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.

2025

08

SK shieldus

Contents

Headline

Shadow AI : Detection, Control, and Governance for Manufacturing Confidentiality

Won-jun Song, SK Shieldus

1. Strategic Security Threats Posed by Shadow AI in Manufacturing Environments

Since 2023, the commercialization LLM-based generative artificial intelligence (Generative AI) has dramatically accelerated innovation across all industries, driving advances in process automation, engineering optimization, and knowledge refinement. In particular, the manufacturing sector has witnessed the rapid emergence of AI's utility in diverse functional domains, including product design, quality management, process control, and productivity enhancement. However, these technological advancements have simultaneously reshaped the security landscape, introducing new vectors of risk—foremost among them is the phenomenon of Shadow AI.

Shadow AI refers to the unauthorized and informal use of AI services by individuals or departmental units without official organizational approval. In knowledge-intensive industries such as manufacturing, the unsanctioned external transmission of critical assets—including trade secrets, production recipes, routing information, design blueprints, and equipment logic—can pose severe security threats. Such practices can circumvent the detection capabilities of existing security infrastructures (such as DLP, EDR, CASB), resulting in a substantial degradation of security visibility within the organization.

For instance, consider scenarios in which R&D engineers describe CAD designs to LLM-based chatbots in order to solicit technical feedback, or cases where manufacturing technology teams input proprietary process data to optimize production recipes. The critical issue here is that most of these inputs are transmitted as unstructured API traffic over HTTPS, thereby evading internal audit and access control mechanisms. Should such prompts be leveraged as training data or stored long-term by external AI services, there exists a tangible risk that proprietary information may be repurposed in subsequent model training within the same industry sector.

Moreover, the risks associated with Shadow AI extend beyond information leakage, encompassing secondary threats such as regulatory non-compliance, legal disputes, and violations of industrial protection statutes. Notably, under domestic and international regulatory frameworks—such as the Industrial Technology Protection Act, GDPR, and ITAR—the mere loss of control over confidential information is sufficient grounds for forfeiting its protected status. As a result, even a single instance of external transmission may irreversibly compromise the legal protection afforded to patent assets.

Accordingly, Shadow AI must not be dismissed as a mere 'user behavior issue'; rather, it should be recognized as a structural vulnerability within knowledge-driven security strategies for the manufacturing sector. This reality underscores the urgent need to establish a comprehensive governance model encompassing proactive detection, behavioral control, and prompt-level risk assessment frameworks.

In this Insight, we examine the operational dynamics of Shadow AI and its manufacturing-specific threat scenarios, and propose an effective security model that encompasses both technical detection mechanisms and policy-driven response strategies. Furthermore, drawing upon global regulatory trends and response guidelines, we present a reference framework designed to support the establishment of practical, operations-oriented governance systems.

2. Conceptual Overview and Threat Model Analysis

2.1 Definition and Behavioral Characteristics of Shadow AI

Shadow AI refers to the practice whereby individuals or departmental units utilize unauthorized generative AI tools—such as large language models (LLMs), Vision AI, or AutoML—without passing through the organization's established security or IT management frameworks. In this process, users often engage in the following behaviors, frequently without adequate awareness of security protocols or data handling regulations.

- Directly inputting internal documents, blueprints, or process information into external AI systems in the form of prompts
- Integrating code or documents generated by external AI into operational systems without proper validation
- Failing to recognize that sensitive data may be automatically stored or cached on external servers outside the corporate perimeter

While the use of Shadow AI may ostensibly aim to enhance workplace productivity and support individual tasks, from a security perspective it constitutes the high-risk transmission of sensitive data through unauthorized channels.

2.2 Shadow AI Threat Model Classification (Manufacturing-Centric)

The following section delineates the various threat types associated with Shadow AI in manufacturing environments, structured around behavior, risk, impact, and illustrative examples.

① Leakage of Design and Technical Documentation

Element	Description
Behavior	Requesting explanations of CAD drawings or summaries of product design structures
Risk	Exposure of design expertise, component specifications, and positioning information to LLMs
Impact	Potential exploitation for imitation of similar products or acquisition of proprietary technology by competitors
Example	Including the complete design structure in a prompt such as, “Is there any overall issue with this design?”

② Exposure of Manufacturing Recipes and Process Parameters

Element	Description
Behavior	Querying AI for process condition adjustments or methods to improve yield
Risk	Transmission of internal variables such as production temperature, speed, and material ratios
Impact	Loss of quality competitiveness; transfer of proprietary information to OEM/ODM competitors
Example	Prompting with questions like, “Analyze the causes of defects for this material ratio.” thereby disclosing sensitive process details

③ Leakage of Sensitive Information via Quality Data

Element	Description
Behavior	Inputting defect occurrence databases, inspection images, or defect types into AI systems
Risk	Product defect data and structural vulnerability information are learned by external entities
Impact	Potential identification of vulnerable products, which could be exploited to maliciously trigger recalls
Example	Prompts such as, “Explain why this photo was classified as a grade B defect.” inadvertently disclose sensitive quality data

④ Leakage and Compromise of Automation Code or Sequences

Element	Description
Behavior	Requesting AI to diagnose PLC control code or sequence logic
Risk	Exposure of code logic, or incorporation of insecure logic from AI-generated code
Impact	Potential for equipment shutdown, safety incidents, or propagation of attacks targeting operational technology (OT) systems
Example	AI-generated code omits authentication procedures, enabling injection of external commands

⑤ Indirect Leakage of User Credentials and System Information

Element	Description
Behavior	Supplying LLMs with development code or API examples
Risk	Disclosure of authentication tokens, account names, and system port configurations
Impact	Unintentionally furnishing attackers with a blueprint of internal APIs
Example	Requests such as, “Show me how to integrate this API with the quality management system.” which may inadvertently reveal sensitive system architecture details

⑥ Information Inference via Training Data Reuse

Element	Description
Behavior	Repeatedly inputting prompts containing internal information into LLMs
Risk	Subsequent prompts from other users may elicit generated responses that reproduce the previously entered confidential data
Impact	Loss of confidentiality, effectively equivalent to public disclosure of the information
Example	Requests such as, “Show me the production recipe I provided earlier.” resulting in sensitive data being resurfaced in model outputs

⑦ Regulatory and Compliance Violations

Element	Description
Behavior	Transmitting confidential information to overseas AI servers, potentially violating regulations such as GDPR, ITAR, or the Industrial Technology Protection Act
Risk	Non-compliance with regulatory requirements, exposure to legal action, and risk of certification revocation
Impact	Damage to corporate reputation and loss of external contracts
Example	Transmission of design blueprints from a defense component manufacturer to OpenAI

As demonstrated by the aforementioned cases, Shadow AI exhibits the following multifaceted characteristics:

- Low-intent, High-impact : While user actions may be well-intentioned, their consequences can prove catastrophic.
- Technical Undetectability : Information embedded within prompts is inherently difficult to identify and classify using conventional methods.
- Governance Externality : Such activities occur outside the purview of traditional information security management frameworks.
- Expansion of the Attack Surface: External API and model invocations effectively create new security perimeters.

2.3 Derivation of Key Issues

Within manufacturing organizations, Shadow AI should not be dismissed as mere employee negligence; rather, it constitutes a warning sign that exposes fundamental deficiencies in the organization's security governance framework. Even in the absence of an external attacker, critical assets can be exfiltrated internally, and any leaked information remains irretrievable—necessitating that such incidents be classified as irreversible security breaches.

Accordingly, the detection, prevention, mitigation, and incident response for Shadow AI must be regarded not as optional measures, but as indispensable elements of security strategy in the era of digital manufacturing.

3. Technical Response Strategies: Detection, Control, and Mitigation

3.1 Detection Strategy

Visibility is paramount for the effective detection of Shadow AI. To accurately identify HTTPS-based AI API calls, dynamic domains, and unstructured prompts, the following response framework is recommended.

① Leakage of Design and Technical Documentation

- AI platform calls can be identified through Server Name Indication (SNI), User-Agent, and Domain Name System (DNS) request patterns
- Advanced Cloud Access Security Broker (CASB) solutions enable real-time detection and policy enforcement for external Large Language Model (LLM) API calls
- However, conventional CASB platforms provide limited detection capabilities for Shadow AI; therefore, Data Security Posture Management (DSPM) features capable of capturing AI-related data flows are required

② Prompt Content-Based Anomaly Detection

- Implement policies to detect high-risk keywords such as “design,” “confidential,” “process,” or “revenue,” flagging prompts that contain sensitive information
- Apply AI-aware Data Loss Prevention (DLP) mechanisms or prompt injection detection rules to monitor unstructured natural language requests

③ Shadow AI Tool Intelligence and Inventory

- Enhance detection capabilities to identify emerging tools such as Perplexity and DeepSeek, in addition to unofficial instances of ChatGPT and Gemini
- Establish blacklist mechanisms and detection baselines using DNS, IP, and User-Agent data—comparable to Advanced Persistent Threat (APT) prevention measures

3.2 Control Strategy

Following detection, the structural management of Shadow AI usage requires the implementation of stringent access control policies and the provision of authorized internal alternative models.

① Restrict AI Usage Permissions Based on RBAC

- Implement differentiated AI access permissions for each department using Role-Based Access Control (RBAC)
- Block external AI usage for functions such as design and R&D, while allowing only secure summarization capabilities for roles like marketing
- 'Establish and automate policies in accordance with the principle of least privilege

② Proxy-Based Blocking and AI SaaS Blacklisting

- Block access to AI services offered in a Software as a Service (SaaS) model at the HTTPS proxy layer
- Expand visibility and control by implementing automated discovery of newly emerging AI services

③ Internal Operation of Private LLM Environments

- Promote the adoption of internal AI models, such as Azure OpenAI Private Endpoint
- Preemptively control external access to maintain security governance within the organizational infrastructure boundary

④ Real-Time Sensitive Information Filtering via AI-aware DLP

- Detect sensitive data types, including PII (Personally Identifiable Information), IP (Intellectual Property), and mCAD (manufacturing CAD data)
- Leverage AI-specific DLP solutions and related products

3.3 Mitigation Strategies

The mitigation phase encompasses Zero Trust-based data flow controls, enhanced user awareness mechanisms, and the establishment of robust incident response protocols.

① Zero Trust-Based Prompt Pathway Control

- Regulate external LLM request channels through Zero Trust Network Access (ZTNA) authentication and authorization mechanisms'
- Analyze and restrict "data transmission" at each stage, from internal networks to internet gateways and ultimately to external AI services

② Security Nudging: Policy-Driven Alerts and Awareness

- Utilize KPIs (Key Performance Indicators) such as departmental AI usage frequency, detection counts, and policy violation trends
- Reinforce organizational awareness and shared accountability through regular reporting

③ KPI-Driven Monitoring and Executive Reporting

- Apply Role-Based Access Control (RBAC) for differentiated departmental permissions
- Prohibit external AI usage for design/R&D roles, while permitting only secure summarization for marketing and similar functions
- Establish and automate least-privilege policies

④ Incident Response Preparedness – Prompt Logging, Backup, and Analysis

- Integrate prompt/response log analysis within the Security Operations Center (SOC)
- Include the capability to track the scope of exposure, API usage records, and user identities in the event of an incident

4. Governance and Policy-Driven Organizational Response Framework

Technical countermeasures alone are insufficient to address the threats posed by Shadow AI. Because these risks are compounded by employee unawareness, habitual usage patterns, and the absence of robust policies, it is imperative to establish an organization-wide management framework through comprehensive security governance and informed decision-making processes.

4.1 Shadow AI Policy Framework

① AI Usage Policy

- Clearly document the criteria for prohibiting or permitting Shadow AI usage to ensure organization-wide understanding.
- The policy must specify: which AI tools may be used (maintaining allow/conditional/deny lists); what types of data are prohibited from input (e.g., PII, CAD files, design documents, source code, with illustrative examples); the disciplinary measures for violations; and procedures for exception approvals.

② AI Risk Classification (Business-Driven Risk Grading)

- Assign and manage risk levels for AI usage based on department, role, and business process.
- Establish differentiated approval and control mechanisms for each risk tier (e.g., RBAC + AI Usage Scope Matrix).

③ AI Usage Approval Process

- Require prior review by the security team or AI governance committee for any requests to use new AI tools.
- Implement technical evaluation processes for API communications, browser extensions, and internal network access requests.
- Mandate administrative approval procedures for exceptional use cases.

4.2 Education and Organizational Awareness Enhancement Strategy

More than 80% of Shadow AI incidents result from unintentional use without security awareness. Accordingly, comprehensive awareness programs—rather than simple restrictions—are indispensable at the enterprise level.

① AI Security Awareness Training Program

- Deliver regular training (at least semi-annually) and develop dissemination materials.

② Distribution of Prompt Authoring Guidelines

- Publish practical, field-oriented guides highlighting examples of “strictly prohibited prompts.

③ Operation of a Security Accountability System

- Appoint security leaders within each department to monitor AI usage, conduct campaigns, and report issues.
- Facilitate channels of communication between departments and the security team.
- Maintain continuous operation of internal security issue-sharing platforms.

