

NETWORK AND COMPUTER SECURITY

RETAIL: GROOVEGALAXY



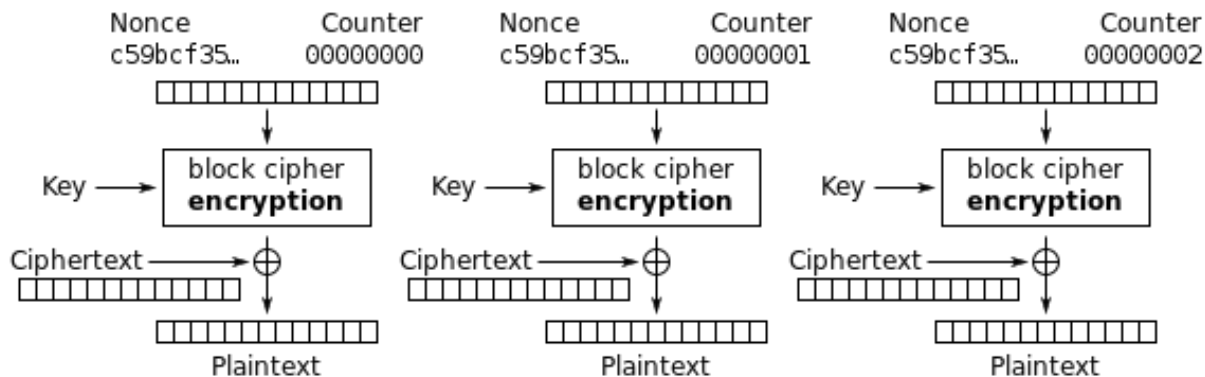
99236 Inês Pissarra
99261 Juliana Marcelino
99275 Mário Santos

SECURE DOCUMENT LIBRARY AND TOOL

SECURE DOCUMENT LIBRARY AND TOOL

- Protect with CTR

This encryption method allows you to decrypt content that is in the middle of the file.



Counter (CTR) mode decryption

SECURE DOCUMENT LIBRARY AND TOOL

- Protect with Nonce

To ensure the freshness, a nonce composed of a counter and a timestamp is implemented. This prevents replay attacks and old messages from being processed.



SECURE DOCUMENT LIBRARY AND TOOL

- Protect with MAC

To ensure the integrity of the file, we used a MAC (with 256 bits) algorithm.

- Encrypt-then-MAC:

It was used EtM method so that we can test the integrity of the file without having to decrypt it

SECURE DOCUMENT LIBRARY AND TOOL

- Check

Uses MAC algorithm and checks the freshness of the file by comparing the nonce components, counter and timestamp with the current ones.

- Unprotect

Reverses the protection process. Calls the check class to verify the integrity, and decrypts the file content using AES in CTR mode with the key and initialization vector (this last provided on the JSON).

SECURE DOCUMENT LIBRARY AND TOOL

Encrypted File Structure:

iv: _____

encrypted File: _____

Nonce:

Counter: _____

Timestamp: _____

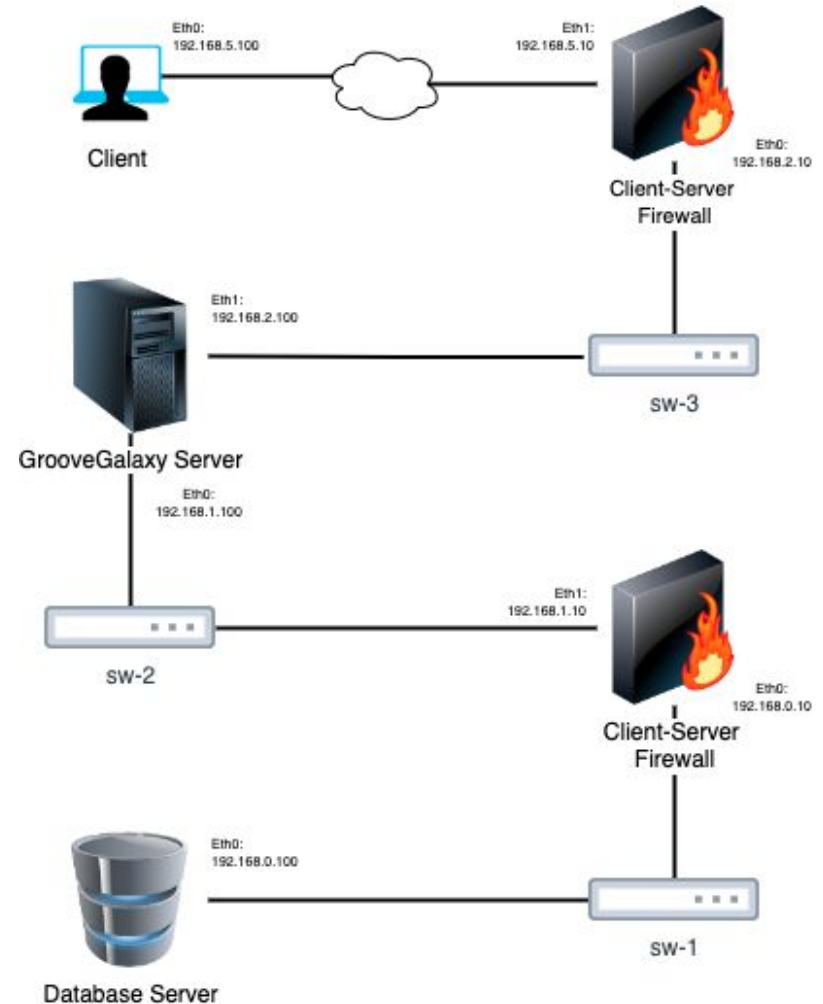
MAC: (all previous data encrypted)

