International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Integrated Security Assessment of WebRTC Video Conferencing Systems: A Complementary Approach Using STRIDE and OWASP ZAP

Si Hwan Kim[1]; Geun Hyeong Kim[2]; Taek Lee[3]

[1,2,3]Division of Computer Science and Engineering, Sun Moon University, Republic of Korea

[1] hwan0212@sunmoon.ac.kr; [2] dko07160@sunmoon.ac.kr; [3] comtaek76@sunmoon.ac.kr
(Corresponding Author: Taek Lee)

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i08.004

Abstract: The rapid expansion of remote collaboration has made video conferencing systems complex web services that combine authentication, session management, media streaming, recording repositories, and external integrations. These systems include not only general web application vulnerabilities, but also structural threats such as meeting-room authorization, WebRTC media paths, IP exposure through ICE candidates, and group-key renewal. This study evaluates an integrated security assessment framework combining design-phase STRIDE threat modeling and runtime OWASP ZAP Baseline Scan for a WebRTC-based video conferencing architecture. While unauthenticated passive dynamic scanning without active attack payloads efficiently identifies deployment misconfigurations and browser-policy weaknesses such as A05 Security Misconfiguration, STRIDE addresses structural and access-control risks such as A01 Broken Access Control and A07 Identification and Authentication Failures. In Experiment A, 9 STRIDE threat elements and DREAD risk-prioritization results derived from Microsoft Threat Modeling Tool analysis were used as design-stage evidence. In Experiment B, the same web interface with security headers applied was reassessed with Docker-based OWASP ZAP Baseline Scan, collecting 4 alerts and 8 instances as runtime evidence. The two results were mapped according to OWASP Top 10:2021 to explain detection scope rather than tool superiority. The analysis showed that STRIDE is useful for identifying logical risks, trust boundaries, and authorization relationships that are difficult to observe from HTTP responses alone, whereas ZAP Baseline Scan is effective for verifying configuration errors in deployed web interfaces. This study interprets the two methods as complementary processes combining design-stage threat-scope definition with implementation-stage configuration verification.

Keywords: OWASP Top 10, OWASP ZAP, STRIDE, Threat Modeling, Video Conferencing

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

I. INTRODUCTION

Video conferencing systems have become essential platforms for remote collaboration, education, and distributed work. Modern WebRTC-based video conferencing systems integrate multiple components and functions, including user authentication, meeting-room authorization, signaling, real-time media transmission, recording services, and data storage. As a result, security risks may arise not only from implementation and deployment errors but also from architectural decisions, trust boundaries, and authorization relationships among system components. Examples include meeting-room access control failures, privilege escalation, ICE candidate IP exposure, insecure media paths, recording-data exposure, and browser security misconfigurations.

This complexity presents an important challenge for security assessment. Different assessment methods provide different types of security evidence. Design-stage threat modeling can identify potential structural and logical risks by analyzing system components, data flows, and trust boundaries. STRIDE, for example, provides a systematic approach for identifying threats related to Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. However, threat modeling cannot directly verify whether a deployed system has been securely implemented or configured.

Conversely, automated security scanning can provide concrete evidence of weaknesses observable in a running system. OWASP ZAP Baseline Scan can efficiently identify security issues related to HTTP responses, security headers, and browser policies and can be repeatedly applied to verify deployment hardening. However, because an unauthenticated passive scan does not analyze all authentication workflows, authorization relationships, or internal system interactions, the absence of an alert does not necessarily indicate the absence of structural or logical security risks.

Therefore, relying on only one assessment approach may provide an incomplete understanding of the security posture of a WebRTC-based video conferencing system. This study addresses this problem by investigating how design-stage threat modeling and runtime security scanning can be used as complementary assessment layers. Rather than comparing STRIDE and OWASP ZAP to determine which method is superior, this study focuses on the differences and overlaps in the security evidence produced by the two approaches.

