**클라우드 #6주차 과제**

21660019 3A김세진

**1. CPU, Core**

**2. 표준**

**3. 리피터, 라우터, 허브, 공유기**

**4. IPS/ IDS**

**5. Dos, DDos, DRDos**

**6. firewall, web-firewall**

**7. UPS**

**8. NMS**

**9. test**

**1. CPU, Core**

1-(1) CPU, Core의 개념

CPU : 컴퓨터에서 구성 단위 중 기억, 해석, 연산, 제어라는 4대 기능을 종합하는 장치인 Central Processing Unit의 줄임말  
Core : CPU의 구성 중 하나이며, 주로 연산작업을 처리하는 요소이다.

1-(2) 내가 생각하는 CPU, Core의 개념과 차이

CPU는 인체와 비교하여 인체의 뇌의 역할을 들 수 있다. 정보가 들어오면 계산하여 처리하는 방식의 부품이며, 컴퓨터의 속도나 성능을 나타낸다고 볼 수 있다.   
 그렇다면 core는 무엇일지 궁금했는데, 처음에 코어는 CPU당 하나만 있었기에 CPU=Core라고 생각했다. 하지만 컴퓨터의 성능을 늘리기 위해서 Core의 클럭을 늘려야 했는데, 이로 인해 발열량과 소비전력이 높아졌기에, 차라리 Core의 수를 늘려서 연결하는 방식으로 만들었는데, 이를 멀티 코어 CPU라고 한다.   
 차이점을 찾기는 애매한 종속관계인데, Core는 CPU의 구성요소 중 하나로, CPU의 성능을 발전시키는 과정에서 1개 -> 2,4,6,8개로 점점 늘어나게 되었고, Core의 개수는 CPU의 연산 속도를 나타내게 되었다.

**2. 표준**

2-(1) 표준의 개념

표준이란 카오스(Chaos)를 코스모스(Cosmos)로 바꿔주는 것이다. 즉, 무질서의 상태에 질서와 조화를 부여해 체계화하는 것을 말한다.

2-(2) 표준의 필요성

표준은 많은 사람이 같이 쓰기 때문에 혼란을 방지하기 위해서 필수적이다. 예시를 들어보면, 사람들간의 소통(말, 언어)를 생각해보자. 세상에는 다양한 사람들이 있고, 그들의 언어나 행동, 문화가 다르다. 만약에 그런 사람들이 서로 협력할 일이 필요할 때, 어떤 언어로 소통할지 정해놓지 않는다면, 일을 할 수 없는 사태가 일어날 것이다. 이렇듯 표준이라는 것은 모두가 인정할만한 규칙을 정하여 혼돈이 일어나지 않게 하는데 필수적이다.

2-(3) 클라우드 VM 세팅 시 표준화의 필요성

클라우드 서비스의 VM을 구축함에 있어서도 많은 클라우드 서비스 제공자들과 VM이 있다. 따라서 다른 VM을 썻다고 해서 그 방식이 완전히 다르게 바꿔야 된다면, 결국에 자신의 어플리케이션은 기존의 클라우드서비스에 종속될 것이고, 호환이 힘들것이다. 따라서 클라우드 VM이나 클라우드 서비스 제공자들의 프로그램 사이에도 이런 표준의 필요성이 있다. 따라서 VM을 세팅할 때, 어떤 성능의 인프라를 쓸지, 보안은 어떻게 설정하고, 네트워크나 포트는 어떻게 설정할지, 등에 대한 것들은 미리 표준을 정해 놓았다.

**3. 리피터, 라우터, 허브, 공유기**  
 3-(1) 리피터, 라우터, 허브, 공유기의 개념

리피터 : 전자기 또는 광학 전송매체 상에서 신호를 수신하고, 증폭하며, 매체의 다음구간으로   
재전송 시키는 장치이다.

라우터 : 패킷의 위치를 추출하여, 그 위치에 대한 최적의 경로를 지정하며, 이 경로를 따라 데이터 패킷을 다음 장치로 전향시키는 장칭이다.

허브 : 여러대의 컴퓨터를 연결해 네트워크를 만들어주는 장비이다. 주로 멀티포트 기능과 리피터 기능으로 사용된다.

공유기 : 한 개의 인터넷 주소를 여러 대의 컴퓨터, 노트북, 폰 등이 인터넷을 공유할 수 있는 기능을 제공하는 장치이다.  
  
 3-(2) 리피터, 라우터, 허브, 공유기의 차이점 and OSI 7계층과 연관  
 일단 가장 낮은 계층의 물리단계에서 작동하는 것이 리피터와 허브이다. 여기서 리피터는 네트워크 선로를 통해 전달되는 신호를 중간에서 증폭하는 역할을 하는데, 쉽게 말하면 네트워크 사이의 거리를 증가시키기 위해서 사용한다.

