5장. 악성 소프트웨어

Malicious Software

박종혁

서울과학기술대학교 컴퓨터공학과

jhpark1@seoultech.ac.kr

목차

5-1. 악성 소프트웨어의 유형

1. 악성 소프트웨어의 유형

- 악성 소프트웨어와 콘텐츠는 컴퓨터 시스템에 대한 가장 치명적인 위협

- 악성 소프트웨어 정의 : 피해자의 데이터, 응용 프로그램 또는 운영체제의 기밀성, 무결성 또는 가용성을 손상시킬 의도로 은밀하게 시스템에 삽입된 프로그램

- 악성 소프트웨어의 분류

명칭	설명
지능형 지속 위협 (APT)	• 다양한 침입 기술과 악성 소프트웨어를 사용 • 상업적 또는 정치적 목적을 위해 특정 공격 대상에 지속적이며 효율적으로 적용함 • 조직의 기밀 정보 획득을 위하여 사용됨
애드웨어 (Adware)	• 소프트웨어에 통합된 광고, 팝업 광고를 포함 • 상업 사이트로 연결하는 경로 변경으로 사용자에게 부정적인 결과를 초래할 수 있음
공격 킷 (Attack kit)	• 다양한 전파와 페이로드 기법을 사용 • 새로운 악성 소프트웨어를 자동으로 발생시키는 도구의 모음
오토 루터 (Auto-rooter)	• 새로운 기계를 원격으로 손상시키기 위하여 사용되는 공격 툴
백도어 (Backdoor)	• 특정 지점에 우회하여 정상적인 방법으로 통과하는 기법 • 프로그램이나 시스템에 있는 기능을 인증 없이 접근 가능
다운로더	• 공격을 위해 시스템에 다른 서비스/어플등을 설치하는 코드 • 악성 소프트웨어 코드가 하나의 손상된 시스템에 우선적으로 삽입된 후 대량의 악성 소프트웨어 패키지에 주입됨

명칭	설명
다운로드에 의한 구동	• 클라이언트 시스템을 공격하기 위하여 브라우저의 취약점을 이용하는 손상된 웹사이트에 있는 코드를 사용하는 공격
익스플로잇(Exploit)	• 하나의 취약점 또는 다수의 취약점에 특화된 코드(취약점 공격)
플러더(Flooder)	• 네트워크에 연결된 컴퓨터를 공격하기 위해 대량의 데이터를 만듦 • 만들어진 데이터를 이용해 DoS 공격의 형태를 수행
키로거 (Key logger)	• 공격된 시스템으로부터 자판 입력을 수집하는 프로그램
논리 폭탄 (Logic bomb)	• 공격자에 의하여 악성 소프트웨어에 삽입된 코드 • 특정 조건이 만족될 때까지 잠복하고 조건이 만족되면 논리 폭탄이 트리거 되어 공격 동작을 실행
매크로 바이러스	• 전형적으로 문서에 포함된 매크로나 스크립팅 코드를 사용 • 문서를 열거나 편집할 때 자동으로 실행됨
모바일 코드	• 다양한 장치의 플랫폼에 변경 없이 설치되어 실행되는 소프트웨어
루트킷 (Rootkit)	• 공격자가 컴퓨터 시스템에 침입하여 루트 권한 접근을 얻은 후에 사용되는 크래킹 공격 도구의 모음

명칭	설명
스팸 프로그램	대량의 전자메일을 보내기 위하여 사용됨
스파이웨어 (Spyware)	키스트로크, 스크린 데이터, 네트워크 트래픽 등을 모니터링하는 것에 의하여 컴퓨터로부터 정보를 수집하여 다른 시스템에 전송하는 소프트웨어
트로이 목마 (Trojan)	유용한 함수를 갖는 것으로 보이지만 시스템의 합법적인 인증을 이용하여 안전한 기계를 공략하는 숨겨진 악성 함수를 수행하는 컴퓨터 프로그램
바이러스 (Virus)	자신을 다른 실행 가능한 기계에 복제하려고 시도하는 악성 소프트웨어. 성공하였을 경우 이를 감염되었다고 말함. 감염된 코드가 실행될 때 바이러스 또한 실행됨
웜 (Worm)	독립적으로 실행할 수 있고 목표 시스템에서 소프트웨어의 취약성을 이용하여 네트워크상의 다른 서버에게 자신의 복사본을 전파할 수 있는 컴퓨터 프로그램
좀비-봇 (Zombi-Bot)	다른 기계를 공격하도록 감염된 기계에서 활성화되는 프로그램

- 악성 소프트웨어의 유형
  1. 목표에 도달하기 위한 확산과 전파하는 방법 기반
  2. 목표에 도달하였을 때 수행되는 동작이나 페이로드 기반

  - 전파 기법은 다른 시스템으로 계속적으로 전파되는 바이러스로 인해 기존의 실행 파일이 감염되는 것을 포함함

  - 소프트웨어 취약점, 웜에 의한 네트워크, 다운로드 등을 통하여 콘텐츠 복제가 수행됨

  - 보안 메커니즘을 사용자에게 우회하도록 유도하여 트로이 목마를 설치

  - 스미싱, 피싱 공격 등을 통해 정당한 설치를 응답하게 유도하여 이루어짐

- 악성 프로그램의 분류에 대한 최초의 접근 방법
  - 기생하기 위한 호스트 프로그램을 필요로 하는 것

  예) 바이러스 등

  - 시스템에서 실행하는 독립적인 프로그램을 갖는 것

  예) 웜, 트로이 목마 등

  - 복제되지 않는 콘텐츠

  예) 트로이 목마, 스팸 메일

  - 복제되는 악성코드

  예) 바이러스, 웜

- 목표 시스템에 도달한 악성 소프트웨어에 의하여 수행된 페이로드 동작은 시스템 또는 데이터 파일의 손상을 일으킬 수 있음
  - 서비스 및 정보의 도난, 악성코드에 자신의 숨김 기능 등을 포함

- 공격 킷(Attack kit)
  - 처음에는 악성 소프트웨어의 개발 및 보급 기술은 소프트웨어 작성자에 의한 상당한 기술이 요구되었음
    - 1990년대 초반 바이러스 생성 툴킷의 개발에 의하여 변화
    - 2000년대에는 악성코드의 개발 및 보급을 쉽게 지원할 수 있는 일반화된 공격 키트가 개발됨
  - 크라임웨어(Crimeware)로 발전되어 알려짐
    - 초보자가 결합, 선택, 보급할 수 있도록 다양한 전파 방법과 페이로드 모듈을 포함하여 제공
    - 시스템의 약점과 광범위한 패치관련 정보에서 발견된 최신 취약점을 이용하여 쉽게 사용자가 공격 방법을 정의 할 수 있음
    - 툴킷으로 만들어진 악성소프트웨어가 전문 공격자로부터 설계된 악성 소프트웨어보다 정교하진 않지만 시스템 방어를 어렵게 하는 새로운 변종의 공격 방법이 쉽고 다양하게 생성될 수도 있음

      예) 제우스, 블랙홀, 사쿠라, 피닉스 등

5-2. 지능형 지속 위협(APT)

2. 지능형 지속 위협(APT)

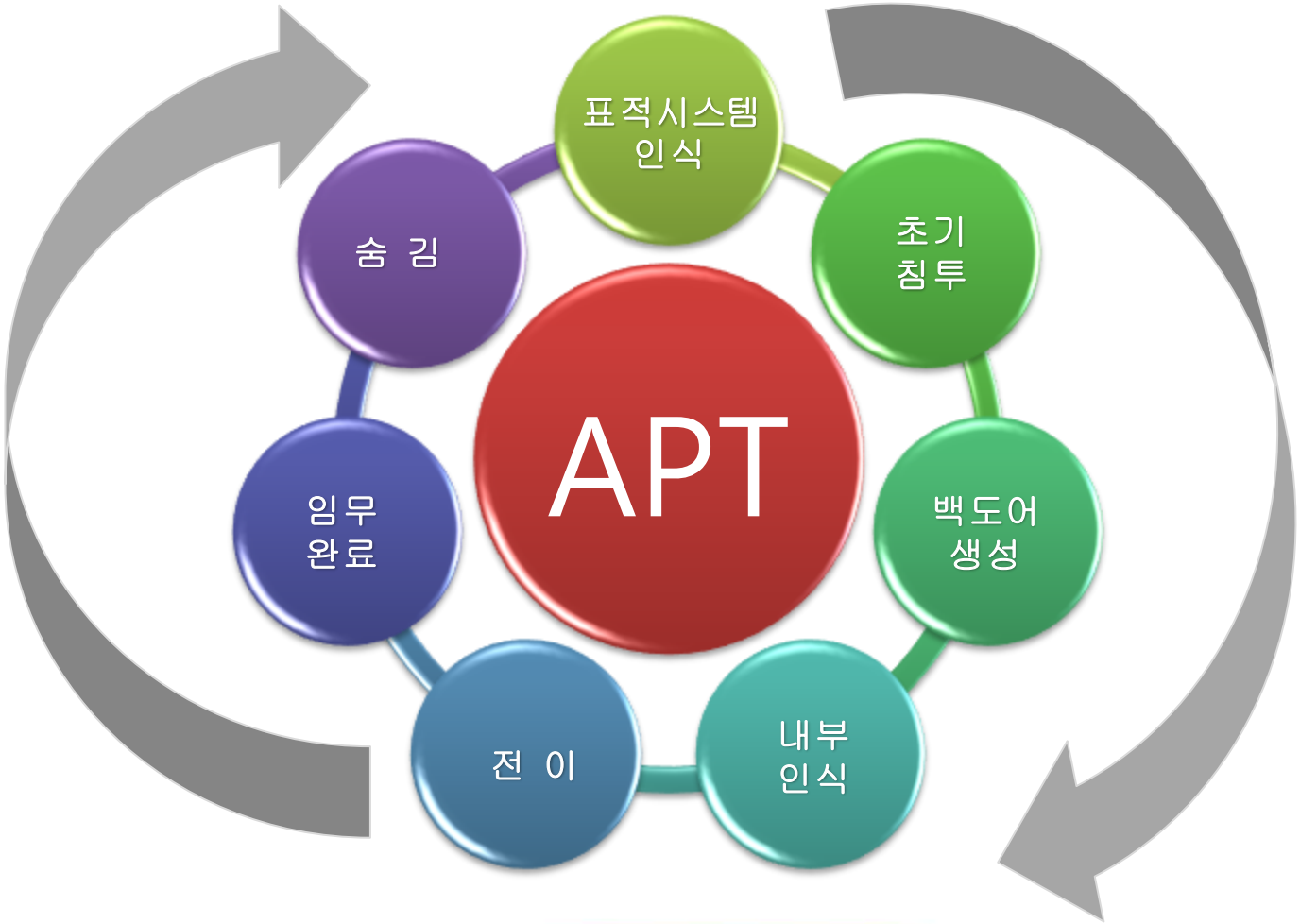
- **APT 공격이란?**
  - 다양한 IT 기술과 방식들을 이용해 조직적으로 경제적인 목적을 위해 다양한 보안 위협들을 생산해 지속적으로 특정 대상에게 가하는 공격 기법
  - 지능형 지속 위협(Advanced Persistent Threat)을 의미함