To this end, this study combines STRIDE threat modeling with OWASP ZAP Baseline Scan for a WebRTC-based video conferencing architecture. STRIDE is used to identify structural threats based on system components, data flows, and trust boundaries, while the DREAD model is used to prioritize the identified threats. OWASP ZAP Baseline Scan is then used to examine observable security configuration issues in the deployed web interface. The assessment results are organized according to the OWASP Top 10 categories to analyze the scope, overlap, and complementary roles of the two assessment layers. The comparison does not attempt to directly combine DREAD scores and ZAP alert severities or establish a capability ranking between the two methods.

The main contributions of this study are as follows. First, this study provides a structured security assessment mapping for a WebRTC-based video conferencing architecture by connecting STRIDE-derived threats involving meeting-room access, authentication and authorization, media paths, ICE candidate exposure, recording access, and internal trust boundaries with the OWASP Top 10 categories.

Second, this study analyzes the complementary assessment scope of STRIDE threat modeling and OWASP ZAP Baseline Scan by distinguishing design-stage threat evidence from runtime passive scanning evidence and identifying their overlapping and method-specific security categories.

Third, this study proposes an integrated assessment workflow in which threat modeling is used to define and prioritize the security review scope, while automated scanning is used to verify observable implementation and deployment configurations and to support repeated security validation.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

II. LITERATURE REVIEW

WebRTC security research shows that browser-based real-time communication must protect against man-in-the-middle attacks, communication tampering, and media-path exposure [1][2][3][4][5]. Studies on WebRTC IP address leakage explain that private or public IP addresses may be exposed during ICE candidate collection, and that P2P restrictions and TURN relay policies may be required for high-privacy meetings [6][7]. Research on video conferencing platforms also indicates that authentication, authorization, signaling security, and data processing must be considered together rather than as isolated web functions [8][9].

From the threat modeling perspective, STRIDE classifies system threats into Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege [10]. In a video conferencing architecture, these categories can be mapped to users, browser clients, authentication functions, signaling servers, media relay servers, and data stores. OWASP Top 10 and OWASP ZAP are also widely used as web application security references and testing tools [11][12]. Prior studies imply that architecture-based modeling and scanner-based dynamic testing produce different kinds of security evidence.

III. METHODOLOGY

A. Experimental Target

The experimental target was a local experimental environment that simulated a WebRTC-based video conferencing architecture. The analysis scope included users, browser clients, authentication functions, session management, meeting-room permissions, media paths, personal data handling, and security header configuration. The open-source Jitsi Meet project was used to understand the video conferencing environment and organize the validation procedure, while the integrated assessment was conducted using local experimental code and OWASP ZAP Baseline Scan results [1][4][9][12][13].

For reproducibility, the supplementary Jitsi Meet validation was organized around the Docker-based web, Prosody, Jicofo, JVB, and Jibri components, while the scanner evidence used in Experiment B was generated from the local HTTP/HTTPS web endpoint. The passive ZAP workflow exported JSON, Markdown, and HTML reports with a command form equivalent to: docker run --rm -t -v reports/zap/secure:/zap/wrk/:rw ghcr.io/zaproxy/zaproxy:stable zap-baseline.py -t https://host.docker.internal:8443 -J zap-secure-report.json -w zap-secure-report.md -r zap-secure-report.html. This command records the use of ZAP Baseline Scan, the Docker image channel, the target endpoint, and the report-output options used for the runtime evidence.

For reproducibility, the versions of the main software components used in the experimental environment were recorded. The experiment used Jitsi Meet [v1.0.3], with Prosody [0.11.x], Jicofo [1.1-90-gd4aa6f2c], Jitsi Videobridge (JVB) [1.0-565-g2e2ce1c], and Zap [v2.13.0]. These versions correspond to the software configuration used during the experimental evaluation.