4.3 AI Governance Organizational Model

① AI Risk Control Taskforce

- Composition: Security team (CISO), IT (CIO), Legal, Internal Controls, and representatives from each business unit
- Role: Manage an internal AI tool whitelist, share weekly Shadow AI detection reports, and coordinate new policies and violation responses

② AI Risk Steering Committee

- Operates as an executive reporting structure, enabling rapid decision-making in response to elevated risk levels
- KPIs: Shadow AI detection rate, number of violations, security guideline training completion rates, etc.

③ Integration with Audit and Internal Control

- Incorporate internal audit items relating to AI usage
- Regularly report on security logs, prompt usage history, and external access records

4.4 Industry Standards and Compliance Alignment

In addition to strengthening security governance within the manufacturing sector, alignment with both domestic and international legal and industry standards is essential.

Regulatory Standard	Application Area	Response Strategy
ISO/IEC 42001	Establishment of a governance framework for generative AI operations	Classification of AI risk levels; operation of oversight committees
NIST AI RMF	AI risk management framework	Inclusion of Shadow AI risk response measures
KISA AI Security Guidelines	Domestic industry-based AI security recommendations	Incorporation of AI prompt filtering and sensitive data detection
GDPR/Personal Information Protection Act	Automation processing and sensitive data leakage	Implementation of pre-input AI detection and data masking mechanisms

5. Conclusion and Response Roadmap Proposal

5.1 Conclusion

Shadow AI has rapidly emerged as a novel security risk that transcends conventional IT controls, posing direct threats to organizational confidentiality and competitiveness. This risk is particularly acute for manufacturing enterprises, where industrial trade secrets—such as design blueprints, proprietary process know-how, and cost data—are increasingly susceptible to external leakage via LLM (Large Language Model)-based AI tools.

This Insight has provided an integrated response strategy to Shadow AI threats, spanning technical, policy, and governance dimensions. The key elements of this response are as follows:

- Detection: Securing AI usage visibility through AI-aware DLP, CASB, DSPM, and related tools
- Control: Establishing AI usage policies, enforcing proxy-based blocking, and implementing role-based access control (RBAC)
- Mitigation: Controlling data pathways via Zero Trust principles, deploying alert UIs, and establishing robust incident response systems
- Governance: Instituting enterprise-wide policies, departmental risk classification, continuous education, and structured internal audits

Such measures should not be viewed as one-off policies, but rather must be embedded into organizational culture and security governance frameworks.

5.2 Proposed Response Roadmap

Outlined below is a three-phase roadmap for responding to Shadow AI:

[Phase 1: Visibility and Awareness Enhancement]

- Objective: Identify and understand the presence and risks of Shadow AI
- Key Actions:
  - → Identify the existence and risks of Shadow AI
  - → Conduct an internal assessment of Shadow AI usage
  - → Distribute educational materials on Shadow AI incident cases
  - → Establish departmental frameworks for sensitive data classification

[Phase 2: Policy and Technical Control Establishment]

- Objective: Control and minimize the use of Shadow AI
- Key Actions
  → Establish and disseminate AI usage policies
  → Configure RBAC-based AI access permissions
  → Apply and test DLP policies for sensitive data
  → Implement proxy-based blocking mechanisms for LLM access

[Phase 3: Organizational Embedding and Governance]

- Objective: Institutionalize the response framework within the organization
- Key Actions
  → Operate an AI governance committee and implement a security accountability system
  → Monitor AI usage and produce regular reports
  → Conduct ongoing AI security awareness training
  → Refine compliance response systems for AI, aligning with standards such as ISO and NIST

5.3 Future Tasks and Recommendations

- Consideration of Internal LLM Deployment: Establish private LLM environments to leverage generative AI capabilities without incurring security risks, thereby reducing reliance on external Shadow AI services.
- Expansion of AI-Specialized Security Solutions: As existing security appliances struggle to detect the unstructured nature of LLM interactions, it is essential to adopt AI-aware DLP, prompt security filtering, and data flow detection technologies.
- Evolution of the Security Team's Role: Responding to Shadow AI threats requires security teams to transition from mere monitoring to serving as AI utilization advisors and security consultants.
- Advancement of Legal and Regulatory Compliance Systems: With generative AI-related regulations evolving rapidly, dedicated organizational structures and the integration of audit criteria are necessary to ensure compliance.

Shadow AI is not merely a matter of technological adoption, but a security imperative that fundamentally determines the protection of trade secrets and, ultimately, the survival of the organization. It is now essential to implement multilayered countermeasures—spanning technology, policy, and culture—in an integrated manner.

If your organization requires the development of security policies to safeguard industrial trade secrets from Shadow AI threats, we encourage you to leverage SK Shieldus's extensive expertise in technology and policy to initiate a robust AI security governance framework.

■ References

[1] Structured, Shadow AI – The Hidden Threat to Governance & Compliance, 25.04

[2] Inteleca, Shadow AI in the Workplace: The Hidden Security and Compliance Risks, 25.03

[3] CIODIVE, AI-generated code leads to security issues for most businesses, 24.01

[4] Nightfall AI, The Nightfall Approach: 5 Ways Our Shadow AI Coverage Differs from Generic DLP, 25.07

[5] NIST AI RISK MANAGEMENT FRAMERK (AI RMF), 23.01

■ Additional Resources

[1] Paloalto, What Is Shadow AI? How It Happens and What to Do About It (Cyberpedia)

[2] ISO/IEC 42001:2023, Information technology — Artificial intelligence — Management system

[3] Ministry of the Interior and Safety(South korea), AI Security Guidelines for Public Institutions, Oct. 2023

[4] NIPA, Report on Generative AI Utilization and Security Threats by Industry, 2024

[5] SK Shieldus, EQST Insight Blog Series (2023–2024)

Keep up with Ransomware

Gunra Ransomware Targeting the Korea Financial Sector

■ Overview

In July 2025, the number of ransomware incidents recorded in the South Korea sector declined to 483 cases, down from 512 incidents in June. Although the total number of cases exhibited a slight decrease, the sophistication and strategic diversity of the attacks were, in fact, further intensified. Notably, the exploitation of vulnerabilities for initial intrusion, attempts at automating negotiations through AI-driven mechanisms, and direct confrontations with law enforcement agencies emerged as defining characteristics of ransomware activity in July.

Evidence has also surfaced indicating that the Gunra group, which emerged in April, conducted attacks targeting Korea financial institutions. On July 14, a victim institution experienced a temporary disruption in service delivery due to a ransomware attack, but was able to complete recovery and resume operations within approximately four days. However, the impact of the incident extended beyond mere service interruption, as it escalated into a data breach. The perpetrator claimed, via their dedicated Leak Sites, to have exfiltrated the institution's database and posted messages soliciting collaborators to assist in data analysis. The leaked data was confirmed to comprise compressed files totaling 13.2 terabytes, with the attacker further escalating pressure on the victim by threatening to release the stolen data incrementally.

The Akira group appears to have achieved infiltration even within environments protected by multi-factor authentication (MFA[1]), exploiting patched SonicWall SSL-VPN [2]appliances. This has raised concerns regarding the potential existence of a zero-day vulnerability, underscoring the persistent threat posed by sophisticated attacks that remain difficult to defend against until such vulnerabilities are officially disclosed and patched. Another notable case involved the proliferation of Warlock ransomware through exploitation of the ToolShell vulnerability (CVE-2025-53770) in Microsoft SharePoint. This attack affected approximately 400 servers, including those belonging to critical U.S. government agencies such as the National Nuclear Security Administration (NNSA) and the National Institutes of Health (NIH).

Attackers are exhibiting not only heightened technical sophistication but also continuous evolution in their operational strategies. The Global group, for instance, incorporated AI chatbots into negotiations with victim organizations, automating the interface and seeking to expedite the negotiation process. This development is regarded as a representative example of the growing trend toward service-oriented and automated operations within the RaaS[3] ecosystem.

[1] MFA (Multi-Factor Authentication): An authentication mechanism that enhances security by requiring users to provide two or more distinct authentication factors when accessing an account.

[2] SSL-VPN (Secure Sockets Layer Virtual Private Network): A device that enables remote access to internal networks over the internet via an encrypted communication channel.

[3] RaaS (Ransomware-as-a-Service): A business model in which ransomware is offered as a service, enabling virtually anyone to easily create and deploy ransomware attacks.

Law enforcement agencies are further intensifying their measures in response to adversarial activities. The U.S. Federal Bureau of Investigation (FBI) has initiated legal proceedings to seize approximately $2.4 million worth of Bitcoin held by members of the Chaos ransomware group. On July 25, the FBI, Europol, and the police forces of Germany and the Netherlands jointly succeeded in seizing the dedicated Leak Sites operated by the BlackSuit ransomware group. This group had listed more than 180 victims on its site, with total ransom demands reportedly amounting to nearly $500 million.

Meanwhile, there have also been instances of groups resuming activity despite law enforcement sanctions. BreachForums—a hacking forum that had been shut down following the arrests of five key operators by the French Cybercrime Brigade (BL2C) in February and June—was restored on July 26. According to an announcement by the forum's administrator, the individuals apprehended did not possess actual administrative privileges and were merely assigned titles to obscure the identities of the true operators. The administrator further acknowledged that the forum's temporary suspension in April was indeed caused by a vulnerability in the MyBB forum software, but asserted that the issue has since been resolved and that the previous domain was taken down at the request of law enforcement agencies.

In contrast, the Russian hacking forum XSS remains offline. In July, international law enforcement agencies—including Europol—arrested an individual in Ukraine who is believed to have been one of the forum's administrators. The arrested suspect is reported to have been active within the cybercrime ecosystem for approximately two decades, accumulating around 7 million euros through advertising and brokerage commissions. Since this arrest, the XSS forum has not been restored.

Ransomware News

Gunra group ransomware and data theft attack on South Korean financial institution

- On 14 July, a ransomware attack on a South Korean financial institution caused a service outage.
- Despite service restoration, a 13.2TB database was exfiltrated.
- Stolen data analysed and disclosure announced

Seizure of illicit cryptocurrency proceeds from the Chaos group

- The U.S. FBI initiated legal proceedings to seize approximately USD 2.4 million worth of Bitcoin assets held by the Chaos ransomware group
- The U.S. Department of Justice seeks forfeiture of ransomware proceeds from victim wallets
- This action, aimed directly at criminal proceeds, is regarded as a significant expansion of law enforcement efforts beyond infrastructure takedowns.

International joint operation to dismantle BlackSuit group infrastructure

- Through a coordinated operation, the FBI, Europol, and the German and Dutch police seized BlackSuit group's Dedicated Leak Site
- This action, which directly dismantled infrastructure, is regarded as a notable case of delivering tangible impact on the ransomware ecosystem.

BreachForums re-emerges despite law enforcement pressure

- BreachForums, previously shut down following the operator's arrest, returned on 26 July.
- The operator claimed that those arrested never held actual administrative authority and had only been given titles to attract attention.
- Resumption of Operations Following Completion of Security Vulnerability Patching and Domain Replacement

Hacking forum XSS shut down by law enforcement

- In July, Europol and other law enforcement agencies arrested the XSS forum operator in Ukraine.
- The operator is estimated to have been active for 20 years, generating approximately € 7 million in profits.
- Following the arrest operation, the XSS forum remains closed.

