119 회 기출풀이

컴퓨터시스템응용기술사

- KPC 기술사회 -

SMART KPC
한국생산성본부

교육 문의 및 상담 : 한 승 연

- Tel : 02) 724-1831/1223
- Fax : 02) 724-1875
- Email : syhan@kpc.or.kr
- Web Site : www.kpc.or.kr
cafe.naver.com/81th

kpc 기술사회

120 회 합격대비 심화반 신청 안내

[토요일 명품심화반]

- O 단합반(SPP 반) (안경환 PE @ KPC) : 정규심화 9. 28. 개강
- O FB(Future Builders) (강희석 PE @ KPC) : 정규심화 9. 28. 개강
- O 열 정 반 (박상욱 PE @ KPC) : 정규심화 9. 28. 개강
- O 정 주 행 (서정훈 PE @ KPC) : 정규심화 9. 28. 개강
- O ITPE Makers (박제일 PE @ KPC) : 정규심화 9. 28. 개강
- O MP 필통반 (구환회 PE @ KPC) : 정규심화 9. 28. 개강
- O 공 감 반 (공수재 PE @ KPC) : 조기심화 8. 31. 개강

[일요일 명품심화반]

- O T.O.P 반 (유술사 PE @ KPC) : 조기심화 8. 25. 개강
- O NS 반 (강정배 PE&박주형 PE @ 강남아지트): 조기심화 9. 1. 개강

[평일 명품심화반]

- O 강남평일야간반 (강정배 PE&전일 PE&박찬렬 PE @ 강남아지트/화,금):
  조기심화 9. 3. 개강

~~ KPC 홈페이지에서 신청 가능합니다. ~~

국가기술자격 기술사 시험문제

기술사 제 119 회 제 2 교시 (시험시간: 100 분)

분야	정보통신	종목	컴퓨터시스템응용기술사	수험번호		성명	

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 25 점)

1. OpenStack 의 특징과 구성도 및 서비스에 대하여 설명하시오.

2. VPN(Virtual Private Network)을 구현 방식과 서비스 형태에 따라 비교하여 설명하고, SSL VPN 방식에 대하여 설명하시오.

3. 4 차 산업혁명을 주도하고 있는 인공지능 머신러닝 기술은 실제로 적대적 공격(Adversarial Attack)에 취약한 것으로 알려져 있다. 다음에 대하여 설명하시오.

   1) 자율주행자동차에 대한 적대적 공격

   2) 적대적 공격을 위한 적대적 샘플(Adversarial Sample) 제작기법

   3) 적대적 공격에 대한 방어기법

4. DNS 를 은닉채널(Convert Channel)로 사용하는 이유를 설명하고, DNS Convert Channel 공격모델 및 방어기법을 설명하시오.

5. TTA 기반으로 정보시스템의 H/W 용량을 산정하고자 한다. 다음에 대하여 설명하시오.

   1) H/W 규모산정 방법에 대한 개념 및 장·단점

   2) 규모산정 대상

   3) CPU 및 스토리지의 성능 기준치

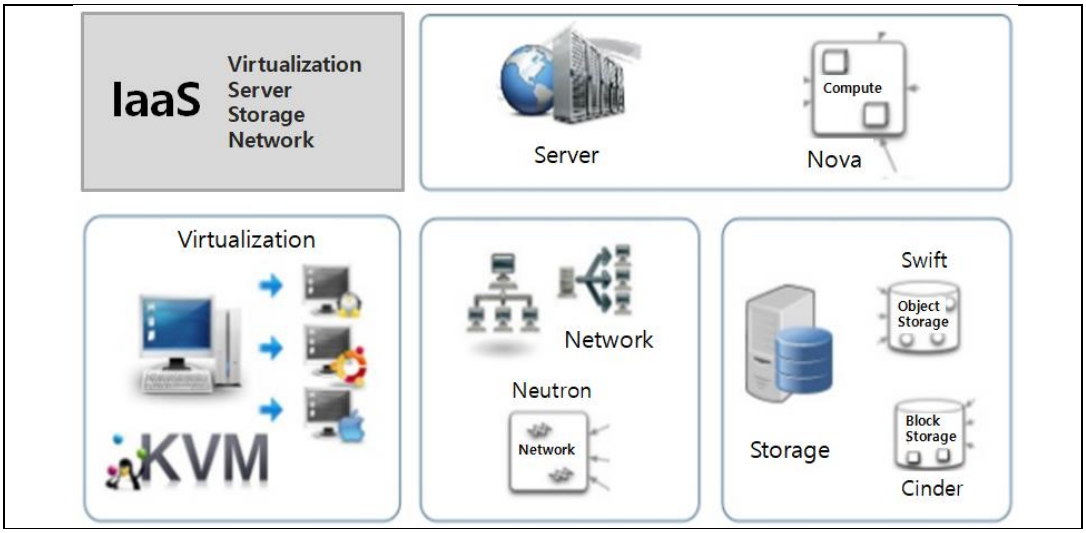
6. 클라우드 시스템 구축을 위한 핵심 기술인 가상화 관련 기술 중 가상머신과 컨테이너를 비교하여 설명하시오.

Notes

1	OpenStack
문제	OpenStack 의 특징과 구성도 및 서비스에 대하여 설명하시오.
도메인	디지털 서비스 (클라우드)
정의	클라우드 인프라에 필요한 서버 가상화, 스토리지 가상화, 네트워크 가상화 기술들을 종합적으로 구현한 IaaS 형태의 클라우드 컴퓨팅 오픈 소스 프로젝트
키워드	Nova, Swift, Glance, Horizon, Neutron
출제의도분석	기업의 클라우드와 오픈소스 활성화에 따라 Openstack 을 통한 클라우드 플랫폼의 설계 및 서비스 구성 전략 확인
답안작성 전략	1) 오픈스택의 특징과 구성도를 명확하게 작성 2) 오픈스택의 모든 서비스를 작성하기 보다는 주요 서비스 중심 서술
참고문헌	Etsuji Nakai 씨의 "OpenStack: Inside Out" 한글 번역본(Jaehwa Park 수석) 오픈테크넷 서밋 2014 발표자료(Nalee Jang, Platform Engineer of Red Hat Korea)
풀이 기술사님	이술사 PE(정보관리 기술사 / onlyitpe@gmail.com)

1. IaaS 구축 플랫폼, OpenStack 의 개념과 특징

가. Openstack 의 개념



- 클라우드 인프라에 필요한 서버 가상화, 스토리지 가상화, 네트워크 가상화 기술들을 종합적으로 구현한 IaaS 형태의 클라우드 컴퓨팅 오픈 소스 프로젝트

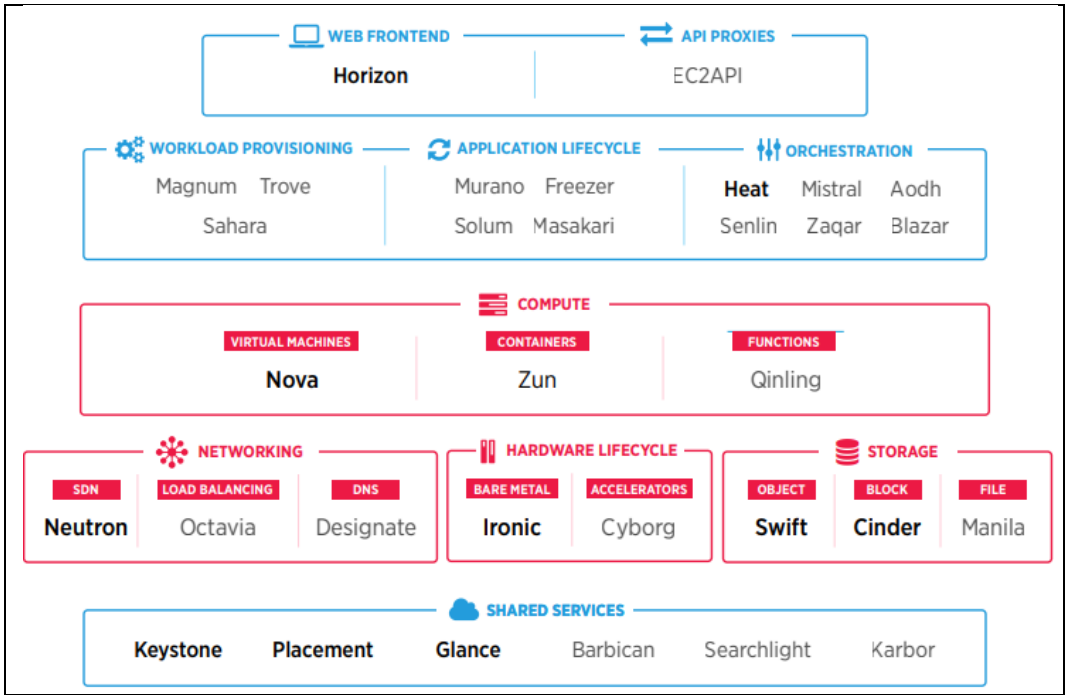
나. Openstack 의 특징

구분	상세 설명	세부기술
자원할당	-프라이빗 테넌트에서 컴퓨팅 자원을 생성하고 설정	웹 콘솔, REST API
자원소유	- 동일 프로젝트 사용자들은 공통 컴퓨팅 자원 공유 - 다른 프로젝트와 독립된 컴퓨팅 자원을 소유	보안그룹, 인증 토큰
관리	-자동배포와 Production 환경관리 위한 배포툴 활용	Ansible, Chef, Puppet, Salt

- Openstack 프로젝트는 간단한 구현, 대규모 확장, 다양한 기능의 지원으로 클라우드 환경에 대한 모든 타입을 지원하는 목표로 한다.