B. Experiment A: STRIDE Threat Modeling

In Experiment A, the analyst group used STRIDE results derived with Microsoft Threat Modeling Tool. The material was based on a Jitsi Meet data flow diagram and treated Nginx, Prosody, Jicofo, JVB, Jigasi, and the User/Session Data Store as major components. Threats were classified as Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Each threat was scored on a 10-point scale for the five DREAD factors: Damage, Reproducibility, Exploitability, Affected Users, and Discoverability. The final STRIDE input data were organized as 9 findings in reports/stride/stride_findings.json [10][12]. The resulting STRIDE threat elements are summarized in Table 1.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

TABLE I
SUMMARY OF STRIDE THREAT ELEMENTS

No.	Threat Element	Key Content	Main OWASP Mapping
T1	Meeting-room access control bypass	Unauthorized users may access a meeting through leaked room links or insufficient permission validation	A01, A07
T2	JWT authorization confusion	Errors in token role or permission validation may incorrectly grant host or administrator privileges	A01, A07
T3	WebRTC media-path manipulation	Media-path or relay misconfiguration may cause communication tampering and session disruption	A03, A04
T4	ICE candidate IP exposure	Private or public IP addresses and network information may be exposed during ICE candidate gathering	A01, A05
T5	Group-key renewal failure	Delayed or missing key renewal after participant changes may allow previous participants to infer communication contents	A02, A04
T6	Insufficient repudiation and audit logging	Audit logs for meeting creation, permission changes, and recording access may be insufficient for accountability	A09
T7	Security header and browser policy misconfiguration	Missing CSP, X-Frame-Options, Permissions-Policy, and related policies may increase the browser attack surface	A05
T8	Recording repository access and residual data	Recording files or meeting metadata may be exposed because of insufficient authorization checks or retention policies	A01, A08
T9	Ambiguous trust boundaries among internal components	Unclear authentication and authorization boundaries among Nginx, Prosody, Jicofo, JVB, and related components may cause privilege escalation	A01, A07

C. Experiment B: OWASP ZAP Dynamic Assessment

In Experiment B, a local web interface was assessed using OWASP ZAP Baseline Scan. The first baseline results were stored as reports/zap/baseline/zap-report.json, reports/zap/baseline/zap-report.md, and reports/zap/baseline/zap-report.html. After CSP, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Permissions-Policy, and Cross-Origin related headers were applied, the local server was rescanned in a Docker environment, and the results were stored as reports/zap/secure/zap-secure-report.json, reports/zap/secure/zap-secure-report.md, and reports/zap/secure/zap-secure-report.html. The assessment in this paper used the post-hardening rescan results as runtime evidence. Because the Baseline Scan does not authenticate to the application or inject attack payloads, it analyzes response headers and HTML structure rather than authorization workflows or behavior-based vulnerabilities. Therefore, non-detection of injection or access control bypass is interpreted as a limitation of this passive scan configuration, not as a general limitation of OWASP ZAP [12].

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

D. Integration Metrics

Table 2 lists the complementary assessment metrics used to organize the two experiments.

TABLE II
COMPLEMENTARY METRICS FOR STRIDE AND OWASP ZAP

Metric	Description
Detection Count	Number of STRIDE threat elements and ZAP alert types or instances; used as scope evidence, not as a direct ranking
OWASP Top 10 Mapping	Number of OWASP categories linked to each method's evidence; not a direct capability ranking
Overlap Detection	Categories detected or explained by both design-stage and runtime evidence
Standalone Scope	Categories represented only by one assessment layer in this experiment
Non-critical Informational Alert Analysis	ZAP informational alerts that should be separated from direct vulnerability findings
Risk Severity Distribution	DREAD severity bands and OWASP alert severities reported separately because their scales differ

IV. RESULTS AND DISCUSSION

A. Quantitative Results

Table 3 summarizes the quantitative assessment evidence from STRIDE and OWASP ZAP, and Fig. 1 illustrates the OWASP Top 10 mapping by assessment layer. In the first ZAP Baseline Scan before applying security headers, 13 alerts and 19 instances were collected. After the security-header-enabled server was rescanned using Docker-based ZAP, the result decreased to 4 alerts and 8 instances. This shows that ZAP does not replace the design threats suggested by STRIDE, but it is effective for repeatedly verifying implementation hardening effects such as security headers and browser policies [12]. These before-and-after ZAP alert changes are shown in Fig. 2.