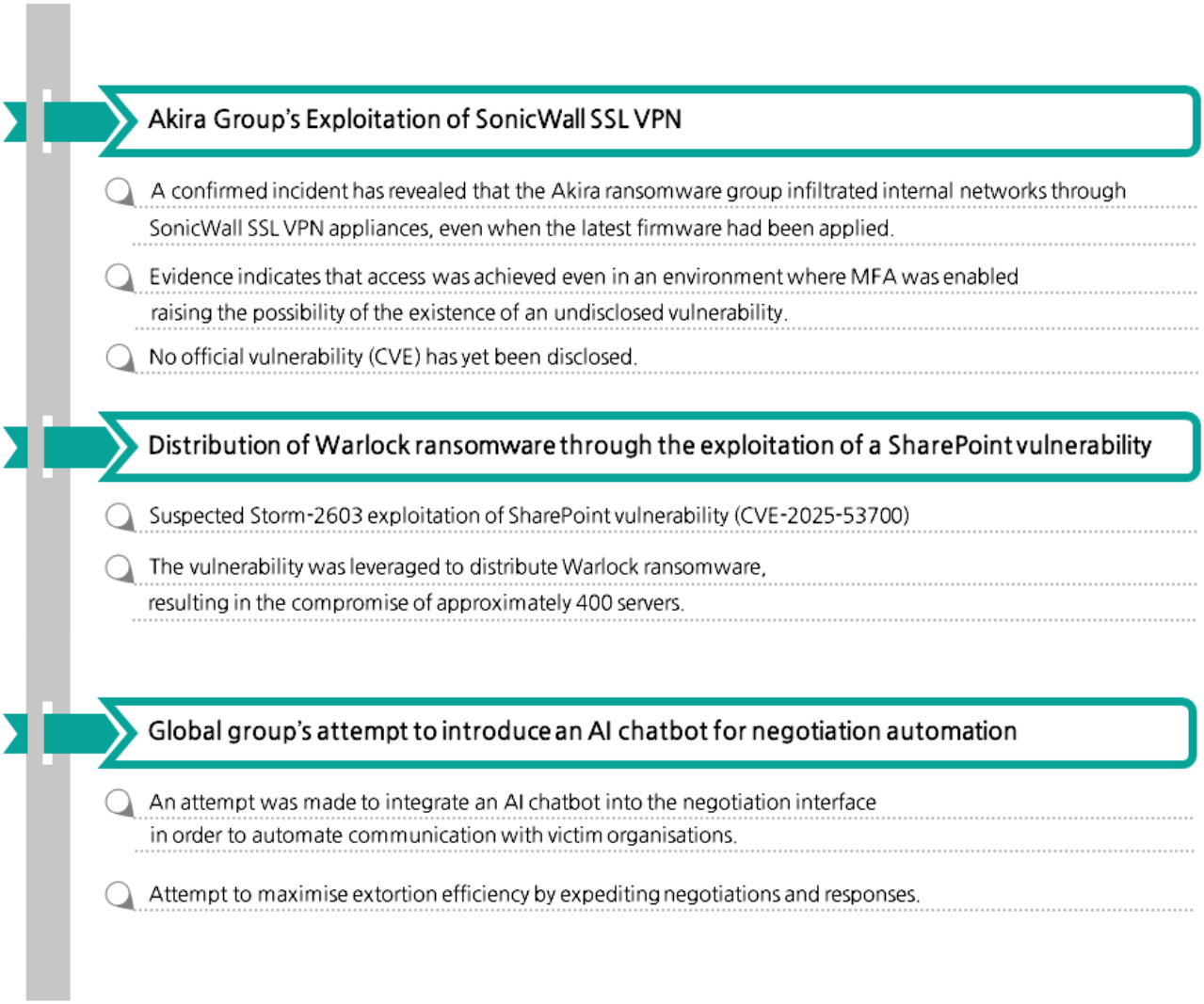


Figure 1. Ransomware Trends

Ransomware Threats

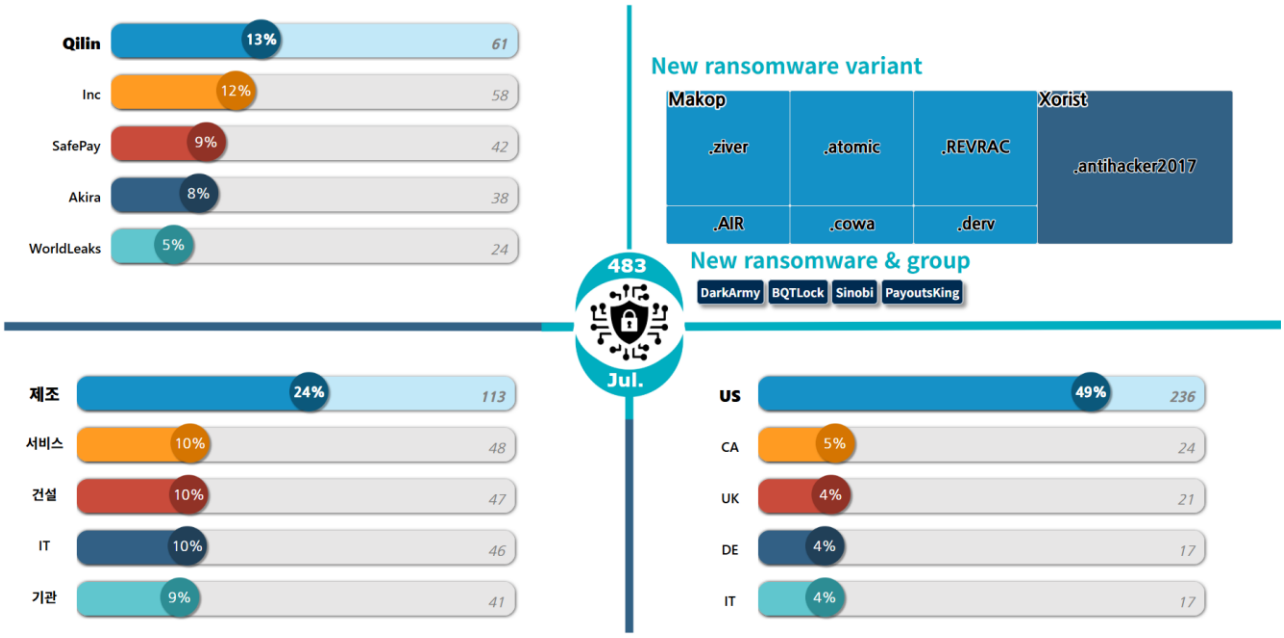


Figure 2. Status of Ransomware Threats in July 2025

New Threats

A total of 483 ransomware incidents were confirmed in July, during which four new ransomware groups emerged. Each of these groups published details of their attacks on dedicated Leak Sites under their own operation. The confirmed number of incidents attributed to each group was as follows: DarkArmy with 11 cases, BQTLock with 2 cases, Sinobi with 5 cases, and Payoutsking with 18 cases.

Figure 3. DarkArmy's dedicated Leak Sites

At the bottom banner of DarkArmy's dedicated Leak Sites, the Chinese slogan "睡觉的乌鸦" (which translates to "Sleeping Crow") is prominently displayed, alongside contact information listing both QQ and WeChat accounts. Taken together, these elements strongly suggest that the developers and operators are likely Chinese-speaking individuals or entities.

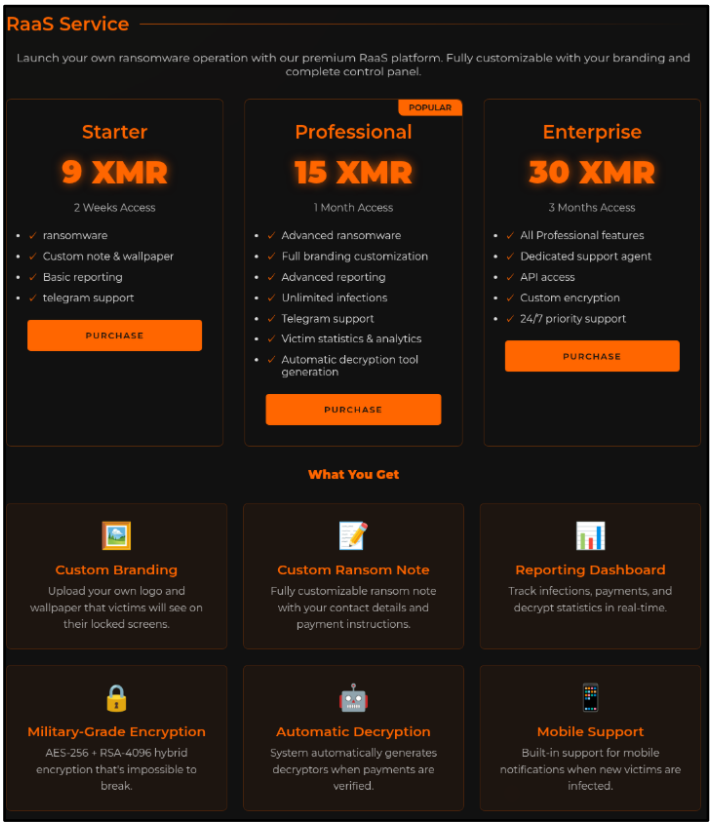


Figure 4. BQTLock RaaS Dashboard

BQTLock operates its own portal, known as BQT RaaS, which provides subscribers with a fully customizable builder[4] and a comprehensive statistics dashboard. The portal delineates three tiers of subscription plans—Starter, Professional, and Enterprise—priced at 9 XMR, 15 XMR, and 30 XMR, respectively. Subscriptions at the Professional level or higher unlock additional features, including ransom note branding customization, victim statistics and reporting, and automatic decryption tool generation. This all-in-one RaaS platform structure is poised to accelerate market proliferation by enabling even non-developer threat actors to rapidly establish and manage ransomware campaigns.

[4] Builder: A tool that enables attackers to configure detailed options—such as the ransomware's encryption algorithm, ransom amount, victim message, and target directories—via a graphical user interface (GUI) or command-line interface, and automatically generates the final executable file.

Top 5 Ransomware

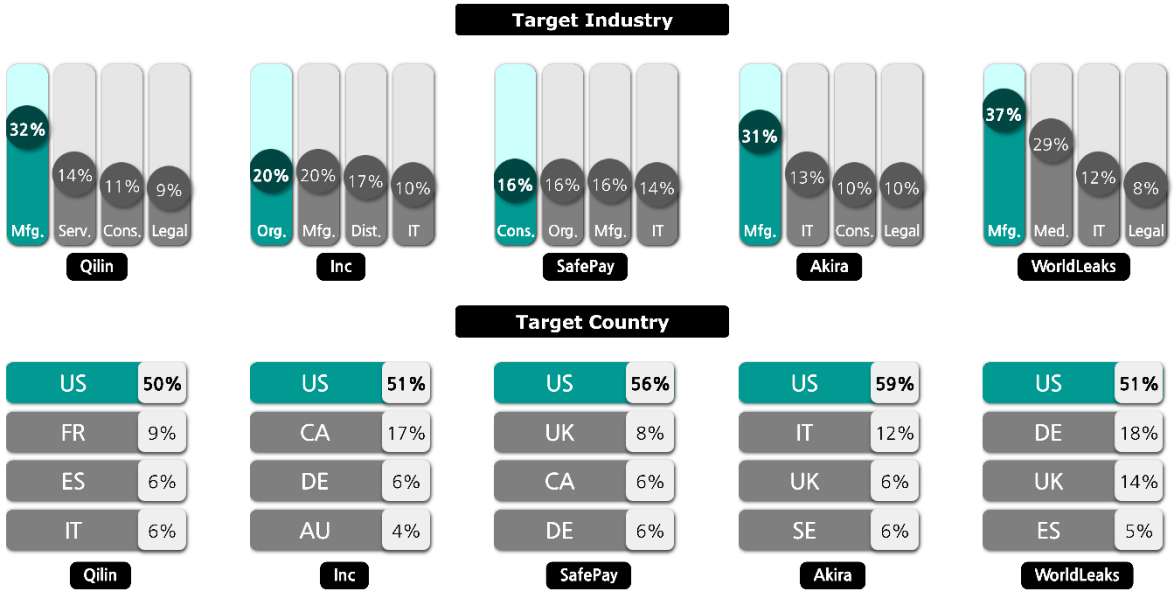


Figure 5. Major Ransomware Attacks by Industry and Country

On July 29, the Qilin group claimed responsibility for an attack on Custom Food Ingredients, a Malaysian food ingredient manufacturer, asserting that they had penetrated the company's core manufacturing systems and exfiltrated internal data. The group heightened pressure on the victim by publishing a list of production and operations-related documents on their dedicated leak site.

On July 31, the Inc group disclosed that it had compromised West Virginia Primary Care Association, a public healthcare organization in the United States. The group posted details of the incident on its dedicated leak site, demanding contact from the victim and warning that the stolen data would be published if negotiations failed. Additionally, Inc targeted the administrative office of Albemarle County, Virginia, exfiltrating thousands of personal records—including residents' and employees' names, addresses, Social Security Numbers (SSN), and driver's license numbers.

In early July, the SafePay group claimed responsibility for an attack on Ingram Micro, a global IT distribution company. In the immediate aftermath, the company's website and order processing systems experienced temporary outages. SafePay subsequently listed the victim's name on its dedicated leak site and threatened to release approximately 3.5TB of exfiltrated data. On July 26, the group further announced an attack against Southwest Florida Dermatology, a U.S. dermatology clinic, stating that they had exfiltrated sensitive internal data, including patients' medical records.

On July 25, the Akira group claimed to have exfiltrated data from Dunlap Codding, an intellectual property law firm based in Oklahoma City, United States. On its dedicated leak site, the group posted a notice announcing the impending release of approximately 19GB of data, including client files, financial documents, and records related to patents and court proceedings. Additionally, the Spanish online beauty retailer Druni fell victim to an attack, resulting in the leakage of 40GB of data, which included employee identification cards, financial records, and customer information.

On July 21, the WorldLeaks group claimed responsibility for an attack against Proactive Engineering Consultants, a U.S.-based engineering services firm. The group subsequently disclosed the incident on its dedicated leak site and released approximately 5.3TB of design and project-related data. WorldLeaks also targeted the American construction company Thomas Bennett & Hunter, publishing internal project and operational data exfiltrated from the organization.