2. Openstack 의 구성도와 구성요소

가. Openstack 의 구성도



- 초기 오픈 스택은 Nova, Swift, Glance 3 개의 프로젝트에서 시작하여 Nava 의 Nova API, Nova Scheduler, Nova Volume, Nova Network 기능이 각 Cinder, Neutron 으로 격상됨
- 릴리즈 마다 새로운 프로젝트가 추가되어 기능을 보완하고, 보안요소를 추가하게 됨

나. OpenStack 의 구성요소

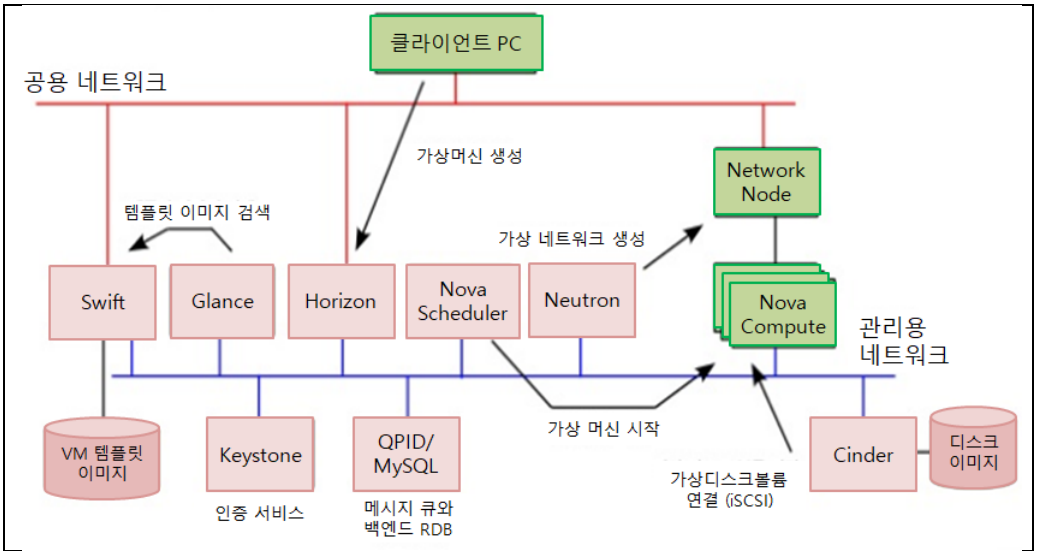
구성요소	역할	요소 설명
Nova	Compute	서버의 가상화를 지원, 가상 머신 라이프 사이클 관리
Swift	Object Storage	오브젝트 스토리지로 Rest API 를 통해 파일을 관리
Glance	Image	Nova 에서 서버를 가상화할 때 필요한 OS 이미지를 관리
Keystone	Identity	모든 프로젝트의 인증을 통합 관리, 중앙집중식 인증과 서비스 카탈로그 시스템
Neutron	Networking	SDN(Software Define Network)를 지원, 가상 네트워크 관리
Cinder	Block Storage	블락 디바이스 볼륨을 관리, 아마존의 Elastic Block Store
Trove	Database	MySQL, Redis 와 같은 데이터베이스를 관리
Horizon	Dashboard	웹 기반의 셀프서비스 포탈

- 각각의 프로젝트간의 통신인 Rest API 의 identity 를 통합하여 인증 모듈을 구현하고 각 프로젝트마다 가지는 고유한 Rest API URL 관리를 하는 Keystone 이 중요한 역할 함.

- Heat(Orchestration), Sanara(Data Processing), Ironic(Bare metal), Zaqau(Message), Barbican(Key management), Designate(DNS), Manila(Shared File System), Monasca(Monitoring)등 프로젝트 발전

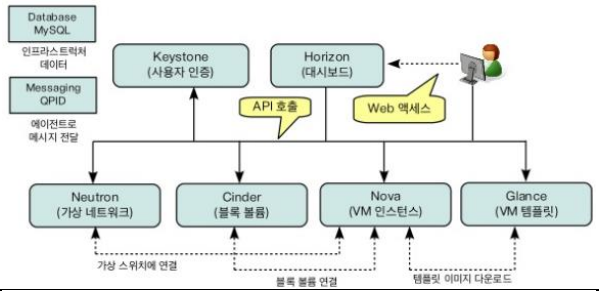
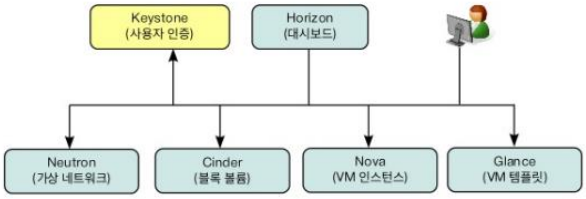
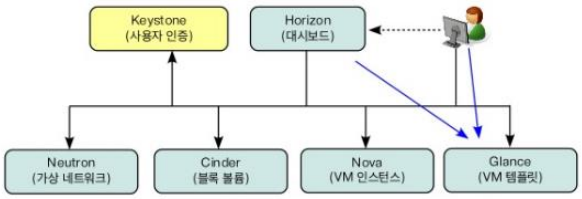
3. Openstack 의 서비스

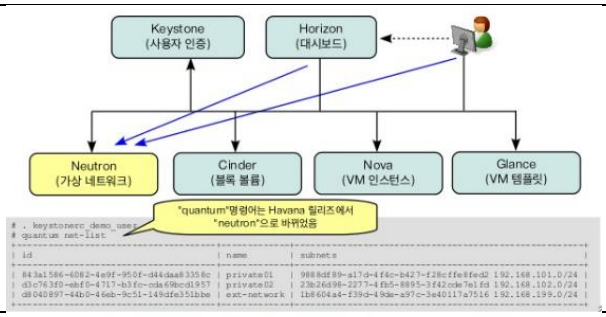
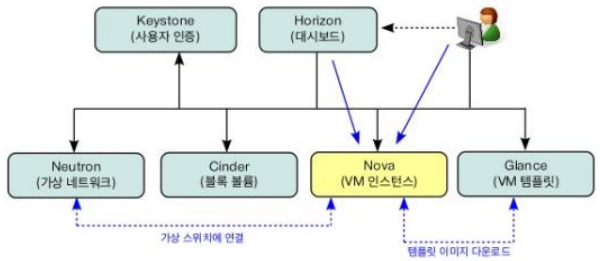
가. Openstack 의 서비스 흐름도



- 모듈들은 REST API 호출과 메시지 큐를 통해 동작한다.
- 조작은 REST API 를 통한 외부 프로그램으로 자동화 할 수 있다.

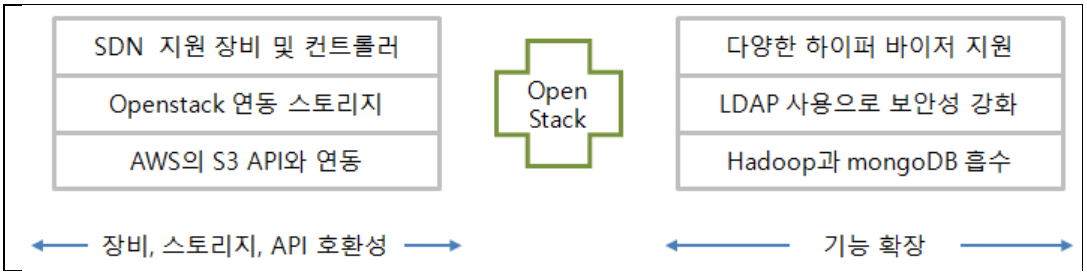
나. Openstack 의 서비스 상세

구분	상세 설명	세부기술
API 호출		- 사용자가 대시보드(혹은 직접) 요청 호출 - 일부 컴포넌트가 다른 컴포넌트에게 요청 호출
사용자 인증		- API 요청 보내기 전에 Key ston 으로부터 동작에 대한 토큰을 구하고, 이를 통해 대상 API 의 URL 획득
이미지 등록		- Glance 를 이용해 새로운 템플릿 이미지를 등록. - 등록된 이미지는 Nova 에서 사용

가상네트워크 운영		- Neutron API 를 통해 사용자는 자신의 테넌트 전용 가상 네트워크를 생성
VM 생성		- Nova 가 인스턴스 생성요청을 수신 하면, API 를 통해 Glance 및 Neutron 과 통신

- Neutron 을 이용해 IP 연결, Nava 를 통한 보안 그룹에 대한 작업, Cinder 를 이용한 블록 볼륨 생성 등의 작업을 수행하며 에이전트와 내부서비스는 메시징 서버를 통해 통신한다.

