클라우드 보안 실시간 모니터링 시스템

팀	명	구 름
지 도 교 수		양환석 교수님
팀	장	하승범 91813248
팀	원	김찬욱 91812218
		김다빈 92103561
		장진호 92113839
		고예진 92014954

2024. 11.

중부대학교
정보보호학과

목 차

1. 서 론

1.1 연구 배경

클라우드 컴퓨팅 기술은 다양한 산업에서 널리 사용되며, 효율성과 비용 절감 측면에서 큰 이점을 제공하고 있다. 그러나 클라우드 환경에서 발생하는 보안 문제, 특히 제로데이 공격과 같은 새로운 위협에 대응하는 것이 점차 중요해지고 있다. 클라우드 서비스 제공업체들은 보안 강화를 위해 노력하고 있지만, 여전히 실시간 탐지 및 대응을 강화할 필요성이 든다. 이에 따라, 본 연구에서는 클라우드 환경에서 실시간으로 보안 위협을 탐지하고 관리자가 신속하게 대응할 수 있는 모니터링 시스템을 개발하고자 한다.

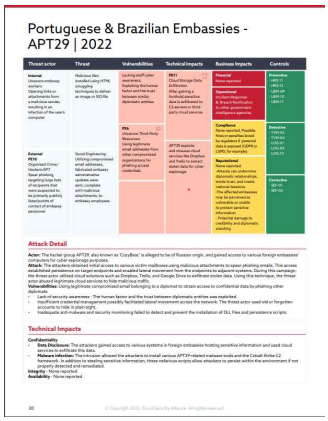
1.2 연구 필요성

클라우드 서비스는 기업과 개인에게 필수적인 인프라로 자리 잡았지만, 보안 위협에 대한 대응은 여전히 중요한 과제이다. 특히 실시간으로 보안 위협을 탐지하고 즉각적으로 대응할 수 있는 시스템이 없다면, 클라우드 환경에서 발생하는 보안 사고를 빠르게 차단하기 어려울 수 있다. 기존의 보안 시스템은 주로 사후 대응에 초점을 맞추고 있어, 사전 예방 및 신속한 대응이 필요한 상황이다. 따라서 실시간 모니터링 시스템을 통해 이러한 문제를 해결할 필요성이 절실한 상황이다.

1.3 연구 목적 및 주제선정

본 연구의 목적은 클라우드 환경에서 발생할 수 있는 다양한 보안 위협을 실시간으로 탐지하고, 관리자가 이를 즉각적으로 대처할 수 있도록 돕는 시스템을 구축하는 것이다. Python을 활용한 GUI 시스템과 Bash Shell을 이용하여, 보안 위협이 탐지될 때 이를 실시간으로 알리고, 자동 보고서를 생성하는 기능을 포함하는 모니터링 시스템을 구현하는 것을 목표로 한다. 이를 통해 클라우드 보안에

대한 대응 시간을 단축시키고, 보안 사고로 인한 피해를 최소화할 수 있다. 현재 클라우드는 여러 회사와 나라들의 보고서를 보면 공격의 빈도수와 위험성이 크게 증가하는 것을 알 수 있다. 하지만 보안이 방어에 성공하는 것뿐만 아니라 피해를 최소화 하고 공격을 늦추는것 또한 중요하기에 이 주제를 선택하였다.



위 그림은 프로그램을 제작하는데 참고한 보고서 중 일부이다.

2. 관련연구

2.1 클라우드 보안 탐지 시스템

클라우드 보안 탐지 시스템에 대한 연구는 꾸준히 발전해 왔다. 클라우드 환경에서의 공격을 실시간으로 탐지하기 위한 연구는 인공지능과 머신러닝 기반의 침입 탐지 시스템(IDS)을 통해 비정상적인 패턴을 분석하는 데 중점을 두고 있다. Sharma 등(2017)은 클라우드 기반 IDS 시스템을 제안하여, 실시간으로 네트워크 트래픽을 분석하고 보안 위협을 탐지하는 방법을 제시하였다. 이러한 시스템은 보안 위협에 대한 빠른 탐지와 대응을 가능하게 해준다.

