협성대학교 소프트웨어공학과

정보보호

〈정보 보안 개론〉 03, 04, 05, 06 장 내용 정리

이시헌

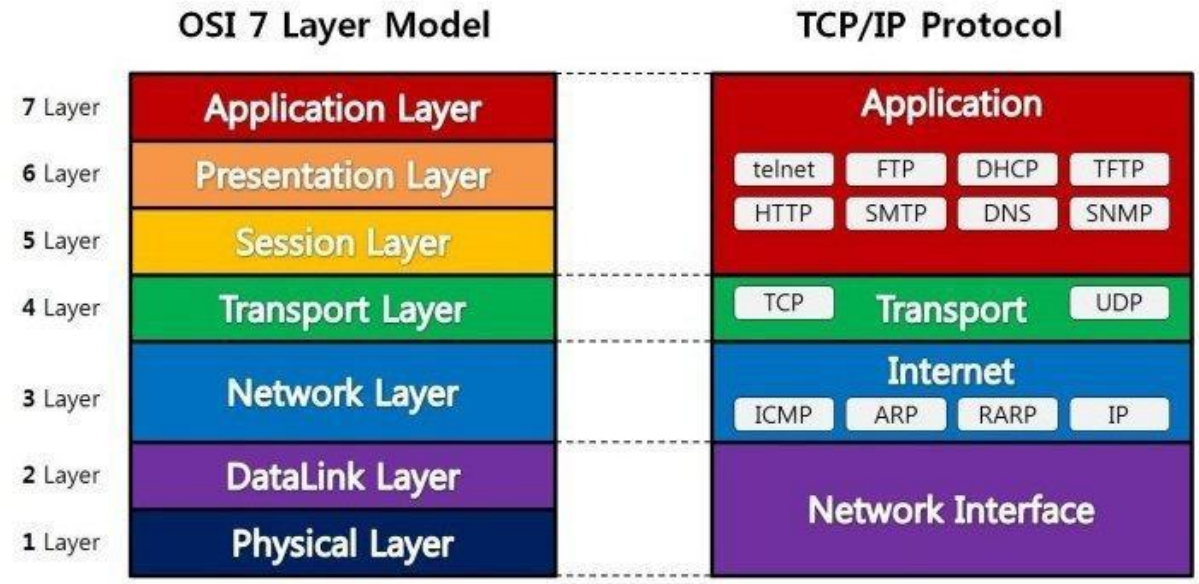
2025-4-28

시험범위는 03 ~ 06장 (4주차부터 03장 진행.)

4주차 강의 내용 정리 시작.

1. OSI 7 Layer (p.113, [그림 3-1]참고)

   = 표준 네트워크 모델



(출처: https://shlee0882.tistory.com/110)

A. 계층 설명

i. 7 - 응용 소프트웨어 계층

- 다양한 서비스를 실제로 제공하는 계층이다.
  (HTTP, FTP, SMTP ….)

ii. 6 - 표현 계층

- 데이터의 구조를 번역하는 계층이다. (코드 간 번역 수행.)
- 응용프로세스들의 독립성을 보장한다.

iii. 5 - 세션 계층

- 독립된 두 컴퓨터에 있는 양 끝단의 응용 소프트웨어간 논리적 연결을 구성하고 해제하는 계층이다.
- TCP/IP의 세션 또한 5계층에서 생성, 소멸한다.

iv. 4 - 전송 계층

- 7 ~ 5계층은 논리적인 계층이고,
  4계층부터 물리적인 계층이다.
- TCP 또는 UDP 프로토콜로 통신을 수행한다.
- 응용 소프트웨어를 위한 논리적인 **[출발지 Port, 목적지 Port] 정보를 패킷에 추가**하는 계층이다.