■ Focus on Ransomware

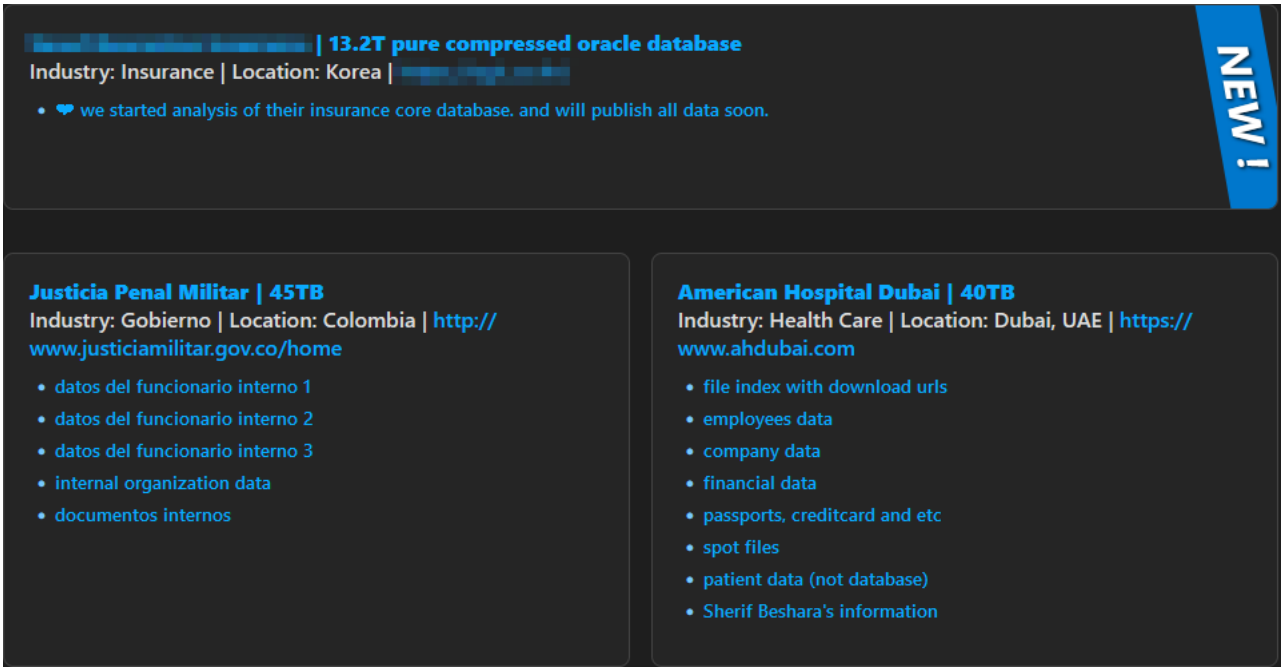


Figure 6. Gunra Dedicated Leak Site

The Gunra ransomware group was first identified in April 2025 and has since listed a total of 16 victims on its dedicated leak site. The site operates on the Tor network and specifies, for each victim, details such as company name, industry, country, the types of data exfiltrated, date of posting, and negotiation deadline. Gunra publicly discloses the nature and posting time of stolen data for each victim, and, if negotiations fail or the designated deadline passes, the group proceeds to release the exfiltrated materials in full on the dark web.

Gunra is characterized by the imposition of short negotiation deadlines and the use of multiple anonymous communication channels. Its ransom notes emphasize that victims must make contact within five days, providing both a Tox ID and an email address to facilitate communication. Initially, the group may offer complimentary decryption of selected files; if the victim fails to respond, Gunra escalates the pressure by listing the victim on its dedicated leak site and releasing a portion of the exfiltrated data. Should negotiations become protracted, the group threatens to publish additional data or even the entire dataset, thereby applying incremental pressure on the victim. Notably, among the published victims is a Korea financial sector company that suffered an attack in July 2025, for which Gunra issued an explicit warning regarding the release of the compromised data, significantly intensifying the coercion.

To date, two variants of the Gunra ransomware have been identified: one targeting Windows and the other targeting Linux environments. The Windows version leaves a ransom note in each directory following encryption, whereas the Linux variant does not generate a ransom note. Instead, it selectively encrypts files based on the path, file extension, and encryption ratio specified as execution parameters. Both versions employ a combination of full and partial encryption techniques—determined by file size and type—to maximize efficiency and speed. This report analyzes both variants, systematically outlining Gunra's operational methodologies and technical characteristics in order to facilitate effective preparedness against ransomware threats.

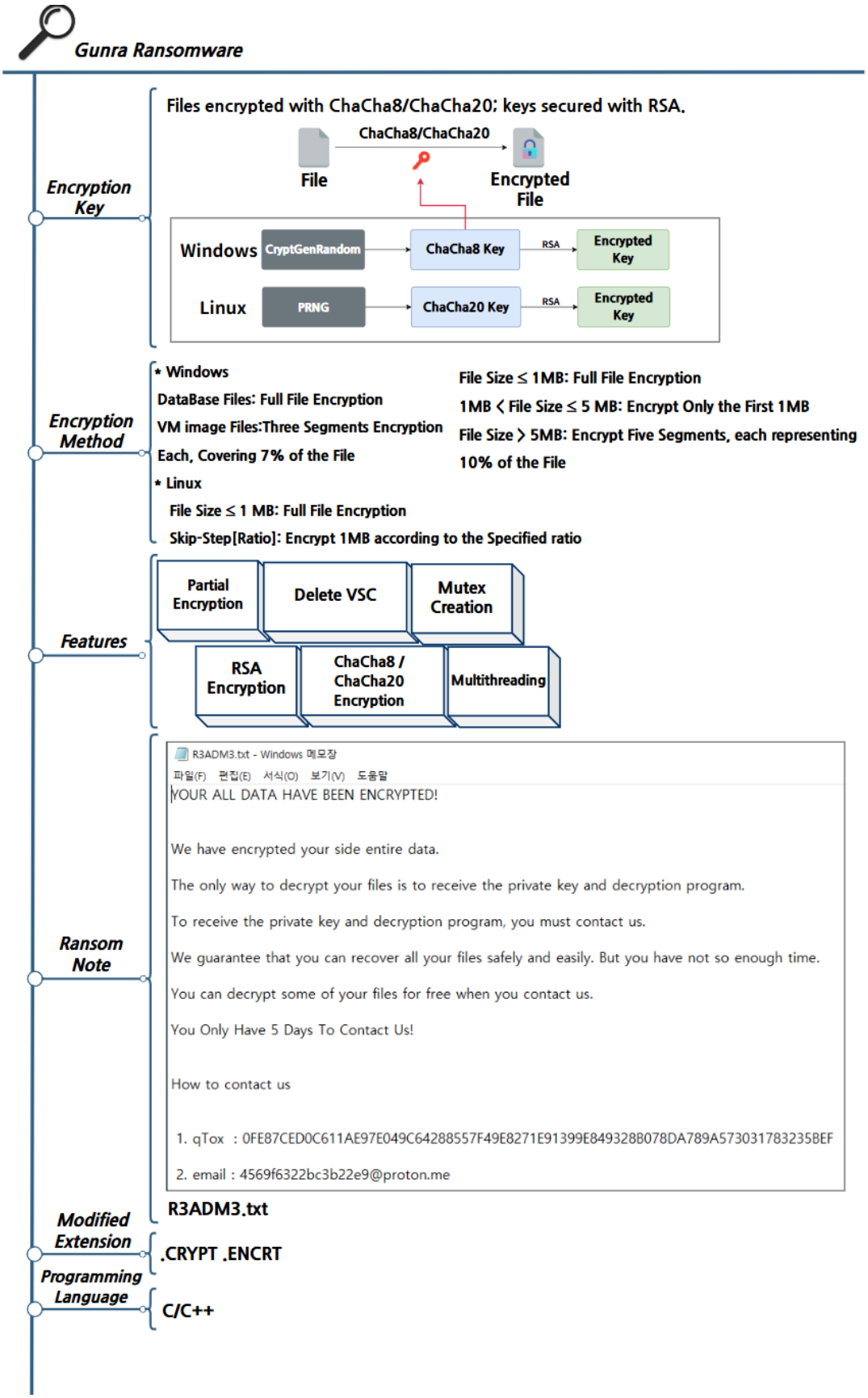


Figure 7. Overview of Gunra Ransomware

Gunra Ransomware Strategy

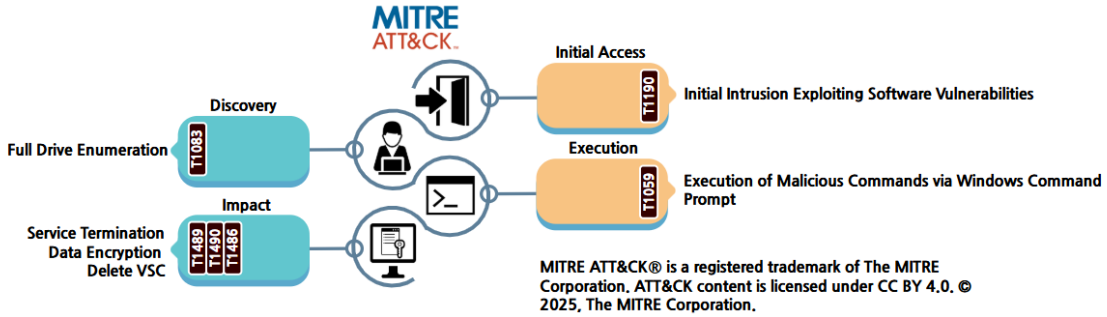


Figure 8. Gunra Ransomware Attack Strategy

Gunra Ransomware (Linux Version)

The Linux variant of Gunra ransomware is engineered to enable precise control over encryption behavior through a wide array of execution parameters, allowing attackers to flexibly specify target file locations, extensions, encryption intensity, and key storage methods. The arguments and functionalities of the Linux version are summarized in the table below.

Category	Description
--threads / -t	Specify the number of file encryption threads
--path / -p	Designate encryption targets
--exts / -e	Specify extensions of files to be encrypted (all: all files, disk: block devices)
--ratio / -r	Set encryption interval (in MB)
--keyfile / -k	Path to RSA public key file (.pem)
--store / -s	Path to store the encryption key
--limit / -l	Maximum encryption size (GB; 0: encrypt the entire file)

Table 1. Execution Parameters for Gunra Ransomware (Linux Version)

In the Linux version, the -p parameter is used to specify the target file or directory for encryption. If the specified target is a single file, only that file will be encrypted; if a directory is provided, the ransomware recursively traverses the directory and its subdirectories, encrypting all eligible files within.

-The -e option designates the file extensions to be targeted for encryption. If this option is omitted or set to 'all', all files—except those with explicitly excluded extensions—will be subject to encryption. When set to ‘disk’, only block device files present on the system are encrypted. Notably, files with the extension .ENCRT (indicating already encrypted files) and ransom note files named R3ADM3.txt are included in the list of extensions excluded from encryption.

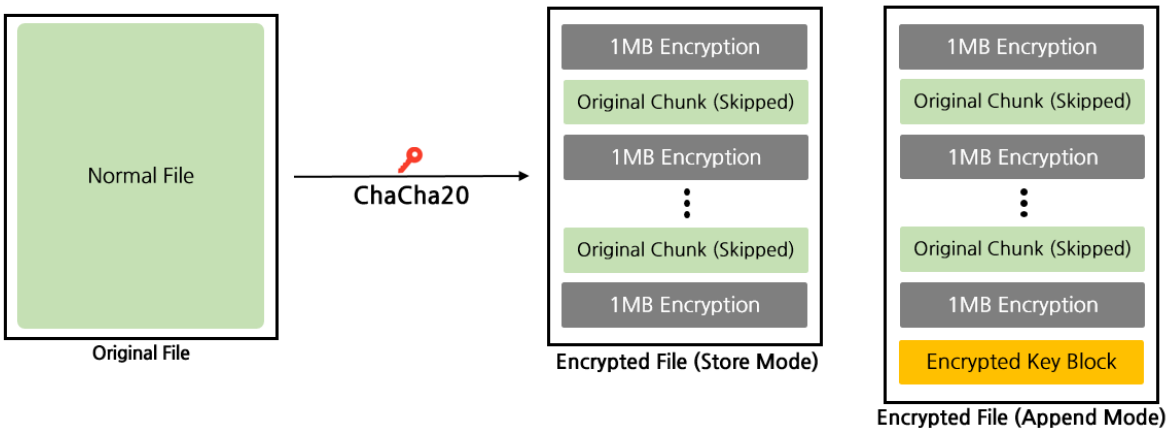


Figure 9. Encryption Process of Gunra Ransomware (Linux Version)

The Linux version of Gunra ransomware utilizes the ChaCha20 algorithm for file encryption. Target files are selected according to the path and file extension specified via the -p and -e parameters, and encryption is performed using a partial encryption method based on the value of the -r parameter. Gunra encrypts files in 1MB segments and, by employing the -r parameter, defines the size of the interval to skip after each encrypted 1MB block. For example, with -r=5, the ransomware encrypts 1MB, skips the next 5MB, then encrypts another 1MB, repeating this pattern throughout the file. Although the interval between encrypted segments varies according to the parameter, the size of each encrypted segment remains fixed at 1MB. Additionally, the -l parameter allows the attacker to specify the maximum encryption size in gigabytes. Upon completion of encryption, the ChaCha20 key, nonce,[5] as well as the values for the -r and -l parameters are stored separately, with the storage method determined by the presence or absence of the -s parameter.

When the -s parameter is used, Store Mode is enabled, and the encrypted key block is stored separately in the specified directory. During this process, the ransomware verifies the existence of the target directory and generates a key file named [Filename].keystore, based on the original file name. If the -s parameter is not specified, Append Mode is applied, and the encrypted key block is appended to the end of the encrypted file.

[5] Nonce: A randomly generated value used in encryption to ensure security and uniqueness.