같은 물리단계에서 작동하는 허브는 2가지 기능을 하는데 리피터 기능과 포트의 증가이다. 리피터가 하는 역할을 수행하기도 하면서 여러 장비를 하나의 네트워크에 연결시킬 수 있는 포트를 늘리고 포트를 정해주는 기능을 수행한다.

다음은 3계층의 네트워크 계층의 라우터이다. 라우터는 패킷(데이터 묶음)을 최적의 경로를 통해 목적지까지 보낼 수 있게 경로를 지정하는 역할을 한다. 이전 1계층은 물리적으로 연결하는 선, 선들을 받아들이는 장치라고 본다면, 3계층의 라우터는 수많은 이러한 선과 장치중에 어디로 연결할지를 결정하는 장치라고 생각하면 된다.

공유기는 라우터, 허브의 기능을 가지고 있는 네트워크 장비로 보면 되는데. 일단은 3계층 네트워크 계층으로 본다. 허브의 여러 장치와의 연결 기능을 가지고 있고 라우터의 패킷을 어디로 전송시킬지 결정하는 라우팅 기능도 가지고 있다. 그렇다면 차이점은 무엇일까.

라우터는 한 장치가 외부 네트워크 어디로 연결하게 해줄지 결정하는 역할을 하며, 공유기는 하나의 IP를 가지고 여러 컴퓨터가 인터넷을 사용할 수 있도록 하는 것이다.

**4. IPS, IDS**

4-(1) IPS, IDS의 개념  
IPS : (Intrusion Prevention System) 직역하면 침입 예방 시스템이며, 비인가 프로그램이나 해커의 공격에 대해서 미리 rule파일에 정의가 되있다면 차단을 시키거나 통과시켜주거나 하는 식의 설정이 된 시스템이다.  
   
IDS : (Intrusion Detection System) 직역하면 침입 탐지 시스템이며, 비인가 프로그램이나 해커의 공격이 이뤄지면 해당 트래픽을 복사하여 로그를 만들고 해당 로그파일을 가지고 관리자가 대응을 할 수 있게 하는 시스템이다.   
  
 4-(2) 내가 생각하는 IPS, IDS의개념  
일단 IPS, IDS는 모두 비인가 프로그램이나 해커의 공격을 막기 위해서 만들어진 보안관련 소프트웨어인 것 같다. 나는 이것을 메이플이라는 게임으로 이해를 해 보았다. 이 RPG게임의 특성상 매크로라는 프로그램은 꽤나 매력있게 다가왔기 때문에 매크로에 대해서 공부를 조금 한적이 있는데, 그 때에 그 게임의 보안 프로그램에서 들은 것이 있다. 클라이언트의 메모리를 탐색하여 매크로 프로그램과 관련된 프로그램이 실행 중이라면, 서버와 해당 클라이언트와의 연결을 끊어 버리는 식이었다. 이 방식은 IPS라고 볼 수 있는데, 미리 짜여진 rule파일에 이러한 작업들이 미리 되있기 때문에, 매크로 프로그램이 실행 중이라면 끊어지는 것이다. 해결 방법은 매크로 프로그램을 허용가능한 프로그램으로 속이면 간단히 통과가 되는 식이었다. 간단히 매크로 프로그램을 internet explorer가 실행중인 것처럼 처리하는 방식이었다. 이러한 경우에 아무런 제지없이 프로그램이 사용되어졌는데, 이러한 방식의 IPS의 취약점 때문에 아마 더 강한 개념이 IDS인 것 같다. 실시간으로 불법 프로그램이나 해커의 공격의 트래픽을 복사하여 로그를 만들고 사람이 직접 차단하거나 대응할 수 있게끔 말이다.

4-(3) 현업에서 어떻게 쓰일지  
 IPS는 위의 예시에서 보듯이 어떤 서비스의 보안 프로그램과 같은 곳에 많이 사용될 것 같다. 수 많은 돈이 오가기 때문에 보안이 철저해야 하며, 또한 수많은 트레픽이 몰려들기 때문에 관리자가 일일이 처리하기가 곤란하기 때문에 일차적으로 부정프로그램이나 해커를 막는데 도움이 될 것이다.   
 IDS는 왠지 IPS에서 처리하지 못한 심오한 단계의 보안을 처리하는데 사용할 것 같다. 예시를 들어보자면, 범국가적인 해킹에서 처리를 살펴보면 이해가 쉬울 것이다. 우리는 종종 어떤 국가 기관이 뚫렸다던가 어떤 기업에서 데이터가 유출 됬다는 소식을 가끔 듣는다. 그런 기관이나 기업이 과연 보안에 투자를 하지 않고 당한 것일까? 내가 생각하기에는 아니다. 그래도 뚫리는 것이 보안이기 때문에 이중적으로 막고 방어하는 장치가 IDS인 것 같다.   
 따라서 내가 생각하기에 IPS라는 시스템 위에 IDS시스템까지 놓아서 보안을 철저히 하는 식으로 사용할 것 같다.