TABLE III
SUMMARY OF ASSESSMENT EVIDENCE FROM STRIDE AND OWASP ZAP

Metric	STRIDE	OWASP ZAP	Combined Interpretation
Meaningful detections/alerts	9 threats	4 alert types	8 ZAP instances
OWASP Top 10 mapping	8/10 categories	1/10 category	Values describe evidence scope, not tool superiority
Overlapping category	Includes A05	Includes A05	Common evidence around security misconfiguration
Design-stage-only categories	A01, A02, A03, A04, A07, A08, A09	Not observed in passive scan	These areas require design review, authentication, active scanning, or manual validation
Categories outside this experiment	A06, A10	Most categories except A05	Vulnerable components and SSRF require separate experiments

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Non-critical informational alert	0	1	10109 Modern Web Application; separated from direct vulnerability findings
Risk severity distribution	High: 2; Medium: 5; Low: 2; Avg DREAD: 38.5	High: 0; Medium: 1; Informational: 3	Severity distributions are comparable qualitatively, but DREAD and ZAP scores are not additive

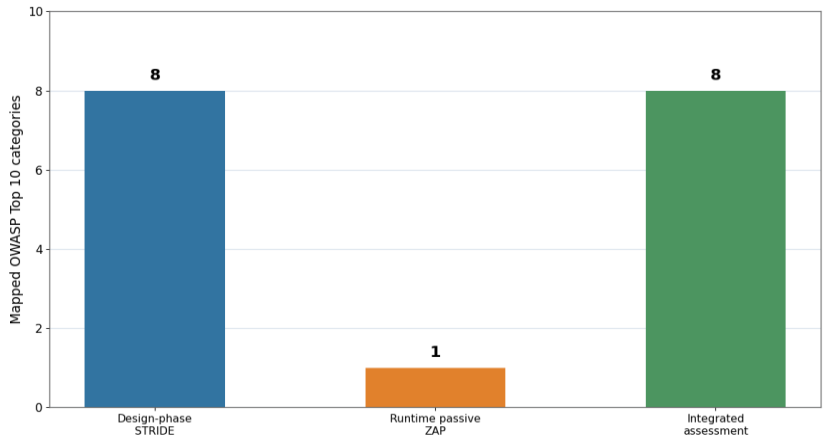

Fig. 1. OWASP Top 10 Mapping by Assessment Layer. Counts indicate evidence scope in this experiment, not a direct ranking of tool capabilities.

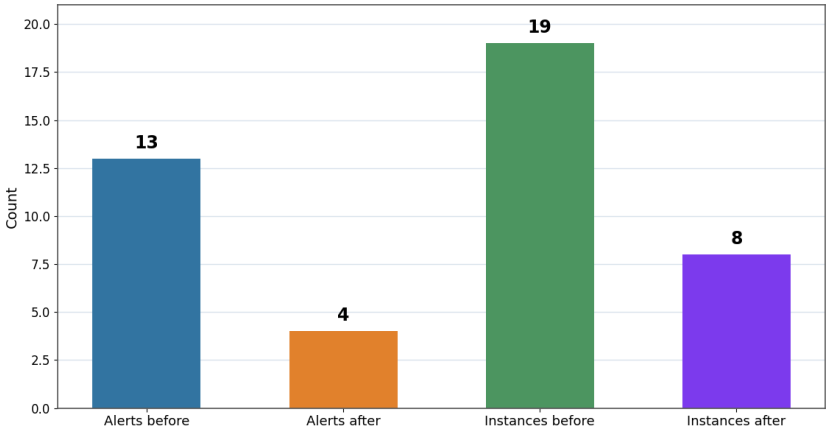

Fig. 2. Changes in ZAP Alerts Before and After Security Header Application

In this study, manual STRIDE analysis, including DFD preparation, STRIDE classification, and DREAD scoring, was organized within approximately 60 to 90 minutes. The assessment log used the median value of 75

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

minutes for the per-minute metric, while the ZAP Baseline Scan was recorded as 5 minutes based on the Docker execution option.

