Universidade Federal do Rio de Janeiro

Processo seletivo GRIS

Thais Angelo Ferreira de Oliveira

TAG WEB:

1) **O que é o protocolo HTTP e Como ele funciona?**

R:

Hypertext Transfer Protocol (HTTP) é um protocolo utilizado para enviar e receber informações na web. A versão mais utilizada atualmente é a 1.1, definida pela especificação RFC 2016. Ele é baseado em requisições e respostas entre clientes e servidores. O cliente (user agent) envia uma solicitação ao servidor no formato de um método de solicitação, contendo a linha de pedido, os campos de cabeçalho do pedido e o corpo do pedido. O servidor responde a solicitação com uma linha de status, os campos da resposta e o corpo dela contendo o código de resposta identificando se a requisição foi concluída. A maioria das comunicações HTTP é iniciada por um user agente e consiste em uma solicitação a ser aplicada a um recurso em algum servidor de origem. As comunicações HTTP geralmente ocorrem através de conexões TCP/IP e em seu caso mais simples é feito por meio de uma única conexão entre o usuário e o servidor. Uma situação mais elaborada ocorre quando um ou mais intermediários então na rede de solicitação/resposta. As três formas mais comuns de intermediários são: Proxy, Gateway e um túnel. O protocolo HTTP é **stateless**, assim, ele não é capaz de reter informações entre requisições diferentes. Para isso deve-se utilizar cookies, sessões, campos de formulário ou variáveis na própria URL.

**2) O que é um Response Code? Cite um exemplo de um programa que você pode fazer com ele?**

R:

Os códigos de status de resposta são emitidos por um servidor em resposta à solicitação de um cliente feita indicando se uma requisição foi corretamnte concluída. Ele inclui códigos de solicitação de comentários IETF (RFCs), outras especificações e alguns códigos adicionais usados em alguns aplicativos comuns do HTTP. O primeiro dígito do código de status define a classe de resposta. Todos os códigos HTTP são separados em cinco classes ou categorias:

- **1XX**: Informativo – a solicitação foi aceita ou o processo continua em andamento;
- **2XX**: Sucesso – a ação foi concluída ou entendida;
- **3XX**: Redirecionamento – indica que algo mais precisa ser feito ou precisou ser feito para completar a solicitação;
- **4XX**: Erro do cliente- indica que a solicitação não pode ser concluída ou contém a sintaxe incorreta;
- **5XX**: Outros erros.

**3) O que é um HEADER? Cite um uso INSEGURO desse cabeçalho.**
R:

O cabeçalho é a parte do pacote que precede os dados e que contém a fonte, o destino e o endereço do IP do receptor, checagem de erros e outros campos. O cabeçalho também é a parte de uma mensagem eletrônica que traz, entre outras coisas, o remetente, dia e hora. Ele permite que o cliente e o servidor passem informações adicionais com a solicitação ou a resposta HTTP.

Um exemplo de uso inseguro desse cabeçalho ocorre quando um site aceita uma conexão por meio de HTTP e redireciona para HTTPS, os visitantes poderão se comunicar inicialmente com a versão não criptografada do site antes de serem redirecionados, criando uma oportunidade para um ataque na infraestrutura da rede do usuário por exemplo. Atualmente a grande maioria dos navegadores web possuem suporte a configuração de segurança que previne que a comunicação seja enviada via HTTP para o servidor, forçando o navegador do usuário enviar os dados somente via HTTPS. Essa característica é importante para prevenir que ataques que façam a navegação passe a ser transmitidas por um meio de comunicação inseguro.

**4) O que é um Método HTTP? Explique o funcionamento do método POST, o funcionamento do método GET. Explique qual é considerado mais seguro e por que.**
R:

O protocolo HTTP determina um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso. Esses métodos também chamados de verbos HTTP e dependendo do verbo o servidor pode dar uma resposta diferente.

O método GET é utilizado para ter o retorno de um recurso pedido e pode ser gerado por um formulário web. Ele possui um limite de tamanho e só aceita strings. Esse método utiliza a URL para enviar os dados ao servidor. Já método POST é usado para solicitar que o servidor de origem aceite a entidade incluída na solicitação como um novo subordinado do recurso identificado. Esse método só pode ser gerado por meio de um formulario web.

