CVE-2021-3737

EP3 – Redes de Computadores

Beatriz Costa e Cássio Cancio

Índice

01

A falha

Resumo da falha e onde ocorre no código fonte

A falha

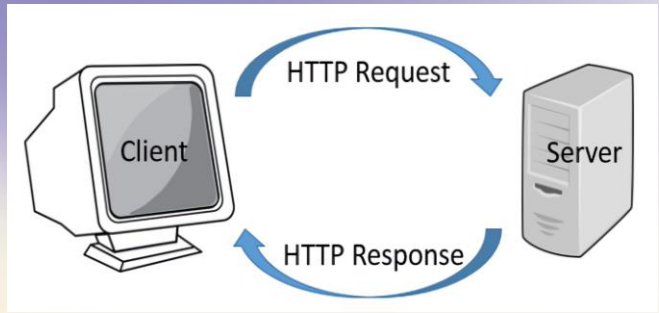
HTTP (Hypertext Transfer Protocol)

Camada de aplicação da web;

Base de qualquer troca de dados na web;

É protocolo cliente-servidor, as requisições são iniciadas pelo destinatário, geralmente um navegador da Web;

As requisições são enviadas pelo cliente e o servidor as responde.

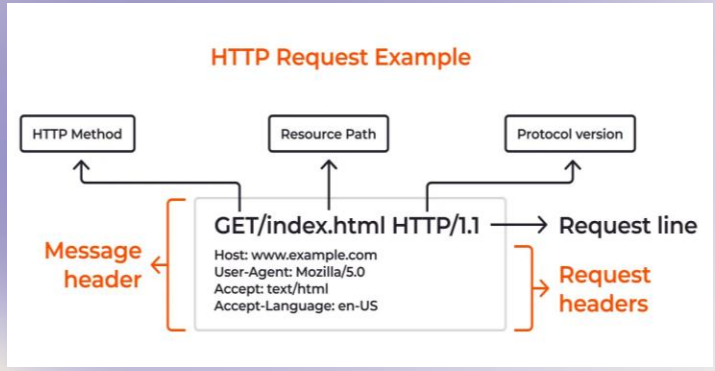


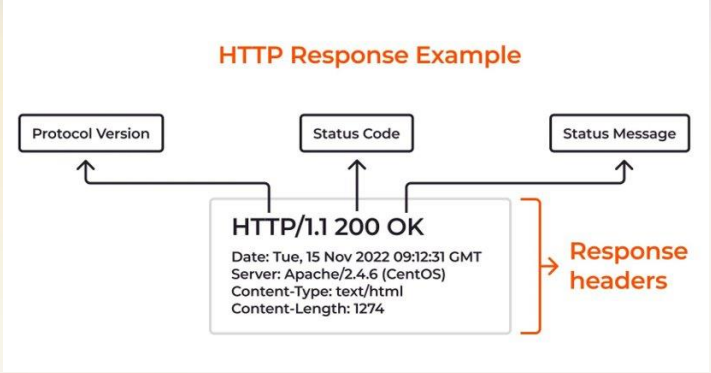
A falha

Abre uma conexão TCP

A conexão é usada para enviar requisições e receber uma resposta

O cliente pode abrir uma nova conexão, reusar uma existente, ou abrir várias conexões





A falha

A falha foi encontrada em python

Publicada em 8 de agosto de 2021

Um erro, no tratamento de respostas HTTP, no cliente, permite ao servidor enviar um script que faz o cliente entrar em loop, consumindo CPU

Ameaça a disponibilidade do sistema do cliente

python

cpython/Lib/http/client.py

```
...

while True:
    version, status, reason = self._read_status()
    if status != CONTINUE:
        break
    # skip the header from the 100 response
    while True:
        skip = self.fp.readline(_MAXLINE + 1)
        if len(skip) > _MAXLINE:
            raise LineTooLong("header line")
        skip = skip.strip()
        if not skip:
            break
        if self.debuglevel > 0:
            print("header:", skip)

...
```

02

Explorar

Mostrar a falha ocorrendo com um caso de teste

Demonstração

03

Corrigir

Mostrar a correção em detrimento da falha

cpython/Lib/http/client.py

```
...

while True:
    version, status, reason = self._read_status()
    if status != CONTINUE:
        break
    # skip the header from the 100 response
    while True:
        skip = self.fp.readline(_MAXLINE + 1)
        if len(skip) > _MAXLINE:
            raise LineTooLong("header line")
        skip = skip.strip()
        if not skip:
            break
        if self.debuglevel > 0:
            print("header:", skip)

...
```

cpython/Lib/http/client.py

```
...



while True:
    version, status, reason = self._read_status()
    if status != CONTINUE:
        break
    # skip the header from the 100 response
    skipped_headers = _read_headers(self.fp)
    if self.debuglevel > 0:
        print("headers:", skipped_headers)
    del skipped_headers



...
```

cpython/Lib/http/client.py

```
...

def _read_headers(fp):
    """Reads potential header lines into a list from a file pointer.
    Length of line is limited by _MAXLINE, and number of
    headers is limited by _MAXHEADERS.
    """
    headers = []
    while True:
        line = fp.readline(_MAXLINE + 1)
        if len(line) > _MAXLINE:
            raise LineTooLong("header line")
        headers.append(line)
        if len(headers) > _MAXHEADERS:
            raise HTTPException("got more than %d headers" % _MAXHEADERS)
        if line in (b'\r\n', b'\n', b''):
            break
    return headers

...
```

04

Verificar

Mostrar que a falha não ocorre no caso de teste

Demonstração

Obrigado