클라우드 보안 실시간 모니터링 시스템

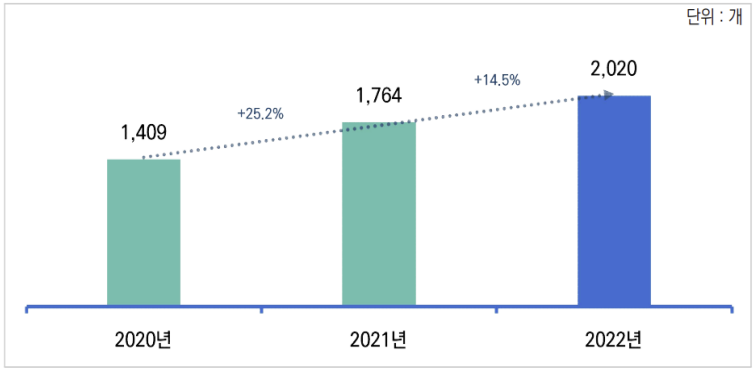
담당 교수 양환석 교수님
팀 장 하승범
팀 원 김다빈
김찬욱
장진호
고예진

CONTENTS

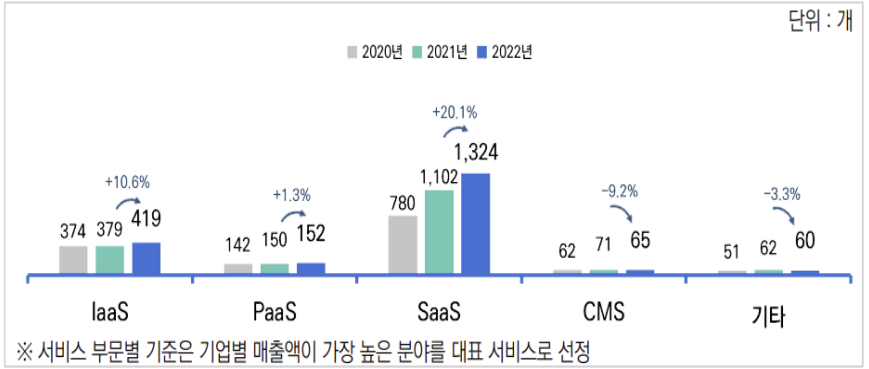
01

프로젝트 개요

A 프로젝트 선정 이유



클라우드 서비스 공급기업 수 3개년 추이



클라우드 서비스 부문별 공급기업 수 3개년 추이

클라우드는 현재 여러 방면에서 활용도가 높고 많이 사용되는 시스템 중 하나로 자리 잡혀 이와 관련된 보안 이슈 또한 늘어나고 있다. 그중 제로데이 공격에 대한 방어뿐만 아니라 빠른 대처 또한 중요하다고 판단하여 관계자에게 직접적으로 빠른 대처를 할 수 있는 프로그램의 필요성을 느꼈다.

출처 2023년 클라우드산업 실태조사 결과보고서

B 프로젝트 주제 및 목적

클라우드 보안 실시간 모니터링 시스템

클라우드에서 발생하는 보안 이슈(공격 탐지)를 즉각적으로 대처하기 위해 Python GUI와 보안 보고서를 통해 확인할 수 있다.

Openstack과 Ubuntu를 이용하여 클라우드 환경을 구축하고 Bash Shell을 이용하여 공격에 대해 탐지를 하면 GUI에 전송하여 실시간으로 확인이 가능하다.

C 팀원 소개 및 역할 분담

하승범	김찬욱	고예진	김다빈	장진호
91813248	91812218	92014954	92103561	92113839
프로젝트 총괄 DB, PW Shell Backdoor Shell API 연결 발표	Hash Shell 총괄 GUI 개발 PPT 제작	Network Shell	Network Shell GUI 개발 PPT 제작	Network Shell 총괄 모의해킹 Backdoor Shell

