

○ 정보 보안 기사 실기

[필기]

1. 시스템보안 (20문항 × 5점 = 100점)
2. 네트워크보안 (20문항 × 5점 = 100점)
3. 어플리케이션보안 (20문항 × 5점 = 100점)
4. 정보보안 일반 (20문항 × 5점 = 100점)
5. 정보보안관리 및 법규 (20문항 × 5점 = 100점)

[실기]

1. 단답형 (10문항 × 3점 = 30점)
2. 서술형 (3문항 × 14점 = 42점)
3. 실무형 (택2문항 × 14점 = 28점) ※ 3문제 중 2문제를 선택해서 답안작성

■ 최종 요약

sticky, Nessus, storm, Clark-Wilson, Challenge, Amplication unreachable, Stealth, Diffie-hellman, heartbleed, Poisoning

■ 정보 보안 및 법규 일반

- SAML, PMI
- 인증 / 인가
- 인증서 : x.509
- 위협(도둑)/취약점(창문,해킹,도청)/위험(발생/손실가능성)
- 정성 위험 분석 : 델파이, 시나리오, 순위결정 : **델시순**
- **위험 접근 : 기준,비정형,상세한 위험 분석,통합된 : 기비상통**
- 통제 : 일반/응용/시점(예방,탐지,교정)
- 이중 서명, 은닉 서명
- 국가사이버안전규정 : 관심, 주의, 경계, 심각 : **관주경심**
- 무결성 = $\sum \square [1\text{-위협(Threat)} * [1\text{-보안(security)}]]$
- 전자서명 : 무결,인증,부인방지,기밀,전접제어 : **인절무기부**

■ 개인 정보 보호

- 개인 정보 내부 관리 계획 : 6가지
- 개인 정보 접근 권한 관리 : 5가지(최소,인사,기록,계정,비번)
- 권한 부여 및 말소 기록 : 5년간 보관
- 기술적 보호 : 접근 통제 장치, 기록 위/변조, 암호화, 백신
- 관리적 보호 : 내부 관리 계획 수립 및 시행
- 물리적 보호 : 개인정보처리 장소 물리적 보호, 시건장치
- 개인정보유출 통지 : 항목, 시점, 경위, 피해최소화방안,구제절차
사고 담당 부서 연락처
- 표준(강제) / 지침(가이드) / 절차(세부)
- 암호화 : 주민,여권,운전,외국인,신용,계좌,바이오 : **주여외신운계바**
- 망 분리 대상 : 1일 100만명, 정보 통신 년 매출 : 100억
- 비식별화 처리 : 가명 처리 / 총계 처리 / 데이터 값 삭제
범주화 / 데이터 마스킹

■ 시스템 보안

- 보안 위한 분리 : 물리,시간,논리,암호
- 데이터보안등급 : 민감 > 기밀 > 비밀 > 대외비
- shadow : 생성경과, 변경허용, 만료기간, 만료경고
- 참조모니터 : 부정 조작 불가, 우회 불가, 검증 가능
- tcpwarpper : inet.conf, hosts.allow, hosts.deny, tcpd(대체)
- corn : 분, 시, 일, 월, 요일(0=일), 작업
- http : 301(move), 403(금지)
- Web Proxy : Paros, Burp
- Apache 정보 : Prod, min(ver), OS(ver), Full(package)
- 교차 조건 : 상호배제, 점유와대기, 비선점, 환경 대기
- 교차 해결 : 예방, 회피, 발견, 복구
- netstat : -a -p -n -e -r -i -s
- Apache DNS 역검색 제한 : HostnameLookups Off
- Spam : Procmail, Sanitizer, Inflex, SpamAssassin
- ISAC : 유사 업무 분야 별 해이나 사이버 테러 공동 대응
- TCSEC : Orange book, 미국국방부
- 오버플로 : 스택가드, 스택실드, ASLR, 안전함수, 경계 체크
- Race Condition : 임시 파일 제한, 파일 존재 여부, 링크 확인
umask 022 설정
- SNORT : alert, log, pass, activate, dynamic, drop, reject, sdrop, msg, content, offset, depth, nocase

■ 최신 보안

- OpenSSL(heartbleed) : 메모리 데이터 탈취 가능, hello
- VENOM : 가상 환경을 무시하는 운영 조작, 하이퍼바이저
- 비대면본인확인: 신분증,영상통화,현금카드,기존계좌 중 2가지
- IOT 보안 : SSDP(단순서비스검색프로토콜) 악용 DDOS
- SSDP : 모바일 기기 검색 시 DDOS 공격 악용 가능,1900port
- Agentless FDS : 별도 어플 불필요, 핀테크 최적,
웹 브라우저 단말의 인자, 인증키 단말 보관,
- blind sql : 시간 기반, 응답 기반, 대체 체널

■ 네트워크 일반

- 계층별 전송 단위 : 비프패세메
- Data Link : L2TP, PPTP, L2F, MAC/LLC
- Network : Router, IPsec
- Transport : SSL/TLS,
- Session :
- Presation :
- Application : SNMP,
- VLAN : Tagging
- **FTP:20, Telnet:23, SMTP:25, DNS:53, POP3:?**
- **SNMP:161, IMAP:143, IMAPv3:220**
- STP : Spanning Tree Protocol
- SNMP : polling(Ser), Trap(Cli), Syslog(실시간 이벤트 전송)
- 스니핑 : 제방, ARP Spoofing/Redirect, ICMP Redirect, Mirror
- ARP : arp -s 123.123.34.212 A : 정적 ARP 설정
- **IDS : 오탐(flase positive), 미탐(flase negative)**
- CIDR : IP절약, 라우팅 테이블 감소
- SLC : 무선 인터넷 인증서, 짧은 유효 기간
- 커버로스 : KDC/AS/TGS, 타임스탬프
- ICMP : Source Quench
- **IP 역추적 : T호내, IPIH**
- IGMP : TTL=1, Report/Query Message
- DNS : TCP - Zone transfer, 메시지>512byte, 로드 높을 때
SNA-MCP, SOA/NS/A/MX/CNAME/PTR
- Resolving DNS / Cache DNS / Authoritative DNS
- SSL : Record(MAC,인증서 교환), BEAST/CRIME
- DHCP : range dynamic-bootp from-IP to-IP
- FTP : ServerType standalone(inetd)
- ARP Redirect : 라우터 가장
- ICMP Redirect : 라우터 경로 재설정

■ find

- find / -perm 2000 -print
- find / -user root -perm 4000 print
- find / -m(a,c)time -10

■ tcpdump

- tcpdump -i eth0 -nn "tcp src/desc port 80"
- tcpdump -i eth0 -nn "tcp and host 123.212.323.221"
- tcpdump -i eth0 -nn "tcp and src host 123.212.323.221"
- tcpdump -i eth0 -nn "tcp port 80 and host 12.21.32.21"

■ Router

PW 암호화 기능 활성화	Router#conf t Router(config)# <u>service password-encryption</u>
PW 암호화 저장	Router#conf t Router(config)# <u>enable secret qlalf123</u>
PW 평문 저장	Router#conf t Router(config)# <u>service password</u>
텔넷 포트 암호 설정	Router#conf t Router(config)# <u>line vty 0</u> Router(config-line)# <u>password qlalf123</u> Router(config-line)# <u>exit</u>
콘솔 암호 설정	Router#conf t Router(config)# <u>line con 0</u> Router(config-line)# <u>password qlalf123</u> Router(config-line)# <u>exit</u>
라우터 이름 설정	#conf t Router(config)# <u>hostname rrrname</u>
e0 I/F에 IP 지정	#conf t Router(config)# interface Ethernet - Router(config-if)# ip address 192.111.111.111 255.255.255.0
black hole 설정	#conf t Router(config)# interface null 0 Router(config-if)# no ip inreachables Router(config-if)# exit Router(config)# ip route 10.0.0.0 255.0.0.0 null 0 : 출발지 10.0.0.0 의 패킷을 static routing 하여 폐기

■ 공격 방식

- Tiny Fragment/ Fragment Overlap / Tear Drop 차이 구분
- SMURF : Directed Broadcast, Echo Request/Reply, Amp
- DDos : Trinoo, TFn, Stacheldracht, Sraft, Trinity
- DDos 대응 : 라우터, i/egress, black hole, ant-ddos
- 스니핑 탐지: ARP,ICMP,Decoy,DNS(pin-sweep),ARP Watch
- no ip source-route, no ip unreachable

■ HTTP 공격 방식

- Slow HTTP Header / Slow HTTP Read / Slow HTTP Post
- HULK 공격, Hash DOS,

■ OWASP Top 10

- 인젝션, 인증/세션 관리 취약점, XSS, 취약 객체 직/참
- 보안 설정 오류, 민감 데이터 노출, 기능 수준 AC 누락
- CSRF, 알려진 취약점 컴포넌트, 미검증 리다/포워

■ 방화벽

- 싱글 홉드 게이트웨이, 듀얼 홉드 게이트웨이,
- 스크린드 서브넷 게이트웨이, 스크린드 호스트 게이트웨이

■ 암호 일반

- 스트림 암호 : one time pad
- 블록 암호 : 전치/지환
- 암호 공격 : COA / KPA / CPA / CCA
- **암호 모드 : ECB / CBC / CFB / OFB / CTR**
- ECB : 고속, 간단, 병렬, 한개 블록 해독 -> 전체 해독
- CBC : ECB 문제점 개선, 복호화시 병렬, IPsec 기밀, 3DES, Kerb
보안성 제일 높음, 병렬 처리 불가
- CFB : 블록 기반의 스트림, 복호화 병렬
- OFB : 블록 기반의 스트림, 병렬 불가, 패딩 불필요
- CTR : 스트림 암호, 병렬, CTR + 1, 빠른 속도 장점, 고속 요구

■ 암호 알고리즘

- 타원곡선 : 이산대수, RSA 대안, 적은 Bit,스마트 카드
- RSA : 소인수 분해
- SHA : 1/2/3, 512bit -> 160bit
- Hash : 약/강 일방향, 충돌 저항성, SHA, MD5, Haval
- MD5 : 512bit -> 128bit,
- DES : 16회전, 64bit, 8개의 s-box, 16라운드 feistel
- DSS : 전자서명 표준, Elgamal 기본, DSA가 핵심 알고리즘
- Diffie-hellman : 이산대수, 기함의, 중간자 공격, (**G^A % P**), **A**
- 키분배 : n명 / n개의 키

■ Snort / IDS

- drop : 패킷 차단, 로그 남김, sdrop(로그 x)
- reject : 패킷 차단, rst / unreachable 메시지 전송

■ IPsec

- **전송(페이로드, 중단) / 터널(헤더+페이로드, 망-게이트)**
- **AH(무결성,송신자) / ESP(무결성, 송신자, 암호화)**
- IKE : main mode(DES/3DES) / Aggressive mode

■ 인증

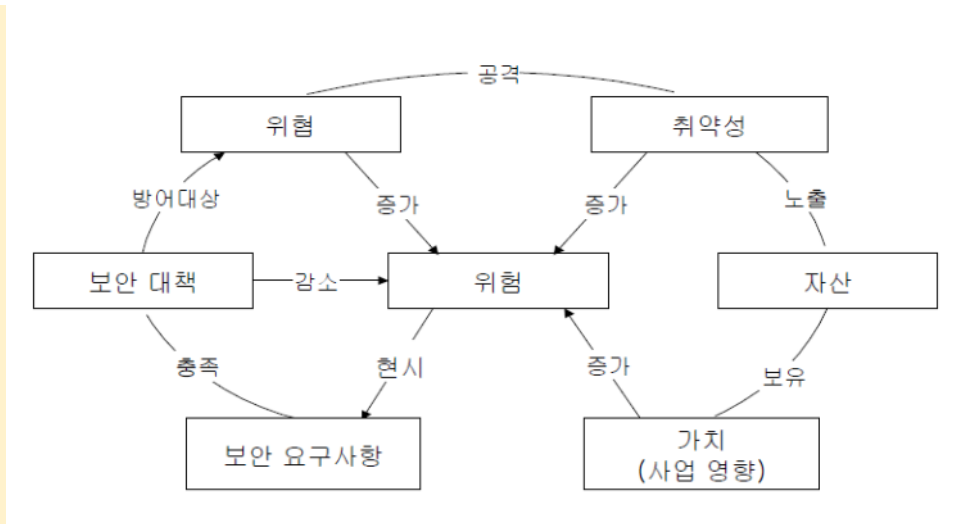
- 생체인식 : FRR / FAR / CER / FER
- 사용자 인증 유형 : 지소존형
- OTP : ETSC

■ 시스템 기록













- 최근 로그인 기록 : lastlog
- 실패한 로그인 기록 : loginlog
- utmp(w/who,whodo,finger) / wtmp(last) / btmp(lastb)
- lastlog : 계정별 가장 최근 로그인 기록,lastlog(cmd)
- dmesg(driver), message(콘솔 출력), boot.log(부팅)
- acct / pacct : 시스템(명령) 사용 내역, lastcomm, acctcom

> 정보보안관리 및 법규

일반		
<div>!</div> 보안의 3요소 정보보호 3대 요소	○ 기밀성, 무결성, 가용성 ○ 정보 자산의 중요도 파악 위해, 머저 자산을 안전하게 보호하기 위해 요구되는 일반적인 특성	[기무가]
<div>!</div> 정보보호의 5대 목표	○ 기밀성, 무결성, 가용성, 인증, 부인방지	
기밀성	○ 오직 인가된 사람, 프로세스, 시스템만이 알 필요성에 근거하여 시스템에 접근해야 한다는 원칙	
무결성	○ 네트워크를 통해 송/수신되는 정보가 변조/변경/추가/삭제되지 않도록 보호되어야 하는 성질	
가용성	○ 정당한 사용자가 필요할 때 지체없이 사용할 수 있는 성질	
정보보호정책	○ 최소한 중요 자산이 무엇이며, 어떤 특성을 만족해야하는지 목적 을 제시 ○ 조직에 미치는 영향을 고려, 업무, 서비스, 조직, 사람, 자산 등 정책의 적용 범위 정의 ○ 경영진, 정보보호조직, 일반 직원의 책임을 명확하게 정의 ○ 최고 경영자의 의지를 확인할 수 있도록 문서로 승인	
정보 보호 계획 수립	○ 정보 보호 정책 수립 ○ 표준, 절차, 기준선, 지침	표절기침
정보 보호 정책	○ 정보 보호에 대한 최 상의 규정, 최고 경영자의 승인 및 배포, 모든 구성원이 숙지 및 준수 필요 ○ 정보 보호 활동의 기본적인 방향과 근거를 제시, 조직 구성원의 정보보호에 대한 책임과 역할 정의	
사업 영향 분석		
BIA	○ 업무 중단이 비즈니스에 미치는 정성/정량/기능적 분석 ○ 발생 가능 모든 재해 고려, 잠재적인 손실을 추정, 재난을 분류, 우선 순이의 부여, 실행 가능 대안 개발 ○ 컴퓨터/통신 서비스 심각한 중단 사태에 따라 각 사업 단위가 받게 될 재정적 손실 영향 파악	
위험 관리		
정보 자산 그룹핑	○ 기밀성, 무결성, 가용성 평가에 기초, 자산 유형/보안 특성/중요도가 같은 것을 묶어 공통 자산 그룹 형성 ○ 비슷한 유형의 정보 자산을 그룹핑 함으로써 위험 분석 비용 절감 및 관리 효율 기대	
위험 관리	○ 조직의 당면 위험을 인식, 조직에 적절한 수준으로 통제하는 것, ○ 측정된 조직의 위험, 보안 대책의 비용 및 효과 등을 비교, 정보 보호 목적과 일치하도록 대책을 도출 ○ 측정 및 평가된 위험을 줄이거나 제거하기 위한 과정으로 위험을 일정 수준으로 유지/관리하는 것	
<div>!</div> 위험 평가 기본 요소	○ 자산, 위협, 취약점	[자위취]
<div>!</div> 자산 중요도 평가 <div>!</div> [자산 목록]	○ 자산 목록 : 자산의 중요도 평가를 위해 작성	
<div>!</div> 자산 중요도 평가 <div>!</div> [자산 분석]	○ 자산의 평가, 관리 용이를 위해 재분류를 위해 실시 ○ 자석의 특성을 고려, 사용 용도 / 피해 규모 / 사용 환경 등	
위험 관리 절차	○ 자산 식별 > 위험 분석 > 취약성 분석 > 위험 평가	[자위취평]
<div>!</div> 위험 평가	○ 위협의 종류, 위협의 영향, 위협의 발생 가능성을 등을 평가하는 과정	
<div>!</div> 위험 관리	○ 정보 보호의 위협을 인식, 적절한 비용 내 필요한 통제 방안 선택하여 위협을 적절히 통제하는 과정	
<div>!</div> 위험관리 계획	○ 선택된 통제의 목적과 통제 방안의 무엇인지, 선택한 이유 등을 문서로 정리한 것 ○ 선택된 통제 방안 등을 누가 언제 어디서, 무엇을 어떻게 적용할 것인지 정리	



!	자산	<ul style="list-style-type: none"> ○ 조직이 보호해야 할 대상 : 정보, 하드웨어, 소프트웨어, 시설, 인력 ○ 위험을 보유하고 있는 대상, 위험이 발생할 경우 피해 규모 측정위해 반드시 포함되는 요소 	
!	위협 (Threat)	<ul style="list-style-type: none"> ○ <u>자산의 손실을 초래할 수 있는 원치않는 사건의 잠재적 원인</u> - 의도적 : 도청, 정보변조, 시스템 해킹, 악성 코드, 절도, 테러 - 사고 : 실수, 누락, 파일 삭제, 부정확한 라우팅, 물리적 사고 - 환경적 요인 : 지진, 번개, 홍수, 누수, 화재 ○ 손실이나 손상의 원인이 될 가능성이 제공하는 환경의 집합, 보안에 해를 끼치는 행동이나 사건 ○ 외부에서 발생, 자산에 손실을 일으키는 요소로서 발생 가능성으로 측정하기도 함. 	
!	취약점 (Vulnerability)	<ul style="list-style-type: none"> ○ 자산의 잠재적 속성으로 위협의 이용 대상이 되는 것 ○ 보안 대책의 미비로 정의하기도 함. ○ 자산에 취약성이 없다면 위협이 발생해도 손실이 나타나지 않을 ○ 자산과 위협을 연결하는 개념 ○ 자산의 잠재적 속성이나 처한 환경으로 위협의 이용대상, 관리적/물리적/기술적 약점 의미 ○ 자산의 내에 존재하는 약점으로 위협은 취약점을 활용하여 위험을 발생 	
	위험 (Risk)	<ul style="list-style-type: none"> ○ 위협이 취약점을 이용하여 조직의 자산에 손실/피해를 가져올 가능성 ○ 비정상적인 일이 발생할 수 있는 가능성 ○ 예상되는 위협에 의해 자산에 발생할 가능성이 있는 손실의 기대치 ○ 자산의 가치 및 취약점, 위협 요소의 능력, 보호 대책의 효과 등에 의해 영향 	
	위협/취약점/위험	<p>나보석씨의 금고방 점포는 도둑을 맞을 가능성이 있습니다. 점포의 문, 창문, 천장 등을 통해 도둑이 침입을 할 수 있습니다. 여기서 물건을 훔치로 오는 도둑놈을 Threat(위협) 라고 볼수 있습니다. 도둑놈은 금고방의 각종 보석들을 훔쳐갈 잠재적인 요인입니다. 이 도둑놈은 점포의 문, 창문, 천장 등의 빈틈을 찾습니다. 그 빈틈을 통해 점포에 들어가고 보석을 훔칩니다. 여기서 빈틈이 Vulnerability(취약점) 입니다. 이처럼 보석들을 훔쳐갈 잠재적인 요인인 도둑놈이 빈틈을 통해 보석을 훔쳐갈 가능성을 Risk(위험)이라고 합니다. 만약 빈틈이 존재하지 않는다면 도둑놈은 침입할수 없고 보석을 훔쳐갈 가능성은 없습니다.</p> <p>Vulnerability(취약점)이 존재하지 않는다면 Threat(위협)은 Risk(위험)이 될수 없다고 할 수 있습니다.</p> <p>Threat(위협) - 도둑놈, 보호해야할 대상을 훼손하거나 훔쳐갈 잠재적인 요인 Vulnerability(취약점) - 빈틈, 대상을 보호하는것이 가진 흠 Risk(위험) - 도둑놈이 빈틈을 노려 보석을 훔침, Threat(위협)이 Vulnerability(취약점)를 이용하여 보호할 대상을 훼손하거나 훔쳐갈 가능성</p>	
!	위험 대응	<ul style="list-style-type: none"> ○ 수용 / 감소(완화) / 회피 / 전가 : ATMA ○ 수용 : 위험을 받아 들이고, 잠재적 손실 비용을 감수 : ○ 감소 : 위험 감소 시킬 수 있는 대책을 수립, 구현하는 것 : 암호화, 보안 솔루션 등 ○ 회피 : 위험이 존재하는 프로세스를 취소, 위험 사업의 포기 : 온라인 절차 포기, 오프라인 전환 ○ 전가 : 보험, 외주 등으로 제 3자에게 위험을 이전하는 방법 	[수감회전]
!	위험 관리 과정	<ul style="list-style-type: none"> ○ 위험 분석 > 위험 평가 > 대책 설정 ○ 위험 분석 : 통제되거나 받아 들어들여질 필요가 있는 위험을 확인 	[분평대]

	<ul style="list-style-type: none"> ○ 위험 평가 : 적절하고, 적당한 보안 대책의 수립을 위해 시스템 및 그 자산이 노출된 위험을 평가/식별 ○ 대책 설정 : 허용 가능 수준으로 평가된 위험을 줄이기 위해 적절하고 적당한 대책을 식별 및 선정 	
위험 분석 방법론		
 위험 분석 방법론 분류		
 정량적 위험 분석	○	
 정성적 위험 분석	○ 델파이법, 시나리오법, 순위결정법	
		
 [정성적 위험 분석] 델파이법	<ul style="list-style-type: none"> ○ 각 분야의 전문가 그룹을 구성, 위험을 분석 및 평가, 정보 시스템의 직면한 다양한 위험 및 취약성 토론 ○ 위험 분석을 짧은 시간에 도출가능하며 비용 및 시간 절약 ○ 위험 추정의 정확도가 낮음 	
 [정성적 위험 분석] 시나리오법	<ul style="list-style-type: none"> ○ 어떤 사건도 기대대로 발생하지 않는다는 사실에 근거, 일정 조건하의 위험 발생 가능한 결과를 추정 ○ 적은 정보를 가지고 전반적인 가능성을 추론 ○ 정확도, 완성도, 이용 기술 등의 수준이 낮음 	
 [정성적 위험 분석] 순위결정법	<ul style="list-style-type: none"> ○ 각 위험을 상호 비교, 각종 위험 요인의 우선 순위 도출하는 방법 ○ 위험 분석에 소모되는 시간과 분석 자원의 양이 적다는 장점 ○ 위험 추정의 정확도가 낮은 단점 	
 위험 분석 접근법	○ 기준 접근법, 비정형화된 접근법, 상세한 위험 분석, 통합된(복합) 접근법	[기비상통]
 기준 접근법	○ 일반적 수준에서 통제, 기준 문서, 실무 규약 등, 체크리스트	
 비정형화된 접근법	○ 분석을 수행하는 개개인의 지식과 전문성에 기반한 접근	
 상세한 위험 분석	○ 구조화된 접근법, IT 시스템에 대한 상세한 위험 평가, 시간 및 비용 과도하게 소요	
 통합된 접근법	○ 고위험 영역은 상세한 위험 분석, 다른 영역은 베이스라인 접근법 활용, 고위험 영역 식별이 관건	
손실 추정		
관련 문제	<ul style="list-style-type: none"> ○ B회사는 5년에 1번 화재 발생, 화재 발생 시 평균적으로 40억의 회사 시설 중 30%의 손실 발생(564p) ○ 자산 가치, 노출 계수, 단일예상손실액, 연간 발생률, 연간예상손실액 산출 	
노출 계수	○ 실현된 위험, 침해 발생 시 조직이 입게되는 손실 비율 : 회사 시설에 30%의 피해 발생	
예상 손실액(SLE)	○ 자산가치(AV) X 노출 계수(EF)	
연간예상손실(ALE)	○ 예상 손실액(SLE) X 연간 발생률(ARO)	
연간 발생률(ARO)	○ 1년 안에 성공적으로 발생할 확률	
자산		
자산 목록	○ 자산의 중요도 평가를 위한 기초 자료 확보, 자산의 누락없이 최대한 자세하게 나열	
자산 분석	○ 자산을 평가하고 관리하기 용이하게 재분류, 자산의 특성을 고려, 사용 용도, 피해 규모, 사용환경 고려	
노출 계수	<ul style="list-style-type: none"> ○ 자산의 가치에 대한 손실이나 영향을 크기를 측정한 값 ○ 어떤 위험 사건으로 부터 발생하는 자산 가치의 손실을 백분율로 나타내는 수 ○ 어떤 자산에 대한 실현된 위험으로 인해 침해가 발생할 때 조직이 입게 되는 손실 비율 	
재해 복구		
재해 복구 방식	○ 미러(AA,RTO:0) / 핫(AS:4) / 웜(중요 자원 부분 보유) / 콜드(장애시 조달) / 상호 지원 계약(동종 업계)	
RTO		
RPO		
ISMS		
ISMS	○ 정책 및 조직 구성, 위험 관리, 대책 구현, 사후 관리 등의 관리 과정 정리	
5단계 관리 과정	○ 정보보호 정책 수립 및 범위 설정 > 경영진 책임/ 조직 구성 > 위험 관리 > 정보보호대책 구현 > 사후관리	
통제 방식		

