



Escola
Superior
de Redes
RNP

Teste de Invasão de Aplicações Web

Capítulo 2

Reconhecimento e mapeamento

- **Apresentar uma metodologia de teste de invasão de aplicações web e as principais ferramentas que podem ser utilizadas no processo, abordando, especialmente, as fases de reconhecimento e mapeamento.**

- **Tipos e metodologia de teste de invasão, proxy de interceptação, web spiders, fuzzers, varredores de portas e serviços, varredores de vulnerabilidades, reconhecimento, mapeamento, controles no lado cliente.**

- **Introdução**
- **Metodologia de teste de invasão**
- **Ferramentas básicas**
- **Reconhecimento**
- **Mapeamento**
- **Descoberta de vulnerabilidades e exploração**
- **Contramedidas**

Teste de invasão, também chamado de teste de intrusão, teste de penetração ou pentest, é um método utilizado para verificar a segurança de um ambiente, plataforma ou sistema, por meio da simulação de ataques reais explorando as vulnerabilidades encontradas.

Pentest é um processo cíclico que depende principalmente do conhecimento técnico do auditor de segurança que o realiza.

Testes de penetração podem ser classificados nos seguintes tipos principais:

Teste
caixa-preta.

Teste
caixa-branca.

Teste
caixa-cinza.

Segundo o OSSTMM, testes de invasão também podem ser classificados nos seguintes tipos:

Teste
duplo-cego

Teste
duplo-cinza

Teste
reverso



Exercício de Nivelamento 1

Teste de invasão

- O que se entende por teste de invasão?

É fundamental realizar testes de invasão de acordo com métodos pré-definidos, de modo a permitir que diferentes pessoas alcancem resultados semelhantes, que possam ser reproduzidos.

O primeiro passo, antes de iniciar qualquer teste de invasão, é obter do cliente, por escrito, uma autorização para execução do teste e o escopo que será coberto pela atividade.

Outro ponto que deve ser considerado é se os testes serão realizados internamente, externamente ou a partir de ambos os posicionamentos.

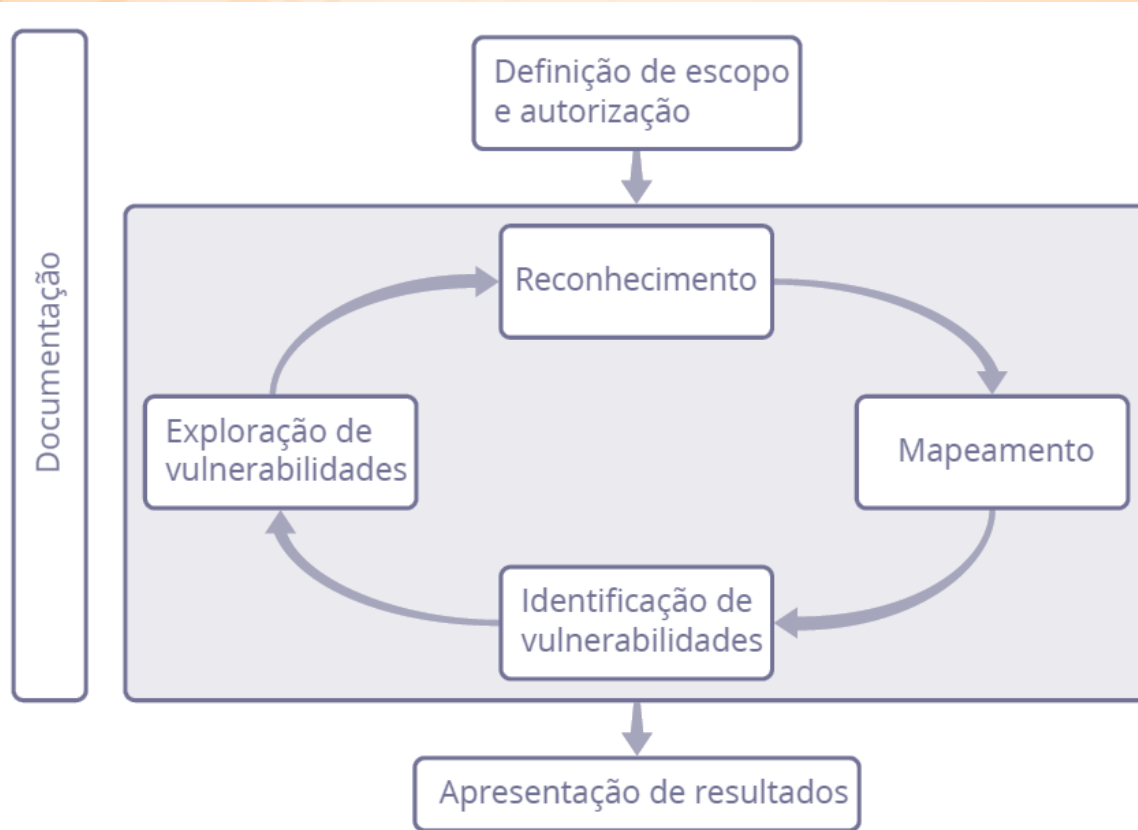


Figura 2.1 - Etapas de um teste de invasão.

Seja uma aplicação de correio eletrônico o alvo do teste de invasão.

Durante o reconhecimento, descobriu-se um diretório “conf”, acessível pelo servidor web.

As únicas páginas mapeadas foram a de autenticação e a de recuperação de senhas.

Neste cenário, que vulnerabilidades devem ser testadas?



Exercício de Fixação 1

Etapas de um teste de invasão

1. Quais são as etapas de um teste de invasão?

Para realizar qualquer teste de invasão, o ideal é que o auditor prepare um notebook, com bom processador e bastante memória, instalando uma ou mais ferramentas de cada uma das classes a seguir:

- Navegadores web
- Web spiders
- Varredores de portas e serviços
- Proxies de interceptação
- Fuzzers
- Varredores de vulnerabilidades
- Outras ferramentas

Os navegadores web são uma peça fundamental de um teste de invasão de aplicações web.

Cada navegador apresenta particularidades no uso do protocolo HTTP e na maneira como documentos HTML são manipulados, ainda mais quando são utilizados elementos não padronizados.

Isto faz com que alguns tipos de ataques não funcionem em todos os tipos de navegadores.

É interessante saber a fatia de mercado de cada navegador, para que os testes sejam direcionados para os mais utilizados apenas.

**Google Chrome – 69,42%;
Apple Safari – 8,74%;
Mozilla Firefox – 8,48%;
Microsoft Edge – 3,45%;
Opera – 2,39%.**

Os proxies de interceptação são uma das ferramentas mais utilizadas em testes de invasão de aplicações web.

Permitem inspecionar requisições e respostas HTTP e alterá-las conforme desejado, em tempo real.

Eles são executados localmente, na própria estação em que roda o navegador web, e interceptam todo tráfego deste baseado em HTTP/HTTPS, dissecando o conteúdo de cada pacote.

