Chapter No. 4
"SSL and Other Advanced Settings "
In this package, you will find:

- The author’s biography
- A preview chapter from the book, Chapter no.4 "SSL and Other Advanced Settings"
- A synopsis of the book’s content
- Information on where to buy this book

About the Author

**Akash Mahajan** is "That Web Application Security Guy." He has more than 10 years of experience in application and network security. Before starting his own company, he was a technical lead for one of the leading American commercial security software companies specializing in endpoint security. He then started working on the security of the web infrastructure for the Government of India.

He is the founder and community manager at null - The Open Security Community, where he has made major contributions in making null a national-level group and null Bangalore the biggest and most vibrant chapter.

He is currently a chapter leader of Open Web Application Security Project Bangalore (OWASP Bangalore).

He is the founder of AppSec Labs, a company focused on application security, where he works with small- and medium-sized companies in securing their web server security, web security, and mobile security, and guiding them to stay secure while being competitive.

Currently, his areas of research include DevOps, SecOps, security in SDLC, cloud security, and security awareness through community building. He conducts a lot of training as well, including the extremely popular Xtreme Web Hacking.

For More Information:

www.packtpub.com/networking-and-servers/burp-suite-essentials
He was actively involved with the Bangalore Barcamp Planners group and has organized events such as AppJam and MobileCamps all over India, where he has evangelized security to small- and medium-sized enterprises.

Acknowledgments

I would like to thank my parents for their constant guidance and encouragement. They gave me all the independence in the world to break rules and question fundamentals and have fun while doing all that. I also want to thank my wife, Lubaina; finding someone like her to be with me and agreeing to marry me rekindled my faith in fate and destiny. She takes care of the house and all my tantrums with a brilliant smile all the time. I want to thank my elder sister, who has the unshakeable faith in my abilities that only elder sisters can have. Apart from my family, I would like to thank my friends, Riyaz and Anant, whom I met at null - The Open Security Community and who are far more brilliant and knowledgeable than I will ever be.

I would like to thank Neeshma, whose help, support, and eternal patience in dealing with me is the only reason this book was completed. I would like to thank the technical reviewers, David Shaw, Luca De Fulgentis, and Rejah Rehim, who read through all that I wrote—most of which in the first draft was pretty sad. They guided me to make this book much better than I ever imagined it could be. Their insights and comments helped me understand and correct numerous mistakes and blunders. I will always be grateful to them for giving their time and effort to this endeavor.

For More Information:
www.packtpub.com/networking-and-servers/burp-suite-essentials
**Burp Suite Essentials**

This book on Burp is meant for web security testers. You might be using browser plugins or automated scanners or even other interception proxy tools. In this book, you will see how Burp Suite is a versatile tool that allows almost any kind of web security testing based on your needs. This book will build on how Burp can be used with upstream proxies, SSL certificates, and more. You will learn how to search, extract, and do pattern matching for requests and responses and use that knowledge to test complex and simple web applications. You will learn to use different tools and components together to form a powerful chain of tools for web testing. As a professional tester, we need to be able to report our work, safeguard it, and sometimes even extend the tools that we use.

You will learn how different components of Burp Suite can be used together and how to use Burp Suite like a pro. You will learn to embrace the user-driven workflow for testing web applications. You can customize and extend Burp according to your needs for maximum testing and minimum software.

This book has an easy-to-follow style, where we focus on understanding what the problem is that we are trying to solve and how Burp can make it easy for us to solve. Looking at scenarios, real-world use cases, and applying the philosophy of how Burp is designed makes for an easy read and a highly actionable list of items for you to take back to your workplace.

**What This Book Covers**

*Chapter 1, Getting Started with Burp*, starts with an introduction to Burp Suite. We will cover some of the advanced flags that can be passed to the software when we invoke it from the command line. By the end of this chapter, you will have a pretty good idea of running Burp Suite in various operating systems, while being able to tweak it for maximum performance.

*Chapter 2, Configuring Browsers to Proxy through Burp*, explains that interception proxies work best when used with a browser software. Even though it is quite simple to get Burp working with a browser, advanced users can use additional browser extensions to perform powerful and customized integrations. By the end of this chapter, you will have configured your browser to use Burp as an interception proxy. Additionally, using browser extensions, you will create a powerful chain of tools to perform web security testing.
Chapter 3, *Setting the Scope and Dealing with Upstream Proxies*, shows how more and more complex web applications are being tested, including the ones that run primarily on mobile platforms. How does one configure Burp Suite to intercept in such cases? Testing web applications available on the Internet is quite simple with Burp, but how do we test applications that are inside corporate networks, running on company intranets? By the end of this chapter, you will know how to work with SSH port forwarding, SOCKS-based proxies, and intercept HTTP traffic coming from mobile devices.

Chapter 4, *SSL and Other Advanced Settings*, teaches that SSL-enabled applications sometimes require additional configuration. Usually, you add the Burp Suite CA certificate to your browser and start testing, but sometimes this is not desirable or possible at all. Some additional settings make it possible for nonbrowser-based HTTP applications and thick clients to be tested. By the end of this chapter, you will be able to set up and test SSL-enabled applications without any errors. You will also be able to test thick clients or clients that are not proxy-aware.