통제 방식	○ 일반 통제, 응용 통제, 시점별 통제	[일용시]
일반 통제	○ 정보시스템 SDLC 환경에 대한 통제, 모응 응용시스템에 공통으로 적용 가능한 것, IT조직 관리, 직무 분리, 시스템 개발, 물리/논리적 보안, 백업 및 비상 계획, 하드웨어 통제 등	
응용 통제	○ 입력 통제 / 처리 통제 / 출력 통제	
시점별 통제	○ 예방 통제 / 탐지 통제 / 교정 통제	[예탐교]
정통방법		
! 기술적 보호 조치	<ul style="list-style-type: none"> ○ <u>내부관리계획 수립</u> ○ <u>침입 차단 시스템 등 접근 통제 장치 설치 및 운영</u> ○ <u>접속 기록의 위/변조 방지를 위한 조치</u> ○ <u>개인 정보의 안전한 저장/전송을 위한 암호화 기술</u> ○ <u>백인 소프트웨어 설치, 운영 등 바이러스 침해 방지 조치</u> ○ <u>개인 정보 보호를 위한 필요한 보호 조치</u> 	내접기압백개
관리적 보호 조치		
개인정보보호법		
개인 정보 수집 시 동의 항목		
! 제3자 제공 시 고지 및 동의 항목	<ul style="list-style-type: none"> ○ 개인정보를 <u>제공 받는자</u> ○ 개인정보를 제공 받는자의 <u>개인 정보 이용 목적</u> ○ 제공하는 개인 정보의 <u>항목</u> ○ 개인 정보를 제공 받는 자의 개인 정보 <u>보유 및 이용 기간</u> ○ 동의를 거부할 권리가 있다는 사실, 동의 거부에 따른 불이익이 있는 경우 그 <u>불이익</u> 내용 	[자목항기불]
! 유출 통지 기준	<ul style="list-style-type: none"> ○ 1만명 이상 정보 주체 개인 정보 유출된 경우 통지 및 조치 결과를 행자부/전문 기관에 신고 ○ 5만명 이상 정보 주체 개인 정보 유출된 경우 통지, 홈페이지에 7일간 고지 ○ 유출 사고 발생 확인 후 5일 이내 정보 주체에게 알려야 함. 	
! 개인 정보 유출 통지 사항	<ul style="list-style-type: none"> ○ 유출된 개인 정보 항목 ○ 유출된 시점 및 그 경위 ○ 유출로 인해 발생할 수 있는 피해를 최소화하기 위해 정보 주체가 할 수 있는 방법 ○ 개인 정보 처리자의 대응 조치 및 피해 구제 절차 ○ 정보 주체에게 피해가 발생할 경우 신고 등을 접수할 수 있는 담당 부서 및 연락처 	
내부 관리 계획	○ 취합하는 개인 정보가 분실/도난/누출/변조, 훼손되지 않도록 안전성 확보하기 위해 개인 정보 보호 활동에 대한 조직 내부의 개인 정보 관리 계획을 수립	
내부 관리 계획 주요 사항	<ul style="list-style-type: none"> ○ 개인 정보 보호 책임자 지정에 관한 사항 ○ 개인 정보 보호 책임자 및 개인 정보 취급자의 역할 및 책임에 관한 사항 ○ 개인 정보의 안전성 확보에 필요한 조치에 관한 사항 ○ 개인 정보 취급자에 대한 교육에 관한 사항 ○ 개인 정보 위탁 경의 수탁자에 대한 관리/감독에 관한 사항 ○ 개인 정보 보호를 위해 필요한 사항 	
개인정보처리자	○ 업무를 목적으로 개인 정보 파일을 운용하기 위해 개인 정보를 처리하는 공공기관,법인,단체,개인 의미	
개인정보처리시스템	○ 개인 정보를 처리할 수 있도록 체계적으로 구성한 데이터베이스시스템	
개인 정보 파일	○ 개인 정보를 쉽게 검색할 수 있도록 일정한 규칙으로 체계적 배열, 구성한 개인 정보 집합물	
개인 정보 시스템 권한 부여 기록 유지	○ 권한 부여, 변경, 말소에 대한 기록은 최소 3년 간 보관	
개인 정보 처리 시스템 접속 기록	○ 접속한 기록은 최소 6개월 보관	
개인 정보 암호화 대상 정보	<ul style="list-style-type: none"> ○ 고유식별정보, 비밀 번호, 바이오 정보는 암호화 보관 ○ 비밀 번호는 단방향 암호화 저장 	고비바
! 개인정보 안정성 확보 조치 기준	<ul style="list-style-type: none"> ○ <u>관리적 조치 : 내부관리계획 수립 및 시행</u> ○ <u>기술적 조치</u> 	

	<ul style="list-style-type: none"> - 접근 통제 및 접근 권한 제한 - 개인 정보의 안전한 전송/저장을 위한 암호화 기술 적용 및 이에 상응하는 조치 - 침해 사고 발생에 대한 위한 접속 기록의 보관 및 위/변조 방지 조치 - 개인 정보 보안 프로그램 설치 및 갱신 <p>○ 물리적 조치</p> <ul style="list-style-type: none"> - 개인 정보의 안전한 보관을 위한 잠금 장치등의 물리적 조치 	
! 접근 권한 관리 기준	<p>○ 업무 수행에 필요한 최소한의 범위로 차등 부여</p> <p>○ 전보/퇴직 등 인사 이동 발생 시 접근 권한의 변경 또는 말소</p> <p>○ 권한의 부여, 말소, 변경에 대한 기록은 최소 3년간 보관</p> <p>○ 계정을 개인 정보 취급자 별 1인 1계정 발급</p> <p>○ 안전한 비밀 번호 체계 설정</p>	
전자서명법		
! 인증	○ 전자서명생성정보가 가입자에게 유일하게 속한다는 사실을 확인, 이를 증명하는 행위	
! 인증서	○ 전자서명생성정보가 가입자에게 유일하게 속한다는 사실을 확인, 이를 증명하는 전자적 정보	
! 전자서명	○ 서명자를 확인, 전자 문서에 서명하였음을 증명하는 전자 문서에 첨부/논리적 결합된 정보	
! 전자서명 효과	○ 무결성, 인증, 부인방지, 기밀성, 접근제어	무인기부접
개인정보		
개인정보분쟁 조정위원회		
! 개인 정보 영향 평가 (PIA)	<p>○ 개인 정보가 수집/활용되는 사업 추진 시 오남용으로 인한 프라이버시 침해 위험 존재 여부 조사,예측</p> <p>○ 평가 대상 시스템 활용에 따른 잠재적 위험을 평가, 개인 정보 침해에 따른 피해를 줄일 수 있는 지 미리 검토, 반영하는 것</p>	
! 개인 정보 영향 평가 대상	<p>○ 구축/운용, 변경하려는 개인 정보 파일에 민감정보/개인식별정보가 5만건 이상인 경우</p> <p>○ 구축/운용하고 있는 개인정보파일 내/외부와 연계하려는 경우, 연계 결과과 50만건 이상인 경우</p> <p>○ 구축/운용, 변경하려는 개인정보파일이 100만명 이상을 가지는 경우</p> <p>○ 개인 정보 영향 평가 받은 후 변경된 부분</p>	
내부망	○ 물리적 망분리, 접근통제시스템 등을 통해 인터넷 구간에서 접근이 통제 /차단되는 구간	
! 위험도 분석	○ 개인정보처리시스템에 적용되는 개인정보보호를 위한 수단과 유출 시 정보 주체의 권리를 해할 가능성 과 그 위험 정도를 분석하는 행위	
기타		
침해 사고 대응 절차	○ 사고전 준비 과정 > 사고 탐지 > 초기 대응 > 대응 전략 체계화 > 사고 조사 > 보고서 > 해결	
ISAC (정보공유/분석센터)	<p>○ 유사 업무 분야 별 해킹, 바이러스 등 사이버테러 와 정보 침해 대해 효과적 공동 대응 체계</p> <p>○ 회원가 간 정보 보호 공동 대처, 운영 비용 절감</p>	

> 네트워크 보안

일반		
	ICMP, DMZ, CIDR, PCRE, Subnet, DNS 재귀쿼리, 포트번호(p287) CIDR 표기법 확실한 이해 필요	
OSI 7 계층		
데이터 전송 단위	○ 비트 스트림(1, 물) > 프레임(2, 데) > 패킷(3, 네) > 세그먼트(4, 트) > 메시지(5이상)	[비프패세메]
1계층 물리 계층	○ RS-232C, I430, RS-449, 리피터	
2계층 데이터 링크 계층	○ 물리 계층에서 전송한 비트에 대한 동기 및 식별, 원할한 데이터 전송 관리 ○ MAC + LLC(Logical Link Control) ○ 회선제어, 흐름제어(정지-대기, 슬라이딩), 오류제어(FEC, ARQ) ○ ARP, RARP, PPP, SLIP, 브리지, MAC, CSMA/CD, VLAN ○ L2TP, L2F, PPTP	
3계층 네트워크 계층	○ 데이터가 목적지까지 올바르게 도달할 수 있도록 경로 선택 및 라우팅 기능 수행 ○ 논리적 링크 설정, 상위 계층 데이터를 작은 크기의 패킷으로 분할하여 전송하는 역할 ○ IP, ICMP, IGMP, RIP, OSPF : 라우팅/포워딩, 라우터	
4계층 트랜스포트 계층	○ 이 계층을 기준으로 하위 계층 / 상위 계층으로 구분 ○ 두 종단 간제어를 담당 ○ TCP, UDP, SSL	
5계층 세션 계층	○ 세션 : 전송 모드(전이중/반이중), NFS, SQL, RPC	
6계층 표현 계층	○ 통신 장치의 데이터 표현 방식, 상이한 부호 체계 간의 변화, 데이터 압축 및 해제 ○ 암호화 / 복호화, 인코딩 / 디코딩 담당 ○ ASCII, MPGE, ipg, 암호화 등	
7계층 응용 계층	○	
2계층		
회선제어	○ 점대점, 멀티포인트 구성방식, 단방향, 반이중 및 양방향 등 전송링크에 대한 제어 규범 ○ 복수의 통신 회선 사이에서 데이터의 상호 전송을 제어하는 것	
흐름제어	○ 수신 장치의 용량 이상으로 데이터가 넘치지 않도록 송신 장치 제어 ○ 정지-대기 기법(Stop-and-wait), 슬라이딩 윈도우 기법	
오류제어	○ 다양한 원인으로 전송된 데이터가 발생할 수 있는 오류 해결을 위한 제어 방식 ○ 오류 정정 부호에 의해 오류를 정정하는 순방향 오류 정정(FEC) ○ 오류 검출 부호를 사용, 재송신하는 자동 재송신 요구(ARQ)	
하위 구성 계층	○ MAC(Media Access Control) : 장비가 네트워크 매체에 대한 접근 통제, 충돌 회피를 위한 CSMA/CD 등 ○ LLC(Logical Link Control) : 물리적 장치를 논리적으로 연결, 연결을 유지, 네트워크 계층에 서비스 제공	
흐름 제어		
정지-대기 (Stop and Wait)	○ 흐름 제어의 가장 간단한 방법, 송신 후 수신측으로부터 ACK, NAK(오류) 받을 때까지 대기하는 방식	
슬라이딩 윈도우	○ 한번에 여러 개의 프레임을 전송하는 방식, 수신측에 n개의 프레임 버퍼 할당 ○ 송신측은 수신측의 ACK를 기다리지 않고 보내는 방식, 각 프레임에 순서 번호 부여	
오류 제어		
FEC		
ARQ		
MAC		

구성	○ 6Byte : 앞 3바이트(벤더) + 뒤 3바이트(벤더 내 코드) ○ 데이터 링크 계층(2계층)에서 사용	
브로드캐스트	○ ff-ff-ff-ff-ff-ff	
IPv4		
	○ IPv4(32bit 표기법) : 헤더 가변 길이, 브로드캐스트 - A Class(0), B class(10), C Class(110), D Class(1110, 254), E Class(1111) ----> 0 > 10 > 110 > 1110 > 1111 : 1이 계속 추가되는 방식 - CIDR은 기존 A/B/C 와 같은 주소 개념 무시하고, 자유롭게 구성 ○ 서브네 마스크는 32bit 길이 가짐	
브로드캐스트	○ 서브넷 호스트 ID 비트가 모두 1인 주소 ○ 166.132.4.0/22의 경우 하위 10비트가 모두 1인 주소 --> 166.134.7.255	
전송 방식		
유니캐스트	○ 하나의 송신자가 하나의 수신자에게 패킷을 전송하는 방식(특정인에게 전송)	
멀티캐스트	○ 하나의 송신자가 멀티캐스트 수신자에게 패킷을 보내는 경우, 일대다 패킷 전송 방식 ○ 네트워크 장치가 멀티캐스트를 지원해야하며, 멀티캐스트 그룹에 가입되어야 함.	
브로드캐스트	○ 같은 네트워크 모든 호스트에게 패킷을 보내는 방식 ○ 호스트 ID 비트를 모두 1로 설정 후 전송	
IPv6		
	○ IPv6(128bit 표기법) : 헤더 고정 길이, 유니/애니/멀티 캐스트 지원, 8개 필드로 주소 표현 - 삭제 필드 : 헤더 길이, 식별자, 플래그, 분할 옵션, 체크섬 필드 - 변경 필드 : 프로토콜 타입, TTL 필드 이름 변경 - 애니캐스트 : [IPv6만 제공], 가장 가까이 있는 애니캐스트 그룹 중 한나에게만 전송(?) - 멀티캐스트 : 그룹의 각 컴퓨터가 복사본 수신, 브로드캐스트 대체 - 주소 변환 : Dual Stack, 터널링, 해더변환(일부 ipv4 남은 경우)	
	○ CIDR : 클래스가 없는 도메인간 라우팅 지원, IPv4를 보다 효율적으로 사용 xxx.xxx.xxx.xxx/x 표기 - CIDR 표기법 공부 필요	
프로토콜		
프로토콜 포트	○ SMTP : 25, TCP ○ FTP : 21, TCP ○ DNS : 53, TCP / UDP ○ SNMP : 161, 162 UDP ○ SSH : 22 ○ TLS/SSL : 443	
ICMP		
	○ Echo Request(문제 해결 메시지) > Echo Reply(Request에 대한 응답) ○ Redirect : 더 적합한 경로 있음 안내 ○ Source Quench : 라우터 집중으로 패킷 손실, 혼잡 제어(전송 속도 조절) 필요	
TCP		
	○ 흐름 및 오류 제어 사용 ○ 연결 종료 : 4way, 3way 두가지 방법	
SNMP		
TCP 상태 전이	p287	
UDP		
	1) 흐름 제어 없음, 오류 제어 미지원, 혼잡 제어 미 제공 2) 멀티 캐스팅을 위한 전송 프로토콜, SNMP 응용, 라우팅 경로 갱신	
UDP		
	1) 7계층 프로토콜, 네트워크 관리 담당, 분산 네트워크 관리 기능 미 포함, V3는 보안 기능 제공	
SCTP		

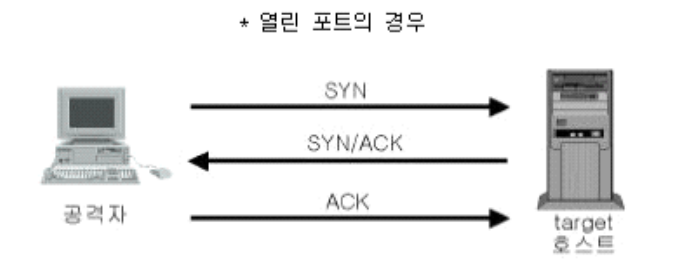
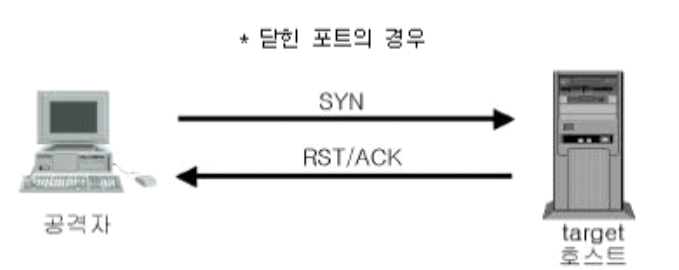
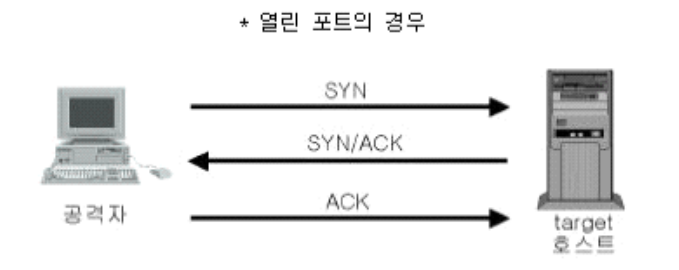
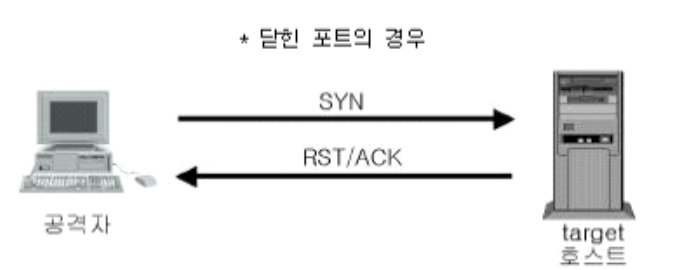
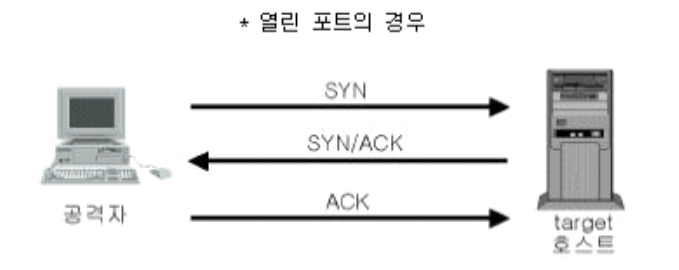
	1) TCP와 UDP의 장점을 취합 전송 방식	
Router		
Posison Reverse	○ 루프 문제 해결, 도달 홉 카운터 증가	
Triggered update	○ 라우팅 정보의 비 효율적 개선, 변경될 경우에는 정보 업데이트	
split horizon	○ 수신한 동일 인터페이스에 재전송하지 않음. 루프 방지 목적	
hold down	○ 제어 과정의 주요 사항을 일시 보류. 경로가 나빠지면 잠시 유보 등	
매스캐레이딩 라우터	○ 리눅스의 네트워크 기능으로 상용 방화벽, 네트워크 라우터의 기본 기능인 1 대 다 방식의 NAT	
리플트 라우팅	○ 라우팅에 등록되지 않은 모든 패킷에 지정하는 경로 ○ 목적지 네트워크 경로를 알지 못하거나, 라우팅 테이블에 맞는 경로가 없을 때 적용 ○ 모든 패킷을 처리하는 기본 경로 ○ 라우팅 테이블의 공간 부족시 활용	
Ingress 필터링	○ Standard, Extended Access-List를 활용, 라우터 내부, 즉 사내로 유입되는 패킷 소스를 허용 또는 거부	
Egress 필터링	○ 내부에서 라우터 외부로 나가는 패킷에 IP를 체크, 허용 또는 거부하는 것	
블랙홀 필터링 (Null 라우팅)	○ 특정한 목적지IP(대역)에 대해 null 이라는 가상의 인터페이스로 보내 패킷 통신을 제한하는 것	
라우팅 방식	○ 정적 / 동적 라우팅으로 분려	
no ip unreachable 설정	○ 송신할 수 없는 패킷 수신 시 소스에게 unreachable 메시지를 전송하지 않음 ○ Black-hole 필터링 후 DDOS 공격 대응 위해 반드시 필요한 설정	
no ip source-route	○ 소스 라우팅을 금지하는 설정	
소스 라우팅	○ 패킷은 목적지 주소만 가짐, 송신자 쪽에서 IP 옵션 헤더에 경로 리스트 작성 시 지정된 경로 이용 가능	
라우팅 모드	○ User 모드 : 테스트를 위한 모드 ○ Privileged 모드 : 모든 명령 사용 가능, 환경 설정 파일 조정(enable 명령) ○ Configuration 모드 : 라우터 구성 파일 변경 ○ RXBOOT 모드 : 패스워드 분신, 이미지 문제 등 복구가 필요한 경우 ○ Setup 모드 : 처음 구매 후 초기 설정하는 모드	
디폴트 라우터 설정	○ Network, Mask 모드 0 으로 설정	
[라우팅 프로토콜] 내부	○ RIP : 최대 홉 15개 제한, 포워딩 테이블, 경로 회선 품질 정보 반영 불가, 경로 변경 사항 즉시 반영 불가 Bellman-Ford 프로토콜 적용, 이웃 라우터와 30초마다 정보 교환 ○ OSPF : 링크 상태 라우팅 기반, 전체 네트워크 변화 생길 경우 플러딩 수행, Dijkstra 알고리즘 사용 ○ GRP : RIP 단점 보완, 네트워크 파라미터(품질)를 경로에 반영,대역폭/지연/신뢰도/부하/MTU CIDR/VLSM 지원 불가, 라우팅 루프 문제 여전히 발생, 다이나믹 프로토콜 ○ EIGRP : IGRP 라우팅 루프문제 해결, CIDR/VLSM 지원	
[라우팅 프로토콜] 외부	○ BGP : 유일한 인터 도메인 라우팅 프로토콜, 경로 백터 기반 알고리즘, 홉수 대신 AS 번호 매트릭 사용	
라우팅 알고리즘	○ 거리 벡터 라우팅(DV) : 인접 노드의 기초 정보, 최소 비용 트리, 목적지 까지 최소 비용 제공 - RIP(제록스) ○ 링크 상태 라우팅 : LSDB를 기초하여 경로 생성, 플러딩 과정 통해 DB 구축, 다익스트라 적용 - OSPF : ○ 경로 벡터 라우팅(PV) : 스패닝 트리로 결정(최소 비용 트리 아님) - BGP	
Router 명령		
패스워드 암호화 기능 활성화	○ Router#configure terminal ○ Router(config)# service password-encryption	
패스워드 암호화 저장	○ Router#configure terminal ○ Router(config)# enable secret [비밀번호]	
패스워드 평문 저장	○ Router#configure terminal ○ Router(config)# enable password	

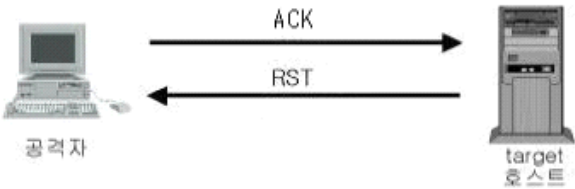
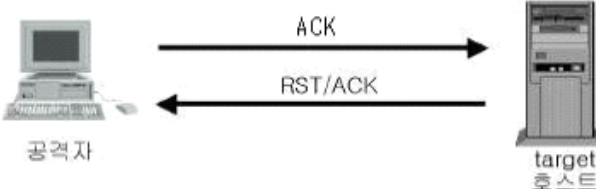