Table 4 therefore reports risk distribution and evidence scope without combining DREAD-based design priorities with ZAP alert severities. The DREAD values represent analyst-scored design threats on a 0-50 scale, while ZAP severities represent alert categories observed from passive runtime scanning. Treating these values as separate severity distributions avoids an invalid arithmetic comparison and better supports the complementary assessment interpretation.

TABLE IV
COMPLEMENTARY ASSESSMENT SCOPE OF EXPERIMENT A AND EXPERIMENT B

Evaluation Metric	STRIDE Threat Modeling	OWASP ZAP Baseline Scan
Scope	Internal Architecture (Prosody, JVB, Jicofo, Nginx)	External Web Endpoint (HTTP/HTTPS)
Assessment Type	Design-based structural analysis with threat enumeration	Unauthenticated passive dynamic assessment without attack payloads
Risk Metric	DREAD Model (0-50 scale per threat)	OWASP Risk Severity Rating
Severity Distribution	High: 2, Medium: 5, Low: 2 (Avg DREAD: 38.5)	High: 0, Medium: 1, Informational: 3
Primary Target	Structural and access-control flaws, privilege-escalation paths	Transport, header, CSP, cache, and browser-policy misconfigurations
OWASP Category Evidence	Design evidence mapped to A01, A02, A03, A04, A05, A07, A08, and A09	Runtime passive evidence concentrated on A05; one unmapped informational alert
Evidence Type	Design-stage inference from DFDs, trust boundaries, and data flows	Runtime observation from passive response/header analysis
Analysis Time	Approximately 60-90 minutes; median value 75 minutes used for reporting	Approximately 5 minutes per Docker-based baseline scan
Repeatability	Requires analyst review when architecture or assumptions change	Automated and suitable for rapid reassessment after hardening changes
Appropriate Interpretation	Defines security review scope and prioritizes architectural risks	Verifies deployment hardening and repeatable configuration checks

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

B. Detection Scope Matrix

Table 5 presents the detection scope matrix based on OWASP Top 10, and Fig. 3 summarizes the distribution of method-specific and shared assessment scope. Although there was no OWASP Top 10 category detected by ZAP alone in this passive Baseline Scan, one non-critical informational alert that did not directly map to OWASP Top 10 existed. Therefore, this study separated that item as a non-critical informational alert and did not include it in the standalone Top 10 detection scope of ZAP. Major structural threats identified by STRIDE, such as Broken Access Control and Injection, did not appear in the ZAP result because the scan mode used in this experiment operated without authentication and without active attack payloads. This demonstrates the passive assessment strength in quickly identifying static configuration issues such as A05, while also indicating that behavior-based vulnerabilities such as actual injection or authorization bypass require design-stage threat modeling, authenticated scanning, active scanning, or manual validation [10][11][12].

TABLE V
DETECTION SCOPE MATRIX BASED ON OWASP TOP 10

OWASP Category	STRIDE	ZAP Alerts	ZAP Instances	Detection Scope
A01 Broken Access Control	5	0	0	STRIDE only
A02 Cryptographic Failures	1	0	0	STRIDE only
A03 Injection	1	0	0	STRIDE only
A04 Insecure Design	2	0	0	STRIDE only
A05 Security Misconfiguration	2	3	7	Both
A06 Vulnerable and Outdated Components	0	0	0	None
A07 Identification and Authentication Failures	3	0	0	STRIDE only
A08 Software and Data Integrity Failures	1	0	0	STRIDE only
A09 Security Logging and Monitoring Failures	1	0	0	STRIDE only
A10 Server-Side Request Forgery	0	0	0	None
Unmapped	0	1	1	Unmapped ZAP informational

