Curso de Extensão em Tecnologias Microsoft

INF0998 Programação segura (segurança de software)

Atividades práticas de testes de segurança com Burp Suite Community

Testes de segurança manuais são demorados e, quando realizados em grande quantidade, podem causar atrasos em projetos de desenvolvimento de software, ou mesmo não estarem disponíveis em tempo hábil para que correções sejam efetuadas.

Testes de segurança auxiliados por ferramentas aceleram a geração de resultados que podem ser tratados rapidamente. Porém, vale lembrar que ferramentas automáticas de verificações de segurança não substituem o testador experiente.

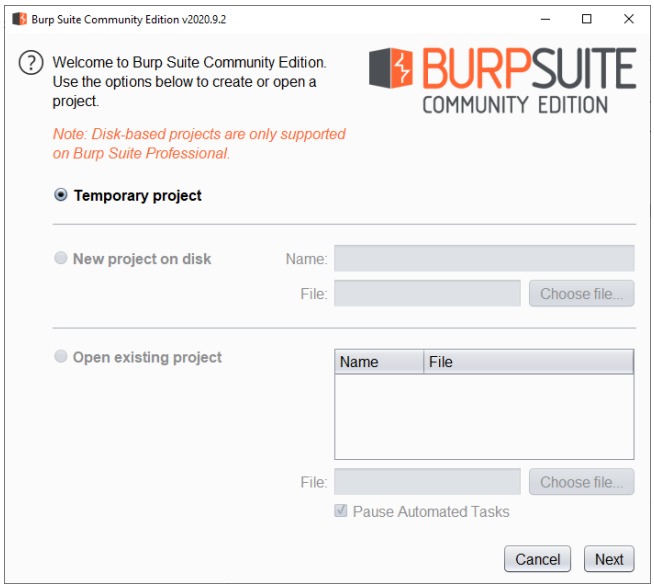
Esta aula usa a ferramenta de testes de intrusão ***Burp Suite Community*** para auxiliar a descoberta e a exploração de vulnerabilidades presentes nas aplicações estudadas nas aulas anteriores.

Burp Suite Community Edition

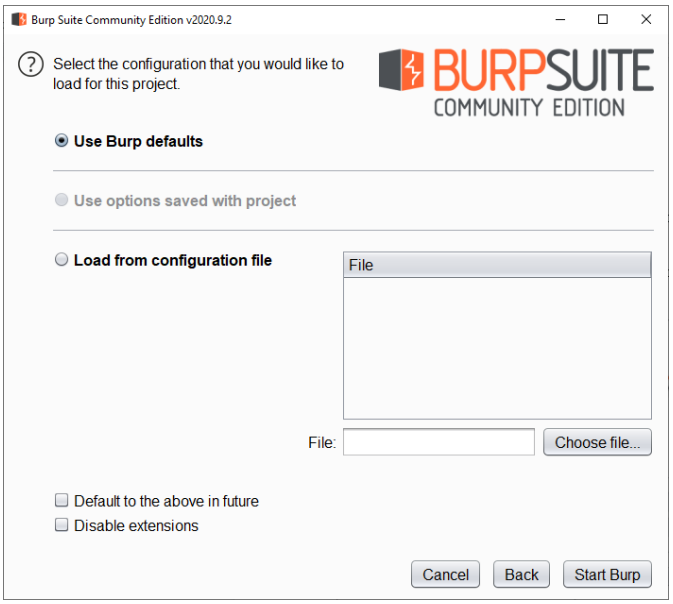
Apresentando o Burp Suite https://portswigger.net/burp

Configurações iniciais (Target, Site map, Scope)