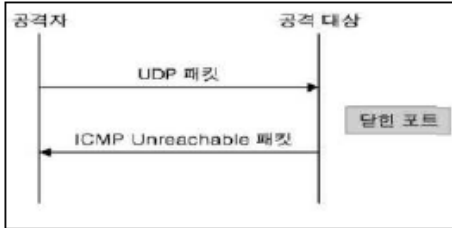
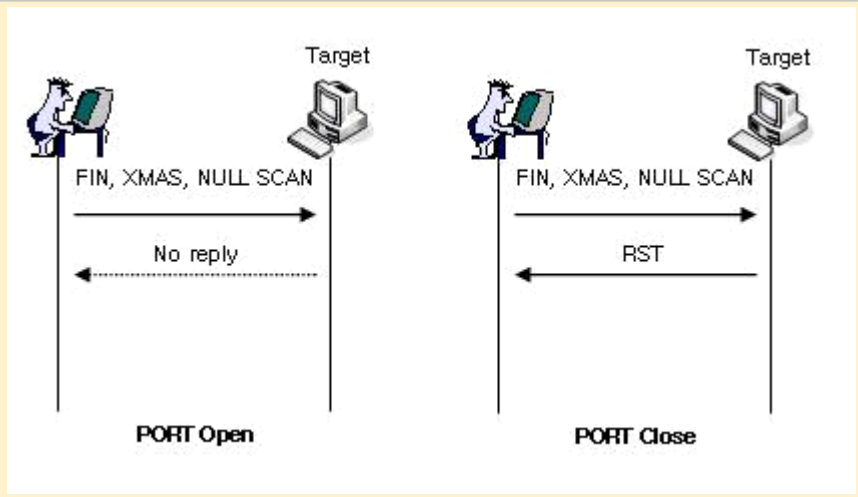
! 텔넷 포트에 암호 설정	<ul style="list-style-type: none"> ○ #conf t ○ Router(config)# line vty 0 ○ Router(config-line)# password qlalf123 ○ Router(config-line)# exit 	
! 콘솔 비밀번호 설정	<ul style="list-style-type: none"> ○ #conf t ○ Router(config)# line con 0 ○ Router(config-line)# password [비밀번호] 	
! privilege 모드 암호 설정 및 암호화 저장	<ul style="list-style-type: none"> ○ #conf t ○ Router(config)# enable password qlalf123 ○ Router(config)# service password-encryption 	
! 사용자 모드에서 privileged 모드 전 환	<ul style="list-style-type: none"> ○ enable 	
! 라우터 이름 지정	<ul style="list-style-type: none"> ○ #conf t ○ Router(config)# hostname rrrname 	
! e0 I/F에 IP 지정	<ul style="list-style-type: none"> ○ #conf t ○ Router(config)# interface Ethernet - ○ Router(config-if)# ip address 192.111.111.111 255.255.255.0 	
! black hole 설정	<ul style="list-style-type: none"> ○ #conf t ○ Router(config)# interface null 0 : 응답 설정 차단 ○ Router(config-if)# no ip unreachableables ○ Router(config-if)# exit ○ Router(config)# ip route 10.0.0.0 255.0.0.0 null 0 : 출발지 10.0.0.0 의 패킷을 static routing 하여 폐기 	
<u>HUB</u>		
HUB 특성	<ul style="list-style-type: none"> ○ 한 포트에서 수신한 패킷을 Broadcast로 전파 ○ 트래픽 조작이나 검사 불가 ○ 기타 제어 기능 없으며, Collision과 Broadcast Domain 구분 불가 	
<u>Switch</u>		
Switch 특성	<ul style="list-style-type: none"> ○ 한포트에 수신한 패킷을 특정 포트로 전달 ○ 확장에 제한 없으며 이중화(Spanning Tree Protocol) 구성 가능 	
Spanning Tree Protocol	<ul style="list-style-type: none"> ○ 스위치나 라우터에서 발생하는 루핑을 막아주기 위한 프로토콜 ○ 출발 부터 목적지까지 두개 이상의 경로 존재시 한개의 경로만 남겨두고 나머지 경로는 차단 ○ 사용 경로에 문제가 생기면 다른 경로를 살려주는 형태로 구성 ○ 이를 통해 이중화 구성 가능 	
<u>NAT</u>		
Normal NAT	<ul style="list-style-type: none"> ○ 내부 사설 IP 주소를 가지고 있는 클라이언트가 외부로 접속하는 경우 	
Reverse NAT	<ul style="list-style-type: none"> ○ 내부 네트워크에서 작동하는 서버에서 외부 클라이언트가 접속하는 경우 	
Redirect NAT	<ul style="list-style-type: none"> ○ 목적지 주소를 재 지정하는 경우(서버 주소가 변경될 경우 패킷의 목적지 주소를 변경 주소로 바꾸어줌) 	
Exclude NAT	<ul style="list-style-type: none"> ○ Normal NAT 적용을 받지 않고 방화벽을 지나도록 설정 ○ 방화벽과 라우터 사이에 서버가 있는 경우와 같이 특수 상황에 적용 	
<u>Switch 공격 기법</u>		
Switch Jamming	<ul style="list-style-type: none"> ○ 변조한 MAC 정보를 담고 있는 ARP Reply 패킷을 계속해서 네트워크에 전송 ○ 스위치는 MAC 정보를 테이블에 저장시도하다가 테이블이 가득 차면 브로드 캐스팅 함 ○ 보안 장비는 장애 시 Fail Close 규칙을 따르나 Switch는 이 규칙을 따르지 않음. 	
ARP Spoofing	<ul style="list-style-type: none"> ○ 특정 호스트의 MAC 주소를 공격자의 MAC 주소로 위조한 ARP Reply 패킷 전송 ○ 특정 호스트의 패킷을 공격자가 가로채기 할 수 있음. 	
ARP Redirect	<ul style="list-style-type: none"> ○ 공격자는 자신이 라우터처럼 위조한 ARP Reply 패킷을 전송, 네트워크에 연결된 모든 HOST가 착각 	
ICMP Redirect	<ul style="list-style-type: none"> ○ ICMP Redirect 메시지는 라우터에서 호스트 또는 라우터간 경로 재설정 메시지 ○ 특정 IP, 특정 대역에서 나가는 패킷의 라우팅 경로를 자신의 주소로 위장, 희생자의 라우팅 T를 공격 	
Span/Monitor Port	<ul style="list-style-type: none"> ○ 모니터링 포트에 물리적인 접근 후 패킷 스니핑 	
<u>스니핑 방어</u>		

방어	○ 암호화, ARP 캐쉬 Static 설정, 스니핑 탐지 도구 활용 모니터링, VLAN 활용 브로드캐스팅 도메인 축소	
스니핑 탐지	○ ping : 의심 호스트에 존재하지 않은 MAC 주소로 ping 전송 ○ ARP : 의심 호스트에 존재하지 않은 MAC 주소의 ARP Request 전송 ○ DNS : ping sweep 전송 후 inverse lookup 패킷을 감시 ○ Decoy : 가짜 계정과 비밀번호를 네트워크에 뿌린 후 접속 시도를 감시 ○ ARP watch : 초기 MAC + IP를 보관 후 이의 변동 사항을 감시 ○ Tool : antsniiff, Sentinel, Hunt	
통신		
CSMA / CD	○ 유선 전용, 지연 시간 예측 불가, 다수 사용자 매체 공유, 통신량 증가->충돌 횟수 증가 ○ 데이터링크 접근 통제 기술 중 이더넷에 사용하는 기술 ○ Data 충돌을 감지, 채널이 비어있는 경우 누구라도 이용하게 하는 방식	
Backoff Time	○ 충돌 발생 후 대기했다가 전송하기까지의 소요 시간	
CSMA / CA	무선 지원	
Broadcast Domain	○ 다른 장치의 브로드캐스트 메시지를 수신하는 장비들의 집합 ○ 동보 패킷의 전달이 허용되는 영역, 허용 영역은 라우터를 기준으로 분할	
Collision Domain	○ 물리적 매체에 연결된 장치의 집합, 2개의 장치가 동시에 매체에 접근하는 경우 두 신호가 충돌하는 것	
패리티 체크	○ 전송 장치에서 각 문자에 하나의 비트를 추가, 전송 오류를 체크하는 기법	
블루투스		
블루재킹	스팸, 명함을 익명으로 뿌림.	
블루스나핑	보안 취약점 통해 데이터 접근, OPP(OBEX Push Profile, 무인증 정보 교환) 기능 악용	
블루 버킹	사용자 모르게 이동 전화 명령 사용, 전화 통화 가능, 전화 번호	
무선 프로토콜		
WAP	p289	
WSP		
WTLS		
WTP		
War Driving	○ 무선 네트워크를 찾기 위해 주위 AP를 찾는 과정	
무선 암호화		
무선 암호화 기술	○ TKIP : WEP의 RC4 적용 문제 해결, key Mixing, Dy WEP Key 적용, ○ CCMP : AES 암호화 블록 사용, TKIP보다 진일보	
무선 인증 프로토콜	○ WEP(Wired Equivalent Privacy) : 유선 네트워크 수준의 보안 제공 목표, 802.11 표준 에 정의 - 대칭키 구조 암호화 알고리즘, RC4 알고리즘 적용으로 보안 취약 2) WPA : 암호화 기법은 TKIP 사용, WEP 약점 보완, RADIUS, Kerberos 등 적용 3) WPA2 : TKIP 대신 AES 기반의 CCMP 암호화 방식 적용 4) WPK : Wireless PKI 4) EAP ○ WAP 1) Transport layer : WDP 2) Security Layer : WTLS : SSL 보완한 방식, TLS 기반, 무결성 제공, 부인 방지 미제공 3) Transation Layer : WTP 4) Session Layer ; WSP	
공격 기법 정리		
Tiny Fragment	○ 패킷을 잘개 쪼개어 포트 정보가 두번째 패킷에 포함되도록 해서 방화벽을 우회하는 공격 기법	
Teardrop Attack	○ IP 단편화 조립 위한 Offset 고의 조작, 패킷 재조립 오류 및 오버헤더 유발	
Land Attack	○ 출발지 및 공격지 IP 동일하게 조작하는 공격	
SMURF Attack	○ 출발지 IP를 위조(공격 대상)한 패킷으로 반사 서버에 전송, 반사 서버는 공격 대상에게 ICMP 패킷 회신	



	<ul style="list-style-type: none"> ○ IP Spoofing / ICMP Broadcast / 중복 ○ 대응 : 유입되는 Directed Broadcast 패킷 차단, ICMP Echo Request 패킷 응답 거부 설정 	
TCP Syn Flooding	<ul style="list-style-type: none"> ○ TCP 3-way handshaking 단계에서 마지막 ACK 를 고의로 전송하지 않은 공격 기법 ○ netstat -an 으로 확인 시 SYN_RCVD(RECV) 상태의 연결이 다수 존재 ○ 대응 : backlog queue 확대, Syn Cookie 설정, SYN+ACK 대기 시간 축소, 방화벽/DDOS 장비 설정 	
NTP 증폭 공격		
Trinoo 공격	○ Trinoo DDOS 공격 툴을 사용하는 방법으로 Attacker / Master / Agent 로 구성	
Get Flooding 분석	○ ngrep -l ser.pcap -tW byline grep GET sort uniq -c sort -rn <- r + n p.344	
DOS		
TCP Syn Flooding Attack	<ul style="list-style-type: none"> ○ 3-Way Handshaking 과정에서 2단계 ack를 회신하지 않는 공격 방식 ○ Backlog Queue 소진을 목표로 하는 공격 <pre> 1 0.000000 10.23.70.52 5.9.3.36 TCP 66 http > 43485 [SYN, ACK] Seq=0 20 0.062365 10.23.70.52 5.9.3.36 TCP 66 http > 56801 [SYN, ACK] Seq=0 60 0.109198 10.23.70.52 5.9.3.36 TCP 66 http > 33771 [SYN, ACK] Seq=0 61 0.113911 5.9.3.36 10.23.70.52 TCP 66 26119 > http [SYN] Seq=0 Win=0 62 0.113988 10.23.70.52 5.9.3.36 TCP 66 http > 26119 [SYN, ACK] Seq=0 96 0.171590 10.23.70.52 5.9.3.36 TCP 66 http > 54847 [SYN, ACK] Seq=0 104 0.190848 5.9.3.36 10.23.70.52 TCP 66 29719 > http [SYN] Seq=0 Win=0 105 0.190951 10.23.70.52 5.9.3.36 TCP 66 http > 29719 [SYN, ACK] Seq=0 135 0.113930 5.9.3.36 10.23.70.52 TCP 66 26119 > http [SYN] Seq=0 Win=0 151 0.190873 5.9.3.36 10.23.70.52 TCP 66 29719 > http [SYN] Seq=0 Win=0 529 0.389328 5.9.3.36 10.23.70.52 TCP 66 57179 > http [SYN] Seq=0 Win=0 531 0.389371 10.23.70.52 5.9.3.36 TCP 66 http > 57179 [SYN, ACK] Seq=0 547 0.389293 5.9.3.36 10.23.70.52 TCP 66 57179 > http [SYN] Seq=0 Win=0 573 0.599599 5.9.3.36 10.23.70.52 TCP 66 34881 > http [SYN] Seq=0 Win=0 580 0.599595 5.9.3.36 10.23.70.52 TCP 66 34881 > http [SYN] Seq=0 Win=0 581 0.599681 10.23.70.52 5.9.3.36 TCP 66 http > 34881 [SYN, ACK] Seq=0 589 0.735720 5.9.3.36 10.23.70.52 TCP 66 61036 > http [SYN] Seq=0 Win=0 591 0.796641 5.9.3.36 10.23.70.52 TCP 66 8767 > http [SYN] Seq=0 Win=0 598 0.867315 5.9.3.36 10.23.70.52 TCP 66 28125 > http [SYN] Seq=0 Win=0 605 0.735745 5.9.3.36 10.23.70.52 TCP 66 61036 > http [SYN] Seq=0 Win=0 606 0.735790 10.23.70.52 5.9.3.36 TCP 66 http > 61036 [SYN, ACK] Seq=0 609 0.796635 5.9.3.36 10.23.70.52 TCP 66 8767 > http [SYN] Seq=0 Win=0 </pre>	
TCP Syn Flooding Attack 대응 방법	<ul style="list-style-type: none"> ○ Syn Cookie 설정 ○ Backlog Queue 크기 확장 ○ SYN + ACK에 대한 대기 시간 축소 ○ 방화벽 또는 DDOS 대응 장비를 통해 동일 Client의 단위 시간 당 요청 건수의 제한 설정 	
SMURF Attack	<ul style="list-style-type: none"> ○ ICMP 출발지 조작, 출발지 IP를 공격하고자하는 IP로 설정 ○ Broadcast, ICMP Echo Request / Echo Reply 	
SMURF Attack 대응	<ul style="list-style-type: none"> ○ 중간 매개지 쓰임 차단하기 위해 라우터에서 외부에서 들어오는 IP broadcast 패킷 차단 ○ 호스트는 boardcast address로 전송된 ICMP 패킷에 대해 응답하지 않도록 시스템 설정 	
UDP Flooding Attack	○ UDP의 비 연결적 특성 활용, 대량의 UDP 패킷을 전송, 희생자의 네트워크 대역폭을 소진하는 공격	
Trinoo Attack	○ DDOS 공격 툴인 Trinoo를 사용하는 공격(Attacker, Master, Agent)로 UDP Flooding 공격 수행	
Flooding Attack	○ ICMP, UDP, TCP Sync	
Land Attack	○ 출발지 IP / 목적지 IP를 동일하고 조작하여 공격하는 방식	
Ping of Death	○ 대량의 ICMP 패킷 전송	
Teardrop Attack	○ IP 단편화 조립 위한 Offset 고의 조작, 재조립 오버헤더 발생	
Targa	○ 다양한 공격 방식을 종합	
Bonk	○ 패킷의 순서 번호를 모두 1번으로 조작하는 공격	
Boink	○ 중간에 패킷 번호(시퀀스 번호)를 비 정상적으로 보내는 공격	
패킷 쪼개기	○ 패킷을 작게 쪼개어 방화벽으로 통과하는 공격	
Slow HTTP Post Slow HTTP Header Slow HTTP Read	<ul style="list-style-type: none"> ○ 동시 연결 갯수 제한 ○ Connect Timeout 설정 : httpd.conf / Timeout 120 ○ 요청헤더와 바디에 각각 Timeout 설정 : httpd.conf : RequestResponseTimeout header=5 body=8 	
HTTP Get Flooding	○	
HTTP Get Flooding	○	

with Cache-Control		
Syn Cookie		
Syn Cookie	<ul style="list-style-type: none"> ○ 클라이언트 SYN에 대한 응답(SYN/ACK)에 Cookie 값을 넣어 전송 ○ Syn Cookie를 통해 상대방의 유효성 확인 전까지 Backlog Queue 사용하지 않음. ○ 이를 바탕으로 Syn Flooding 공격 대응 가능 	
Syn Cookie 설정	<ul style="list-style-type: none"> ○ echo 1 > /proc/sys/net/ipv4/tcp_syncookies ;0:미설정, 1:설정 ○ sysctl -w net.ipv4.tcp_syncookies=1 	
DOS 대응		
! 라우터 필터링	<ul style="list-style-type: none"> ○ Ingress 필터링 적용 : 사설 IP의 유입을 차단 ○ Egress 필터링 적용 : 관리하는 IP 대역이 아닌 경우 외부 나가는 것 차단 	
! Rate-Limit 적용	○ 특정 서비스, 특정 패턴을 가진 패킷이 단위 시간 당 일정량 이상 초과하면 패킷 통과 금지	
! uRPF (unicast Reverse Path Forwarding)	<ul style="list-style-type: none"> ○ 라우터가 패킷을 받으면 출발지 IP 주소를 확인하여 해당 IP로 갈 수 있는 역 경로가 존재하는지 확인 ○ 출발지 IP의 스푸핑 여부 판단 목적 	
! ICMP 차단	○ ICMP-Broadcast 및 ICMP Redirection 비 활성화	
! MRTG / 보안 장비	<ul style="list-style-type: none"> ○ MRTG를 통해 급작스럽게 유입되는 트래픽 모니터링, 감당 힘든 경우 Null 라우팅 처리 ○ 방화벽, IDS/IPS, Anti DDOS 장비 운용 	
DDOS		
Trinoo	○ 트리누 마스터가 여러개의 IP에게 공격을 지시하는 방식	
TFN	○ 투리누와 유사, 다양한 형태의 유사 공격 제공	
TFN2K	○	
Stacheldraht	○ 트리누와 TFN 참고 작성	
DRDOS		
DRDOS	<ul style="list-style-type: none"> ○ TCP 3-way handshake 취약점 악용 ○ BGP의 취약점 악용 ○ 별도의 에이전트 필요 없음 ○ 정상적인 SYN 패킷을 이용, 반사 서버로 부터 SYN/ACK 패킷 발생 ○ Raw Socket 생성 가능 호스트는 반사 공격을 위한 SYN 패킷 생성 쉬움 ○ 반사 서버 목록을 생성하여 희생 서버에 대한 공격을 수행하도록 함 	
DRDoS 공격 기법	<ul style="list-style-type: none"> ○ TCP 3-way handshake 취약점 공격 ○ ICMP 프로토콜 활용, echo request 패킷을 반사 서버에 전달, echo reply가 공격 대상에 전송 ○ UDP 프로토콜 활용, DNS/NTP/SNMP/CHARGEN 등의 서비스 이용 위조 요청 및 응답 요구 	
DNS DRDoS 방어	<ul style="list-style-type: none"> ○ DNS 설정을 통해 내부 사용자만 재귀 쿼리(Recursive Query) 허용, allow-recursion ○ 서버 방화벽을 통해 특정 파이트 이상의 DNS 응답 차단 	
포트 스캐닝		
TCP Open Scan 방식	○ SYN/ACK 스캔, 완전한 세션 성립하여 스캔하는 방식, 로그 기록 남고, 속도 느림	
TCP Flag	<ul style="list-style-type: none"> ○ URG : 긴급 플래그 (데이터를 전송하는 중간에 [CTRL + C]와 같은 행동 했을시 발생 플래그) ○ ACK : 수신 확인 플래그 (데이터가 제대로 전송되었다고 알려주는 플래그) ○ PSH : 푸시 플래그 (버퍼에 데이터가 차지 않아도 데이터 즉시 전송하겠다는 플래그) ○ RST : 리셋 플래그 (재설정을 요구하는 플래그 이상 패킷이 도착 시 잘못 전송했다고 알려주는 플래그) ○ SYN : 동기화 플래그 (상대에게 연결을 해도 되는지 제의하는 플래그) ○ FIN : 종료 플래그 (세션을 종료하고자 할때 사용하는 플래그, 접속 종료하겠다는 플래그) 	
Stealth Scan 방식	<ul style="list-style-type: none"> ○ 공격 대상 속이고 스캔, 완전한 세션 성립 않음, 로그 남지 않음 ○ TCP Half Scan(syn scan) ○ Fin Scan : 열린 경우 응답 없음, 닫힌 경우 RST 응답 ○ NULL Scan ○ XMAS Scan ○ UDP 스캔 : 방화벽, 라우터에 의해 손실, 신뢰성 낮음 	

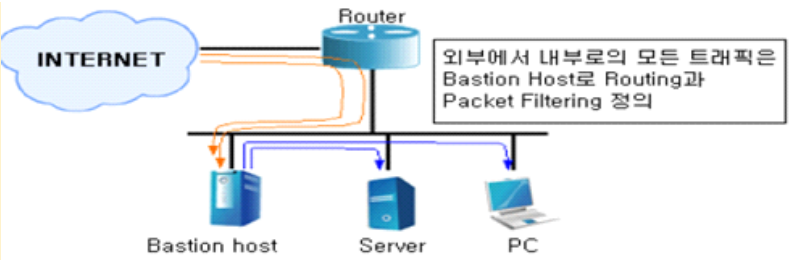
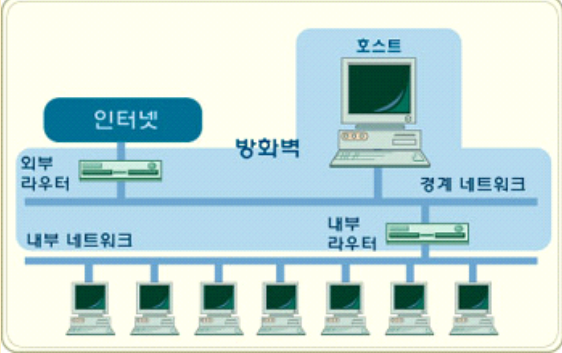
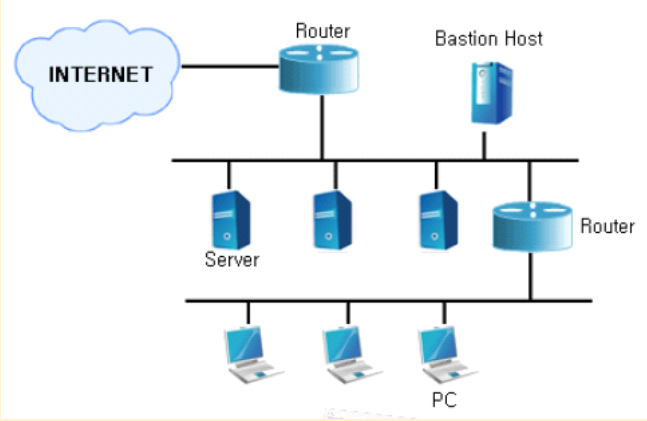
TCP Open Scan	<div><p>* 열린 포트의 경우</p><p>* 닫힌 포트의 경우</p></div>	
TCP Half Open Scan SYN Scan TCP Sync Scan	<div><p>○ SYN/RST 스캔, SYN 전송 -> Syn/ack 수신 -> 연결 종료, 닫힌 경우 rst/ack 수신</p><p>○ TCP 3-way handshake 이용 포트 접근, 마지막 신호(ack)를 생략 연결에 대한 log를 남기지 않는다.</p><div><div><p>공격자</p><p>공격대상</p><p>SYN</p><p>SYN+ACK</p><p>RST</p><p>열린 포트</p><p>포트가 열려있는 경우</p></div><div><p>공격자</p><p>공격대상</p><p>SYN</p><p>RST+ACK</p><p>"오지매"</p><p>닫힌 포트</p><p>포트가 닫혀있는 경우</p></div></div></div>	
Fin Scan	<div><p>* 열린 포트의 경우</p><p>* 닫힌 포트의 경우</p></div>	○ Fin 비트가 포함된 경우 허용하는 시스템이 많으며, Fin 비트 타입에 대해서는 로그 남기지 않는 특성
ACK Scan	<div><p>* 열린 포트의 경우</p></div>	

	<p>* 열린 포트의 경우</p>  <p>공격자 target 호스트</p> <p>* 닫힌 포트의 경우</p>  <p>공격자 target 호스트</p>	
UDP Scan	 	
XMAX Scan NULL Scan		
Decoy Scan	○ 스캔을 당하는 호스트에서 스캔 공격자 주소를 식별하기 어렵도록 다양한 위조 주소를 활용하는 공격	
UDP Scan	○ 공격 대상 포트 오픈 시 응답 없음 ○ 공격 대상 포트 닫힌 경우 -> ICMP Unreachable 응답	
! TCP 닫힌 포트 대상 스캔 방법	○ FIN / NULL / XMAX Scan	
VLAN		
VLAN	○ 데이터 링크 계층에서 브로드캐스트 도메인 나누기 위해 사용하는 기술, 보안성 향상 기대	[PMPI]
Tagging	○ 단일 링크 네트워크에서 VLAN 트래픽 통과 위해 라우터와 스위치 사이에 필요한 것 ○ 프레임이 어떤 VLAN에 속하는지 식별하기 위해 사용하는 기능 ○ 전송되는 패킷이 어떤 VLAN에 속하는지 다른 스위치에게 알리는 기능	
Port 기반 VLAN	○ 스위치 포트를 기준으로 vlan 구성, 가장 일반적인 구성 방식	
MAC 기반 VLAN	○ MAC 주소를 등록하여 VLAN을 구성하는 방식, 맥 관리 부담으로 잘 사용하지 않음	