No método post, ao contrário do get, os parâmetros podem ser enviados na URL e também no corpo da requisição tornando-o mais seguro, pois os dados não ficam expostos na URL evitando que as informações da requisição fiquem salvas no histórico. Por mais que isso seja uma forma melhor de preservar os dados, não significa que eles estejam totalmente protegidos.

**5) O que é Cache e como ele funciona? Cite os principais HEADERS de Request e Response responsáveis pelo controle de Cache.**
R:

O cache é um recurso que permite aos navegadores de Internet armazenar páginas visitadas com frequência. Cada site visitado é guardado no cache do navegador, como um arquivo temporário, para que, na próxima visita, a página seja visualizada mais rapidamente. Como as páginas ficam armazenadas, elas ocupam espaço no disco rígido. Um exemplo conhecido do uso do cache é um o Google, ou seja, isso significa que uma cópia da página é feita por bots, e é armazenada no servidor. Assim, em caso do site estar indisponível, é possível

acessar uma versão mais antiga que já estava armazenada no servidor do Google. Isso se deve ao fato do cache do navegador salvar a estrutura básica das páginas que você está habituado a visitar, poupando o tempo de download delas em exibições futuras. Ele salva os planos de fundo das páginas, principais links e diversos outros dados dela, o que torna a navegação mais rápida.

Os principais headers de Request e Response responsáveis pelo controle de Cache são: Cash-control e Pragma.

**6) O que é Cookie? Qual é o principal ataque relacionado a ele?**

R:

Os cookies são arquivos de texto simples, enviados pelo site ao navegador, na primeira visita. No próximo acesso, o navegador reenvia os dados ao site para que suas informações sejam configuradas de forma automática. Cada vez que o usuário visita o site novamente, o navegador envia o cookie de volta para o servidor para notificar atividades prévias do usuário. Por exemplo, a maioria dos sites armazenam informações básicas, como endereços IP e preferências sobre idiomas, cores, etc. Contudo, em portais que exigem cadastros, os nomes de usuários e senhas de email também fazem parte dos Cookies. O site determina quanto tempo o arquivo vai ficar armazenado, o que pode variar entre dias e anos.

Um dos ataques mais comuns a eles é a captura de cookies, essa captura pode ser feita através de ferramentas que analisam o tráfego de rede. Esses ataques ocorrem quando os cookies são enviados sob uma requisição HTTP, permitindo que o atacante ou algum código malicioso tenha acesso indevido aos dados gravados nele.

**7) O que é OWASP-Top-Ten?**

R:

O Top 10 da OWASP é um documento de conscientização padrão para desenvolvedores e segurança de aplicativos da web. Representa um amplo consenso sobre os riscos de segurança mais críticos para aplicativos da Web. As empresas devem adotar este documento e iniciar o processo de garantir que seus aplicativos da web minimizem esses riscos. O uso do Top 10 da OWASP é talvez o primeiro passo mais eficaz para mudar a cultura de desenvolvimento de software dentro da organização para a produçao de código mais seguro.

**8)O que é Recon e Por que ela é importante?**

R:

A fase de reconhecimento (recon) da aplicação web fornecerão informações detalhadas sobre os recursos (páginas, arquivos, diretórios, links, imagens etc.) que a compõem. São informações muito importantes, posteriormente utilizadas na frase de exploração de falhas da aplicação web. Realizar o reconhecimento da aplicação envolve descobrir todo e qualquer recurso com o qual interage para buscar vulnerabilidades. Somente os recursos descobertos durante essa fase serão percorridos, portanto é muito importante que o máximo possível seja

encontrado. O critério mais importante para realizar o reconhecimento da aplicação para descobrir os recursos relacionados é entender o comportamento da aplicação.

## 9)Command Injection (SO-Injection)

### a) O que é Command Injection?

R:

Command Injection é um ataque no qual o objetivo é a execução de comandos arbitrários no sistema operacional host por meio de um aplicativo vulnerável. Os ataques de injeção de comando são possíveis quando um aplicativo passa dados inseguros fornecidos pelo usuário (formulários, cookies, cabeçalhos HTTP etc.) para um shell do sistema. Nesse ataque, os comandos do sistema operacional fornecidos pelo invasor geralmente são executados com os privilégios do aplicativo vulnerável. Os ataques de injeção de comando são possíveis em grande parte devido à validação de entrada insuficiente. Esse ataque difere da code injection , pois ela permite que o invasor adicione seu próprio código que é executado pelo aplicativo. Na Injeção de Comando, o invasor estende a funcionalidade padrão do aplicativo, que executa comandos do sistema, sem a necessidade de injetar código.