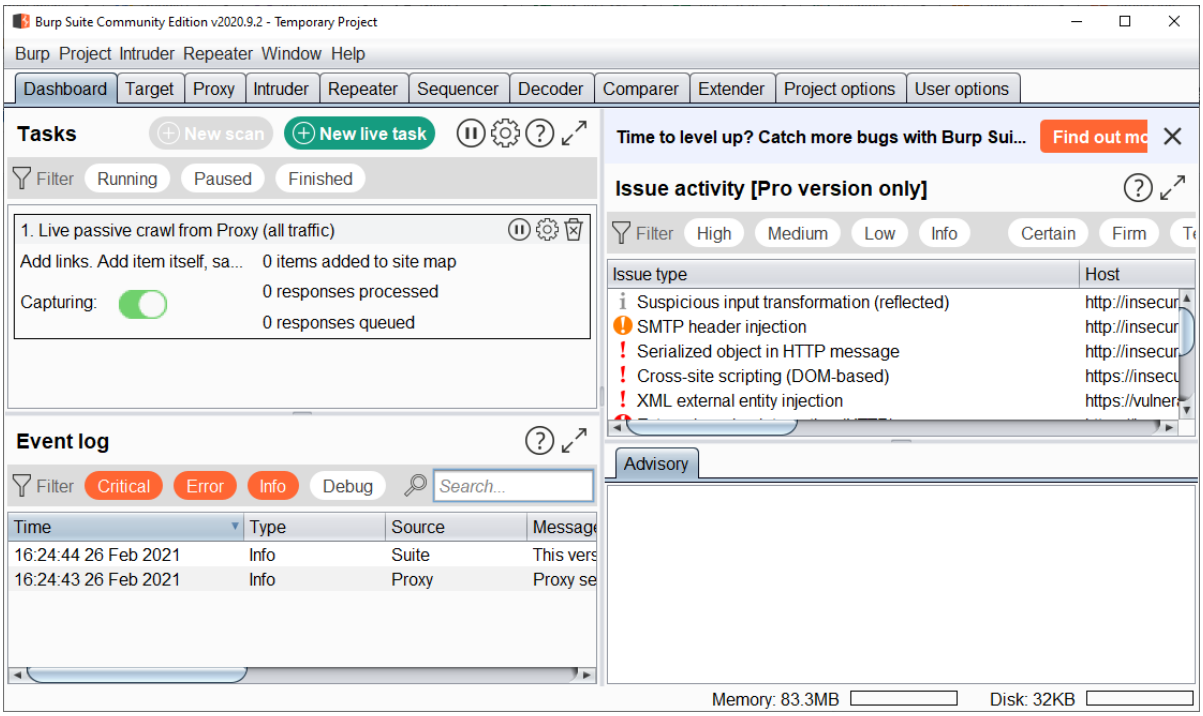
- A ferramenta Burb Suite Community Edition pode ser baixada no link https://portswigger.net/burp/communitydownload.
- Baixar a versão apropriada para o sistema operacional e que seja a mais recente e estável.
- Seguir o processo de instalação conforme instruções do programa.
- Ativar a ferramenta Burp.



- Burp Community Edition só aceita projetos temporários, Clicar no botão ***Next***.



- Usar as opções default. Clicar no botão ***Start Burp***.



- Observar que nada acontece na aba (ou guia) ***Target***.
- Ir para a aba ***Proxy***.
- Burp possui um browser embarcado e configurado como proxy. Clicar no botão ***Open browser***.
- Na guia ***Proxy*** → ***Intercept***, desativar o controle ***Intercept is on*** (de ***on*** para ***off***)
- Na janela do browser, digitar a URL ou IP do bWAPP (que deve estar ativado).
    - Selecionar a aplicação bWAPP, logar na aplicação com usuário/senha bee/bug.
- No Burp, guia ***Target*** → ***Site map***, abrir a pasta (🗀) bWAAP e a engrenagem (⚙) login.php
    - Observar que as informações de login e senha estão expostas!
    - Clicar com botão direito em 🗀 bWAPP e selecionar a opção ***Add to scope*** e em seguida apertar o botão ***Yes***.
- Clicar na barra Filter e selecionar a opção ***Show only in-scope items.***

Opções de proxy

- Acessar a aba ***Proxy*** → ***Options***.
    - Na seção ***Intercept Client Requests***, desativar a regra ***File extension Does not match***.
    - Na seção ***Intercept Server Responses***, ativar as regras de interceptação e ativar a regra ***Content type header Matches text.***
    - Na guia ***Proxy*** → ***Intercept***, ativar o controle ***Intercept is on*** (de ***off*** para ***on***).
- Testar o comportamento destas modificações da interceptação da bWAPP.