네트워크 주소 기반 VLAN	○ 네트워크 주소를 기준으로 같은 네트워크에 속한 호스트 간 통신 가능, 주로 IP 네트워크 사용														
프로토콜 기반 VLAN	○ 같은 통신 프로토콜 간 통신 가능하도록 구성														
VPN															
VPN	○ 2계층 터널링 프로토콜 : PPTP, L2F, L2TP ○ 3계층 터널링 프로토콜 : IPSec, MPLS(도입 간편, 저가, 운영관리 간단)														
PPTP	○ 2계층 사용, MS가 개발한 터널링 프로토콜, TCP 연결 사용, IP/IPX/NetBEBU 암호화 ○ IP 헤더 캡슐화 및 하나의 터널에 하나의 연결 지원														
L2F	○ 시스코에서 제안한 터널링 프로토콜, 사용자는 별도의 S/W 불필요														
L2TP	○ 시스코에서 제안한 L2F + PPTP 기술을 결합한 형태, 호환성 우수														
IPSec															
운영 모드	○ 전송 모드 / 터널 모드														
[운영모드] 전송 모드	○ Payload 보호, IP 헤더는 노출 되는 운영 모드														
[운영모드] 터널 모드	○ 패킷 전체에 대한 보호, 게이트웨이 간의 보안 기능, 전체 패킷을 캡슐화 하는 운영 모드														
프로토콜	○ AH / ESP														
[프로토콜] AH	○ 페이로드의 무결성 보호를 위해 개발된 프로토콜 ○ AH 프로토콜을 패킷의 암호화를 지원하지 않음 ○ 단말과 라우터 간의 IP 패킷에 대한 송신 인증 및 무결성 제공 <div>Original IP packet<table><tr><td>IP header</td><td>IP data</td></tr></table>AH in transport mode<table><tr><td>IP header</td><td>AH header</td><td>IP data</td></tr></table>AuthenticatedAH in tunnel mode<table><tr><td>Outer IP header</td><td>AH header</td><td>IP header</td><td>IP data</td></tr></table>Authenticated</div>	IP header	IP data	IP header	AH header	IP data	Outer IP header	AH header	IP header	IP data					
IP header	IP data														
IP header	AH header	IP data													
Outer IP header	AH header	IP header	IP data												
[프로토콜] ESP	○ 발신지 인증, 무결성, 프라이버시 보호, AH보다 나중에 개발, 공유된 대칭키로 암호화 ○ 송신자 인증 및 데이터 암호화 <div>Original IP packet<table><tr><td>IP header</td><td>IP data</td></tr></table>ESP in transport mode<table><tr><td>IP header</td><td>ESP hdr</td><td>IP data</td><td>ESP trailer</td><td>ESP auth</td></tr></table>EncryptedAuthenticatedESP in tunnel mode<table><tr><td>Outer IP hdr</td><td>ESP hdr</td><td>IP hdr</td><td>IP data</td><td>ESP trailer</td><td>ESP auth</td></tr></table>EncryptedAuthenticated</div>	IP header	IP data	IP header	ESP hdr	IP data	ESP trailer	ESP auth	Outer IP hdr	ESP hdr	IP hdr	IP data	ESP trailer	ESP auth	
IP header	IP data														
IP header	ESP hdr	IP data	ESP trailer	ESP auth											
Outer IP hdr	ESP hdr	IP hdr	IP data	ESP trailer	ESP auth										
SA	○ IPsec에서 송신용, 수신용 2개의 SA가 양단에 구성 ○ Sequence Number, AH/ESP 관련 정보, Lifetime														

!	IKE	○ Main 모드 / Aggressive 모드(보안 취약, 키 평문 전송)	
!	SAD		
!	Pre-Shard Key	○ 상대방 인증을 위해 사전에 공유된 키로 상대방을 인증하는 방식으로 IKE 단계에서 사용 ○ 간단하게 상대방의 인증이 가능하나, 통신 대상이 많은 경우 키 관리 어려움과 키 파일 도난시 문제	
MPLS VPN			
		1) 2계층 스위칭 속도 + 3계층 라우팅 기능 접목 2) 짧고 고정된 길이의 레이블 이용 스위칭 3) 패킷 지연 감소, 레이블 부여는 LER에서 수행 4) QoS, VoIP, TE 등	
스위치			
	Port Mirroring	○ 스위치 환경에서 모니터링과 트래픽 분석을 위해 특정 포트에 관한 패킷을 특정 포트로 전달하는 것	
	스위칭 방식	○ Cut-Through, Interim Cut-Through, Store-and-Forward	
	Cut-Through	○ 전체 프레임 수신 기다리지 않고, 송/수신지 정보 위해 몇바이트만 읽음, 중계 시간 최소화	
	Interim Cut-Through	○ 작은 런트 프레임 중계를 막는 기능을 보강한 방식	
	Store-and-Forward	○ 전체 프레임 수신 후 에러 검사하고 중계, CRC 검출 가능, 대기 시간 길어짐	
	스위치 재밍		
	MAC Flooding		
	Fail Close		
DNS			
	재귀 커리	○	
	증폭 공격	○	
	위임 권한 (Authoritative)		
방화벽			
	참고 사항	○ ICMP 패킷은 타입과 코드, 체크섬 들을 기준으로 필터링 ○ 사용자 인증 기능 제공하지 않음	
	배스천호스트	○ 내/외부 사이 게이트웨이 역할, 방화한 방어 기능이 구성된 시스템 ○ 라우터(스크리닝 라우터 역할) 뒤에 구성되는 방식, 방화벽 구성 시 필수 요소 ○ 외부 공격에 방어 정책이 구현되어 있는 네트워크에서 외부 접속에 대한 일차적인 연결을 받는 시스템	
			
	방화벽 구성	○ 스크리닝 라우터 구조 : 라우터를 이용 I/O 패킷 필터링 진행 ○ 이중 네트워크(듀얼 홈드) 호스트 구조 : 두개의 인터페이스 가짐, 배스천 호스트 역할 ○ 스크린드 호스트 게이트웨어 구조 : 듀얼 홈드 / 스크리닝 결함 구조 ○ 스크린드 서버넷 구조 : DMZ 완충 지역 역할	
	[방화벽 구성] 스크리닝 라우터	○ 내부와 외부 네트워크 사이에서 패킷 트래픽을 인거/거부하는 라우터를 활용하는 방식	
			

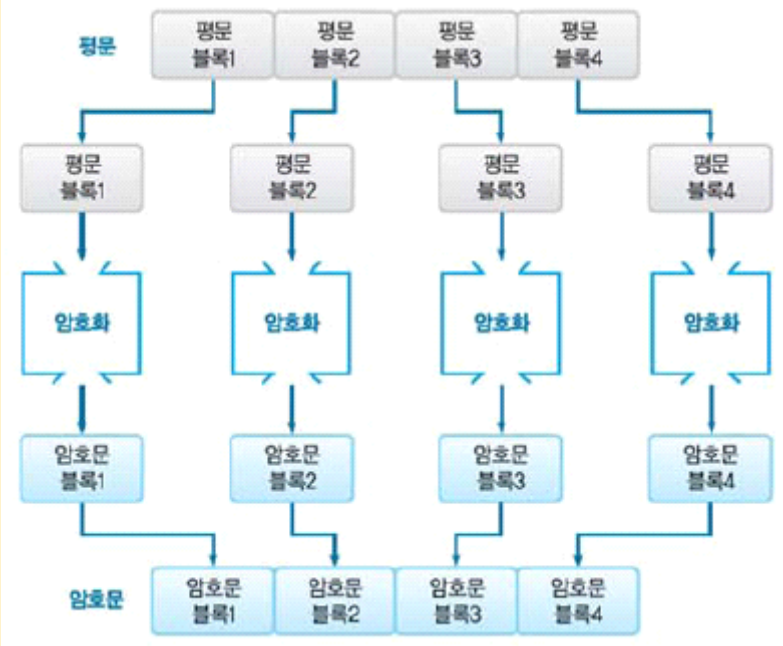
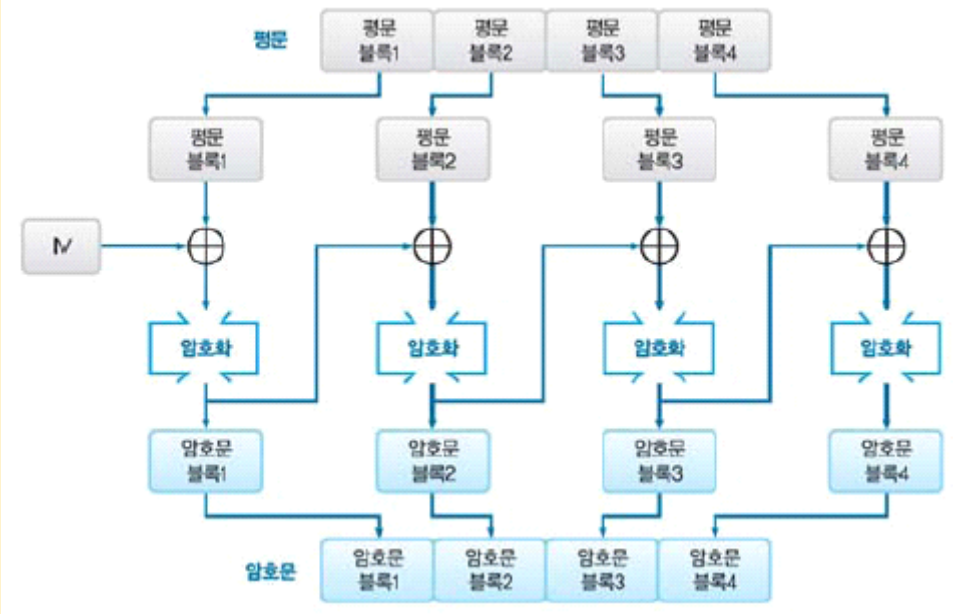
<p>단일 홈 / 듀얼 홈</p>	<div data-bbox="271 134 850 474"> <p>단일 홈 게이트웨이</p> </div> <div data-bbox="271 501 888 842"> <p>이중 홈 게이트웨이</p> </div>	
<p>[방화벽 구성] 듀얼 홈드</p>	<p>○ 외부 네트워크, 내부 네트워크 사이 2개의 인터페이스를 가지며, 라우팅 기능은 없음</p> <div data-bbox="271 918 831 1263"> </div> <div data-bbox="271 1263 1222 1509"> </div>	
<p>[방화벽 구성] 스크린드 호스트</p>	<p>○ 패킷 필터링 또는 스크리닝 라우터의 한 포트가 외부 네트워크에 연결, 다른 포트는 내부 네트워크 연결</p> <p>○ 전체적으로 보면 패킷 필터링 라우터와 호스트가 복합되어 방화벽 열할 수행</p> <div data-bbox="271 1624 831 2000"> </div> <div data-bbox="271 2027 1058 2145"> </div>	

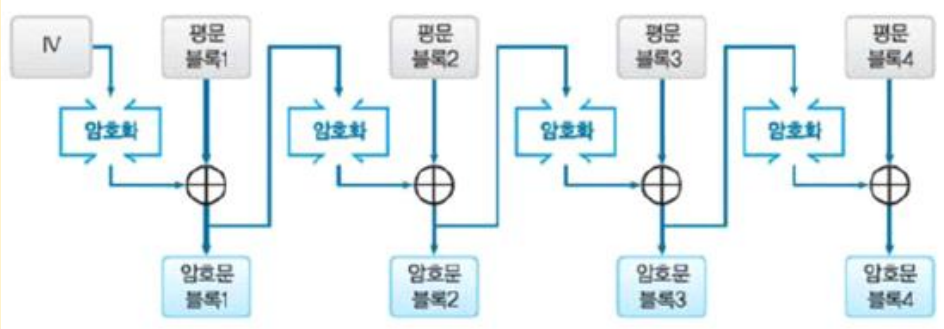
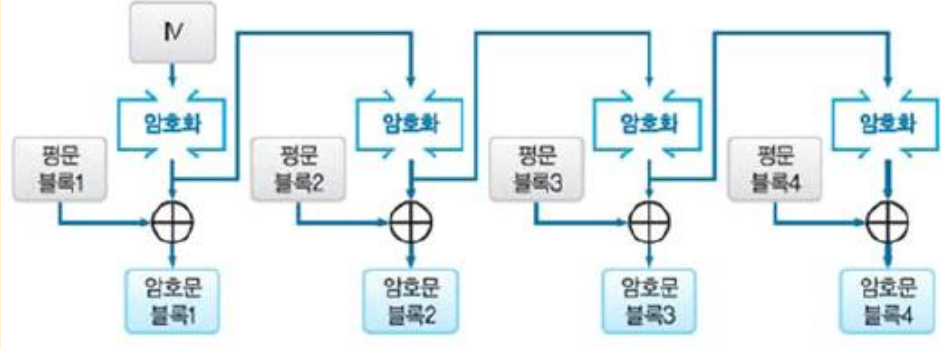
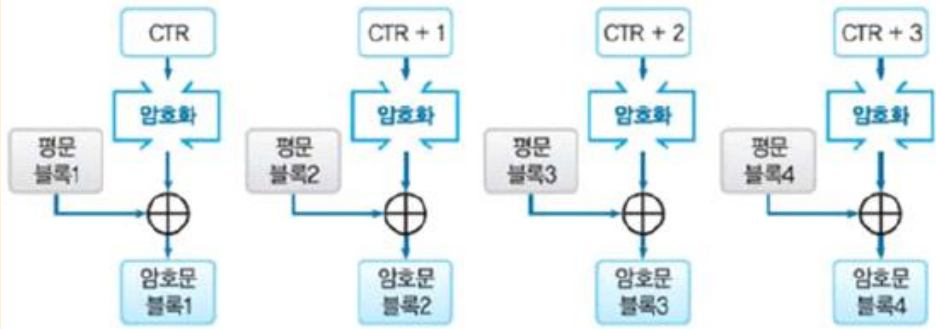
		
<p>[방화벽 구성] 스트린드 서브넷</p>	<p>○ 스트린드 호스트의 보안상의 문제점을 보완, 외부 네트워크와 내부 네트워크 사이 하나 이상의 경계</p>  	
방화벽 유형	<p>○ 패킷 필터 방화벽</p> <p>○ 스테이트풀 패킷 검사 방화벽</p> <p>○ 프록시 방화벽 : 응용 레벨 게이트웨이, 7계층에서 작동</p> <p>○ 어플리케이션 게이트웨 방식 : 전체 패킷 검사 후 내용에 기반하여 허용/거부를 결정</p> <p>○ 서킷 게이트웨이 : 하나의 프락시로 모든 서비스 제공, 클라이언트 수정이 필요한 경우 있음.</p>	
<p>! 스테이트풀 패킷 검사 방화벽 (Stateful Inspection)</p>	<p>○ 세션에 대한 추적 기능을 보완한 형태</p> <p>○ TCP 연결에 대한 정보 기록, 일련의 패킷 순서 인지 및 탐지</p> <p>○ 일정 시간 프로토콜의 상태 정보를 유지해서 보다 빠르고 높은 보안성 제공</p>	
설치 방식	○ 배스천호스트, 호스트-기반	
호스트-기반	○ 개별 호스트에 구성	
	단일 홈게이트웨이 / 이중(듀얼) 홈 게이트웨이	
IDS		
<p>! 탐지 방식</p>	<p>○ 지식 기반/오용 탐지 : 패턴 기반, 오탐률 낮음, 신규 유형의 공격에 취약 (상패전키, 전문가 시스템, 키 모니터링, 상태 전이 분석, 패턴 매칭)</p> <p>○ 행위기반/비정상행위 탐지 : 이상 행동에 반응, 인공 지능, 오탐률 높음, 신규 공격에 대응 (통신예, 통계적 접근, 예측 가능 패턴 생성, 신경망)</p> <p>○ 상태 전이 분석 : 상태 전이도 기반 분석</p> <p>- 전문가 시스템 : 침입/오용의 패턴을 실시간 입력되는 감사 정보와 비교</p>	
구성 방식	○ N-IDS, H-IDS	
<p>! N-IDS</p>	○ 서버 성능 저하 없음, 패킷 암호화 시 분석 불가	

	<ul style="list-style-type: none"> ○ 무차별 모드(Promiscuous mode)에서 동작하는 NIC가 설치 ○ 모든 트래픽을 캡처, 분석기를 통해 분석하며, 자신을 향하는 트래픽 분석 불가(H-IDS 필요) 	
! H-IDS	<ul style="list-style-type: none"> ○ 서버 설치, 다양한 로그 기록 접근, 명백한 침투에 대응, 호스트 성능 의존 ○ 단일, 다중(에이전트 간 정보 통합 및 교환) 구성 방식 존재 함. 	
탐지 시점 분류	○ 사후 분석 탐지, 실시간 탐지	
탐지 방식	○ 행위 기반, 지식 기반(시그니처)	
! 오탐	<ul style="list-style-type: none"> ○ 공격이 아닌 것으로 공격으로 인지 ○ false Positive ○ 행위 기반 IDS에서 주로 발생 	
! 미탐	<ul style="list-style-type: none"> ○ 공격을 탐지 못하는 문제 ○ false Negative ○ 시그니처 기반의 IDS에서 주로 발생하는 문제 ○ 통계적 임계치 설정의 부적절에 따라 발생 	
	오용탐지 / 비정상행위 탐지	
IPS		
ESM		
agent		
manager		
Console		
NMS		
! 정보 수집 방식	○ Polling 방식으로 네트워크 장비의 상태 정보 및 통계 정보를 주기적으로 수집	
! 정보 전달 방식	○ 클라이언트는 장비의 특정 이벤트를 실시간으로 Event Reporting / Trap 방식으로 전달	
SIEM		
! 운영 방식	○ 로그 수집 -> 로그 분류 -> 로그 변환 -> 로그 분석	[수분변분]
제공 기능	○ 데이터 통합, 상관 관계 분석, 알림, 대시 보드	
역추적		
TCP 패킷 역추적	○ TCP 연결 기반을 우회 공격하는 해커의 실제 위치를 실시간으로 추적하는 기법	
IP 패킷 역추적	○ IP 주소가 변경된 패킷의 실제 송신지를 추적하는 기술	
TCP 패킷 역추적 구현 방법	<ul style="list-style-type: none"> ○ 호스트 기반 ○ 네트워크 기반 	
IP 패킷 역추적 구현 방법	<ul style="list-style-type: none"> ○ PPM(확률적 패킷 마킹) 기법 ○ ICMP 역추적 기법 ○ 해시 기반 역추적 기법 	
CIDR		
Mac Address		
주소 구성	○ 48bit로 구성, 상위 24bit는 벤더 코드, 하위 24bit는 벤더의 일련 번호	
ARP		
ARP 스푸핑	<ul style="list-style-type: none"> ○ 호스트의 주소 매칭 테이블에 위조된 MAC 주소를 설정되도록 하는 공격 ○ 흔히 동일 랜에 연결된 내부자에 의해 공격 발생 ○ 공격자는 캐시 유효 시간(약 5분) 내 다시 캐시를 갱신해야 함. 	
ARP 스푸핑 대응	○ ARP cache table을 정적으로 구성	
ICMP		
ICMP	○ IP 프로토콜의 비신뢰성을 보완하기 위한 오류 메시지 통보 기능, 네트워크 상태 진단 목적 프로토콜	

ICMP Redirect	○ 악성으로 사용되는 메시지 타입 중 한가지 ○ 라우터가 보내는 패킷으로 호스트 라우팅 테이블 무력화, 새로운 경로 알려 줌	
ping	○ ICMP를 사용하는 대표적 프로그램으로 Echo Request / Echo Reply 메시지 이용	
tracert	○ ICMP를 사용하는 대표적 프로그램	
작업 Type	○ echo, timestamp, redirects	
<u>SNMP</u>		
개요	○ 7계층 프로토콜로 UDP 사용하며 161포트 사용	
메시지 구성	○ get-request, get-response, get-next-request, set-request, trap	
SNMPTRAP	○ 162포트 사용	
Community String	○ SNMPD와 클라이언트가 데이터 교환 전 인증하는 일종의 패스워드	
Community String 초기값	○ public ○ private ○ read only	
SNMP V3	○ 인증을 위한 암호화 제공(가장 최근 버전)	
응용 프로그램	○ MRTG가 사용하는 프로토콜	
MIB (Management Information Base)	○ 관리자에 의해 조회되거나 설정할 수 있는 대행자에 의해 유지되는 변수 ○ 망 기기 감시, 제어를 위해 사용되는 체계화된 관리 정보 항목들, 망 내 장치가 정적/동적으로 유지	
<u>IPtable</u>		
iptables	○ 넷 필터 프로젝트에서 개발, 광범위한 프로토콜 상태 추적, 패킷 어플 계층 검사 및 속도 제한 ○ 필터링 정책 명시를 위한 강력한 매커니즘 제공	
<u>SNORT</u>		
SNORT	○ 오픈소스 IDS, 프로토콜 분석, 콘텐츠 검색/비교, 다양한 공격과 스캔 탐지 ○ 패킷 로거, 스니퍼, 네트워크 침입 탐지 모드 등 다양하게 작동 가능	
[action] alert	○ 정해진 방식에 따라 alert 발생시키고 패킷을 기록	
[action] log	○ 패킷을 로그로 기록	
[action] pass	○ 패킷을 무시	
[action] activate	○ alert을 발생시키고 대응하는 dynamic 시그니처를 유효하게 함	
[action] dynamic	○ activate 시그니처에 의해 유효하게 된 경우에 log 처럼 작동	
[action] drop	○ 패킷 차단 후 로그에 저장	
[action] reject	○ 패킷 차단 후 TCP reset / unreachable 회신	
[action] sdrop	○ drop과 같지만 로그를 남기지 않음	
[option] msg	○ alert나 로그 출력 시 이벤트명으로 사용	
[option] content	○ 패킷, 페이로드 내부를 검색	
[option] offset	○ content에서 지정한 문자열의 오프셋 지정	
[option] depth	○ offset 부터 검사할 바이트 수 지정	
[option] nocase	○ 대소문자 무시	
[Rule Set] web scan	○ alert tcp any any -> 123.21.22.1 80 (content:"/admin"; msg:"web scan");	
<u>세션하이재킹</u>		
<u>증폭 공격</u>		
<u>WPA 해킹</u>		
airmon-ng	○ 모니터링 모드 생성 명령	





	○ airmon-ng start <무선 NIC> -c <공격할 AP 채널>	
airodump-ng	○ WPA 패킷 덤프 명령 ○ airodump-ng -c 3 -bssid 00:00:00... -w wpa mon0	
aireplay-ng	○ AP 또는 Station에 DOS 공격 : 재연결을 환경을 만들어 핸드 셰이킹 패킷 가로채기 위함 ○ aireplay-ng -0 1 -a 00:00:00:00:00:00 -c 00:00:00:00:00:00 mon0 ○ -0 : 공격 프로그램 일련 번호(0=dos), 1 : 한번 공격 ○ -a : BSSID(AP) , -c : station , mon0 : 모니터링 모드 무선 NIC	
aircrack-ng	○ 가로챈 핸드셰이크 패킷을 기준으로 사전을 통해 패스워드 크랙 시도 ○ aircrack-ng -w pw.lst -b 00:00:00:00:00:00 *.cap ○ -w : 패스워드 목록 파일, -b : 공격할 BSSID , *.cap : 패킷 캡처 파일	
<u>IGMP</u>		
IGMP	○ 멀티캐스트 그룹을 인근의 라우터에게 알리는 수단을 제공하는 프로토콜 ○ 첫번째 라우터 사이에 이루어지는 메시지로 TTL = 1로 설정되는 것이 특징 ○ 네트워크 계층에서 사용	
<u>3-Way Handshaking</u>		
연결 끊는 플래그	○ FIN : 정상 종료 ○ RST : 즉시 강제 종료	
<u>APT</u>		
위터링 홀 공격	○ zero-day 취약점 활용, 공격 타겟이 주로 이용하는 사이트 파악 후 해당 사이트 공격 ○ 해당 사이트에서 공격 타겟 공격 후 악성 코드 유포, 공격 대상자의 업무 시스템 접근	
스피어 피싱 공격	○ 특정인(공격 대상자가 잘 인지하는)으로 위장, 위조된 형태의 공격 메일 발송하는 형태	
<u>tcpdump</u>		
개요	○ 유닉스 계열에서 사용하는 패킷 캡처 도구 ○ -i 옵션 : 인터페이스 정보 지정 ex) eth0 ○ -nn 옵션 : reverse lookup 수행하지 않음, ip/port 번호 형식 출력	
80 포트 캡처	○ tcpdump -i eth0 -nn "tcp port 80"	
소스 80 포트캡처	○ tcpdump -i eth0 -nn "tcp src port 80"	
목적 80 포트 캡처	○ tcpdump -i eth0 -nn "tcp dest port 80"	
특정 IP 캡처	○ tcpdump -i eth0 -nn "tcp and host 123.212.323.221"	
소스 IP 캡처	○ tcpdump -i eth0 -nn "tcp and src host 123.212.323.221"	
80 포트, 특정 IP	○ tcpdump -i eth0 -nn "tcp port 80 and host 123.212.323.221"	
<u>netstat</u>		
netstat -an	○ 네트워크 연결 현황을 IP 그대로 보여주는 명령	
-a	○ 모든 소켓 정보를 출력	
-p	○ 지정한 프로토콜의 연결을 표시	
-n	○ 네트워크 주소를 숫자로 표현	
-e	○ 받은 패킷, 보낸 패킷의 이너넷 통계	
-r	○ 라우팅 정보 출력	
-i	○ 네트워크 인터페이스의 정보 출력	
-s	○ 각 네트워크 프로토콜의 통계 정보 출력	
<u>PCRE</u>		
문제점 파악	alert tcp any any -> any any { pcre:"post.*Content\Wx2dLength~~~~~";}	