### b)Mostre um exemplo de Command Injection ( PoC da exploração )

R:

## 10) SQL INJECTION

### a) O que é SQL injection?

R:

O SQL Injection éuma falha na codificação de uma aplicação (seja web ou local) que possibilita, por meio de um input qualquer, a manipulação de uma consulta SQL. Ele é um termo que indica um tipo de ameaça, que usa falhas existentes em sistemas, para interagir com o banco de dados dos mesmos através de comandos SQL. Este tipo de ataque serve para alterar e manipular informações do banco, comprometendo a integridade dos dados armazenados.

### b)O que é Union Based Attack?

R:

Quando um aplicativo é vulnerável à injeção de SQL e os resultados da consulta são retornados nas respostas do aplicativo, a palavra-chave UNION pode ser usada para recuperar dados de outras tabelas no banco de dados. Isso resulta em um ataque UNION de injeção SQL. Ele permite que um invasor extraia informações do banco de dados estendendo os resultados retornados pela consulta original. O operador Union só pode ser usado se as consultas originais / novas tiverem a mesma estrutura (número e tipo de dados das colunas). O operador UNION estende os resultados retornados pela consulta original, permitindo que os usuários executem duas ou mais instruções se elas tiverem a mesma estrutura que a original. A palavra-chave

UNION permite executar uma ou mais consultas SELECT adicionais e anexar os resultados à consulta original.

c**) O que é Blind-SQL-I?**
R:

Blind SQL injection é um tipo de ataque SQL injection que faz perguntas verdadeiras oufalsas ao banco de dados e determina a resposta com base na resposta dos aplicativos. Esse ataque geralmente é usado quando o aplicativo Web está configurado para mostrar mensagens de erros genéricas, mas não atenua o código vulnerável a SQL injection. Quando um invasor explora a SQL injection, às vezes o aplicativo Web exibe mensagens de erro do banco de dados reclamando que a sintaxe da consulta SQL entá incorreta. A blind SQL é quase idêntica à SQL injection, a única diferença é a maneira como os dados são recuperados do banco d dados. Quando o banco não gera dados para a página web, um invasor é forçado a roubar dados fazendo ao banco uma serie de perguntas verdadeiras ou falsas. Isso torna a exploração da vulnerabilidade SQL injectio mais difícil.

**d)Mostre um exemplo de um Blind SQL-Injection ( PoC da exploração ).**
R:

**11) XSS**

**a)O que é XSS?**
R:

É uma vulnerabilidade presente em aplicações web que permite que o cibercriminoso insira códigos Java Script para obter certos tipos de vantagem sobre as vítimas. O Cross-Site Scripting (XSS) é normalmente aplicado em páginas que sejam comuns a todos os usuários, como por exemplo a página inicial de um site ou até mesmo páginas onde usuários podem deixar seus depoimentos. Para que o ataque possa ocorrer é necessário um formulário que permita a interação do atacante, como por exemplo em campos de busca ou inserção de comentários.

**b)Quais são os tipos de XSS? Explique-os.**
R:

Existem três tipos de ataques XSS:

Reflected XSS:

Nele uma área do site que não armazena informações é utilizada para injetar o código malicioso, como por exemplo, o campo de busca. A URL com o código malicioso normalmente é espalhadas aos usuários através de SPAM, assim que as vítimas acessam a URL, o script malicioso.

Stored XSS:

Essa forma de ataque exige que os cibercriminosos possuam uma forma de escrever dados diretamente na página, como por exemplo, campos de comentários. É mais perigoso que o reflected por manter os dados armazenados permanentemente na página, fazendo com que todos os usuários que visitem esta área específica executem o script malicioso sempre que a acessem.

DOM Based XSS:

A vulnerabilidade DOM (Document Object Model) Based XSS executa todos os códigos Java Script maliciosos localmente no browser da vítima, sem ter contato direto com o servidor. Esse tipo de ataque é menos frequente, pois depende que a página alvo tenha componentes específicos que permitam que a ativação dos códigos aconteça em tempo de execução.