Chapter 5, *Using Burp Tools As a Power User – Part 1*, shows that Burp Suite is powerful due to its amazing set of tools. We will start with Target, covering Site map and Scope, and then we will move to Proxy, which is the workhorse for testers. Then, we will move to the attack tool of choice, Intruder. After Intruder, we will cover the Scanner tool and discuss when we should use the Scanner tool. We will end the chapter with the Repeater tool, which supercharges the manual testing part by making it dead simple to repeat requests and see responses.

Chapter 6, *Using Burp Tools As a Power User – Part 2*, covers the other tools that make up the Burp Suite software and shows us how tools such as Spider, Sequencer, Decoder, Comparer, and Alerts work in sync to provide us with what we need to test web applications.

Chapter 7, *Searching, Extracting, Pattern Matching, and More*, explains that the suite of tools provided by Burp is quite powerful in terms of performing the heavy lifting of crafting HTTP requests and responses based on our actions on the web applications. An important aspect of this power is the ability to match, extract, find, grep, and search all the requests and responses based on our requirements. In this chapter, you will learn the various ways in which we can search, extract, and pattern match data in requests and responses, which allow us to complete our testing.

Chapter 8, *Using Engagement Tools and Other Utilities*, covers something called the engagement tools of Burp suite. These tools allow us to automate some of the more mundane and boring parts of the security testing process. Engagement tools is a Pro-only feature of Burp Suite. Apart from the engagement tools, we will look at some smaller utilities that aid the testing process such as Search, Target Analyzer, Content Discovery, Task Scheduler, CSRF PoC Generator, and Manual Testing Simulator.

For More Information:  
www.packtpub.com/networking-and-servers/burp-suite-essentials
Chapter 9, *Using Burp Extensions and Writing Your Own*, shows that not only does Burp Suite come with its own rich set of tools, but it also provides API interfaces to extend its functionality. Many security researchers have written extensions that enhance the native functionality or add to the already rich toolset. By the end of this chapter, you will be able to use Burp Extensions and even write a sample extension in Python.

Chapter 10, *Saving Securely, Backing Up, and Other Maintenance Activities*, states that Burp Suite is just like any other testing tool. As with any software, it is imperative that you make regular backups and carry out other maintenance activities. By the end of this chapter, you will have all the knowledge about ensuring that your Burp Suite data is backed up properly and securely and how you can run scheduled tasks for backup and other maintenance activities.

Chapter 11, *Resources, References, and Links*, provides a number of great resources and references that you can rely on. It provides you with the primary references that you should follow to get more insight into how web security practitioners use Burp. We will list useful and informative resources for application security as well.

For More Information:

www.packtpub.com/networking-and-servers/burp-suite-essentials
Until now, we have successfully managed to intercept HTTP traffic. This is incredibly useful for a security professional tasked with the testing of applications that talk about HTTP. However, in our experience, we know that most secure applications are not served over HTTP, which is plain text, but over HTTP over Secure Socket Layer (SSL).

HTTPS is a combination of HTTP over SSL/TLS to prevent eavesdropping, tampering, and MITM attacks.

To intercept traffic over HTTPS, we need to configure some more settings.

Browsers and servers exchange X.509 certificates, which are signed by certificate authorities. Since Burp runs at a layer below the layer in which encryption takes place, the content of the web page is already encrypted when it reaches Burp.
The only way Burp can see the data is if the SSL/TLS connection terminates here. So, Burp generates a per-site certificate, which the browser needs to accept. Since this certificate is not signed by a certificate authority known to us, we encounter an invalid certificate error, as shown in the following screenshot:

![This Connection is Untrusted]

At this point, we can accept the error and interception will work exactly how we expect it to work. A more elegant approach is to import the Burp Suite root certificate into the browser we use for our testing. This way, we will get per-site unique certificates, and also, there will be no more errors about wrong/untrustworthy certificates.

**Importing the Burp certificate in Mozilla Firefox**

Starting with Mozilla Firefox, it is quite simple to import the certificate:

1. While Burp is running, go to `http://burp`.
2. Click on **CA Certificate**. Note where this file is downloaded:

   ![CA Certificate]

For More Information:

www.packtpub.com/networking-and-servers/burp-suite-essentials
The method is very convenient for testers, but it does open the tester to a malicious user who could perform MITM attacks against the pentester, abusing the trust related to the Burp Suite root certificate.

3. Open Firefox **Options**, click on **Advanced**, **Certificates**, and **View Certificates**. Have a look at the following screenshot:

4. Click on **Authorities**, click on the **Import** button, and navigate to the place where you downloaded the certificate, as shown in the following screenshot:

5. You will get another window about whether you trust the new certificate authority. Select **Trust this CA to identify web sites**. And if you like, click on **View** to examine the CA certificate:

---

**For More Information:**

SSL and Other Advanced Settings

6. Click on the OK button and then navigate to https://burp. If there are no errors or warnings about the certificate, you installed it successfully.