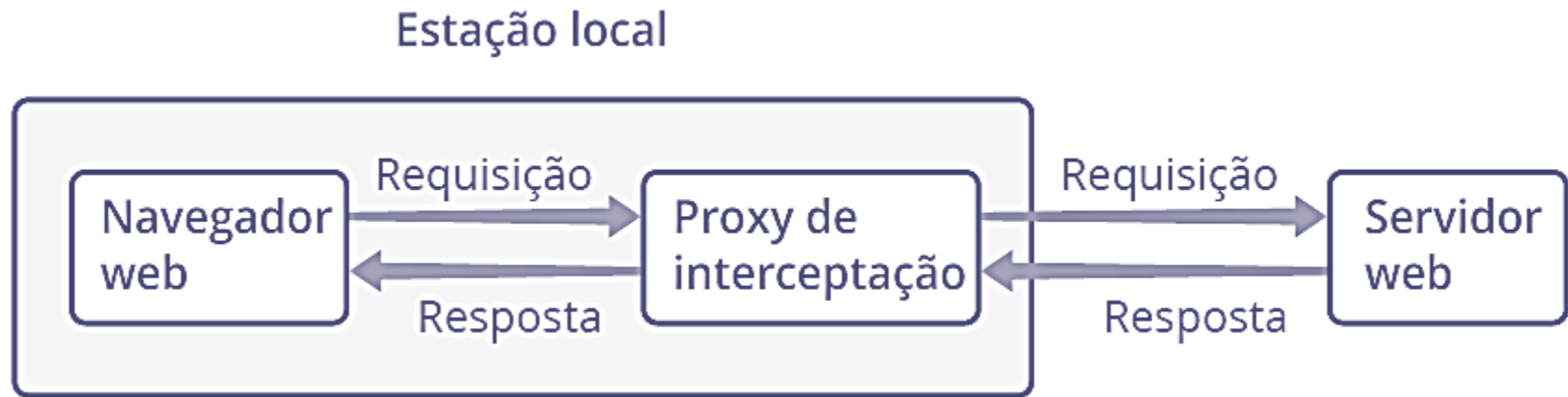


Figura 2.2 – Funcionamento de um proxy de interceptação.

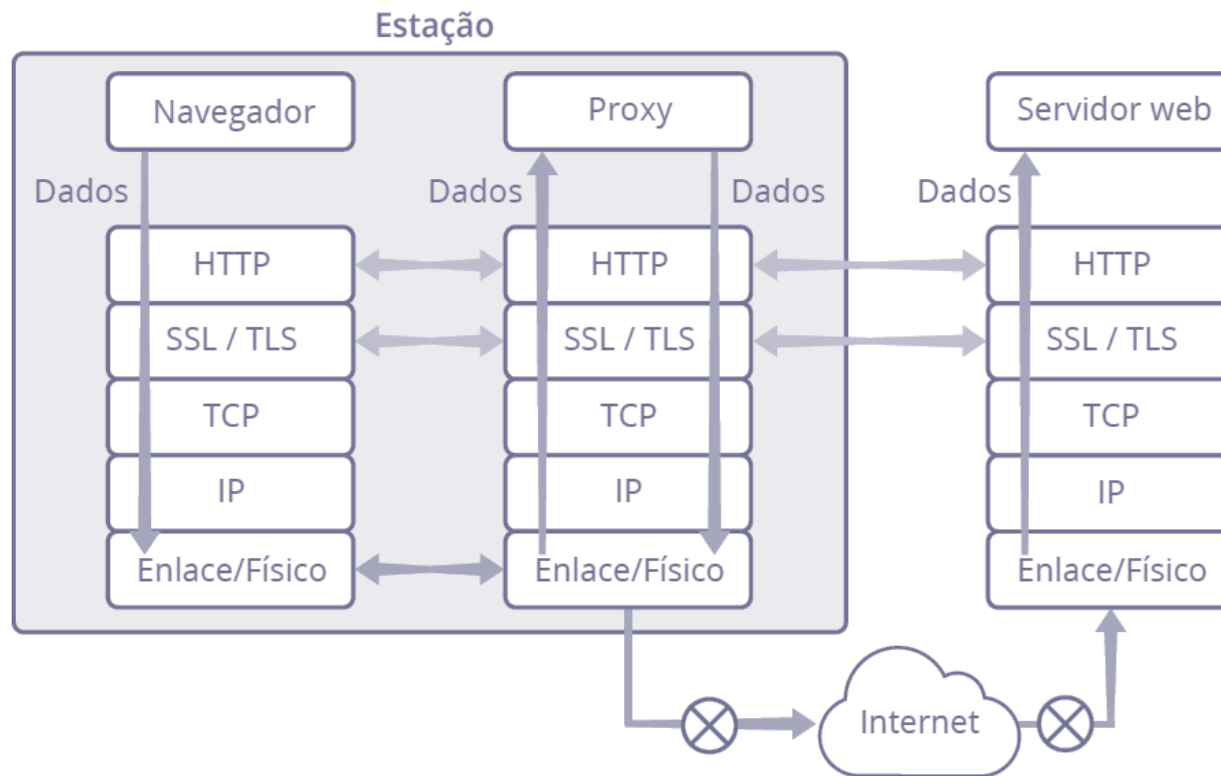


Figura 2.4 – Funcionamento de proxy de interceptação para conexões HTTPS.

Além da capacidade básica de interceptação, estas ferramentas fornecem diversas outras funcionalidades:

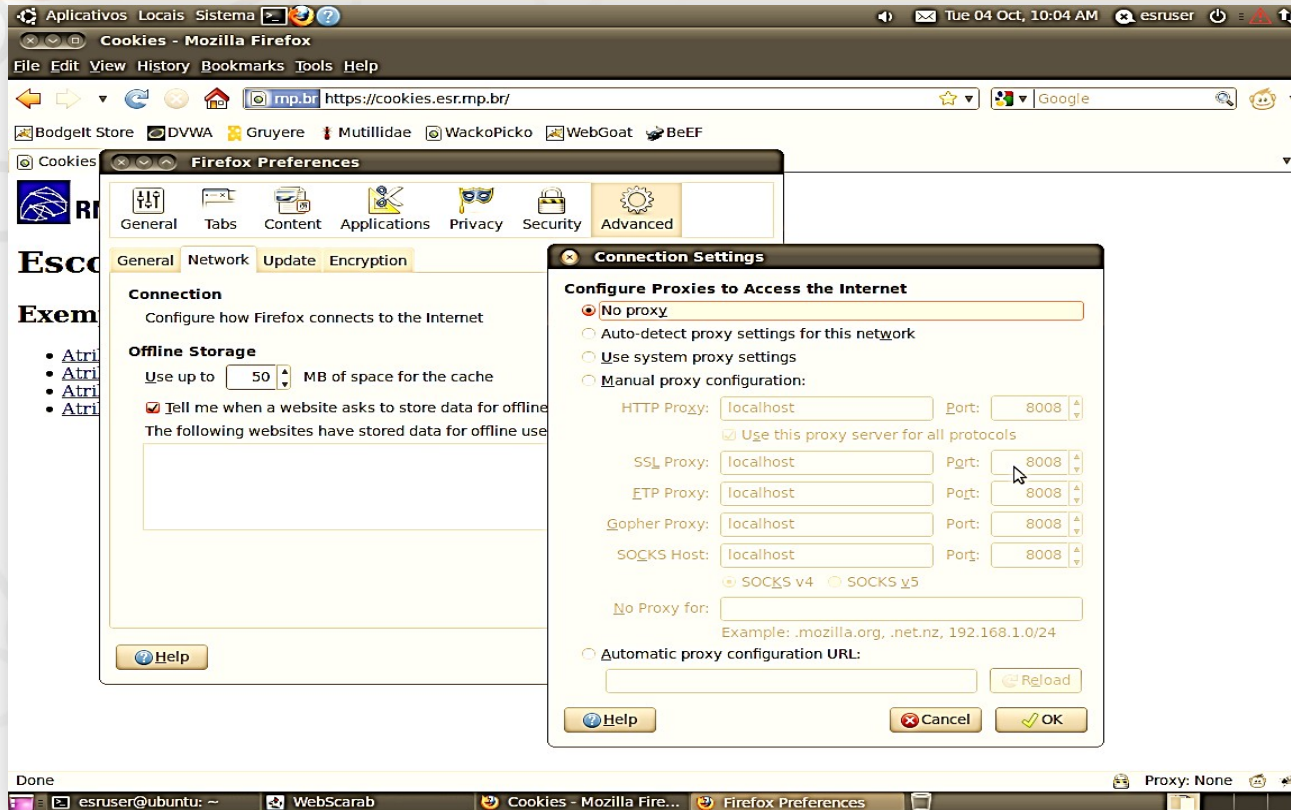
- Definição de filtros para seleção de mensagens;
- Manutenção de histórico detalhado de requisições realizadas e respectivas respostas;
- Substituição automática de valores nas mensagens por meio de regras definidas pelo usuário;
- Manipulação das interceptações diretamente no navegador;
- Revelação de campos escondidos, para visualização direta no navegador;
- etc.

Os exemplares mais sofisticados dos proxies de interceptação fazem parte de suítes integradas de teste de aplicações, dentre as quais pode-se citar:

OWASP
ZAP

Burp Suite.

Proxies de interceptação



Também chamadas de web crawlers, têm por objetivo montar automaticamente o mapa da aplicação, visitando cada página disponível e realizando uma cópia local de cada recurso encontrado, para fins de análise posterior.

Esse processo é realizado facilmente para sítios web com conteúdo estático, por meio de uma busca recursiva de recursos.

Quando executadas para mapear aplicações web, porém, devido à natureza dinâmica destas, diversas dificuldades surgem, que nem sempre podem ser resolvidas satisfatoriamente.

- ▲ **Exemplos:** suítes integradas de teste, wget e webshag.

A técnica de fuzzing consiste em fornecer automaticamente valores para campos e parâmetros que aceitam dados de usuário, em uma aplicação, serviço ou protocolo, com o objetivo de descobrir vulnerabilidades que possam ser exploradas.

Cabe ao analista de segurança descrever o domínio de valores a ser testado em cada item de entrada, mas, normalmente, listas pré-definidas estão disponíveis para os casos mais comuns.

Há uma diversidade de fuzzers disponíveis para uso, por exemplo o OWASP ZAP.

Um varredor de portas e serviços é um software que analisa uma ou mais máquinas e identifica as portas abertas e os serviços específicos sendo executados em cada uma delas.

Opcionalmente, pode também detectar o tipo e a versão do sistema operacional utilizado.

O princípio de funcionamento de um varredor de portas se baseia na suposição que o ativo implementa corretamente a camada de transporte da pilha de rede, de acordo com a RFC 793.

Os principais métodos de detecção são:

Varredura TCP

Varredura SYN

Varredura UDP

Varredura FIN

A identificação dos respectivos serviços pode ser realizada por meio de duas técnicas principais.

A primeira, chamada de verificação nula e aplicável ao protocolo TCP, consiste em se conectar na porta e esperar alguns segundos, pela possibilidade de apresentação do banner de boas vindas. Se isto ocorrer, ele é comparado contra uma base que tem o mapeamento para versões específicas de serviços.

O segundo método, mais confiável e aplicável a portas TCP e UDP, resume-se em interagir com o serviço e realizar a identificação, de acordo com o comportamento apresentado e com uma base de respostas características de serviços conhecidos.

Varredores de portas e serviços

```
esruser@ubuntu: ~  
Arquivo Editar Ver Terminal Ajuda  
esruser@ubuntu:~$ nmap -A -T4 exemplo.esr.rnp.br  
  
Starting Nmap 5.00 ( http://nmap.org ) at 2011-10-04 16:00 BRT  
Interesting ports on exemplo.esr.rnp.br (192.168.213.200):  
Not shown: 993 filtered ports  
PORT      STATE SERVICE  VERSION  
22/tcp    open  ssh      OpenSSH 5.5 (protocol 2.0)  
|_ ssh-hostkey: 1024 c0:df:2d:0b:bd:91:c0:05:80:e5:ec:ef:bf:46:cd:f4 (DSA)  
|_ 2048 f0:c9:22:fe:61:ab:61:f1:7a:08:2a:00:f5:df:3a:96 (RSA)  
80/tcp    open  http     Apache httpd 2.2.17 ((Fedora))  
|_ html-title: Escola Superior de Redes  
81/tcp    open  http     lighttpd 1.4.26  
|_ html-title: Powered by lighttpd  
443/tcp   open  ssl/http Apache httpd 2.2.17 ((Fedora))  
|_ html-title: Escola Superior de Redes  
|_ sslv2: server still supports SSLv2  
3000/tcp  open  ppp?  
8080/tcp  open  http     Apache Tomcat/Coyote JSP engine 1.1  
|_ html-title: Apache Tomcat  
8090/tcp  open  http     nginx web server 0.8.53  
|_ html-title: Test Page for the Nginx HTTP Server on Fedora  
1 service unrecognized despite returning data. If you know the service/version,  
please submit the following fingerprint at http://www.insecure.org/cgi-bin/servi  
cefp-submit.cgi :
```

Varredores de portas e serviços

```
esruser@ubuntu:~$ nmap -A -T4 192.168.1.1
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2011-01-06 16:40 BRST
```

```
Interesting ports on 192.168.1.1:
```

```
Not shown: 997 filtered ports
```

```
PORT      STATE SERVICE VERSION
```

```
21/tcp open  ftp      Netgear broadband router or ZyXel VoIP adapter  
ftpd 1.0
```

```
23/tcp open  telnet   Netgear broadband router or ZyXel VoIP adapter  
telnetd
```

```
80/tcp open  http     Embedded Allegro RomPager webserver 4.07  
UPnP/1.0 (ZyXEL ZyWALL 2)
```

```
|_ html-title: Protected Object
```

```
| http-auth: HTTP Service requires authentication
```

```
| Auth type: Basic, realm = 550B-4P2
```

```
|_ HTTP server may accept admin:admin combination for Basic  
authentication
```

```
Service detection performed. Please report any incorrect results at  
http://nmap.org/submit/ .
```

```
Nmap done: 1 IP address (1 host up) scanned in 15.75 seconds
```

Estas ferramentas têm por objetivo encontrar, de maneira automatizada, o maior número possível de vulnerabilidades em um ativo.