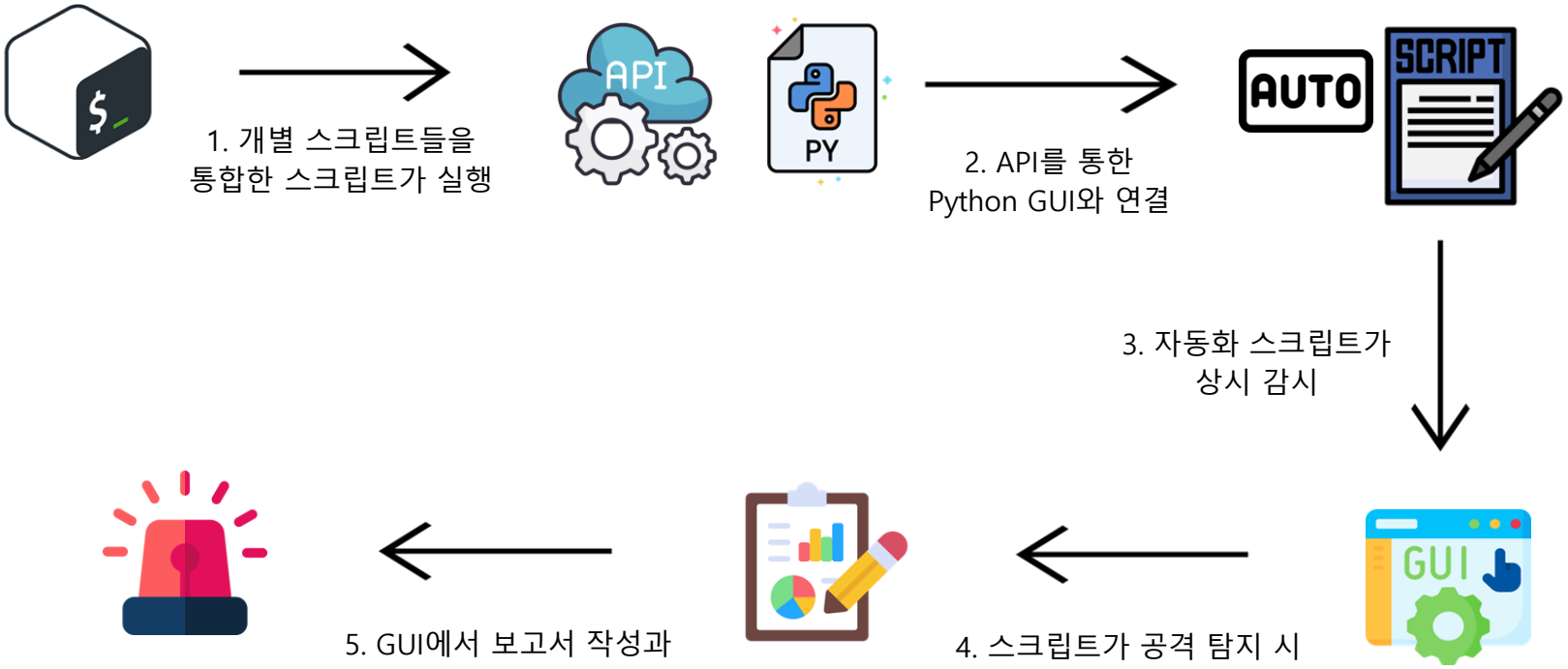
D 프로젝트 진행 일정

	3월	4월	5월	6월	7월	8월	9월	10월
팀 구성 및 주제 선정								
자료 조사								
기본 설계								
제작 및 구현								
테스트 및 오류 수정								
보고서 작성								

02

프로젝트 개발

A 프로젝트 구상도


1. 개별 스크립트들을
통합한 스크립트가 실행
API
PY
2. API를 통한
Python GUI와 연결
AUTO
SCRIPT
3. 자동화 스크립트가
상시 감시
GUI
4. 스크립트가 공격 탐지 시
GUI로 공격 탐지 데이터 전송
5. GUI에서 보고서 작성과
동시에 공격 탐지 알림

B 프로젝트 개발 환경

사용 환경

ubuntu

openstack

GUI

python

Visual Studio Code

API

Flask

진단 스크립트

BASH & Shell Scripts

C 프로젝트 구성

한국융합보안학회 2024년 하계학술대회 논문집

클라우드 보안 위협에 대한 연구 동향 분석

하승범[1]*, 김찬욱[2], 장진호[3], 김민수[4]
중부대학교[1,2,3,4]