**5. DoS, DDoS, DRDoS**

5-(1) Dos, DDos, DRDos의 개념

DoS : (Denial of Service)서비스 거부 공격/ 서비스 방해 공격이라는 뜻으로, 한 사용자가 시스템의 리소스를 독점하거나 모두 사용, 또는 파괴함으로써 다른 사용자들이 이 시스템의 서비스를 올바르게 사용할 수 없도록 만드는 것

DDoS : (Distributed Denial of Service)분산 서비스 거부 공격/ 분산 서비스 방해 공격이라는 뜻으로, DoS의 또 다른 형태이며, 인터넷에 연결된 일련의 시스템들을 이용해 단일 사이트에 대한 플러드 공격을 시도하는 것

DRDoS : (Distributed Reflection Dos)분산 반사 서비스 거부 공격, 분산 반사 서비스 방해 공격이라는 뜻으로, 공격자가 IP주소를 공격받는 시스템의 IP주소로 조작하고, 요청을 보낸다. 이에대한 응답이 시스템 자신에게 반복하고 이를 고속으로 처리하는 방식으로 공격하는 DoS이다.

5-(2) 내가 생각하는 DoS, DDoS, DRDoS의 개념  
 세가지 모두 특정한 대상을 향하여 악의적인 공격을 하는 것인데, 공통점으로는 공격받는 대상의 리소스나 트래픽을 초과하는 정도의 요청을 함으로써 그를 처리하는 네트워크, 서버 장비들을 과부화시키는 것이다.

처음에는 개인의 pc에서 해당하는 서버로 악의적인 패킷을 보내는 방식의 DoS형태였지만, 방어법으로 공격자의 주소를 확인하여 차단하는 방법이 있었다. 따라서 그 이후에 나온 것이 좀비PC를 이용한 DDoS방식이다. 공격자가 여러 PC를 감염시켜 모든 PC에서 DoS공격을 실행하므로써 모든 공격을 차단하기 힘들어지기 때문이다.

하지만 이러한 DDoS조차 좀비PC를 조사하면 공격자의 위치를 알 수 있었기에, DRDoS라는 공격 방법이 생겨났다. PC를 감염시키지 않고 다른 PC를 반사체로 이용하여 공격한다. 이 방식은 공격자의 위치를 알기 힘들고, 공격의 규모를 수십배 이상으로 키워서 공격하기에 막기가 힘들다.

5-(3) 나라면 어떤 상황에서 쓸지? -> 어떻게 막아야 할지.

일단 원리 자체는 의외로 간단한데, 처음 DoS공격은 F5를 계속 눌러서 응답을 엄청 늘리면 되고, DDoS의 경우 여러 사람이 짜거나, 좀비PC를 이용해 많은 PC에서 DoS를 일으키는 것, DRDoS의 경우 공격자가 안보이게 반사판을 이용하여 간접적으로 공격하는 방법이다.

실제 사례로는 경인 대첩을 들 수 있는데, 내가 알기로는 사이버 한일전 정도로 생각하면 되고, 2010년에 국내 커뮤니티 사이트의 인원들이 힘을 합쳐 일본 사이트 몇 군데를 DoS공격으로 박살낸 사건이라고 알고 있다. 몇 천명이 같은 시간에 수많은 요청을 쏟아내니까 서버는 견디지 못하고 실제로 데이터를 복구할 수 없을 정도 파괴됬다고 들었다.

그렇다면 이 공격을 막기 위한 방법은 무엇일까?  
  
DoS의 경우에는 개인의 PC에서 공격하는 방식이다 보니 이상한 트래픽이 감지되었을 때에 해당하는 트래픽을 주는 PC의 접근을 차단하면 되서 막기가 쉽다.   
 하지만 DDoS단계 부터는 많은 수의 접근이 대역망과 트래픽을 초과하기에 대처가 쉽지않는 것이 정석이다, 따라서 이 부분에서는 공격의 시작 기점이 되는 좀비PC가 되는 것부터 막는 것이 효과적이며, 서버, 컴퓨터 보안 업체를 이용하여 미리 예방을 해놓는 것이 필요하다.

