Agentic Framework for Automated Security Analysis in Multi-User Extended Reality

Mutahar Ali, Habiba Farrukh

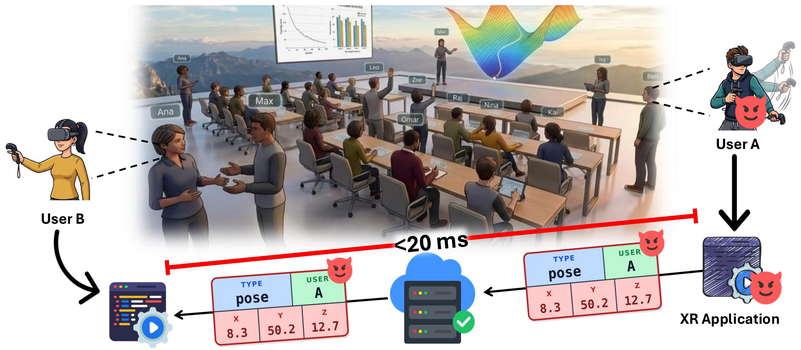
UC Irvine

acm XR Security '26

Problem

Multi-user XR apps

- Groups of users share one 3D space in real time, each represented by an avatar.
- Used for education, training, healthcare, design review and social events.

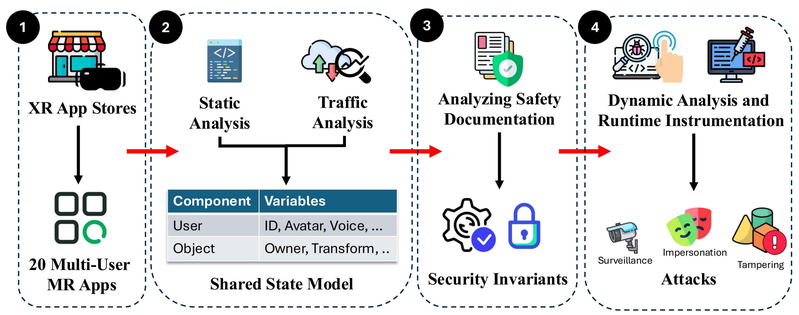


Every state update is relayed to every other participant's headset.

- Perceptual stability needs updates in under **20 ms**, so the relay server just forwards state updates to other clients.
- A malicious user can modify their application to broadcast false state updates.

Can a single malicious participant, running a modified client, subvert the privacy, integrity, and safety of a shared XR session?

Security Analysis [1]



Multi-User XR Apps

4 Platforms

Meta, Android, SteamVR, VIVE

20 Apps

VRChat, Rec Room, Engage, Arkio, Gorilla Tag, VictoryXR

150M+	6M+	280K+
installs	monthly active users	peak concurrent users

Use Cases

Attacks [1]

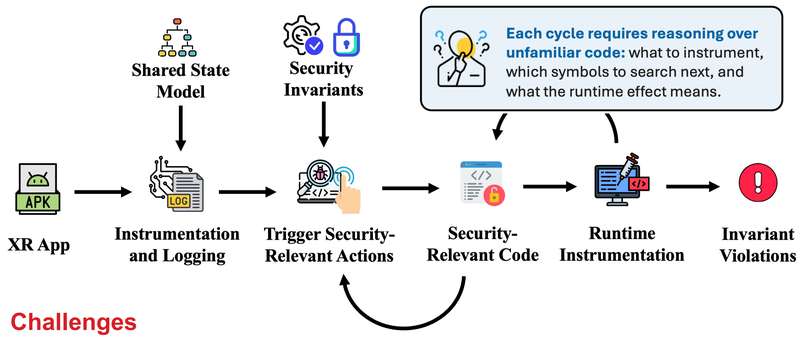
Surveillance
- Video surveillance
- Audio surveillance