4. Openstack 사용의 확대

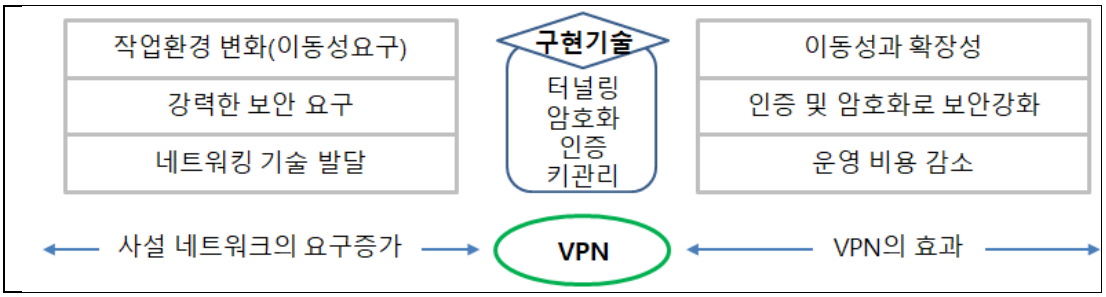


- Openstack 의 호환성과 기능 확장 으로 기업에서의 활용 사례가 증가되고 있으며, 계속되는 업데이트로 안정성이 향상되어 Private 클라우드 시스템은 물론 Public 클라우드 시스템으로 사용이 확대되고 있음.

"끝"

2	VPN(Virtual Private Network)
문제	VPN(Virtual Private Network)을 구현 방식과 서비스 형태에 따라 비교하여 설명하고, SSL VPN 방식에 대하여 설명하시오.
도메인	Network
정의	인터넷 등의 공중망(Public NW)에서 터널링(Tunneling) 기법을 사용하여 서버와 클라이언트간에 암호화 된 통신을 지원하는 전용회선(Private NW)을 구성한 가상 네트워크
키워드	공중망의 전용회선化, 터널링, 인증, 암호화, 키관리, LAN-to-LAN, LAN-to-Client, IPSEC VPN, MPLS VPN, VoVPN, MVPN SSL, 프로토콜(PPTP,L2TP,IPSec,SOCKET V5)
출제의도분석	네트워크 기술 발달 및 원격근로 확대와 IoT 장비 확산으로 터널링을 통한 암호화 통신의 요구가 증가함에 따라, VPN 의 방식과 서비스 구분 및 SSL VPN 에 대한 이해도를 확인하고자 출제.
답안작성 전략	빈출 되는 토픽이므로 정확한 서술 필요. VPN 의 구성요소, 기술요소 보다는 문제에 충실하여 답안을 구성하여 성의있게 작성하고, 가능하다면 SSL VPN 의 최근 동향 및 적용방안에 대한 실무적 서술.
참고문헌	https://ensxoddl.tistory.com/184 https://ict-story.tistory.com/53 ssl vpn 이란 무엇일까요?
풀이 기술사님	이술사 PE(정보관리 기술사 / onlyitpe@gmail.com)

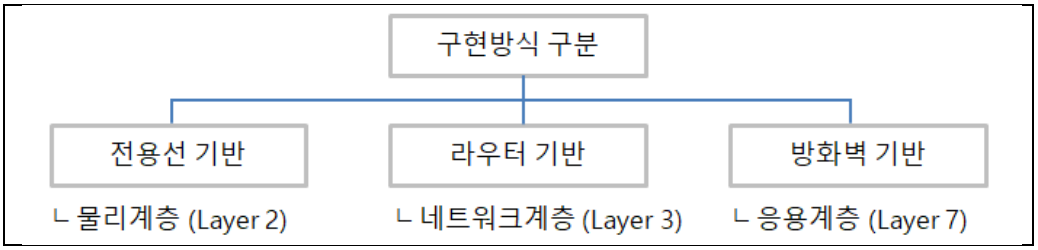
1. 전용 네트워크 구성 기술, VPN(Virtual Private Network)의 개념



- 인터넷 등의 공중망(Public NW)에서 터널링(Tunneling) 기법을 사용하여 서버와 클라이언트간에 암호화 된 통신을 지원하는 전용회선(Private NW)을 구성한 가상 네트워크

2. VPN 의 구현방식과 서비스 형태에 따른 비교

가. VPN 의 구현 방식에 따른 비교



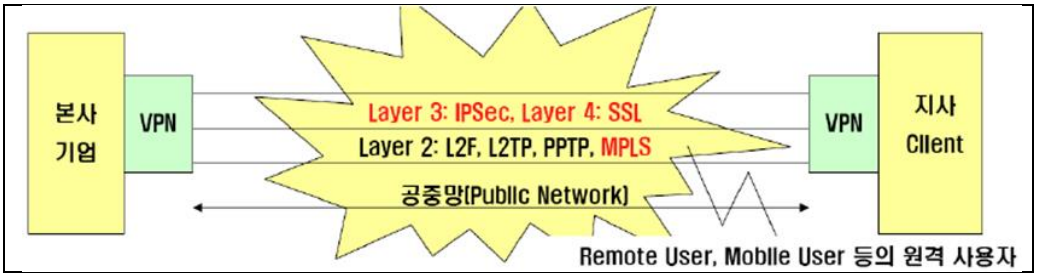
- 구현 방식에 따라서 전용시스템 방식, 라우터 방식, 방화벽 방식으로 분류

구분	전용선기반	라우터기반	방화벽기반
개념	LAN to LAN 구현,	전송경로상의	방화벽에 가상 사설망

	게이트웨이와 암호화장비가 결합한 형태	라우터가 VPN의 기능을 수행하는 형태	기능을 추가하여 구현
장점	기존 보안장비와의 호환성 불필요	전용장비에 비해 비용이 저렴	보안성이 가장 우수함
단점	설치 비용이 높음	방화벽 외부설치로 라우터와 방화벽 사이의 데이터 노출 위험	네트워크 성능저하
속도	빠름	중간	느림
가격	고가	중간	저가
구현	하드웨어	하드웨어	소프트웨어
적용대상	대규모 네트워크	소규모 네트워크	소규모 네트워크

- VPN 을 구현 방식 외에 서비스 형태에 따라 유형을 구분하는 것이 더 일반적임.

나. VPN 의 서비스 형태에 따른 비교



- 자사의 업무특성 파악 및 구성을 판단하며, Remote Access VPN 과 Lan to Lan VPN 중 서비스 형태를 결정하는데, 터널링 방식에 따라 L2TP, IPSec, SSL 등 여러 가지 서비스 유형이 있음.

표	MPLS	IPSec	SSL
개념	패킷 스위칭 기술인 MPLS 환경을 통해 VPN 구현	-IP 프로토콜의 일부인 IPSec 프로토콜을 이용하여 VPN 을 구현	보안통신 프로토콜(SSL)을 통해 VPN 구현
제공계층	Layer 2	Layer 3	Layer 4
보안	없음	IPSec	SSL
암호화	없음	패킷단위	메시지단위
성능	QoS 보장	다소 취약	취약
표준	RC2547	RFC2401	TLS
구성	Lan to Lan, Remote Access	Lan to Lan, Remote Access	Remote Access
활용	QoS 보장 App	C/S 기반 App	웹 기반 App
장점	뛰어난 제어 , 낮은 도입/ 관리비	보안수준/암호화우수 App 와 독립(투명성)	웹 브라우저로 VPN 구현 뛰어난 사용성
단점	통일 ISP 내부운영,	높은 초기비용 , 트래픽	SSL 자체 부하 ,

	고유정책반영 어려움	제어 , QoS 미약	UDP 불가

- 원격근로자의 증가, IoT 장비의 보안요구 강화 등의 네트워크 환경 변화로 장비 설치 없이 VPN 의 구성이 가능한 SSL VPN 이 활성화 되고 있음.

3. SSL VPN 방식의 개념과 동작절차

가. SSL VPN 의 개념도

- 사용자들은 별도의 장비나 클라이언트 소프트웨어 없이 브라우저에서 URL 을 통해 접속
- 애플리케이션 단의 강력한 사용자 인증을 실시 가능
- IPSec 에 비해 가벼운 Agent 를 설치 하여 모든 애플리케이션을 지원

나. SSL VPN 의 동작절차

No.	절차	상세 설명
①	Client 의 접속요청	클라이언트가 서버에 접속요청 하는데 사용 가능한 암호화 및 해싱 방식을 서버에 제시
②	서버의 지원 가능 방식 통보	서버는 지원 가능한 방식을 선택하여 클라이언트에 통보 서버는 자신의 ID 를 디지털 인증서 형식으로 전송을 하는데 여기에는 서버명과 CA 주소, 공개키를 포함 하여 전송
③	인증서검증	클라이언트는 필요 시 인증서 발급 CA 서버에 접속하여 인증서를 검증
④	세션키 생성	확인이 되면 보안접속에 필요한 세션 키를 생성하기 위해 서버의 공개키를 이용하여 임의의 수를 암호화 시킨 후 서버에 전송
⑤	터널링 형성	사용한 임의의 수를 이용하여 암호화 / 복호화 필요한 키를 생성하게 되며 터널링이 형성

- SSL VPN 은 구성과 관리 용이성으로 사용의 확대와 함께 비 인가자의 접근, VPN 계정관리, 모니터링 제약 등의 보안 위험이 존재하며, 이에 대한 방안도 함께 고려되어야 함.