v. 3 - 네트워크 계층

- **IP를 기반으로 데이터를 전송하고 수신**하는 계층이다.
- 데이터 전송의 신뢰성을 보장하지 않으며,
  연결 여부도 따지지 않는다.
  신뢰성은 상위 계층(TCP/IP 〈- 4계층)에 의존한다.
- 목적지 IP에 도달할 때 까지, MAC 주소를 기반으로 라우팅이 계속 수행된다. (L3 스위치(라우터)를 거쳐가며 계속 이동.)
- 개별 라우터에 도착할 때 마다, 이전 계층에서 사용한 **[출발지 MAC, 목적지 MAC]을 계속 갱신**한다. (라우팅, 흐름 제어, 세그멘테이션…, 등을 수행하는 계층.)

vi. 2 - 데이터 링크 계층

- MAC 주소를 기반으로 데이터를 송수신하는 계층이다.
- 1계층의 정보 송수신 흐름과 오류를 관리하여,
  물리 계층에서의 오류를 찾아내고 해결하는 기능을 제공한다.
- 즉, 물리 계층의 신뢰성을 보장하는 계층이며,
  MAC 주소를 기반으로 스위칭을 수행하는 계층인 것이다.

vii. 1 - 물리 계층

- 전기 신호를 기계적으로 전달하는 계층.
- 데이터의 송신과 수신만 수행하며, 다른 어떠한 기능도 존재하지 않는다.

2. DoS (Denial of Service)

= 서버의 자원을 고갈시켜 서비스의 정상 작동을 방해하는 공격 기법.

A. 주요 방법

i. 취약점 공격: TCP 오류 제어의 취약점을 주로 공략. (오류 제어 작업을 반복시키는 것.)

ii. 자원 고갈 공격: 서버에 높은 부하를 주는 다양한 방법을 사용.

B. 취약점 공격의 대표 예시:

i. TearDrop: TCP 시퀀스 번호를 속여서, 누락된 구간의 정보를 재요청하도록 하는 공격이다.

ii. Land: 출발 IP를 도착 IP로 하여, 서버가 자신에게 요청을 보내도록 해 과부하를 유도한다.

C. 자원 고갈 공격의 대표 예시

i. Ping of Death

= 초기의 DoS 기법. (윈도우 95, 98. 리눅스 6.0 이하)

- ping 요청을 최대 길이(65,500byte)로 전송하는 것.
  이때 ping 요청은 네트워크에서 작은 패킷으로 나뉘어지며, 결과적으로 공격 대상 시스템은 대량의 작은 패킷을 처리하느라 시스템 자원 고갈을 맞는다.
- IMCP(Internet Control Messaging Protocol) 프로토콜 차단으로 막을 수있다. (ping 요청에 사용하는 프로토콜이 IMCP이다.)

ii. **SYN 플러딩**

= 3-Way Handshake 의 허점을 공략하는 공격법.

- 3-Way Handshake 과정:

  A. 클라이언트가 SYN 패킷을 전송.

  B. 서버는 SYN+ACK 패킷을 클라이언트에게 전송.

  C. 클라이언트가 서버에 ACK 패킷을 전송.

- 이때 C 단계를 의도적으로 수행하지 않는 Handshake 요청을 대량 보내는 것으로, 서버에 존재하는 가용 최대 접속자 숫자가 소진되도록 할 수 있다.

iii. **HTTP GET 플러딩**

= 정상적으로 3-Way Handshake 과정을 마친 후, 특정 페이지에 있는 GET 메서드를 이용하여 특정 페이지를 무한대로 실행하는 공격.
(GET 방식으로 특정 변수를 전송하는 것을 반복. (하나의 페이지 접속을 유지하며 동일 변수의 요청 반복.))

iv. **HTTP CC**

= HTTP 1.1 버전의 CC(Cache-Control) 헤더 옵션을 사용하여, 웹 서버가 갖고 있는 캐시를 비활성화 하는 공격.
(이 경우, 서버는 캐시를 사용하지 않고 웹 페이지를 새로 구성하여 전송하므로, 많은 리소스를 사용한다.)

v. **동적 HTTP 리퀘스트 플러딩**

= 앞서 설명한 HTTP GET 플러딩, HTTP CC 의 쿼리 URL을 동적으로 갱신하여, 공격 대상의 차단 시스템을 우회하는 공격 기법.

vi. **스머프 공격**

= 다이렉트 브로드캐스트를 악용하는 것.