O mecanismo básico de funcionamento consiste no envio de requisições e análise das respostas obtidas, em busca de evidências que uma dada vulnerabilidade está presente.

As vulnerabilidades que podem ser encontradas automaticamente são aquelas para as quais é possível descrever claramente a assinatura de ataque e o resultado que deve ser verificado para inferir a presença da fraqueza.

É importante conhecer também os tipos de vulnerabilidades que normalmente não são detectados corretamente.

Falhas no controle de acesso;

Problemas que, para serem explorados, precisam que parâmetros sejam alterados considerando a semântica associada;

Uso inadequado de criptografia; e

Falhas de lógica decorrentes de situações não previstas, como a remoção de um parâmetro obrigatório.

Exemplos de varredores de vulnerabilidades:

Acunetix

AppScan

Burp

N-Stalker

w3af

Webinspect



Há inúmeras outras ferramentas disponíveis, que podem ser úteis em situações em um teste de invasão:

Netcat

OpenSSL

Metasploit

Nikto



Exercício de Fixação 2

Tipos de ferramentas

1. Que tipos de ferramentas podem ser empregados em um teste de invasão?

A fase de reconhecimento tem por objetivo levantar o máximo possível de informações da aplicação alvo, principalmente nos casos de teste caixa-preta, em que quase nada é fornecido de antemão ao analista de segurança.

Embora esta etapa seja fundamental para um teste bem sucedido, muitas vezes, não é executada sistematicamente pelo auditor.

Para que o reconhecimento seja realizado com êxito, é importante se ter uma ideia do tipo de informação que deve ser procurada:

- Nomes de funcionários.
- Identificadores de usuários.
- Informações diversas sobre usuários.
- Tecnologias empregadas.

Para que o reconhecimento seja realizado com êxito, é importante se ter uma ideia do tipo de informação que deve ser procurada:

- Servidores e topologia de rede.
- Configurações dos componentes.
- Recursos disponibilizados pelos servidores web.
- Arquivos “robots.txt”.

Levantamento de infos em fontes públicas

Muitas informações interessantes para o teste de invasão podem ser obtidas em fontes públicas, sem um grande esforço. Vejam-se alguns exemplos:



Redes sociais.

Grupos de
discussão.

Anúncios de
emprego.

WHOIS

DNS

É uma técnica que utiliza o mecanismo de busca do Google para encontrar vulnerabilidades de software e de configuração em sistemas acessíveis pela Internet.

Embora o termo remeta à ferramenta específica do Google, os conceitos são gerais e podem ser aplicados a outros serviços similares.

A técnica não é aplicável para sistemas acessíveis somente pela rede interna, uma vez que os recursos da aplicação não podem ser catalogadas pelo mecanismo de busca.

Regras básicas

O comportamento padrão do Google é considerar todos as palavras fornecidas, exceto as comuns, que podem ser ignoradas.

O símbolo “*” pode ser usado para substituir um ou mais termos desconhecidos na consulta.

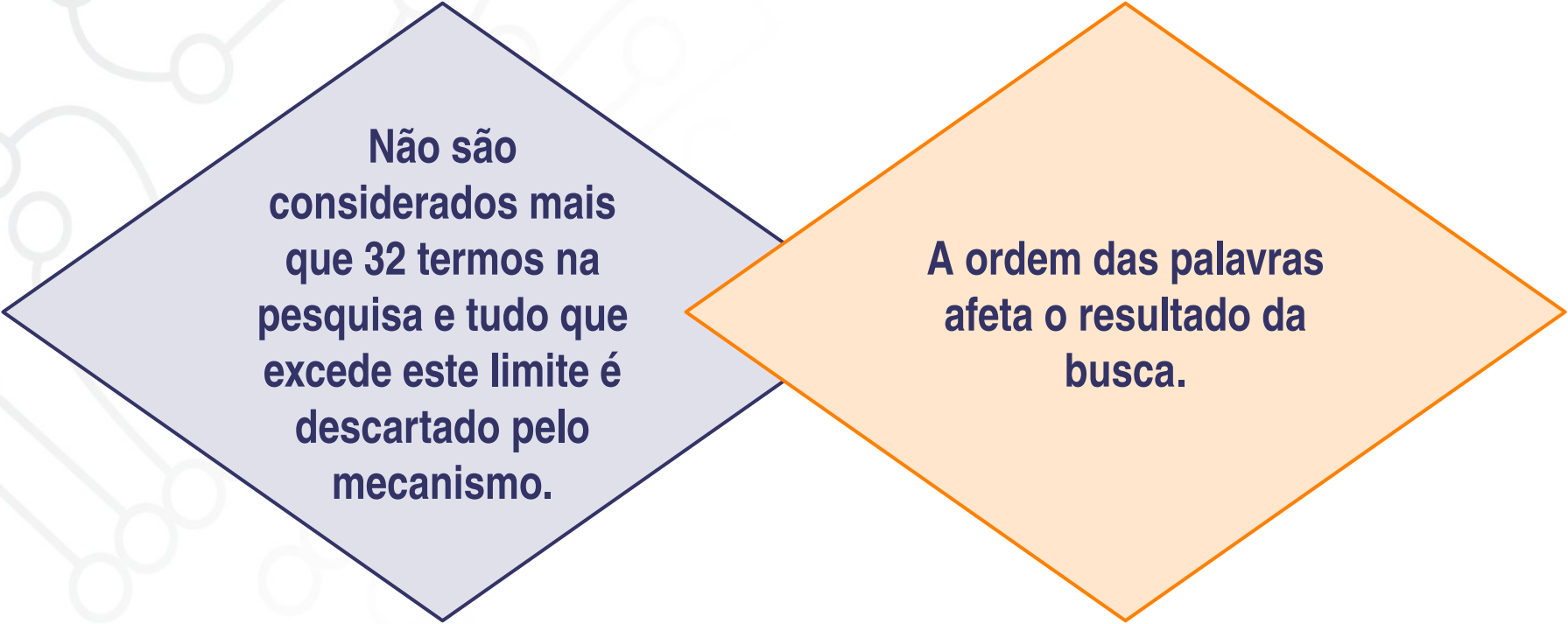
Não há diferenciação entre letras maiúsculas e minúsculas, exceto para o operador “OR”.