Analysis of research trends on cloud security threats

Ha Seung Bum[1]*, Kim Chan Wook[2], Jang Jin Ho[3], Kim Min Su[4]
Jungbu University, Department of Information Protection[1,2,3,4]

요약 : 본 연구에서는 현재 대부분의 기업과 국가에서 운영하는 Cloud가 가지고 있는 보안 위협에 관해서 설명하고 최근 있었던 Cloud를 대상으로 하는 공격들을 분석하여 앞으로의 대응 방안에 관해서 연구하고자 한다. 관련 연구로써 CSA(Cloud Security Alliance)에서 제공한 데이터를 기반으로 클라우드 보안 위협에 최근 동향을 설명한다. 이런 사례들을 종합하여 클라우드 서비스라다 발생했던 공격들을 정리해서 클라우드 최신 공격 매트릭스를 제작하고 표를 제작하는 과정에서 유난히 자주 발생했던 공격들과 취약점을 파악하고 보완할 수 있는 최신 연구를 제안하고자 한다.

Key Words : Cloud / 취약점

3.1 클라우드 최신 공격 매트릭스

DDoS의 경우 지금까지도 발전하고 많이 발생하는 공격이다. 하지만 다른 공격들에 대해서는 충분히 대처할 여지가 있었다. DDos 공격을 제외한 사례에서는 저속 DDoS로 인한 피해가 발생했다.

저속 DDoS 공격은 은밀하고 낮은 속도의 트래픽으로 인해서 탐지하기가 어렵다.[3] 이런 저속 DDoS에 대해서도 연구가 이루어지고 있지만 저속 DDoS에 공격 종류에 따라서 탐지 방법이 다르기 때문에 더욱 더 연구가 필요할 것으로 생각된다. [3]에서 저속 DDoS에 대한 여러 연구들이 설명되었다.

3.3 시스템 영역

시스템 영역 내에서 가장 눈에 띄게 많았던 공격은 이메일 공격 혹은 이메일 서버를 노렸던 공격들이다. 이메일은 아직도 보안이 취약한 부분이며 공격자들의 주 목표라는 것을 보여준다.

현재는 스피어 피싱을 통해 한 사람 또는 한 시스템을 향한 공격들이 많았었고 대부분 관리자의 실수들로 인해서 시스템에 공격을 허용한 것이 대부분이다. 나머지는 시스템 내 자체 취약점이 존재했었던 것들이었다. 그리고 이런 공격들에 대해서 현재 머신러닝을 이용한 탐지와 관련된 연구들이 진행되고 있다.

2024년 1월에 ICIDCA 2024에서 피싱 메일에 공격 유형이 변화함에 따라 지금의 머신러닝 딥러닝, AI 모델을 이용한 피싱 메일 탐지가 많이 무용지물이 되었기에 앙상블 기법을 이용한 새로운 모델을 제안했다.[4] 이런 앙상블 기법으로 머신러닝을 이용하는 것으로도 피싱뿐만 아니라 다른 취약점이 존재하는 시스템 내에서 공격을 탐지하기 더 효율적일 것으로 생각한다.

3.4 데이터 영역

데이터 영역은 APT 공격, 중간자 공격들로 데이터에 해손, 탈취 등 데이터에 대한 공격들이 메인이다. 대부분의 공격은 사이트의 이용자를 즉 고객들의 데이