1. 지능형(Advanced)
   - 각양각색의 침략 기술과 특화된 악성코드를 이용
   - 여러가지 공격 요소들이 선정된 목표에 적합하도록 알맞은 공격 기술을 선택

2. 지속(Persistent)
   - 성공률을 극대화하기 위하여 오랜 시간에 걸쳐서 선정된 목표에 대한 공격
   - 목표에게 목적이 달성 될 때까지 다양한 형태의 공격이 점진적으로 또는 은밀하게 이루어짐

3. 위협(Threat)
   - 공격자는 보통 풍부한 자원 등을 가진 개인/조직들을 목표로 삼음
   - 공격에 성공 할 경우 매우 큰 규모의 피해를 초래함
   - 컴퓨터 기술의 발전으로 자동화된 공격 도구와 향상된 공격기법은 위협의 수준을 증가시킴

- APT공격은 표적 지향의 공격이기 때문에 공격방식을 규정할 수는 없지만 다수의 공격사례를 통해 7가지의 공격 요소를 도출



1. 표적시스템 인식 (Targeted System Recognition)
   - 침입 전 표적 시스템에 대한 정보 수집 활동
   - 기술적인 방법는 포트 스캔을 통해서 침입 가능한 포트(Port)를 검색
   - 비기술적인 방법으로는 Social engineering을 통해서 보안 정보에 접근권한이 있는 담당자의 이메일 주소 등을 획득하여 이후 Spear phishing, USB감염 등을 위한 수단을 사용

2. 초기 침투 (Initial Intrusion)
   - 실질적인 APT 공격이 이루어지는 단계로써 표적 시스템에 초기 침투를 수행
   - 수집한 정보를 토대로 Spear phishing 공격을 시도
   - 보안담당자, 네트워크담당자와 같은 표적 시스템을 관리하는 담당자가 주요 표적
   - 공격자는 주로 악성코드에 감염된 웹사이트나 문서를 이메일에 첨부하여 표적에게 전송

3. 백도어 생성 (Backdoor Establishment)
   - 최초 침입 이후 공격자는 향후에 더욱 쉽게 시스템에 접근하기 위하여 백도어를 생성
   - 공격자는 자신이 생성한 C&C (Command and Control) Server와 표적 시스템이 직접적으로 정보를 교환
   - 정상적인 트래픽처럼 보이는 암호화된 데이터를 C&C서버와 주고받아 Antivirus 제품의 감시망을 벗어나 정보를 쉽게 획득

4. 내부 인식 (Internal Recognition)
   - 신뢰된 컴퓨터를 통해서 네트워크 스캐닝을 시도
   - 내부네트워크를 통해서 정찰이 이루어지며 신뢰되는 관계를 통해 표적 시스템의 네트워크 정보를 획득

5. 전이 (Metastasis)
   - 내부 정찰을 통해서 얻은 정보를 이용하여 주요 정보가 있는 End-Point로 악성코드를 전이동일 네트워크를 사용하는 모든 End-Point를 감염
   - 공격자는 원하는 정보를 유출시킬 준비를 마침

6. 임무 완료 (Mission Complete)
   - 공격자의 목적에 맞는 행동을 성공적으로 수행하는 단계
   - 공격자의 목적은 대표적으로 시스템 파괴와 기밀 정보의 유출
   - Antivirus에 탐지되지 않게 하기 위해서 정보는 암호화, 파일압축, 파일 분할 등의 기법을 사용

7. 숨김 (Hiding)
   - APT 공격은 지속적인 공격을 위해서 특별한 이벤트가 없는 동안 악성 소프트웨어를 숨김
   - 몇몇의 APT 공격은 Rootkit을 포함하고 있으며 이를 통해 시스템의 루트 권한을 획득
   - Rootkit을 이용하여 Malware의 행동을 숨김

- **APT 공격 사례**

구분	정의
IceFog	• 일본과 대한민국의 정부출연연구소, 방위산업체, 조선해양사, 통신사 및 고도의 기술을 보유한 회사 등을 타겟으로 활동하고 있는 APT공격 • 스피어피싱 이메일이 이용, 이메일에 첨부된 악성코드를 열거나 악성 웹사이트를 방문 • 시스템을 감염시킨 뒤 공격자는 감염 시스템의 특성을 파악하고 확인하기 위해 폴더 목록, 어댑터 목록, IP구성, 네트워크 정보 등을 윈도우 주소록, WAB파일, HWP, XLS, DOC등 문서파일 그리고 사용자 계정 자격 정보 등을 탈취
Net Traveler	• 항공 우주 관련 기술 및 에너지 생산 관련 데이터나 통신 관련 데이터를 탈취 • 이메일 주소를 확보한 후 악성코드가 포함된 첨부 파일의 클릭을 유도하는 메시지를 발송 • 이메일에 첨부되는 마이크로소프트 오피스 형식(확장자 doc, xls 등) 및 PDF 파일에 잠복 • 시스템에 침입할 경우 시스템 내 모든 정보에 대해 모니터링 실시 뿐만 아니라 데이터 추출, 사용자의 키 입력 기록 확인 및 각종 문서 파일의 탈취 등도 가능

구분	정의
Stuxnet	• 시스템을 감염시켜 원자로의 PLC(programmable logic controller)에 악의적인 프로그램을 삽입 • 사용된 취약점은 Window shell LNK 취약점, Window server service 취약점, Window printer spooler 취약점, 공유 네트워크 서비스 취약점 • 장치인 USB를 공격에 이용하여 네트워크 망이 분리되어 있는 시스템에도 악성코드를 전파 • Stuxnet은 대상시스템이 아닌 경우에는 특별한 활동을 하지 않았으며 자신의 존재를 지우기 위해서 Rootkit을 사용
Duqu	• Duqu는 정보 유출을 목적으로 하여 표적 시스템의 정보를 수집 • 감염된 Micro Office 문서를 첨부하는 Spear phishing을 통해 초기 침투를 시도 • 성공적으로 시스템에 잠입하고 백도어를 설치함으로써 C&C (Control and Command) 서버와 통신 • 패스워드와 같은 주요 정보를 수집하며, 수집된 정보는 공격자가 네트워크상의 다른 시스템에 접근할 권한을 얻는데 사용

구분	정의
Red October	• 동유럽과 중앙아시아의 외교/정부의 에너지/원자력 기관, 교역/항공 우주 산업 전반을 표적 • 감염된 Microsoft Office 제품의 파일을 메일로 첨부하여 보냄으로써 표적 시스템에 침입 • 처음 유출된 정보를 바탕으로 동일 네트워크의 다른 기밀 시스템에 침입 • IPhone, Nokia, Windows Mobile를 Operating System으로 사용하는 스마트 기기의 데이터와 이동식 디스크에 있는 정보를 유출
Mask	• 다양한 환경에 적응하는 APT 공격으로 Rootkit을 포함하고 있으며 Window 32bit, Window 64bit, Mac OS X, Linux환경에서 동작 • Adobe Flash의 취약점을 이용한 Spear phishing을 통해 표적 시스템에 초기 침입 • 설치된 악성코드는 탐지를 회피하기 위해 표적 시스템 내의 Antivirus 를 찾아내고 동작을 휴내 • 네트워크 트래픽, 키스트로크, 스카이프 대화, WiFi 트래픽 분석, 노키아로 부터의 패치 정보, 스크린 캡쳐의 로그정보 등을 수집

- **APT 공격 특징**
  - 특정 조직에 최적화된 공격 수행
  - 충분한 시간과 비용을 투자
  - 조직 및 구성원 개인에 대한 충분한 정보 수집 (사회공학적인 방법 이용)
  - 탐지 회피 기법을 병행하여 공격 수행
    - low and slow 전략
    - 알려지지 않은 악성코드(Zero-Day Attack) 사용
    - 이상징후를 파악하지 못하도록 장기간에 걸쳐 은밀히 활동
  - 다양한 방향으로 공격, 사용자 및 Endpoint에 집중

- **APT 공격 대응 방안**
  - 공격자가 원하는 정보에 접근하기 까지 소요 시간 지연
    - 네트워크 분리 (망 분리)
    - 내부 시스템 인증 강화
  - 공격자가 원하는 정보에 접근하기 이전 단계에서 탐지/제거
    - 알려지지 않은 악성코드(Zero-Day Attack) 탐지 / 제거
    - 악성코드/원격접속/명령 및 제어 지점 접근 트래픽 탐지/차단

- **APT 공격 기법**

관찰	• 공격자들은 표적으로 삼은 조직, 시스템, 프로세스, 협력업체를 포함해 사람들을 파악하기 위해 수개월에 걸쳐 공격 목표를 분석함
사회공학	• 공격자가 특정 조직의 내부자를 노린다면 공격자는 사전에 개인 SNS, P2P, 홈페이지 등을 검색하여 공격대상의 개인정보와 관련 키워드 등을 수집함 • 공격자는 피싱 메일을 보내는 방식 등을 이용하여 실수나 부주의로 링크를 클릭하여 첨부된 파일을 열도록 하는 사회공학 기법을 접목시킴
제로데이 취약점	• 개발자들이 보안패치 등을 제공하기 전까지 소프트웨어는 특정 알려진 취약점에 대해 무방비 상태에 노출됨
수동	• 각각의 개별 시스템과 사람을 표적으로 삼아 고도의 정교한 수동 공격을 감행 • 일반적인 해킹 공격에서는 효과를 극대화하기 위해 자동화를 선택하는 방식과 비교해 볼 때 대조적임
지속적인 분석	• 전략적인 기회를 포착하기 위해 공격자가 분석하는 활동
유출	• 웹 메일 혹은 암호화된 패킷이나 압축파일 형태로 공격자에게 정보를 전송
중단	• 소프트웨어 및 시스템의 강제 종료 또는 악의적인 시스템 중단 문제를 유발