Regras básicas

**Os termos fornecidos
são procurados em
qualquer lugar de uma
página, incluindo
título, corpo e URL.**

**Delimitar com aspas
duplas um conjunto
de palavras determina
que estas sejam
agrupadas como uma
frase, que deve
aparecer exatamente
igual nos resultados.**

Regras básicas



**Não são
considerados mais
que 32 termos na
pesquisa e tudo que
excede este limite é
descartado pelo
mecanismo.**

**A ordem das palavras
afeta o resultado da
busca.**

Regras básicas

O operador “OR”
deve ser escrito em
maiúsculas e
seleciona as páginas
que contêm pelo
menos um dos
termos.

O operador “-”,
quando precedido de
um espaço, indica que
nenhuma página com
o termo imediatamente
posposto deve fazer
parte do resultado.

Regras básicas

O operador “+” deve ser precedido de um espaço e usado quando o termo posposto é relevante para a pesquisa, mas é ignorado pelo Google, por ser uma palavra comum ou caractere.

Operadores avançados

A sintaxe básica de um operador avançado é “operador:termo de busca”, sem nenhum espaço entre os elementos.

O termo de busca pode ser uma única palavra ou uma frase entre aspas duplas.

Uma pesquisa pode conter termos simples misturados com operadores avançados, desde que as sintaxes sejam respeitadas.

Operadores avançados

Operadores que começam com “all”, normalmente, não se dão bem com outros operadores e, portanto, devem ser empregados sozinhos.

Os operadores “-” e “OR” podem ser aplicados a operadores avançados.

Operadores avançados

Exemplos de operadores avançados:



site

intitle

allintitle

inurl

allinurl

filetype

link

author

Operadores avançados

Os exemplos abaixo foram extraídos do Google Hacking Database, de Johnny Long.

site:<domínio> login OR logon.

**site:<domínio> inurl:temp OR inurl:tmp OR
inurl:backup OR inurl:bak.**

site:<domínio> intitle:index.of people.lst.

site:<domínio> intitle:"index of" etc passwd.

**site:<domínio> "This file was generated by
Nessus".**

Para cada ativo descoberto ou listado pelo cliente, deve-se identificar o nome e versão do sistema operacional, além dos serviços e portas habilitados.

Uma das maneiras de realizar esta tarefa é escutar passivamente todo o tráfego de entrada e saída do elemento e, com base em detalhes específicos de plataforma, inferir as informações desejadas.

A outra estratégia é ativa, isto é, a identificação ocorre por meio de interação com o servidor ou elemento de rede.

Um utilitário que pode ser empregado com esta finalidade é o Nmap.

Existem métodos e utilitários que podem ser utilizados especificamente para reconhecer servidores web, e sempre é interessante conferir o resultado com mais de uma solução.

A possibilidade mais simples é solicitar um recurso inexistente para o servidor web e observar a página de erro resultante. Caso uma página personalizada não tenha sido configurada, a padronizada é exibida, a qual contém informações sobre o servidor utilizado.

Um método mais robusto de reconhecimento de servidores web envolve analisar o comportamento deles em diversas situações, pois este varia bastante de fornecedor para fornecedor, servindo assim como um discriminante.

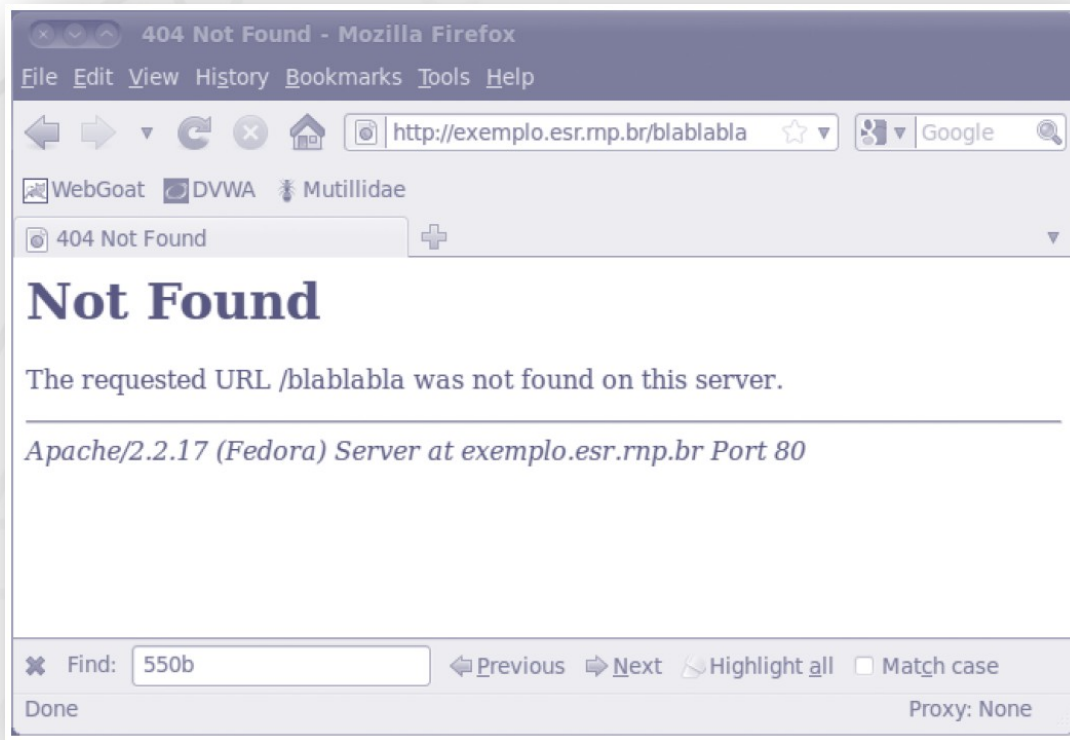


Figura 2.6 - Tela padrão de erro do Apache.

Apache 2.2.17:

HTTP/1.1 200 OK

0 Date: Wed, 05 Jan 2011 13:53:25 GMT

1 Server: Apache/2.2.17 (Fedora)

2 Last-Modified: Mon, 22 Nov 2010 01:40:24 GMT

3 ETag: "61c27-5056-4959a57036427"

4 Accept-Ranges: bytes

5 Content-Length: 20566

6 Connection: close

7 Content-Type: text/html; charset=iso-8859-1

```
lighttpd 1.4.26:  
  HTTP/1.0 200 OK  
  7 Content-Type: text/html  
  4 Accept-Ranges: bytes  
  3 ETag: "852229764"  
  2 Last-Modified: Mon, 22 Oct 2007 12:13:49 GMT  
  5 Content-Length: 844  
  6 Connection: close  
  0 Date: Wed, 05 Jan 2011 13:56:56 GMT  
  1 Server: lighttpd/1.4.26
```