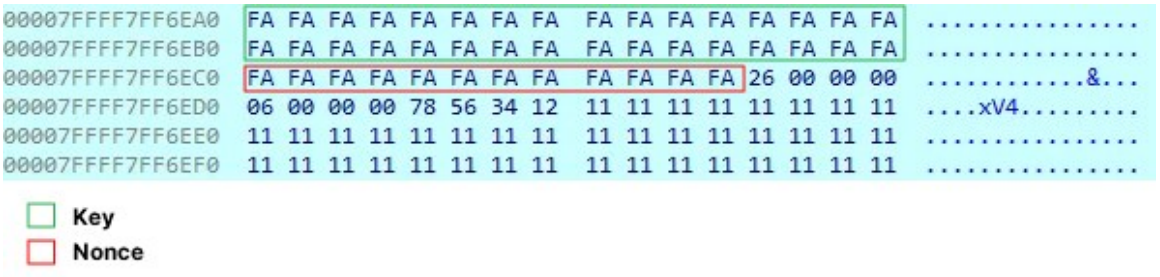


Figure 10. Vulnerable Key Generation in Gunra Ransomware (Linux Version)

Additionally, a design-level vulnerability has been identified in the key generation process of the Linux version. In standard implementations, the encryption key and nonce should be generated with sufficient randomness to ensure unpredictability, and then encrypted with an asymmetric key so that only the attacker can decrypt them. However, Gunra's Linux variant employs an inefficient approach, generating random values one byte at a time and concatenating them. During this process, the program calls the time(0) function for each byte, setting the current time in seconds as the seed. Since generating a 32-byte key and a 12-byte nonce takes less than one second, it is highly likely that multiple bytes will be generated using the same seed, resulting in repeated values. Even if the time changes during the generation process, the change in the seed is minimal, making it relatively easy for an attacker to predict the key and nonce.

Gunra Ransomware (Windows Version)

Unlike its Linux counterpart, the Windows version of Gunra ransomware is designed to operate without any external execution parameters. Critical values—such as the mutex name used to prevent repeated infections during execution and the RSA public key employed to protect the encryption keys—are embedded directly within the binary.

Upon execution, the ransomware creates a mutex named '375345635adfwef39' to prevent duplicate instances from running simultaneously. It then sequentially scans all system drives to generate ransom notes and identify files for encryption. During this process, only the user folder and its subdirectories within the C drive are traversed, whereas all other drives are scanned from their root directories. Specific folders, file extensions, and filenames are excluded from encryption, and the identified exceptions are listed in the table below.

Folder Name	Extension and File Name
tmp, winnt, temp, thumb, $Recycle.Bin, $RECYCLE.BIN, System Volume Information, Boot, Windows, Trend Micro	.exe, .dll, .lnk, .sys, msi, R3ADM3.txt, CONTI_LOG.txt

Table 2. Encryption Exceptions for Gunra Ransomware (Windows Version)

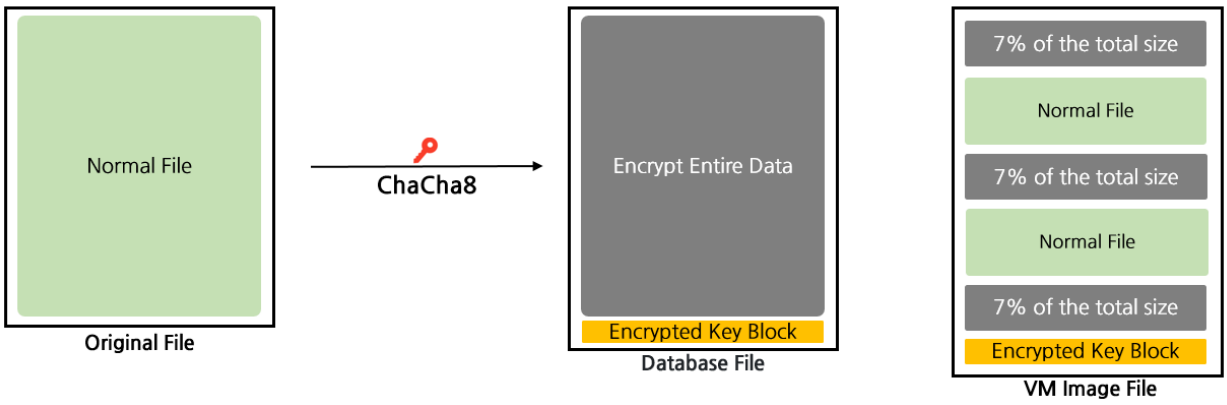


Figure 11. Gunra Ransomware Windows Version Encryption Method (Based on File Extension)

The file encryption method is determined by both the file extension and its size. Files associated with databases are fully encrypted regardless of their size. In contrast, files related to virtual machine (VM) images are partially encrypted: specifically, 7% of the file is encrypted at the beginning, middle, and end of the file—totaling 21% of the entire file, irrespective of its overall size. The corresponding file extensions for each category are listed in the table below.

Database-Related Extensions	VM-Related Extensions
.4dd, .4dl, .accdb, .accdc, .accde, .accdr, .accdt, .accft, .adb, .ade, .adf, .adp, .arc, .ora, .alf, .ask, .btr, .bdf, .cat, .cdb, .ckp, .cma, .cpd, .dacpac, .dad, .dadiagrams, .daschema, .db, .db-shm, .db-wal, .db3, .dbc, .dbf, .dbs, .dbt, .dbv, .dbx, .dcb, .dct, .dcx, .ddl, .dlis, .dp1, .dqy, .dsk, .dsn, .dtsx, .dxl, .eco, .ecx, .edb, .epim, .exb, .fcd, .fdb, .fic, .fmp, .fmp12, .fmpsl, .fol, .fp3, .fp4, .fp5, .fp7, .fpt, .frm, .gdb, .grdb, .gwi, .hdb, .his, .ib, .idb, .ihx, .itdb, .itw, .jet, .jtx, .kdb, .kexi, .kexic, .kexis, .lgc, .lwx, .maf, .maq, .mar, .mas, .mav, .mdb, .mdf, .mpd, .mrg, .mud, .mwb, .myd, .ndf, .nnt, .nrmlib, .ns2, .ns3, .ns4, .nsf, .nv, .nv2, .nwdb, .nyf, .odb, .oqy, .orx, .owc, .p96, .p97, .pan, .pdb, .pdm, .pnz, .qry, .qvd, .rbf, .rctd, .rod, .rodx, .rpd, .rsd, .sas7bdat, .sbf, .scx, .sdb, .sdc, .sdf, .sis, .spq, .sql, .sqlite, .sqlite3, .sqlitedb, .te, .temx, .tmd, .tps, .trc, .trm, .udb, .udl, .usr, .v12, .vis, .vpd, .vvv, .wdb, .wmdb, .wrk, .xdb, .xld, .xmlff, .abcddb, .abs, .abx, .accdw, .adn, .db2, .fm5, .hjt, .icg, .icr, .kdb, .lut, .maw, .mdn, .mdt	.vdi, .vhd, .vmdk, .pvm, .vmem, .vmsn, .vmsd, .nvram, .vmx, .raw, .qcow2, .subvol, .bin, .vsv, .avhd, .vmrs, .vhdx, .avdx, .vmcx, .iso

Table 3. Database and VM-Related Extensions

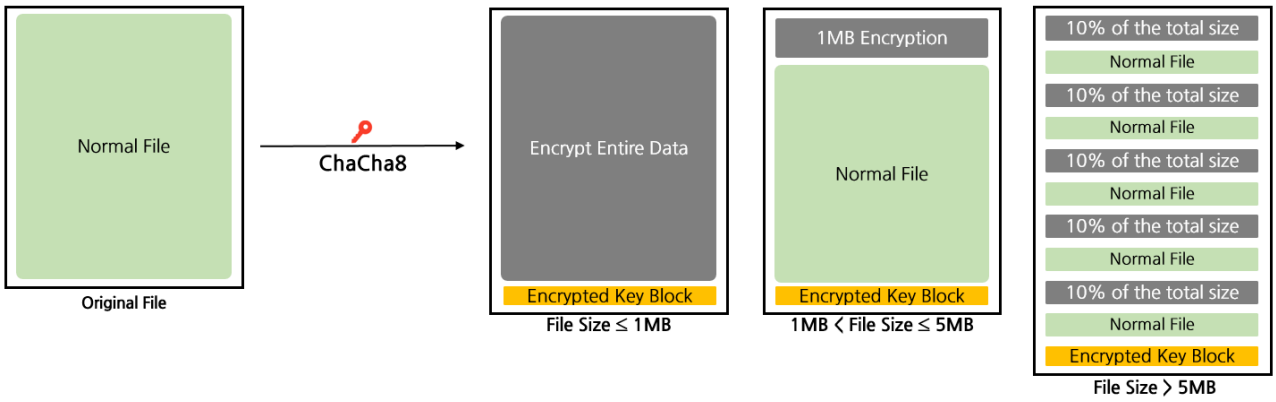


Figure 12. Gunra Ransomware Windows Version Encryption Method

All files other than those associated with virtual machines and databases are encrypted according to their size. Files that are 1MB or smaller are fully encrypted, while files larger than 1MB but not exceeding 5MB have only the first 1MB encrypted. For files exceeding 5MB, the entire file is divided into blocks, each representing 10% of the total file size, and only the odd-numbered blocks are encrypted.

After file encryption, the data required for recovery is appended to the end of the file. This includes the ChaCha8 key and nonce used for encryption, the original file size, and a 2-byte identifier specifying the encryption method employed. All of this information is encrypted with the attacker's RSA public key before being appended.

Gunra ransomware disables the system restore functionality to prevent victims from recovering their data. To achieve this, it enumerates all Volume Shadow Copy Service (VSS) entries and systematically deletes each one. During this process, the ransomware executes the query SELECT * FROM Win32_ShadowCopy via WMI [6]to identify all shadow copies present on the system. It then extracts the unique ID of each volume shadow copy from the query results and proceeds to generate and execute the following command to delete each identified shadow copy.

cmd.exe /c C:\Windows\System32\wbem\WMIC.exe shadowcopy where "ID='%s'" delete

Table 4. VSC Deletion Command

[6] WMI (Windows Management Instrumentation): A management interface that enables standardized querying and administration of components, status, and operational information within the Windows operating system.

Response Strategies for Gunra Ransomware

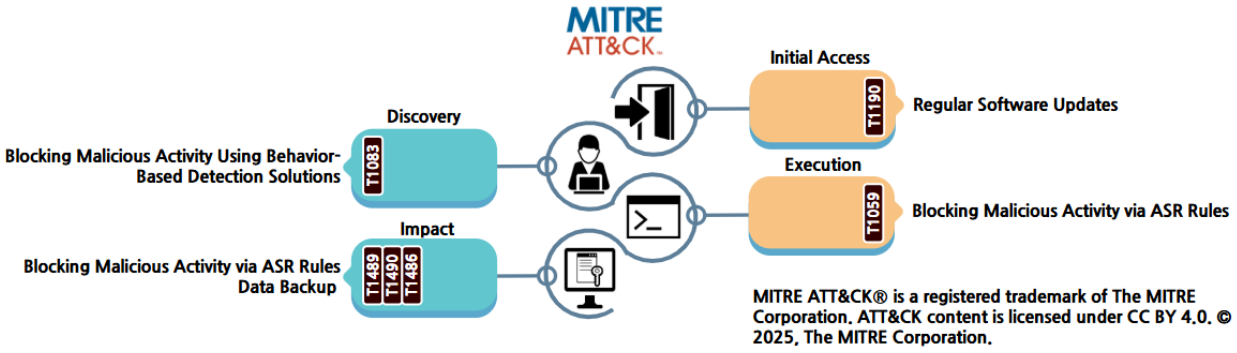


Figure 13. Mitigation Strategies for Gunra Ransomware

The Windows variant of Gunra ransomware utilizes the Windows command prompt to delete backup copies on the system prior to initiating file encryption. Consequently, enabling ASR (Attack Surface Reduction) rules allows for the proactive detection and blocking of abnormal processes related to backup deletion and encryption, thereby effectively mitigating malicious activity. In particular, it is critical to establish an environment capable of detecting and blocking actions such as the deletion of system restore points. Careful pre-configuration of security policies and the immediate blocking of unnecessary script execution attempts can also significantly contribute to the prevention of ransomware damage.

In addition, it is essential to deploy an EDR (Endpoint Detection and Response) solution and apply the latest security patches to swiftly identify and block intrusions exploiting known vulnerabilities or anomalous activities initiated locally. Such measures enable the real-time detection of behavior-based patterns that occur during the file encryption process and allow for the immediate termination of malicious processes. Furthermore, integrating EDR, antivirus, and log analysis systems for centralized monitoring of alert events ensures a robust response capability, even in the event of simultaneous attacks across multiple endpoints.

Additionally, regularly distributing backup copies across separate network segments, external storage, or offline media ensures data recoverability even if the primary system is encrypted. It is crucial to minimize access privileges to backup devices and conduct routine recovery tests to guarantee the integrity of backup data. Furthermore, dispersing backup data across different networks or storage solutions, as well as diversifying backup schedules and retention periods, can effectively mitigate the risk of ransomware attempts to delete backup copies.

These mitigation strategies are equally applicable to the Linux variant of Gunra ransomware as well as to Windows environments. In Linux environments—where critical infrastructure such as servers is frequently targeted—it is essential to enforce access control policies, restrict service ports, and strengthen administrator account management in accordance with the specific characteristics of the operating system. Regardless of the platform, implementing a multilayered security architecture can minimize damage in the event of ransomware infection and ensure rapid recovery.

