12 FEPEG
FÓRUM
ENSINO · PESQUISA · EXTENSÃO · GESTÃO
CIÊNCIA E TECNOLOGIA: IMPLICAÇÕES NO ENSINO, PESQUISA E EXTENSÃO
REALIZAÇÃO: Unimontes PIBID Unimontes
APOIO: FAPEMIG FADENOR
ISSN: 1806-549X

USO DAS FERRAMENTAS SQLMAP E NIKTO PARA TESTAR SEGURANÇA DE DADOS EM SITES

Autores: ROGER MATHEWS ARRUDA SANTOS, ANDERSON ALVES DE SOUZA, INGRID RAUANY DIAS SOARES, MARCELLA SOUZA DAYRELL, RAFAEL ANTONIO GONÇALVES LIMA, VICTOR DE FREITAS ARRUDA

Introdução

A tecnologia faz parte da rotina das pessoas. A troca de informações entre os aparelhos e as aplicações exigem que os usuários cadastrem seus dados em serviços diversos. Atualmente, nossos dados são cadastrados e estão à disposição de inúmeras empresas, por exemplo, as informações que prestamos às redes sociais, e-mails, serviços bancários e a uma infinidade de aplicativos que fazemos uso.

Desta maneira, é notável que as informações que fornecemos se tornaram significativas e preciosas, criando nos usuários e nas empresas a preocupação e questionamentos sobre como se dá o meio de armazenamento, como está sendo realizada a comunicação entre aparelhos e aplicações, sobre a infraestrutura e a forma como as pessoas utilizam os sistemas.

É indiscutível que se necessita constantemente de implementação de medidas de segurança, dificultando o extravio de informações que podem ser utilizadas para outros fins que não o pretendido, além de que podem ser modificadas e destruídas, ocasionando prejuízo para ambas as partes.

Este trabalho objetiva alertar sobre falhas de segurança em sites e apresentar o funcionamento das ferramentas Nikto e SQLMap. A finalidade é despertar nos desenvolvedores atenção e preocupação sobre a segurança das informações quando desenvolvem, realizando testes em busca de possíveis falhas e imperfeições no código-fonte que possam ser exploradas por hackers.

Nikto

Nikto é uma ferramenta para avaliação de servidor *web*, de forma que examina um servidor *web* para encontrar potenciais problemas e vulnerabilidades de segurança, por exemplo: configurações incorretas de servidor e *software* , arquivos e programas padrões, arquivos e programas inseguros, servidores e programas desatualizados. O Nikto é um bom scanner para obter informações iniciais sobre o alvo, muitas vezes ele passa informações úteis para a invasão, porém ele não passa todas as informações nem todos os pontos de invasão.

Nikto é construído em *LibWhisker2* (por RFP) e pode rodar em qualquer plataforma que tem um ambiente Perl. Suporta SSL, *proxies*, autenticação de host, codificação de ataque e muito mais. Ele pode ser atualizado automaticamente a partir da linha de comando e suporta o envio opcional de dados de versão, atualizados para os mantenedores.

A maioria das ferramentas de segurança da *web* (incluindo o Nikto 1.32 e posteriores) depende muito da resposta HTTP para determinar se existe uma página ou *script* no destino. Como muitos servidores não aderem adequadamente aos padrões RFC e retornam uma resposta "OK" para solicitações que não são encontradas ou proibidas, isso pode levar a muitos resultados falso-positivos. Além disso, as respostas de erro para várias extensões de arquivo podem diferir - a resposta "não encontrada" para um arquivo .html geralmente é diferente de um arquivo .cgi.

A partir da versão 2.0, o Nikto não mais assume que as páginas de erro para diferentes tipos de arquivo serão as mesmas. Uma lista de extensões de arquivo exclusivas é gerada em tempo de execução (do banco de dados de teste) e cada uma dessas extensões é testada no destino. Para cada tipo de arquivo, o "melhor método" de determinação de erros é encontrado: resposta RFC padrão, correspondência de conteúdo ou hash MD4 (em ordem decrescente de preferência). Isso permite que o Nikto use o método mais rápido e mais preciso para cada tipo de arquivo individual e, portanto, ajude a eliminar os falsos positivos vistos em alguns servidores na versão 1.32 e abaixo.

ISSN: 1806-549X

SQLMap

O SQLMap é uma ferramenta de teste de penetração de código aberto, que automatiza o processo de detecção e exploração de falhas de injeção SQL. Com essa ferramenta é possível assumir total controle de servidores de banco de dados em páginas *web* vulneráveis, inclusive de base de dados fora do sistema invadido. Ele possui um motor de detecção poderoso, empregando as últimas e mais devastadoras técnicas de teste de penetração por SQL *Injection* , que permite acessar a base de dados, o sistema de arquivos subjacente e executar comandos no sistema operacional.