STRUCTURE OF THE SYSTEM

STRUCTURE OF THE SYSTEM

Virtual machines each with one of the following (total of 5 VMs):

- 1 Client
- 2 Firewalls
- 2 Servers (1 Main and 1 Database)



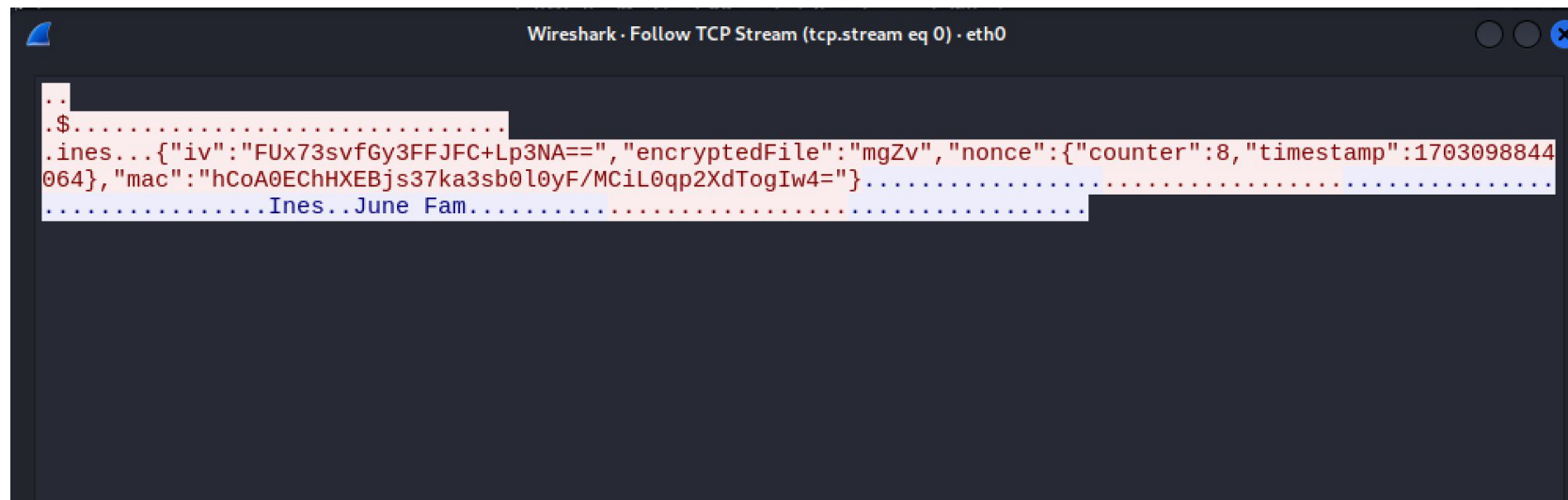
SECURE CHANNELS

SECURE CHANNELS

- Client and the server communicate via gRPC
- Both client-server and server-database communications use Transport Layer Security (TLS) and one CA for each communication.

SECURE CHANNELS

Encryption without TLS (client-server)