암호학		
암호 모드	<ul style="list-style-type: none"> ○ ECB, CBC, CFB, OFB, CTR ○ 블록 모드(ECB, CBC), 스트림 모드 이용 가능한 블록 모드(CFB, OFB, CTR) 	
ECB (Electronic Code Book, 전자 부표호)	<ul style="list-style-type: none"> ○ ECB 모드, 간단, 고속, 병렬 처리, 평문반복이 암호문 재현, 재전송 공격가능 ○ 동일한 평문에 대해 동일한 암호문 출력, 해독 가능성 높음 ○ 평문 블록 분할 후 블록 별 별도 암호화 수행 후 조합 ○ 중간 블록 단위 별도 암호화 가능, 데이터베이스 암호화 시 병렬 처리 가능 	
CBC(Cipher Block Chaining, 암호 블록 연쇄) 모드	<ul style="list-style-type: none"> ○ ECB의 결점 개선, 복호화 시 병렬 처리 가능 ○ 각 평문 블록은 이전의 암호문 블록과 XOR 후 암호화 진행 ○ 최초 단계는 이전 단계가 없어 IV를 사용 ○ 각 블록의 결과는 이전 블록의 결과에 영향을 받는 구조 ○ 중간 블록만 암호화 불가하며, 이전 모든 블록이 필요, 복호화 시 1개 블록 파손, 2개 블록 영향 ○ 평문 블록의 한비트 오류는 출력된 모든 암호문에 영향 ○ IPSec 통신 기밀성 응용, 3DES, Kerberos 등에 활용 	

CFB(Cipher Feedback, 암호 피드백) 모드	<ul style="list-style-type: none"> ○ 블록 암호화 기반의 스트림 암호, 복호화 시 병렬 처리 가능 ○ 이전 블록의 결과를 암호화 수행 후 현 평문과 XOR 수행, 결과 활용 및 다음 단계 전달 ○ 암호화 완성된 결과물을 다음 암호화 수행에 반영 ○ IV는 암호화 후 첫 블록과 XOR 수행 ○ 블록 암호를 스트림 암호로 전환 가능 	
OFB(Output Feedback, 출력 피드백) 모드	<ul style="list-style-type: none"> ○ 블록 암호화 기반의 스트림 암호, 병렬 처리 불가, 패딩 불필요, ○ CFB와 유사하나, 암호문 블록은 이전 암호문 블록과 독립적, 오류 파급 영향 회피 가능, IV 사용 ○ 블록 암호화 기반의 스트림 암호 ○ 이전 블록의 암호화 결과 다시 암호화 후 평문과 XOR 수행 ○ IV 암호화 후 첫 블록과 XOR 수행 ○ 암호화 된 내용을 다음 암호화 수행에 전달 	
CTR(CounTeR, 카운터) 모드	<ul style="list-style-type: none"> ○ 스트림 암호, 병렬 처리 가능 ○ 각 단계별 피드백 없음, 키 스트림에 의사 난수 적용, ECB와 같이 독립적인 암호 블록 생성 함. ○ 카운터 + 1을 암호화 수행하고 평문 블록과 XOR하여 암호 블록 완성 	
Kerckhoff 원리	○ 암호 해독자는 암호 방식을 알고 있다는 것을 전제로 암호 해독을 시도한다는 원리	
스트림 암호	<ul style="list-style-type: none"> ○ 평문과 길이가 같은 키 스트림 생성, 평문과 키의 배타적 논리합으로 암호화 ○ 오류 확산 현상이 없으며, 암호화에 시간 많이 소요 	
블록 암호	<ul style="list-style-type: none"> ○ 전치와 치환을 반복해서 평문과 암호문, 키의 관계를 식별이 어렵도록 구성 ○ 혼돈과 확산으로 암호 고도화 	
혼돈	○ 암호문과 비밀키의 관계를 숨기는 기법	[돈키]
확산	<ul style="list-style-type: none"> ○ 전치와 치환을 통해 암호문과 평문의 관계를 숨기는 기법 ○ 평문의 통계적 성질 숨길 	[산문]

Key Clustering	○ 서로 다른 키가 동일한 메시지에 대해 동일한 암호문을 생성하는 현상	
ECC	○ RSA 암호화 방식의 대안으로 제안된 암호 알고리즘 ○ 코블리치와 밀리가 처음 제안 ○ 암호 방식 설계 용이하며, H/W와 S/W로 구현 용이하여 스마트카드나 무선 통신 단말 등 처리 능력 제한된 응용 분야에 효율적 ○ 전자 상거래 핵심 기술, 160bit ECC는 1024Bit RSA와 동일 보안 수준	
암호 공격		
! 암호 공격 기법	○ COA, KPA, CPA, CCA	
! COA(Ciphertext Only Attack, 암호문 단독 공격):	○ 암호문 유일	
! KPA(Known Plaintext Attach, 기지 평문 공격)	○ 몇개의 평문 + 암호문 쌍	
! CPA(Chosen Plaintext Attach, 선택 평문 공격)	○ 평문 선택 시 -> 암호문 획득 가능, 암호기 접근 가능	
! CCA(Chosen Cipertext Attach, 선택 암호문 공격)	○ 암호문 선택 시 -> 평문 획득 가능, 복호기 접근 가능	
! 차분 공격	○ 두 개의 평문 블록의 비트 차이에 대응되는 암호문 블록의 비트 차이를 이용 암호키를 찾는 방식	
! 선형 공격	○ 알고리즘 내부의 비 선형 구조를 선형화 시켜 키를 찾는 방식	
! 전수 공격	○ 암호화 할 때 일어 날 수 있는 모든 경우의 수를 조사하여 키를 찾는 방식	
! 통계적 분석	○ 평문의 각 단어 빈도를 사용, 통계적 방식을 찾는 방식	
암호 알고리즘		
! RSA	○ 소인수 분해의 어려움을 근거를 두는 대표적인 공개키 알고리즘 ○ 키의 길이는 가변적이며, 1024 or 2018 bit 이상 권장 ○ $c = m^e \bmod n$ (m:평문, e,n:공개키, d:개인키, c:암호문) ○ 원칙 : 암호/복호화는 한쌍, 키는 암호/복호 중 한쪽에 사용, 타인은 개인키 ○ 두개의 지수 e(공개용), d(비밀) 사용, $C = P^e \bmod n \rightarrow P = C^d \bmod n$, n : 매우 큰수, 키 생성 프로세스 ○ 키 생성 : p, q의 곱, 유클리드 호제법, 권장 소수(p,q)는 512 bit, 10진수 154 자리수, 모듈러 n은 2014 비트 ○ RSA 공격 : 소인수분해 공격(모듈러가 매우 큰값, 시간내 소인수분해 불가 착안), 중간자 공격	
! MD5	○ 유닉스가 기본으로 지원하는 128bit 해시함수	
! 해시 알고리즘	○ MD2, MD4, MD5, SHA-1(160), SHA-2(SHA-256, SHA-384, SHA-512), SHA3, RIPEMD, HAVAL	
! SHA-1	○ 512비트 블록에서 160비트의 출력 생성	
! SHA-256		
ECC (타원곡선알고리즘)	○ RSA 방식의 대안으로 제안된 알고리즘 ○ 키의 비트수가 작고 빨라 스마트 기기 등 용량/성능이 제한된 기기에 적용 용이	
전자 서명	○ 메시지 복원형 / 부가형(해시 함수)으로 구분	
! DSS	○ NIST에서 제안한 전자 서명 표준안, Elgamal에 기본, DSA를 핵심 알고리즘으로 사용	
Feistel 암호	○ DES 등 대부분 대칭 블록 암호가 사용하는 방식 ○ 구성 : 자기 자신을 역으로 갖는 것, 역 함수가 존재하는 것, 역함수 존재하지 않는 것 ○ 특징 : 원하는 만큼 라운드 실행, 암호/복호 알고리즘 동일 ○ DES, LOKi, CAST, Blowfish, MISTY, RC5, RC6, CAST256, Twofish, Mars	
Non-Feistel 암호	○ 역함수가 존재하는 요소만 사용, 동일한 입출력 갯수를 사용하는 S-BOX 등	

	○ feistel과 같이 평문이 반으로 분할될 필요 없음	
SPN 구조	○ 라운드 함수가 역 변환 되어 하는 제약 존재 ○ 병렬 처리가 용이하여 암호/복호화를 고속 구현 가능 ○ 입력을 여러개의 소블럭 편성, 소 블럭을 S-Box 입력 후 출력을 P-box로 전치하는 과정 반복 ○ AES(Rijndael), IDEA, SAFER, SHARK, Square, SRYPTON, Serpent	
DES ! (Data Encryption Standard)	○ 대칭키 블록 암호 ○ 평문 64bit, 키 64bit(7bit 당 오류 검출 1비트, 56bit), Feistel 네트워크 변형 구조, 16 라운드 ○ 암호화는 상호 역순 암호화 시 생성된 서브키를 복호화 시 역순을 사용 ○ 두개의 전치(P-BOX, 초기 전치, 최종 전치)와 16개의 Feistel 라운드 함수로 구성 ○ 안전성은 비선형 함수로 구성된 s-BOX에 의존 ○ 16라운드 파이스텔 구조, 대체와 전치를 반복 적용 ○ DES 함수 : 확장 P-BOX + 키 XOR + 8개의 S-BOX + 단순 P-BOX로 구성 -S-BOX : 비선형, 1Bit 변경 시 출력은 2Bit 이상 변경 ○ 취약점 : 키의 크기가 작아 문제됨(56bit), 키 전수 조사는 2의 56 키 조사 하면 해결 ○ 이중 DES : 중간 일치 공격(가지 평문 공격 등) 시 2의 57승의 키 조사 필요(보안성 개선 효과 미미)	
3DES	○ DES가 무차별 공격에 취약해짐에 따라 이를 대체하기 위해 개발 ○ 두개의 키 방식, 세개의 키 방식 존재, 키 크기 : 112bit or 168bit ○ 두개의 방식 : 암호화(K1) > 복호화(K2) > 암호화(K1) : 기지 평문 공격 가능 위험성 ○ 세개의 방식 : 암호화(K1) > 복호화(K2) > 암호화(K3) : PGP 등 많은 응용 프로그램에서 사용 ○ 모든키를 동일하게 하면 DES와 호환성 가짐	
SEED	○ 128bit key, 16 round, 64bit round key, 128 bit block, 변형 Feistel, f 함수의 비선형성 안전도 의존) ○ DES 참조	
! AES	○ 128 Bit Block, 128/192/256 Key, Non-Feistel, Rijadael 알고리즘, Triple DES보다 안전 ○ 128 bit key(10라운드), 192 bit key(12라운드), 256 bit key(14라운드) ○ 키 크기에 관계 없이 평문과 암호문의 크기는 128 bit	
ARIA	○ SPN 구조, 128bit block, 128/192/256 bit key), AES 참조	
공개키	○ 공개키 : 무결성, 부인방지, 전자 서명 구현에 활용, RSA 알고리즘 ○ 공개키(검증용 키) + 개인키(비밀키, 서명용키)로 구성 ○ 원칙 : 암호/복화키는 한쌍, 키는 암호/복화 중 한쪽에 사용, 타인은 개인키 사용 불가 ○ 2048bit 이상 키 필요, RSA, ECC, DSA ○ 취약점 : 입수 공개키가 송신자것 검증 필요, 공개키 인증이 없을 경우 중간자 공격능, 암호화 시간 과다	
기타	○ IDEA(스위스, 128bit key, 64 bit block, 8 round) 1) DES 대비 2배 속도, 무차별 대입 공격 대응, PGP 암호화 소프트웨어 사용 ○ Blowfish(대칭형 블록 알고리즘, Feistel, 32bit~448bit 키 교환) 1) 가장 빠른 알고리즘, 구현시 메모리 많이 소요 ○ OAPE(Optmal Asymmetric Encryption Padding) 1) 짧은 메시지는 짧은 메시지 공격 성공 가능, 의미 없는 데이터를 붙여 공격이 어렵도록 조치 필요 ○ Rabin : RSA의 변형, 2차 합동에 근거, 암호화 매우 간단, 빠른 속도로 스마트 카드 등 낮은 사양에 활용 ○ ElGamal : 이산 대수 문제 근거, 암호문 길이는 원문의 2배로 커짐 1) 이산 대수를 고속으로 구하는 알고리즘 없다는 사실에 근거, Diffle-Hellman이 이에 근거하여 구현	
! 해시 함수의 특성	○ 약 일방향성(Weak Onewayness) : $h(M) = H$ 에서 M 값을 찾는 것이 불가능해야 함. ○ 강 일방향성(Strong Onwayness) : $h(M) = H$ 에서 $h(M') = H$ 를 만족하는 M'를 찾는 것이 불가능해야 함. ○ 충돌 회피성(Collision freeness) : $h(M) = h(M')$ 의 서명문 쌍(M, M') ($M \neq M'$) 찾는것 불가능 해야 함.	
전자 서명		
전자 서명 조건	○ 위조 불가, 서명자 인증, 부인 방지, 변경 불가, 재사용 불가	
전자 서명 방식	○ 메시지 복원형 / 메시지 부가형	

은닉 서명	○ 전자 화폐가 되기 위한 요구 조건으로 프라이버시 보장을 위해 사용되는 특수 서명 기법 ○ 사용자의 익명성과 송신자의 익명성 보장하여 기밀성을 보장하는 방식	
부인 방지 서명	○ 자체 인증 방식을 배제, 서명을 검증할 때 서명자의 도움이 있어야 검증이 가능함 전자 서명	
이중 서명	○ 고객 결제 정보가 판매자에게 노출 가능성, 판매자에 의해 결제 정보 위/변조 가능성을 제거 서명 방식	
서명 방식		
전자 투표 요구 사항	○ 완전성, 익명성, 건전성, 이중 투표 방지, 정당성, 책임성, 검증 가능	
전자 화폐 요구 사항	○ 독립성, 프라이버시 보장, 안전성, 오프라인 사용성, 양도성	
PKI	○ 개방 네트워크에서 안전하고, 건전한 서비스 제공 목표 ○ 비밀성, 인증성, 무결성, 부인 장치 등의 기본적인 보안 서비스의 효과적인 제공	
PKI 구성 요소	○ 인증기관, 등록 기관, 디렉토리 서비스, 사용자	
LDAP	○ X.500 디렉토리 표준 ○ 쓰기보다 읽기에 최적화된 PKI 시스템의 인증 저장 ○ PKI 정보를 추가/삭제/변경을 수행하기 위한 프로토콜	
디렉토리	○ 인증서와 관련된 정보, 상호 인증서, 인증서 취소 목록 등의 정보를 저장/검색하는 역할	
전자 서명 인증서	○ X.509	
CRL	○ 인증서 폐지 목록 ○ 폐기 이전에 사용된 전자 거래 문서의 거래 타당성 증빙을 위해 사용 ○ X.509에 CRL 세부 규격 정의 됨. ○ 인증서 폐기는 본인 또는 인증 기관의 직권으로 가능	
델타 목록	○ 금번 업데이트 <> 다음 업데이트 간 변동 사항)	
간접 CRL (Indirect CRL)	○ 발행 CA가 아닌 다른 CA가 발행하거나 여러CA의 CRL을 통합	
OCSP(온라인 인증서 상태 확인 프로토콜)	○ CRL 다운없이 인증서의 상태 확인 : good, revoked, unknown	
WPKI	○ 서버/클라이언트 인증을 위한 무선 환경에 적합한 공개키 기반 구조	
SLC (short Lived Certificate)	○ 무선 인터넷 환경에서 발급되는 인증서, CRL의 갱신/목록 관리가 사실상 불가능 ○ CRL을 대신하여 인증서 폐지 메커니즘을 개발, 25/48시간의 짧은 유효 기간을 가지는 인증서	
생체 인식		
생체 인식 기술 요구	○ 보편성, 유일성, 영속성, 획득성	
 인가	○ 권한 부여의 의미, 사용자가 응용 시스템에 대한 사용 권한을 부여 받는 것 ○ 정보 전송 주체가 되는 송/수신자가 정당한지 확인하는 절차	
 인증	○ 어떤 사람, 사물이 등록된 바로 그 사람인지 판단하는 과정	
생체인증시스템	○ 참조 프로파일, 템플릿에 의존하는 인증 기술 ○ 시스템에 접근을 원하는 개인의 프로파일, 템플릿을 생성 및 저장하여 활용	
 FAR (False Acceptance Rate, 오인식률)	○ 생체 인식의 정확성을 측정 평가, 허가되지 않은 사용자가 시스템의 오류로 인해 접근이 허용되는 오류율	
 FRR (False Rejection Rate, 오거부율)	○ 인가된 사용자와 인식되지 못한 사용자간의 측정율, 허가된 사용자가 오류로 접근이 거부되는 비율	
CER (Crossover Error Rate)	○ FRR과 FAR 교차점,잘못된 거부의 비율과 잘못된 허용의 비율의 교차점	
FER (Failure to Enroll Rate, 등록실패율)	○ 생체 인식 데이터 레코드를 등록할 수 없는 사용자가 발생하는 확률 측정치	

인증		
인증 유형	○ 지식(ID/PW) > 소유(토큰/스마트카드) > 존재(생체인증) > 행위(서명, 음성, 움직임)	[지소존행]
Type1(지식)	○ 알고 있는 것 / 패스워드, 핀, i-Pin	
Type2(소유)	○ 거지고 있는 것 / 메모리카드(토큰), 스마트 카드	
Type3(존재)	○ 그를 나타내는 것 / 생체 인증, 지문, 홍채, 장문, 성문, 얼굴 이미지	
Type4(행위)	○ 그가 하는 것 / 서명, 움직임, 음성	
커버로스		
커버로스 구성	○ 사용자, KDC, TGS, AS로 구성 ○ 재전송 공격	
커버로스 작동	○ 사용자는 AS 서버를 통해 인증을 받고, 세션키로 암호화된 서비스 티켓을 받은 후 암호화된 서비스 티켓 복호화 ○ 클라이언트는 접속을 원하는 서비스에 확보한 티켓을 통해 인증 시도	
커버로스 재전송 공격 방지	○ 타임 스탬프 사용	
접근 통제		
무결성 3가지 목표	○ 비인가자의 수정 방지, 내/외부 일관성 유지(정확한 트랜잭션), 합법자의 불법 수정 방지	
접근 통제 원칙	○ 최소 권한 정책 : 알필요 원칙, 활동을 위한 최소한의 정보, 객체 접근의 강력한 통제 ○ 최대 권한 정책 : 데이터 공유의 장점 극대화, 가용성의 원리 기반	
! 접근 통제 모형	○ <u>MAC, DAC, RBAC</u> , CBAC, MLS	[MDRCM]
! MAC (강제적 접근통제)	○ 벨라파둘라 모델 근거, 객체 등급, 주체 보안 레벨로 접근 여부 설정, 주로 군사용 ○ 엄격한 보안 적용과 중장 집중식 관리 특징, 모든 접근에 대한 레이블링 필요, 상업적 환경 적용 어려움 ○ 다른 그룹의 낮은 등급 정보에는 접근 불가 함. ○ 임의적 접근 통제(DAC)보다 안전한 구조, 다단계 보안 등급 ○ 규칙 기반 정책 <u>○ 어느 하나의 주체/객체 단위로 접근 제한을 설정할 수 없음</u>	
! DAC (임의적 접근통제)	○ 한 개체가 다른 개체의 접근을 관리, ACL 사용하여 구현, 분산형 보안 관리 ○ 소속되어 있는 <u>그룹의 ID에 근거</u> 하여 객체에 대한 접근 제한을 통제하는 방식 ○ 객체 소유자의 임의적 판단에 따른 접근 권한 승인, 시스템의 전체적 보안 관리 취약 우려 ○ 윈도우, 유닉스의 파일에 대한 접근 통제 정책이 대표적 <u>○ ACL 사용하여 구현 하는 것이 일반적</u> ○ 신분기반 정책, 소속 그룹의 신분에 근거하여 객체에 대한 접근 제한	
! RBAC (역할기반 접근 통제)	○ 금융, 정보 등 특정한 일을 수행하는 조직에서 책임과 권한을 구체화 ○ 그룹은 전형적인 사용자 집합, 역할(롤)은 권한들의 집합 <u>○ 임의적 접근 통제의 단점과 강제적 접근 통제의 단점을 보완한 통제 대책</u> ○ 인사 이동이 잦은 조직에 적합한 접근 통제 정책 ○ 보안 관리와 감사를 용이하게 하는 접근 통제, 운영자는 모든 자원에 접근 가능하나 접근 권한 변경 불가 ○ Netware나 Windows NT와 같은 네트워크 운영 체제에서 볼 수 있음. ○ 편리한 관리 능력, 비기술적 정책 입안자 이해 용이, 관리 효율성, 최소 권한의 원칙 준수, 직무 분리 <u>○ RBAC의 3가지 기본 보안 정책 : 특권 최소화, 직무 분리, 데이터 추상</u> ○ RBAC 종류 : Role-Base(역할 기반), Task-Base(임무 기반-책무기 제조), Lattice-Based(상하한 정의)	
! CBAC (상황 기반 접근 제어)	○ 네트워크에서 어플과 네트워크 경계서 모든 트래픽을 통제하는 규약	
! MLS (다중 등급 보안)	○ 사용자, 프로세스 등 구성 요소에 대한 보은 수준과 업무 영역에 따른 보안 등급 및 보호 범주를 부여 하는 방식	
! Clark-Wilson 모델	○ 사용자가 직접 객체에 접근 불가, 프로그램을 통해서만 객체에 접근 가능한 모델	

