사이버보안개론

Introduction to Cybersecurity

유준수 (Joon Soo Yoo, Ph.D.)

Fall 2026

Defense & Advanced Security Lab (DAS Lab)
Department of Advanced Defense Technology and Industry
Jeonbuk National University

Chapter 6. Malicious Software

6.1 Types of Malicious Software

Malware, Propagation, and Payloads

Learning Objectives

이번 Section에서는 Malware의 기본 개념과 실제 공격에서 Malware가 사용되는 방식을 학습한다.

Learning Goals

- Malware, Vulnerability, Exploit의 차이를 이해한다.
- Propagation과 Payload를 구분할 수 있다.
- Virus, Worm, Trojan의 동작 원리를 설명할 수 있다.
- Spyware, Backdoor, Rootkit, Bot의 역할을 이해한다.
- Ransomware와 Wiper의 공격 목적을 구분한다.
- Blended Malware와 Attack Kit의 개념을 이해한다.
- 실제 Malware 사례를 Security 관점에서 분석한다.

1. What Is Malware?

Malware는 **Malicious Software**의 줄임말이며, 일반적으로 악성코드라고 부른다.

Malware는 시스템에 침투하거나 실행되어 공격자가 의도한 악의적인 기능을 수행한다.

대표적인 악성 행위는 다음과 같다.

- Sensitive Information을 수집하거나 외부로 전송
- File 또는 System Configuration을 변경하거나 파괴
- Unauthorized Access를 위한 Backdoor 설치
- 시스템 자원을 이용하여 다른 시스템 공격
- 악성 행위와 자신의 존재를 숨김

핵심

Malware는 단순히 시스템에 오류를 발생시키는 프로그램이 아니라, 악의적인 기능을 수행하는 Software이다.

1-1. Malware vs. Vulnerability vs. Exploit

세 가지 개념은 서로 관련되어 있지만 의미가 다르다.

Basic Security Concepts

Concept	Meaning
Vulnerability	시스템의 Design, Implementation 등에 존재하는 Weakness
Exploit	Vulnerability를 악용하는 방법 또는 Code
Malware	악의적인 기능을 수행하는 Software

예를 들어 공격자는 Server의 Vulnerability를 이용하여 시스템에 침입한 뒤 Malware를 설치할 수 있다.

$$\text{Vulnerability} \xrightarrow{\text{Exploit}} \text{Compromise} \xrightarrow{\text{Installation}} \text{Malware}$$

핵심

Exploit은 취약점을 악용하는 수단이고, Malware는 악의적인 기능을 수행하는 Software이다. Malware가 반드시 Exploit을 통해 설치되는 것은 아니다.

1-2. How Should We Analyze Malware?

Malware를 분석할 때는 단순히 이름을 분류하기보다 다음 세 가지 질문을 생각하는 것이 중요하다.

1. **Propagation**
   Malware가 어떻게 Target System에 도달하고 확산되는가?
2. **Payload**
   Malware가 실행된 후 어떤 악의적인 기능을 수행하는가?
3. **Objective**
   공격자가 궁극적으로 달성하려는 목표는 무엇인가?

예를 들어 Ransomware에서는

$$\text{Phishing} \longrightarrow \text{File Encryption} \longrightarrow \text{Financial Gain}$$

와 같은 관계를 생각할 수 있다.

핵심

Propagation은 감염 및 확산 방식, Payload는 악성 기능, Objective는 공격자의 최종 목표를 의미한다.

1-3. What Does Payload Mean?

Payload는 분야에 따라 서로 다른 의미로 사용된다.

Different Meanings of Payload

Context	Meaning
Network	Packet에서 Header 등을 제외한 실제 Data
Exploit	공격 과정에서 전달하거나 실행하는 Data 또는 Code
Malware	감염 후 수행하는 악의적인 기능 또는 행위

예를 들어 Ransomware를 생각해보자.

Propagation: Phishing

⇓

Payload: File Encryption

⇓

Objective: Ransom Payment

핵심

Malware에서 Payload는 공격자의 최종 목적 자체라기보다 그 목적을 위해 실행하는 악의적인 기능을 의미한다.

2. Propagation Mechanisms

Malware가 Target System에 도달하는 방식은 다양하다.

교재에서는 대표적으로 세 가지를 설명한다.

1. **File Infection**
   기존 Executable 또는 Document에 악성코드를 삽입한다.
2. **Vulnerability Exploitation**
   Software Vulnerability를 악용하여 시스템에 침투하거나 다른 시스템으로 확산한다.
3. **Social Engineering**
   사용자를 속여 Malware를 다운로드하거나 실행하도록 유도한다.

핵심

Propagation Mechanism은 Malware가 어떻게 Target에 도달하고 감염을 확산시키는지 설명한다.

