Common Attack Pattern Enumeration and Classification (CAPEC)

Adversarial Behaviours

CAPEC

Adversarial Behaviours

- The motivation for Common Attack Pattern Enumeration and Classification (CAPEC) is to **better understand adversaries**.
- A central aim is to **standardise the language** and **improve information sharing** between companies, both in terms of attack to systems and internal investigations.
- MITRE Corporation advocates attack patterns as **effective way to communicate attacker the perspective**.
- Distill from the consideration of **software and systems in the wild** and aimed at strengthening systems.
- MITRE Corporation maintains a public **catalogue** of attack patterns in a similar vain to software engineering patterns.

Attack Patterns

CAPEC

- Attack patterns can be thought of as a standardise language or blueprint to discuss specific types of attacks.

- Focus on the perspective of the adversary when considering the development of systems, deployment of systems and the use of systems.

Attack Patterns

CAPEC

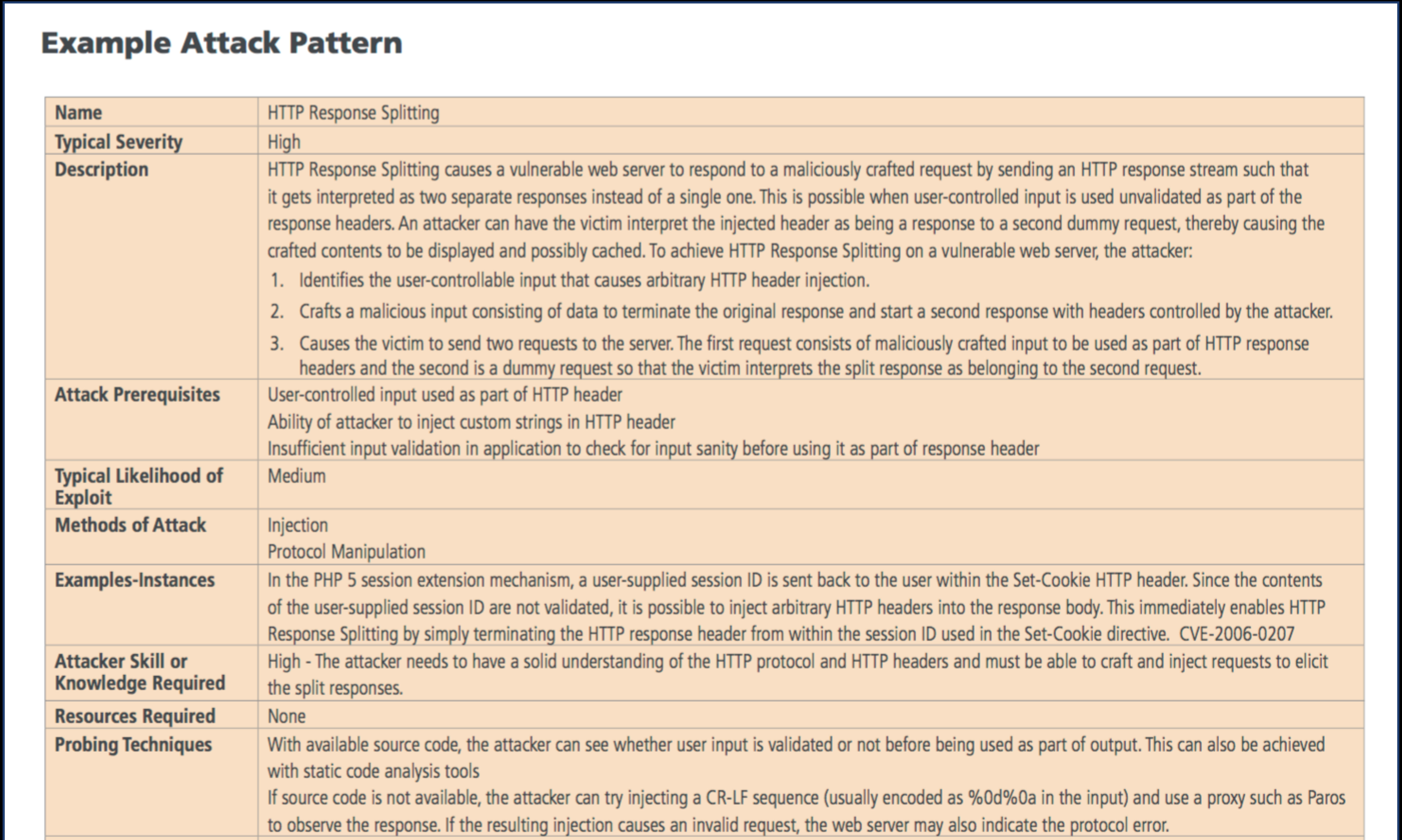
- Attack patterns can be classified in of *architecture*, *artefact* and *external*.
    - *architecture* refers to the elements and connections itself, includes protocols and processes.
    - *artefact* refers to actual source, platform and specifics of the actual system.
    - *external* attack patterns can be considered a grab bag of attacks that exploit various aspects of architecture and artefacts, for example viruses and worms.