Identificação do servidor web

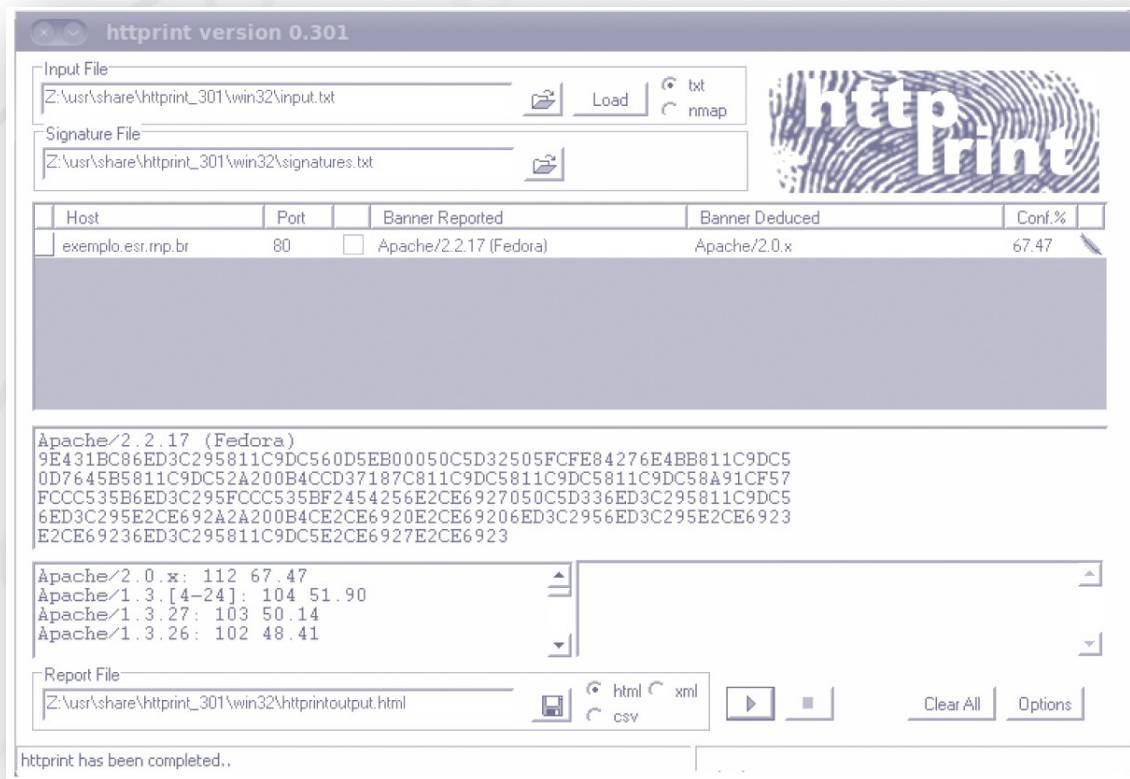


Figura 2.7 - Saída do httpprint em teste de servidor Apache.

Há métodos interessantes, como PUT, que permitem copiar arquivos para o servidor. Isso pode ser muito útil para carregar ferramentas em uma máquina da rede, que esteja servindo de pivô para atacar outro ativo do ambiente.

Assim, é vantajoso saber os métodos suportados por um servidor web, e uma maneira direta de conseguir isto é por meio do método OPTIONS, caso esteja habilitado.

```
~$ nc exemplo.esr.rnp.br 8080  
OPTIONS / HTTP/1.0
```

```
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS  
Content-Length: 0  
Date: Wed, 05 Jan 2011 16:16:12 GMT  
Connection: close
```


É muito comum hospedar diversos sítios web em um único servidor, com o objetivo de melhor alocar os recursos disponíveis.

Duas técnicas podem ser usadas:

Hospedagem virtual baseada em IP

Hospedagem virtual baseada em nomes.



Um teste simples para detectar o uso de servidor compartilhado por nomes consiste em resolver o nome de domínio da aplicação e tentar o acesso por endereço IP.

Outra maneira de testar o compartilhamento de servidor é por meio de ferramentas web que mapeiam endereços IP para nomes de domínio, de maneira similar a um DNS reverso.

WebHosting.Info's Power WHOIS Service

200.219.245.136 - IP hosts 13 Total Domains ...
Showing 1 - 13 out of 13

	Domain Name ^
1	AZULCOMERCIO.COM.
2	CABELEIREIROASADELTA.COM.
3	CENTRALDECONVENIOS.COM.
4	CORALRENOVATION.COM.
5	EDGOSURFBOARDS.COM.
6	ERCONSULTORIA.COM.
7	IZABELNOIVOS.COM.
8	JGIMOVEISRIO.COM.
9	MISSAI.COM.
10	NURATEXTIL.COM.
11	PLANO-SAUDE.NET.
12	REALMOVEISANTIGOS.COM.
13	ROBBINSOARS.COM.
1	

Figura 2.8 - Consulta realizada ao serviço WebHosting Info.

Ferramentas como o Nikto testam a presença de diversos itens de configuração, de acordo com uma base pré-cadastrada:

```
~$ nikto -host dvl.esr.rnp.br
- Nikto v2.03/2.04
...
+ Server: Apache/1.3.37 (Unix) PHP/4.4.4
+ No CGI Directories found (use '-C all' to force check
all possible dirs)
- Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE
+ OSVDB-877: HTTP method ('Allow' Header): 'TRACE' is
typically only used for debugging and should be disabled.
This message does not mean it is vulnerable to XST.
```



Exercício de Fixação 3

Fase de reconhecimento

1. Que informações devem ser obtidas na fase de reconhecimento?

Na fase de mapeamento, deve ser criado um mapa da aplicação, que reflita a estruturação dos arquivos componentes, as funcionalidades ofertadas, os pontos de entrada de informação e as tecnologias utilizadas. Tudo isso é realizado por meio dos seguintes passos:

Cópia das
páginas e
recursos da
aplicação;

Identificação dos
pontos de
entrada de
informação

Relacionamento
com as
informações de
reconhecimento.

Este primeiro passo consiste em realizar uma cópia integral das partes acessíveis da aplicação, iniciando pela página inicial e incluindo quaisquer páginas que tenham sido descobertas na fase de reconhecimento.

Uma abordagem híbrida deve ser empregada, na qual o auditor navega pela aplicação, enquanto que um web spider grava as páginas e monta o mapa automaticamente.

Neste processo, é comum que recursos ainda não mapeados sejam revelados, por meio de links ou comentários nas páginas capturadas.

Recursos escondidos, muitas vezes, podem ser encontrados a partir dos elementos já mapeados. Para isso, inicialmente, é necessário analisar o resultado obtido até o momento e observar os nomes de arquivos, diretórios e parâmetros.