	Network			System				Data	
	Discovery	Command and Control (C2)	Reconnaissance	Initial Access	Execution	Privilege Escalation	Credential Access	Collection	Defense Evasion
IaaS	Cloud Infrastructure Discovery Cloud Storage Object Discovery	WS-Discovery Flood Attack SSDP DDoS Attack IP spoofing	Cross-cloud attack Orchestration attack	Exploit Public-Facing Application Hardware Additions	Cloud Administration on Command Remote Code Execution (RCE)	Exploit Misconfigured Identities in the Cloud Malware using Privilege Escalation		Data from Cloud Storage	Modify Cloud Compute Infrastructure
SaaS	Cloud Service Dashboard Cloud Service Discovery		Cross-tenant attack	Drive-by Compromise Email Phishing Attack Arp spoofing	DDoS		Indiscriminate college admissions Attack Email Phishing Attack	Emotet	
PaaS	Cloud Service Discovery Cloud Storage Object Discovery	Zombie Botnet	Cross-cloud attack Cryptojacking Cross-tenant attack Orchestration attack	Exploit Public-Facing Application Hardware Additions Drive-by Compromise Email Phishing Attack	Cloud Administration on Command Remote Code Execution (RCE) Remote code execution			Emotet APT Attack (Advanced Persistent Threat)	
FaaS		Zombie	Cryptojacking Serverless attack					Data from Cloud Storage APT Attack (Advanced Persistent Threat)	
비고	Cloud Storage Object Discovery (IaaS,PaaS) Cloud Service Discovery SaaS,PaaS)	Zombie (PaaS,FaaS)	Cross-cloud attack (IaaS,PaaS) Cryptojacking (PaaS,FaaS) Cross-tenant attack(PaaS, SaaS) Orchestration attack (IaaS,PaaS)	Exploit Public-Facing Application (IaaS,PaaS) Hardware Additions (IaaS,PaaS) Drive-by Compromise (SaaS,PaaS) Email Phishing Attack: (SaaS,PaaS)	Cloud Administration on Command (IaaS,PaaS) Remote Code Execution (RCE) (IaaS,PaaS) DDoS Remote code execution			Data from Cloud Storage (IaaS,FaaS) Emotet (SaaS,PaaS) APT Attack (Advanced Persistent Threat) (PaaS,FaaS)	

프로젝트 구성 (스크립트)

```bash
# 초기 해시값 저장 함수
store_initial_hash() {
    local FILE="$1"
    local HASH=$(sha256sum "$FILE" | awk '{print $1}')
    echo "$(basename "$FILE"):$HASH" >> "$HASH_STORE_FILE"
}

# 해시값을 딕셔너리에서 로드
declare -A initial_hashes
if [ -f "$HASH_STORE_FILE" ]; then
    while IFS=":" read -r filename hash; do
        initial_hashes["$filename"]="$hash"
    done < "$HASH_STORE_FILE"
fi
```

```bash
# 초기 해시값과 비교
if [[ "${initial_hashes[$FILENAME]}" != "" && "${initial_hashes[$FILENAME]}" != "$
    STATUS="1"  # 해시값이 변조된 경우 1 (취약)으로 설정
fi

# 초기 해시값이 없으면 저장
if [[ "${initial_hashes[$FILENAME]}" == "" ]]; then
    initial_hashes["$FILENAME"]="$HASH"
    store_initial_hash "$FILE"
fi

# 해시값 출력
echo "해시값: $HASH"
```

```bash
#!/bin/bash

# 데이터 전송 함수
send_data() {
    url="http://192.168.55.103:5000/receive-data"
    data=$(jq -n --arg id "PW" --arg status "$1" '{id: $id, status: $status}')
    echo "Sending data: $data to $url"
    curl -X POST -H "Content-Type: application/json" -d "$data" "$url" -v
}

# 파일 변경 감지 및 상태 전송
echo "Watching for changes in /etc/shadow..."
while true; do
    # 파일 수정 감지
    inotifywait -e modify /etc/shadow

    # 파일 수정이 감지되면 상태 1을 전송
    send_data 1

    # 일정 시간 대기 후 상태 0을 전송
    sleep 10
    send_data 0
done
```

각 취약점 진단별 스크립트 구성 → (Hash / PW)

D 프로젝트 구성 (GUI)

```python
# 보고서 열기 함수
def open_report(attack_type, data_to_add):
    # 보고서 생성
    create_report(attack_type, data_to_add)

    # 보고서가 열렸으므로 상태를 초록불로 변경
    filename = f"{attack_type}_report.pdf"
    if sys.platform == "win32":
        os.startfile(filename)
    else:
        subprocess.call(["open", filename])

    # 보고서 확인 후 상태를 초록불로 변경
    report_checked[attack_type] = True  # 보고서 확인 플래그
    locked_status[attack_type] = False  # 빨간불 고정 해제
    update_status(attack_type, False)  # 상태를 초록불로 변경
    delete_uploads_folder()
```

```python
# 상태 표시기를 업데이트하는 함수
def update_status(script_id, status):
    # 보고서가 확인되기 전까지 상태를 락함
    if locked_status[script_id]:  # 빨간불 고정 상태인 경우
        status_indicators[script_id].itemconfig("circle", fill="red")
        report_buttons[script_id].config(state=tk.NORMAL)
        return
```