Ele suporta uma ampla gama de servidores de banco de dados, onde a maioria dos servidores de banco de dados mais populares já estão inclusos. Também suporta vários tipos de ataques de SQL *injection*, incluindo *boolean-based blind*, *time-based blind*, *error-based*, UNION *query-based*, *stacked queries* e *out-of-band*.

Uma boa característica da ferramenta é que ela vem com um sistema de reconhecimento de *hash* embutido por senha. Ele ajuda a identificar o hash da senha e, em seguida, decifrar a senha, verificando um dicionário.

Depois de conectar a um servidor de banco de dados, esta ferramenta também permite que você pesquise por um banco de dados específico, tabelas ou colunas específicas no servidor de banco de dados inteiro. Esta é uma característica útil quando você quer pesquisar por uma coluna específica, mas o servidor de banco de dados é em grade e contém muitos bancos de dados e tabelas.

Material e métodos

Para o desenvolvimento do trabalho foram selecionados 5 sites encontrados em buscas no google, são eles: Home of Acunetix Art (http://testphp.vulnweb.com/), Site 1, Site 2, Site 3 e 4.

Para a análise foram utilizadas as ferramentas Nikto 2.1.5 e SQLMap 1.2.3.50#dev já descritas nas seções anteriores. Para utilizar o Nikto em cada um dos sites foi utilizado o comando “perl nikto.pl -C all -h host -o resultado.html”, onde *host* é o endereço do site escaneado e resultado.html é o resultado do escaneamento em formato HTML. O relatório apresenta as possíveis vulnerabilidades de acordo com o endereço do site que foi utilizado. Após a utilização do Nikto, procurou-se no site alguma URL que coincidisse com o padrão imposto pelo SQLMap e o Havij para a execução de seus testes e, assim, pode-se saber qual era a vulnerabilidade SQL que o site tinha e ter acesso a todo o banco de dados. Em alguns dos sites foi possível efetuar o login do usuário, mas a maioria não continha dados comprometedores.

Resultados e discussão

*A. **Nikto***

Home of Acunetix Art

O servidor utilizado é nginx/1.4.1. Através do cabeçalho, aparentemente utiliza-se PHP/5.3.10-1 e de acordo com a análise o cabeçalho anti-clickjacking X-Frame-Options não está presente; servidor vaza inodes via ETags; cabeçalho encontrado com o arquivo ‘/clientaccesspolicy.xml’, que contém uma entrada completa de *wildcard*; ‘/crossdomain.xml’ contém uma entrada completa de *wildcard* e contém 0 linhas que devem ser exibidas manualmente para domínios ou *wildcards* impróprios. Em ‘/index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000’, o PHP revela informações potencialmente confidenciais através de certas solicitações HTTP que contêm *strings* QUERY específicas. Diretórios de indexação encontrados: ‘/admin/’, ‘/images’, ‘/login.php’, ‘/images/?pattern=/etc/*&sort=name’ e ‘/CVS/Entries’, que contém arquivo de entradas CVS com informações de listagem de diretórios.

Site 1

ISSN: 1806-549X

O servidor utilizado é o Apache. De acordo com a análise o Cookie PHPSESSID criado sem a bandeira httponly; o cabeçalho anti-clickjacking X-Frame-Options não está presente. Cabeçalho incomum 'x-cache-status' encontrado, com conteúdo: BYPASS; DEBUG HTTP verb pode mostrar informações de debug do servidor. Diretórios de indexação encontrados: '/config/' onde configuração da informação pode estar disponível remotamente, '/cgi-bin/' onde provavelmente um sistema shell seria encontrado, '/img/', '/login/', '/images/?pattern=/etc/*&sort=name'.

Site 2

O servidor utilizado é o Microsoft-IIS/8.5. Através do cabeçalho, aparentemente utiliza-se ASP.NET e de acordo com a análise o cabeçalho anti-clickjacking X-Frame-Options não está presente; cabeçalho incomum 'x-powered-by-plesk' encontrado, com conteúdo: PleskWin; cabeçalho encontrado x-aspnet-version: 2.0.50727; servidor vaza inodes via ETags; cabeçalho encontrado com o arquivo '/'. Métodos HTTP permitidos: OPTIONS, TRACE, GET, HEAD, POST. Métodos HTTP públicos: OPTIONS, TRACE, GET, HEAD, POST; DEBUG HTTP verb pode mostrar informações de debug do servidor.