IoCs

Hash(SHA-256)
91F8FC7A3290611E28A35A403FD815554D9D856006CC2EE91CCDB64057AE53B0
22C47EC98718AB243F2F474170366A1780368E084D1BF6ADCD60450A9289E4BE

References

• ArcticWolf (https://arcticwolf.com/resources/blog/arctic-wolf-observes-july-2025-uptick-in-akira-ransomware-activity-targeting-sonicwall-ssl-vpn/)

• Microsoft (https://www.microsoft.com/en-us/security/blog/2025/07/22/disrupting-active-exploitation-of-on-premises-sharepoint-vulnerabilities/)

• Thehackernews (https://thehackernews.com/2025/07/newly-emerged-global-group-raas-expands.html)

• Securit affairs (https://securityaffairs.com/180578/cyber-crime/fbi-seizes-20-btc-from-chaos-ransomware-affiliate.html)

Special Report

Zero Trust Security Strategy: System

Byung-gwon Hwang, SK Shieldus

■ Overview of the System Pillar

In the context of Zero Trust Architecture, the System Pillar encompasses all servers responsible for operating critical applications or storing and managing sensitive data. This domain includes not only physical and virtual servers, but also virtual machines running on hypervisors, databases, file servers, database servers, container and Kubernetes nodes, as well as public cloud instances—all of which fall within the scope of the System Pillar.

When applying Zero Trust Architecture to the System Pillar, the foremost consideration is the diverse range of system (server) operational environments. Unlike the past, when systems were predominantly on-premises, today's environments have expanded to include public, hybrid, and private clouds. Servers are now rapidly created, modified, and migrated at the level of virtual machines and containers. As the operational landscape grows more heterogeneous, the number of management items in the system domain—such as accounts, access paths, configurations, patches, and backups—increases, and these elements are often managed in a fragmented manner rather than through integrated processes. Therefore, it is imperative to standardize management policies for the System Pillar and to establish a consistent and unified management framework in conjunction with related systems.

Figure 1. Diverse System (Server) Operating Environments

In a Zero Trust environment, the significance of the System Pillar lies not merely in enhancing the security of individual servers, but in managing servers operating across heterogeneous environments according to unified standards. Given the diversity of operating systems and middleware, as well as the coexistence of physical servers, virtual machines, and containers, it is inherently challenging to implement distinct security policies for each server. Thus, the central imperative is to establish an integrated management framework grounded in centrally defined policies.

Unlike traditional approaches that assume "internal servers can be trusted," the System Pillar's methodology does not rely on such a presumption. Even after access is granted, accounts, sessions, commands, queries, and modification activities are continuously logged and monitored; privileges are assigned only for the necessary period and scope, and are automatically revoked upon expiration. The status of each server—such as patch levels, configuration compliance, vulnerability assessments, and backup verification—must also be evaluated in conjunction with account and privilege management, verifying both the user's identity and the system's current security posture.

Within Zero Trust Architecture, the System Pillar is not limited to the role of servers alone, but functions as the foundational space in which an organization's most critical resources reside. Accordingly, it is imperative to implement mechanisms that can identify and collectively manage systems deployed across a wide range of environments. The core elements and systems comprising the System Pillar, as outlined below, serve as essential reference points for establishing a robust Zero Trust environment.

■ Key Elements of the System Pillar

Within Zero Trust Architecture, the System Pillar serves as the central axis for the direct management and protection of all systems comprising an organization's core assets—including servers, critical applications, and data repositories. Systems distributed across diverse environments—on-premises, cloud, and hybrid—constitute the primary aggregation points for information and business operations within today's complex IT infrastructures, while simultaneously representing prime targets for both external and internal threats.

Notably, in a Zero Trust paradigm, trust based on the singular identity or physical location of a system is no longer valid; instead, continuous and granular verification and integrated management must be enforced across all systems, as well as accounts, resources, logs, and processes residing within them. Only through the synergistic integration of various administrative and technical elements—such as system inventory, account management, access control, security policies, patch management, vulnerability management, visibility, system segmentation, and policy administration—can organizations achieve genuine security levels that encompass data protection, operational continuity, and legal compliance across the enterprise.

The following section outlines the principal elements of the System Pillar and details specific management and technical measures required for their implementation, structured according to the Zero Trust maturity model.

1. System Asset Inventory

In a Zero Trust environment, system inventory management constitutes the foundational step in physically and logically identifying all core systems—such as servers, critical applications, and data repositories—operated within an organization, and ensuring that their status is continuously updated. This encompasses a broad spectrum of systems, including not only on-premises but also cloud and hybrid infrastructures: physical and virtual servers, containers, databases, and file servers. All such systems must be centrally registered and catalogued within an integrated asset management framework. Rather than relying on one-time registration at deployment, it is essential that key attributes (such as system owner, IP address, operating system, role, and configuration location) and state information are dynamically updated in real time throughout the entire system lifecycle—covering addition, modification, migration, and decommissioning—which forms the basis for policy automation.

Every asset within the system inventory should be logically grouped by operational environment (on-premises, cloud, hybrid) and functional role (e.g., web server, database server, file server). Grouping information must not be confined to static documentation or ad hoc data entry; instead, integrated asset management systems and monitoring tools must automatically reflect any changes, reclassify assets, and update group policies in real time as system changes occur. This enables unified management of security policies, access permissions, and monitoring frameworks for each group, and allows for the immediate identification and response to policy violations or anomalous activity.

Furthermore, the definition of system zones must transcend simple physical or logical segmentation by enabling multi-layered management according to business purpose, data criticality, network topology, and required security posture. Zone-specific controls should include differentiated access policies, micro-segmentation, session-based multi-factor authentication (MFA), real-time risk assessment, and policy automation. Traffic flows and access rights—both between and within zones—must be continuously and automatically adjusted via integration with asset management systems, ICAM (Identity, Credential, and Access Management), and unified monitoring tools, thereby ensuring real-time visibility.

In an optimized system inventory management framework, all changes in the status of system assets are reflected instantaneously. This real-time information underpins granular privilege control, efficient policy deployment, rapid anomaly detection and incident response, as well as systematic auditing and compliance management, thereby establishing the essential foundation for Zero Trust implementation.

2. System Account Management

In a Zero Trust environment, system account management entails the comprehensive cataloging and unified oversight of all accounts with access to organizational servers (including Unix, Linux, Windows, and others) and critical systems, based on their respective purposes and functions. Administrators must systematically manage not only privileged accounts, but also user-level and service accounts, classifying each by key attributes—such as usage status, privilege level, and group affiliation—and overseeing their entire lifecycle, including modification and decommissioning.

It is insufficient to rely on manual documentation for scattered account information across different systems. Instead, all account data must be centrally managed and updated in real time through an account management system or portal. Essential attributes for each account—such as privileges, affiliation, expiration, and lock status—should be automatically reflected and updated. Any changes in account status (creation, modification, deletion) must be immediately scrutinized for anomalies, with unauthorized or high-risk accounts promptly deactivated or otherwise remediated.

Account management must go beyond mere inventorying, enabling both manual and automated classification and grouping of accounts by criteria such as criticality, privileges, group membership, and usage status. The resulting categorized account information should be integrated with relevant systems to simultaneously enhance availability and security across the infrastructure.

Securing system accounts is paramount. To prevent unauthorized access and account misuse, security settings for each account—including access restrictions, expiration, and least privilege—must be consistently enforced, whether natively or through integration with account management systems or ICAM (Identity, Credential, and Access Management) platforms. When accounts are created or deleted, pre-defined security policies—tailored by operating system (Linux, Windows, macOS, etc.)—should be automatically applied. Security configurations must be linked with unified monitoring and log analysis systems to support real-time monitoring and auditing.

For password management, each account should adhere to stringent password policies (such as minimum length, complexity requirements, and regular rotation) and multi-factor authentication (MFA) should be enforced for high-value accounts. Password status, policy compliance history, and change records must be centrally managed via integration with ICAM, authentication, and monitoring systems, while all related activities are logged. Furthermore, linkage with SIEM/SOAR platforms should enable immediate detection and remediation of at-risk accounts or anomalous password changes.

3. System Access Control

System access control, in accordance with Zero Trust principles, mandates that all permissions be granted based on the principle of least privilege, with granular access rights configured for each system according to its specific function and operational requirements—referencing the system inventory as the authoritative baseline. Access control must encompass a comprehensive range of components, including authentication, authorization, and access management, and should not be limited to controls within individual systems. Instead, enterprise-wide privilege management should be achieved by leveraging integrated management systems such as ICAM, in conjunction with network and application layers.

Each system should enforce access restrictions based on various criteria—such as IP addresses, ports, and accounts—through access rights settings. Where appropriate, integration with SSO (Single Sign-On) and IAM (Identity and Access Management) solutions should facilitate the assignment of granular permissions using RBAC (Role-Based Access Control) or ABAC (Attribute-Based Access Control) models. For real-time monitoring and analysis, access permissions must be managed dynamically via integration with ICAM, SIEM (Security Information and Event Management), SOAR (Security Orchestration, Automation, and Response), and XDR (Extended Detection and Response) systems, ensuring that privileges can be automatically adjusted in response to emerging threats.

Command control within systems involves identifying and managing high-risk or vulnerable commands used in actual operations. Robust policies must be established to specify which commands require control and auditing. Rather than relying solely on per-system settings or shell-based restrictions, command control should be systematically applied through access control platforms, Secure OS, or unified management systems, supporting comprehensive change tracking and real-time monitoring. The usage patterns and anomalies associated with controlled commands should be analyzed and remediated in real time via integration with SIEM, SOAR, and similar platforms, with ongoing verification of the effectiveness and timeliness of these controls.

Real-time session control entails managing the granting and revocation of system access on a per-session basis. Access control systems must monitor and manage the entire lifecycle of each session—including initiation, maintenance, extension, and termination—in real time. Upon detecting anomalous activity, immediate measures such as session termination or additional authentication must be enforced. Rather than relying exclusively on individual system configurations, session policies should be finely tailored per account, group, or business function, through close integration with account and access control systems. Real-time session data should also be linked with unified monitoring and analytics platforms to elevate the overall security posture.

4. System Security

In a Zero Trust environment, system security management entails defining security policies for critical systems such as servers and establishing an effective governance framework to enforce them. Security policies must be explicitly specified for each system group or asset, and enforcement should extend beyond native system security configurations to include integration with asset management platforms, unified monitoring systems, and other centralized controls. Whenever systems are added or modified, security settings should be automatically applied according to predefined policies, and real-time monitoring must enable immediate response to any changes in system status.

For system components and critical data, regular backup and recovery mechanisms must be maintained. To protect against risks such as hardware failures, software errors, or intrusions, backup systems should periodically capture essential configuration files, databases, logs, and other critical information. Recovery plans should be in place to ensure rapid restoration in case of incidents. Redundant configurations and disaster recovery (DR) centers should be employed to facilitate swift recovery during catastrophic events, and backup and recovery policies must be automatically enforced whenever system changes or anomalies occur.

Internal system processes—including creation, execution, monitoring, and termination—must be systematically managed. Critical or high-risk processes should be clearly defined for each system, with execution controlled based on privileges. Monitoring systems should observe the real-time status of designated processes, and any abnormal termination or unexpected behavior must trigger immediate investigation and corrective action. Process anomalies should be managed through automated mechanisms, such as alerts, to ensure rapid response.

Regarding system security functions, regular checks should be conducted on areas including software updates, integrity of critical files, antivirus and malware defenses, and log health. Rather than relying on manual checklists, results should be continuously visualized through integration with antivirus solutions, inspection systems, and vulnerability management platforms. Any detected issues must be addressed immediately. Inspection criteria and items should be periodically updated and applied via automated tools such as monitoring systems and machine learning, and the results should be systematically documented in reports or other formats for ongoing review and compliance.

5. System Segmentation

System segmentation refers to the practice of managing specific servers and critical systems through physical or logical separation based on the system inventory. Segmentation can be implemented via physical infrastructure modifications, network isolation, virtualization (VMs), or access control policies for logical separation, and should be applied according to each system's role, criticality, and service characteristics. When implementing segmentation, designs must maintain compatibility with existing systems while allowing flexible expansion as new systems are introduced. Additionally, dedicated monitoring, surveillance, and security controls must be established for segmented systems. Integration with asset management systems should ensure that when new systems are added to the segmentation scope, labeling policies are automatically inherited and both logical and physical separation are systematically maintained.

For highly critical systems, more granular measures are required to ensure effective management and protection. Various administrative and technical controls—including incident management, change management, patch management, and backup and recovery—should be applied in accordance with the segmentation policy. Services such as web servers and database instances should be physically or logically separated and managed systematically. Changes in the status of critical systems, as well as anomalies, should be continuously monitored in real time, with automated response mechanisms in place where necessary. When new systems are added or the environment changes, detailed management policies and technical measures should be automatically applied to preserve the overall security posture through a dedicated management framework.

6. System Policy Management

System policy management refers to the establishment and consistent enforcement of administrative policies designed to ensure the secure and efficient operation of system environments. Management policies for systems should encompass a wide range of functions, including system operations, access control, security, documentation, reporting, and analysis, while also satisfying non-functional requirements such as accuracy, completeness, consistency, user convenience, scalability, and security. When developing policies, considerations must include compatibility with existing systems, relevant legal and regulatory compliance requirements, and internal standards. Policies should be systematically documented and managed, drawing on corporate guidelines and standard frameworks. Utilizing centralized management systems, policies must be applied consistently across all systems, with integration into monitoring platforms to allow automatic updates based on operational analysis and seamless application to new systems.