**6. Firewall, Web-Firewall**  
  
6-(1). Firewall, Web-Firewall의 개념  
 Firewall : 방화벽(Firewall)은 가장 오래된 정보보호시스템으로 \*TCP/IP 레벨에서 IP/Port 신뢰 여부를 기준으로 구분해주는 접근 제어 기반의 보안 솔루션이다.

Web-Firewall : 웹 서버만을 보호하기 위해 설계된 정보 보호 시스템입니다.

웹 해킹 시도로부터 홈페이지 서비스를 전문적으로 방어하기 위해 HTTP 트래픽을 검사해 보호 대상 홈페이지 서버로 유입되는 해킹을 차단하는 정보 보호 시스템 이다.

6-(2) 두가지 개념의 차이점

기본적으로 외부의 악의적인 공격을 탐지하거나 차단하는 기능을 수행하지만, 방화벽은 OSI 7계층의 네트워크 계층을 , 웹 방화벽은 응용계층을 방어한다는 점이 다르다.  
 처음에는 네트워크 계층으로 여러 공격을 탐지하고 차단했지만, 네트워크 계층만으로는 암호화된 내용을 복호화할 수 없었기에 7계층 응용계층의 다양한 웹 공격을 막을 수가 없었기에 별도의 웹 공격 차단 솔루션이 필요했고, 이것이 web-firewall이다.  
  
6-(3) 현업에서 어떻게 쓰일지  
 일단 일반 Firewall은 외부 방화벽과 내부 방화벽으로 나뉘는데 이전에 설명한 DMZ의 개념에서 나왔었다. 외부의 요청에 대해 내부에서 드러내야 할 데이터는 DMZ에 드러내고 내부의 IP주소를 보호하며, 여러 보안 정책을 실현, 바이러스 검사를 하는 역할을 맡는다.

그에 비해 Web-Firewall은 패킷을 목적지 IP에 보내는 방식이 아닌 수 많은 요청을 가함으로써 HTTP서버 과다 접속, 서비스 부하로 인한 장애를 발생 시키므로, 요청에 여러 제약을 두어 다른 방식으로의 요청을 차단하는 방식으로 많이 사용된다.  
  
 **7. UPS**

7-(1) UPS의 개념  
무정전 전원 장치(UPS)는 상용 전원에서 일어날 수 있는 전원 장애를 극복하여 좋은 품질의 안정된 교류 전력을 공급하는 장치이다.

7-(2) 내가 생각하는 UPS의 개념   
평상시에 전기를 저장해 놓았다가 비상시(정전시)에 전원이 끊기면 가동중인 기계나 시스템이 문제없이 돌아가게끔 전원을 공급해주는 일종의 비상용 베터리라고 생각한다.

가끔보면 전자제품에도 마이크로5핀 연결부분이 있음에도 건전지가 따로 들어가는 제품이 있듯이, 평소에는 일반 전기로 비상시에는 전기를 저장해 놓은 축전기로 시스템을 돌리는 장치라고 생각한다.

7-(3) 현업에서 어떻게 쓰일지

현업에서는 연구소, 금융업, 대기업, 공장 등등 매우 많은 곳에서 필수적으로 갖추고 있을 것 같다. 우리나라 같은 경우 겨울에는 한파가 여름에는 열대의 기후를 가지고 있기에 전기줄이나 여러 장비들에 문제가 가해질 수 있고, 더욱이 우리나라는 나라가 좁아서 여러 산업들이 한 곳에 뭉쳐있고, 유기적으로 연결되있는 것들이 많다. 따라서 일부 지역의 정전이라도 연쇄반응이 크게 올 수 있는데, 이럴 때 평상시의 전력을 모아놨다가 비상시에 쓰는 방법은 필수적이며 당연하다고 생각한다.

**8. NMS**

8-(1) NMS의 개념  
 NMS : (Network Management System) 네트워크 상의 모든 장비들을 관리할 수 있는 중앙 감시 체계를 의미한다.  
  
 8-(2) NMS가 하는 역할, 기능  
 장애 관리 : 장애 발생시 관리자에게 신속하게 알리고 해결방안을 제시하는 기능

구성 관리 : 네트워크 통신망을 구성하고 있는 통신 자원에 대해서 물리적 위치, 구성 정보, 장치 수량, 예비물품 수량등의 관리기능

보안 관리 : 통신 장비 접근 제어 및 통신망 관련 대외비 정보의 관리 기능

성능 관리 : 네트워크 통신망을 구성하고 있는 장비의 활용율, 에러율 등을 파악하고 향후 수요 예측

계정 관리 : 서비스 사용자에 대한 비용을 정하는 역할 수행

네트워크가 거대한 규모로 되감으로써 일일이 관리를 할수 없어졌기 때문에 이를 효과적으로 관리할 수 있는 시스템이 만들어졌다고 생각하면 되겠다.