- 공격 대상의 네트워크에서 사용되는 브로드캐스트 주소를 사용하여, 원격지의 네트워크에서 브로드캐스트를 수행한다.
- 기본적으로 ICMP Echo Request 에 대하여, ICMP Echo Reply 응답이 이루어지는 것.
- 이때, **원격지의 라우터에 연결된 모든 컴퓨터에 브로드캐스트로 위조된 IP(IP Spoofing)로 ping을 보내면, 위조된 IP로 Request 가 송신된다.**
- 이를 통하여 수많은 request ping이 위조된 IP로 송신되고, 시스템이 과부하 된다.

vii. **메일 폭탄**

= 대량의 메일을 보내어, 메일 서버에 할당된 디스크 공간을 가득 메우는 것.

viii. DDoS

= 복수의 컴퓨터를 감염시켜, DoS 공격을 수행하도록 하는 것.

- 좀비 PC 양산 -> DoS 공격 수행.
- 최근의 DDoS 공격은 악성코드와 결합된 형태가 많다. (스스로 감염 전파.)

3. 스니핑 공격

= 방어수단을 무력화하고 시스템의 정보를 열람하는 공격.

A. 특징.

i. 수동적이다. 대표 예시: 도청.

B. 원리

i. 일반적으로 IP와 MAC이 자신을 목적지로 하지 않는 모든 패킷은 무시되는 것이 정상이나, 프러미스큐어스(Promiscuous)모드를 활성화 해둔 경우, 그러한 필터링이 해제된다.

ii. 이때 프러미스큐어스 모드를 활성화시켜 스니핑을 수행한다.

C. 종류

i. 스위치 재밍 공격: MAC 주소 테이블을 포화시켜, 스위칭 기능을 마비시킨다.

ii. SPAN 포트 태핑 공격: 포트 미러링 기능을 악용하여 데이터 감청.

D. 탐지 방법:

i. 거의 없다.

ii. 프러미스큐어스 모드의 활성 여부를 검토하는 것을 권장.

iii. P.146을 참고.

4. 스푸핑(Spoofing) 공격

= 어떤 정보(IP, MAC …)를 속이는 것을 통해 수행하는 공격.

A. ARP 스푸핑

i. ARP: Address Resolution Protocol

2계층 프로토콜.

MAC 과 IP를 연결하는데 사용된다. (IP 기반으로 MAC을 알아낸다.)

ii. 원리:

- 사실상, MAC 기반의 패킷 하이재킹을 수행하는 것이다.
- 공격자는, 정상적인 클라이언트-서버 통신의 당사자들에게, 가짜 MAC 주소와 IP를 알린다.
- 이후, 통신은 공격자를 경유하여 진행되며, 공격자는 자신에게 온 패킷을 열람하고, 본래 목적지로 향하도록 IP와 MAC을 조작하여 다시 송신한다.

B. IP 스푸핑

= 특정 IP를 지닌 사용자인 것처럼, 사칭하는 것이다.

i. 원리:

- '특정 IP'를 지닌 사용자가, 서버와 트러스트(trust)관계를 맺고 있을 때, 해당 IP로 서비스 거부 공격을 수행하여 접속을 차단해 두고, 순조롭게 사칭을 수행한다.

ii. 방어 방법:

- 사실상 없다.
- 트러스트 IP를 설정하지 않는 것으로, 원천 차단해야 할 뿐.