- Em ***Proxy*** → ***Options,*** na seção ***Response Modification***, ativar a opção ***Unhide hidden form fields*** e a opção ***highlight unhidden fields.***
- Na guia ***Proxy*** → ***Intercept***, desativar o controle ***Intercept is on*** (de ***on*** para ***off***).

- Testar o comportamento destas modificações da interceptação da bWAPP.
    - Selecionar o bug "***Insecure DOR (Change Secret)"*** e clicar no botão ***Hack.***
    - Um campo hidden aparecerá na janela do browser.

Repeater

Repeater é uma ferramenta muito útil em testes manuais. Ele permite a manipulação (alteração) de parâmetros das requisições individuais para a repetição de uma requisição com parâmetros personalizados pelo testador.
Para usar o Repeater:

- Na aba ***Proxy*→ *HTTP History***, localizar a requisição da página de login da aplicação bWAPP.
- Clicar na requisição com o botão direito do mouse e selecionar a opção ***Send to Repeater***.
- O rótulo da aba ***Repeater*** vai mudar de cor para vermelho, indicando que foi ativada.
- A aba ***Repeater*** mostra a requisição que será repetida.
- Ativar o botão Send para enviar a requisição como está.
- Visualizar a resposta obtida. (Spoiler, o login foi bem-sucedido.)
- Editar o campo password=bug para password=bag e ativar o botão Send.
- Visualizar a resposta obtida, o botão Render ajuda na visualização HTML. (Spoiler, o login foi mal-sucedido.)

Intruder

Intruder é uma ferramenta muito útil em testes de força bruta. Ele permite a manipulação (alteração) de parâmetros das requisições e o envio de diversas requisições personalizadas em massa.
Para usar o *Intruder*:

- Na aba ***Proxy*→ *HTTP History***, localizar a requisição da página de login da aplicação bWAPP.
- Clicar na requisição com o botão direito do mouse e selecionar a opção ***Send to Intruder***.
- O rótulo da aba ***Intruder*** vai mudar de cor para vermelho, indicando que foi ativado.
- Na aba ***Intruder*** → ***Positions***, manter attack type em Sniper.
    - Observar todos os campos que podem ser manipulados. Clicar no botão ***Clear §***.
    - Selecionar o parâmetro bug, clicar no botão ***ADD §***, o parâmetro fica assim: ***§bug§***.
- Na aba ***Intruder*** → ***Payloads***, manter o ***Payload type*** em ***Simple list***.
    - Na caixa de *Payload options*, adicionar (ADD) os itens da lista manualmente.
    - Digitar cada item da lista bag, beg, big, bog, bug seguido de enter.
    - bag <ENTER>beg<ENTER>big<ENTER>bog<ENTER>bug
    - Clicar no botão ***Start Attack*** no canto superior direito da janela do Burp.
    - O progresso do ataque é mostrado em outra janela.

Exploração de HTML Injection com Burb Suite

Ainda no bWAPP em proxy pelo Burb.

- Selecionar o bug ***HTML Injection - Reflected (POST)***.
- Preencher os campos de nomes e sobrenome e clicar em ***Go!***,
    - p.ex., preencher com “Tony” e “Stark”.
    - O conteúdo fornecido pelo usuário é mostrado no navegador.
- No Burp, ***Proxy*→ *Intercept***, ativar a Interceptação (***Intercept is on***).
- No browser, preencher de novo com “Tony” e “Stark”. Clicar em ***Go***.
- No Burp, verificar que a requisição interceptada
    - Observar os parâmetros `firstname=Tony&lastname=Stark&form=submit`.
    - Substituir a linha de parâmetros por
        - `firstname=<h1><a href="http://www.google.com">Clique aqui!</a></h1>&lastname=<h2>Hahah!</h2>&form=submit`
    - Clicar no botão Forward enquanto houver requisições para tratar.
- No Browser, observa-se que o HTML foi inserido na página e apresentado ao usuário.