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

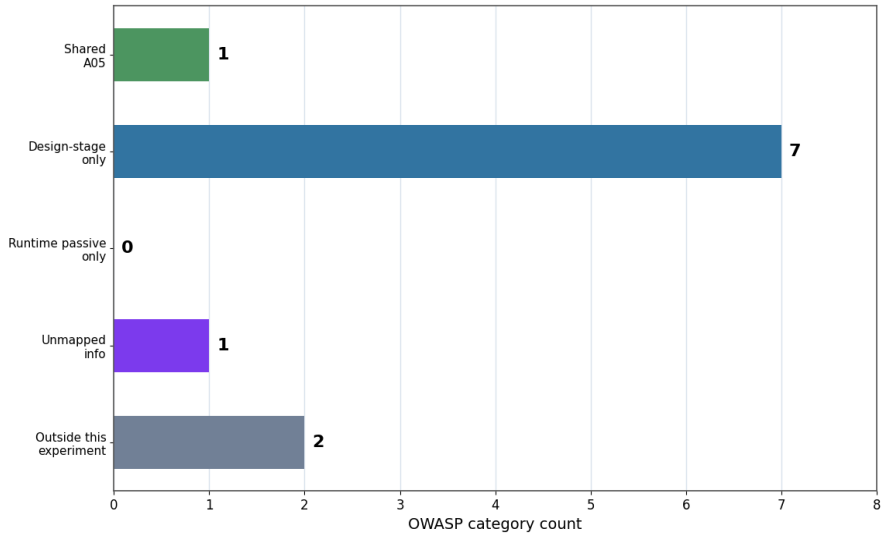

Fig. 3. Distribution of Method-Specific and Shared Assessment Scope

C. Non-critical Informational Alert Analysis

In the ZAP rescan results, the items mapped to A05 Security Misconfiguration were 10055 CSP: Meta Policy Invalid Directive and 10049 Non-Storable Content. Alert 10055 may occur when an HTTP response header and an HTML meta policy coexist or when some CSP directives are not sufficiently aligned with browser policy behavior; because it is related to script and frame control in a video conferencing interface, it was treated as a valid alert. Alert 10049 was classified as an environment-dependent alert whose impact depends on static-file cache policy. Alert 10109 Modern Web Application was classified as a non-critical informational alert because it is closer to an application-structure identification signal than to a direct vulnerability. Before security headers were applied, additional missing-header alerts such as 10020, 10021, 10038, 10063, and 90004 appeared, but those items changed to PASS in the rescan [12].

D. Interpretation of Results

Because STRIDE derives threats from trust boundaries and data flows in the video conferencing architecture, it identified structural risks such as meeting-room access control, JWT authorization confusion, WebRTC media paths, ICE candidate IP exposure, group-key renewal, and missing logs. In contrast, ZAP observes actual responses from a running web interface, so it concretely detected implementation conditions such as CSP, X-Frame-Options, X-Content-Type-Options, Permissions-Policy, and Cross-Origin policies.

The area commonly detected by both methods was A05 Security Misconfiguration. This is because STRIDE information-disclosure and denial-of-service scenarios can be connected to security misconfiguration, and most ZAP alerts were concentrated on missing security headers and browser policies. The A01, A02, A03, A04, A07, A08, and A09 areas identified by STRIDE were not directly detected by ZAP Baseline Scan because Baseline Scan is a passive assessment method that focuses on response headers and HTML structure. Therefore, the two methods should be interpreted as complementary approaches that detect vulnerabilities at different layers rather than as substitutes [10][11][12].

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

E. Strengths, Limitations, and Explanation Scope of Experiment A and B

Experiment A and Experiment B address the same video conferencing target, but they present different layers of information. Experiment A explains threats that may arise from design diagrams and data flows, while Experiment B narrowly but concretely shows the current implementation state observed from the running web interface. Therefore, this paper interprets the two experiments not as a ranking, but as methods with different roles: Experiment A is suitable for setting the security review scope and prioritizing threats, while Experiment B is suitable for confirming implementation hardening and repeated verification. Table 6 summarizes the strengths, limitations, and explanation scope of the two experiments.