C. ICMP 리다이렉트 공격

i. 원리:

- 특정 호스트가 다른 컴퓨터와 통신할 때,
  호스트와 연결된 다수의 라우터가 있다고 하자.
- 호스트의 주 라우터A가, 호스트와 연결된 라우터B로 패킷을 전달할 때, 라우터A는 호스트에게 다음 통신은 라우터B와 하라는 신호로 'ICMP 리다이렉트 패킷'을 보낸다.
- 이때 공격자가 라우터B의 역할을 대신하여, 호스트의 모든 패킷을 스니핑 할 수 있게 된다.
- 라우터B의 자리를 획득하기 위해 ICMP 리다이렉트를 공격자가 직접 송신한다. 즉, 호스트는 공격자를 라우터로 인지하고 데이터 송신을 수행한다.

D. DNS 스푸핑 공격

= 공격 대상이 DNS에 질의를 할 때, DNS의 실제 대답보다 빨리 공격자가 준비된 응답을 보내는 것.

i. 원리:

- DNS 패킷은 UDP이므로, 먼저 도착한 응답 패킷만 얻고, 이후 패킷은 무시한다.
- 즉, 가짜 응답을 먼저 전송하기만 하면, 공격 대상이 공격자가 원하는 IP로 접속하게끔 할 수 있다.

ii. 방어 방법:

- 마땅히 없다.
- 그나마 유효한 방법은, DNS 질의 자체를 보내지 않는 것. (hosts에 알려진 IP를 최대한 준비.)

5. 세션 하이재킹

= 문자 그대로, 세션을 가로채는 공격법이다.

A. 방법:

i. 네트워크 공격으로 세션 탈취.

- 여하튼… 어떻게든 빼앗으면 된다.

ii. TCP 취약점으로 세션 탈취.

- ARP 스푸핑 등의 공격을 통해, 모든 패킷을 감청 시작.
- RSTreset 패킷을 보내어, 서버가 재설정 된 시퀀스 넘버에 따라 3-Way Handshake를 다시 수행하도록 함.
- 공격자는 클라이언트 대신 세션(TCP 연결) 획득.

6. 무선 네트워크 공격

A. AP 보안의 기초

i. 물리적 전파 범위 조절 (최소한으로)

ii. ID, PW 설정 하기

iii. 암호화 통신 하기 (WEP, WPA-PSK, WPA-EAP…)

- WPA-EAP <- 802.1x 표준에 따름. (ID 와 PW 기반의 사용자 인증 수행)

iv. SSID 브로드캐스팅 피하기 (AP 존재를 숨기자)

4주차 강의 내용 정리 끝. (3장 종료.)

5주차 강의 내용 정리 시작. (4장)

1. Gateway

   = 프로토콜을 해석하여 다른 프로토콜로 전환하여 다른 시스템으로 전송하는 객체.

   A. 게이트웨이의 개발 이후, 네트워크라 부를 수 있는 것이 생겼다.

2. HTTP (Hyper Text Transfer Protocol)

   = 웹에서 가장 널리 쓰이는 데이터 송수신 프로토콜.

   A. 계속하여 버전업이 되었다.

      i. 0.9: 단순 읽기 지원 + 개별 송신마다 Connect 과정 거친다.

      ii. 1.0: 1회의 Connect 이후, Request, Response 반복.

      iii. 현대에는 1.1 이상의 버전이 사용된다.

3. HTTP Request 의 종류

   A. GET: URL에 요청 데이터 인수를 붙여서 전달. (2048자 이하)
      (참고: 브라우저에서 자동으로 캐싱을 수행한다.)

   B. POST: HTTP 헤더에 데이터 적재. (2GB 정도)
      (참고: 브라우저에서 자동으로 캐싱하지 않는다.)

   C. 기타 방식:

      i. HEAD: 서버 측의 데이터를 검색하고 요청.

      ii. OPTIONS: 정보 요청 이전에, 서버에 상태 질의.

      iii. PUT / DELETE / TRACE <- 이런 것이 있다고만 알아 두라.