Attack Patterns

CAPEC

Attack Patterns

CAPEC

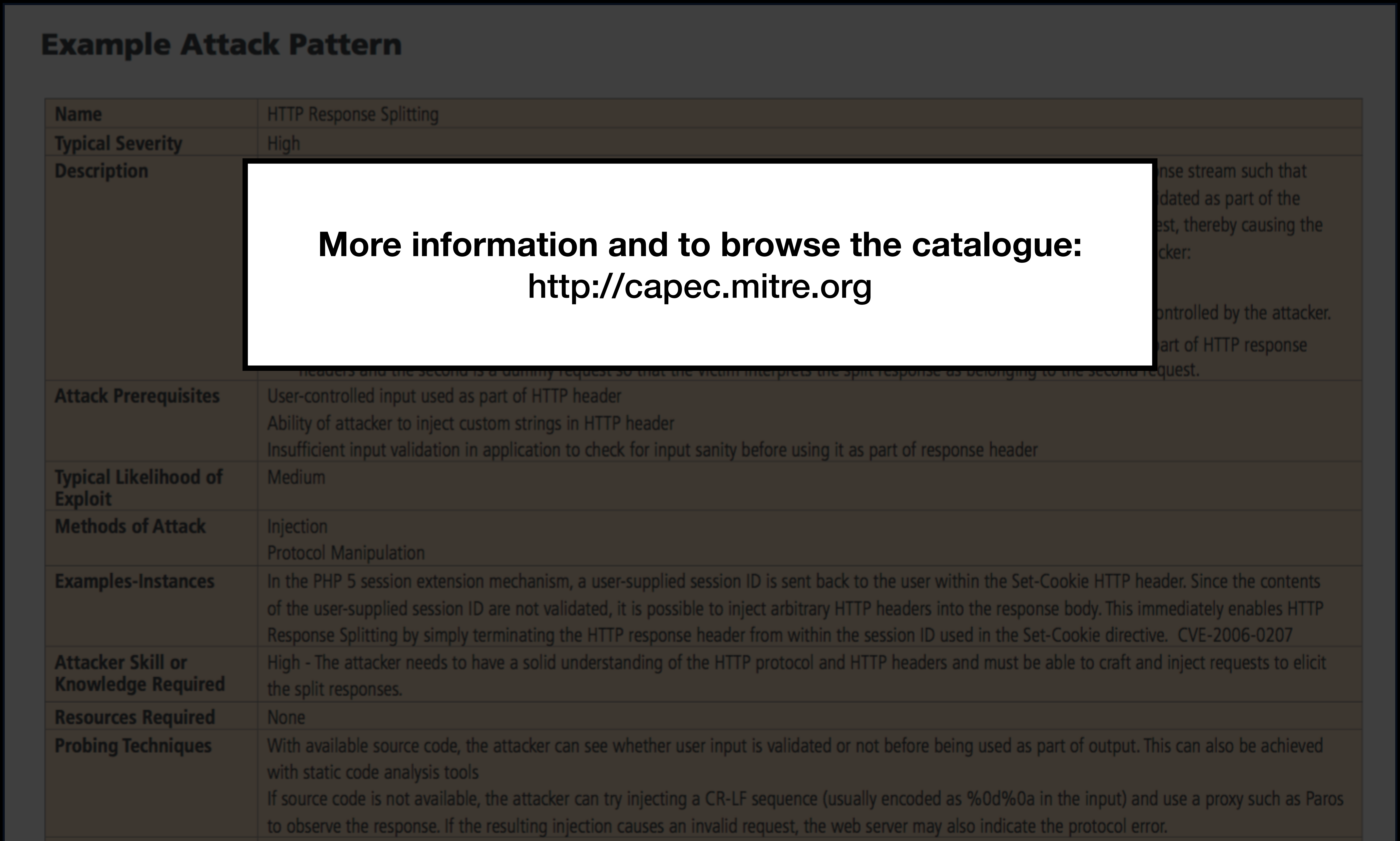


Example Attack Pattern

Name	HTTP Response Splitting
Typical Severity	High
Description	HTTP Response Splitting causes a vulnerable web server to respond to a maliciously crafted request by sending an HTTP response stream such that it gets interpreted as two separate responses instead of a single one. This is possible when user-controlled input is used unvalidated as part of the response headers. An attacker can have the victim interpret the injected header as being a response to a second dummy request, thereby causing the crafted contents to be displayed and possibly cached. To achieve HTTP Response Splitting on a vulnerable web server, the attacker: 1. Identifies the user-controllable input that causes arbitrary HTTP header injection. 2. Crafts a malicious input consisting of data to terminate the original response and start a second response with headers controlled by the attacker. 3. Causes the victim to send two requests to the server. The first request consists of maliciously crafted input to be used as part of HTTP response headers and the second is a dummy request so that the victim interprets the split response as belonging to the second request.
Attack Prerequisites	User-controlled input used as part of HTTP header Ability of attacker to inject custom strings in HTTP header Insufficient input validation in application to check for input sanity before using it as part of response header
Typical Likelihood of Exploit	Medium
Methods of Attack	Injection Protocol Manipulation
Examples-Instances	In the PHP 5 session extension mechanism, a user-supplied session ID is sent back to the user within the Set-Cookie HTTP header. Since the contents of the user-supplied session ID are not validated, it is possible to inject arbitrary HTTP headers into the response body. This immediately enables HTTP Response Splitting by simply terminating the HTTP response header from within the session ID used in the Set-Cookie directive. CVE-2006-0207
Attacker Skill or Knowledge Required	High - The attacker needs to have a solid understanding of the HTTP protocol and HTTP headers and must be able to craft and inject requests to elicit the split responses.
Resources Required	None
Probing Techniques	With available source code, the attacker can see whether user input is validated or not before being used as part of output. This can also be achieved with static code analysis tools If source code is not available, the attacker can try injecting a CR-LF sequence (usually encoded as %0d%0a in the input) and use a proxy such as Paros to observe the response. If the resulting injection causes an invalid request, the web server may also indicate the protocol error.

