10장. 악성 소프트웨어

10.2 바이러스

바이러스 속성 (1)

- 다른 프로그램을 변형시켜 '감염(infect)'시키는 프로그램
- 변형된 형태의 바이러스
  - 원래 프로그램에 루틴을 주입해서 바이러스 프로그램의 복제를 제작하는 것이 있음
- 1980년대 초반 등장
  - 프레드 코헨 : 바이러스란 용어 처음 사용

바이러스 속성 (2)

- 컴퓨터 바이러스 3개 부분
  - 감염 메커니즘(Infection mechanism):
    - 바이러스가 퍼지는 수단
    - 자신을 복제
    - 감염 벡터(infection vector)라고도 함
  - 트리거(Trigger):
    - 페이로드 활성화나 전달 시기를 정하는 사건이나 조건
  - 페이로드(Payload):
    - 바이러스가 자기 자신을 퍼뜨리는 일 외에 하는 일
    - 페이로드는 피해를 줄 수 있고 심각한 피해를 끼치지 않을 수도 있지만 분명히 그 활동을 알 수 있음

바이러스 속성 (3)

- 바이러스 활동 4단계
  - 잠복단계(Dormant phase):
    - 어떤 조건을 만족하기를 기다리는 단계
    - 모든 바이러스가 이 단계를 갖는 것은 아님
  - 확산단계(Propagation phase):
    - 자신과 동일한 복제를 다른 프로그램이나 시스템의 특정 영역에 투입
  - 트리거단계(Triggering phase):
    - 바이러스가 작동되어 의도한 기능 수행 시작
  - 실행단계(Execution phase):

바이러스 구조 (1)

- 간단한 바이러스의 예
  - 감염여부 판정 및 복제
  - 유해 행동
  - 원래 프로그램 동작
- 원래 프로그램보다 길어지므로 쉽게 탐지 가능

```
     program V :=

{goto main;
     1234567;

     subroutine infect-executable :=
          {loop:
          file := get-random-executable-file;
          if (first-line-of-file = 1234567)
             then goto loop
             else prepend V to file; }

     subroutine do-damage :=
          {whatever damage is to be done}

     subroutine trigger-pulled :=
          {return damages is to be done}

main:    main-program :=
          {infect-executable;
          if trigger-pulled then do-damage;
          goto next;}
next:

}
```

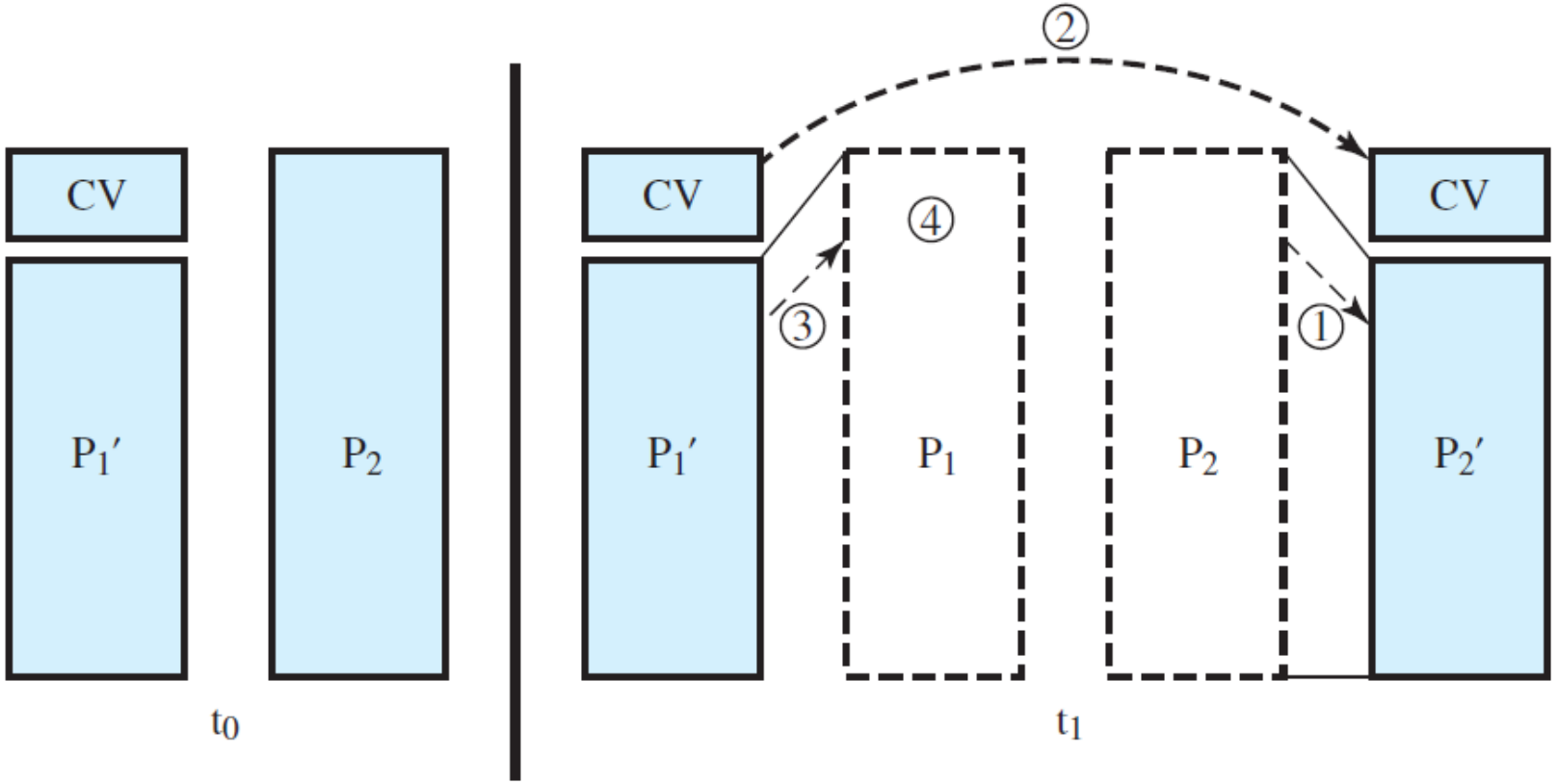
바이러스 구조 (2)