4. HTTP Response

A. 기본 정보: 프로토콜 버전, Request 처리 결과 코드(상태코드)…

B. 추가 정보: (IME 형식으로 적재됨) 전달할 데이터의 형식, 길이…

C. 위의 두 정보가 '헤더 정보'이다.

D. 헤더 이후에 실제 데이터가 적재된다. (HTML, 이미지…)

E. 실제 데이터 이후에, 서버의 연결 종료가 수행된다.

5. 웹페이지의 종류와 안전도

A. 위험도가 낮은 것부터 -> 높은 것 까지

i. 정적 웹페이지

ii. 서버사이드의 동적 웹페이지

iii. 클라이언트사이드의 동적 웹페이지

6. REST API

= HTTP 기반의, 상태 저장 없는 웹 구현 방식.

A. 그런데 교재의 정의가 마음에 들지 않는다.

B. REST: HTTP URI로 자원 명시 + HTTP Method(POST, GET…)로 자원에 대한 CRUD 수행.

C. 즉, REST API는, HTTP URI로 자원에 대한 접근을 처리하기 위한 API.

7. 웹 해킹

A. 웹 해킹을 위해 해볼 수 있는 사전 작업은 다음과 같다.

i. 웹 취약점 스캐너로 정보 수집.

-> 수집된 취약점의 유효성 검사.

ii. 웹 프록시를 사용하여 웹 구조 파악, 취약점 점검. (클라이언트 사이드에서 패킷 확인 및 변조 시험.)

iii. 구글 고급 검색 기능 활용

-> 특정한 문서 존재 검색 (admin, password…)

-> 디렉터리 리스팅 여부 검색 (Intitle 키워드 사용)

8. 웹의 주요 취약점 (OWASP TOP 10)

A. Injection (**명령 삽입 취약점**)

i. 서버로 명령을 전달하는 모든 경우에 발생 가능.

B. Broken Authentication and Session Management

i. 취약한 PW로 발생.

ii. 사용자 데이터를 이용한 인증으로 발생 가능.
(필요 이상의 데이터-세션 등-를 클라이언트에게 전송하는 경우 주로 발생한다.)

C. Cross-Site Scripting (XSS)

i. 웹 브라우저로 악성 스크립트를 전송하는 것.

D. Broken Access Control

i. 인증된 사용자에 대한 접근제어 코드 미비로 발생.

ii. SW 개발시, 표준 인증 로직을 우선 구현하여 해결할 수 있다.

iii. Directory Traversal 공격이 대표적인 사례.

E. Security Misconfiguration

i. Directory Listing 설정 상태 인지 실패.

ii. Backup Files, Temporary files … 등을 방치.

iii. 미흡한 주석 관리.

iv. File upload limit 설정 미비.

v. Reverse Telnet -> Shell Access 가능성.

(일반적으로 인바운드 방화벽은 철저하지만, 아웃바운드 방화벽은 허술하다.

이를 공략하여, 서버에서 telnet 을 주도적으로 실행하도록 하면 공격자가 Shell 을 얻을 수 있다.

이를 위해, 공격자는 파일 업로드 제한이 미비한 서버를 노린다. (실행 파일 업로드로 telnet 실행 유도.))

F. Sensitive Data Exposure

i. 암호와 로직이 미비하거나, 암호화 구조에 문제가 있는 경우이다.

ii. 그리하여 민감한 정보가 노출된다.

G. Insufficient Attack Protection (공격 방어 취약점)

i. 공격의 자동 탐지 기능이 미비한 경우이다.

ii. 취약점이 쉽게 드러나는 상태라고 할 수 있다.

H. Cross-Site Request Forgery (CSRF, 교차 사이트 요청 위조)

i. 악성 스크립트를 서버로 보내어, 서버가 악성 스크립트를 실행하도록 한다.

ii. 일반적으로 신뢰할 수 있는 사용자를 사칭하여 서버에 명령을 보내므로, 각 사용자를 구별하는 인수의 철저한 검토가 필요하다.