	<ul style="list-style-type: none"> ○ 불법 수정 방지를 위한 보안 모델, 무결성이 유지되는 특정 시스템을 통해서만 객체에 접근 가능 ○ 상업 환경에서 적합하게 개발된 불법 수정을 방지하기 위한 보안 모델(금융 자산, 회계 등) 	
키 분배		
키 분배 알고리즘 분류	○ KEY 분배 알고리즘 / KEY 합의 알고리즘(Diffie-Hellman)	
Diffie-Hellman	<ul style="list-style-type: none"> ○ 키 합의 알고리즘, 중간자 공격에 취약한 문제 발생, 최초의 비밀키 교환 프로토콜 ○ 이산 대수 문제를 근거로 키 교환을 뒷 받침 ○ 중간자 공격에 취약한 구조 	
OTP		
OTP	○ ID/PW 스니핑 공격에 대응하기 위한 일회용 패스워드 활용 사용자 인증 기법	
동기식 / 비동기식	정리 필요	
S/Key	<ul style="list-style-type: none"> ○ 유닉스 계열의 운영 체제에서 인증에 사용, 해시 체인에 기반한 알고리즘 ○ 서버에 저장된 OTP 목록이 유출될 경우 보안이 대단히 취약해지는 문제 발생 	
시간 동기화	<ul style="list-style-type: none"> ○ OTP 생성을 위한 입력값으로 시각을 이용하는 방식 ○ 클라이언트는 현재 시각을 입력값으로 OTP 생성, 서버도 현재 시각 기준 OTP 생성 후 상호 비교 ○ 미국 RSA 사의 시큐어ID가 대표적 	
Challenge-Response	<ul style="list-style-type: none"> ○ 난수 생성 등을 통해 임의의 수를 생성하고 클라이언트에 전송하면, 클라이언트는 그 값을 OTP 생성 ○ 입력값이 매번 임의의 값이 된다는 측면에서 안정성 보장, 네트워크 모니터링에 의해 전송값 노출될 경우 매우 취약한 문제 발생 	
이벤트 동기화	○ 서버와 클라이언트가 카운트 값을 동일하게 증가, 해당 카운트를 입력값으로 OTP 생성	
기타		
워터마킹		
영지식 증명 프로토콜	한사람이 다른 사람에게 사실 증명의 어떤 정보도 주지 않고, 알고 있음을 증명하는 방법 정보를 전혀 주지 않고 상대방에게 정보를 알고 있음을 증명	
SAML (보안 보장 생성 언어)	<ul style="list-style-type: none"> ○ XML 표준 보안 표준 언어, 인증정보/권한부여정보 등을 안전하게 교환할 수 있는 표준 ○ 기업 내부, 기업 간의 SSO를 제공하고, 기업 보안 인프라에 종속되지 않는 장점 가짐 	
허니팟 구현 방법	○ Port Monitor, Deception System, Multi-Protocol Deception System	
피싱	○ 금융 기관 등을 사칭한 이메일을 통해 가장 링크로 접속 유도, 개인 정보/금융 정보를 수집하는 기법	
파밍	○ 도메인 탈취, DNS 변조 등을 통해 사기 사이트로 접속을 유도하는 기법	
혹스(Hoax)	○ 남을 속이거나 장난을 목적, 허위 바이러스 경고 메일 형태, 공신력 가진 기관 사칭, 특정 파일 삭제 유도	
은닉 채널	○ 높은 등급 주체가 낮은 등급에게 정보를 전달하는 방법, 스테가노그래피가 대표적	
은닉 채널 은닉 시간 채널	○ 자원 사용 조작 초과(cpu 사이클 등)	
은닉 채널 은닉 저장 채널	○ 어떤 저장소에 데이터 기록, 다른 프로세스가 직간접적으로 데이터 열람	
암호 사용 효과	○ 기밀성, 무결성, 메시지 인증, 사용자 인증, 부인방지	
침투 테스트 유형	○ 외부 테스트(무지식, 부분 지식) / 내부 테스트(완전 지식, 더블 블라인드-담당자 배제)	
HSM(Hardware Security Module)		

> 어플리케이션 보안






FTP		
!	FTP	<ul style="list-style-type: none"> ○ 21(명령), 20(data) 포트 사용, 클라이언트가 Active/Passive 모드 선택 ○ TimeoutSession : 지정 시간 이후 무조건 로그아웃, RootLogin : root 로그인 허용 여부 ○ /etc/ftpuser에 등록된 ID는 FTP 차단 ○ PASV 명령은 데이터 하이재킹에 이용될 수 있음 ○ ftp 디렉토리의 소유자는 ftp로 절대 금재
	FTP Log (/var/log/xferlog)	<pre># more xferlog Sun Feb 27 20:40:31 2000 6 file.test.kr 3191923 /home/user/fire.mp3 b _ o r user ftp 0 * c Sun Feb 27 20:40:38 2000 7 file.test.kr 4728392 /home/user/노래다.mp3 b _ o r user ftp 0 * c Sun Feb 27 20:40:31 2000 이 파일을 전송한 시간 6 전송 소요 시간 file.test.kr 전송한 호스트 네임 3191923 파일 크기 /home/user/fire.mp3 파일의 이름 b 전송 방식 _ special action flag Special action flag는 C, U, T, _의 값을 가지며 각 플래그의 의미는 다음과 같다. C 압축된 파일 (Compressed file) U 비압축된 파일 (Uncompressed file) T 묶인 파일(Tar'ed file) _ No action was taken o direction is the direction of the transfer. Can be one of: o outgoing i incoming d deleted r access 방식 access mode는 a, g, r의 세 가지 값을 가진다. a는 anonymous를 의미하고, g는 guest, r은 real을 의미한다. authentication 방식은 o 또는 l 값을 가지며, 여기서 o는 none을 l은 RFC931의 인증방식을 사용한다는 것을 의미한다. 완료 상태는 c, i의 값을 가지며, c는 완료된 상태, i는 불완료된 상태를 의미한다. user 사용자 이름 ftp Service방식 0 Authentication 방식 * 인증된 사용자 이름 c 완료 상태</pre>
	Proftpd 설정	<ul style="list-style-type: none"> ○ ServerType : standalone / inetd ○ RootLogin : on / off ○ MaxInstances(or MaxClients) : n ○ DefaultRoot : 자신의 홈 디렉토리 벗어나지 못하도록 설정
!	FTP Bounce Attack	<ul style="list-style-type: none"> ○ FTP 서버가 data 전송 시 목적지를 확인하지 않는 설계상의 문제점을 악용 ○ FTP Port 명령 사용하여 포트 스캐닝에 사용
!	FTP 보안 취약점	<ul style="list-style-type: none"> ○ FTP는 id/pw를 평문으로 전송, 명령어 및 데이터 또한 평문 전송 ○ FTP 세션 자체가 암호화 되지 않아 프라이버시 보호 불가
	Anonymous FTP 공격	<ul style="list-style-type: none"> ○ 패스워드 없는 FTP 서버 공격, 악성 코드 배포지로 활용 가능
!	TFTP	<ul style="list-style-type: none"> ○ UDP/69, Secure Mode 지원, 사용자 인증 관련 명령 지원, 인증 없이 파일 읽고 저장 가능 ○ TFTP 는 inetd.conf(-s 옵션은 chroot() 보안 옵션)에서 주석 처리 권장 ○ TFTP 접속 시 /TFTPboot 디렉토리만 접근 가능 ○ /etc/ftpusers 파일 무시함.
	TFTP 보안 대책	<ul style="list-style-type: none"> ○ tftp 불필요한 경우 서비스 중단 ○ 필요한 경우 secure mode로 운영 ○ '-s' 지정하여 지정한 상위 디렉토리 접근하지 못하도록 설정
	FTP 서비스 중단	○ inetd.conf 파일에서 FTP 서비스 부분의 주석 처리(#) 후 inetd 데몬 재 가동
Mail		
	메일 기본	○ MTA : 메일 메시지 저장, 전달, 전송 기능(Sendmail 등)

	<ul style="list-style-type: none"> ○ MUA : 사용자 메일 어플리케이션, 아웃룩 등 ○ MDA : 사용자에게 메일을 전달, 메일 서버에서 사용자 메일 박스로 복사하는 기능, procmail 등 ○ POP3 : 110번 ○ IMAP : 143번(IMAP4/220) ○ PGP(Pretty Good Privacy) : 	
SMTP	<ul style="list-style-type: none"> ○ SMTP : 25번(Sendmail 포트 동일), TCP ○ RCPT 명령은 수신자 지정 명령 ○ MAIL FROM : 메일 송신자 지정 ○ EHLO : 인사 및 송신자가 SMTP 확장 기능을 지원한다는 것 	
스팸 방지	<ul style="list-style-type: none"> ○ SPF(Sender Policy Framework) : 이메일 발신자의 도메인 인증 방식 ○ RBL(Real-time Spam Black Lists) ○ Sender ID ○ Sanitizer : 이메일 공격 효과적 대응 	
SPF (Sender Policy Framework, 메일 서버 등록제)	<ul style="list-style-type: none"> ○ 메일 서버 정보를 사전에 해당 도메인 DNS에 공개 등록, 수신자는 발송자 정보가 실제 메일 서버의 정보와 일치하는 확인 ○ 발신자 : 자신의 메일 서버 정보와 SPF 레코드를 DNS TXT 레코드에 등록 ○ 수신자 : 발송자의 DNS에 등록된 SPF 레코드 확인하여 발송자 IP와 대조 후 결과에 따라 메일 수신 	
SPF 정책	<ul style="list-style-type: none"> ○ Hard Fail(-) : 인가된 메일 서버로 부터 메일이 아니면 거부 ○ Soft Fail(~) : 인가도니 메일 서버에서 전송된 메일은 아니지만 기록 남기고 통과 ○ Neutral(?) : 중립(위/변조 판단하지 않음) ○ Positive(+) : 무조건 통과(정책 미설정 시 디폴트 설정) 	
SPF 정책 확인	○ nslookup > set type=txt > 이후 도메인 조회	
DomainKeys	<ul style="list-style-type: none"> ○ 야후에서 개발한 스팸 메일 관리 기술, 메일 헤더를 암호화 후 해시값 생성 ○ 해시값은 메일 헤더에 기록, 받은 메일 서버는 이를 검증 후 변조 여부 확인 	
DKIM	○ 야후의 DomainKeys와 시스코의 IIM 결합 기술	
PGP	<ul style="list-style-type: none"> ○ 필 짐머만이 개발, 키 관리는 RSA, 데이터 암호화는 IDEA ○ 메시지의 무결성 보장 ○ 메일 암호화/복호화 프로그램, 메일 수신 시 송신자의 신원을 확인하여 메시지 미 변경을 확신 ○ PKI 표준과 일치하지는 않음 ○ 키링을 통해 공개키 디렉토리를 공유, 대규모 전자 상거래에는 부적합. ○ 기밀성, 무결성, 인증 및 부인 방지 제공을 위해 디지털 서명 및 공개키 사용 ○ 키 관리의 그래픽 인터페이스 지원 ○ 공개키는 4096 bit 까지 지원, RSA, DSS/Diffie-Hellman 공개키 지원 ○ 공개키 서버와 직접 연결되어, 공개키 분배 및 취득이 간편 ○ 세션키는 공개키로 암호화 하고, 메시지는 대칭 암호화로 암호화 	
PEM	<ul style="list-style-type: none"> ○ 암호화, 디지털 서명, 상호 운영성 목표, 암호화/인증/무결성/부인방지 지원 보안 메일 표준 ○ 송수신측이 상대방을 상호 인증 하는 방식, CA를 통한 인증, 인증서 교환 인증 등 지원 ○ 중앙 집중식 관리 	
S/MIME	<ul style="list-style-type: none"> ○ X.509 인증서 표준 준수, RSA 기반, 송신부인(DSA), 기밀성(3DES), 무결성(SHA-1) ○ signedData : 서명데이터, 무결성 보장 ○ envelopedData : 봉인된 데이터, 비밀성 보장 ○ digestedData : 무결성 ○ encrypedData : 암호화된 데이터 ○ authenticateData : 데이터 인증 	
폭탄 메일	○ 특정인, 특정 단체에 다량의 메일을 일시에 보내는 공격, 다량의 첨부 및 바이러스 전송	
MTA	○ 한 호스트로 부터 메일을 받아 다른 호스트로 메일을 전달하는 역할	
RBL (Real Time Block List)		
메일 서비스 작동 방식 3가지	<ul style="list-style-type: none"> ○ RELAY : host에 지정된 메일의 수/발신을 허용 ○ REJECT : relay와 반대로 송/수신을 허용하지 않음. 	

	○ DISCARD : 메일 받은 후 폐기, 발신자에게 폐기 통보 하지 않음.	
메일 설정 파일	○ sendmail.cf	
DRAC (Dynamic Reply Authorization Control)	○ 동적 IP에서 메일 Relay를 지원하기 위한 방식	
! 스팸 방지 도구	○ Procmal, Sanitizer, Inflex, SpamAssassin	
! 스팸 방지 도구 Procmal	○ Sendmail.cf에 포함, 사용자 디렉토리 forward 파일에 위치, 플러그인 형식 ○ 받은 메일에서 보낸 사람, 제목, 크기, 내용 등 기준으로 메일 필터링 가능	
! 스팸 방지 도구 Sanitizer	○ 이메일 이용 공격에 효과적인 대응하는 Procmal Ruleset ○ 확장자를 통한 필터링, 오피스 메크로, 악성 메크로, 감염 메시지의 보관 장소 설정 기능 제공	
! 스팸 방지 도구 Inflex	○ 메일 서버에서 나가는 메일에 대한 Inbound / outbound 정책 세워 메일 필터링 가능(첨부 파일만 가능)	
! 스팸 방지 도구 SpamAssassin	○ Apache 그룹을 통해 오픈 소스로 진행, Perl로 개발, 헤더 내용, 화이트/블랙 리스트, RBL 등 필터링 지원	
HTTP		
S-HTTP	○ EIT에서 제안, WWW 보안 강화, 암호화/서명 등의 보안 서비스 제공하는 프로토콜	
상태 코드	○ 100(Continue) ○ 200(OK) ○ 201(Created) : PUT 메소스 전형적 응답 ○ 202(Attepted) : 요청 접수, 아직 미처리 ○ 203(Non-Authoritative) : 요청 성공, 일부 자원은 제 3자로 부터 ○ 204(No Content) : 요청 성공, 콘텐츠는 미 제공 ○ 300(Multiple Choices) ○ 301(Move Permantly) ○ 400(Request Syntax Error, Bad Request) ○ 401(Unauthorized) ○ 403(Forbidden) ○ 404(Not Found) ○ 500(Server Error)	
주요 웹 취약점	○ Injection, XSS, CSRF ○ 취약한 인증 및 세션 관리, 안전하지 않은 객체 직접 참조, 보안상 잘못된 구성 ○ 민감 데이터 노출(개인 정보, 금융 정보 등), 기능 레벨 통제 누락, 알려진 취약점 구성 요소 사용 ○ 검증되지 않은 리다이렉트와 포워딩	
XSS cs CSRF	○ 두 공격 기법의 비교 설명, p463	
cache-Control	○ no-cache - 원본 서버(웹서버)에서 무조건 다시 읽어서 응답하라는 의미 ○ max-age=0 - 원본 서버와의 동일성 유무에 유무에 대해 매번 체크하라는 의미	
CC-Attack	○ user-agent의 cache-control를 조작하여 웹 서버 및 DB 서버에 부하를 발생시키는 공격 기법 ○ cache-control 부분에 no-store, must-revalidate(no-cache 보다 강한 의미) 포함됨	
SQL-Inject	○ Error 기반, 논리적 에러 기반, Union 기반, Blind 기반 ○ 공격 샘플 코드 리뷰 및 설명 연습	
SQL-Injection 방어	○ 파라메터 값 검증 강화 ○ 특수 문자 사용 금지 : 싱글/더블 쿼테이션, 공백, --, # 등 사용 금지 ○ AD-HOC쿼리 사용 지양	
! Blind SQL-Inject	○ 조건절의 참/거짓에 따른 응답의 차이를 이용한 공격 기법, 참/거짓에 결과에 따른 웹 서버의 응답을 통해 DB 데이터 추출	
웹셸		
! 웹 Proxy	○ 웹 해킹 시 가장 많이 사용되는 도구, 클라이언트 요청을 가로채 필터링 우회, 서버 전송 데이터 조작 등 ○ 관련 툴 : Paros, Burp	


Access_log	○ 웹 서버 접근 내역에 대한 로그	
agent_log	○ 웹 브라우저 이름, 버전, O/S 등에 대한 로그	
error_log	○ 없는 페이지 요청, 실행시 오류 등에 대한 로그	
아파치 환경 설정	○ /usr/local/apache/conf/httpd.conf	
아파치 환경 설정 파라미터	○ ServerType standalone ○ HostnameLookups off ○ MaxClienots 512 ○ ServerAdmin : 관리자의 이메일 주소 지정 ○ PidFile : 웹 서버 프로세스의 PID를 기록하는 파일, 기본값은 logs/httpd.pid ○ MaxKeepAliveRequest 000	
! 아파치 서버 정보 숨김 처리	○ Prod[uctOnly] : 웹 서버 종류만 제공 ○ min[imal] : 웹 서버 정보 + 웹 서버 버전 ○ OS : 웹 서버 정보 + 웹 서버 버전 + O/S 정보 ○ Full : 웹 서버 정보 + 웹 서버 버전 + O/S 정보 + 설치된 패키지 목록	
아파치 DNS 역검색 차단	○ DNS 역검색 로깅 기능을 로깅 시 많은 부하 및 성능 저하의 원인 ○ 이를 차단하는 설정 : HostnameLookups Off	
웹 개발 보안 기본 원칙	○ 최종 통제는 서버에서 수행 ○ 중요 정보 전송 시 POST method 및 SSL 적용 ○ 중요 정보를 제공하는 페이지는 no-cache 설정	
HTTP Request Type	○ GET ○ HEAD : 요청 받은 정보 피드백 없음 ○ POST ○ OPTIONS : 요청 받은 리소스의 가능한 통신 옵션 정보, 웹 서버에서 사용 가능한 동사 목록 요청 ○ PUT ○ DELETE : 명시된 리소스 삭제 요청 ○ TRACE : 루프백을 위한 요청 송신	
! 디렉토리 리스팅 취약점 방지 (Apache / IIS)	○ 디렉토리 리스팅 취약점 : 백업 파일, 환경 설정 파일, 웹 응용 프로그램 구조 노출 등의 취약점 발생 ○ Apache : httpd.conf 파일에서 indexes 옵션 제거 ○ IIS : IIS 설정 화면에서 디렉토리 검색 옵션 해제	
SSL		
BEAST (Browser Exploit Against SSL/TLS)	○ 사용자 브라우저의 취약점 이용, HTTPS 쿠키를 훔쳐서 HTTPS 세션을 가로채 수 있는 공격	
CRIME Attack (Compression Ration Info-Leak Mass Exploitation)	○ HTTPS의 압축에 따른 취약점을 이용하여 HTTPS 쿠키를 훔쳐서 HTTPS 세션을 가로챌 수 있는 공격	
SSL 프로토콜	○ HandShake Protocol, Change Cipher Spec Protocol, Alert Protocol, Record Protocol	H-CAR
Handshake Protocol	○ 클라이언트/서버 간 암호화 통신을 위한 준비 단계에서 사용하는 프로토콜 ○ 비밀키 알고리즘, 공개키 알고리즘, 메시지 다이제스트 알고리즘 정의 ○ 세션키 생성하여 서버 및 클라이언트 공유(데이터 암호화에 사용)	
Change Cipher Spec Protocol	○ 협상된 암호화 스펙을 적용하는 신호 ○ 암호 통신 설정 후 클라이언트/서버가 handshake를 다시 시도하여 암호 스펙 변경 가능	
Alert Protocol	○ 통신 양단에 오류 / 주의 발생 시 상대방에게 통보하는 프로토콜	
Recode Protocol	○ 상위 프로토콜의 내용을 감싸고, 내용 인증을 위한 MAC 추가 및 암호화 기능 제공 ○ 상호 인증을 위한 상호 인증서 교환 및 검증 등의 역할 수행	

SSL 작동 과정	<p>The diagram illustrates the SSL handshake process between a Client and a Server. The process is divided into three main phases: 초기 협상 단계 (Initial Handshake), 서버 인증 단계 (Server Authentication), and 클라이언트 인증 단계 (Client Authentication). The steps are as follows: 1. Hello Request (Server to Client), 2. Client Hello (Client to Server), 3. Server Hello (Server to Client), 4. Server Certificate or Server Key exchange (Server to Client), 5. Certificate Request (Server to Client), 6. Server Hello Done (Server to Client), 7. Client Certificate (Client to Server), 8. Client Key Exchange (Client to Server), 9. Certificate Verify (Server to Client), Change Cipher Specs (Client to Server), 10. Finished (Client to Server), Change Cipher Specs (Server to Client), 11. Finished (Server to Client). Finally, Application Data is sent from the Client to the Server.</p>	
SSL 핸드셰이크 과정	○ 초기 협상 단계 > 서버 인증 단계 > 클라이언트 인증 단계 > 종료 단계	
SSL vs TLS	○ SSL : 넷스케이프에서 개발 ○ TLS : SSL 3.0을 기초로 개발, TLS 1.1은 대칭 암호 알고리즘으로 AES 추가	
SSL / TLS	○ 사용 포트 : 443 ○ 사용자 상호 인증, 데이터 기밀성, 메시지 무결성 제공 ○ 기밀성 위해 DES, RC4 사용 ○ TLS 핸드 셰이크 프로토콜은 클라이언트와 서버의 암호 통신에 사용할 알고리즘과 공유키를 결정 ○ 특정 암호 알고리즘에 의존하지 않음 ○ Transport 계층 ~ Application 계층 사이 작동 ○ SSL 2.0은 중간자 공격 취약, SSL 3.0은 해시값 유지, 공격 막을 수 있음. ○ 무선 환경에서는 WTLS 사용 ○ 연결 시 연결 정보와 세션 정보로 나뉘며, 세션 정보는 재 사용, 연결 정보는 매번 재 생성 ○ 세션키(대칭키)는 수신자의 공개키로 암호화	
DNS		
DNS 보안 취약점	○ Zone Transfer 취약점 ○ Dynamic Update ○ DNS 코드 취약점 ○ Address Spoofing	
Zone	○ 도메일과 IP가 보관된 DB 파일	
DNS Cache Poisoning	○ 취약한 DNS 서버에 조작된 쿼리 전송, DNS 서버의 주소 정보를 임시적으로 변조하는 공격 ○ DNS의 캐시 정보를 위조하여 위조된 사이트로 접속하게 하는 공격 기법	
Resolving DNS (Cache DNS)	○ 이름 풀이 결과를 일정 기간 별도로 저장, 같은 요청이 올 경우 저장해둔 정보로 응답하는 DNS 서버	
DNS 설정 파일	○ /etc/named.conf	

DNS 서버 등록	○ /etc/resolv.conf	
DNS 레코드	○ SOA, NS, A, MX, CNAME, PTR	
DNS 레코드 SOA(Start of Authority)	○ zone의 등록 정보 제공	
DNS 레코드 NS(Name Server)	○ 네임 서버 리소스 레코드로 DNS 서버 나열	
DNS 레코드 A(Address)	○ 도메인에 IP 주소를 부여해 주는 레코드	
DNS 레코드 MX(Mail Exechage)	○ 메일 인스턴서로 메일 서버를 설정하는데 중요한 레코드	
DNS 레코드 CNAME(Canonical Name)	○ 하나의 IP에 여러 개의 별칭을 가질 수 있게 해주는 레코드	
DNS 레코드 PTR(Pointer)	○ IP Address --> Hostname으로 변환	
Resolving DNS 서버	○ 클라이언트 요청 시 자신의 캐시에서 응답, 없을 경우 계층적으로 질의하여 그 결과를 리턴	
Authoritative DNS 서버	○ 위임 권한을 가지고 있는 도메인에 대해 응답하는 DNS 서버	
DHCP	○ 동적인 IP 할당을 위한 프로토콜 ○ randge dynamic-bootp 121.212.12.1 121.212.12.99	
DNS Spoofing 방어		
Database		
	○ 주체 : 객체나 데이터에 대한 접근을 요청하는 능동적인 객체, ○ 객체 : 접근 대상이나 될 수동적인 개체, 혹은 행위가 일어날 아이템 ○ 접근 제어 매트릭 : Table, Access Lists, Capability lists, Authority item list ○ ms sql 기본 인증 : window dlswwd ahem ○ oracle port : 1521 ○ 부적절한 접근 방지, 추론 방지, 데이터 무결성, 감사 기능, 사용자 인증	
DB 주요 보안 문제	○ 추론 : 통계성 데이터에서 개별적인 데이터 항목을 추적, 질의제한, 조회 데이터 한정, 일관성 부재 ○ 집합 : 데이터의 조각에 개별적 접근 후 전체 조각을 완성하는 문제	
DB 보안 강화 도구	○ Onstat : 공유메모리 정보와 서버 통계 정보 제공 ○ Oncheck : 디스크 공간 체크, 수리 ○ Ontape : 백업과 복구	
 GRANT	○ 사용자에게 데이터베이스 객체 에 대한 권한 부여 ○ GRANT 권한(ex SELECT 등) ON DB 객체 TO 사용자	
 REVOKE	○ 사용자에게 부여된 데이터베이스 객체 권한을 취소 ○ REVOKE 권한 ON DB 객체 FROM 사용자	
 DENY	○ 사용자에게 데이터베이스 객체에 대한 권한 금지(GRANT와 중복되는 경우 DENY 우선) ○ DENY 권한 ON DB 객체 TO 사용자	
개발 보안		
 7가지 유형	○ 입력 데이터 검증 및 표현 ○ API 악용 : 표준 API의 잘못된 사용으로 발생하는 보안 취약점 ○ 보안 기능 : 인증, 접근제어, 권한 관리, 암호화 등 보안 기능과 관련한 취약점 ○ 시간 및 상태 : 멀티 프로세스 / 스레드에서 발생할 수 있는 보안 취약점 ○ 에러 처리 : 에러 처리와 관련된 보안 취약점 ○ 코드 품질 : 어플리케이션 안전성 및 신뢰성 확보 위한 소스 코드 품질 관련 보안 취약점 ○ 캡슐화 : 어플리케이션이 다른 값을 참조할 때 발생하는 보안 취약점	[입A보시에 코랩]
 외부 명령 실행	○ exec(), system(), shell_exec()	