**c)Mostre um exemplo de um XSS Stored ( PoC da exploração ).**
R:

**d)Mostre um exemplo de um DOM-XSS (PoC da exploração ).**
R:

## 12) LFI , RFI e Path Traversal

**a) O que é LFI?**
R:

Local File Inclusion (LFI) é o processo de inclusão de arquivos, que já estão presentes localmente no servidor em questão, através da exploração de processos de inclusão vulneráveis, implementados na aplicação web. Ela permite que o atacante inclua um arquivo para explorar o mecanismo de dynamic file inclusion( inclusão dinâmica de arquivo ) implementado na aplicação web. A falha ocorre devido ao fato de que o atacante pode passar qualquer valor para o parâmetro da aplicação alvo e a mesma não faz a validação correta do valor informado antes de executar a operação. Esse tipo de falha faz com que a aplicação web mostre o conteúdo de alguns arquivos, mas dependendo da severidade, essa falha também permite, por exemplo, execução de código no servidor, negação de Serviço (DoS) e vazamento de informações sensíveis.

**b)O que é RFI?**
R:

Remote File Inclusion (RFI) é o processo de inclusão de arquivos remotos, através da exploração dos processos de inclusão vulneráveis, implementados na aplicação web. Esta falha ocorre, por exemplo, quando uma página recebe como entrada, o caminho para o arquivo que

será incluído, e esta entrada não é validada de forma correta pela aplicação web, permitindo assim que uma URL externa seja injetada na aplicação.

**c) O que é Path Traversal?**

R:

Path traversal é um exploit via HTTP que permite ao atacante acessar diretórios restritos e executar comandos fora do diretório no qual a aplicação web esteja rodando. O ataque de Path Transversal tem o objetivo de acessar arquivos e diretórios que estão armazenados fora do diretório utilizado pela aplicação. Ao Acessar a aplicação pelo browser o atacante pode olhar os links absolutos para arquivos armazenados no Web Server. Manipulando as variáveis que referencia os arquivos utilizando "dot-dot-slash (../)" utilizando diferentes sequências e variações podem permitir acesso arbitrário a arquivos e diretórios no file system, incluindo código fonte da aplicação, arquivos críticos e de configuração do sistema operacional entre outros. Utilizando "../" o atacante consegue direcionar a aplicação para diretórios acima do diretório padrão.

d**)Como aliar Path Traversal e LFI**

R:

A LFI e as vulnerabilidades da Path Traversal ocorrem quando os dados fornecidos pelo usuário conseguem analisar o sistema de arquivos subjacente do servidor. Em outras palavras, um invasor pode, entre outras coisas, ler arquivos do servidor. Devido à essas vulnerabilidades algumas consequências podem ser esperadas como por exemplo, Listagem de nomes de arquivos ou diretórios no sistema de arquivos, negação de serviço do servidor completo.

**e) Mostre um exemplo de LFI utilizando a contaminação de LOGS (PoC da exploração).**

R:

**13) CSRF e SSRF**

**a) O que e CSRF?**

R:

O Cross site request forgery (CSRF) é um ataque pelo qual uma entidade maliciosa engana a vítima para executar ações em nome do invasor. O impacto do ataque dependerá do nível de permissões que a vítima está sendo explorada. As ações praticadas pelo atacante certamente terão um efeito maior se a vítima que estiver executando as ações estiver em um nível administrativo. Os ataques de CSRF tiram proveito do fato de que um aplicativo da Web confia completamente em um usuário, uma vez que pode confirmar que o usuário é de fato quem ele diz ser.

**b)Mostre um exemplo de CSRF ( PoC da exploração )**

R:

**c)O que é SSRF?**

R:

A falsificação de solicitação do lado do servidor (SSRF) é uma vulnerabilidade que permite que um invasor induza o aplicativo do servidor a fazer solicitações HTTP para um domínio arbitrário de sua escolha. Em casos comuns de SSRF, o invasor pode fazer com que o servidor faça uma conexão de volta para si mesmo ou para outros serviços baseados na Web na infraestrutura da organização ou para sistemas externos de terceiros.

**d)Mostre um exemplo de SSRF ( PoC da exploração )**

R:

**e) Como evitar ataques de CSRF?**

R:

- Desconectar de aplicativos Web quando terminar de usá-los.
- Limpe os cookies do navegador periodicamente.
- Usar corretamente o post e o get
- Não alterar o estado em uma solicitação get.
- Desativar os plug-ins que negam os pedidos de cross-site.