주요기능에서 보듯이 어떤 장비들에 문제가 생길지, 얼마나 자주 에러가 발생하는지 등에 대한 정보를 바탕으로 관리자가 네트워크 장비를 다룸에 있어서 필요한 정보를 제공하는 역할을 하는 시스템이다.

8-(3) 현업에서 어떻게 쓰일지



위의 사진은 NMS시스템의 일부 인터페이스 부분이다.

얼마나 많은 라우터가 있고, 얼마나 실행중이며, 여러 정보가 들어있고, 자세하게 들어가 보면 실제 장애율, 트래픽관리, 포트관리 등 여러 기능이 들어가 있을 것이다.

이를 사용함으로써 실제로 관리자가 200개의 라우터를 일일이 관리 하지 않아도 시스템이 필요한 정보를 보여줌으로써 네트워크 장비를 관리하는데 아주 편리하게 쓰일 것이다.

**9. TEST**

9-(1) Test란 시스템이 완전히 통합되어 구축된 상태에서 시스템이 총체적으로 잘 작동 되는지 검사하는 것으로 검사를 통과한 소프트웨어 컴포넌트와 필요한 하드웨어를 통합한 소프트웨어 시스템 전체를 대상으로 한다. 이 과정에서 소프트웨어가 정상 운행 되는지. 하드웨어와의 결합에서 이상이 생기지 않는지를 검사하는 것이다.

9-(2) 테스트 기법

화이트 박스 테스트(코드기반) :   
 statement coverage : 모든 코드가 한번은 실행되게 입력. 이 방식은 코드의 에러여부를 파악한다.

Decision coverage : 코드의 흐름에서 모든 진행을 테스트. 이 방식은 if, for 등의 구문에서 모든 흐름의 입력을 해보는 것이다.

Condition coverage : 조건문이 있을 때 조건과 맞지않는 input을 넣어본다.

Multiple condition coverage : 조건문의 true, false에 모든 상황을 고려, 조건이 (A&B)일 경우 T,T T,F F,F F,T 같이 매개변수가 여러 개인 if문에 사용

블랙 박스 테스트(input, output 기반의 시험)

Syntax testing : 입력에 올바른 값과 올바르지 않는 값을 넣어서 테스트한다

Equivalent partitioning : 입력을 동등하게 쪼개서 각 영역의 대표값을 입력하여 테스트한다. 예시로 성적이 0~100일 때, 0~20은 10, 20~40은 30 을 넣어서 테스트한다.

Boundary testing : 입력값의 경계의 값을 넣어 테스트한다.

Decision table : 입력값의 종류를 결정하여 테스트한다.

9-(3)알파, 베타 테스트

게임 광고에서 종종 듣던 말이 있는데 바로 베타테스트, 베타테스터라는 말이었다. 이 테스트는 게임이 실제 시장에 나오기 전에 게임을 테스트하는 방식이다.

알파와 베타테스트의 차이는 실시하는 시기와 대상에서 차이가 나는데, 알파 테스트의 경우, 게임을 다 만들고 회사 내 관련자들이 실제 게임에서 프로그램이 버그없이 잘 작동하는지 테스트를 실행하는 것이고, 베타 테스트는 정시로 출시하기 전에 사용자들이 사용함에 있어 불편한 점, 버그 등을 찾는 테스트이다. 또한 베타 테스트를 한정된 인원으로 진행하면 클로즈 베타, 모든사람들에게 자유롭다면 오픈베타라고 부른다.

10-(4)내가 생각하는 테스트

실제로 작년에 안드로이드 프로그램으로 간단한 앱을 만든적이 있다. 실제로 내가 만들고 동작의 방식을 알고 있기에 내가 짯던대로 실행을 시킬때는 이상이 없었다. 하지만, 사용자들은 나와 다르다는 것을 잊었었다. 실제로 발표 때는 이거 눌러봐요 저거 눌러봐요 했을 때, 어? 이게 아닌데? 이런 생각이 많이 들었다. 조그마한 프로그램도 내 생각대로 작동시키지 않는데, 큰 프로그램에서는 얼마나 다양하게 사용할것인가. 그런 점에서 테스트는 잘 작동 되는지를 판단하는 것이 아니라, 얼마나 오류가 있는지를 살피는 것이 테스트인 것 같다.