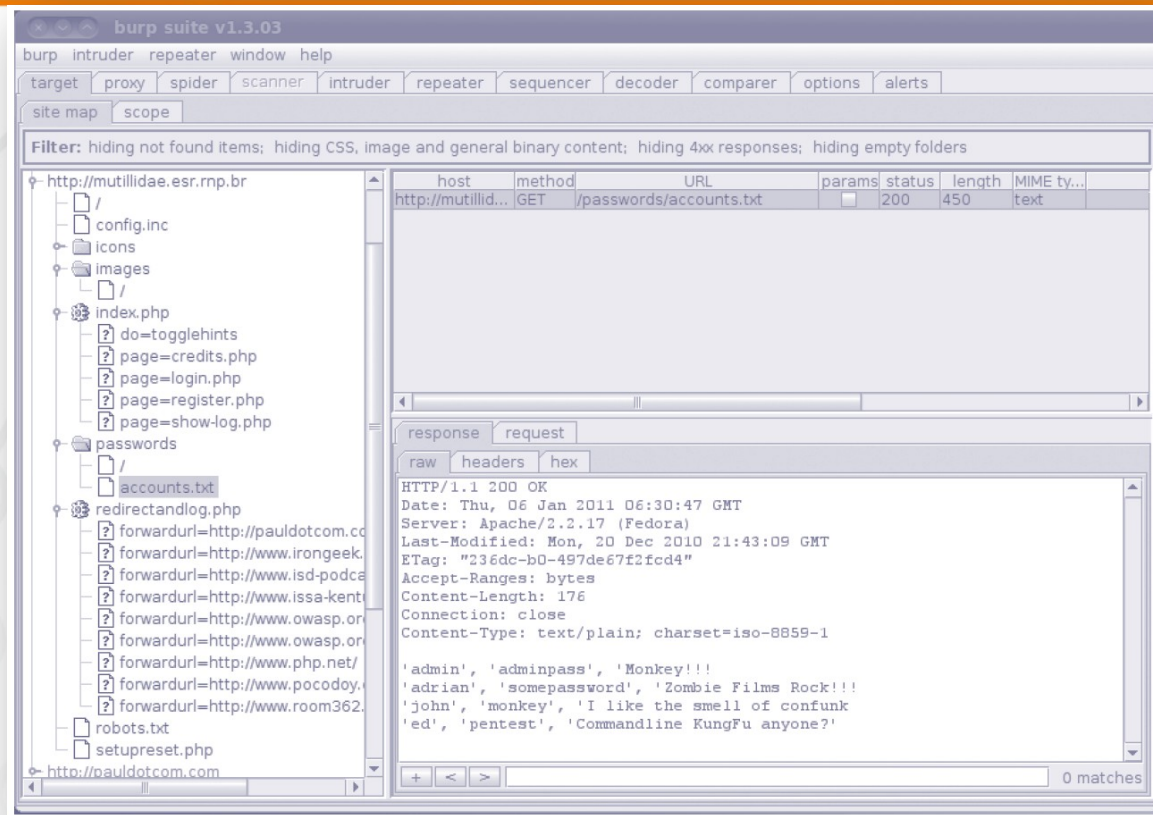


Figura 2.9 - Cópia de aplicação com Burp Suite.

É por meio da manipulação de itens de entrada que muitos problemas na aplicação são descobertos.

Os seguintes elementos devem ser levantados:

Parâmetros passados no corpo de requisições POST, principalmente, campos escondidos, que não são visíveis pela interface da aplicação.

Parâmetros passados via URL em requisições GET.

Cookies e os lugares em que são definidos e modificados.

Cabeçalhos que tendem a ser processados pela aplicação, como User-Agent, Referer e Host.

Cabeçalhos não padronizados utilizados em requisições.

Canais secundários que podem ser controlados pelo usuário.

Neste ponto, algumas das informações levantadas na fase de reconhecimento já devem ter sido utilizadas no processo de cópia dos arquivos da aplicação.

Falta ainda, porém, relacionar todas as funcionalidades mapeadas aos servidores e tecnologias identificados.

Alguns motivos para este passo:

Há vulnerabilidades que afetam apenas linguagens e plataformas específicas.

Ataques que envolvem interação com o sistema operacional devem utilizar os comandos específicos da plataforma.

Sistemas gerenciadores de bancos de dados relacionais possuem peculiaridades na sintaxe de comandos SQL.



Exercício de Nivelamento 2

Validação unilateral

- ▮ É comum encontrar aplicações que validam entradas somente no lado cliente?

Após realizar todo o levantamento e mapeamento da aplicação, prossegue-se à etapa de descoberta de vulnerabilidades, para posterior exploração.

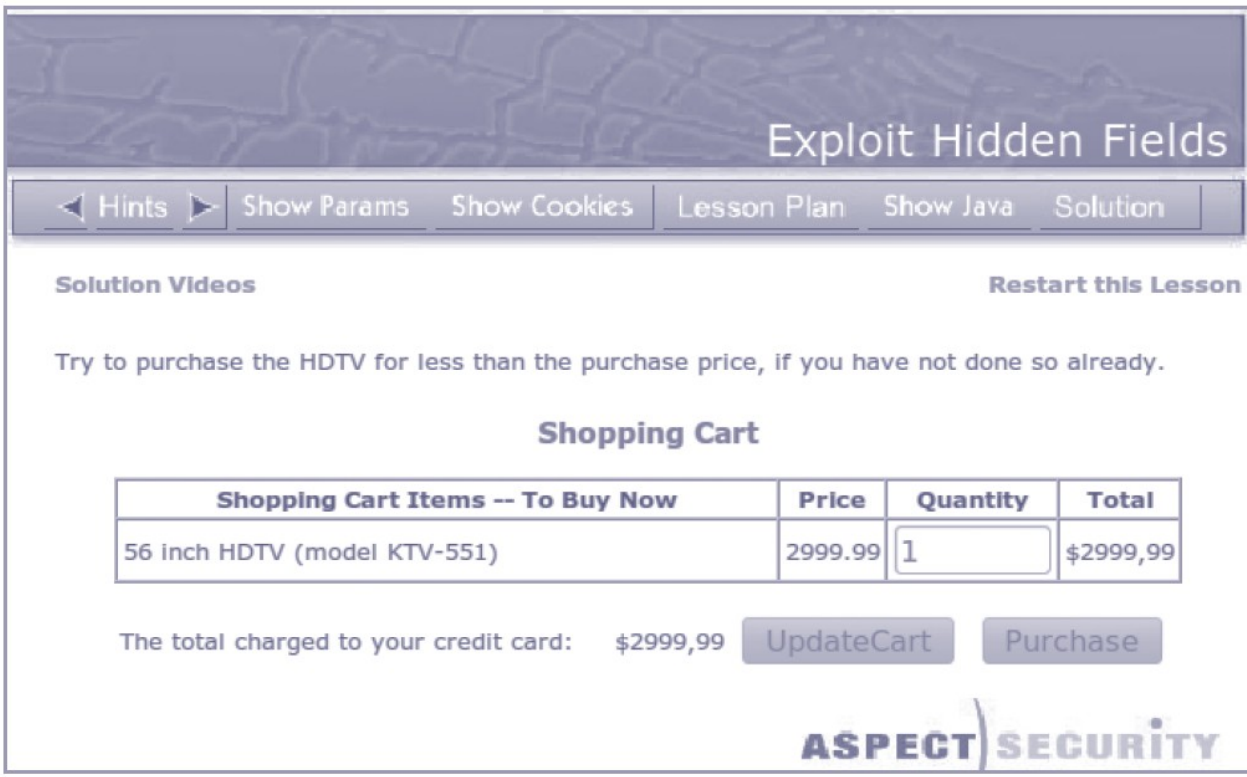
Para a maioria dos casos, porém, testes específicos para cada tipo de problema possível devem ser executados, sempre guiados pelas informações coletadas.