5-3. 전파: 손상된 내용,바이러스

3. 전파 : 손상된 내용, 바이러스

- 바이러스의 성질
  - 프로그램을 수정하는 것에 의하여 다른 프로그램들을 감염시키는 일련의 소프트웨어
    - 원본 코드가 바이러스 코드를 복제하기 위한 루틴을 갖도록 수정함
    - 바이러스 코드는 다른 내용을 감염시키는 데 계속 사용될 수 있음
    - 초기 감염 형태는 의심되지 않는 사용자들로부터 프로그램, 디스크 파일, USB 스틱을 상호 교환하는 것에 의하여 컴퓨터에서 컴퓨터로 전염
    - 인터넷의 보편화로 네트워크를 통해서도 전염됨
  - 실행할 수 있는 프로그램에 첨부된 바이러스는 프로그램에 허용된 어떠한 것이든 할 수 있음
  - 바이러스 코드가 실행되면 사용자의 권한에 의해 허용된 파일이나 프로그램을 지우는 것과 같은 다양한 기능을 수행할 수 있음

- 컴퓨터 바이러스의 유형

1. 감염 메커니즘
   - 바이러스가 퍼지거나 전파하는 수단으로서 감염 벡터로 표현됨
2. 트리거
   - logic bomb 공격
   - 페이로드가 활성화하거나 전달될 때를 결정하는 이벤트 혹은 조건
3. 페이로드
   - 바이러스를 수행하는 부분
   - 손상을 포함할 수 있음

- 많은 악성코드의 형태는 각 부분에 하나 이상의 변형을 포함

- 바이러스의 수행 4단계
  1. 휴지 단계
     - 바이러스는 활성화되지 않음
     - 바이러스는 날짜, 다른 프로그램이나 파일의 생성, 디스크 용량의 초과 등과 같은 특정한 사건에 의하여 활성화됨
     - 모든 바이러스가 휴지단계를 거치는 것은 아님
  2. 전파 단계
     - 바이러스는 자신을 복제하여 다른 프로그램 또는 디스크 상의 시스템 영역에 삽입
     - 바이러스는 탐지를 피하기 위하여 변형되어 복제된 바이러스는 복제 이전의 바이러스와 같지 않을 수 있음
     - 감염된 프로그램은 바이러스의 복제를 실행하고 다시 전파 단계로 들어감

3. 트리거 단계

- 바이러스는 의도한 기능을 수행하도록 설정
- 휴지 단계에서처럼 다양한 시스템 사건에 의하여 활성화
- 바이러스가 자기 자신을 복제하는 경우도 하나의 사건으로 포함

4. 실행 단계

- 바이러스내의 의도한 기능이 수행
- 단순히 스크린에 메시지(광고)를 실행하는 등 시스템에 무해할 수도 있고 데이터 파일과 프로그램 등을 파괴하는 것과 같이 손상을 입힐 수도 있음

- 실행 가능한 바이러스 구조
  - 전통적인 기계 실행 코드 바이러스는 어떤 실행 프로그램의 열에 또는 뒤에 붙여질 수 있거나 다른 형태로 끼워질 수 있음
  - 코드의 첫 번째 줄은 바이러스에 의해 잠재적인 피해 프로그램이 이미 그 바이러스에 감염되었는지 안 되었는지 결정하기 위해 사용되는 특별한 표시
  - 프로그램이 시작될 때 제어 권한은 우선적으로 바이러스 코드를 포함한 주요한 활동 영역으로 옮겨짐
  - 바이러스는 우선적으로 감염되지 않은 실행 파일을 찾고 그 파일을 감염시킴
  - 요구되는 트리거 조건이 만족된다면 바이러스는 페이로드를 실행
  - 바이러스는 활성화 된 후에 원래 프로그램으로 제어 권한을 옮김

```
program V
1234567;

procedure attach-to-program;
begin
   repeat
        file := get-random-program;
   until first-program-line ≠ 1234567;
   prepend V to file;
end;

procedure execute-payload;
begin
   (* perform payload actions *)
end;

procedure trigger-condition;
begin
   (* return true if trigger condition is true *)
end;

begin (* main action block *)
   attach-to-program;
   if trigger-condition then execute-payload;
   goto main;
end;
```

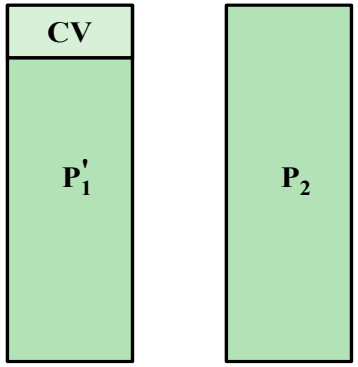
(a) 단순 바이러스

```
program CV
1234567;

procedure attach-to-program;
begin
   repeat
        file := get-random-program;
   until first-program-line ≠ 1234567;
   compress file;    (* t1 *)
   prepend CV to file;   (* t2 *)
end;

begin (* main action block *)
   attach-to-program;
   uncompress rest of this file into tempfile;   (* t3 *)
   execute tempfile;   (* t4 *)
end;
```

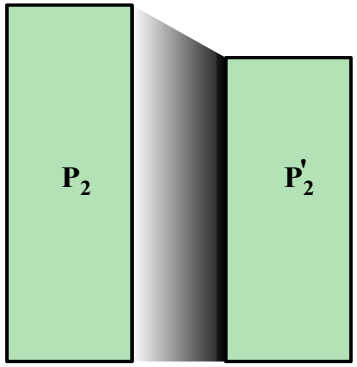
(b) 압축 바이러스

바이러스 로직 예

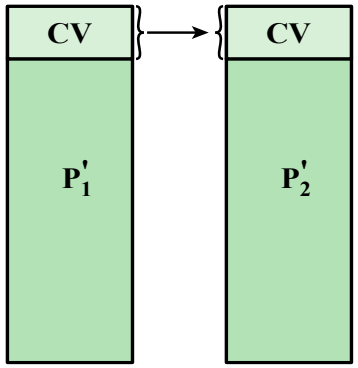
- 프로그램은 시간 t_0에서 시작
- P_1은 바이러스 CV에 감염된 프로그램이고, P_2는 CV에 감염되지 않은 깨끗한 프로그램
- P_1이 실행될 때 제어는 바이러스로 이동되고 다음의 단계를 수행
  - t1 : 감염되지 않은 파일 P_2에 대하여 바이러스는 우선 원래의 프로그램보다 바이러스 크기만큼 짧은 P'_2파일을 만들기 위해 압축함
  - t2 :복사된 CV를 압축된 프로그램 앞에 붙임
  - t3 : 감염된 원래의 프로그램의 압축 버전(P'_1)은 압축하지 않음
  - t4 : 압축되지 않은 원래의 프로그램 P_1이 실행됨



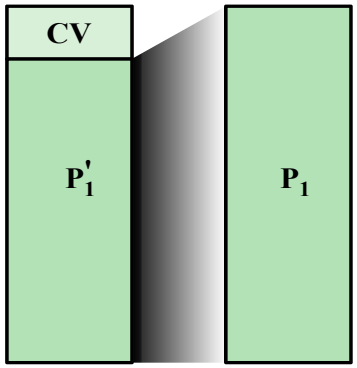
t_0:P′$_1$은 P_1의 감염된 버전
P_2는 감염되지 않음



t_1:P_2는 P′$_2$로 압축



t_2:CV는 P′$_2$에 붙음



t_3:P′$_1$이 원래 프로그램 P_1에
압축해제 된다

압축 바이러스

- 바이러스 분류
  - 표적에 의한 바이러스 분류
    1. 부트 섹터 감염자
       - 마스터 부트 레코드 또는 부트 레코드에 감염되고, 시스템이 바이러스를 포함하고 있는 디스크에서 부팅할 때 전파됨
    2. 파일 감염자
       - 운영체제 또는 쉘이 실행하기 위한 파일을 감염시킴
    3. 매크로 바이러스
       - 응용프로그램에 의해서 해석되어진 매크로 또는 스크립트 코드를 갖는 파일을 감염시킴
    4. 다중 부분 바이러스
       - 여러 가지 방법을 혼합하여 파일을 감염시킴
       - 여러 종류의 파일을 대상으로 함

- 은닉 전략에 의한 바이러스 분류
  1. 암호화 바이러스
     - 내용을 모호하게 하기 위해 암호화를 사용하는 바이러스 형태
     - 바이러스의 일부분은 바이러스의 나머지 부분을 암호화하는 임의의 암호키를 생성
     - 키는 바이러스와 함께 저장됨
  2. 스텔스 바이러스
     - 소프트웨어에 의한 감지로부터 숨기도록 설계된 바이러스 형태
  3. 다형성 바이러스
     - 바이러스 검사 프로그램을 격파시키기 위해서 기능적으로 동일
     - 탐지 알고리즘 기법을 피하기 위해 다른 비트 패턴을 가진 사본을 생성
  4. 변성 바이러스
     - 다형성 바이러스와 같이 모든 감염을 변화시킴
     - 검출의 어려움을 증가시키면서 다수의 형질 전환 기술을 이용하여, 변성 바이러스를 각각의 반복에서 완전히 재기록함

- 매크로 및 스크립팅 바이러스
  - 1990년대 중반에 매크로와 스크립트 코드 바이러스가 가장 널리 퍼짐
  - 문서 형태의 다양한 활성 콘텐츠를 지원하는 데 사용되는 스크립트 코드를 감염시킴
  - 워드 프로세싱 문서 및 파일의 다른 유형에서 임베디드, 매크로 언어의 스크립트를 사용하여 활성 콘텐츠를 지원하는 장점이 있음
  - 일반적으로 사용자는 반복적인 작업을 자동화하기 위하여 매크로를 사용하고 키 입력을 저장
  - 동적 콘텐츠, 형태 검증, 이들 문서와 연관된 다른 유용한 작업을 지원하기 위해 사용됨

- 매크로 및 스크립팅 바이러스

1. 매크로 바이러스는 플랫폼 독립적임

2. 많은 매크로 바이러스는 Microsoft Word 문서, Microsoft Office 문서에서 매크로나 Adobe PDF 문서에서의 스크립트 코드와 같이 일반적으로 사용되는 응용의 활성 내용을 감염시킴

3. 응용프로그램을 지원하는 하드웨어 플랫폼 및 운영체제에 감염시킴

4. 매크로 바이러스는 실행 코드가 없는 문서에 감염시킴

5. 컴퓨터 문서은 일반적으로 공유되어 사용되기 때문에 매크로 바이러스는 전자메일 등을 통해 쉽게 확산됨.

