# Availability

Source: Malicious User

Stimulus: Dos Attack

Artifact: Sever

Environment: Normal run condition

Response: It should cancel the connections that the attacker has

Response measure: How long server can survive before crashing

Source: Users wanting to make requests

Stimulus: A user makes a request

Artifact: Server

Environment: Normal run conditions

Response: Server handles requests in parallel

Response measure: Total memory consumed by application

# Performance

Source: A user who wants a file

Stimulus: A user requests a file

Artifact: Sever

Environment: Normal run condition

Response: The server serves the file

Response measure: The time between request and serving the content

Source: A malicious user

Stimulus: A dos attack on our server

Artifact: Sever

Environment: stressed run condition

Response: The server stops the dos attack to allow other users to not get slower performance

Response measure: Latency for legitimate client under dos attack

# Security

Source: A malicious user

Stimulus: A dos attack on our server

Artifact: Server

Environment: Normal run condition

Response: The server stops the dos attack to allow other users to not get slower performance

Response measure: How long the survive before crashing

Source: A malicious user

Stimulus: User uploads virus

Artifact: Sever

Environment: Normal run condition

Response: The server should serve the request

Response measure: The server maintains an audit trail so the managers can find the malicious user

# Tactics For Improvement

## Dos Attack – Tactic for Availability, Performance, and Security

After a threshold number of connections we ban the user from being able to make new connections and cancel the current connections. This threshold is based upon number of current connections, we allow each user to have a maximum number of concurrent connections. If a user exceeds this number of concurrent connections they will be blacklisted for a time period.

## Parallel Requests – Tactic for Availability

Tactic is to increase the capacity number of threads that can be running simultaneously on the server. This means limiting the memory impact of each individual thread. Implement the server such that it never needs to have entire files in memory or read entire contents of incoming requests before starting to service the connection. Increasing the total memory of the server could also be used for this tactic.

## Time Between Request and Content – Tactic for Performance

This tactic is to implement a caching service for the server. For the most commonly requested resources, it should be in the cache, in memory, instead of needing to be read from disk with each request.

## Audit Trail – Tactic for Security

In order to track and trace the actions of potentially malicious individuals, maintain a log of all connections, requests, and origins such that they can be looked up at a later date to determine the actions of any user on the server. This will be implemented as a log file.

# Testing Plan

## Dos Attack

We first use the dos client to attack the server to see how many connections the user can generate and have the server attempt to serve. Then we implement the improvement and see how many connections the server will serve the malicious user.

## Parallel Requests

We use the dos client to generate a lot of heavy resource requests to the server and see how many connections the server can handle at once. Then we will implement our fix tactic and see how many connections we can maintain and serve.

## Time Between Request and Content

We generate back to back requests of similar files and measure the time the server takes to responds. Then we implement our cashing technique and send the requests again and check to see if we get a better overall average response time.

## Audit Trail

Tester 1 does some actions and tester 2 has to tell what different actions the first tester did. If he can successful do this for a variety of request then our audit system works. (he can send requests from different ip addresses)

Dos Attack Test Results:

Server seems to survive indefinitely without crashing, however while performing a SYN flood, it takes 13.12 seconds of time before the server starts rejecting connections periodically. 1000 requests per second.

It takes around 19 seconds for the system to start rejecting a significant number of connections.

Connection Latency No DOS:

Initial Connection between 5ms and 7ms

Average connection latency after 100 requests of index page: 0.48623853211009177ms

Connection Latency with DOS:

Average connection latency after 30 request while DOS is hit: 11.072162087149598ms

Memory Usage:

During a DOS style attack, 18 threads run with an allocated memory of ~160MB on the heap of which it peaks to 40MB used.

Requesting a single large file uses a single thread which used a maximum of 3 MB change from the server when not requesting anything. (Difference from 33MB to 36MB) when requesting a video file of 118MB.