4. SSL VPN 의 위험과 대응방안

위험 구분	위험 항목	대응정책
관리적 위험	권한 및 계정관리	권한 발급 기준수립, 프로세스 모니터링
	계정발급/회수 절차	계정,권한 발급/회수의 절차화

Notes

기술적 위험	모니터링/차단의 한계	SSL 복호화을 통한 탐지/차단기술 적용
물리적 위험	비 인가자 인식 불가	Multi Factor 인증을 통한 본인인증 강화

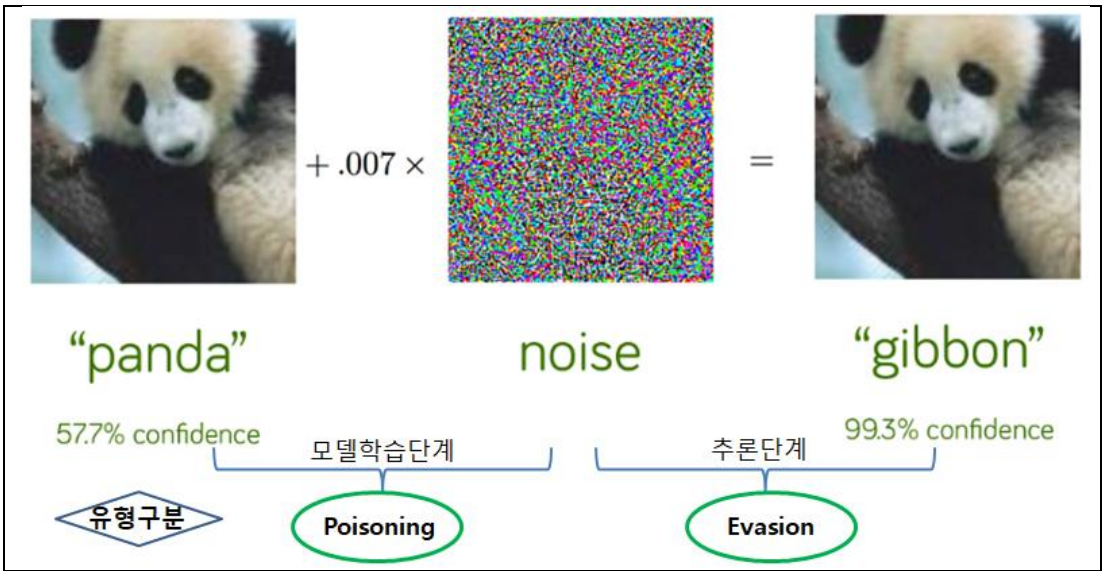
- 정책 수립 및 시스템 도입을 통한 보안강화로 VPN 의 보안위험을 감소시키는 노력필요

"끝"

Notes

3	적대적 공격(Adversarial Attack)
문제	4 차 산업혁명을 주도하고 있는 인공지능 머신러닝 기술은 실제로 적대적 공격(Adversarial Attack)에 취약한 것으로 알려져 있다. 다음에 대하여 설명하시오. 1) 자율주행자동차에 대한 적대적 공격 2) 적대적 공격을 위한 적대적 샘플(Adversarial Sample) 제작기법 3) 적대적 공격에 대한 방어기법
도메인	인공지능
정의	분류 성능이 우수한 DNN 을 이용한 Classifier 들에 적대적 교란을 적용하여 분류 알고리즘이 Misclassification 을 발생하도록 만드는 인공지능의 인식 오류를 이용한 공격
키워드	Misclassification, 표적공격, 비표적 공격, gradient-descent (기울기 하강), Perturbation, 적대적훈련, Gradient Masking/Distillation
출제의도분석	DNN 의 취약점을 악용하여 손쉽게 이미지 인식오류를 일으키는 적대적 공격이 자율주행 자동차에 미치는 연구가 이루어지고 있으며, 이러한 적대적 공격에 대한 이해와 대응 방안을 확인하기 위하여 출제
답안작성 전략	GAN 의 개념과 혼동하여 유추하여 작성을 피할 것. 적대적 공격에 대한 개념을 정확하게 작성하고, 샘플 제작기법과 방어 기법에 대해서는 전문지식 기반의 알고리즘 나열이 불가하다면, 패치 적용 기술을 바탕으로 덧붙여 서술하고, 방어기법에 대한 본인 의견을 작성할 것
참고문헌	애드버세리얼 패치(Adversarial Patch).pdf 딥러닝 기반 의료 영상 인공지능 모델의 취약성: 적대적 공격(학회지) https://rain-bow.tistory.com/entry/Adversarial-Attack 적대적 공격 동향
풀이 기술사님	이술사 PE (정보관리 기술사 / onlyitpe@gmail.com)

1. 딥러닝 취약점, 적대적 공격의 개념

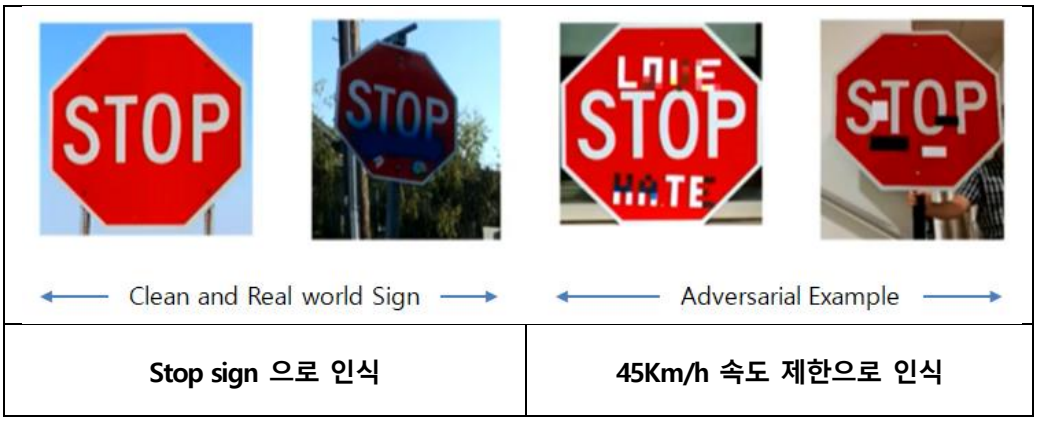


- 분류 성능이 우수한 DNN 을 이용한 Classifier 들에 적대적 교란을 적용하여 분류 알고리즘이 Misclassification 을 발생하도록 만드는 인공지능의 인식 오류를 이용한 공격

- 학습단계에서 모델을 방해하는 "Poisoning"과 추론단계에서 오작동을 유발하는 "Evasion" 구분됨
- 인공지능에 의존하는 영역이 늘어남에 따라 불가피하게 기계만 식별할 수 있는 정보의 종류와 형태가 증가하게 되고, 특히 이미지 인식이 중요한 자율주행에 큰 위협이 되고 있음

2. 자율주행자동차에 대한 적대적 공격

가. 자율주행 자동차에 대한 적대적 공격 사례



Stop sign 으로 인식	45Km/h 속도 제한으로 인식

- 사람이 못 보는 특징 기반해 판단 버그가 아니라 보는 방식이 다른 것에 기인하는 AI '취약점'을 이용하여 자율 주행을 위한 인지기술에 오류를 발생시킴

나. 자율주행 자동차에 대한 적대적 공격 유형

구분	표적 공격	비표적 공격
개념설명	출력을 사전에 특정한 대상 레이블로 의도적으로 변경하려고 하는 공격	단지 분류자가(무엇이 되었든) 잘못 된 레이블을 선택하도록 하는 공격
공격 사례	자율 주행 자동차가 표지판을 오인식하여 특정 속도, 특정 방향으로 주행하도록 유도. 예) 2017 년 7 월 미국 워싱턴대·미시건대 공동연구진은 '정지' 교통 표지판에 스티커를 붙이자, 인공지능 이미지 인식모델이 '속도제한 시속 45 마일' 표지판으로 인식	자율 주행 자동차가 표지판을 정확하게 인식하지 못하도록 방해 유도. 예) 2019 년 3 월엔 중국의 텐센트 산하의 킨시큐리티 연구소가 이러한 적대적 사례를 이용해 자율주행 차 시스템이 점이 3 개 찍힌 도로를 역 주행 하게 하는 실험 성공

- 인공지능의 이미지 인식의 결함을 드러내는 이러한 '적대적 사례'는 자율주행뿐만 아니라 군사공격, 의료진단과 같은 영역에서 사용될 경우 치명적 결과로 이어져, 인공지능의 안전성과 신뢰성을 근본적으로 위협하는 요인이 됨

3. 적대적 공격을 위한 적대적 샘플(Adversarial Sample) 제작기법과 방어 기법

가. 적대적 샘플 제작기법

제작방법	설명	요소 기술
Fast Gradient Sign Method (FGSM)	학습 시 기울기 하강(gradient-descent) 과정에서 사용된 기울기 방향과 반 대 방향에 해당하는 perturbation 을 이미지에 추가하여 학습을 저하시키는 방법	gradient-descent (기울기 하강), perturbation