Attack Patterns

CAPEC



Example Attack Pattern

Name	HTTP Response Splitting
Typical Severity	High
Description	nse stream such that idated as part of the est, thereby causing the cker: ontrolled by the attacker. art of HTTP response headers and the second is a dummy request so that the victim interprets the split response as belonging to the second request.
Attack Prerequisites	User-controlled input used as part of HTTP header Ability of attacker to inject custom strings in HTTP header Insufficient input validation in application to check for input sanity before using it as part of response header
Typical Likelihood of Exploit	Medium
Methods of Attack	Injection Protocol Manipulation
Examples-Instances	In the PHP 5 session extension mechanism, a user-supplied session ID is sent back to the user within the Set-Cookie HTTP header. Since the contents of the user-supplied session ID are not validated, it is possible to inject arbitrary HTTP headers into the response body. This immediately enables HTTP Response Splitting by simply terminating the HTTP response header from within the session ID used in the Set-Cookie directive. CVE-2006-0207
Attacker Skill or Knowledge Required	High - The attacker needs to have a solid understanding of the HTTP protocol and HTTP headers and must be able to craft and inject requests to elicit the split responses.
Resources Required	None
Probing Techniques	With available source code, the attacker can see whether user input is validated or not before being used as part of output. This can also be achieved with static code analysis tools If source code is not available, the attacker can try injecting a CR-LF sequence (usually encoded as %0d%0a in the input) and use a proxy such as Paros to observe the response. If the resulting injection causes an invalid request, the web server may also indicate the protocol error.

More information and to browse the catalogue:
http://capec.mitre.org

Attack Patterns

CAPEC

- CAPEC is not a **modelling technique** and **would not be used to formulate the anatomy of a cyber attack**.
- CAPEC is designed to largely be a resource to support secure software development.
- CAPEC cements **models in evidence** and affords further detail on what aspects to focus on.
- CAPEC **does not prioritise attacks** and as such is not particularly valuable in guiding the specific attacks that should be focused on.
- CAPEC is **largely technically focused** and can be limiting when it comes to some non-technical attacks.

Common Attack Pattern Enumeration and Classification (CAPEC)

Adversarial Behaviours