Exception management is also essential in system policy administration. Servers requiring exception policies typically include those with unique functions, systems used for testing new technologies or features, and systems handling critical data. A separate exception policy must be established, detailing the list of exempted servers, prioritization, and monitoring and reporting procedures. Systems requiring exceptions should be systematically cataloged and managed through either manual processes or centralized management systems. Integration with monitoring tools must ensure that exception-related items are reflected in real time, allowing immediate response to policy violations or anomalous activity.

In a Zero Trust framework, system policies are not static. As system environments continuously evolve, policies must undergo ongoing evaluation, modification, training, and documentation updates. Analysis systems should be leveraged to assess potential risks associated with policies, while automated policy generation ensures that changes are propagated and enforced across systems without manual intervention. The maturity of a policy management framework is determined by the extent to which these continuous management and automation processes are implemented, thereby enhancing both organizational security posture and operational efficiency.

Similarly, in a Zero Trust environment, network segmentation strategies extend beyond simple physical boundaries. By combining granular access controls tailored to diverse business environments and asset characteristics with automated policy enforcement, organizations can minimize internal risks, prevent lateral movement, and simultaneously achieve a flexible and resilient security environment.

7. System Patch Management

A system patch management policy must explicitly define the standards and procedures for applying security patches to all system components, including operating systems, applications, and firmware. The policy should cover the entire patch lifecycle, including the selection of target systems, patch deployment and installation procedures, backup and recovery measures in case of patch failure, and real-time monitoring of patch compliance. Management should be conducted consistently and systematically through centralized management systems or Patch Management Systems (PMS), ensuring that integration with external patch servers allows immediate reflection of the latest patch policies and continuous maintenance of systems in a secure, up-to-date state.

Patch deployment and execution must be applied accurately and consistently across all servers and systems in accordance with the management policy. All stages of patch deployment—such as patch listing, prioritization, distribution and installation, and pre-deployment functional testing—should be standardized and automated. Approved deployment tools (e.g., PMS) must deliver patch files to target systems, with backup and recovery mechanisms enabling rapid rollback in the event of failure. Newly released patches should first be validated in isolated environments, such as sandboxes, before deployment; any issues detected in the sandbox should trigger automatic exception handling according to PMS policies. These procedures ensure both operational stability and user convenience.

System patch monitoring involves real-time oversight of the entire patch deployment process, enabling immediate detection of installation status, omissions, failures, or delays. Utilizing PMS, integrated monitoring systems, and analytics tools, organizations should visualize patch status in real time, track patch adoption trends, identify causes of failures, and manage follow-up actions such as redeployment and automatic rollback. Monitoring outputs should be automatically generated and distributed in report formats, with instant alerts and analysis enabling automated remediation processes to maintain system security and compliance.

8. System Log Management

System log management requires clearly defining which logs should be collected for each system and establishing a framework for real-time collection and storage according to a formal log collection policy. Collection methods may include built-in system tools, log agents, and custom scripts, ensuring comprehensive capture of all necessary logs, such as custom application logs and audit logs. Real-time and periodic collection targets should be managed separately to optimize efficiency. A centralized environment must be established to enable enterprise-wide real-time collection and integration of system logs, while ensuring data integrity and security through secure storage, transmission, and retention practices.

An effective log management framework must include a robust indexing system. Real-time indexing, search, filtering, pagination, highlighting, and visualization capabilities are essential. Tools such as Splunk, Elasticsearch, or Graylog can be employed to assign and manage index values for critical logs, while continuously improving the overall log management process. Integration with log collection and analysis systems enables real-time monitoring and response capabilities.

System log analysis should provide actionable insights into system activity, performance, and security posture through real-time, correlation, and visual analysis. Using centralized log systems and SIEM platforms, diverse log sources can be correlated to identify patterns, inform system improvements, and anticipate issues such as errors, security threats, or performance degradation using historical data and machine learning. Automated response mechanisms can also be incorporated to mitigate risks proactively.

Within the overall log management framework, periodic automated generation and management of summary, detailed, and comparative reports are necessary to maintain comprehensive visibility into system health. Reports should be deliverable in multiple formats, including HTML, PDF, and CSV, with scheduled distribution. Applying machine learning–based automated analysis allows for real-time detection of critical events and risk factors, enabling immediate remediation. Continuous refinement of the report management process ensures rapid and effective operational response.

9. System Vulnerability Management

A system vulnerability management policy must clearly define the objectives and scope of vulnerability management to maintain system security levels and ensure regulatory compliance. The policy should establish a systematic process covering the entire lifecycle of vulnerability management, including identification, remediation, analysis, and reporting. It should specify the definition of vulnerabilities, severity assessment criteria, diagnostic methods, patch deployment or code remediation procedures, mitigation strategies, and reporting and management protocols. To maintain currency, the policy must be continuously updated and automatically applied across all systems through integration with SOCs, threat intelligence (TI) platforms, and other sources to collect the latest vulnerability information and reflect updates in real time.

Vulnerability detection and remediation should employ both automated and manual scanning, as well as publicly available vulnerability databases such as CVEs, to rapidly and accurately identify weaknesses. Identified vulnerabilities must be prioritized according to policy, and swift remediation—such as patch deployment, code modification, or mitigation measures—must be implemented. Periodic assessments and the integration of the latest vulnerability information into the system ensure consistent coverage, while real-time diagnostics and automated patching should be applied according to risk levels.

Impact assessment of vulnerabilities involves analyzing the root cause and evaluating their effects on the system from multiple perspectives to determine priority. Factors such as exploitability, operational impact, and the necessity for preventive or mitigating measures must be objectively assessed. Based on this evaluation, appropriate responses—including patching, mitigation, or acceptance—should be defined. Integration with ICAM, TI systems, and real-time database feeds facilitates automated analysis and policy enforcement.

Vulnerability management extends beyond mere detection and remediation. Integration with SIEM and other monitoring systems enables real-time tracking and analysis of vulnerability events. All management records, including vulnerability reports, remediation status, and mitigation plans, should be documented through automation tools. Deep analysis using machine learning and big data should be employed to continuously improve both the effectiveness and efficiency of responses. Moreover, integration with antivirus, monitoring, and vulnerability management systems ensures that automated response processes are triggered during risk events, with reports and status updates generated and distributed on a scheduled basis.

10. System Visibility and Analytics

Ensuring system visibility involves establishing a framework for real-time monitoring and analysis of server status, performance, anomalous activities, and security threats, enabling early detection and rapid response to issues. Monitoring policies should be applied across all systems, continuously collecting and analyzing key metrics such as CPU, memory, and disk utilization, critical process states, and other system indicators, while implementing mechanisms for anomaly detection and alerting. Monitoring systems should provide not only system-level visibility but also comprehensive insights across the entire infrastructure. Any changes in system states must be automatically reflected in the monitoring framework, with integration into centralized log and analytics platforms to ensure consistent operations and management.

System analytics capabilities are central to achieving deep understanding and optimization of the system environment through in-depth analysis of data collected from servers. Beyond simple data collection, logs and other system data should be visualized in real time or stored for long-term analysis to identify issues and derive improvement measures. Collected data must undergo cleansing, transformation, and integration, enabling examination through a variety of analytical techniques. Insights from monitoring and analytics processes should feed directly into operational and security policy adjustments, ensuring real-time improvements and enforcement. This framework should be continuously refined through regular reviews and process improvement activities to enhance overall effectiveness and responsiveness.

11. Policies and Processes

System operational procedures constitute a critical component of an organization's IT system management, playing a vital role in achieving both business continuity and system security through efficient and stable operations. Server operational procedures must provide clear and consistent standards across key areas, including system installation, configuration, monitoring, maintenance, and documentation, and should be established and managed from a governance perspective. From a Zero Trust standpoint, operational procedures must embed security-enhancing principles such as least privilege, continuous authentication and authorization, decoupling security from network location, data-centric protection, and rapid incident response. Beyond traditional perimeter-based models, these procedures should incorporate real-time feedback and enforcement mechanisms.

Minimum privilege management is a core principle. Access should be restricted to the smallest necessary group, and all unauthorized access must be proactively blocked. Users should be granted only the minimum permissions required to perform their tasks, implemented through mechanisms such as RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control). In accordance with least privilege principles, permission management must be integrated with account and authentication management systems, with procedures for revoking privileges and approval workflows established to prevent misuse across the system. High-value information systems should employ additional layers of control, such as isolated environments or dedicated equipment, to establish multi-tiered defense mechanisms.

Management of personal data systems is equally important. Systems storing personal information must implement both administrative and technical privacy protection policies, aligned with applicable laws and compliance standards. Integration with privacy management systems and portals enables centralized oversight of all systems containing personal data, with automated lifecycle management to monitor and control data flow. Policy development should encompass personal data lifecycle management, flow tracking, and access controls to minimize risks of leakage or misuse.

Building on these elements, the System Pillar functions as the central axis for the direct management and protection of all organizational core assets—including servers, critical applications, and data repositories—within a Zero Trust Architecture. In today's complex and distributed IT infrastructure, systems not only handle the majority of operational data and business processes, but also represent high-impact targets in the event of security incidents, necessitating robust protection against both external and internal threats.

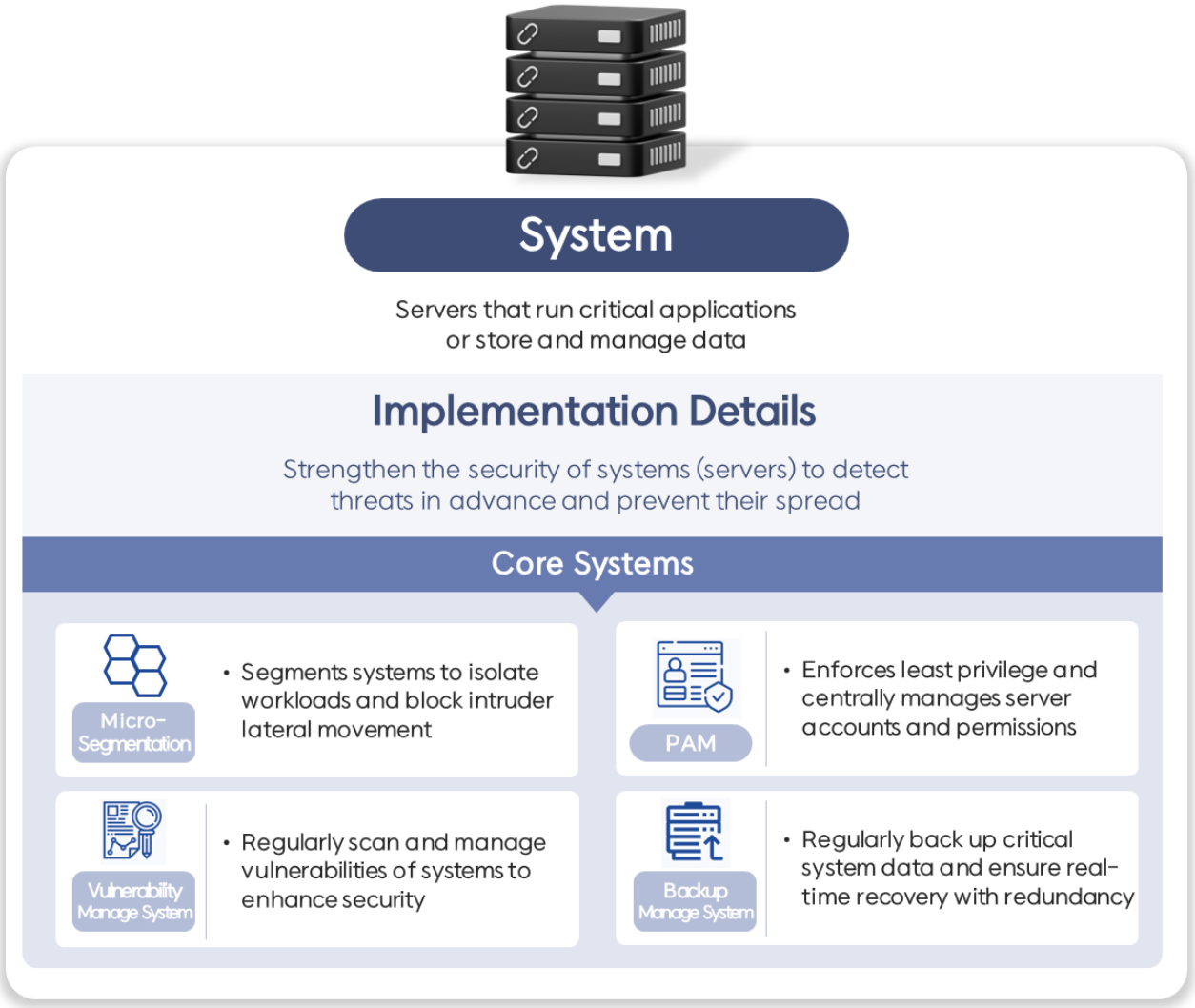
In a Zero Trust environment, access control based on server location or pre-existing trust relationships is no longer sufficient. Each system must implement strong individual authentication, least privilege principles, real-time security monitoring, granular access control, and process management, with these mechanisms interlinked and mutually reinforcing. Additionally, integrated operational and technical controls—including system inventory, account management, security policies, vulnerability and patch management, system segmentation and policy automation, log collection and analysis, and real-time visibility—are required to achieve a substantive Zero Trust security posture across all systems.

