mmn15 q2 – Defensive Systems Programming – Daniel Greiner

Threat	Client impersonation
Affected component	Authentication and files
Vulnerability class	Sensitive information in plaintext
Description	In order to connect through a client all a user need is the client name and the uuid, which are both transmitted without being encrypted. So, an attacker might just grab a packet, catch the cardentials, and use them to impersonate the client
Result	Attacker has client's cardentials, he may overwrite user files (which is the worst) but also change all the keys and stuff of the client. Basically everything that the client can do, he can do (maybe not everything because he cannot re-register, but he can steal the first packet so yeah).
Prerequisites	Being able to connect to the network and sniff the packets
Business impact	Possible loss of client data, or transforming client data without him knowing. Client's data might get exposed and his account might become inacceisble if the attacker nesses with the keys and stuff
Proposed remediation	Encrypt everything
risk	Damage potential: 9 Reproducibility: 8 Exploitability: 10 Affected users: 10 Discoverability: 10 Overall: 9

Threat	Client impersonation
Affected component	Authentication, files
Vulnerability class	Weak cardentials
Description	In order to login we need client name and uuid, client name is easy to guess (because it is literally the client's name), and the uuid might also be easy to guess because there is no requirement to use a secure way to generate it. Moreover, all the cardentils are stored in the victim's machine as plaintext, so if we have something in it we may read them.
Result	Same as the vuln above
Prerequisites	Being able to connect to the network and see traffic, or being able to get user's cardentils file
Business impact	Same as the vuln above
Proposed remediation	Use stronger authentication tools, keep everything in secret, don't rely on no one, not on the client keeping his cardentils safe, and not on the server to implement a secure uuid
risk	Damage potential: 8 Reproducibility: 7 Exploitability: 9 Affected users: 8 Discoverability: 8 Overall: 8

Threat	Server impersonation
Affected component	Authentication, files
Vulnerability class	Server could be any one
Description	Basically, MITM. An attacker may pretend to be the host, meanwhile getting all client's packets and delivering them to the real host, and vice versa. This way, the attacker can do literally anything that the client or the server can do, to the other side.
Result	Attacker can be the client to the server, or the server to the client, so basically anything. All the problems in the vulns above are present here, and also more because we don't need the client almost at all.
Prerequisites	Attacker can connect to the network and preform a MITM attack
Business impact	Client's and server's cardentials and secret info are being sent through a third party without their knowledge
Proposed remediation	Use certifications in the server side and validations in the client side, in order for each to know for sure that he is talking to the real other side.
risk	Damage potential: 9 Reproducibility: 8 Exploitability: 6 Affected users: 7 Discoverability: 9 Overall: 7

Threat	Dos or ddos (making the server inaccessible)
Affected component	Server
Vulnerability class	Dos, ddos
Description	Protocol doesn't limit connections or connection rates and things like that, which may be exploited by sending tons of packets to the server, making it –
Result	Unavailable! It will collapse and be inaccessible
Prerequisites	Being able to send lots of packets to the server
Business impact	Server unavailable
Proposed remediation	Limit number of connections to a small number so not too many clients can connect. Limit rate of requests from users in a certain interval.
risk	Damage potential: 6 Reproducibility: 6 Exploitability: 8 Affected users: 10 Discoverability: 10 Overall: 5 (ddos is for losers)

Threat	Replay attack
Affected component	Authentication
Vulnerability class	Replay attack
Description	A malicious actor can intercept and capture packets, then transmit them to trick the server into thinking they are new and preform actions again and again
Result	Changing keys (if sending again reconnection packets), override user files (if send again a file that has been overwritten by the client), etc.
Prerequisites	Being able to connect to the network, read packets and send packets.
Business impact	A bit like user impersonation but weaker
Proposed remediation	Use session tokens, timestamps, or unique values for each request in order to not be able to send one twice
risk	Damage potential: 8 Reproducibility: 6 Exploitability: 7 Affected users: 6 Discoverability: 5 Overall: 6

Threat	Override server's data with path injection
Affected component	File management in server (not really a vuln in the protocol, it depends on how you implement it. You can do it safe but you don't have to)
Vulnerability class	Path injection
Description	Any client can send any file name to write to, including names like "../../../../etc/passwd" which will go back and back in the filesystem and overwrite files if the server and not of the client
Result	Server's data being overwritten, possibly everything, even RCE (inject code to a dll) or changing setting and stuff
Prerequisites	Being able to be a client
Business impact	Living hell. Basically control the server, not only in the sense of the network connections but also the server's machine itself
Proposed remediation	Check file name to not include any "/"s
risk	Damage potential: 10 Reproducibility: 5 Exploitability: 10 Affected users: 10 Discoverability: 10 Overall: 10 (but not in reality cuz its really easy to check for it