Importing the Burp certificate in Microsoft IE and Google Chrome

Google Chrome uses the same certificate store as Microsoft Internet Explorer. Adding the certificate from either one of them is enough for us. Since IE is almost always installed by default, let’s install the certificate in that first:

1. Open Internet Explorer options, and click on the Content tab, as shown in the following screenshot:

2. Internet Explorer provides us with a simple Certificate Import Wizard. Do note that the extension for the certificate is .der, which might not be visible in the file-browse dialog. Just select all files and you will be able to see them:

   ![Certificate Import Wizard]

   **Welcome to the Certificate Import Wizard**

   This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

   A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

   To continue, click Next.

For More Information:
www.packtpub.com/networking-and-servers/burp-suite-essentials
3. Accept the security warning about adding a root CA, and we are good to go, as shown in the following screenshot:

4. Navigate to `https://burp` to confirm that the certificate is installed correctly and working fine.

### Installing the Burp certificate in iOS or Android

The basic steps remain the same. We need to figure out where the certificate should get installed. For iOS, since there is no simple way to add external files, Burp documentation suggests e-mailing the certificate file to yourself and saving it from there.

As long as we have proper privileges, we can install the root certificate on our devices.

### SSL pass-through

Sometimes due to the way applications and websites are set up, it may not be possible to intercept SSL traffic. Usually, Burp will show an SSL negotiation error in the Alerts tab. One of the most common cases is when a mobile application utilizes certificate pinning. In such a scenario, when we still want to keep working with the other parts of the application, we can add the host in the SSL pass-through list.

---

For More Information:

www.packtpub.com/networking-and-servers/burp-suite-essentials
SSL and Other Advanced Settings

This can be automated by checking an option, which will kick in as soon as Burp encounters an SSL negotiation error. Have a look at the following screenshot:

<table>
<thead>
<tr>
<th>Add</th>
<th>Enabled</th>
<th>Host / IP range</th>
<th>Port</th>
</tr>
</thead>
</table>

- Automatically add entries on client SSL negotiation failure

Invisible Proxy

Sometimes, while intercepting a thick client, you might need to enable Invisible Proxy. A thick client is a software that usually runs outside of the browser framework. This means that some of this software doesn’t have an option for HTTP proxies. When the client is not proxy-aware and is incapable of sending requests that are used by a proxy, such as Burp, we need to use the option of Invisible Proxy.

Since a thick client has no proxy options, we need to trick it into sending all its traffic to the machine where the Burp proxy can listen. For example, if the nonproxy-aware thick client needed to connect to https://example.com, this is what we need to do:

1. Add a mapping for a domain to the loopback IP address in the default Hosts file. This file is usually found in the following paths for Windows and Linux / OS X:
   - Windows/System32/drivers/etc/hosts
   - /etc/hosts

The mapping will look like this:

```
127.0.0.1 example.com
```
2. Once this is set, we need to add a new listener running on the default port for HTTP TCP port number 80 or, if the traffic was meant to be over HTTPS, then TCP port number 443. Most likely, you need to be a privileged user on the system to be able to listen on those two ports, as shown in the following screenshot:

3. If the expected traffic is going to be over SSL, we need to ensure that we can present an SSL certificate to the thick client with an accurate domain name, as shown in the following screenshot:

For More Information:
www.packtpub.com/networking-and-servers/burp-suite-essentials
4. Now, we need to send the traffic from Burp to the original server, which is expecting it. We set this in **Options | Connections | Hostname Resolution**. We can't miss this step because we told the operating system to send all the traffic meant for `example.com` to the loopback interface earlier. Have a look at the following screenshot:

```
<table>
<thead>
<tr>
<th>Enabled</th>
<th>Hostname</th>
<th>A   IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>example.com</td>
<td>93.184.216.119</td>
</tr>
</tbody>
</table>
```

5. At this point, we have tricked the thick client (the nonproxy-aware application) to send all the traffic meant for `example.com` to the Burp Suite listener, and if required, present a correct SSL certificate as well. Once the traffic reaches Burp Suite, it can send it on its way to `example.com`.

6. There is a small caveat in all of this. Burp Suite uses the `Host` header in the request to figure out where to send the request further. If the `Host` header is not present in the request (rare, but can happen), we can configure Burp Suite to send all the traffic reaching a particular listener on to another server.

7. If the traffic is meant for multiple servers and we need to see all of it, then the only suggested solution is to create additional virtual interfaces where we can start loopback listeners, and if that is not an option, start Burp Suite on multiple computers to do what we did.

**Summary**

After this chapter, we can intercept SSL-enabled traffic for any website. While using SSL/TLS certificates is desirable in terms of security, it does pose a challenge when we wish to use an interception proxy, such as Burp, to test the website for flaws.

Burp provides a simple interface to set up SSL/TLS connections with minimal fuss. Once a root certificate authority is imported, all certificates generated by Burp and signed by the same root CA are identified as valid in the browsers. With this configuration, we have basically covered all that we needed to move on to learning about the various tools of Burp Suite that make it such a powerful tool to security test applications that work over HTTP.

---

For More Information:

www.packtpub.com/networking-and-servers/burp-suite-essentials
Where to buy this book

You can buy Burp Suite Essentials from the Packt Publishing website:

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.