Procura e Análise Automatizada de Superfícies Expostas e Passíveis de Ataque a Partir da Internet

Rúben M. Meneses
António Casimiro
FC/UL
fc37063@alunos.fc.ul.pt
casim@ciencias.ulisboa.pt

José A. Alegria
Paulo T. Serrão
PT Portugal
jose-alegria@telecom.pt
paulo-t-serrao@telecom.pt

I. Introdução e Motivação

Num mundo cada vez mais tecnológico e digital, onde os ataques informáticos têm surgido também eles com maior frequência, é de extrema importância para as empresas fornecedoras de serviços de base tecnológica aumentar o seu nível de segurança contra possíveis ataques.

Estes ataques acontecem frequentemente devido a vulnerabilidades no software usado nestes sistemas, que podem ter diversas origens. Podem resultar de ataques realizados anteriormente com sucesso, ou podem resultar de erros humanos, criados inadvertidamente, e que podem ocorrer, por exemplo, durante a programação do software utilizado ou na configuração de portos que por algum motivo foram abertos e alguém se esqueceu de fechar.

Os resultados associados a este tipo de ataques podem ser devastadores. Se bem realizado, o ataque, pode levar ao roubo de informação confidencial tanto do fornecedor de serviço como dos clientes do mesmo, sendo o resultado mais provável o da perda de clientes.

Existem atualmente soluções que permitem a estes fornecedores de serviço terem controlo sobre possíveis vulnerabilidades deste género. Ferramentas como o Nessus ou o OpenVAS (derivado do Nessus desde que este deixou de ser *open source*) têm capacidade para analisar uma rede inteira em busca deste tipo de problemas e gerar um *feedback* para o cliente com os resultados obtidos.

Contudo, estas soluções têm alguns problemas associados à sua utilização, sendo um dos mais sérios o da interferência no desempenho do sistema alvo. Na realidade, a utilização destas ferramentas pode mesmo provocar a falha das máquinas sobre as quais os testes são executados.

Nesta comunicação apresentamos o sistema HAS (*Hound Attack Surface*), a solução que desenvolvemos para realizar a procura e análise automatizada de superfícies expostas e aumentar o nível de segurança das redes internas da PT Portugal. Apresentamos os elementos principais da solução, descrevendo as opções tomadas para satisfazer os objetivos estabelecidos, nomeadamente para permitir um melhor desempenho com base na distribuição de tarefas, para assegurar a escalabilidade e expansibilidade da solução, para permitir a cobertura de diferentes tipos de vulnerabilidades e para permitir uma fácil integração das soluções com a infraestrutura já existente.

II. Arquitetura do HAS

O HAS é uma ferramenta desenvolvida em Ruby e que recorre a outras ferramentas *open source*. O funcionamento do HAS é baseado na utilização de *plugins*, uma base de dados *NoSQL* à qual está associada uma interface gráfica para visualização de dados e um canal de comunicação que faz a ligação entre todos estes componentes, como podemos observar na Figura 1.

A utilização de *plugins* oferece modularidade à ferramenta. Com esta modularidade a ferramenta torna-se facilmente expansível e escalável, ajustando-se a novas necessidades no que respeita à procura de vulnerabilidades.

Os diferentes *plugins* podem ser categorizados de acordo com as funções que realizam (Figura 1), considerando-se *plugins* de parametrização da ferramenta, de orquestração dos dados entre os diversos *plugins*, de realização de vários *scans* e testes para diferentes vulnerabilidades e ainda de visualização dos dados recolhidos.

A importância do *plugin* de parametrização prende-se com a inicialização da ferramenta. Será este o responsável por tratar e encaminhar o *input* da ferramenta. Este *plugin* encontra-se centralizado numa máquina protegida no interior da rede, com acesso controlado.

O *plugin* de orquestração, por sua vez, é re-

sponsável pela distribuição dos *hosts* pelos diferentes *scans* que lhes serão realizados. Encontra-se centralizado numa zona segura da rede. Contudo pode ser replicado noutras máquina por questões de desempenho ou para garantir que a falha de uma máquina não compromete a execução tarefa da ferramenta.