2.2 자동화된 보안 모니터링 시스템

자동화된 보안 모니터링 시스템은 보안 담당자의 개입 없이 시스템 자체가 보안 위협을 탐지하고 대응하는 데 큰 역할을 한다. Zhang 등(2020)은 클라우드 환경에서 보안 위협을 자동으로 탐지하고 보고서를 생성하는 시스템을 개발하였다. 이 시스템은 보안 위협이 탐지될 경우 즉각적인 알림과 보고서를 제공하여, 관리자가 신속하게 대응할 수 있도록 설계되었다. 이러한 자동화 시스템은 보안 관리

의 효율성을 높이고, 보안 사고의 영향을 최소화할 수 있다.

2.3 OpenStack 기반 보안 연구

OpenStack은 많은 클라우드 환경에서 사용되는 오픈 소스 플랫폼으로, 보안 연구의 주요 대상이 되고 있다. OpenStack 환경에서의 보안 취약점을 실시간으로 탐지하는 연구가 활발히 진행되고 있으며, Hwang 등(2021)은 OpenStack 기반 클라우드 환경에서 발생하는 보안 위협을 실시간으로 탐지하는 시스템을 제안하였다. 이 시스템은 네트워크 트래픽을 분석하여 비정상적인 활동을 감지하고, 관리자에게 경고를 제공함으로써 보안 위협에 대한 신속한 대응을 가능하게 한다.

2.4 Python

Python은 쉽고 직관적인 구문을 제공하는 고수준 프로그래밍 언어로, 다양한 라이브러리와 프레임워크를 통해 데이터 처리, 자동화, 웹 개발, 그리고 보안 시스템 개발 등 여러 분야에서 폭넓게 사용되고 있다. 본 프로젝트에서는 Python을 이용하여 GUI(그래픽 사용자 인터페이스) 및 보안 위협 탐지를 위한 스크립트를 작성하였다. Python의 다양한 라이브러리를 활용하여 클라우드 환경에서 발생하는 보안 위협을 실시간으로 탐지하고, 데이터를 시각화하여 관리자가 즉각적인 대응을 할 수 있도록 구현하였다.

2.5 Bash Shell

Bash Shell은 유닉스 기반 운영 체제에서 사용되는 명령어 인터프리터로, 본 프로젝트에서는 클라우드 서버의 취약점을 실시간으로 분석하고 탐지하는 데 사용되었다. Bash Shell 스크립트는 OpenStack 환경에서 서버의 네트워크 활동과 파일 시스템을 감시하며, 비정상적인 활동을 탐지했을 때 Python GUI로 데이터를 전송하여 관리자가 이를 확인할 수 있도록 구현되었다

2.6 OpenStack

OpenStack은 오픈 소스 클라우드 플랫폼으로, 클라우드 인프라스트럭처에서 리소스를 관리하고 제어하는 데 사용된다. 본 프로젝트에서는 OpenStack을 사용하여 클라우드 환경을 구축하였고, 보안 모니터링 시스템의 테스트 환경으로 활용하였다. OpenStack을 통해 가상 서버와 네트워크를 설정하고, 클라우드 기반의 보안 위협을 탐지하는 Bash Shell 스크립트를 실행하는 데 필요한 인프라를 제공하였다.

2.7 Ubuntu

Ubuntu는 오픈 소스 리눅스 기반 운영 체제로, 안정성과 보안이 뛰어난 특성으로 인해 서버 환경에서 널리 사용되고 있다. 본 프로젝트에서는 Ubuntu 운영체제를 기반으로 클라우드 보안 모니터링 시스템을 구축하고, Bash Shell 스크립트를 실행하여 실시간으로 보안 위협을 탐지하고 보고하는 역할을 수행하였다. Ubuntu의 오픈 소스 특성을 활용하여 다양한 보안 도구와 연동이 가능하며, 본 시스템의 핵심 환경으로 사용되었다.