TABLE VI
STRENGTHS, LIMITATIONS, AND EXPLANATION SCOPE OF EXPERIMENT A AND EXPERIMENT B

Category	Strengths	Limitations	Scope Represented in This Paper
Experiment A: STRIDE Threat Modeling	Can structurally infer privilege escalation, repudiation failure, residual data points, and missing encryption paths that automated tools struggle to detect	Difficult to directly detect implementation-stage coding mistakes; focus may vary depending on analyst judgment	Logical risks expected at the design stage, inter-server communication rules, OWASP category mapping, DREAD-based priorities, and mitigation directions
Experiment B: OWASP ZAP Dynamic Assessment	Automatically collects responses from a running web interface and repeatedly validates security headers, CSP, cache policy, and browser security policy	Baseline Scan is centered on the web interface, so meeting-room permission models, WebRTC ICE information, and TURN/XMPP/JVB internal threats are not sufficiently revealed	Observed deployment misconfigurations, alert instances, before-and-after security-header changes, and non-critical informational alerts

A detailed strength of Experiment A is that it first draws the data-exchange structure among multiple servers and components based on a DFD and then infers threats. In this process, trust boundaries and communication rules among components such as Nginx, Prosody, Jicofo, JVB, and data stores can be reviewed together. This was useful for identifying privilege-escalation possibilities, repudiation risks, internal communication paths lacking encryption, and residual data points that are difficult to reveal through simple web response scanning [10].

However, Experiment A is based on design material and analyst interpretation, so it cannot directly prove bugs or deployment mistakes introduced during implementation. It also lacks the immediate validation process provided by automated tools, so inaccurate assumptions may cause attention to be placed on lower-priority threats. As system scale increases, DFD preparation, STRIDE classification, and DREAD scoring become more complex, and static design material alone cannot reflect runtime traffic pattern changes or network environment changes in real time [10].

This comparison shows that Experiment A is suitable for explaining what risks may exist, while Experiment B is suitable for confirming what configuration problems are actually visible in the current implementation. In this experiment, STRIDE mapped design-stage threats to 8 OWASP Top 10 areas, whereas the ZAP Baseline Scan provided runtime evidence concentrated on A05. This supports the conclusion that video conferencing security assessment should combine design review and automated scanning to present a more balanced result [10][11][12].

F. Supplementary Validation Results for Jitsi Meet

Apart from the integrated assessment, web UI, Prosody, Jicofo, JVB, and Jibri were executed in a Docker-based Jitsi Meet self-hosted environment, and authentication, JWT, ICE candidates, Nmap, and ZAP Baseline results were checked. This supplementary validation was not intended to change the STRIDE-ZAP scope

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

mapping, but to confirm what operational issues remain when applying the local experimental web-client-centered results to an actual video conferencing stack [12]. Table 7 summarizes the supplementary validation results for Jitsi Meet.

TABLE VII
SUPPLEMENTARY VALIDATION SUMMARY FOR JITSI MEET

Validation Area	Observation	Paper Interpretation
Web/ZAP/Nmap	Verified web ports 8000/tcp and 8443/tcp and JVB media port 10000/udp; ZAP Baseline still reported CSP, clickjacking, and COOP/COEP hardening items	ZAP is useful for repeated validation of operational web responses and browser security policy hardening
Authentication/JWT/Lobby	Verified guest waiting before host entry, room creation by a valid JWT user, and guest participation after host entry; observed that JWT authentication success is not always identical to administrator privilege	Meeting-room authorization and authentication-flow threats identified by STRIDE should lead to operational configuration validation
ICE/TURN	Browser ICE candidates included host candidates, while relay candidates were not observed	High-privacy environments require validation of enforced TURN relay and P2P restriction
Jibri	Confirmed Jibri container health OK and Jicofo availability detection, but recorder domain errors and missing recording UI remained	Recording functionality is not sufficiently validated by startup status alone; permissions, recorder VirtualHost, and actual file generation must be tested end to end

V. CONCLUSION

WebRTC-based video conferencing systems involve security risks arising from both system architecture and implementation or deployment configurations. Therefore, relying on a single assessment method may provide an incomplete understanding of system security. This study addressed this issue by evaluating STRIDE threat modeling and OWASP ZAP Baseline Scan as complementary assessment layers.