I. Under protected APIs

i. 보안에 취약한 API를 가져다 쓰거나, 보안에 취약점이 있는 API를 만들었을 때 문제가 생긴다.

J. 위의 10가지 취약점을 방어하는 방법

i. 특수문자 필터링 수행. (서버측에서, 클라이언트의 모든 입력에 대한 검증을 수행한다.)

ii. 지속적인 세션 관리. (모든 페이지에서 세션에 대한 인증을 반복적으로 수행해야 한다.)

5주차 강의 정리 끝. (4장 정리 끝.)

6주차 강의 정리 시작. (5장)

1. **보안에 취약한 소스코드**

   = 데이터의 형태와 길이에 대한 불명확한 정의에 기반하여 작성된 코드.

2. **메모리 구조**

   A. 상위 메모리 - 스택 (OxFFFF)

   B. 하위 메모리 - 힙 (0x0000)

   C. 레지스터 (메모리와 상호 데이터 교환, CPU의 임시 메모리.

3. **레지스터와 어셈블리어**

   A. 대표적인 몇 가지만 기술하겠다.

      i. EAX - 누산기(accumulator) - 산술연산 함.

      ii. ECX - 카운트 레지스터(count Register) - (반복 횟수)입출력 수행.

      iii. EBP - 베이스 포인터(base pointer) - 변수 값 읽기

      iv. ESP - 스택 포인터(stack pointer) - 스택의 끝 주소 읽기. (top())

4. **셸 (Shell)**

   = 명령어 해석기

   A. 버퍼 오버플로 공격, 포맷 스트링 공격의 최종 목표 = 관리자 권한의 셸.

   B. 일반적으로 셸 자체를 기계어 코드로 바꾸어 메모리에 적재하여 실행시킨다. (기계어로 바꾼 셸 경로(/bin/sh)를 메모리에서 실행.)

5. SetUID

   = 파일 실행시 소유자 권한을 대여하도록 허용하는 모드.

   A. rws <-이때 Access 권한 자리에 들어간 s 가 SetUID.

      i. 이 파일을 실행하면, 소유자 권한을 대여하여 파일이 실행된다.

   B. 이 설정을 가급적이면 하지 말아야 한다.

6. **버퍼 오버플로우 공격**

A. 데이터의 길이를 명확히 규정하지 못했을 때 발생하는 취약점.

B. 준비된 메모리 공간을 초과하는 영역에 접근을 시도하여 생기는 취약점.

C. 해결책

i. strcpy(), strcat() 등의 문자열 관련 함수를 사용하지 않는 것이 좋다.

ii. 최신 운영체제를 사용하자.

7. **포맷스트링 공격**

= 데이터의 형태를 규정하지 않았을 때 발생하는 취약점.

A. 원리:

i. 출력형식지정자가 변수에 있는 값을 적절한 포맷으로 바꾸기 위해서는, 해당 값을 메모리에 적재할 필요가 있다.

ii. 이때, 메모리에 적재되는 정보가 셸의 기계어 번역 문자열이라면 어떻게 되겠는가? 버퍼 오버플로와는 전혀 다른 경로로, 원하는 코드를 메모리에 넣을 수 있는 것이다.

8. **메모리 해킹**

= SW가 실행되는데 필요한 정보를 저장해 둔 메모리를 조작하는 공격.

A. 실행 중인 SW가 점유하는 메모리를 알아내어, 조작할 수 있다.

B. 대응: 메모리에 저장되는 값을 암호화 하여 저장해야 한다.

6주차 강의 정리 끝. (5장 끝.)

7주차 강의 정리 시작. (6장)

1. **바이러스**

   = 스스로 복제하는 악성코드

2. **악성 코드의 동작에 따른 분류**

   A. 복제와 감염 - 바이러스 (다른 SW에 기생, 실행파일에 포함)

      => 파일 파괴 및 손상 추구.