6. 매크로 바이러스는 시스템 프로그램보다는 사용자 문서를 감염시키기 때문에 기존의 파일 시스템 액세스 제어는 전파를 방지하는 데 제한적임

5-4. 전파: 사회 공학 , 스팸 전자메일, 트로이 목마

4. 전파: 사회 공학, 스팸 전자메일, 트로이 목마

- 스팸 전자메일
  - 일부 스팸은 합법적인 메일 서버로부터 보내지는 반면 대부분의 최신 스팸은 사용자 시스템이 사용하는 봇넷을 이용
  - 피싱 공격과 악성코드를 전파하는 주요 수단
  - 사용자의 시스템상에 악성코드를 설치하기 위해 보내는 파일의 소프트웨어 취약성 이용
  - 사용자를 온라인 은행 사이트와 같은 합법적인 사이트로 가장하여 가짜 웹사이트로 연결시키고, 이를 이용해여 사용자의 로그인 정보와 비밀번호를 알아냄
  - 사용자에게 전자메일과 첨부 파일을 볼 것인지 또는 프로그램을 수행할 것인지에 대한 적극적인 선택을 요구

- 트로이 목마 (Trojan horse)
  - 원하지 않거나 해로운 기능을 행하는 숨겨진 코드를 포함한 프로그램이나 유틸리티
  - 공격자들이 직접적으로 수행하지 않고 간접적으로 기능을 수행하기 위하여 사용
  - 제작자는 게임이나 유용한 유틸리티 프로그램, 배포되는 사이트, 앱 스토어 등 잘 알려진 소프트웨어를 이용하여 실행하도록 유도
  - 사용자의 도움 없이 자동으로 설치 및 실행되기 위하여 소프트웨어의 취약성을 이용함
  - 자기 복제를 하지 않음
  - 트로이 목마의 세 가지 공격 형태
    1. 원래 프로그램 기능의 수행을 계속하면서 분리된 악의적인 행동을 추가적으로 수행
    2. 원래 프로그램 기능 수행을 계속하지만 악의적인 행동을 수행하기 위하여 업데이트를 통해 기능을 수정 또는 다른 악의적인 행동을 추가
    3. 원래 프로그램의 기능을 완벽하게 대치하여 악의적인 기능 수행

- 모바일 트로이목마 (Mobile phone trojans)
  - Skuller의 발견하였으며 2004년에 처음 출현
  - 이동식 웜과 같이 주 목표 디바이스는 스마트폰
  - 아이폰 O/S의 여러 버전은 그래픽 유형이나 PDF 취약성을 이용
  - 휴대전화의 잠금 장치를 해체시키는 주된 수단으로 이용

5-5. 페이로드: 시스템파괴

5. 페이로드: 시스템 파괴

- 악성 소프트웨어가 목표 시스템상에서 활성화되면 다음 관심은 이 시스템상에서 어떤 활동을 할 것 인가임
- 몇몇 악성 소프트웨어는 비존재 또는 비활동적인 페이로드이고, 이것의 목적은 감염 대상을 확산 시키는 것임
- 많은 바이러스와 웜에서 보여준 초창기의 페이로드는 트리거의 조건이 만족될 때 감염된 시스템의 데이터 파괴 수행함
- 사용자 시스템 상에 원하지 않는 메시지나 내용을 보여주거나 더 심각하게는 다른 변종이 시스템상에서 대한 손상을 입힘
- 페이로드의 주 동작은 컴퓨터 시스템의 소프트웨어, 하드웨어 또는 사용자 데이터의 무결성을 공격함
- 페이로드는 바로 발생하지는 않을 수도 있지만 특별한 트리거 조건이 부합될 때 활성화됨

- 데이터 파괴
  - 체르노빌 바이러스 (Chernobyl virus)
    - 윈도우 95, 98의 메모리상에서 기생하면서 데이터를 파괴시키는 초창기의 예
    - 1998년에 처음 발견되었으며, 감염은 파일을 열 때 실행됨
    - 트리거 데이터가 도착할 때 감염된 시스템상의 하드디스크의 처음 수 megabyte에 0의 데이터를 쓰는 것에 의하여 원래 데이터가 삭제되며, 궁극적으로는 전체 파일시스템의 파괴를 초래함
  - Klez mass-mailing warm
    - 윈도우 95가 XP 시스템을 감염시키는 파괴적인 웜의 초창기 예
    - 2001년 10월에 처음 발생하였고, 전자메일 주소를 주소록이나 시스템상의 파일 등에서 복사하여 퍼뜨림
    - 안티 바이러스 프로그램의 실행을 멈추거나 삭제할 수도 있음
  - 랜섬 웨어 (Ransomware)
    - 사용자의 데이터를 암호화하여 이 정보를 복구하는 데 필요한 키를 받는 대가로 돈을 요구함

    예) Cyborg Trojan (1989), Gpcode Trojan (2006)

- 시스템과 하드웨어 손상
  - 시스템 파괴 페이로드의 변종은 물질적인 장치의 손상을 목표로 하였음
  - 체르노빌 바이러스는 데이터를 파괴할 뿐만 아니라 컴퓨터를 부팅하는 데 사용되는 BIOS 코드에 중복 쓰기를 시도함
  - BIOS 칩을 다시 프로그램하거나 교체될 때까지 시스템을 사용할 수 없게 됨
  - Stxnet 웜은 특수 산업 제어 시스템 소프트웨어를 목표로 함
  - 제어 시스템이 감염되었다면 제어 시스템이 정상적으로 동작할 수 없도록 원래의 시스템 소프트웨어를 다른 코드로 변경시켜서 부착된 장비의 고장을 초래함

- 논리 폭탄 (Logic Bomb)
  - 데이터 파괴 악성 소프트웨어의 주요 구성 요소
  - 조건이 만족되었을 때 폭발하는 악성 소프트웨어에 내장되어 있는 코드
  - 논리 폭탄에 트리거로써 사용될 수 있는 조건의 예
    - 시스템 상에서 파일 또는 장치의 존재 또는 부재
    - 특정 날짜 혹은 업데이트 시기
    - 특정 소프트웨어의 구성 요소 실행
    - 응용프로그램을 실행하는 특정 사용자
  - 트리거가 되면 데이터나 파일 전체를 변경시키거나 삭제하여 기계를 다운시키거나 손상을 초래함

5-6. 페이로드: 공격 에이전트, 좀비, 봇

6. 페이로드: 공격 에이전트, 좀비, 봇

- 좀비, 봇은 페이로드의 한 종류로 공격자의 사용에 대하여 감염 시스템의 컴퓨팅이나 네트워크의 자원을 개조하는 악성 소프트웨어임
- 비밀리에 다른 인터넷이 부착된 컴퓨터를 인수한 뒤 그 컴퓨터를 다른 공격을 관리하거나 시작하는 데 사용
- 다른 매개체를 이용하므로 추적하기 어려움
- 로봇은 평범한 제 3자를 통해 수백 또는 수천 대의 컴퓨터에 이식
- 로봇 무리는 종종 상호작용 방식으로 행동하는데 이러한 집합을 봇넷(botnet)이라고 함
- 감염된 시스템의 무결성과 가용성을 공격함

- 봇의 사용
  - 봇의 용도
    1. 분산 서비스 거부(DDoS) 공격
       - 사용자에 대한 서비스의 손실을 야기하는 컴퓨터 시스템 또는 네트워크에 대한 공격
    2. 스팸
       - 봇 넷 또는 수천의 봇의 도움으로 공격자는 대량의 스팸 메일을 보낼 수 있음
    3. 스니핑 트래픽
       - 봇은 타협된 시스템에 의해 통과된 관심 텍스트 데이터를 보고 전송하기 위하여 패킷 스니퍼를 이용할 수 있음
       - 스니퍼는 이름이나 비밀번호 같은 민감한 정보를 검색하는 데 사용됨

4. 로깅
   - 타협된 기계가 암호화된 통신 채널(예, HTTPS, POP3S)을 사용할 경우, 희생 컴퓨터의 네트워크 패킷은 패킷을 해독하는 적절한 키가 없기 때문에 쓸모가 없음
   - 감염된 컴퓨터에 키 입력을 캡처하는 키 로거를 사용
   - 공격자는 공격대상의 기밀한 정보를 검색할 수 있음

5. 새로운 악성 소프트웨어 전파
   - 봇넷은 새로운 봇을 전파하는 데 사용됨
   - 모든 봇들은 HTTP나 FTP를 통해 파일을 다운받아 실행하는 메커니즘을 구현하기 때문에 전파가 매우 쉬움
   - 웜과 메일 바이러스의 전진 기지로서 동작하는 10,000개의 호스트를 가진 봇넷은 더 빨리 전파하여 더 많은 해를 입힘

6. 광고 추가 및 BHO 설치
   - 가짜 웹사이트에 광고를 설정하는 것에 의하여 동작함
   - 웹사이트의 운영자는 광고 클릭에 대해 지불하도록 일부 호스팅 업체와 계약을 맺기도 함
   - 봇넷의 도움으로 이런 클릭이 수천 개의 봇이 팝업을 클릭하도록 자동화 될 수 있음

7. IRC 채팅 네트워크 공격
   - 클론공격: 봇넷은 인터넷 릴레이 채팅(IRC)에 대한 공격에 사용
   - 피해자는 복제된 봇에 의해서 수천 개의 봇이나 수천 번의 채널 요구에 의하여 트래픽이 폭증
   - 피해 IRC 네트워크는 DDoS 공격과 유사하게 마비를 초래

8. 온라인 설문/게임 조작
   - 봇넷으로부터 조작하기 쉬움
   - 각각의 봇은 다른 IP 주소를 가지고 있기 때문에 모든 투표는 진짜 사람이 투표를 한 것과 같은 신뢰성을 가짐

- 원격 제어 기능
  - 웜은 자기 스스로 전파하고 활성화하는 반면 봇은 명령 및 제어(C&C) 서버 네트워크에 의해 제어됨
  - 원격 제어 기능의 초기 수단: IRC 서버가 사용
  - 모든 봇은 이 서버의 특정 채널에 가입하고 명령으로써 들어오는 메시지를 처리함
  - 최근 봇넷은 IRC 기법을 피하고 HTTP와 같은 프로토콜을 통해 비밀 통신 채널을 사용
  - 제어 모듈이 인터넷의 특정 위치로부터 악성 파일을 다운로드하고 실행하도록 봇에게 지시하는 변경된 명령어를 발생할 수 있음