Com um pouco de sorte, algumas fraquezas, como utilização de usuários e senhas-padrão, são encontradas logo nas fases iniciais do teste de invasão.

**Controles que são executados
no lado cliente podem ser
violados com as ferramentas e
conhecimentos apropriados.**

O caso clássico deste problema, que afetou diversas lojas virtuais, nos princípios do comércio eletrônico, é a manutenção de preços em campos escondidos, para uso no cálculo da fatura a ser paga pelo cliente.

Um método de ataque consiste em utilizar um proxy de interceptação, para alterar as informações desejadas, em tempo de execução, antes que a requisição deixe a máquina do usuário.



Exploit Hidden Fields

< Hints Show Params Show Cookies Lesson Plan Show Java Solution

Solution Videos [Restart this Lesson](#)

Try to purchase the HDTV for less than the purchase price, if you have not done so already.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
56 inch HDTV (model KTV-551)	2999.99	<input type="text" value="1"/>	\$2999,99

The total charged to your credit card: \$2999,99 [UpdateCart](#) [Purchase](#)

ASPECT SECURITY

Figura 2.10 - WebGoat: exercício sobre exploração de campo escondido.

**POST http://webgoat.esr.rnp.br:8080/webgoat/attack?
Screen=60&menu=1700 HTTP/1.1**

Host: webgoat.esr.rnp.br:8080

...

Cookie: JSESSIONID=325E4532DAA7BD43EA8C2AE5824783C0

Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=

Content-Type: application/x-www-form-urlencoded

Content-length: 35

QTY=1&SUBMIT=Purchase&Price=2999.99

Exploração de controles no lado cliente

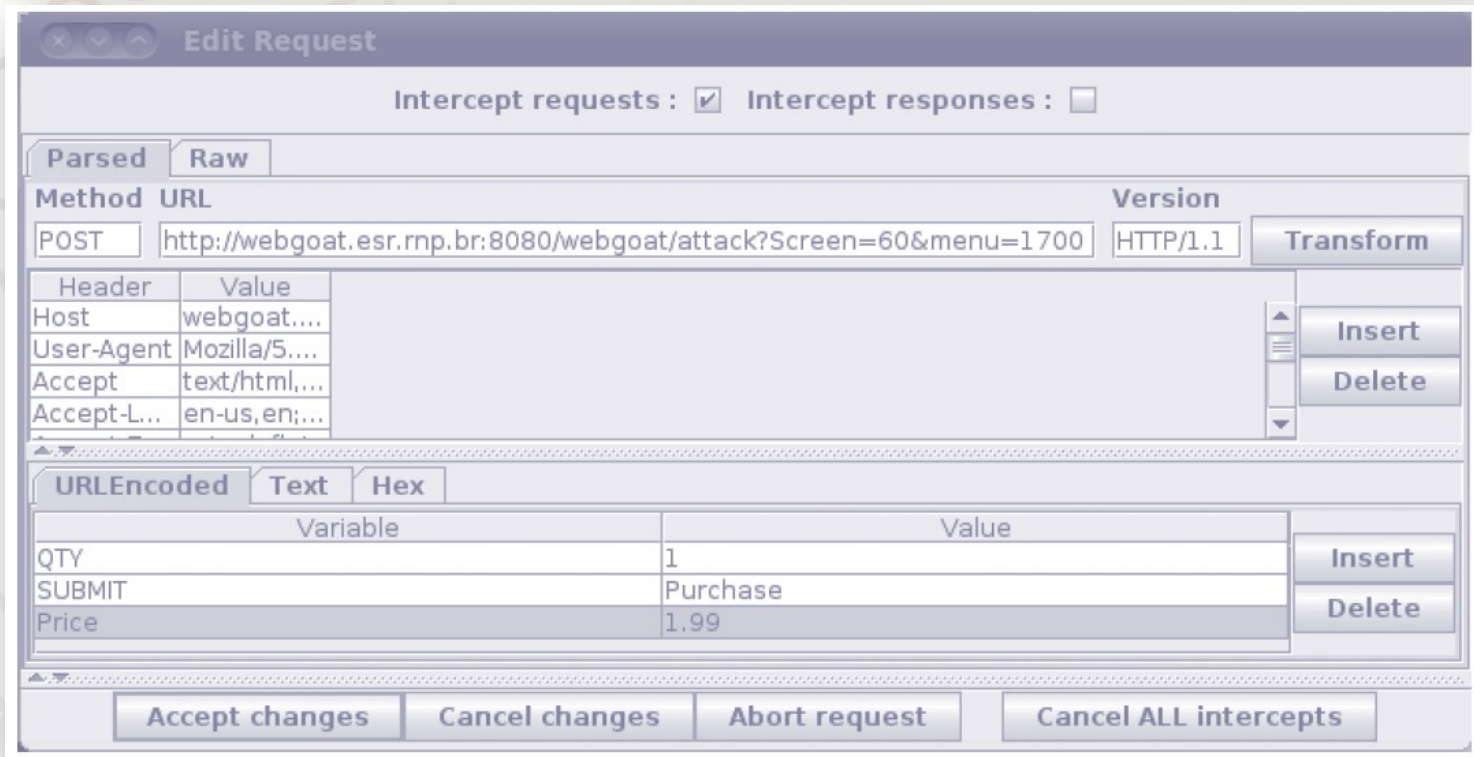


Figura 2.11 - Intercepção da requisição e alteração do valor do campo “Price”.



Exercício de Fixação 4

Segurança de controles

1. Controles no lado cliente são seguros? Em caso negativo, explique o motivo.

Parta da premissa de que todos os usuários da aplicação são maliciosos.

Não assuma que informações armazenadas em campos escondidos não podem ser alterados por um usuário.

Implemente controles no lado cliente da aplicação, apenas para evitar que um usuário bem intencionado submeta, por erro, formulários com campos inconsistentes e consuma banda de rede desnecessariamente.

Nunca espere que todos os parâmetros previstos sejam recebidos como parte da requisição.

Perguntas

?

?

?

?

?

?



Caderno de Atividade 1

1

Ferramentas básicas



Caderno de Atividade

1

2

Varredores de Portas e Serviços



**Caderno de
Atividade**

1

3

Reconhecimento



**Caderno de
Atividade
1**

4

Mapeamento



Caderno de Atividade 1

5

Descoberta e exploração de vulnerabilidades



Teste de Invasão de Aplicações Web

Capítulo 2

Reconhecimento e mapeamento



MINISTÉRIO DA
DEFESA

MINISTÉRIO DA
CIDADANIA

MINISTÉRIO DA
SAÚDE

MINISTÉRIO DA
EDUCAÇÃO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA,
INOVAÇÕES E COMUNICAÇÕES