	손실함수를 증가시키는 perturbation 을 영상에 추가, 알고리즘의 단순성으로 인해 공격 성공률은 낮은 편	
One-Step Target Class Methods	한 특정 레이블 Y target 에 속할 가능성이 적은 어떠한 이미지 X 에 대해서 그 이미지가 특정 레이블에 속할 확률 p (Y target | X)을 최대화하는 방향으로 perturbation 을 추가하는 학습을 진행 유사하지 않거나 전혀 연관성이 없어 보이는 레이블로 오인하도록 수행, 효과 최대화	Targeted FGSM, 확률 p (Y target | X)
DeepFool	특정 입력 포인트에서 손실함수를 반복적으로 선형화하고(해당 입력 포인트에서 손실함수에 접하는 접선 벡터를 구하여 근사 한다), 이 선형 근사가 정확하다면 학습모델의 추론 결과를 전환하는 데 필요한 최소한의 perturbation 이 적용	탐욕 알고리즘, 선형화
Jacobian-Based Saliency Map Attack (JSMA)	분류 결과를 판단하는 데 있어 입력 이미지에서 어느 특정 부분들이 주요하게 영향을 미쳤는지를 나타낸 일종의 주의맵(attention map)인 saliency map 을 gradient 기반으로 구성하여 이를 토대로 입력 이미지에서 어느 부분들에 대해 최소한의 수정을 가하면 원하는 레이블로 분류 모델이 오작동하도록 하는 것이 가능할지를 판단하는 방법	표적공격, saliency map
NewtonFool	입력 이미지에서 화소 단위로 특정 레이블에 높은 확률로 속하는 화소와, 반대로 해당 레이블에 높은 확률로 속하지 않는 화소를 뉴턴 알고리즘을 통해 찾는 방법	최소한의 perturbation 을 탐색

- 구글은 이러한 알고리즘을 통해 Adversarial 패치를 생성하는 스티커 제작방법도 공개함.

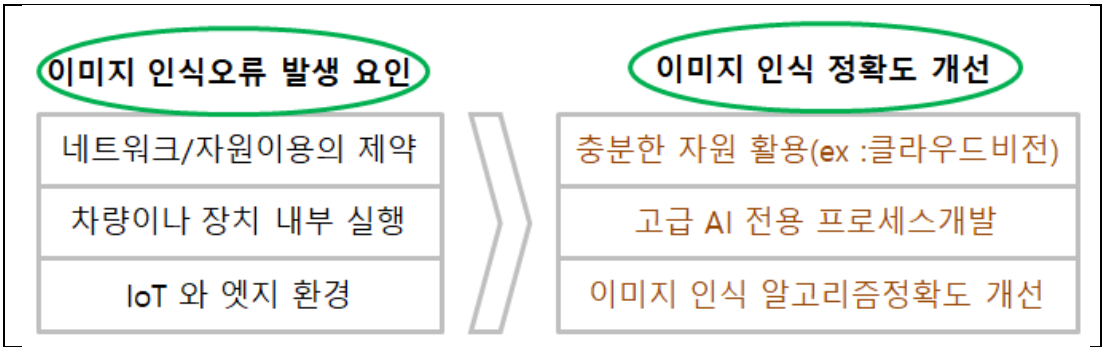
나. 적대적 공격 방어기법

방어기법	설명	효과
적대적 훈련(Adversarial Training)	자동 진단 모델을 학습시킬 때, 적대적 사례로서 작동할 수 있는 모든 경우의 수를 미리 학습 데이터셋에 포함시키는 것	모델의 정규화 (regularization)에 도움 및 과적합(over- fitting) 을 방지
Gradient Masking/Distillation	학습모델의 gradient 가 출력으로서 그대로 노출되는 것을 방지하거나(gradient masking), 학습 모델의 구조상 gradient 자체를 일종의 정규화 방법과 같이 두드러지지 않게 하여 적대적 공격의 학습 방향에 힌트를 주지 않도록 하는 방법(distillation)	과한 gradient 에 패널티를 주는 일종의 정규화 방 식으로 학습모델의 강인성을 확보
Feature Squeezing	본래의 학습모델과 별도로, 주어진 입력에 대해 적대적 사례인지 아닌지를 판단하는 학습모델을 덧붙이는 것 특징 공간(feature space)에서의 불확실성 (uncertainty)	주어진 이미지에서 의 perturbation 을 감지하여 적대적 사례 여부를 판별

	의 평가(특히, dropout 을 이용함으로 인한) 및 밀도 추정 (density estimation)을 수행	

- 그 외에 다수의 학습모델을 앙상블(ensemble)하여 시스템을 구성하면, 특정 모델에 대한 화이트박스 공격을 피할 수 있음은 물론이고, 다수의 학습모델에 범용적으로 적용되는 적대적 공격은 개발이 어렵다는 점으로 인해 좋은 방어법이 된다는 연구 결과도 있음

4. 적대적 공격 방어를 위한 이미지 인식 환경 개선

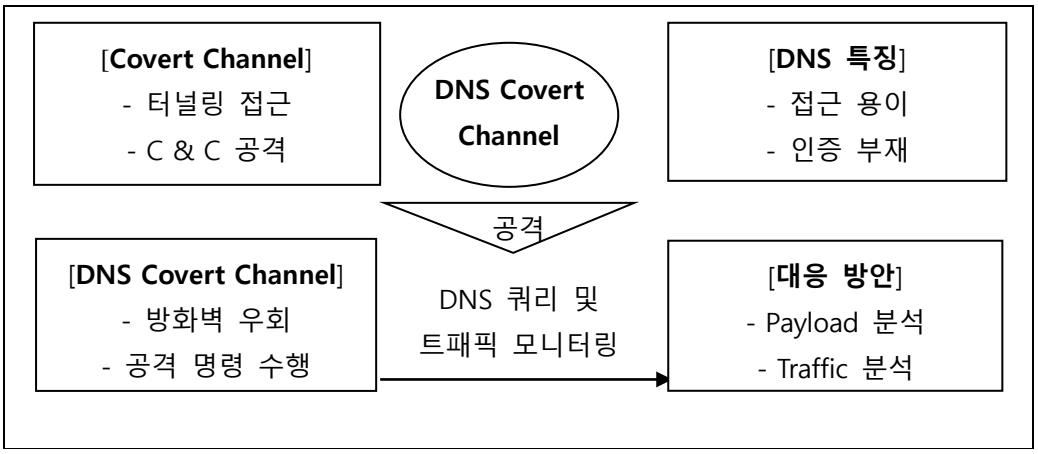


- 보안업체 뿐 아니라 해커들도 AI 를 이용하므로 이미지 알고리즘 정확도 개선 노력이 요구되며, 차량이나 장치 내부 등 대규모 연산의 제공이 어려운 환경에서의 최적화를 위하여 객체 판정 알고리즘과 고급 AI 전용 프로세스 개발이 필요함.

"끝"

4	DNS Convert Channel 공격
문제	DNS 를 은닉채널(Convert Channel)로 사용하는 이유를 설명하고, DNS Convert Channel 공격모델 및 방어 기법을 설명하시오.
도메인	보안
정의	방화벽을 통과하는 DNS 패킷 데이터 Payload 에 공격을 위한 데이터를 탑재하여 내부 Host 를 공격하는 공격 기법
키워드	방화벽 우회, 켑슐화, C&C 서버, 터널링, Traffic/Payload 분석
출제의도분석	다양하게 등장하고 있는 보안 공격 기법중에 취약한 DNS 를 통한 공격 기법을 파악
답안작성 전략	물어본 DNS 사용 이유, DNS Covert Channel 공격모델 설명과 방어기법을 집중하여 작성
참고문헌	https://www.slideshare.net/barryjcoatesworth/cyber-espionage-tinker-taylor-soldier-spy https://slidesplayer.org/slide/11291263/
풀이 기술사님	이민홍 PE (제 117 회 정보처리기술사/ iminred@naver.com)

1. DNS 를 통한 터널링 공격, DNS Covert Channel 개념



- **은닉채널(Covert Channel):** 데이터 전송 시 평소에 잘 사용하지 않는 프로토콜 패킷의 공간에 공격 명령어 및 코드를 담아 전송하는 공격 방법
- Covert Channel 은 **Covert Storage Channel**(시스템에 저장된 정보를 변경하여 정보 전달)과 **Covert Timing Channel**(시스템 자원의 타이밍을 수정하거나 성능을 변화시켜 정보 전달)하는 기법이 있으며 DNS Covert Channel 은 Storage Channel 의 유형임
- **DNS Covert Channel**: 방화벽을 통과하는 DNS 패킷에 공격을 위한 데이터를 탑재하여 내부 Host 를 공격하는 공격 기법

2. DNS 의 은닉 채널 사용 이유

가. DNS 접근성 이유

이유	설명	특징
외부 오픈	내외부의 접근 허용 시스템	내부 및 외부에서 접근이 용이한 취약
	해커의 주요 접근 경로	개방성으로 인해 해커에게 기본적 노출 경로
방화벽	Query 로 방화벽 통과	Open 된 알려진 Port(Port 54)

우회	내부 데이터 캡슐화	Payload 에 데이터를 Base32/64 로 인코딩 전달
질의 처리	Query 로 데이터 전송 악용	Query 에 데이터를 담아 DNS 를 통해 외부 전송
	Query 결과의 응답 악용	Query 결과에 데이터 및 명령어를 담아 응답