The results showed that STRIDE identified nine design-stage threats mapped to eight OWASP Top 10 categories, revealing structural risks related to access control, authorization, trust boundaries, privilege escalation, and communication paths that may not be observable through passive response scanning. In contrast, the OWASP ZAP Baseline Scan produced four alert types and eight instances, primarily related to A05 Security Misconfiguration, and was useful for verifying security headers, browser policies, and the effects of deployment hardening.

These findings support the central argument of this study that STRIDE and OWASP ZAP should be interpreted as complementary rather than competing approaches. STRIDE can first define and prioritize the security review scope based on system architecture, while OWASP ZAP can repeatedly verify observable implementation and deployment configurations. Integrating these two assessment layers can therefore provide a more comprehensive basis for evaluating the security of WebRTC-based video conferencing systems.

This study was limited by its local experimental environment and the use of an unauthenticated OWASP ZAP Baseline Scan. Future work should extend the assessment through authenticated and active scanning and broader validation of WebRTC components and operational security issues, including TURN relay configurations, recording access, operational logs, mobile applications, and personal-data protection policies [1][4][9][10][11][12][13].

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Si Hwan Kim *et al*, Volume 15, Issue 8, August 2026, pg. 59-70
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

ACKNOWLEDGEMENT

This research was supported by the MSIT (Ministry of Science, ICT), Republic of Korea, under the National Program for Excellence in SW, supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation) in 2026 (2024-0-00023).

REFERENCES

[1]. Luis López-Fernández, Micael Gallego, Boni García, David Fernández-López, and Francisco Javier López, "Authentication, Authorization, and Accounting in WebRTC PaaS Infrastructures," IEEE Internet Computing, Vol. 18, No. 6, pp. 34-40, 2014. DOI: 10.1109/MIC.2014.111.
[2]. E. Rescorla, "Security Considerations for WebRTC," IETF RFC 8826, 2021.
[3]. Alan B. Johnston and Daniel C. Burnett, WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web, Digital Codex LLC, 2014.
[4]. Ben Feher, Lior Sidi, Asaf Shabtai, and Rami Puzis, "The Security of WebRTC," arXiv:1601.00184, 2016. DOI: 10.48550/arXiv.1601.00184.
[5]. Tuukka Koistinen, Implementation and Evaluation of Security on a Gateway for Web-based Real-Time Communication, Master's Thesis, Aalto University, 2014.
[6]. N. M. Al-Fannah, "One Leak Will Sink A Ship: WebRTC IP Address Leaks," arXiv:1709.05395, 2017. DOI: 10.48550/arXiv.1709.05395.
[7]. S. Tang, E. Alowaisheq, X. Mi, Y. Chen, X. Wang, and Y. Dou, "Stealthy Peers: Understanding Security Risks of WebRTC-Based Peer-Assisted Video Streaming," arXiv:2212.02740, 2022. DOI: 10.48550/arXiv.2212.02740.
[8]. Raiful Hasan and Ragib Hasan, "Towards a Threat Model and Security Analysis of Video Conferencing Systems," Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), pp. 1-4, 2021. DOI: 10.1109/CCNC49032.2021.9369505.
[9]. Georgios Achilleos, Konstantinos Limniotis, and Nicholas Kolokotronis, "Exploring Personal Data Processing in Video Conferencing Apps," Electronics, Vol. 12, No. 5, Article 1247, 2023. DOI: 10.3390/electronics12051247.
[10]. Microsoft, "Microsoft Threat Modeling Tool Threats," accessed May 13, 2026.
[11]. OWASP Foundation, "OWASP Top 10:2021," accessed May 13, 2026.
[12]. OWASP Foundation, "OWASP Zed Attack Proxy Documentation," accessed May 14, 2026.
[13]. Jitsi, "Jitsi Meet Handbook," accessed April 30, 2026.