Outra injeção de HTML via GET

- Selecionar o bug ***HTML Injection - Reflected (GET)***.
- Preencher os campos de nomes e sobrenome e clicar em ***Go!***,
    - p.ex., preencher com “Tony” e “Stark”.
    - O conteúdo fornecido pelo usuário é mostrado no navegador e na URL também!
- No Burp, ***Proxy*→ *Intercept***, ativar a Interceptação (***Intercept is on***).
- No browser, preencher de novo com “Tony” e “Stark”. Clicar em ***Go***.
- No Burp, verificar que a requisição interceptada
    - Observar os parâmetros `firstname=Tony&lastname=Stark&form=submit`.
    - Substituir a linha de parâmetros por
        - `firstname=<h1><a href="http://www.google.com">Clique aqui</a></h1>&lastname=<h2>Hahah!</h2>&form=submit`
    - Clicar no botão Forward enquanto houver requisições para tratar.
- No Browser, observa-se que o HTML foi inserido na página e apresentado ao usuário.

Outra injeção de HTML via GET

- Selecionar o bug ***HTML Injection - Reflected (URL)***.
    - Não esquecer de clicar em ***Hack!***
    - A página mostra a URL atual (Current URL)
- No Burp, ***Proxy*→ *Intercept***, ativar a Interceptação (***Intercept is on***).
- No browser, recarregar a página (F5 ou botão de recarregar no navegador).
- No Burp, verificar que a requisição interceptada
    - Substituir o endereço IP do header Host por algum outro endereço.
        - P.ex., www.google.com
    - Clicar no botão Forward enquanto houver requisições para tratar.
- No Browser, observa-se que a URL atual (Current URL) foi adulterada.

Exploração de XSS JSON auxiliada pelo Burb Suite

Ainda com o bWAPP em proxy pelo Burb.

- Selecionar o bug ***XSS - Reflected (JSON)***.
- Testar a busca com qualquer nome de filme, p.ex., “Matrix”.
    - Observar que o conteúdo do campo de busca volta para o usuário;
        - Matrix???
- No Burb, aba ***Proxy*** → ***HTTP History***, identificar a última requisição (Request) que contém a busca “Matrix”.
    - Observar o conteúdo da resposta (Response). Onde está a palavra “Matrix”?
        - Dica: Usar a função de busca na parte inferior da baixa da Response.
    - A palavra buscada (“Matrix”) está inserida entre tags `<script> … </script>`
        - O JSON de resposta é construído no trecho.
        - Um Eval vulnerável também está exposto.
- O ataque consiste em customizar um JSON de resposta com um script embutido.
    - Finalizar o JSON, fechar o script e inserir o script malicioso.
    - Observar de onde o JSON é finalizado e copiar os caracteres finais
        - Dica: a linha do JSON termina com `"}]}'`;
    - Fechar o script do JSON e inserir o script malicioso
        - `"}]}'; </script><script>alert(1);</script>`
    - Outro exemplo de script malicioso
        - `"}]}'; </script><script>alert(document.cookie);</script>`

Exploração de XSS AJAX/JSON com Burp Suite

Ainda no bWAPP em proxy pelo Burb.

- Selecionar o bug ***XSS - Reflected (AJAX/JSON)***.
- Testar a busca com qualquer nome de filme, p.ex., fazer uma busca por “iron man”.
- Nesse formulário, não há botão de Submit. E agora?!
    - A busca é feita a medida em que o campo é preenchido.
    - Isso indica alguma tecnologia responsiva. Neste caso, o AJAX.
- No Burp, visualizar as sequência de requisições assíncronas AJAX. Selecionar a última.
    - Verificar que o código 200 na resposta indica sucesso.
- Confirmar no código fonte. Visualizar código fonte no navegador.
- Trecho do código 200 usa eval (linha 137) para processar o JSON da resposta.