전자상거래		
전자 상거래 기술 요건	○ 부인 방지, 인증, 프라이버시	
전자 지불 기술 요건	○ 거래 상대방 신원 확인, 전송 내용의 비밀 유지, 문서 위/변조 방지, 거래 정보 접근 통제 ○ 전자 지불에 대한 불추적성, 분할성, 익명성 보장	
전자 화폐 안전성 요구 사항	○ 익명성, 오프라인성(은행배제), 양도성, 분할성, 독립성, 복사 및 위조 방지, 익명성 취소	
SET (Secure Electronic Transaction)	○ VISA & Master Card, 인터넷 상거래 저넷 결제, 공개키 기반 구조 ○ 이중 서명 : 고객의 지불 정보는 상점이 모르고, 주문 정보는 은행이 모르도록, 고객의 프라이버시 ○ 디지털 서명, 해시 함수, 공개키 암호 적용, 영지식증명 프로토콜 미적용 ○ 상점/고객 별도의 소프트웨어 필요, 지불게이트웨이는 별도의 h/w, s/w 요구	
ebXML	○ 구성 요소 : 비즈니스 프로세스, 핵심 컴포넌트, 등록 저장소, 거래 당사자, 전송/교환 패키징 ○ 재활용 수준을 문서 수준과 시나리오 수준까지 확대 적용 가능 ○ XML 보안 기술 : XML 전자 서명, XKMS, SAML(인증,속성,승인), XACML(접근제어)	[비핵등거선]
XML		
Web Service	○ SOAP / WSDL / UDDI 구성	
SOAP	○ 웹 서비스 호출 후 그 결과를 전달 받을 때 사용되는 메시징 프로토콜	
WSDL	○ 해당 서비스에 대한 상세한 설명이 포함되어 있는 서비스 기술서	
UDDI	○ 웹 서비스를 등록하고 검색할 수 있는 레지스트리	
기타		
Unix 버퍼 오버플로 우 공격 방지	○ /etc/system, set noexec_user_stack = 1	
소프트웨어 무결성 산출 공식	○ 무결성 = SUM [(1 - 위협) + (1 - 보안)] ○ 위협 : 특정한 공격 유형이 주어진 시간 내에 발생할 확률 ○ 보안 : 특정한 유형의 공격을 물리칠 수 있는 확률	[위보]
! CVE (Common Vulnerability & Exposures)	○ 보안 취약점 표기 방법 ○ CVE - 2014 - 4321	
나선형 개발 모델 (Spiral)	○ 시스템 개발 시 생기는 위험을 최소화하여 관리하고자하는 것이 주된 목적인 개발 방법론 ○ 반복적/점증적 생명 주기 모델 -> 계획 수립 > 위험 분석 > 구축 > 고객 평가	
TCPWarpper 설정	○ <u>Telnet 서비스 접근 제어 설정</u> - <u>inetd.conf 파일에서 TELNET 서비스 항목의 실행 경로 부분을 in.telnet --> in.tcpd로 수정</u>	

> 시스템 보안

운영체제 일반		
O/S 입출력 방식 [직접 제어 방식]	○ 데이터 이동에 CPU가 관여하는 방식 - Polling 입출력 방식 : CPU가 I/O에 관여, CPU 시간 낭비 발생 - 인터럽트 입출력 방식 : I/O 발생 시 CPU 다른 일 진행 가능, I/O 종료 시 인터럽트 신호 보냄	
O/S 입출력 방식 [간접 제어 방식]	○ CPU 독립적으로 제어하는 방식 - DMA : CPU를 통하지 않고 고속의 별도 채널 구성, CPU Bus 이용 / Cycle Stealing - 채널 제어기 : 입출력 전용의 제어기, PPU 또는 채널로 명명	
로더(Loader)	○ 어떤 프로그램 실행 위해 해당 목적 프로그램을 메모리에 적재하고, 배치주소를 옮기는 프로그램	
링커(Linker)	○ 프로그램 실행 가능한 상태로 만들기 위해 목적 모듈간의 상호 참조를 해결 ○ 여러개의 목적 모듈을 하나로 만들기 위해 사용되는 프로그램	
커널	○ 메모리 접근, 프로세스 생성 및 관리, 입출력 디바이스 관리, 파일 관리, 서비스 제공	
운영 체제 보안을 위한 분리 방법	○ 물리적, 시간적, 논리적, 암호적 분리	[물시논암]
운영 체제 보안 기능	○ 메모리 보호, 파일 보호, 접근통제, 사용자 인증	[메파접사]
MBR	○ 디스크의 첫번째 파티션에 생성 시 만들어지며, 항상 디스크의 첫번째 섹터에 위치 ○ 디스크의 파티션 테이블과 부팅에 필요한 작은 실행 파일 저장 ○ 512byte 구성 : 446Byte(실행 파일) + 64Byte(파티션 테이블) + 2Byte(시그니처) + 0x55AA	
운영 체제의 기능적 분류	감시 프로그램, 작업관리프로그램, 데이터관리 프로그램	[감작데]
 리다이렉션	○ STDERR : 2 : 표준 에러, 2 >> ○ STDOUT : 1 : 표준 출력 1 >>	
이중 모드 (Dual Mode)	○ 사용자 모드 : 실행중인 프로그램의 오류가 다른 프로그램에게 영향을 주지 않은 제한적인 모드 ○ 모니터 모드 : 다른 프로그램에 영향을 줄 수 있는 하드웨어 제어 등의 특권 명령 등을 사용하는 모드	
운영체제 보안		
운영 체제 보안을 위한 분리 기법	○ 물리적 분리, 시간적 분리, 논리적 분리, 암호적 분리 ○ 구현 복잡도가 낮음에서 높은 순서 : 물리 > 시간 > 논리 > 암호	[물시논암]
물리적 분리	○ 사용자별 별도의 장비만 사용하도록 제한하는 방법, 강한 형태의 분리 ○ 서로 다른 보안 수준을 요구하는 프로세스 별 분리	
시간적 분리	○ 프로세스가 동일 시간 대 하나씩만 실행되도록 구성, 동시 실행으로 발생하는 문제를 제거 ○ 프로세스를 서로 다른 시간에 운영하는 방법	
논리적 분리	○ 미 허용된 객체의 영역에 대한 접근을 제어하는 방법 ○ 프로세스에게 논리적 구역 지정, 구역 내 활동은 자유로우나, 구역 밖의 활동은 엄격한 제한	
암호적 분리	○ 다른 프로세스가 인식할 수 없는 방법으로 자신의 데이터를 감추는 방법 ○ 내부에서 사용되는 정보를 외부에서 알 수 없도록 암호화 조치	
보안 장비		
ESM (Enterprise Security Management)	○ 방화벽, 침입탐지시스템, 침입 방지시스템 등 각종 보안 시스템을 로그를 통합 관리 ○ 보안 장비의 수가 늘어 남에 따라 관리, 인력 등의 어려움을 해결하기 위한 통합 시스템	
기타		
보안 일반		
데이터 보안 등급	○ 민감 > 기밀 > 비밀 > 대외비 ○ 민감 : 데이터의 무결성 요구 ○ 기밀 : 정보의 공개가 제한되는 등급	[민기비대]

	<ul style="list-style-type: none"> ○ 비밀 : 조직의 내부 사용자에게만 허용되는 정보 ○ 대외비 : 정보 공개 시 우위가 약화될 수 있는 기술에 해당되는 등급 	
/bin/false		
/sbin/nologin		
/etc/passwd	<ul style="list-style-type: none"> ○ UID < 100, UID > 6000 이상은 시스템이 사용하는 로그인 필요 없는 계정 의미 ○ 해당 계정은 셸 지정 컬럼에 /sbin/nologin 또는 /bin/false 를 지정하여 로그인 차단 <div> <div>tux:x:1001:100:The Linux penguin:/home/tux:/bin/bash</div> <div> <div>Standard shell</div> <div>Home directory</div> <div>Comments field</div> <div>GID of primary group</div> <div>UID</div> <div>Password</div> <div>User name</div> </div> </div> <p>Figure 7-2</p>	
/etc/shadow	<ul style="list-style-type: none"> ○ <div> <div>Username</div> <div>user created since unix epic time</div> <div>Forever lasting Account</div> <div>Days before the password will expire</div> <div>Days before password can be changed</div> <div>Password Placeholder</div> </div> <div>testuser1:!!:15670:0:99999:7:::</div>	
	○ 다중 프로그래밍 제어 기술 정리(임계 영역, 세마포어 등)	
접근 통제		
유닉스 파일 무결성 검증	○ 일반적으로 MD5 Check Sum을 사용	
참조 모니터		
참조모니터	<ul style="list-style-type: none"> ○ Secure O/S의 핵심 모델 ○ 운영 체계 매커니즘 중 접근 및 사용하는 주체의 각 요청을 점검, 보안 정책에 따른 요청임을 보증 ○ 접근 통제 모델에서 정보를 사용하는 주체가 객체에 접근하는 규칙을 통제하고 감사하는 것 ○ 주체의 객체에 대한 접근 통제를 담당하는 추상 머신 ○ 보안 커널 데이터베이스를 참조하여 객체에 대한 접근 허가 여부를 결정해야 하는 것 	
참조 모니터 역할	○ 객체에 주체가 접근할 때 발생하는 모든 시스템 콜 및 행위를 모니터링하는 것	
SKDB (Security Kernel Database)	○ 접근 허가 여부를 결정할 때 참조 모니터가 참조하는 DB	
참조 모니터 3가지 규칙	<ul style="list-style-type: none"> ○ 부정 조작 불가능 : 항상 부정 조작이 없어야 함. ○ 우회 불가능 : 항상 무시되지 않고 호출되어야 함. (직접 접근 가능한 경우 발생) ○ 검증 가능성 : 모든 동작에 대해 항상 분석 및 테스트 통해 확인 	

프로세스		
프로세스	○ CPU에 의해 수행되는 시스템 및 사용자 프로그램	
PCB	○ 프로세스 이름, 상태, 소유자, 실시간 통계, 스레드, 관련 프로세스 및 자식, 주소 공간, 자원, 스택	
준비 상태(Ready State)	○ 프로세스가 기억 장치를 비롯한 모든 필요한 자원을 할당 받은 상태에서 프로세서를 할당 받기 위해 기다리고 있는 상태	
보류(Pending) 상태	○ 프로세스의 작업이 일시 중지되거나, 디스크에 수록된 상태	
실행(Running) 상태	○ 프로세스가 CPU를 차지하고 있는 상태	
대기(Blocked) 상태	○ 프로세스가 CPU를 차지하고 실행되다가, 입출력 처리와 같은 사건이 발생하면 CPU 양도, 입출력 완료 시까지 대기 큐에 대기하는 상태	
완료(Terminated) 상태	○ CPU를 할당 받아, 주어진 시간 내 완전히 수행을 종료한 상태, 프로세스 제거, PCB 삭제됨	
좀비 프로세스	○ 코드, 자료, 스택, 세그먼트 등 아무것도 가지지 않은 프로세스, 프로세스 테이블에는 계속 존재 ○ 모든 프로세스는 좀비 프로세스가 된 후 정상 종료 됨. ○ 프로세스는 종료 시 부모에게 알리고, 부모가 확인해야 정상 종료됨. 아닌 경우 좀비 ○ 좀비가 생기는 이유는 부모 프로세스의 결함, 오류, 시그널 처리 미숙 ○ 종료 프로세스는 [좀비] 상태로 전환 시 소유한 자원 반환, 프로세스 테이블 제외 자신의 문맥 파괴	
CPU 스케줄링		
CPU 스케줄링	○ CPU 스케줄링 정리 필요	
비선점 CPU 스케줄링	○ FIFO : 먼저 들어 오면, 먼저 처리 ○ SJF(SPF) : 가장 짧은 작업 우선 처리 ○ HRN : SJF 개선, 긴/짧은 작업 불평등 해소, 우선 순위 = (대기+서비스 받을 시간)/서비스 받을 시간	
선점형 CPU 스케줄링	○ RR : Time Quantum 단위 CPU 할당 ○ SRT : 가장 짧은 시간을 소요할 작업을 우선 처리 ○ MLQ : 5개의 독자 큐 구성(시스템, 편집, 대화형 등), 각 큐별 독자적인 스케줄링 알고리즘 ○ MFQ : 각 단계큐별 우선 순위를 낮춤, 제일 마지막 큐는 RR 운영, 유닉스 채택 알고리즘	
CPU 스케줄링 라운드 로빈	○ 스케줄링 시간 할당량의 커지면 FIFO 유사 ○ 시간 할당량의 작아지면 문맥 교환 부담 커짐 ○ 각 프로세스는 동일한 CPU 시간을 가짐 ○ 다중 프로그래밍, 시분할 방식 효과적 ○ 할당 크기는 효과적인 동작에 절대적 영향	
페이지 교체 기법		
FIFO (First In First Out)	○ 타임스탬프, 가장 먼저 들어온 페이지 교체	
LRU (Least Recently Used)	○ 각 페이지당 카운터, 가장 오랫동안 사용되지 않은 페이지 교체 ○ 국부성(지역성, 최근의 상태가 가까운 미래 척도), 시간 오버헤드(시간 기록)발생, 구현 복잡 ○ 불러왔던 시간 관리로 인해 오버헤드 발생, 실제 구현하기 어려움	
LFU (Least Frequently Uesd)	○ 호출 빈도가 가장 낮은 페이지를 교체 ○ 최근에 로드된 페이지가 교체될 가능성이 높음	
NUR (Not Used Recently)	○ LRU의 단점을 개선한 모델, 2개의 관리 비트로 교체 페이지 관리 ○ 참조 비트 : 미 참조(=0), 참조(=1) ○ 변형 비트 : 내용 미 변경(=0), 내용 변경(=1)	
Clock 기법	○ FIFO와 최소 사용 알고리즘을 결합 ○ 제거하기 전 한번의 기회를 더 줌 ○ 참조 비트가 소거된 페이지는 최근 참조되지 않았음을 의미	
OPT(최적 교체)	○ 앞으로 가장 오랫동안 사용되지 않을 페이지 찾아 교체	
SCR(Second Change Replacement)	○ FIFO의 단점을 보완한 기법 ○ 가장 오래 머문 페이지 중 자주 사용 페이지가 교체되는 FIFO의 단점을 보완한 기법	



무작위 방식	○ 임의의 페이지를 교체하는 기법	
메모리 관리		
단편화	○ 내부 단편화 : 분할을 사용하고 남은 일부분, 세그먼테이션 기법으로 해결 가능 ○ 외부 단편화 : 분할이 작아 프로그램을 탑재할 수 없는 경우, 페이징 기법으로 해결 가능	
기억장치 분할	○ 페이지고정/외부 페이지(고정 블록) : 블록 사이즈가 고정된 방식, 기계적인 페이지 할당 - 내부 단편화 발생 가능(프레임의 정수배로 할당, 외부 단편화 미 발생) - 페이징은 기법은 주기억 장치 오버헤드 발생, 페이지 사상표 보관 장소 필요 ○ 세그먼트(가변 블록) : 내부 단편화 미 발생, 외부 단편화 발생 ○ 페이징 / 세그먼트 혼용 : 세그먼트 단위 분할 후 페이지 단위로 다시 분할 - 주기억 장치에는 페이지 단위 적재, 가상 주소는 $v=(s,p,d)$ 형태로 표현	
스왑 (Swap)	○ 메모리는 운영 체제의 주요 부분과 응용 프로그램, 사용 중인 데이터를 저장하고 있다. 이 메모리가 부족할 경우 사용되는 영역 ○ 유닉스에서 swap 공간을 할당하는 설정 파일 위치는 <code>/etc/vfstab</code>	
버퍼링	○ 주기억 장치를 버퍼로 활용하는 기법	
스폴링	○ 디스크를 버퍼처럼 활용하는 기법	
스래싱	○ 한번에 한 프로세스만 자원 사용 가능, 다른 프로세스는 임계구역 밖에서 대기상호배제 프로세스 처리 시간 보다 페이지 교체 시간이 더 많아지는 현상 ○ 원인 : 페이지 부재로 프로세스 실행에 필요한 수만큼 충분한 페이지를 갖지 못해 발생 ○ 해결 방안 : 다중 프로그래밍 정도를 낮추거나, 주 기억 장치내 워킹세트를 유지	
지역성 (Locality)	○ 지역성 : Denning 교수의 증명한 프로세스의 계산 영역 참조가 밀집되는 현상 ○ 시간 지역성 : 최근 참조된 기억장소가 계속 참조될 가능성이 높다. ○ 공간 지역성 : 기억 장소가 참조되면 인근의 기억장소가 참조될 가능성이 높다.	
Working set	○ 하나의 프로세스가 자주 참조하는 페이지들의 집합	
멀티스레드		
교착 상태 (dead lock)	○ 두개의 컴퓨터 프로그램이 상대방 자원을 접근하는 것을 서로 방해하며, 서로 기다리는 상황 ○ 다중 프로그램 시스템에서 아무리 기다려도 발생하지 않은 사건을 기다리는 상황 ○ 둘이상의 프로세스가 서로 원하는 자원을 점유, 상대방의 자원을 요구하며 대기하는 상황	
교착 상태 4가지 필요 조건	○ 상호배제 ○ 점유와 대기 ○ 비선점 ○ 환형 대기	[상점비환]
교착 상태 해결 방안	○ 예방 ○ 회피 ○ 발견(탐지) ○ 복구	[예피발복]
Semaphore [보충 필요]	○ 운영체제의 자원을 경쟁적으로 사용하는 다중 프로세스에서 행동을 조정하거나 또는 동기화 시키는 기술	
	다중 프로그래밍 제어 기술 정리(임계 영역, 세마포어 등)	
문맥 교환	○ 실행 상태의 프로세스를 다른 프로세스로 교체하는 과정 ○ 보통 특정 프로세스와 관련된 정보들의 총집합	
악성코드		
웜	○ 독립적 실행 가능하며, 네트워크를 연결을 따라 기기들을 옮겨 다닐 수 있는 프로그램 ○ 데이터를 파괴하고, 엄청난 양의 컴퓨터 및 통신 자원 사용	
트로이 목마	○ 정상적인 기능을 가장한 프로그램 ○ 프로그램 내 숨어 의도하지 않은 기능 수행 ○ 백 오리피스가 대표적	
키로거 공격	○ PC에 몰래 설치해서 사용자 키보드, ID, PW 등의 중요한 개인 정보 수집	

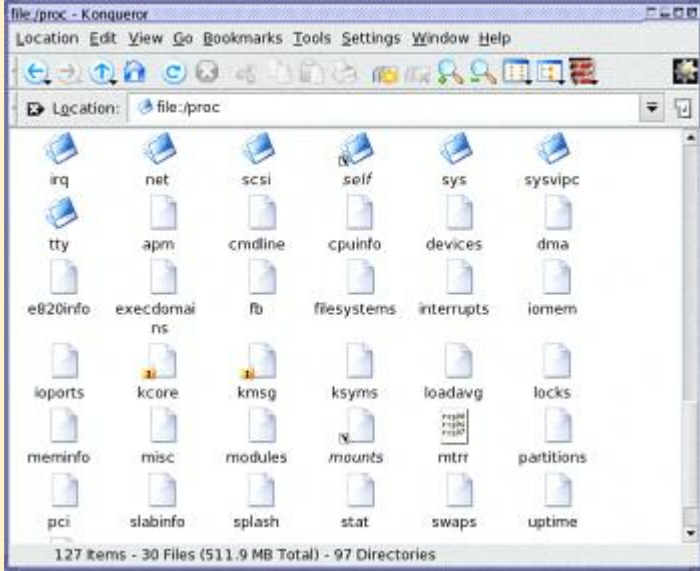
바이러스 구분	<ul style="list-style-type: none"> ○ 부트 바이러스 ○ 파일 바이러스 ○ 부트/파일 바이러스 ○ 메모리 상주 바이러스 ○ 매크로 바이러스 													
웜, 바이러스 트로이목마	<div>웜, 바이러스, 트로이목마 구분 조건</div> <table> <tr> <th>구분</th><th>독립실행</th><th>자기복제</th></tr> <tr> <td>웜</td><td>○</td><td>○</td></tr> <tr> <td>바이러스</td><td>X</td><td>○</td></tr> <tr> <td>트로이</td><td>○</td><td>X</td></tr> </table>	구분	독립실행	자기복제	웜	○	○	바이러스	X	○	트로이	○	X	
구분	독립실행	자기복제												
웜	○	○												
바이러스	X	○												
트로이	○	X												
Windows														
Windows 네트워크 구성 방식	<ul style="list-style-type: none"> ○ 워크 그룹 방식 <ul style="list-style-type: none"> - 각각의 계정의 자원을 시스템 별 관리 - 피어 투 피어라고 하며, 전용 서버 없이 개별 관리 - 보안은 각각 시스템의 SAM DB에 의해 제공 - Active Directory 없는 상황에서 운영하는 모델 ○ 도메인 방식 <ul style="list-style-type: none"> - 모든 계정과 자원을 특정 서버에서 관리하는 중앙 집중식 방식 - 사용자에게 설정된 권한은 다른 컴퓨터에 자원에서 적용 													
Window 로컬 계정	○ %SystemRoot%\System32\config\SAM 에 기록													
SAM DB	정리 필요													
Windows 로그	<ul style="list-style-type: none"> ○ 시스템 로그 <ul style="list-style-type: none"> - Windows 구성 요소에서 기록한 이벤트 수록 ○ 보안 로그 <ul style="list-style-type: none"> - 파일이나 다른 객체 만들기, 열기 또는 삭제 등의 리소스 사용과 관련된 이벤트 - 올바른 로그인 및 잘못된 로그인 시도 등의 이벤트 기록 ○ 시스템 로그 vs 보안 로그 <ul style="list-style-type: none"> - 시스템 로그는 로그 유형이 시스템의 의해 사전에 정의 - 보안 로그는 사용자가 정의하기 기록 남김 													
이벤트 뷰어	<ul style="list-style-type: none"> ○ 로그를 조회하고 관리할 수 있는 도구 ○ 응용프로그램 로그, 보안 로그, 시스템 로그 등 3가지 로그로 구성 	[용보시]												
공유 폴더 제거 명령	○ net share c\$ /delete													
레지스트리														
저장 위치	○ 레지스트리 구성(H\windows\User.Dat, System.Dat 파일에 저장)													
HKCR (Hkey_Class_Root)	○ 확장자와 연결 프로그램 설정, automation 정보, 바로가기 키 정보													
HKCU(Current_User)	○ 환경 설정, HKEY_USER보다 우선, Current_User 변경 시 HKEY_USer도 변경													
HKLM (Local_Machine)	<ul style="list-style-type: none"> ○ 하드웨어와 구동에 필요한 드라이버, 설정 관련 정보 ○ Hardware, Software, System 등 3개의 서브키로 구성 													
HK_USER	○ 데스크탑 정보, 네트워크 연결 정보, user.dat에 저장													
해킹														
사회 공학 해킹	○ 인간의 심리적 약점을 이용한 공격 / 정보 획득하는 방법													
Zero-Day Attack	○ 보안 취약점이 발견된 후 이를 막을 수 있는 패치가 발표되기 전까지의 공백 기간													
BOF														