- 압축 바이러스 논리
  - 실행 파일을 압축하여 감염 안된 것과 동일한 길이로 만드는 것

```
      program CV :=

{goto main;
      01234567;

      subroutine infect-executable :=
            {loop:
                  file := get-random-executable-file;
             if (first-line-of-file = 1234567) then goto loop;
          (1)    compress file;
          (2)    prepend CV to file;
          }

main:     main-program :=
             {if ask-permission then infect-executable;
          (1)    uncompress rest-of-file;
          (2)    run uncompressed file;}
          }
```

바이러스 구조 (3)

- 압축 바이러스 활동 절차



바이러스 구조 (4)

- 압축바이러스 활동 절차 (계속)
  - 프로그램 P1이 바이러스 CV에 의해 감염되었다고 가정
  - 이 프로그램이 실행요청을 받으면 제어는 바이러스에게로 넘어가서 다음 단계 수행
    - 1. 발견된 감염이 안 된 파일 P2에 대해, 바이러스는 우선 그 파일을 압축하여 P'2를 생성
      - 압축파일은 원래 프로그램보다 바이러스 크기만큼 작게 제작
    - 2. 바이러스를 압축된 프로그램의 앞에 추가
    - 3. 원래의 감염된 프로그램의 압축버전인 P'1의 압축을 풀어냄
    - 4. 압축이 풀린 원래 프로그램 실행

초기 감염

- 초기 예방이 중요
- 바이러스는 시스템의 외부에 있는 어떤 프로그램의 일부일 수 있기 때문에 예방이 어려움
- 자신의 시스템과 모든 응용을 직접 만들지 않는 한 취약할 수 밖에 없음
- 정상적인 사용자가 시스템 상의 프로그램을 수정할 권리를 박탈하여 예방
- 초기 PC : 접근통제수단을 갖추고 있지 않아 빠르게 확산
- UNIX : 시스템 상의 접근통제수단이 있어 쉽게 차단
- 현대 : 보다 효율적인 접근통제수단을 갖추고 있어 전통적 머신코드는 덜 사용

바이러스 유형 (1)

- 목표별 바이러스 유형(classification by target)
  - 부트 섹터 감염자(Boot sector infector):
    - 마스터 부트 레코드나 부트 레코드를 감염시키고 바이러스가 포함된 디스크로부터 시스템이 부팅될 때 퍼짐
  - 파일 감염자(File infector):
    - 운영체계나 쉘이 실행 가능하다고 여기는 파일을 감염
  - 매크로 바이러스(Macro virus):
    - 응용 프로그램으로 나타낼 수 있는 매크로 코드를 가진 파일을 감염

바이러스 유형 (2)

- 은닉 전략에 따른 바이러스 유형
  - 암호화된 바이러스(Encrypted virus):
    - 랜덤 암호화키 생성, 남은 부분 암호화
    - 감염된 프로그램이 호출될 때 바이러스 복호화
    - 복제될 때 새로운 암호화키 선택
    - 다른 키로 암호화되므로 비트 패턴이 일정하지 않음
  - 스텔스 바이러스(Stealth virus):
    - 안티바이러스 소프트웨어에 의해 탐지되지 않기 위해 자신을 감추도록 정교하게 설계된 바이러스
  - 폴리모픽 바이러스(Polymorphic virus):
    - 감염시킬 때마다 변형하기 때문에 특징(signature) 이용 탐지 불가
  - 메타모픽 바이러스(Metamorphic virus):
    - 감염시킬 때마다 변형, 모양만 변형하는 것이 아니라 행동까지 변화