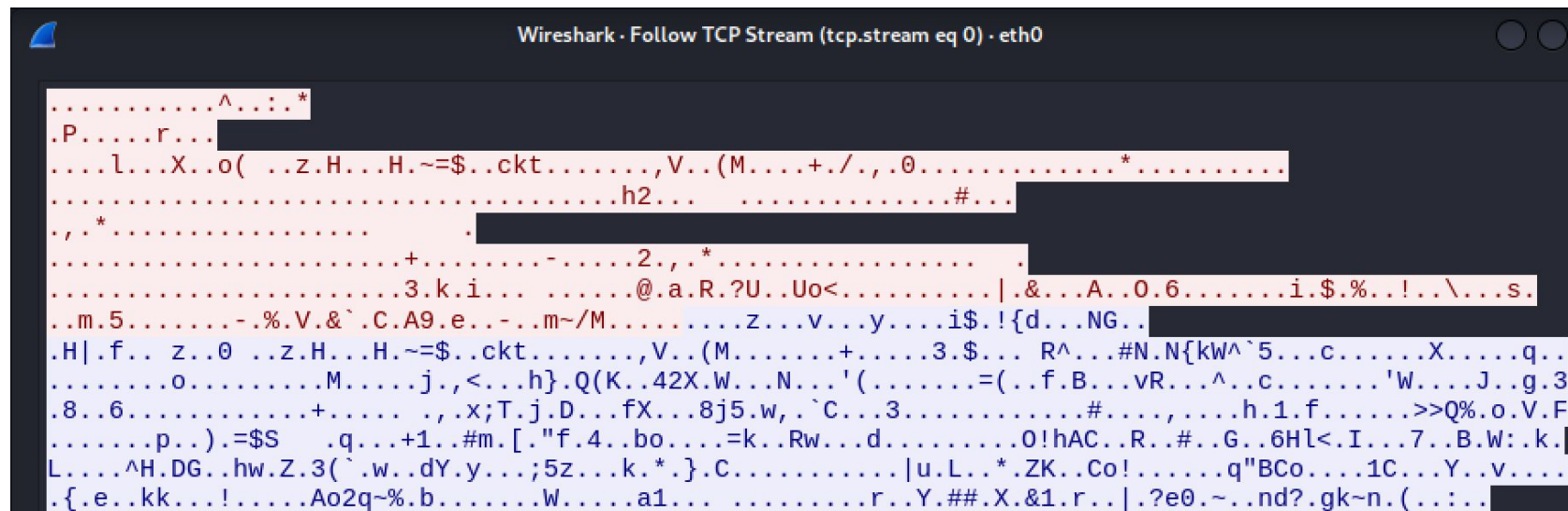


The image shows a Wireshark window titled "Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0". The packet list on the left shows a single packet of type "Hypertext Transfer Protocol". The packet details pane on the right shows the "Hypertext Transfer Protocol" section expanded, displaying a JSON object. The JSON object contains fields for "iv", "encryptedFile", "nonce", and "mac". The "encryptedFile" field is highlighted in red. The "nonce" field is a JSON object with "counter" and "timestamp" fields. The "mac" field is a long hexadecimal string. The packet bytes pane at the bottom shows the raw data of the packet, which is mostly obscured by a large redacted area.

```
..  
.$.....  
.ines...{"iv":"FUx73svfGy3FFJFC+Lp3NA==", "encryptedFile":"mgZv", "nonce":{"counter":8, "timestamp":1703098844  
064}, "mac":"hCoA0EChHXEBjs37ka3sb0l0yF/MCiL0qp2XdTogIw4="}.....  
.....Ines..June Fam.....
```

SECURE CHANNELS

Encryption with TLS (client-server)



Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0

```
.....^...*
.P....r...
...l...X...o( ..z.H...H.~=$..ckt.....,V..(M...+./.,.0.....*.....
.....h2... ..#...
.,*.....
.....+.....2.,*.....
.....3.k.i.....@.a.R.?U..Uo<.....|.&...A..0.6.....i.$.%..!..\...s.
..m.5.....-.%.V.&`.C.A9.e.-..m~/M.....z...v...y...i$..!{d...NG..
.H|.f.. z..0 ..z.H...H.~=$..ckt.....,V..(M...+.....3.$... R^...#N.N{kW^`5...c.....X.....q..
.....o.....M.....j.,<...h}.Q(K..42X.W...N...'('.....=(.f.B...vR...^..c.....'W....J..g.3
.8..6.....+.....,x;T.j.D...fX...8j5.w,`C...3.....#.....,.....h.1.f.....>Q%.o.V.F
.....p..).=$S ..q...+1..#m.[."f.4..bo....=k..Rw...d.....0!hAC..R..#.G..6Hl<.I...7..B.W:.k.
L....^H.DG..hw.Z.3(`.w..dY.y...;5z...k.*.}.C.....|u.L...*.ZK..Co!.....q"BCo....1C...Y..v...
.{.e..kk...!.....Ao2q~%.b.....W.....a1.....r..Y.##.X.&1.r..|. ?e0.~..nd?.gk~n.(.....
```

SECURITY CHALLENGE

FAMILY SHARING

Each family has its own key that is sent to the family members when they join the family. The keys are encrypted with the member's key when distributed.

This way, each family member has their own key and the family key.



PLAYBACK IN THE MIDDLE OF AN AUDIO STREAM

Using CTR we can decrypt from the middle of a file, so it's possible to decrypt the song from any place we want.

It works like a normal command “play” but when unprotecting the file for the client, we securely decrypt the part from where the client wants until the end, leaving the beginning encrypted.



DEMONSTRATION