [9] Norse. <http://map.norsecorp.com/>. Acessado: junho/2015.

...