- DNS 는 기본적으로 외부에서 Query 를 받아 응답하는 시스템으로 공개되어 있는 시스템임
- DNS 는 시스템 목적상 개방 및 노출이 우선되어야 함으로 접근성으로 인해 외부의 1 차적 공격 목표로 떠오름

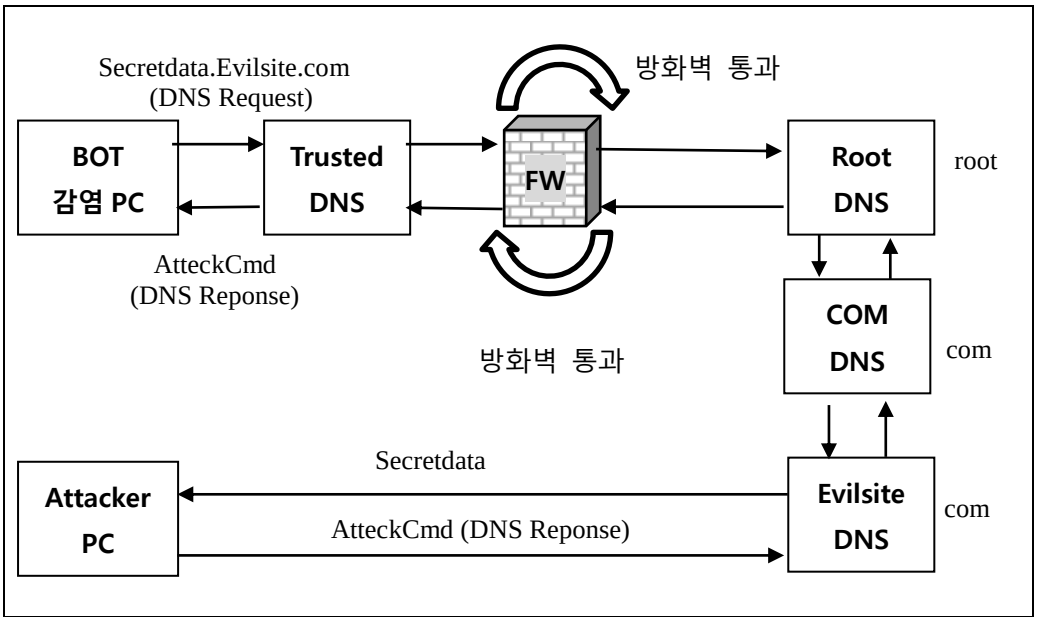
나. DNS 취약성으로 인한 사용 이유

이유	설명	특징
인증 부재	접근시 인증 부재	임의 사용자의 사용을 위해 접근시 인증 부재
	임의 사용자의 사용 허용	모든 DNS 질의 사용자의 DNS 사용 가능
UDP 사용	서버간 세션의 부재	Trust Server 에 대한 인증 불가(비 DNSSEC)
	DoS 공격에 취약	DNS 자체가 공격에 이용 가능
캡슐화 가능	Encoding 에 의한 Data 은닉	DNS 레코드(TXT, CNAME 등)로 IP 트래픽 캡슐화
	미사용 프로토콜 영역 존재	악의적 데이터 및 명령어 은닉 사용 가능

- DNS Covert Channel 공격은 DNS 를 통한 우회 공격 경로의 모델을 제공함
- DNS 를 통한 내부 호스트와 외부 공격자간의 터널링을 생성하여 외부에서 내부 호스트에 악의적 행위를 가할 수 있음

3. DNS Covert Channel 공격모델 및 방어 기법

가. DNS Covert Channel 공격모델



- DNS Query 의 Request 및 Response 를 통해 데이터와 명령어를 전달 가능함
- DNA Covert Channel 을 방어하기 위해서는 Payload 분석과 Traffic 분석을 수행함

나. DNS Covert Channel 방어 기법

유형	상세 방어 기법	설명
Payload Analysis	Payload 사이즈 분석 (Size of request and response)	Payload 에 공격 데이터로 인한 사이즈 변화 유무를 분석

	요청 Host 의 변동율 분석 (Entropy of hostnames)	요청 호스트가 평소 정상적 요청을 한 호스트 여부인지를 확인
	DNS Name 에 공격 문자 여부 분석 (Statistical Analysis)	DNS Name 의 특정 문자 분석
	평소 미사용 레코드 사용여부 확인 (Uncommon Record Types)	잘 사용 않는 레코드의 사용시 사용여부 및 사용 목적 확인
	금지된 정택의 요청 여부 확인 (Policy Violation)	금지된 정책의 요청이 잦은 경우 공격인지 질의 점검
	Specific Signatures	특정 DNS 터널링의 시그니처 확인
Traffic Analysis	IP 당 과다 DNS 트래픽 확인 (Volume of DNS traffic per IP address)	DNS 패킷 정보제약으로 필요 데이터 전송을 위한 IP 당 DNS 트래픽 과다 확인
	IP 당 과다 DNS 트래픽 확인 (Volume of DNS traffic per domain)	DNS 패킷 정보제약으로 필요 데이터 전송을 위한 도메인당 DNS 트래픽 과다 확인
	도메인에 호스트가 과다 활당 확인 (Number of hostnames per domain)	공격을 위해 도메인당 여러 호스트 네임 할당 확인
	DNS 의 지리적 위치 확인 (Geographic location of DNS server)	비즈니스 무관한 지역 DNS 사용시 점검
	도메인 이력 점검(Domain history)	A/N 레코드 추가시 도메인 이력 확인
	NXDomain 응답량 점검 Volume of NXDomain responses	미존재 도메인 응답의 과다 확인
	Visualization	트패픽 가시화를 통한 모니터링
	Orphan DNS requests	상호 응답이 없는 DNS 요청을 확인
	General covert channel detection	Covert Channel 탐지도구 사용

- DNS 터널링을 위해서는 한정된 DNS Query 내에서 데이터를 주고받아야 함을 DNS 패킷의 수가 증가하게 됨으로 **DNS 요청 및 응답 쿼리의 크기를 모니터링**하여 감지 가능함 (터널링 된 트래픽은 DNS 에서 64 자를 초과 가능성이 있음).
- 특정 소스의 DNS 트래픽이 매우 많은 경우 트리거하도록 **SIEM 에서 규칙 구성** 필요.
- DNS 서버에서 많은 수의 **DNS TXT 를 모니터링**하도록 규칙 구성 필요.
- DNSTrap 은 인공 신경망을 사용하여 **DNS 터널링 탐지 도구**
- Paloalto 및 Fire Eye 와 같은 **차세대 방화벽**의 DNS 터널링 감지 기능 활성화 가능.

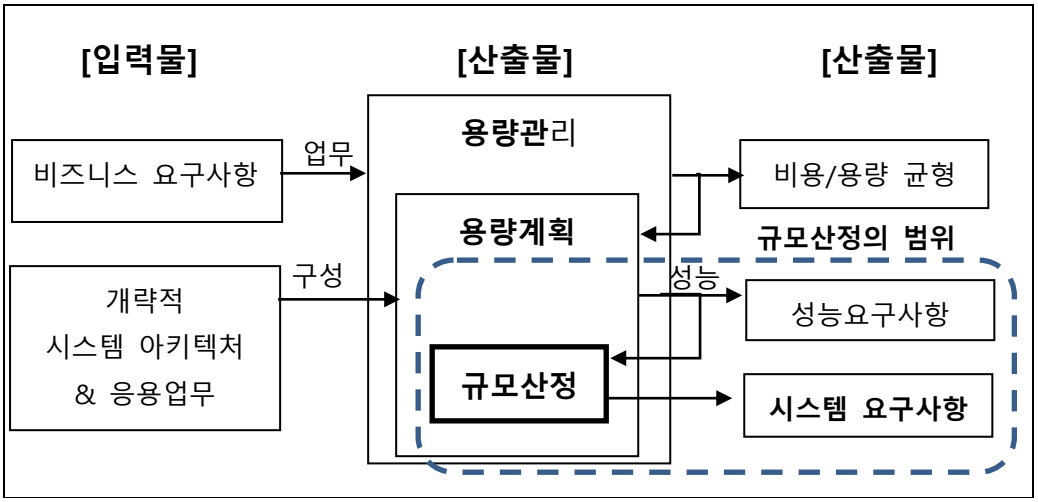
"끝"

-

5	H/W 용량 산정
문제	ITA 기반으로 정보시스템 H/W 용량을 산정하고자 한다. 1) H/W 규모산정 방법에 대한 개념 및 장·단점 2) 규모산정 대상 3) CPU 및 스토리지의 성능 기준치
도메인	CA/OS
정의	기본적인 용량과 성능요구사항이 제시되었을 때, 그것을 시스템 요구사항으로 변환하는 것
키워드	수식계산법, 참조법, 시뮬레이션법, tpmC/TPC-C, Max-jOPS/SPECjbb2015, IOPS/ SPC-1
출제의도분석	2018 년 개정된 규모산정 지침에 대한 이해 필요
답안작성 전략	개념적인 풀이가 아닌 HW 규모산정 지침을 기본으로 하여 답안작성
참고문헌	(2018-1727)정보시스템 하드웨어 규모산정 지침(2018-08-29)
풀이 기술사님	이민홍 PE (제 117 회 정보처리기술사/ iminred@naver.com)