5-7. 페이로드: 정보 도용, 키로거, 피싱, 스파이웨어

7. 페이로드: 정보 도용, 키로거, 피싱, 스파이웨어

- 자격 증명 도난, 키로거, 스파이웨어
  - 사용자는 네트워크 패킷의 모니터링에 의해 캡처되는 것을 보호하기 위해 암호화된 통신 채널(예 HTTPS, POP3S)을 이용
  - 금융, 게임 및 이와 관련된 사이트에 자신의 로그인 및 비밀번호 자격 증명을 보냄
  - 공격자는 감염된 컴퓨터에 중요한 정보를 모니터링할 수 있도록 감염된 기계에 키 누름을 캡처할 수 있는 키로거를 설치
  - 이것은 감염된 시스템에 입력된 모든 텍스트의 사본을 수신하기 때문에 키로거는 전형적으로 원하는 키워드에 가까운 정보 (예 :"로그인", "비밀번호", "paypal.com")만을 반환하도록 필터링 기법을 구현
  - 키로거 사용의 대처로 일부 은행 및 다른 사이트들은 패스워드와 같은 중요한 정보를 그래픽 애플릿을 사용하여 전환

- 피싱 및 신원 도용
  - 초기에는 사용자의 로그인 및 패스워드 인증 정보를 수집하는 데 사용
  - 최근 사적 또는 개인적인 정보를 수집하기 위하여 사용
  - 공격자는 가짜 웹사이트로 연결되는 스팸 메일의 URL을 포함시킴
  - 스팸 메일은 가짜 웹사이트로 사용자를 유도하거나 밀봉된 형식을 작성하여 호기심을 자극하여 접근 하도록함
  - 피싱 공격
    - 충분한 정보가 주어지면, 공격자는 다른 자원에 대한 신용이나 민감한 접근을 얻을 목적으로 사용자의 신원을 추측할 수 있음
    - 신뢰할 수 있는 주체로 가장하여 사용자의 신뢰를 활용하는 사회공학적 기법을 이용함

- 정찰, 간첩 및 데이터 빼오기
  - 자격 증명 및 신원 도용은 보다 일반적인 정찰 페이로드의 특별한 경우임
  - 목표: 원하는 유형의 정보를 획득하여 공격자에게 반환
  - 2009년에 기술, 보안 및 방위 계약자 회사 시스템 내의 소스코드 스토리지에 접근하기 위해 트로이 목마를 사용
  - APT 공격은 공격자의 목표 조직 및 개인으로부터 대량의 정보가 손실되는 결과를 초래할 수 있음
  - 정보를 빼오는 것을 탐지하고 차단하기 위해서는 정보에 대한 접근을 관리하거나 조직의 네트워크를 통한 전송을 관리하는 "데이터 손실"에 대한 기술적인 대응책이 필요

5-8. 페이로드: 은신, 백도어, 루트킷

8. 페이로드: 은신, 백도어, 루트킷

- 백도어
  - 트랩도어라고 알려져 있으며, 평소의 보안 접속 절차를 통하지 않고 접속을 허용하는 지점
  - 유지 관리 후크: 프로그래머는 수년 동안 프로그램을 디버그하고 시험하기 위하여 백도어를 사용
  - 백도어를 두는 이유: 프로그램을 신속하게 디버그 하기 위해 개발자가 특수한 특권을 얻거나 모든 설정과 인증을 회피하는 것을 선호
  - 부도덕적인 외부자가 인증되지 않은 접속을 위하여 백도어를 사용할 때 매우 위협적임
  - 최근 백도어는 공격자가 감염된 시스템에 연결하고 명령을 내릴 수 있는 비표준 포트의 네트워크 서비스를 도청하는 데 사용될 수 있도록 구현

- 루트킷
  - 가능한 최대의 범위까지 증거를 숨기면서 관리자의 권한으로 해당 시스템에 대한 은밀한 접속을 유지하려는 시스템에 설치된 프로그램의 집합
  - 악의적이고 은밀한 방법으로 호스트의 표준 기능을 변화
  - 공격자는 루트에 접속해서 시스템을 완전히 제어하고 프로그램과 파일, 모니터 프로세스를 추가하거나 바꿀 수 있음
  - 네트워크 트래픽 위조 및 변조가 가능하며 백도어에 접속할 수 있음
  - 공격자는 자신의 존재를 숨기기 위하여 시스템에 변화를 줌으로써 사용자에게 루트킷의 존재와 루트킷이 무엇을 변화시켰는지 알기 어렵게 함
  - 숨기는 방법으로는 백신프로그램의 프로세스, 파일, 컴퓨터의 레지스트리 탐색 및 보고하는 것에 대한 기법을 파괴

- 루트킷의 여러 특성에 따른 공격 유형
  1. 지속성
     - 매번 시스템을 부팅시킴
     - 레지스트리 또는 파일 시스템과 같은 지속적인 저장소 안의 코드를 저장시키고, 코드는 사용자의 간섭 없이 실행되는 방법을 구현함
     - 지속적인 저장소를 스캔하는 것에 의하여 쉽게 감지될 수 있음을 의미

  2. 메모리 기반
     - 지속적인 코드를 가지지 않아서 재시동으로 생존할 수 없음
     - 메모리에 있기 때문에 감지하기 어려움

  3. 사용자 모드
     - API에 대한 호출을 가로채서 반환된 결과를 수정함

     예) 응용프로그램이 디렉터리 도청을 실행할 때 반환된 결과는 루트킷과 관련된 파일을 식별하는 항목을 포함하지 않음

4. 커널 모드
   - API로 가는 호출을 가로챌 수 있음
   - 활성 프로세스의 커널 목록에서 제거하여 악성 소프트웨어의 존재를 숨길 수 있음

5. 가상 기계 기반
   - 루트킷의 가상 기계 기반 타입은 가벼운 가상 기계 모니터를 설치하고, 가상 기계에서 운영체제를 실행
   - 가상화된 시스템에서 발생하는 상태 및 사전을 투명하게 가로채고 수정

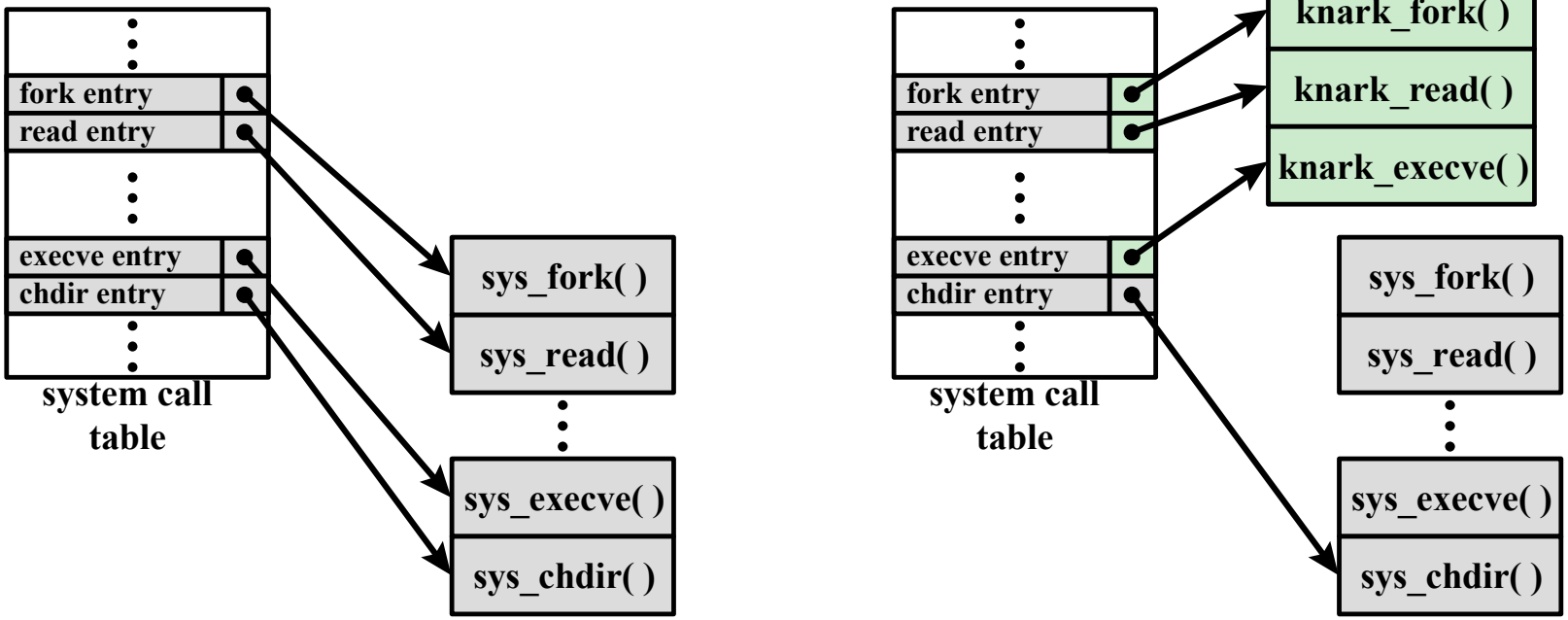
6. 외부 모드
   - 악성 소프트웨어는 목표 시스템의 정상적인 운영 모드 외부(BIOS 혹은 시스템 관리 모드)에 위치하여 하드웨어에 직접적으로 접근할 수 있음

- 커널 모드 루트킷
  - 최근 탐지를 더 어렵게 하기 위하여 커널 내부를 변화시키고 운영체제 코드와 공존하도록 발전하였음
  - 사용자 계층에서 동작하는 프로그램은 시스템 호출을 통해 커널과 상호 연동하기 때문에 프로그램의 시스템 호출 기능(함수)은 은폐하려는 커널 계층 루트킷의 주요 목표 대상임
  - 커널 루트킷의 동작 과정
    1. 각 시스템 호출은 유일한 시스템 호출 번호가 할당됨
    2. 사용자 모드 프로세스가 시스템 호출을 실행할 때 프로세스는 호출 번호에 의하여 시스템 호출을 함
    3. 커널은 시스템 호출 루틴당 하나의 엔트리를 지닌 시스템 호출 테이블을 관리
    4. 각 엔트리는 관련된 루틴에 대한 포인터를 포함함
    5. 시스템 호출 번호는 시스템 호출 테이블에서 색인으로써 사용됨

- 시스템 호출을 변경시키는 기술
  1. 시스템 호출 테이블 변경
     - 공격자는 시스템 테이블에 저장된 선택된 시스템 호출 주소를 변경함
     - 합법적인 루틴이 루트킷으로 대치된 것으로 시스템 호출이 가능

  2. 시스템 호출 테이블 대상 변경
     - 공격자는 악성 코드로 선택된 합법적인 시스템 호출 루틴을 덮어씀
     - 시스템 호출 테이블은 변경되지 않음