진단 결과 보고서 PDF 자동 생성 → 진단 결과 상태 표시 업데이트

D 프로젝트 구성 (GUI)

Hash 보고서

제작자 : 구름

구분	상세 내용
생성 일시	2024-09-27 13:03:58
예상 취약점	41e13d0d-3935-4f08-a498-83289bfb3c62 969c3aa4-b107-40bf-ba1a-dbccf175f3e2 b04e886f-af01-421f-8266-1971dfdbd79a fcc22902-55f8-4fe1-8418-9457d08533fd 1. 클라우드 업로드 데이터중 해시값이 변경됨 클라우드에 업로드하는 데이터의 해시값이 변경될 수 있는 잠재적 취약점이 존재합니다. 이 경우, 파일의 무결성이 의심받게 되며, 해시값이 초기 저장값과 다르다는 것은 파일이 수정되었음을 의미합니다. 해시값 변경의 원인은 여러 가지가 있을 수 있습니다. 예를 들어, 파일이 업로드되는 과정에서 손상이 발생하거나, 시스템 오류로 인해 데이터가 제대로 저장되지 않을 수 있습니다. 2. 중간자 공격 클라우드 업로드 과정에서 중간자 공격이 발생할 수 있습니다. 이 공격은 공격자가 클라이언트와 서버 간의 데이터 전송 경로에 개입하여 데이터를 가로채거나 수정하는 방식으로 진행됩니다. 예를 들어, 사용자가 클라우드에 파일을 업로드하는 동안, 공격자는 네트워크를 통해 데이터 패킷을 가로채고 파일을 변경한 뒤 클라우드 서버로 전송할 수 있습니다. 3. 악성 코드 삽입 바이러스나 웜과 같은 악성 코드가 클라우드 업로드 과정에서 추가될 가능성도 있습니다. 사용자가 업로드하는 파일에 대한 무결성을 검사하는 절차가 부족할 경우, 악성 코드가 포함된 파일이 클라우드에 저장될 수 있습니다.

GUI 초기 상태 → 스크립트 실행 → 취약점 발견 후 GUI 모습 및 보고서

03

프로젝트 결론

A 프로젝트 시연 영상

A 프로젝트 시연 영상

B 프로젝트 결과

본 프로젝트는 클라우드 환경에서 발생할 수 있는 다양한 보안 위협에 실시간으로 대처할 수 있는 **모니터링 시스템**을 개발하는 것을 목표로 하였다. Python GUI와 보안 보고서를 통해 **공격 탐지 및 즉각적인 알림**이 가능하게 하였으며, 이를 통해 클라우드 관리자들이 보안 상황을 빠르게 인지하고 대응할 수 있는 환경을 제공할 수 있었다.

또한 OpenStack과 Ubuntu를 기반으로 클라우드 환경을 구축하고, Bash Shell을 활용한 공격 탐지 기능을 통해 보안 문제를 실시간으로 확인할 수 있었다.

본 프로젝트를 통해 클라우드 보안 모니터링의 중요성을 다시 한 번 확인할 수 있었으며, 향후 보안 관련 기술 발전에도 기여할 수 있는 가능성을 보여주었다.

C 프로젝트 기대 효과

관리자가 실시간으로 확인

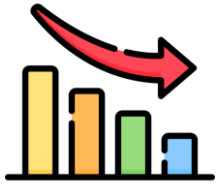
클라우드 환경에서 발생하는 보안 이슈를 즉각적으로 탐지하고 관리자에게 실시간으로 알림을 제공하여 신속한 대응이 가능하다.

확장성

본 시스템은 리눅스 환경에서의 클라우드를 대상으로는 모두 사용 가능하기에, 추후 통합적인 보안 관리가 가능하다.

공격 피해 최소화

잠재적인 보안 위협을 조기에 탐지하고 대응함으로써, 시스템 가동 중단이나 데이터 손실과 같은 피해를 최소화할 수 있다.



대처할 수 있는 시간 증가

보안 위협에 대한 빠른 탐지와 보고를 통해 관리자가 사전에 대처할 수 있는 시간을 확보하여 공격 피해를 줄일 수 있다.

THANK YOU

Q & A