   B. 네트워크를 통해 스스로 전파 - 웜 (독립적인 SW, 스스로 복제 및 전파)

      => 네트워크 트래픽 과부하 유도.

   C. 전파X, 복제X, 침투 후 컴퓨터 조작 - 트로이목마

      => 정보 탈취 및 도용 추구.

   D. 불필요한 SW - PUP (스파이웨어)

      => 사용자를 귀찮게 한다. (불편함 제공 추구.)

3. **목적에 따른 악성 코드의 분류**

   A. 다운로더: 다른 악성코드의 다운로드를 위한 SW

   B. 드로퍼: 트로이목마의 일종. 자신 내부에 숨겨진 악성코드를 설치.

   C. 애드웨어: 사용자 동의 없이 광고를 출력하는 SW.

   D. 스파이웨어: 사용자 동의 없이 정보를 수집하여 전송.

   E. 랜섬웨어: 파일을 암호화하고 인질로 삼아 금전 요구.

   F. 백도어: 정상 보안 절차를 우회하여 시스템에 접근.

   G. 익스플로잇: 시스템 취약점을 이용하여 악성코드 설치 등 수행.

   H. 봇: DDoS 공격을 수행하는데 사용되는 자동화 SW.

   I. 스캐어웨어: 가짜 경고 등으로 자사의 SW 구매를 유도하는 SW.

4. 감염 증상

A. 교재 p.273 참고.

B. 간략히 정리하면, 주요 증상 위치는 다음과 같음.

C. 시스템 / 네트워크 / 하드디스크 / 파일 / 특이점(분류 안되는 것들)

5. 바이러스의 세대 구분

A. 1세대: 자기 복제 + 데이터 파괴 수행

B. 2세대: 1세대 + 암호화된 코드 (메모리 적재시에 복호화 수행)

C. 3세대: 2세대 + 잠복기 (SW 확산 이후에 활동 개시)

D. 4세대: 2세대 + 바이러스 감지를 위한 식별자를 메모리에서 능동적으로 수정하는 조합(mutation) 코드를 포함한다.
(제작도 진단도 매우 어렵다.)

E. 5세대: 스크립트 실행 환경을 타겟으로 하는 바이러스.
(예: MS-Office 내부의 VBS 스크립트…)

F. 차세대: 스크립트 형태의 바이러스가 계속 발전하여, 네트워크 전파와 정보 탈취 등의 기능이 점차 추가된다. (웜의 형태로 바이러스가 계속 발전한다.)

6. 웜

= 스스로 증식하고 전파되는 악성 SW

A. 유형

i. 메스메일러: 대량 메일 발송 및 확산.

ii. 네트워크 공격: DoS 공격을 수행하는 SW 전파 및 감염.

iii. 시스템 공격형: 운영체제 취약점 공략.

7. **트로이 목마**

= 악성 루틴이 포함된 SW.

A. 일반적으로 백도어가 포함된 SW가 트로이목마인 경우가 많다.
(단, 백도어 자체는 트로이목마가 아님.)

B. 스스로 전파되지는 않는다.

8. PUP (Potentially Unwanted Program)

= 악성코드라 단언하기는 모호하지만, 불편함을 주는 스파이웨어들.

9. **악성코드 탐지 및 대응책**

A. 네트워크 상태 점검

i. 현재 통신중인 서비스 포트들을 확인해 봐야 한다.

B. 정상 프로세스와 비교

i. 시스템의 정상 프로세스 목록을 확보하고, 이상 프로세스를 탐지한다.

C. 시작 프로그램 레지스트리 확인

i. 시스템 운영과 연관 없는 프로세스를 위한 레지스트리가 있는지 확인해야 한다.

D. 악성 코드 제거 절차

i. 악성 코드 프로세스 중단.

ii. 악성 코드의 실제 파일 찾기. -> 파일 삭제.

iii. 레지스트리 삭제.

7주차 강의 정리 끝. (6장 끝.)