1. H/W 용량관리와 규모산정의 개념

가. HW 용량 관리의 개념도



- H/W 규모산정은 "기본적인 용량과 성능요구사항이 제시되었을 때, 그것을 시스템 요구사항으로 변환하는 것"으로 사전 예측적 성격이 있으므로 규모추정이라고도 함.
- **규모산정은** 시스템의 아키텍처와 응용 기반을 전제로 성능 요구사항과 성능을 결정하는 용량계획이나 용량관리의 하위적 개념

나. 용량관리, 용량계획, 규모산정의 개념 비교

구분	정의	관점	시간성
용량 관리	**비즈니스 요구사항을 충족**시키기 위한 **현재와 미래의 용량 계획을 수립**하고 **비용(Cost)과 용량(Capacity)의 균형**을 맞추는 것	조직	지속적

용량 계획	개략적인 시스템 아키텍처와 응용 업무를 기반으로 **시스템에 요구되는 성능 요구사항과 성능을 결정하기 위한 계획**	조직, 시스템	지속적
규모 산정	**기본적인 용량과 성능 요구사항이 제시**되었을 때, 그것을 **시스템 요구사항으로 변환**하는 것	시스템	일시적

- 시스템 규모산정은 실제 업무와 응용을 기반으로 수학적 방법론을 사용하여 도입하는 시스템 규모를 추정 혹은 계산하는 것으로 수식계산법, 참조법, 시뮬레이션법으로 산정함

2. H/W 규모산정 방법에 대한 개념 및 장·단점

가. H/W 규모산정 방법에 대한 개념

구분	개념	특징
수식계산법	사용자 수 등 규모산정을 위한 수요를 토대로 용량수치를 계산하고, 보정치를 적용하는 방법	- 초기 계획 가능 - 산정 용이
참조법	업무량(사용자 수, DB 크기)에 따라, 기본 데이터를 토대로 대략적인 시스템 규모를 비교하여 비슷한 규모를 산정	- 경험기반 산정 - 유사 사업에 효과적
시뮬레이션법	대상 업무에 대한 작업부하를 모델링하고 이를 시뮬레이션하여 규모를 산정	- 모델링 수행 - 고정밀/고비용

- 공공 부문에서는 각 방식의 장단점을 고려하며 사업 입안 단계에서도 규모산정이 가능하여야 한다는 특성을 고려하여 수식계산법을 적용함
- 산정결과의 정확성과 객관성 향상을 위해서 세 가지 방법 적절하게 활용할 필요가 있음.

나. H/W 규모산정 방법의 장·단점

구분	장점	단점
수식계산법	- 명확한 근거 제시 가능 - 타 방법에 비해 간단한 산정	- 보정치 오차에 영향도 큼 - 정확한 (보정)근거자료 제시 곤란
참조법	- 유사사업에 정확성 - 비교적 안전한 규모산정 가능	- 근거 제시 미약(비교에 의함)
시뮬레이션법	- 상대적으로 정확한 값 획득	- 시간과 비용이 많이 소요

- 규모산정 대상은 CPU 형태나 수, 디스크 크기나 형태, 메모리 크기 등이 있음

3. 규모산정 대상 및 설명

가. 규모산정의 대상

구분			시스템 유형	
대상			**OLTP**	**WEB/WAS**
서버	**CPU**		O	O
	메모리		O	
	디스크	시스템	O	
		데이터	O	
스토리지			O	O

- 시스템 가격 및 성능 측면에서 가장 중요한 CPU, 메모리, 디스크, 그리고 스토리지 등 4 가지 분야를 규모 산정 분야로 정의

나. 규모산정의 대상 설명

대상	설명	특징
CPU	해당 업무를 처리하기 위한 CPU 규모를 계산한 후, 적정한 성능을 지닌 서버기종을 선정.	- OLTP, WEB/WAS 등 서버의 작업부하 특성에 따**라 선정**
메모리	CPU 규모 산정에 따른 서버 구성 방안에 의거하여, 서버별 시스템 S/W, 응용 프로그램 등의 메모리 사용량을 산정	- 메모리 유형 - 메모리 용량
디스크	CPU 규모 산정에 따른 서버 구성 방안에 의거하여, 서버별 OS, 시스템 S/W, DB 의 데이터, DB 의 아카이브(Archive) 및 백업 영역 등의 디스크 사용량을 산정	- 각 SW 및 Data 별 저장 용량 검토
스토리지	CPU 를 기준으로 산정된 서버 규모에 따라 필요한 스토리지의 규모를 산정한다.	- IOPS 등 성능 검토

- 4 가지 대상 중 CPU 및 스토리지는 시스템의 성능에 직접적인 영향을 미치는 요소로 성능 기준치의 측정이 중요함

4. CPU 및 스토리지의 성능 기준치

가. CPU 및 스토리지 성능 기준치

구분	CPU			스토리지
	OLTP 또는 OLTP & 배치 어플리케이션 서버	**WEB 서버**	**WAS 서버**	
성능측정치	tpmC	Max-jOPS		IOPS
참조 성능기준	TPC-C	SPECjbb2015		SPC-1

- TPC-C, SPECjbb2015 및 SPC-1 을 사용하여 성능 측정치를 측정함

나. CPU 및 스토리지 성능 기준치 참조 성능 기준의 상세 설명

성능 기준	측정 방법	측정 결과
TPC-C	다양한 크기와 복잡도를 가진 서로 다르면서도 연관성이 있는 DB 테이블에 수행할 수 있는 5 개 트랜잭션 규정, 분당 트랜잭션 측정	- **tpm**(Transaction Per Minute)을 기본 단위로 사용하고 tpmC 로 표현 - 사용자 생각시간을 고려
SPEC jbb2015	자바 응용 프로그램 기반 시나리오 (Multi JVM 구성)	-**Critical-jOPS**: SPEC 에서 지정된 응답시간 이내 부하처리 성능 -**max-jOPS**: 응답시간과 상관없이 Fail 발생 직전의 최대 부하처리 성능

Notes

SPC-1	일반적 서버급 컴퓨터 시스템에서 온라인, 비휘발성 스토리지로 표시되는 실제 환경 시뮬레이션	- 40 % 읽기 요청과 60 % 쓰기 요청 -백엔드: 3 개의 ASU (Disjoint Application Storage Unit)로 구성

- 기존의 IDC 기반의 규모산정이 클라우드 환경으로 변화되어 가면서 많은 추가 고려 사항 발생

5. 클라우드 환경의 규모 산정의 고려사항

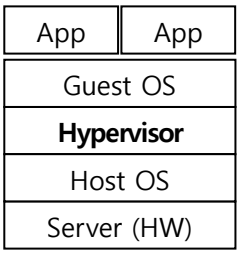
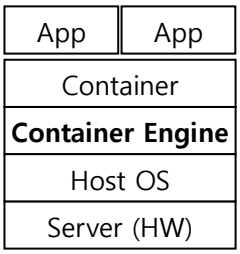
분야	설명	특징
벤더 기반 산정	클라우드 벤더에 의해 성능 및 용량이 상이함	벤더 특성 부상
	클라우드 벤더가 제공하는 용량 기준을 참조	업체 제공 기준 고려
가변성 큼	시스템 용량은 상시 확장 및 축소 가능함	가변성 고려 비용 효율화
	피크타임 기준 용량산정보다는 평상시 기준 용량산정이 필요함(미사용시 OFF)	기준 운영 용량 산정
시스템 아키텍처 복잡	MSA 아키텍처 특성 이해가 필요	가변성 큰 서비스 사용
	FaaS, Spot instance 등 Saas 의 용량 영향 고려	신규 서비스 검증 필요

- 클라우드 서비스의 규모 산정은 클라우드 인스턴서의 상시적 개선으로 규모 산정 역시 상시 클라우드의 특성을 감안하여 수행 필요함

"끝"

6	가상머신(VM: Virtual Machine)와 컨테이너
문제	클라우드 시스템 구축을 위한 핵심 기술인 가상화 관련 기술 중 가상머신과 컨테이너를 비교하여 설명하시오.
도메인	신기술
정의	가상머신: 하이퍼바이저를 사용하여 물리적 인프라의 기능을 SW 로 가상화한 논리적 기능으로 제공하는 기술: 컨테이너: 컨테이너 엔진으로 어플리케이션의 수행의 하나의 프로세스 상에서 격리하여 수행시키며 자원을 공유하게 하는 기술
키워드	하이퍼바이저, 자원 공유 및 격리,
출제의도분석	클라우드 환경을 넘어서는 컨테이너 환경에 대한 이해 확인
답안작성 전략	클라우드와 컨테이너 환경의 다양한 관점에서의 비교 작성
참고문헌	https://www.alibabacloud.com/ko/knowledge/difference-between-container-and-virtual-machine
풀이 기술사님	이민홍 PE (제 117 회 정보처리기술사/ iminred@naver.com)