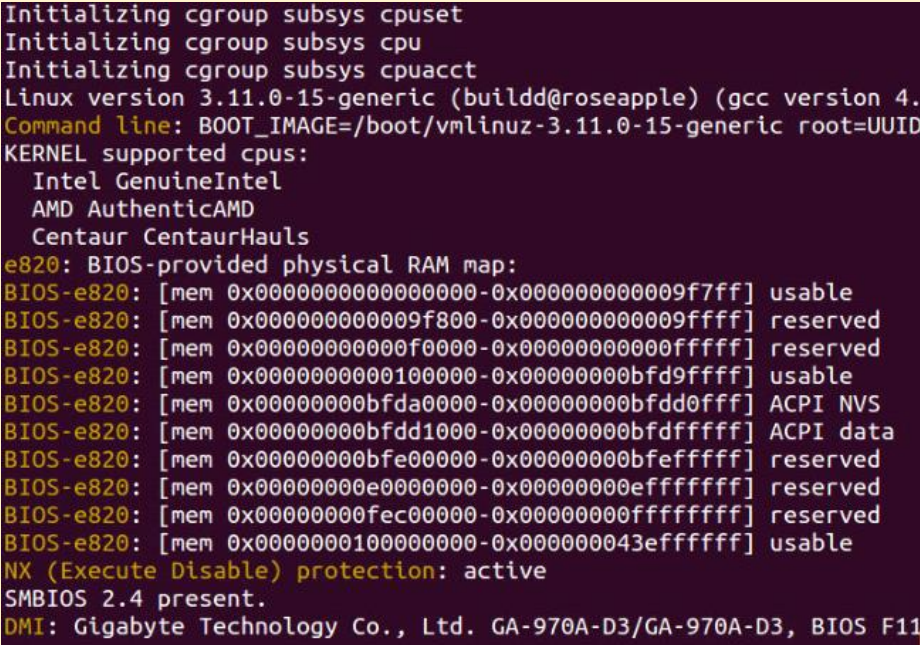
포맷 스트링 버그 (FSB)	<ul style="list-style-type: none"> ○ 개요 ○ 취약점 ○ 주요 공격 방법 ○ 예방 	
ASLR (Address Space Layout Randomization)	<ul style="list-style-type: none"> ○ echo 2 > /proc/sys/kernel/randomize_va_space ○ 버퍼 오버플로우 예방책 ○ 동적 메모리인 힙, 스택의 배치를 랜덤하게 설정하고 프로그램 실행시 마다 다른 주소 실행 ○ 변수나 복귀 주소 변경을 이용한 버퍼 오버플로우 대비 가능 ○ 0 : 무효 , 1 : 힙 이외 랜덤 , 2 : 모두 랜덤 	
스택		
버퍼오버플로 예방책	<ul style="list-style-type: none"> ○ 스택 가드 : 복귀주소와 변수 사이에 특정 값 저장 후 그 값이 변경되면 강제 실행 중단 ○ 스택 실드 : 리턴 주소를 Golbal RET라는 특수 스택에 저장 후 리턴 시 이 값과 스택의 리턴값 비교 ○ ASLR : 실행 시 메모리 주소를 변경 시켜 악성 코드에 의한 특정 주소 호출을 방지 ○ 버퍼 오버플로에 안전한 명령 사용 : strncat(), strncpy(), snprintf(), vsnprintf() 등 	
힙	<ul style="list-style-type: none"> ○ 프로그래머가 필요 시 할당하고, 해제하는 등 동적인 관리가 가능한 메모리 영역 ○ 힙은 Linked-list 구조를 가져 스택보다는 보안이 안전 	
	<p>정해진 메모리 범위를 넘치게 해서 원래 리턴 주소를 변경, 임의의 프로그램이나 함수를 실행하는 해킹 기법</p> <p>버퍼 오버플로우 공격</p> <p>버퍼 오버플로우 공격의 종류 스택 오버플로우, 힙 오버플로우</p>	
스케줄링		
Linux 스케줄링	<ul style="list-style-type: none"> ○ 우선 순위 기반의 선점형 스케줄링 기법 적용 ○ 우선 순위 값이 클수록 우선 순위가 낮음을 의미 ○ 각 CPU 마다 Runqueue라는 자료 구조 사용 , Runnable Task / Expired Task를 구분하여 스케줄링 ○ I/O를 많이 하는 프로세스를 우선 순위를 상대적으로 높게 설정 	
TCPwrapper		
inetd.conf	○ 설정에 관한 사항 확인(p213)	
! TCP-warpper	<ul style="list-style-type: none"> ○ 네트워크 트래픽을 제어, 모니터링, 로깅을 수행하는 UNIX 기반의 방화벽 툴 ○ TCP 연결로 부터 특정 네트워크의 보호 및 각각의 허용 / 거부 정책 적용 가능 ○ hosts.allow : 패킷의 허용 ○ hosts.deny : 패킷의 거부 	
/etc/hosts.allow	○	
/etc/hosts.deny		
! telnet 적용	<ul style="list-style-type: none"> ○ inet.conf 파일 ○ [적용전] telnet stream tcp nowait root /usr/bin/in.telnetd in telnetd ○ [적용후] telnet stream tcp nowait root /usr/bin/tcpd in.telnetd 	
인터럽트		
인터럽트	<ul style="list-style-type: none"> ○ 인터럽트 순위 : 전원 > 기계착오 > 외부 신호 > 입출력 > 명령 잘못 > 슈퍼바이저 콜 ○ 외부 인터럽트(비동기) : 장치, 전원 등 외부 장치, 전원/기계착오/외부 신호(타이머,키보드 등),입출력 <ul style="list-style-type: none"> - 비동기 : 프로세서는 인터럽트가 발생할 것을 전혀 예측하지 못함, 동기할 수 없음. ○ 내부 인터럽트 : 잘못된 명령어 사용, 프로그램 검사 인터럽트(0 나눔, 오버/언더 플로우) ○ 소프트웨어 인터럽트 : 프로그램 처리 중 요청에 의해 발생 SVC ○ IDT(Interrupt Descriptor Table) : X86 아키텍처, 인터럽스 발생 시 어느 루틴을 수행 정보 가짐 <ul style="list-style-type: none"> - 엔트리당 하나의 인터럽트에 대응, 프로세서가 여러개 일 경우 각자의 IDT 가짐 - IDT의 각 엔트리는 ISR의 주소로 가지고 있음. 	
PC	○ 다음 수행할 명령어의 번지를 기억하는 기억 장소, 레지스터, IP라고도 함.	
ISR (Interrupt Service Routine)	○ 인트럽트 발생 시 현재 프로그램 상태를 보존 후 인터럽스 루틴으로 제어를 옮기기 위한 프로그램	
퍼미션		
! 특수 퍼미션	○ 4000 > 2000 > 1000	

UGS, 유지스	<ul style="list-style-type: none">○ -rws rwx rwx --> 4///	
! SetUID	<ul style="list-style-type: none">○ 4000○ 프로그램 소유자 권한으로 프로그램을 실행할 수 있음.○ SetUID 설정되면 실행 권한 중 'x' --> 's'로 표시 <div><pre>-r-sr-sr-x 1 root sys 56808 Jun 17 12:02 /usr/bin/passwd</pre></div> <ul style="list-style-type: none">○ find / -perm 4000 -print	
! SetGID	<ul style="list-style-type: none">○ 2000 <div><pre>-r-x--s--x 1 root mail 71212 Jun 17 12:01 /usr/bin/mail</pre></div> <ul style="list-style-type: none">○ find / -perm 2000 -print	
! sticky-bit [보충/이해]	<ul style="list-style-type: none">○ 1000○ 디렉토리에 적용된 경우 : 파일 소유자, 디렉토리 소유자 또는 권한 있는 사용자만 파일 삭제 가능○ 파일에 적용된 경우 : <div><pre>drwxrwxrwt 7 root sys 400 Sep 3 13:37 tmp</pre></div> <ul style="list-style-type: none">○ find / -perm 1000 -print	
! umask	<ul style="list-style-type: none">○ 파일 생성 시 666에서 umask 값 빼고 적용○ 디렉토리 생성 시 777에서 umask 값 빼고 적용○ umask 적용은 /etc/profile에서 지정 가능하며, 022을 가장 많이 사용	
유닉스 명령어		
top	<ul style="list-style-type: none">○ 시스템의 전체적인 운영 상황 모니터링 가능○ CPU, MEMORY, DIKS 이용 상황, 전체 프로세스의 운영 상황 실시간 모니터링 가능○ 물리적인 스왑, 개별 프로세스 정보 등 시스템의 다양한 자원의 상황 확인 가능	
pstree	<ul style="list-style-type: none">○ 시스템의 모든 프로세스를 트리 구조로 제공하여 프로세스 간의 상호 관계 확인 가능	
nice	<ul style="list-style-type: none">○ 프로세스의 실행 순위를 조정하는 명령	
SetGID	<ul style="list-style-type: none">○ 2000○ find / -perm 2000 -print	
SetUID	<ul style="list-style-type: none">○ 4000○ find / -perm 4000 -print	
sticky Bit	<ul style="list-style-type: none">○ 1000	
find	<ul style="list-style-type: none">○ 특수 퍼미션(SetGID, SetUID) 검색<ul style="list-style-type: none">- find / -perm 2000 -print○ root 소유 파일에 SetUID 설정된 파일 검색<ul style="list-style-type: none">- find -user root -perm 4000 -print○ 수정일 10일 미만 파일<ul style="list-style-type: none">- find /etc/apache/conf -mtime -10 ; atime, ctime(속성 변경)- mtime 10 : 수정한지 10일 째- mtime +10 : 수정한지 10일이 지난	
tail	<ul style="list-style-type: none">○ secure 로그의 실시간 모니터링<ul style="list-style-type: none">- tail -f / var/log/secure	
passwd	<ul style="list-style-type: none">○ -S : 계정의 상태를 표시○ -o : 계정의 패스워드 삭제○ -l : 계정 잠금○ -u : 계정 잠금 해제	
chown/chgrp	<ul style="list-style-type: none">○ 소유자 및 소유그룹 변경	
chmod	<ul style="list-style-type: none">○ 권한 변경○ chmod 1777 file -> chmod ugo+rwxt file○ chmod u+s file	
lsof	<ul style="list-style-type: none">○ list open file○ 실행 중인 프로세스가 참조하고 있는 파일의 정보를 제공 <div><pre>COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME brcm_iscs 2854 root 3w REG 8,2 2376 207618127 /var/log/brcm-iscsi.log syslogd 3200 root 2w REG 8,2 0 207619507 /var/log/kern.log syslogd 3200 root 3w REG 8,2 256389 207619341 /var/log/messages syslogd 3200 root 4w REG 8,2 54364 207619371 /var/log/secure</pre></div>	

	<pre> COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME brcm_iscs 2854 root 3w REG 8,2 2376 207618127 /var/log/brcm-iscsi.log syslogd 3200 root 2w REG 8,2 0 207619507 /var/log/kern.log syslogd 3200 root 3w REG 8,2 256389 207619341 /var/log/messages syslogd 3200 root 4w REG 8,2 54364 207619371 /var/log/secure syslogd 3200 root 5w REG 8,2 0 207619455 /var/log/maillog syslogd 3200 root 6w REG 8,2 174268 207619502 /var/log/cron syslogd 3200 root 7w REG 8,2 0 207619500 /var/log/spooler syslogd 3200 root 8w REG 8,2 0 207619501 /var/log/boot.log acpid 3572 root 1w REG 8,2 4998 207619090 /var/log/acpid acpid 3572 root 2w REG 8,2 4998 207619090 /var/log/acpid mysqld 3811 mysql 1w REG 8,2 418390 207619133 /var/log/mysqld.log mysqld 3811 mysql 2w REG 8,2 418390 207619133 /var/log/mysqld.log httpd 3897 root 2w REG 8,2 578105 207619498 /var/log/httpd/error_log httpd 3897 root 7w REG 8,2 578105 207619498 /var/log/httpd/error_log httpd 3897 root 8w REG 8,2 9807 207619304 /var/log/httpd/access_log smbd 4093 root 2w REG 8,2 22001 207619344 /var/log/samba/samba.log </pre>	
cron	<p>○ 필드 순서: 분, 시, 일, 월, 요일(0=일요일), 작업</p> <p>○ 0 18-23/2 * * 1-5 command</p>	
! w	<p>○ 누가 시스템에 접속해 있는지 확인하는 명령</p> <pre> [root@host3 root]# w 11:46pm up 10:17, 5 users, load average: 0.04, 0.09, 0.08 USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT root pts/0 192.168.0.2 6:39pm 0.00s 0.38s 0.01s w root :0 - 2:41pm ? 0.00s ? - root pts/1 - 2:42pm 9:04m 0.00s ? - bible1 pts/2 192.168.0.202 11:46pm 46.00s 0.05s 0.05s -bash bible2 pts/3 192.168.0.111 11:46pm 1.00s 0.05s 0.05s -bash </pre>	
! who / w 비교	<pre> -bash-3.2\$ w 00:35:45 up 21 days, 11 min, 4 users, load average: 0.26, 0.26, 0.36 USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT ewright4 pts/1 155.70.23.45 19:53 1:57m 0.02s 0.02s -bash gettheco pts/2 ignfwdsyd-nat.au 00:35 0.00s 0.01s 0.00s w everpret pts/5 222.177.9.26 20:46 3:21 0.04s 0.04s -bash bookingp pts/0 cpe-24-25-214-14 18:31 2:49 0.05s 0.05s -bash -bash-3.2\$ who ewright4 pts/1 Jan 19 19:53 (155.70.23.45) getthecoder pts/2 Jan 20 00:35 (ignfwdsyd-nat.aust.csc.com) everpretty pts/5 Jan 19 20:46 (222.177.9.26) bookingpro411 pts/0 Jan 19 18:31 (cpe-24-25-214-144.san.res.rr.com) -bash-3.2\$ </pre>	
dmesg	<p>○ 부팅 시 나오는 각종 메시지를 확인할 수 있는 명령어</p> <pre> Initializing cgroup subsys cpuset Initializing cgroup subsys cpu Initializing cgroup subsys cpuacct Linux version 3.11.0-15-generic (build@roseapple) (gcc version 4. Command line: BOOT_IMAGE=/boot/vmlinuz-3.11.0-15-generic root=UUID KERNEL supported cpus: Intel GenuineIntel AMD AuthenticAMD Centaur CentaurHauls e820: BIOS-provided physical RAM map: BIOS-e820: [mem 0x0000000000000000-0x00000000000009f7ff] usable BIOS-e820: [mem 0x00000000000009f800-0x00000000000009ffff] reserved BIOS-e820: [mem 0x0000000000000f0000-0x0000000000000fffff] reserved BIOS-e820: [mem 0x000000000000100000-0x000000000000bdfd9fffff] usable BIOS-e820: [mem 0x000000000000bfda0000-0x000000000000bfdd0fff] ACPI NVS BIOS-e820: [mem 0x000000000000bfdd1000-0x000000000000bfdfffff] ACPI data BIOS-e820: [mem 0x000000000000bfe00000-0x000000000000bfefffff] reserved BIOS-e820: [mem 0x000000000000e0000000-0x000000000000efffffff] reserved BIOS-e820: [mem 0x000000000000fec00000-0x000000000000ffffff] reserved BIOS-e820: [mem 0x000000000000100000000-0x00000000000043efffffff] usable NX (Execute Disable) protection: active SMBIOS 2.4 present. DMI: Gigabyte Technology Co., Ltd. GA-970A-D3/GA-970A-D3, BIOS F11 </pre>	

init	○ /etc/inittab 파일의 읽어 시스템의 런 레벨을 결정	
vmstat	<div>○ 시스템의 프로세스 정보, 메모리 사용량, IO상태, CPU 사용률 등의 정보 제공</div> <div>○ 각 보고서 항목 정리 및 속지 필요(p214)</div> <div> <pre>procs-----memory-----swap-----io-----system-----cpu----- r b swpd free buff cache si so bi bo in cs us sy id wa st 0 0 0 15816 35400 372092 0 0 77 37 1324 585 16 2 78 4 0 0 0 0 15816 35408 372092 0 0 0 24 1138 411 1 1 96 2 0 0 0 0 15864 35408 372092 0 0 0 0 1141 429 2 1 97 0 0 0 0 0 15912 35408 372092 0 0 0 0 1158 444 2 1 98 0 0 0 0 0 15912 35416 372092 0 0 0 7 1063 276 1 0 99 0 0 0 0 0 15912 35416 372092 0 0 0 0 1062 291 1 0 99 0 0 0 0 0 15960 35424 372092 0 0 0 19 1062 322 1 0 96 3 0 0 0 0 15968 35424 372092 0 0 0 1 1064 290 1 0 99 0 0 0 0 0 15960 35432 372084 0 0 0 7 1063 285 1 0 99 0 0 0 0 0 15960 35432 372092 0 0 0 0 1064 298 1 0 99 0 0 0 0 0 15960 35440 372084 0 0 0 9 1065 284 1 0 97 2 0 0 0 0 16000 35440 372092 0 0 0 0 1070 302 1 0 99 0 0 0 0 0 15992 35440 372092 0 0 0 17 1156 420 2 1 97 0 0 0 0 0 15976 35448 372092 0 0 0 7 1176 460 2 1 97 0 0 0 0 0 15976 35448 372092 0 0 0 0 1128 364 2 0 98 0 0</pre> </div>	
find		
SetGID 파일 검색	○ find / -perm 2000 -print	
SetUID 파일 검색	○ find / -perm 4000 -print	
특수 퍼미션(SetGID, SetUID) 검색	○ find / -perm 2000 -print	
root 소유 파일, SetUID 설정 파일 검색	○ find -user root -perm 4000 -print	
보안 점검 도구		
 Tripwire	무결성 점검 도구	
 Nessus	네트워크 취약점, 패스워드 취약점, TCP/IP 스택, DOS, 취약한 서버 설정 등 알려진 취약점 점검, text/HTML 형태 보고서	
파일 무결성	tripwire, MD5, Fcheck, AIDE 등	
SATAN	해커와 같은 방식으로 침입, 보안 취약점 확인, 그래픽 모드, 해커 악용 가능	
SARA	SAINT 업데이트 종료에 따른 개선 버전	
SAINT	네트워크 취약점 분석, HTML 형태 보고서 제공	
COPS	시스템 내부 취약점 점검, 취약 패스워드 확인	
nmap	포트 스캐닝	
파일 시스템		
proc	<div>○ 커널 동작에 필요한 매개 변수를 제어, 디바이스 드라이버 동작 감시 및 제어 흐름 변경</div> <div>○ 커널에서 제공하는 프로세스 정보를 저장하는 파일 시스템</div> <div>○ 실제 프로세스 목록과 CPU 점유율, 메모리 사용률을 제공</div> <div>○ 일반 파일 시스템은 오버헤드가 발생하지만 proc는 커널이 직접 제공해서 오버헤드 경감 가능</div>	

																						
ext2 / ext3	○ chattr 명령으로 읽기 전용(read only) 속성으로 변경 시 root도 통제 가능																					
아이노드	○ 파일의 이름을 제외한 해당 파일의 모든 정보를 가지고 있는 자료 구조 ○ 각 파일에 부여되는 고유한 번호이며, 파일형태, 위치, 크기, 소유자 등의 정보로 구성																					
수퍼블록	○ 파일 시스템에 의존하는 정보를 가지며, 파일 시스템의 크기 등과 같은 파일 시스템 전체 정보 관리 ○ 슈퍼 블록은 파일 시스템을 보다 더 빠르고 효과적인 파일 시스템 관리를 가능하게 함. ○ 파일 시스템 유형, 디스크 블록 크기, 파일 시스템 디바이스 파일 이름, iNode 갯수 ○ 할당 되지 않은 디스크 블록 및 inode의 갯수와 위치																					
유닉스 자동 로그 아웃	○ TMOUT = 1800; export TMOUT ○ /etc/profile																					
유닉스 핵심 컴포넌트 [KSF]	○ 커널 ○ 셸 ○ 파일 시스템																					
유닉스 디렉토리 구조	<table><tr><td>/</td><td>○ root 디렉토리</td></tr><tr><td>/etc</td><td>○ 시스템 설정 파일</td></tr><tr><td>/dev</td><td>○ 특수 파일 저장 디렉토리</td></tr><tr><td>/usr/bin</td><td>○ 디폴트 사용자 명령어 저장</td></tr><tr><td>/usr/include</td><td>○ C 언어 라이브러리 헤더 파일</td></tr><tr><td>/usr/lib</td><td>○ C 언어 라이브러리</td></tr><tr><td>/usr/sbin</td><td>○ 시스템 관리 명령어</td></tr><tr><td>/home</td><td>○ 사용자 홈 디렉토리</td></tr><tr><td>/tmp</td><td>○ 임시 파일 저장</td></tr><tr><td>/var</td><td>○ 시스템 로그 저장</td></tr></table>	/	○ root 디렉토리	/etc	○ 시스템 설정 파일	/dev	○ 특수 파일 저장 디렉토리	/usr/bin	○ 디폴트 사용자 명령어 저장	/usr/include	○ C 언어 라이브러리 헤더 파일	/usr/lib	○ C 언어 라이브러리	/usr/sbin	○ 시스템 관리 명령어	/home	○ 사용자 홈 디렉토리	/tmp	○ 임시 파일 저장	/var	○ 시스템 로그 저장	
/	○ root 디렉토리																					
/etc	○ 시스템 설정 파일																					
/dev	○ 특수 파일 저장 디렉토리																					
/usr/bin	○ 디폴트 사용자 명령어 저장																					
/usr/include	○ C 언어 라이브러리 헤더 파일																					
/usr/lib	○ C 언어 라이브러리																					
/usr/sbin	○ 시스템 관리 명령어																					
/home	○ 사용자 홈 디렉토리																					
/tmp	○ 임시 파일 저장																					
/var	○ 시스템 로그 저장																					
유닉스 파일 종류	○ 일반 파일 ○ 디렉토리 파일 ○ 특수 파일 : 프린터, 터미널, 디스크와 같은 주변장치, 파이프/소켓 같은 프로세스 상호 통신 자료																					
심볼릭 링크 생성 명령	○ ln -s /path /path ○ -s 옵션 제거 시 하드 링크 생성																					
유닉스 링크	○ 심볼릭 링크 ○ 하드 링크																					
환경 설정																						

/etc/syslog.conf	○ 시스템 로그 데몬이 실행될 때 참조하는 로그 환경 설정 파일	
/etc/services	○ 슈퍼 데몬이 사용하는 포트 번호 정의 파일 ○ 서비스 이름, 포트 번호, 프로토콜 이름, 별명(aliases)	
/etc/hosts	○ DNS Query 전에 참조하는 파일, 보안 위험도 증가하므로 관리 신중 필요	
/etc/hosts.equiv	○ 전체 시스템의 원격 접근에 대한 접근 제어 목록을 가진 파일(???), 이해 필요	
/etc/inittab		
/etc/inet.conf	○ 슈퍼 데몬이 사용하는 환경 설정 파일	
로그 파일 및 명령		
! MAC Time	○ Modify Time : 로그 파일 마지막 수정 시간 ○ Access Time : 로그 파일 마지막 접근한 시간 ○ Change Time : 로그 파일의 속성(퍼미션, 소유자 등)의 마지막 수정 시간	
! 사용 기록 핵심 로그	○ /var/log/message : 리눅스 시스템의 가장 기본 로그파일, 시스템 운영의 전반적 메시지 기록 ○ /var/log/secure : 사용자의 원격 접속 기록 관리 ex)sshd, su, telnet 등	
syslogd	○ 시스템 로그를 관장하는 데몬 ○ /var/log/message이 대표적으로 관장하는 로그 파일 ○ /etc/syslog.conf 파일로 설정 가능	
logrotate	○ 로그 파일이 커지면 분할해서 관리 하기 위한 명령, 적당한 크기로 분할, 삭제, 압축, 메일 전송 지원 ○ cron 명령과 조합하여 주기적으로 로그 파일 핸들링에 활용	
/var/log/dmesg dmesg 명령	○ 부팅 시 출력된 메시지 로그 ○ 부팅 시 나오는 각종 메시지를 확인할 수 있는 명령어 	
/var/adm/loginlog		
/var/log/xferlog	○ ftp log	
! /var/adm/sulog	○ su 명령을 사용한 기록	
! /var/adm/loginlog	○ 실패한 로그인 시도 기록	
/var/adm/lastlog ! lastlog 명령	○ 각 사용자의 최근 사용 기록 확인 ○ 최근 로그인 시각과 접근한 소스 호스트 정보 ○ lastlog -u user01	
/var/adm/btmp	○ 5회 이상 로그인 실패 기록, lastb 명령	
! /var/adm/utmp	○ 현재 시스템에 접속해 있는 사용자의 정보, 바이러리 형태로 저장	

	<ul style="list-style-type: none"> ○ 현재 로그인한 사용자 정보 DB 파일 ○ utmp 로그 참조 명령 : who, w, whodo, users, finger 	
/var/adm/wtmp wtmp 명령	<ul style="list-style-type: none"> ○ 처음부터 접속했던 모든 사용자의 로그인/로그아웃 기록 누적함. ○ 사용자 로그인/아웃, 시스템 셧다운, 부팅 정보, <u>last 명령어로 확인 가능</u> 	
acct	○	
/acct/pacct	○ 로그인 동안 입력한 명령 및 시간, tty 등에 대한 정보	
/var/log/cron	○ 크론 로그	
/var/log/maillog	○ 메일 송수신 로그	
<u>/var/log/secure</u>	<u>○ 원격 접속에 관한 로그, tcp_wrapper 접속 제어에 관한 기록</u> <ul style="list-style-type: none"> ○ 서버 보안에 아주 민감하고, 중요한 파일, sshd, su 관련 실행, telnet/원격 접속 기록 제공 ○ 보안과 관련한 주요 로그 및 사용자 인증에 관련된 로그 포함 ○ 사용자 인증은 telnet, ftp, pop 등 인증에 관련된 모든 네트워크 서비스 대상 	
<u>/var/log/message</u>	<u>○ 시스템 운영에 대한 전반적인 로그</u> <u>○ 가장 기본적인 로그 파일, 주로 시스템 데몬들의 실행 상황과 내역, 사용자 접속 기록 제공</u>	
로그 기록	5번 이상 로그인 실패한 기록(AIX) failedlogin 5번 이상 로그인 실패한 기록(솔라리스)	
history	<ul style="list-style-type: none"> ○ 사용자 HOME 디렉토리에 생성되는 실행 명령어 기록 ○ acct, pacct 보다 상세하게 입력한 인수 및 디렉토리 정보까지 기록 ○ \$HOME/.history, \$HOME/.bash_history 파일로 생성 	