

Roteiro 3

Rodolfo Avelino e João Eduardo

Objetivo: Criação de extensão para navegadores Firefox para detecção de ataques e violação de privacidade em cliente web.

Descrição: Desenvolver uma ferramenta capaz de detectar e apresentar:

- Conexões a domínios de terceira parte em uma navegação web;
- A quantidade de cookies injetados no carregamento de uma página;
- Detectar o armazenamento de dados (storage local) no cliente por intermédio do html5.
- Detectar sincronismo de cookies.

A insegurança dos Browsers

O navegador web, ou Browser, é o processo que fica a mais tempo em execução em um Sistema Operacional, seja por motivos pessoais ou profissionais. Além deste detalhe os navegadores mais utilizados hoje são multiplataforma. Estas duas características chamam a atenção para que fraudadores busquem explorar vulnerabilidades de sistemas no elo mais fraco do ecossistema da tecnologia da informação, o usuário final.

Os browsers quando foram lançados exibiam apenas textos (ASCII e HTML) e imagens (GIF e JPEG). Devido às limitações da linguagem HTML (Hyper-Text Markup Language), logo surgiram alternativas para ampliar a capacidade do browser para torná-lo mais dinâmico e interativo. Uma das primeiras iniciativas neste sentido foi a adoção dos helpers (aplicativos auxiliares) que evoluíram para os atuais plug-ins¹.

1 http://www.training.com.br/lpmaia/pub_seg_browser.htm

Extensões de navegadores ou plug-ins, são programas executáveis que permitem ao browser executar arquivos em formatos diferentes de HTML, como áudio, animações e acesso a banco de dados. Um de seus pontos fracos é que sua confiança e segurança está depositada em quem o desenvolveu. O fato da extensão de navegador estar incorporada ao “core” do browser, ele utiliza o mesmo PID (Process ID), tendo assim acesso irrestrito ao seu sistema, comprometendo assim seu ambiente.

Com a introdução de linguagens/tecnologias como Java, JavaScript, VBscript e ActiveX, o browser passou a receber código ativo pela rede, ou seja, o simples fato de se visitar um site na Internet já é o suficiente para receber um código malicioso que possa ter acesso a qualquer parte do seu sistema (processador, memória, discos e rede) e, até mesmo, desligar sua estação².

Existem inúmeros motivos para uma aplicação client-server executar operações no lado cliente, tal como necessidade de acesso à informações ou funcionalidades que estão disponíveis em sua estação e não no servidor, poupar poder de processamento no servidor.

Garantir a segurança em clientes web é muito mais complexo do que no web server. Essa complexidade se dá devido a impossibilidade dos administradores do sistema gerenciarem de forma efetiva a segurança do dispositivo do usuários.

Browser hijacking

É a modificação não autorizada das configurações de um browser por um código malicioso, com o objetivo de ter acesso aos recursos do sistema ou também inserir anúncios indesejados. É muito comum que a instalação de plugins não confiáveis possa causar ações indesejadas pelo browser. Algumas são instaladas sem o consentimento do usuário.

2 http://www.training.com.br/lpmaia/pub_seg_browser.htm

Outra forma de sequestro de navegadores é por meio da exploração de uma falha de cross site scripting (XSS) em uma aplicação web. A falha de XSS pode permitir que um invasor utilize por exemplo o framework BeEF para injetar um código javascript dentro de uma página web vulnerável. A partir de sua execução no navegador, o invasor poderá enviar comandos ao cliente infectado.

Entregáveis

Desenvolver um plugin para navegador capaz de detectar e apresentar:

- As conexões a domínios de terceira parte em uma navegação web (2,5 pontos);
- Potenciais ameaças de sequestro de navegador (hijacking e hook) (1 ponto);
- Detectar o armazenamento de dados (storage local – html5) no dispositivo do usuário (2,5 pontos);
- A quantidade de cookies e supercookies injetados no carregamento de uma página (se possível diferencie em cookies de primeira e terceira parte, bem como sessão ou navegação) (1 ponto);
- Detecção de Canvas fingerprint (1 ponto).
- Criar uma pontuação a partir de uma metodologia (o grupo pode determinar), indicando se a página respeita a privacidade do usuário (2 pontos).

Indicação para pesquisa:

https://developer.mozilla.org/pt-BR/docs/Mozilla/Add-ons/WebExtensions/sua_primeira_WebExtension

<https://developer.mozilla.org/pt-BR/Add-ons/WebExtensions/Passo-a-Passo>

<https://www.ibm.com/developerworks/br/library/os-extendchrome/index.html>

<https://addons.mozilla.org/en-US/firefox/user/miraculix200/>

<https://github.com/Miraculix200/StoragErazor>

<https://fingerprintable.org>

<https://addons.mozilla.org/pt-BR/firefox/addon/clear-storage-button/?src=api>

Proposta 3 - Relatório Análise de vulnerabilidade

Objetivo: Gerar um relatório de análise de vulnerabilidade em um ambiente real.

Descrição:

- Consiga a autorização para realizar o teste em um domínio publicado na Internet;
- Gere um relatório completo com análise do domínio.

Conceitos

Proposta 1

C+ - Ambiente com VPN, Wordpress e servidor SSH configurado

B+ - Servidor Zabbix configurado gerenciando serviço HTTP e Banco de dados. Deverá estar configurado o chat bot Telegram para envio dos alertas.

A+ - Módulo WAF executando e configurado para os ataques OWASP Top 10 (não pode ser plugin de wordpress)

Proposta 2

C+ - plugin apresentando a quantidade de cookies, web storage e conexões para domínios de terceira parte.

B+ - Plugin identificando sincronismo de cookies

A+ - Plugin classificando cookies e web storages de sessão e persistentes.

Proposta 3

C+ - Relatório apresentando as vulnerabilidades, suas classificações, evidências, recomendações para correção e referências sobre elas.

B+ - Relatório com resumo executivo, metodologia aplicada, referências para testes, Apresentação dos testes e conclusão.

A+ - Encontrar pelo menos uma vulnerabilidade que não tenha sido reconhecida por script ou ferramentas automatizadas.