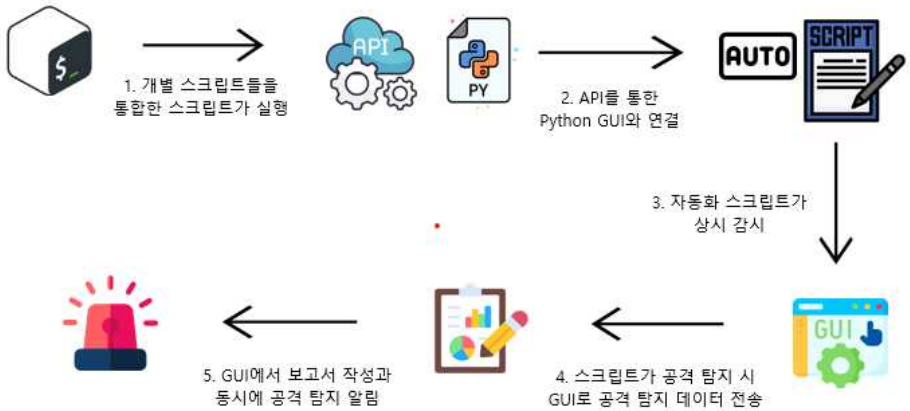
2.8 GUI(Graphical User Interface)

본 프로젝트에서는 Python의 Tkinter 라이브러리를 사용하여 GUI(그래픽 사용자 인터페이스)를 구현하였다. GUI는 보안 위협을 실시간으로 시각화하고, 관리자에게 직관적으로 정보를 제공하는 역할을 합니다. 탐지된 보안 위협은 GUI를 통해 실시간으로 알림이 제공되며, 이를 바탕으로 보안 보고서를 자동으로 생성하는 기능을 포함하고 있다.

3. 본 론

3.1 시스템 구성

클라우드 보안 실시간 모니터링 시스템은 Bash shell로 만들어진 리눅스에서 작동이 가능한 탐지 자동화 shell들이 모니터링하는 컴퓨터에서 실행 중인 Python GUI로 Flask API를 이용해 실시간으로 공격을 받거나 시스템에 이상이 없



는지 GUI 화면을 통해서 사용자가 판단할 수 있는 프로그램으로 구현되었다.

3.2 자동화 스크립트

자동화 스크립트들은 논문에 근거한 공격 표에 따라서 최근 클라우드 환경과 클라우드를 대상으로 했던 공격을 탐지할 수 있게 구성하였다. VMware에 Ubuntu(22.04.4)와 Openstack을 설치하고 이 환경을 기준으로 스크립트를 제작하였다. 스크립트는 실시간으로 공격을 계속 탐지하도록 백그라운드에서 돌아가며 공격을 받을 시 1값을 아닐시 0값을 GUI로 보내도록 설계했다. Hash 스크립트에 경우 어느 Hash 값이 변경됐는지도 XML 파일에 같이 보내서 더욱 빠른 대응을

```
# 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"
```

할 수 있게 하였다. 네트워크는 DDos 2가지 방법으로 탐지와 ARP테이블, port를 체크하고 프로세스 백도어, 파일 백도어, 루트 비밀번호 변경, 클라우드 내 이미지의 Hash 값 변경이 8가지 스크립트로 구성돼 있다.

```
# 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
```

3.3 GUI

GUI는 각각의 공격을 탐지하는 스크립트들이 Flask API를 통해 공격 값이 포

```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
```

함된 데이터 혹은 정상값이 포함된 데이터를 보내면 밑에 알림 등이 정상이라면 초록색 공격을 받고 있다면 빨간색으로 바뀐다. 동시에 공격 값이 포함된 데이터를 받으면 보고서 버튼이 활성화되면서 보고서를 볼 수 있는데 자동화 스크립트들이 현재 어떤 방식으로 탐지하고 있고 어떻게 대응해야 하는지 대응 방안과 예상 취약점을 알려준다.