Disruption
- Forced Disconnection

Tampering
- Object Manipulation

Safety Bypass
- Blocking Feature
- Host Controls

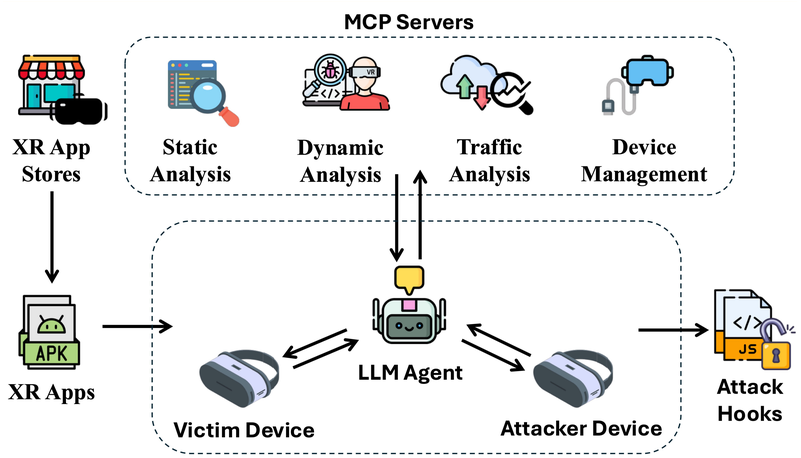
Scaling Challenges



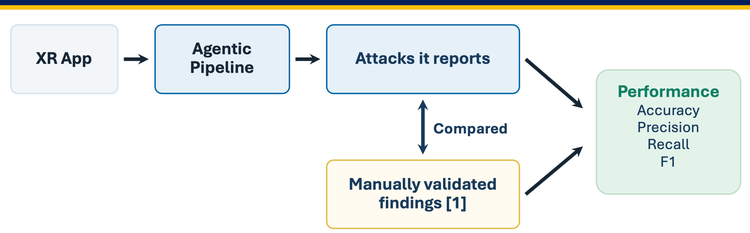
Challenges

- Violations are **semantic**. There is no fixed program signature to search for because the enforcement logic differs in each app.
- Writing instrumentation to violate invariants requires **natural language understanding** and iterative **testing**.

Agentic Framework



Evaluation



Inside One Run

1. Pull the APK, inject the Frida gadget, re-sign, and patch anti-tamper checks. Reinstall and confirm the app runs.
2. Search the decompiled code for pose, sync and broadcast, narrow to three candidates, and load a **read-only logger** on them.

```
PlayerSync.SendState
Photon.OnSerializeView
AvatarRig.Push
```

HUMAN OPERATOR walks around the room

3. One method fired **600 times in 10 s**, about frame rate, and its arguments track the operator. Decompile it.

```
void SendState() {
  net.write(player.pose);
}
```

4. Override the method to always broadcast **one fixed transform**.

```
override SendState() {
  net.write(pos(0,1.7,0), rot(0,0,0));
}
```

HUMAN OPERATOR walks around the room

5. Traffic shows the same pose leaving every frame. On the **victim's** screen, the attacker never moves from the spawn point.

The invariant is violated. The attack is successful.

Results

Attack	Accuracy	Precision	Recall	F1
Video surveillance	1.00	1.00	1.00	1.00
Audio surveillance	0.93	1.00	0.93	0.96
Object manipulation	0.79	1.00	0.79	0.88
Block evasion	0.79	0.57	1.00	0.73
Host-control evasion	0.93	1.00	0.86	0.92
Forced disconnection	0.71	0.57	0.80	0.67
Overall (macro-average)	**0.86**	**0.86**	**0.90**	**0.86**

Takeaways

Client-side trust introduces **security risks** in multi-user MR, and there is a trade-off between security and immersion.

Findings **disclosed** to vendors. Vendors can use this framework to perform **security regression tests** before releases.

Prompts, scripts, and MCP tooling released for reuse and reproducibility.

Defenses
- Selective server-side validation
- Client integrity and attestation
- State minimization

Video Demos of Attacks

[1] Mutahar Ali and Habiba Farrukh. Relay and Betray: Exploiting Client-Side Authority in Multi-User Mixed Reality. USENIX Security 2026.