  3. 시스템 호출 테이블 바꿈
     - 공격자는 전체 시스템 호출 테이블을 새로운 커널 메모리 위치에 새로운 테이블을 참조하도록 변경



(a) 평범한 커널 메모리 계층

(b) knark 설치 후

루트킷에 의한 시스템 호출 표

- 가상 기계와 기타 루트킷
  - 최근의 루트킷은 공격 대상의 운영체제에서 완전히 노출되지 않는 코드를 사용
  - 로그나 절충된 가상 기계 모니터 혹은 최근의 프로세서가 제공하는 하드웨어 가상화의 도움을 받는 하이퍼바이저를 사용하여 이룰 수 있음
  - 루트킷 코드는 대상 운영체제에서 커널 코드의 노출 아래에서 동작하고, 이것은 가상 기계에서 알려지지 않도록 동작하여 몰래 감시하고 공격할 수 있음
  - 방어: 모든 부트 프로세스는 반드시 보호되어야 함
  - 운영체제는 부트 프로세스를 안전하게 로드하고 악성코드의 설치를 못 하도록 하여야 함

5-9. 대비책

9. 대비책

- 악성 소프트웨어 대비 접근 방법
  - 악성 소프트웨어가 시스템에 설치되는 것을 허용하지 않거나 시스템의 수정 능력을 차단하는 것임
  - 이 목표는 감염 방지가 악성코드 공격에 대한 성공의 수를 크게 줄일 수 있을지라도 일반적으로 거의 달성하기 불가능함
  - 예방 정책, 인식, 취약성 완화, 위협 완화
  - 대비책
    1. 시스템에 악용될 수 있는 취약점의 수를 줄이기 위하여 모든 시스템이 현재의 상태에 있도록 확인
    2. 잠재적으로 감염 또는 손상된 파일의 수를 줄이기 위하여 시스템에 저장된 프로그램 및 데이터에 대한 적절한 접근 제어를 설정
    3. 적절한 사용자 인식 및 교육을 이용하여 대비

- 예방에 실패하면 기술적인 기법이 다음의 위협 완화를 지원하는 데 사용 가능
  1. 감지: 감염이 발생하면 감염 여부를 결정하고 악성코드의 위치를 알아냄
  2. 식별: 감지되면 시스템을 감염시킨 특정 악성코드를 식별함
  3. 제거: 특정 악성코드가 식별되면 더 확산되지 않도록 모든 감염된 시스템으로부터 악성 바이러스의 정 모든 흔적을 제거함
- 검출에 성공했지만 식별 또는 제거할 수 없는 경우 감염 또는 악성 파일을 삭제하고 감염되지 않은 백업 버전을 다시 복구
- 심각한 감염의 경우에는 모든 저장 공간을 깨끗하게 소각하고 깨끗한 미디어로부터 감염 시스템을 재구성

- 효과적인 악성코드 대비책에 대한 요구 사항
  1. 일반성
     - 접근 방식은 다양한 공격을 다룰 수 있어야 함
  2. 적시성
     - 접근 방식은 감염된 프로그램이나 시스템의 수를 제한하기 위하여 빠르게 대응
  3. 회복 탄력성
     - 접근 방식은 악성코드의 존재를 숨기기 위해 공격자에 의하여 사용된 기술에 대하여 대응할 수 있어야 함
  4. 최소 DOS 비용
     - 접근 방식은 대응 소프트웨어에 의해 용량이나 서비스에 대한 최소한의 감소를 초래하여야 하며 정상적인 작동을 현격히 낮추어서는 안됨
  5. 투명성
     - 대비 소프트웨어와 장치는 기존의 OS, 응용소프트웨어, 하드웨어의 수정을 요구하지 않아야 함
  6. 전역과 지역 범위
     - 접근 방식은 기업 네트워크의 외부와 내부 모두로부터 공격원을 다룰 수 있어야 함

- 호스트 기반 스캐너
  - 소프트웨어가 목표가 된 시스템과 상호 작용을 하는 악성 소프트웨어의 행동뿐만 아니라 악성 소프트웨어 활성을 위하여 정보에 대한 최대의 접근을 제공
  - 개인 컴퓨터에서 안티바이러스 소프트웨어의 사용은 현재 악성 소프트웨어의 양과 활성의 폭발적인 증가로 인해 널리 퍼져 있고, 이 소프트웨어는 호스트 기반 침입 탐지 시스템의 한 형태로 간주될 수 있음

- 안티바이러스 소프트웨어의 4세대로 구분하였음
  - 1세대: 간단한 스캐너
    - 악성 소프트웨어를 식별하기 위하여 악성 소프트웨어 시그니쳐가 필요
    - 시그니쳐는 "와일드카드"를 포함하지만 악성 소프트웨어의 모든 사본에서 필수적으로 동일한 구조 및 비트 패턴과 일치함
    - 이러한 시그니쳐 특정 스캐너는 알려진 악성 소프트웨어의 탐지에만 제한
  - 2세대: 경험적 스캐너
    - 특정 시그니쳐에 의존하지 않음
    - 가능한 악성 소프트웨어를 검색하기 위하여 경험적 규칙을 사용함
    - 스캐너는 악성 소프트웨어와 관련된 코드의 조각을 찾음
    - 키가 발견되면 스캐너는 악성 소프트웨어를 밝혀내기 위해 악성 소프트웨어를 복호화하여 감염 프로그램을 제거한 후 프로그램을 서비스에 반환
    - 체크섬(checksum)이 각 프로그램에 추가될 수 있음
    - 악성 소프트웨어가 체크섬의 변경 없이 프로그램이 대체되거나 변경된다면 무결성 검사가 이러한 변화를 잡아낼 수 있음

  - 프로그램이 변경될 때 체크섬을 바꿀 만큼 충분히 복잡한 악성 소프트웨어에 대응하기 위하여 암호화 해시 함수를 사용할 수 있음
  - 간단한 체크섬 대신에 해시 함수를 이용하여 악성 코드가 이전과 동일한 해시 코드를 생성하기 위해 프로그램을 조정하는 것을 방지할 수 있음
  - 신뢰할 수 있는 위치의 프로그램 보호 목록이 유지된다면 접근 방법으로 로그 코드나 프로그램을 대치하거나 설치하려는 시도를 탐지할 수 있음
- 3세대: 활성 트랩
  - 감염된 프로그램의 구조보다는 행동에 의해 악성 소프트웨어를 식별하는 메모리 상주 프로그램
  - 다양한 배열에 대한 시그니쳐 및 경험적 방법의 개발이 필요 없다는 장점이 있음
- 4세대: 완전한 기능을 갖춘 보호
  - 다양한 안티바이러스 기술로 이루어진 패키지
  - 시스템에 침투하는 악성코드의 능력을 제한하고 전파하기 위해 파일을 업데이트하는 악성 소프트웨어의 능력을 제한함

- 호스트 기반 스캐너 – 일반적 복호
  - 일반적 복호(Generic Decryption) 기술은 빠른 스캐닝 속도를 유지하면서 안티바이러스 프로그램이 복잡한 다형성 바이러스와 다른 형태의 컴퓨터 악성 프로그램을 쉽게 검출할 수 있게 함
  - 다형성 바이러스가 포함된 파일이 실행될 때 바이러스가 활성화되기 위하여 자신을 복호화하여야 함
  - 이런 구조를 검출하기 위하여 실행 파일은 다음 요소를 포함하는 일반적 복호 스캐너를 통하여 실행됨
    1. CPU 에뮬레이터
       - 소프트웨어 기반 가상 컴퓨터, 실행 파일의 명령어는 프로세서에 의해 실행되기 보다는 에뮬레이터에 의하여 해독됨
    2. 바이러스 시그니쳐 스캐너
       - 알려진 악성 시그니쳐를 찾는 목표 코드를 스캔하는 모듈임
    3. 에뮬레이션 제어 모듈
       - 목표 코드의 실행을 제어함

- 시뮬레이션의 시작에서 에뮬레이터는 목표 코드에 있는 명령어를 한 번에 하나씩 해독하기 시작함
- 코드가 복호화 루틴을 포함하여 악성 소프트웨어에 노출될 경우 그 코드는 해독됨
- 주기적으로 제어 모듈은 악성 소프트웨어 시그니쳐에 대한 목표 코드를 스캔 하기 위하여 해독을 정지함
- 해독하는 동안 목표 코드는 완전히 통제된 환경에서 해독되고 있기 때문에 실제로 개인 컴퓨터 환경에 손상을 일으킬 수 없음
- 일반적 복호 스캐너와 관련된 가장 어려운 설계 문제는 각 해독을 얼마나 오랫동안 실행시킬지 결정하는 것

- 호스트 기반 스캐너 –행동 차단 소프트웨어
  - 악의적인 행동에 대하여 실시간으로 호스트 컴퓨터의 운영체제를 통합하고 프로그램 동작을 감시함[CONE01, NACH02]
  - 시스템의 영향을 미칠 수 있는 기회를 가지기 전에 행동 차단 소프트웨어는 잠재적으로 악성 행동을 차단함
  - 감지 행동은 다음을 포함
    - 파일 열기, 보기, 지우기, 수정 등의 시도
    - 디스크 드라이브 및 다른 복구할 수 없는 디스크 작동의 초기화 시도
    - 실행 파일 또는 매크로의 로직 수정
    - 시작 설정과 같은 중요한 시스템 설정의 수정
    - 실행 내용을 보내기 위한 전자메일과 긴급한 메시징 클라이언트의 스크립팅
    - 네트워크 통신 야기
  - 행동 브로커는 의심되는 소프트웨어를 실시간으로 차단할 수 있기 때문에 확립된 안티바이러스 검출 기술에 비해 장점을 가짐

- 호스트 기반 스캐너 – 스파이웨어 감지 및 제거
  - 일반적인 안티바이러스 제품이 스파이웨어를 감지하는 것을 포함하여도 악성 소프트웨어 같은 위협은 스파이웨어에 특화된 감지 및 제거 유틸리티가 존재함
  - 이것은 스파이웨어의 감지와 제거에 특화되어 있고 더욱 견고한 능력을 제공

- 호스트 기반 스캐너 – 루트킷 대비책
  - 네트워크와 호스트 기반 IDS는 들어오는 트래픽에서 알려진 루트킷 공격의 코드 시그니쳐를 찾을 수 있음
  - 호스트 기반 안티바이러스 소프트웨어는 알려진 시그니쳐를 인식하기 위하여 사용될 수 있음
  - 새로운 시그니쳐를 갖는 또는 변형된 버전의 루트킷이 있을 경우, 시스템 호출의 가로채기나 키보드 드라이버와 연동하는 키로거와 같은 루트킷의 존재를 밝혀낼 수 있는 동작이 시스템에 필요함
  - 커널 레벨 루트킷이 감지된다면 이를 복구하기 위한 안전하고 신뢰성 있는 방법은 감염된 기계에 새로운 OS를 설치하는 것

