# www.theforage.com - Telstra Cyber Task 3

# Model Work Example

# Firewall Server Handler

from http.server import BaseHTTPRequestHandler, HTTPServer

host = "localhost"

port = 8000

def block\_request(self):

self.send\_error(403, "Request blocked due to firewall")

def handle\_request(self):

# List of bad headers from the proof of concept payload

bad\_headers = {

"suffix": "%>//",

"c1": "Runtime",

"c2": "<%",

"DNT": "1",

"Content-Type": "application/x-www-form-urlencoded",

}

bad\_header\_keys = bad\_headers.keys()

# If a request is on the Spring Framework path

if self.path == "/tomcatwar.jsp":

# Iterate through bad headers

for bad\_header\_key in bad\_header\_keys:

# If we find a bad header that matches the malicious payload

if bad\_header\_key in self.headers and self.headers[bad\_header\_key] == bad\_headers[bad\_header\_key]:

# Block request and throw 403 error

return block\_request(self)

# Return successful response

self.send\_response(200)

self.send\_header("content-type", "application/json")

self.end\_headers()

self.wfile.write({ "success": True })

class ServerHandler(BaseHTTPRequestHandler):

def do\_GET(self):

handle\_request(self)

def do\_POST(self):

handle\_request(self)

if \_\_name\_\_ == "\_\_main\_\_":

server = HTTPServer((host, port), ServerHandler)

print("[+] Firewall Server")

print("[+] HTTP Web Server running on: %s:%s" % (host, port))

try:

server.serve\_forever()

except KeyboardInterrupt:

pass

server.server\_close()

print("[+] Server terminated. Exiting...")

exit(0)