The advanced implementation of the System Pillar establishes a management framework and technical foundation that consistently applies Zero Trust principles across the organization's server infrastructure. This enables early detection of anomalies at the system level and rapid response to potential threats. Effective deployment of the System Pillar is essential for securely protecting critical data and core business processes, while safeguarding organizational infrastructure against evolving IT environments and increasingly sophisticated cyber threats.

■ Implementation of Zero Trust Functions for Key Systems

To successfully implement a Zero Trust environment, both technical measures and the systems capable of executing them are essential. Zero Trust Architecture is founded on the principle of "never trust, always verify," and achieving this requires systems that can continuously assess the state of each system, perform ongoing verification, and enforce least-privilege access.

The key systems outlined below play critical roles within a Zero Trust environment and are interconnected to strengthen the organization's overall security posture. For each system, we examine the specific functions necessary to implement Zero Trust principles and the security benefits that organizations can derive from their effective deployment.



* Source: SK Shieldus, "The Beginning of Zero Trust: Realized with SKZT"

Figure 2. Key Systems within the System Pillar

1. PAM (Privileged Access Management)

In the System Pillar, Privileged Access Management (PAM) serves as the central security system. Integrated with the Identity Pillar's IAM (IDP), PAM ultimately enforces and audits "who can access what, from where, when, and to what extent" within a Zero Trust environment. Modern PAM solutions extend beyond traditional server- and database-centric controls to cover enterprise applications, network and security devices, cloud consoles (AWS, Azure, GCP), and various SaaS platforms, allowing comprehensive privileged access management from a single interface. In other words, PAM uniformly governs system access control and database access control while applying consistent principles and procedures across remote access, cloud management consoles, and web- or API-based administration interfaces.

Traditional PAM typically focused on system and database access control, using installed agents to monitor and record sessions. With the expansion of managed environments to applications, SaaS, and cloud infrastructures, web-based architectures have rapidly gained adoption. In this model, a bastion (proxy gateway) intermediates all sessions, allowing users to access required resources via web consoles under least-privilege policies without local keys or accounts. This approach enables consistent control over assets where agent installation is impractical, as well as externally managed services, making it highly favored in operational environments.

Continuous verification—a core principle of Zero Trust—is implemented in PAM through real-time reassessment of privileged sessions. Even after session initiation, signals such as user and device status, access location and time, executed commands, and query patterns are continuously evaluated. Upon detecting risk, PAM can require additional MFA, progressively reduce privileges, or automatically terminate the session. This ongoing verification applies uniformly across servers, databases, and SaaS platforms.

Secure channels and sensitive information management are also fundamental to PAM. SSH keys and privileged passwords are stored and rotated in a secret vault, with access proxied through SSH/SSL/TLS tunnels to avoid key exposure. All activities—including console access, commands, file uploads/downloads, queries, and data extraction—are logged with detailed metadata, supporting incident reconstruction and regulatory audits.

In summary, PAM acts as the single gateway for privileged access within the System Pillar. While IAM (IDP) authenticates “who” the user is, PAM determines and enforces “what, how far, and under what conditions” access is permitted, recording and auditing the results. In a Zero Trust environment, PAM operates as an integrated system across servers and databases via agents/proxies, and across applications and SaaS platforms via web-based proxies, unifying management across both cloud and on-premises infrastructures.

2. Micro-Segmentation

Micro-Segmentation is an advanced security strategy that offers a finer-grained approach compared to traditional macro-segmentation. It separates the network at the OSI Layer 7 (Application layer) level, down to the granularity of business functions, users, and applications, enforcing access controls based on the principle of least privilege.

Where conventional network segmentation primarily relies on physical or logical boundaries such as IP addresses, ports, or VLANs, Micro-Segmentation focuses on the relationships between services and applications, their purposes, and actual traffic flows. This logical division allows organizations to precisely control internal threats and prevent lateral movement by attackers within the network.

Implementation of Micro-Segmentation can be categorized into two approaches: network-based and system (host)-based. Within the System Pillar, Micro-Segmentation is primarily system-based. System-based Micro-Segmentation deploys either agent or agentless solutions on endpoints such as servers or workstations, applying granular security policies and access controls at the individual system level. During this process, the topology between systems and the network is visualized, and actual network traffic flows between applications and services are analyzed to automatically generate and manage segmentation policies. Recent advancements incorporate AI and machine learning to optimize system-to-system paths, detect anomalies, and recommend policy adjustments, enhancing operational efficiency.

The core principle of implementing Micro-Segmentation in the System Pillar is shifting from "zone-level firewalls" to "per-system firewalls." Historically, firewalls were deployed at critical network segments to block major traffic flows. Micro-Segmentation, however, applies policies as if each server has its own firewall, ensuring that lateral movement is blocked even if a breach occurs within the network. While this granular segmentation significantly improves security, it also increases policy complexity. AI and machine learning technologies are leveraged to mitigate this complexity through functions such as learning normal traffic patterns, policy recommendations, consolidation of redundant or unnecessary rules, pre-change simulation, and automated alerts for anomalies. Real-world implementations have demonstrated that AI-enabled Micro-Segmentation solutions effectively enhance policy enforcement and operational management.

3. System (Server) Vulnerability Management System (VMS)

A System (Server) Vulnerability Management System is a critical tool designed to continuously detect, assess, and remediate security weaknesses across an organization's servers, network devices, and cloud instances, thereby reducing risk. It performs periodic or continuous scanning of operating systems, middleware, applications, databases, and web/service processes, organizing the results according to risk levels for effective management. Vulnerability remediation progress, patch deployment status, unresolved issues, and recurrence rates are tracked and visualized through dashboards and reports.

In practice, both agent-based (installed on the server) and agentless (remote authentication scan) methods are employed. Beyond simple version comparisons, authenticated scans assess configuration vulnerabilities such as misconfigurations, unnecessary services, excessive privileges, and weak encryption. In cloud environments, instances that are transient or part of auto-scaling groups are automatically registered and scanned via tags/labels, and golden images (AMIs/templates) are periodically reviewed. Containerized environments are analyzed separately at the host OS and container image levels, with CI/CD pipeline scans performed to identify risks before deployment.

Vulnerability prioritization does not rely solely on CVSS scores. Risk scoring incorporates factors such as known exploited vulnerabilities (KEV), exploit probability (EPSS), internet exposure, business criticality, data sensitivity, and potential for lateral propagation. Based on this prioritized view, patch campaigns are planned and executed according to a standardized remediation playbook, which includes maintenance windows, rollback procedures, and pre/post functional verification. For vulnerabilities that cannot be immediately remediated, exceptions are documented with timeframes and rationale, while compensating controls—such as firewall blocks, WAF virtual patches, privilege reduction, service isolation, and file integrity monitoring—are automatically applied to mitigate residual risk.

From a Zero Trust perspective, a vulnerability management system quantifies the "trust level" of each server and integrates this data with other security tools and policies. For instance, if a server has high-risk unpatched vulnerabilities, access can be restricted through ZTNA or NGFW, PAM can limit privileged access, EDR can isolate the affected server, and IAM/SSO systems can enforce MFA on related administrative sessions. This establishes a dynamic, real-time vulnerability-based framework within the System Pillar, enabling effective implementation of a Zero Trust environment.

4. Backup & Recovery Management System

A Backup Management System is an operational platform designed to create, store, verify, and restore backups to rapidly recover services in the event of system or server failures or security incidents. Rather than merely saving individual files, the system regularly protects complete server images—including operating systems, applications, configurations, and databases—and allows mounting for immediate service restoration or selective recovery of specific files, emails, or database objects. Within a Zero Trust environment, the Backup Management System manages these functions across on-premises, virtualized, cloud, and SaaS environments in a unified manner.

Backup targets are automatically discovered and registered using both agent-based and agentless methods, and snapshots are created to ensure application consistency. Only changed blocks are transmitted to storage, reducing network and storage overhead, while deduplication and compression improve storage efficiency. Backups are distributed across local storage and remote object storage, with critical segments optionally stored in WORM storage to prevent deletion or tampering. Periodic automated verification procedures, including booting and application checks, ensure that backups are recoverable, with results displayed on dashboards and reports.

The same principles apply to cloud and container environments. In the cloud, newly created instances are automatically included in backup policies through tag/label integration, with disk-level backups orchestrated via snapshot APIs. Kubernetes environments preserve etcd, resource manifests, and persistent volumes, enabling namespace-level restoration. CI/CD pipelines are integrated to capture snapshots before and after deployments, allowing rapid rollback. SaaS data—including Microsoft 365, Google Workspace, and Salesforce—is similarly protected and recoverable under the same policy framework.

For advanced backup management, the system must reflect organizational disaster recovery (DR) strategies. Conceptually, cold sites minimize costs but have longer recovery times, relying on backups and configuration (including infrastructure code) to spin up environments as needed. Warm sites use periodic replication and snapshots to pre-stage critical services, achieving intermediate RTO/RPO. Hot sites employ synchronous or low-latency replication and automatic failover to minimize recovery time, albeit at higher cost. The Backup Management System automates these scenarios through runbooks/playbooks—covering sequences, dependencies, and verification—and can conduct uninterrupted DR rehearsals in isolated environments during operational hours, executing failover and failback procedures in actual incidents.

Zero Trust controls are also integrated. High-risk functions such as backup console access and permanent deletion are governed through SSO/IAM and PAM, requiring MFA and approval. Dedicated backup network segments are separated from the operational network using ZTNA, NGFW, or Micro-Segmentation. During backup, suspicious files or anomalous patterns are isolated, and detection results are fed to SIEM/SOAR systems for automated alerts and follow-up actions. Actual recovery is first validated in isolated environments before being applied to production infrastructure.

The Backup Management System ensures business continuity (BCP) and enforces organizational policies, guidelines, and procedures, thereby maintaining the availability and reliability of the System Pillar.

Within the System Pillar of a Zero Trust architecture, key resources stored on servers are centrally controlled using PAM, Micro-Segmentation, Vulnerability Management Systems, and Backup Management Systems. Privileged access is centrally managed, inter-server communications are finely segmented and restricted, vulnerabilities are continuously assessed and remediated, and recovery from failures or incidents is integrated into a single workflow. These core systems interact with other pillars' key systems—including IAM, ZTNA, and SIEM & SOAR—to sustain and strengthen trustworthiness and availability across both on-premises and cloud environments in a comprehensive Zero Trust framework.

■ Conclusion

Within a Zero Trust Architecture, the System Pillar is a concept unique to domestic (Korean) guidelines and does not exist as a separate pillar in most international frameworks. Globally, servers and related resources are typically managed under the Device or Endpoint Pillar. In Korea, however, due to network-segmented environments and a predominance of on-premises operations, the management and protection of systems (servers) are considered critical. Accordingly, KISA included the System Pillar as a distinct category when publishing Zero Trust guidelines tailored to domestic environments.

The primary focus of the System Pillar in a Zero Trust context is the unified management and consistent application of security policies across systems deployed in diverse environments, including on-premises, public cloud, and private cloud. Effective centralized control of the System Pillar requires both appropriate policies and supporting systems. Management standards and policies should be defined based on core elements such as system inventory, account management, access control, policy management, and patch management, and implemented using systems such as PAM (Privileged Access Management), Micro-Segmentation, Vulnerability Management, and Backup Management Systems.

Because the majority of servers are existing operational systems rather than newly deployed, implementing Zero Trust Architecture must account for backward compatibility, which represents one of the greatest challenges. Given the diversity of operating systems and middleware, the System Pillar may employ a mix of agent-based and agentless approaches, while unsupported systems require custom control policies for monitoring and management.

In conclusion, the System Pillar has been classified separately to reflect domestic operational environments, aiming to manage critical system resources under a Zero Trust framework. The System Pillar does not function in isolation; rather, it is designed to integrate organically with other pillars—Identity, Network, and Data—to enable the full implementation of a Zero Trust Architecture tailored to an organization's environment.

■ References

[1] KISA, Zero Trust Guidelines V2.0, December 2024

[2] NIST SP 800-207, "Zero Trust Architecture", 2020.08

[2] NIST SP 1800-35 Final,"Implementing a Zero Trust Architecture:High-Level Document", 2025.06

[3] NIST SP 800-34, "Contingency Planning Guide for Federal Information Systems", 2010.11

[4] DoD, "Zero Trust Overlays", 2024.06

[5] National Cybersecurity Center(South Korea) National Network Security Framework Guidelines (Draft), January 2025

■ Additional Resources

[1] SK Shieldus, "The Beginning of Zero Trust: Realized with SKZT" – Brochure

[2] Gartner, "Best Privileged Access Management Reviews 2025"

[3] Akamai, "What Is Microsegmentation or Micro-Segmentation?"

[4] CyberArk, "What is Privileged Access Management (PAM)? - Definition"

[5] Net & End, "HIWARE Privileged Session Management for System"

[6] Arctera, "Arctera™ System Recovery 24 User's Guide"

EQST INSIGHT

2025.08

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