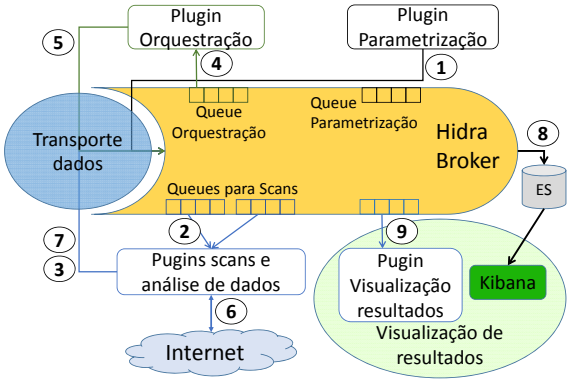

Figure 1: *Arquitetura genérica do HAS*

Os *plugins* de *scans* são os que desempenham o papel mais importante. São estes os responsáveis por obter e analisar toda informação que alimenta a ferramenta. Dadas as suas funções, estes *plugins* encontram-se distribuídos por várias sondas do sistema, para distribuir a carga de trabalho e também para oferecer um melhor desempenho à ferramenta. Estas sondas têm a capacidade de estabelecer ligações quer com a internet quer com a rede interna.

O *plugin* de visualização de dados terá a responsabilidade de gerar um relatório com as informações recolhidas por todos os *scans* realizados pela ferramenta. De acordo com a sua função, este *plugin* encontra-se centralizada numa máquina. Em relação ao transporte de dados, a comunicação entre os diferentes *plugins* é realizada através da utilização de um canal de comunicação denominado "*Hidra Broker*". Este canal recorre à utilização de *queues* e de eventos para a concretização das comunicações, num paradigma *Publish/Subscribe*.

Para o armazenamento dos dados recolhidos utilizamos o ES (*ElasticSearch*), que pode ser visto como uma base de dados *NoSQL*. Para nos ajudar a visualizar esses dados utilizamos o *Kibana*, uma interface gráfica que acede diretamente ao ES e mostra os dados armazenados.

Do ponto de vista funcional, a ferramenta inicia a sua execução através do *plugin* de parametrização (1), responsável por realizar a leitura do ficheiro com os *hosts* a serem analisados, sendo estes *hosts* publicados 1 a 1 para o *Broker*.

Esta informação é colocada numa *queue* do *Broker*, sendo utilizada pelo *plugin* responsável por fazer o *scan* aos portos e serviços, o único que subscreve este canal (2). Após a realização do *scan* a informação obtida é publicada no *Broker* (3) para ser encaminhada para o armazenamento (ES - *ElasticSearch*) (8).

Todos os dados publicados no *Broker* pelo primeiro *plugin* de *scan* são também subscritos pelo *plugin* de orquestração (4). Este *plugin* é responsável por determinar novos *scans* específicos que têm de ser feitos, com base, por exemplo, nos portos abertos identificados no *scan* inicial. O orquestrador envia então dados referentes a cada *host* para cada *plugin* responsável por realizar um *scan* específico (5).

Após realizarem *scans*, desta vez mais específicos, a determinados hosts (6), os plugins enviam os resultados obtidos novamente para o armazenamento dos dados (8), através do *Broker* (7).

Concluídos todos os *scans* necessários, é então executado o *plugin* de visualização de resultados (9). Este *plugin* é responsável por gerar um relatório num formato legível por uma folha de cálculo, com duas tabelas. Uma das tabelas contém informação sobre os portos abertos, a contagem atual de portos abertos por *host*, e a contagem anteriormente realizada. Na outra tabela temos informação sobre os testes mais específicos que foram realizados, como por exemplo informação quanto à vulnerabilidade ao ataque *Heartbleed*, a ataques de *SQL injection*, ou informação sobre certificados expirados, entre outros.

III. Resultados

Esta ferramenta pressupõe um desenvolvimento contínuo de funcionalidades adicionais, sob a forma de *plugins*, tendo sido concretizados os mecanismos básicos de inicialização da ferramenta e de pesquisas por vulnerabilidades. A avaliação inicial da ferramente permite-nos afirmar que é possível distribuir a carga de trabalho ao executar os *plugins* de *scans* de forma paralela, explorando o canal de comunicação (*Broker*) para realizar as diversas troca de informação entre os *plugins* e para armazenar os dados. Os testes iniciais que realizámos mostram também que a capacidade de deteção das vulnerabilidades testadas é efetivo, como seria de esperar. Para tal, foram configuradas máquinas em que se injetaram as vulnerabilidades procuradas (nomeadamente portos abertos, vulnerabilidade ao ataque Heartbleed, vulnerabilidade a ataques de SQL injection), as quais foram corretamente assinaladas.

O *plugin* de visualização dos dados encontra-se em desenvolvimento e, como trabalho futuro, pretende-se adicionar novos *plugins* de forma articulada com as necessidades, de uma forma que será ágil e eficaz.