### 8.3.1 Remote user to a web server:

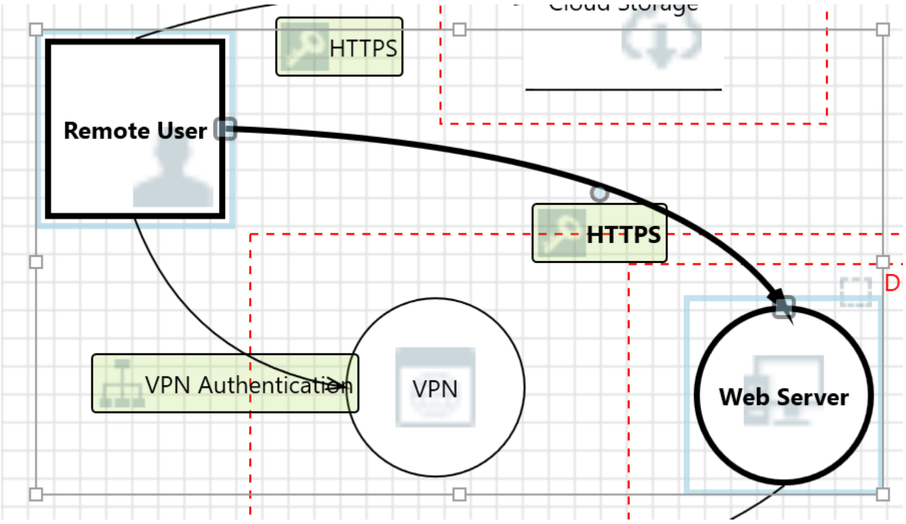

Figure 6: Remote user to Web Server

Figure 6 shows the interaction between a remote user and a web server. Below are threats and mitigations:

**Spoofing the Remote User External Entity**

| | |
|---|---|
| **Category:** | Spoofing |
| **Description:** | Remote User may be spoofed by an attacker and this may lead to unauthorised access to Web Server. Consider using a standard authentication mechanism to identify the external entity. |
| **Mitigation:** | User authentication will be done through Single Sign On (SSO) with MFA enabled for all external connections (Grassi et al., 2017). |

**Cross Site Scripting**

| | |
|---|---|
| **Category:** | Tampering |
| **Description:** | The web server 'Web Server' could be a subject to a cross-site scripting attack because it does not sanitise untrusted input. |
| **Mitigation:** | Access to the Web Server will be protected with Web Application Firewall. A WAF can inspect the input data from user requests and apply strict validation rules. It checks for suspicious or malicious content, such as JavaScript code or HTML tags, and blocks or sanitises them to prevent XSS attacks (Sundar, 2019). |

## Potential Data Repudiation by Web Server

| | |
|---|---|
| **Category:** | Repudiation |
| **Description:** | Web Server claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data. |
| **Mitigation:** | WAF will protect web server access. WAF will provide details of all the access to the webserver, including time of access, IP address and resource accesses. |

### Data Flow HTTPS Is Potentially Interrupted

| | |
|---|---|
| **Category:** | Denial Of Service |
| **Description:** | An external agent interrupts data flowing across a trust boundary in either direction. |
| **Mitigation:** | Strong encryption protocols, such as Transport Layer Security (TLS) or Secure Sockets Layer (SSL), can help protect data in transit and prevent unauthorised interception. Strong Authentication using MFA will also help protect against this attack (blog.hypr.com, n.d.). |

### Elevation Using Impersonation

| | |
|---|---|
| **Category:** | Elevation Of Privilege |
| **Description:** | Web Server may be able to impersonate the context of Remote User in order to gain additional privilege. |
| **Mitigation:** | User authentication will be done through Single Sign On (SSO) with MFA enabled for all external connections (Grassi et al., 2017). |

## Web Server May be Subject to Elevation of Privilege Using Remote Code Execution

| | |
|---|---|
| **Category:** | Elevation Of Privilege |
| **Description:** | Remote User may be able to remotely execute code for Web Server. |
| **Mitigation:** | A key principle of Zero Trust Access is to grant the user only the necessary privileges, known as the least privilege approach. |

### Elevation by Changing the Execution Flow in Web Server

| | |
|---|---|
| **Category:** | Elevation Of Privilege |

| | |
|---|---|
| **Description:** | An attacker may pass data into Web Server in order to change the flow of program execution within Web Server to the attacker's choosing. |
| **Mitigation:** | Access to the Web Server will be protected with Web Application Firewall. A WAF can inspect the input data from user requests and apply strict validation rules. It checks for suspicious or malicious content, such as JavaScript code or HTML tags, and blocks or sanitises them to prevent code injection attacks. |

**Cross Site Request Forgery**

| | |
|---|---|
| **Category:** | Elevation Of Privilege |
| **Description:** | Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting. The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations. |
| **Mitigation:** | Access to the Web Server will be protected with Web Application Firewall. A WAF can inspect incoming requests to the web application and validate them against predefined rules. It can check for specific headers, tokens, or parameters that are typically used to prevent CSRF attacks, such as anti-CSRF tokens. |