Site 3

O servidor utilizado é o Apache/2.2.31. De acordo com a análise o cabeçalho anti-clickjacking X-Frame-Options não está presente; mod_ssl/2.2.31 parece estar desatualizado (current é pelo menos: 2.8.31 e pode depender da versão do servidor); mod_ssl / 2.2.31 OpenSSL / 1.0.1e-fips mod_bwlimited / 1.4 - mod_ssl 2.8.7 e inferior são vulneráveis a um estouro de buffer remoto que pode permitir um shell remoto (difícil de explorar). CVE-2002-0082, OSVDB-756.

Site 4

O servidor utilizado é o Apache. Através do cabeçalho, aparentemente utiliza-se PHP/5.2.17 e de acordo com a análise o cabeçalho anti-clickjacking X-Frame-Options não está presente; servidor vaza inodes via ETags; cabeçalho encontrado com o arquivo /; Métodos HTTP permitidos: HEAD, GET, POST, OPTIONS; /db.php: poderia ser interessante... tem sido visto em logs web de um scanner desconhecido; /icons/: diretório de indexação encontrado; /icons/README: arquivo padrão Apache encontrado.

B. ***SQLMap***

Home of Acunetix Art

O ponto de invasão encontrado foi o http://testphp.vulnweb.com/listproducts.php?cat=1, onde foram encontradas as vulnerabilidades do tipo *boolean-based blind*, *error-based*, AND/OR *time-based blind* e UNION *query*. Os bancos de dados presente foi o 'acuart', que possui 8 tabelas.

Site 1

Foi encontrado ponto de invasão, onde foram encontradas as vulnerabilidades do tipo *boolean-based blind*, AND/OR *time-based blind*, e UNION *query*. Os bancos de dados presente foi: 'mgerais_3' que tem 22 tabelas.

Site 2

Foi encontrado ponto de invasão, onde foram encontradas as vulnerabilidades do tipo *boolean-based blind* , AND/OR time-based blind e UNION query. Os bancos de dados presentes foram 'feeder_feeder' que tem 12 tabelas.

Site 3

Foi encontrado ponto de invasão, onde foram encontradas as vulnerabilidades do tipo *boolean-based blind*, *error-based* , AND/OR *time-based blind* e UNION *query*. Os bancos de dados presentes foram 'russians_russians', 'russians_russians_DE' e 'russians_russians_RU' que possuem 36 tabelas.

Site 4

ISSN: 1806-549X

Foi encontrado ponto de invasão, onde foram encontradas as vulnerabilidades do tipo boolean-based blind, error-based, AND/OR time-based blind e UNION query. Os bancos de dados presentes foram: ‘qoe850ek_boxoffice’ que possui 5 tabelas, ‘qoe850ek_boxoffice_labs’ que possui 7 tabelas, ‘qoe850ek_boxoffice_labs2’ e ‘test’.

Considerações finais

A utilização de ferramentas para verificação de vulnerabilidades ajuda a identificar e prevenir problemas, possibilitando que falhas de segurança sejam corrigidas. Durante a utilização do *SqlMap*, as vulnerabilidades mais encontradas foram: *boolean-based blind*, *error-based*, AND/OR *time-based blind* e UNION *query*.

Este trabalho possui cunho acadêmico, por esta razão foram ocultados os nomes e URLs dos sites utilizados para testes. Os testes executados neste artigo demonstram as formas mais comuns para explorar vulnerabilidades de *SQL Injection* (por meio do *SqlMap*) e vasculhar *websites* em busca de arquivos e configurações que podem ser passivos de um ataque através do Nikto. Estes ataques podem ser prevenidos com: validação de dados digitados pelo usuário, criação de usuários com permissões adequadas, não retornar mensagens do servidor SQL para o usuário, remover objetos não utilizados e habilitação de log de segurança no servidor.

Referências bibliográficas

NIKTO v2.1.5 - The manual. Disponível em: <https://cirt.net/nikto2-docs/>. Acesso em: 04 jun. 2018.

REGO, Jalmacy; GONDIM, Vanderley; FERREIRA, Almedson. **Nikto** : Uma Ferramenta Open Source para Análise de Vulnerabilidades em Servidores Web. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/freerbase/2015/004.pdf>. Acesso em: 04 jun. 2018.

SQLMAP: Automatic SQL injection and database takeover tool. Disponível em: <http://sqlmap.org/>. Acesso em: 04 jun. 2018.

SANTIN, Felipe; FIGUEIREDO, José Antônio Oliveira; MACHADO, Vanessa Lago. **Uso da ferramenta sqlMap para detecção de vulnerabilidades de SQL Injection**. Disponível em: <http://eati.info/eati/2017/assets/anais/Longos/L31.pdf>. Acesso em: 04 jun. 2018.