- No Burp, enviar a última requisição da sequência para o Repeater.
- No Repeater, substituir o payload “iron man” pelo script malicioso.
    - `<script>alert(document.cookie);</script>`
    - Clicar no botão Send.
    - O script malicioso pode ser visto na resposta
- Para executar a exploração, no frame Response, clicar no botão “Actions” ou botão direito do mouse, selecionar a opção Show response in browser
    - Clicar no botão Copy da caixa de diálogo para copiar a URL do exploit
- No browser proxy do burp, abrir uma nova aba e colar a url do exploit.
    - O script é executado, mostrando um alerta.
- (Tem que ser no mesmo browser de onde o teste foi iniciado, lá já existe uma sessão ativa com token de sessão para reuso)

Exploração de XSS AJAX/XML com Burp Suite

Ainda no bWAPP em proxy pelo Burb.

- Selecionar o bug ***XSS - Reflected (AJAX/XML)***.
- Testar a busca com qualquer nome de filme, p.ex., fazer uma busca por “star wars”.
- Nesse formulário, não há botão de Submit. E agora?!
    - A busca é feita e refeita a medida em que o campo é preenchido.
    - Isso indica o uso de uma tecnologia responsiva. Neste, caso é AJAX.
- No Burp, visualizar as sequência de requisições assíncronas AJAX. Selecionar a última.
    - O Campo title está codificado em XML (UTF-8): “`star%20wars`”.
    - A Resposta indica o código 200 OK.
- Queremos usar o payload HTML `<img src="x" onerror="alert(1)"/>`

- No Burp, enviar a última requisição da sequência para o Repeater.
- No Repeater, substituir o payload “`star%20wars`” pela tag html maliciosa.
    - `<img src="x" onerror="alert(1)"/>`
    - Clicar no botão ***Send***.
    - O alerta não foi emitido! Por que? Dica: falta codificar o payload em html.
    - Selecionar o Payload malicioso e clicar no botão direito do mouse.
    - ***Botão direito → Convert selection → HTML → HTML-encode key characters***.
    - O payload codificado `&lt;img src=&quot;x&quot; onerror=&quot;alert(1)&quot;/&gt;`
    - (opcional) para voltar: ***Botão direito → Convert selection → HTML → HTML-decode***.

- De volta para a aplicação bWAPP e bug ***XSS - Reflected (AJAX/XML)***
    - colar o payload codificado no campo de busca
        - `&lt;img src=&quot;x&quot; onerror=&quot;alert(1)&quot;/&gt;`
    - o alerta será emitido.
- Dessa forma, o resultado é o bloqueio da aplicação. Vai ser necessário fechar o browser..

Por que isso acontece?

- No browser, na aba da bWAPP, abrir as ferramentas de desenvolvedor.
- Usar o ponteiro para inspecionar o elemento “`Sorry ...`”
- O parâmetro `result` aparece piscando porque está sendo atualizado continuamente.
- Em `network`, selecionar a requisição mais recente, em `initiators`, clicar no arquivo `.js`
- O código fonte da função process() mostra que `title=encodeURIComponent(...)` .
- Além disso, no processamento da resposta, `xmlResponse = xmlHttp.responseXML;`

Exploração de SQLi com Burp Suite

Para terminar com o bWAPP em proxy pelo Burb.

- No bWAPP
  - Selecionar o bug ***SQL Injection (POST/Select)***.
  - Escolha um filme da lista de seleção e clique em ***Go***.
- No Burp, ***Proxy*** → ***Proxy History***, seleciona a requisição mais recente.
  - Observar como o parâmetro de busca `movie` é usado.
  - Copiar a requisição para o Repeater
- No Burp Repeater
  - Modificar `movie` da requisição para `movie=1 or 1=1#&action=go`
  - Clicar em ***Send***. O que acontece? Dica: o comportamento não foi modificado.
  - Modificar `movie` da requisição para `movie=`200 union select 1,2,3,4,5,6,7#
  - Clicar em ***Send***. O que aconteceu dessa vez?