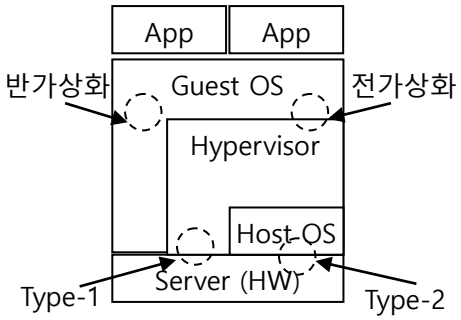
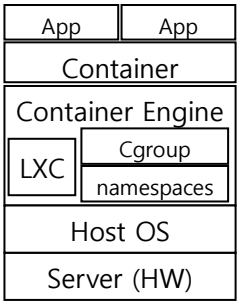
1. 물리 자원의 논리적 재구성, 가상머신(VM: Virtual Machine)와 컨테이너의 개념 비교

구분	가상머신	컨테이너
개념도 비교		
정의 비교	하이퍼바이저를 사용하여 물리적 서버 자원을 논리적 서버로 재 구성하여 제공하는 가상화 기술	컨테이너 엔진으로 어플리케이션에 자원을 공유 및 격리하여 제공하는 가상화 기술
특징 비교	OS 환경의 가상화 제공	Application 가상화 제공
	물리자원의 독점	물리자원의 공유 및 격리

- 가상머신은 하이퍼바이저 기반으로 HW 자원에 대한 App 의 접근을 가상화 하여 독점 제공함
- 컨테이너는 컨테이너 엔진기반으로 물리자원의 공유 및 격리를 제고하여 가상화를 지원함
- 가상 머신은 하드웨어 스택을 가상화함. 컨테이너는 운영체제 수준에서 가상화를 실시하여 다수의 컨테이너를 OS 커널에서 직접 구동함.

2. 가상머신과 컨테이너의 아키텍처 비교

가. 가상머신과 컨테이너의 아키텍처 비교

구분	가상머신	컨테이너
아키텍처 비교		
아키텍처 설명	– 하이퍼바이저를 HW 위에 바로 생성하는 여부에 따라 Type-1/2 로 구성 – Guest OS 의 HW 접근 여부에 따라 전/반가상화로 구성	– 리눅스 환경에서 자원의 격리 및 공유를 담당하는 C-group, namespace, 이를 관리하는 LXC 로 구성

- 가상 머신은 HW 및 OS 에 대한 접근 지원하는 여부에 따라 성능에 많은 차이를 보임
- 컨테이너는 리눅스의 자원 공유 기능을 사용하여 컨테이너를 생성함

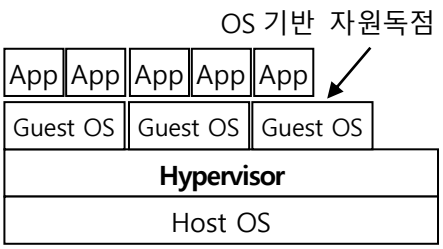
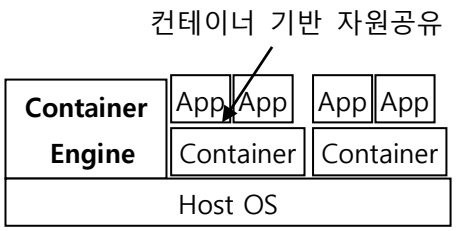
나. 가상머신과 컨테이너의 아키텍처 기술 요소 상세 비교

구분	가상머신	컨테이너
Application 환경	**전가상화**: Guest OS 는 하이퍼바이저로만 HW 접근(저성능,)	**LXC**: namespaces 와 Cgroup 표준을 정의한 OCI(Open Container Initative) 스펙을 구현한 컨테이너 기술의 구현체
	반가상화: Guest OS 는 직접 HW 접근 (고성능, Guest OS 한정)	
HW 관리	**Type-1**: 베어메탈 서버에 하이퍼바이저 기동(고성능, 저호환성)	**Namespaces**: 각 게스트 머신 별로 독립적인 공간을 제공하는 기능
	Type-2: Host OS 에서 하이퍼바이저 기동 (저성능, 고호환성)	**Cgroups**: 자원에 대한 제어를 가능하게 해주는 리눅스 커널 기능
지원 환경	**다양한 OS** (Guest OS 선택) 지원	**단일 OS** 만 선택 가능
	대용량 서버 인프라 **제공**	**경량(프로세스 수준) 실행환경** 제공

- 가상머신과 컨테이너는 자원에 대한 독점 및 공유/격리가 가장 큰 차이점임
- 자원에 대한 독점을 제공하는 가상머신보다 효율적으로 자원 관리가 가능한 컨테이너 부상중

3. 가상머신과 컨테이너의 성능 비교

가. 가상 머신과 컨테이너의 성능에 대한 개념 비교

구분	가상머신	컨테이너
성능 비교		
성능	OS 수준의 가상화로 각 OS 는 독점적으로 자원을 할당 받으므로 자원의 효율적 사용 불가로 성능 한계	컨테이너에서 자원을 공유 받아 사용하며 App 는 격리된 공간에서 안정하게 기동되어 성능의 향상

- 가상머신의 자원 독점으로 인한 성능 제약을 컨테이너에서는 자원공유를 통해 해결하여 한정된 자원에서의 성능 향상이 가능함

나. 가상머신과 컨테이너 성능 비교 상세

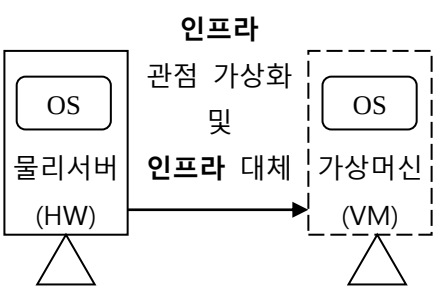
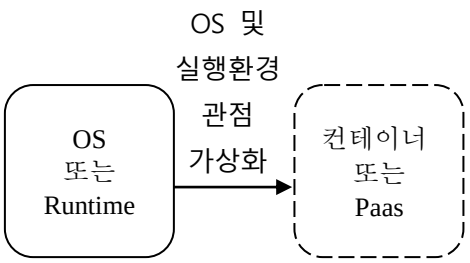
구분	가상머신	컨테이너
자원 활용	자원(HW 및 Host OS) 독점	프로세스 수준의 자원 공유 지원
	미사용 자원 점유	필요 자원에 대한 컨테이너별 격리
성능 관리	자원 부족시 성능 저하 발생	프로세스 기반의 경량화된 성능 지원
	하이퍼바이저 및 Guest OS 부하	프로세스 수준의 처리 한계
유연성	가상 머신 생성 시간 소요(Overhead)	Cold/Warm Start 로 신속 기동
	Cloud 지원 기능 사용(Auto Scale)	다양한 오픈소스 활용(Kubenetes 등)
용량	일반적으로 GB 급 공간 소요	일반적으로 MB 급 공간 소요
	전체 운영 체제와 관련 Tool 포함	필요 라이브러리와 관련 Tool 포함
유지보수 및 업데이트	Host OS 의 HW 별 개별 패치	Host OS 패치 수행
	Guest OS 의 가상 머신별 개별 패치	Guest OS 패치 불필요

- 가상 머신은 IDC 의 인프라 기반으로 확산중에 있으며 컨테이너는 MSA 와 Devops 의 기반 기술로 Application 아키텍처의 변경에 많은 영향을 주며 확산중임

Notes

4. 가상머신과 컨테이너 활용 비교

가. 가상머신과 컨테이너의 활용 현황 개념 비교

구분	가상머신	컨테이너
활용 비교		

- 가상머신은 기존의 IDC 의 서버 기반의 인프라를 대체하여 대용량 어플리케이션 지원의 기존 인프라 역할을 대체중
- 컨테이너는 Application 관점의 실행환경의 효율화를 제공하는 관점에서 Runtime 의 제공(Paas 등)으로 발전중

나. 가상머신과 컨테이너의 활용 현황 상세 비교

구분	가상머신	컨테이너
주요 사용처	IDC 등의 기존 인프라 대체(HW)	Application 아키텍처 최적화(SW)
	인프라 자원의 분할 수요	MSA 기반의 Micro Service 구성
주요 사용자	TA(Technical Architect)	SA(Software Architect)
	Infra Manager, Admin	SW 개발자 및 운영자
적용 Application	다중 어플리케이션 실행 환경	단일 어플리케이션 실행환경
	GB 급의 모놀리틱 Application 실행	MB 급의 마이크로 서비스 실행환경
활용 방법론	인프라 용양관리, 용량계획, 규모산정	Devops, MSA, 콘웨이 법칙
	인프라 아키텍처링	Application 아키텍처링
Trend	SDDC 의 기반 기술	유니커널로 경량화 추구
	가상머신의 신속하고 잦은 개선	Paas 및 Faas 의 기반 기술로 채택

- 컨테이너가 경량화 및 효율화로 급속히 발전 및 보급되고 있음
- 컨테이너 규모, 다중 OS 미지원 등 컨테이너 한계로 상머신과 상호 보완을 하며 발전중
- 기존의 HW 관점의 서버 인프라는 가상머신으로 대체되며 Application 의 실행환경으로 컨테이너가 Cloud Native Application 의 핵심기술로 부상중임

"끝"