- 주변 장치 스캔 접근 방법
  - 안티바이러스 소프트웨어가 사용되는 다음 위치는 조직의 방화벽과 IDS에 있음
  - 일반적으로 시스템상에서 실행되는 전자메일과 웹 프락시 서비스에 포함되고, 또한 IDS의 트래픽 분석 도구에도 포함
  - 이는 안티바이러스 소프트웨어가 악성 소프트웨어에 접근 하는 것을 허용하고, 이러한 소프트웨어는 의심스러운 트래픽의 흐름을 막는 침입 방지 대책이 포함
  - 이러한 접근 방식은 감염된 시스템에서 실행될 때는 접근 권한을 가지지 못하기 때문에, 악성 소프트웨어의 내용을 스캐닝하는 것을 제한

- 감시 소프트웨어의 사용
  1. 입구 모니터
     - 기업 네트워크와 인터넷 경계의 위치하며, 경계 라우터의 진입 여과 소프트웨어의 일부이거나 외부 방화벽 또는 분리된 수동 모니터의 소프트웨어일 수 있음
     - 모니터는 변칙 또는 시그니쳐를 사용할 수 있으며 악성 소프트웨어 트래픽을 검출하기 위해 경험적 방법을 사용

     예) 입력 트래픽이 사용되지 않는 지역 IP 주소를 사용하는지 검사하는 것
  2. 출구 모니터
     - 기업 네트워크와 인터넷 사이의 경계뿐만 아니라 기업 네트워크에서의 개인 LAN의 출구에 위치할 수 있음
     - LAN의 출구에 위치할 경우, LAN 라우터 또는 스위치의 출구 여과 소프트웨어의 부분이 됨
     - 나가는 트래픽을 감시하는 것에 의하여 악성 소프트웨어 공격 원을 잡아낼 수 있도록 설계됨
     - 웜에서 사용된 비정상적인 전자메일 웜이나 스팸 페이로드에서 사용되는 것처럼 비정상으로 높은 전자메일 트래픽을 감지하고 대응할 수 있음

- 분산 정보의 수집 접근 방법
  - 안티바이러스 소프트웨어가 사용되는 마지막 위치는 분산 구성에 있음
  - 많은 호스트 기반과 주변 센서로부터 데이터를 수집하고, 데이터의 연관성을 분석할 수 있도록 수집된 데이터를 중앙 분석 시스템에 전달함
  - 관련된 시스템 모두에게 악성 소프트웨어에 대응하고 방어할 수 있는 갱신된 시그니쳐와 행동 패턴을 되돌려줌
  - 이러한 더 많은 시스템이 제안되었고, 구체적인 예는 분산 침입 방지 시스템(IPS)임

참고문헌

- 제 3판 컴퓨터 보안 원리 및 실습
  William Stallings Lawrie Brown 저자, PEARSON, 2016
- Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, Jong Hyuk Park, "A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions", Journal of Supercomputing(SCI), 2016
- Jong Hyuk Park , Hyungjoo Kim, Jungho Kang, "Security Scheme Based on Parameter Hiding Technic for Mobile Communication in a Secure Cyber World", Symmetry(SCIE), 8(10), 106; pp.1-12, 2016
- Jungho Kang, Geunil Park, Jong Hyuk Park, "Design of secure authentication scheme between devices based on zero-knowledge proofs in home automation service environments", Journal of Supercomputing(SCI), Volume 72, Issue 11, pp 4319–4336, 2016
- Man-Sik Kim, Jeong-Kyu Lee, Jong Hyuk Park, Jung-Ho Kang, "Security Challenges in Recent Internet Threats and Enhanced Security Service Model for Future IT Environments", Jounal of Internet Technology(SCIE), Volume 17, Number 5, pp 947–955, 2016
- Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, Jong Hyuk Park
  " DFA-AD: a distributed framework architecture for the detection of advanced persistent threats",Cluster Computing, 20(1): 597-609 (2017)

Q & A

부 록

4. 전파 – 취약점 이용 – 웜

- 웜 프로그램은 각각의 새로운 시스템에 접근하는 클라이언트 또는 서버 소프트웨어 프로그램의 취약점을 이용함
- 시스템에서 시스템으로 감염시키기 위하여 네트워크 연결을 이용함
- USB 드라이브나 CD, DVD 데이터 디스크 등과 같이 공유 매체를 통해서 확산됨
- 이메일이나 메신저에 첨부된 문서에 포함된 매크로와 스크립트 코드를 이용하여 감염시킴
- 활성화될 때 웜은 복제하고 다시 전파할 수 있음
- 전파 뿐만 아니라 페이로드의 형태를 전달할 수 있음

- 웜 자체를 복제하기 위해 웜은 원격 시스템에 접근하는 방법을 사용함
  1. 전자메일 또는 즉석 메신저
     - 자신의 복제를 다른 시스템에게 이메일로 보내거나 즉석 메시지 서비스를 통해 첨부로 전송함
     - 코드는 이메일 또는 첨부 파일을 수신하거나 봄으로써 실행됨
  2. 파일 공유
     - 자신의 복사본을 만들거나 같은 USB 드라이브와 같은 이동식 미디어에 바이러스처럼 다른 적절한 파일을 감염시킴
     - 소프트웨어의 취약성을 이용하여 자동 실행 기법을 이용하여 다른 시스템에 연결될 때 실행됨
     - 사용자가 표적 시스템에서 감염된 파일을 열 때 실행됨

3. 원격 실행 기능
   - 기본적으로 제공되는 원격 실행 기능을 사용하거나 그 동작을 파괴하기 위해 네트워크 서비스 프로그램의 결함을 이용하여 다른 시스템상에서 자체의 복사본을 실행

4. 원격 파일 접근 또는 전송 기능
   - 하나의 시스템에서 다른 시스템으로 복사하기 위하여 원격 파일 접근 또는 전송 서비스를 사용함

5. 원격 로그인 기능
   - 사용자로서 원격 시스템에 로그인한 후 하나의 시스템에서 다른 시스템으로 복사하기 위하여 명령어를 사용함

- 다중 프로그래밍 시스템에서 시스템 프로세스 자체 명명 또는 시스템 오퍼레이터에 의해 발견될 수 없는 다른 이름을 사용하여 자신의 존재를 은폐할 수 있음

- 시스템상의 기존 프로세스에 자신의 코드를 삽입하고, 자신의 존재를 은폐하기 위하여 추가 쓰레드를 사용하여 실행할 수 있음

- 웜은 컴퓨터 바이러스와 유사하게 휴면, 전파, 트리거링 및 실행 단계로 수행됨

- 전파 단계에서는 다음 기능을 수행
  - 호스트 테이블, 주소록, 친구 목록, 신뢰할 수 있는 동료에 의하여 감염된 다른 시스템으로의 적잘한 접근 기법 탐색
  - 원격 시스템으로 복사본을 전송하기 위하여 발견된 접근 기법 사용

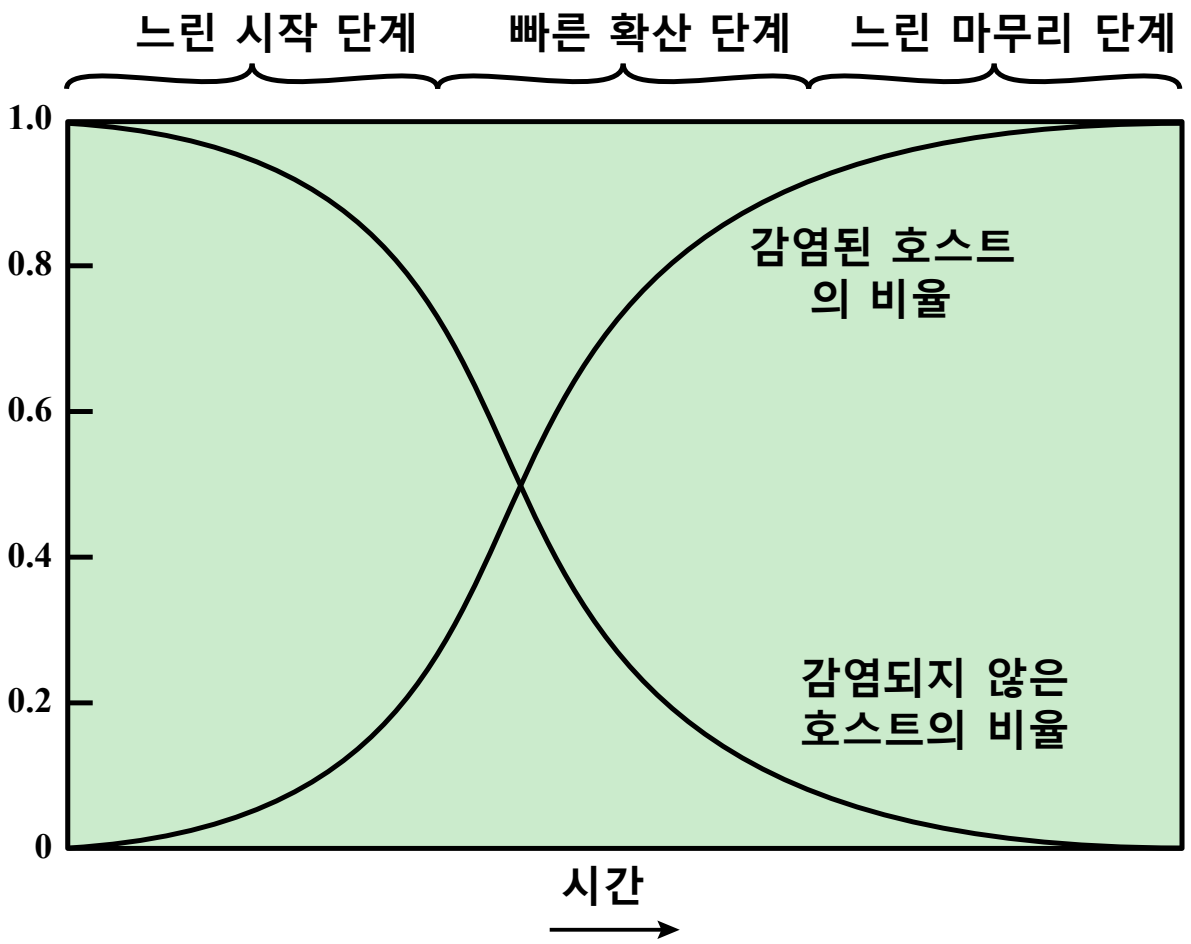
- 대상 탐색 및 선정
  - 웜을 감염시키기 위한 전파 과정에서 첫 번째 기능은 네트워크 스캐닝 또는 트래픽 경로의 역추적 과정을 통해서 감염시킬 다른 시스템을 찾는 것
  - 원격 액세스 네트워크 서비스에서 소프트웨어 취약점을 이용하는 웜의 경우, 취약한 서비스를 실행하는 잠재적인 시스템을 식별하여 감염시킴
  - 일반적으로 감염된 시스템의 대단위 분산 네트워크가 생성될 때까지 동일한 스캔 과정을 반복