스텔스 바이러스(stealth virus)

- 스텔스(Stealth)
  - 탐지를 피하기 위해 바이러스가 이용하는 기술
- 스텔스 기술 이용 예
  - 압축을 이용해서 감염되지 않은 프로그램의 길이와 감염된 프로그램의 길이가 동일하도록 제작
  - 디스크 I/O 루틴 안에 가로채기 논리(intercept logic)를 심어놓고 이 루틴을 이용하는 디스크의 의심스런 부분을 읽으려는 시도가 있으면 원래의 감염되지 않은 프로그램을 되가져다 놓음

폴리모픽 바이러스 (1)

- polymorphic virus
- 복제 과정에서 기능적으로는 동일하지만 비트패턴에서는 명확하게 다른 변형을 만들어냄
- 변형 목적은 안티 바이러스 프로그램을 무력화시키는 것
- 바이러스의 '특징'은 복제 될 때마다 달라짐
- 사용 방법
  - 여분의 명령을 랜덤하게 삽입
  - 독립적인 명령 순서를 바꿈
  - 암호화 이용

폴리모픽 바이러스 (2)

- 암호화 바이러스의 전략
  - 보통 변형엔진(mutation engine)이라고 하는 바이러스의 일부가 랜덤한 암호화키를 생성하여 바이러스의 나머지 부분을 암호화
  - 변형엔진 : 키를 생성하고 암호화/복호화를 담당
  - 변형엔진 자신은 사용될 때마다 변화

바이러스 키트

- 바이러스-생성 툴 키트
- 초보 제작자도 여러 가지 바이러스를 짧은 시간 안에 제작 가능
- 툴 키트를 이용해서 만든 바이러스는 새로 설계해서 만든 바이러스보다 덜 정교
- 생성될 수 있는 새 바이러스의 엄청난 수 때문에 안티바이러스 구조에 영향을 줌

매크로 바이러스 (1)

- 매크로바이러스의 위협성
  1. 플랫폼과 무관하게 작동
     - 거의 모든 매크로 바이러스 는 마이크로소프트 워드 문서를 감염
     - 워드를 지원하는 어떤 하드웨어 플랫폼이나 운영체제도 감염 대상
  2. 문서만 감염시키고 코드 실행부분은 감염시키지 않음
     - 컴퓨터 시스템에 저장되는 정보는 프로그램이라기보다는 문서의 형태
  3. 쉽게 확산
     - 가장 보편적인 방법은 전자메일
  4. 시스템 프로그램보다는 사용자 문서를 감염
     - 파일 시스템 접근 통제를 제한적으로 사용해서 확산 방지

매크로 바이러스 (2)

- 워드나 엑셀과 같은 오피스 응용프로그램에서 사용하는 매크로 기능 사용
  - 문서에 내장된 실행 프로그램
  - 일반적으로 반복적인 작업을 자동화
  - 기본적 프로그래밍 언어의 형태
  - 일련의 키 입력을 하나의 매크로로 정의하고 설정할 수 있음
- MS의 버전 업데이트로 바이러스에 대한 방어 능력 증대
  - 옵션으로 의심스러운 워드파일을 감지하고, 매크로를 포함하고 있는 문서를 열 때마다 위험성 경고
- 안티바이러스 제조업자
  - 다양한 매크로 바이러스 탐지 및 교정

전자메일 바이러스 (1)

- 예 1: 멜리사(Melissa)
  - 첨부된 파일에 내장된 MS 워드 매크로를 이용
  - 수신자가 전자메일에 첨부된 파일을 열면 워드 매크로가 동작
    1. 전자메일 바이러스는 사용자의 전자메일 패키지 안에 있는 메일링 리스트의 모든 사람에게 자신을 전송
    2. 바이러스는 사용자 시스템에 지역적 피해

전자메일 바이러스 (2)

- 예 2: VB 스크립팅 언어 사용
  - 바이러스를 포함하고 있는 전자우편을 열어보기만 해도 작동
  - 전자메일 패키지에서 제공하는 VB 스크립팅 언어 사용
  - 감염된 호스트가 알고 있는 모든 전자우편 주소로 전파 > 수 시간 내 확산 가능
- 전자메일을 통해 도착하고 인터넷을 통해 자신을 복제하기 위해 전자메일 소프트웨어 기능을 사용하는 시대
- PC 응용소프트웨어와 인터넷 유틸리티가 높은 레벨의 보안 체계를 갖추어야 함