Hash 보고서

제작자 : 구름

구분	상세 내용
생성 일시	2024-09-27 13:03:58
예상 취약점	41e13d0d-3935-4808-a498-83289bfb3c62 969c3aa4-b107-40bf-ba1a-dbccf175f5e2 b04e8864-af01-421f-8266-1971dfdbd75a fcc22902-5588-4fe1-8418-9457d08533fd 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.

4. 분 석

4.1 활용 결과 및 성능

클라우드 보안 실시간 모니터링 시스템을 개발해서 Kali Linux와 클라우드 공격 쉘을 이용해서 테스트해 본 결과 탐지를 이상 없이 성공, GUI와 스크립트 사이에 통신도 잘 되는 것을 확인, GUI도 데이터값에 따라 정확하고 빠르게 UI로 공격 상황 여부를 보여주고 공격을 당했을 때 보고서 또한 잘 출력되었다. 이를 통해 충분히 혼자서도 공격을 받는 상황에서 보고서와 UI를 통해서 어느 공격을 받고 있고 어떻게 대처해야 하는지 판단하는 데 있어서 충분할 것으로 파악했다.

4.2 추후 보완사항

현재 GUI의 경우 보고서를 따로 다운받는 기능은 없어서 pdf를 열 때 수동으로 다운 받아야 한다. 그렇기에 공격이 받으면 로그를 따로 남기고 전에 보고서를 다시 볼 수 있는 기능의 추가와 현재 초기에 구상했었던 스크립트들이 전부 제작된 것은 아니기에 다른 탐지 shell 등을 추가하고 Openstack 환경뿐만 아니라 범용적으로 사용되는 AWS 환경에서도 사용이 가능한 스크립트들이 제작되고 앞으로도 클라우드를 향한 공격들과 동향들을 연구해서 기능을 업그레이드하는 것을 계획 중이다.

5. 결 론

5.1 결 론

최근 클라우드를 보편화되고 많은 사람들이 이용하면서 클라우드가 발전한 만큼 클라우드를 대상으로 하는 공격들 또한 발전하고 빈도수도 늘어났다. 그렇기에 클라우드를 대상으로 하는 실시간 보안 모니터링 프로그램도 발전해야 한다. 현재 Openstack 환경에서만 테스트를 해봤지만, 스크립트들이 정상적으로 공격을 탐지하고 GUI에서 확인하고 보고서를 출력하는 것을 확인했다. 여러 업데이트가 더 필요하지만, 이런 탐지 프로그램만으로도 더 빠른 대응을 통해 공격을 막는 것은 아니지만 피해를 줄일 수 있다고 생각하고 이런 탐지 시스템과 프로그램들이 계속해서 발전할 것이라고 여겨진다.

5.2 기대효과

일반적으로 보완하면 방어에 성공해야만 보안이라고 생각하는 경우가 많다. 하지만 요즘 제로데이 공격이 강력한 취약점으로 골치를 썩이고 있는 만큼 탐지를 통해 공격을 빨리 감지하고 빠르게 대응하는 것 또한 중요해졌다. 그렇기에 클라우드 보안 실시간 모니터링 프로그램은 앞으로의 탐지 시스템이 발전해야 하는 예시가 되어줄 것이고 보안이 더욱 견고해지는 것을 기대하고 있다.

6. 별 첨

6.1 팀원 소개

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

6.2 소스 코드

깃허브 주소 : https://github.com/Rudyclo/cloud-security [하승범]

6.3 시연 영상 QR코드 와 링크

https://www.youtube.com/watch?v=qGgeRVHQ69k

6.4 발표 자료

(별첨)

6.5 소개 자료

CLOUD

2024
November 4th

"클라우드 보안 실시간 모니터링 시스템"

개요

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.

구상도

Python GUI

팀 장	하 승 범 | 프로젝트 총괄 DB, PW Shell API 연결 발표	팀 원	김 찬 욱 | Hash Shell 총괄, GUI 개발, PPT 제작
			김 다 빈 | Network Shell, GUI 개발, PPT 제작
			장 진 호 | Network Shell 총괄, 모의해킹
담당 교수	양 환 석 교수님		고 예 진 | Network Shell