- 웜이 사용하는 네트워크 어드레스 스캐닝 전략
  - 무작위
    - 각 호스트는 다른 시드 값을 사용하여 IP 주소 공간에서 임의의 주소를 조사
    - 무작위 기술은 많은 양의 인터넷 트래픽을 발생 시킴
  - 적중 목록
    - 공격자는 잠재적 취약 기계의 긴 목록을 컴파일함
    - 공격이 진행되는 중에 검출을 피하기 위하여 장기간에 걸쳐 느린 프로세스로 수행됨
    - 목록이 컴파일 되면 공격자는 목록에 있는 시스템을 감염시킴
  - 토폴로지컬
    - 스캔할 더 많은 호스트를 찾아 감염된 대상 컴퓨터에 포함된 정보를 사용
  - 로컬 서브넷
    - 호스트가 방화벽 안에서 감염된다면 호스트는 자신의 로컬 네트워크에서 대상을 찾음
    - 호스트는 방화벽에 의하여 보호된 다른 호스트를 발견하기 위하여 서브넷 주소 체계를 이용

- 웜 전파 모델
  - 컴퓨터 바이러스와 웜은 생물학적 바이러스와 유사한 자기 복제와 전파 동작을 나타냄
  - 간단한 고전적인 전염 모델은 다음과 같이 표현될 수 있음

- 전파는 세 단계를 통해서 진행

1. 초기 단계
   - 스캐닝 작업으로 인해 호스트의 수는 기하급수적으로 증가함
   - 단일 호스트에서 실행하고 두 개의 주위 호스트를 감염
   - 호스트들이 각각 두 개 이상의 호스트를 감염시키고, 이와 같은 과정이 반복
2. 중간 단계
   - 감염대상의 수와 확산 속도가 기하급수적으로 증가함
3. 마무리 단계
   - 웜이 식별하기 어려운 남아 있는 호스트들을 찾아내기 위해 다양한 스캐닝 기법을 사용하기 때문에 일시적으로 확산 속도가 느려짐


느린 시작 단계
빠른 확산 단계
느린 마무리 단계
1.0
0.8
0.6
0.4
0.2
0
감염된 호스트
의 비율
감염되지 않은
호스트의 비율
시간

웜 전파 모델

- 모리스 웜
  - 유닉스 시스템에 퍼지도록 설계되었고 많은 수의 다른 전파 기술들을 사용하였음
  - 실행되었을 때 첫 번째 작업은 다른 대상에게 접근 가능하도록 알려진 다른 호스트를 찾는 것임
  - 다른 기계가 선언된 시스템 테이블이 포함된 테이블, 사용자의 메일 전달 파일, 원격 계정의 접속에 대해 사용자가 권한을 준 테이블 그리고 네트워크 연결 상태에 대한 프로그램 상태를 검사하여 작업을 수행

- 각각의 발견된 호스트의 경우 웜은 접근 권한을 얻기 위한 많은 방법들을 시도함
  1. 합법적인 사용자로 원격 호스트에 접속을 시도함.
  2. 이 방법에서 웜은 먼저 로컬 암호 파일을 해독하려고 발견된 암호 및 해당 사용자의 ID를 사용함
  3. 암호를 얻기 위해서 웜은 암호 해독 프로그램을 실행하고 아래의 목록을 참고함
     a. 각각의 사용자 계정 이름과 이것을 조합한 간단한 순열
     b. 모리스가 후보라고 생각한 내장된 암호 432의 목록
     c. 로컬 시스템 사전의 모든 단어
  4. 원격 사용자의 소재를 보고하는 유닉스 프로토콜의 버그를 악용함
  5. 수신 메일을 보내는 원격 프로세스의 디버그 옵션에 트랩 도어를 악용함

- 웜 형태의 발전 과정

이름	연도	특징
멜리사	1998년	• 이메일 형태 웜 • 첫 번째 하나의 패키지에 트로이 목마, 바이러스, 웜의 측면을 포함한 악성코드
Code Red	2001년 7월	• 침투하여 확산하기 위해 Internet Information Server(IIS) 보안의 허점을 이용 • 다른 호스트에 전파하기 위해 임의의 IP 주소를 관찰함 • 특정 시간 동안 퍼뜨리기만 하다가, 정부의 웹사이트에 서비스 거부 공격(DoS)를 시작함
Code Red II	2001년 8월	• Microsoft의 IIS를 목표로 하였음 • 해커가 원격으로 피해자의 컴퓨터에 명령 실행을 허용하도록 백도어를 설치함
Nimda	2001년 9월	• 웜, 바이러스, 모바일 코드의 특징을 가짐 • 전자메일, 윈도우 공유, 웹 서버, 웹 클라이언트, 백도어의 다양한 분산 방법을 이용하여 전파
SQL Slammer	2003년 초	• 마이크로소프트 SQL 서버의 버퍼 오버플로우 취약점을 이용하였음 • 치밀하고 빠르게 전파되며, 10분 이내에 취약한 호스트의 90%를 감염시킴
Sobig.F	2003년 후반	• 감염된 기계들을 스팸 엔진으로 바꾸기 위하여 오픈 프락시 서버를 이용함 • 매 17개의 메시지들 중 하나를 이용하여 24시간 이내에 100만 개 이상의 복사본을 생성

이름	연도	특징
Mydoom	2004년	• 대량 발송이 가능한 이메일 형태 웜 • 비밀번호나 신용카드 번호 등과 같은 데이터에 해커들이 원격으로 접근이 가능하게 함으로써 감염된 컴퓨터에 백도어를 설치하는 경향이 증가함 • 분당 1000번 이상 복제하고, 36시간 내에 10억 개의 감염된 메시지를 전파
Warezov	2006년	• 실행되면 시스템 디렉터리에 실행 파일을 생성하고 레지스트리 항목을 생성하여 윈도우가 시작될 때 매번 자기 자신을 실행시키도록 설정 • 이메일 주소를 위한 몇몇 유형의 파일을 스캔하고 첨부 파일로 자신을 전송
Conficker (Downadup)	2008년 11월	• 초기에 윈도우 버퍼 오버플로우의 취약점을 이용하여 퍼짐 • 이 후 USB 드라이브와 네트워크 파일 공유를 통해 확산되었음
Stuxnet	2010년	• 탐지의 기회를 줄이기 위하여 속도를 제한하였음 • 장비의 작동을 방해할 목적으로 이란의 핵 프로그램 같은 것과 연관된 산업 제어 시스템을 대상으로 하였음 • USB 드라이버, 네트워크 파일 공유, 제로 데이 취약점의 이용 등 다양한 전파 기법을 지원

- 웜 기술의 상태
  - 다중 플랫폼
    - 윈도우에 제한되지 않고 UNIX와 같은 다양한 플랫폼을 공격함
    - 문서에서 매크로나 스크립팅 언어를 이용
  - 다중 활용
    - 웹 서버, 브라우저, 전자메일, 파일 공유, 네트워크 기반 응용 등 다양한 방법으로 시스템에 침투함
  - 초고속 확산
    - 짧은 시간에 가능한 많은 취약한 기계에 웜의 확산을 최적화하기 위해 다양한 기술을 활용
  - 다형성
    - 탐지를 피하고 필터를 통과하고 실시간 분석을 헛되게 하기 위하여 웜은 바이러스 다형성과 같은 기법 이용
    - 웜 각각의 복제는 기능적으로 같은 명령어와 암호 기술을 이용하여 새로운 코드를 발생시킴

- 변성
  - 모양의 변경과 더불어 변성 웜은 다른 전파 단계에서 밝혀지지 않는 다양한 동작 형식의 목록을 가짐

- 확산 기법
  - 웜은 많은 수의 시스템을 빠르게 손상시키기 때문에 DDOS, 루트 킷, 스팸 전자메일 발생기, 스파이웨어 등과 같이 다양한 악성코드를 전파하기 위한 방법이 있음

- 제로 데이 악용
  - 최대한 충격과 분산을 성취하기 위하여 웜이 보급될 때 네트워크 공동체에 의해서만 발견된 알려지지 않은 취약점을 이용

- 모바일 폰 웜
  - 블루투스 무선 연결 또는 MMS를 통해 통신함
  - 대상은 사용자가 셀룰러 네트워크 운영자가 아닌 다른 출처에서 소프트웨어 응용프로그램의 설치를 허용하는 스마트폰임
  - 완전히 사용하지 못하도록 폰의 데이터를 지우거나, 프리미엄 가격의 번호로 비용이 많이 드는 메시지를 강제로 폰으로 보냄
  - 컴 워리어 웜은 블루투스에 의해 수신 영역에서 다른 휴대전화로 복제됨
  - 이동식 메모리 카드에 자신을 복사할 뿐만 아니라 전화 프로그램 설치 파일에 자신을 삽입함

- 클라이언트 측 취약점 및 다운로드에 의한 구동
  - 다운로드에 의한 구동(drive-by-download)
    - 사용자가 공격자에 의해 제어되는 Web 페이지를 볼 때 사용자에게 알림이나 동의 없이 시스템에 악성코드를 다운로드하고 설치하기 위하여 브라우저의 버그를 이용하는 것이 공통적인 기술
    - 대부분의 경우 이 악성코드는 웜처럼 활발하게 전파되지는 않음
    - 악성코드를 확산시키기 위하여 의심하지 않는 사용자가 자신의 시스템에 방문하기를 기다림

- 클릭 재킹
  - 사용자 인터페이스의 구제 공격으로 알려짐
  - 감염된 사용자의 클릭을 수집하기 위해 공격자에 의하여 사용
  - 공격자는 악성코드를 가지고 있을 수 있는 웹사이트에 사용자가 방문하도록 하기 위해 사용자 컴퓨터 설정을 조정
  - 의도적으로 최상위 페이지를 클릭할 때 다른 페이지에서 버튼을 클릭하거나 링크되도록 속이기 위하여 여러 투명 또는 불투명 층을 이용함
  - 유사한 기술을 사용하여 키 스트로크 또한 가로채기를 할 수 있음
  - 스타일 시트, 인라인 프레임 및 텍스트 상자의 교묘한 조합을 통해 사용자는 자신의 전자메일이나 은행 계좌의 비밀번호를 입력하도록 하지만 공격자에 의해 보이지 않는 프레임에 정보가 입력됨