2-1. File Infection

File Infection은 기존의 정상적인 File에 악성코드를 삽입하는 방식이다.

예를 들어 정상적인 Executable Program P가 있다고 하자.

공격자가 Virus Code V를 삽입하면, 감염된 Program P'를 만들 수 있다.

$$P \longrightarrow P' = P + V$$

이는 File Infection의 개념을 나타내는 단순화된 표현이다.

사용자가 감염된 Program을 실행하면

- 원래 Program의 기능이 실행될 수 있고,
- 함께 삽입된 Virus Code도 실행될 수 있다.

핵심

File Infection에서는 정상적인 File이 Malware를 전달하고 실행하는 Host 역할을 할 수 있다.

2-2. Vulnerability-Based Propagation

Malware는 Software Vulnerability를 이용하여 다른 시스템에 침투할 수 있다.

예를 들어 Network에 세 개의 Computer가 있다고 하자.

$$A \longleftrightarrow B \longleftrightarrow C$$

Computer A가 Malware에 감염되어 있고, B와 C에 동일한 Vulnerability가 존재한다고 하자.

A: Infected

⇓

A exploits B's vulnerability

⇓

B: Infected

⇓

B exploits C's vulnerability

⇓

C: Infected

핵심

Vulnerability-Based Propagation은 취약한 시스템을 찾아 악용하는 방식으로 Malware가 확산될 수 있음을 보여준다.

2-3. Social Engineering

모든 Malware가 Software Vulnerability를 악용해야만 설치되는 것은 아니다.

공격자는 사용자의 판단이나 행동을 악용할 수 있다.

예를 들어 공격자가 다음과 같은 메시지를 보낸다고 하자.

"Important Security Update Available"

Download Update

사용자가 정상적인 업데이트라고 믿고 악성 프로그램을 실행할 수 있다.

대표적인 방식은 다음과 같다.

- Phishing Email
- Fake Software Update
- Malicious Installer
- Fake Document or Attachment

핵심

Social Engineering은 Software의 Weakness가 아니라 사용자의 신뢰와 행동을 공격 경로로 활용한다.

3. What Is a Computer Virus?

Virus는 다른 Executable 또는 Script Code에 자신을 복제하여 삽입하는 Malware이다.

일반적인 Virus의 특징은 다음과 같다.

- 기존 File이나 Program을 감염시킨다.
- 감염된 Host가 실행되면 Virus Code도 실행될 수 있다.
- 다른 File을 감염시켜 자신의 복제본을 만든다.
- 추가적인 악성 Payload를 수행할 수도 있다.

중요한 점은 Virus가 **Host Program에 의존하는 형태의 Malware**라는 것이다.

핵심

Virus의 대표적인 특징은 기존 File을 감염시키고 그 File을 통해 자신의 Code를 복제한다는 점이다.

3-1. How Does a Virus Infect a File?

정상적인 Program A와 Program B가 있다고 하자.

처음에는 Program A만 Virus에 감염되어 있다.

$$A' = A + V$$

사용자가 감염된 Program A를 실행하면, Virus Code V가 함께 실행된다.

Virus는 다른 Program B를 찾아 자신의 Code를 삽입할 수 있다.

$$B \longrightarrow B' = B + V$$

이제 Program B도 감염된 상태가 된다.

이 과정이 반복되면 여러 File에 Virus Code가 복제될 수 있다.

핵심

Virus의 Replication은 기존 Host File을 감염시키는 방식으로 이루어진다.

3-2. Virus Replication and Execution

Virus의 Replication과 Payload는 구분해야 한다.

1. Replication

- 감염 가능한 File을 찾는다.
- 자신의 Code를 해당 File에 삽입한다.
- 감염된 File이 다른 환경으로 전달될 수 있다.

2. Payload Execution

- File을 삭제하거나 변경한다.
- 사용자 정보를 수집한다.
- 시스템의 정상 동작을 방해한다.

Virus가 반드시 파괴적인 Payload를 수행해야 하는 것은 아니다.

핵심

Replication은 Virus가 자신을 복제하는 과정이고, Payload는 감염 후 수행하는 악의적인 기능이다.

3-3. (참고) Macro Virus

Macro Virus는 Document에 포함된 Macro 또는 Script를 이용하는 Virus이다.

예를 들어 Office Document에 악성 Macro Code가 포함되어 있다고 하자.

사용자가 Macro 실행을 허용하면 악성코드가 실행될 수 있다.

Document Opened

⇓

Macro Execution Allowed

⇓

Malicious Code Execution

최근에는 Office의 인터넷 출처 Macro 차단 등 보안 정책이 강화되었다.
그러나 공격자는 다른 Script, Attachment, Social Engineering 방식도 활용할 수 있다.

핵심

Macro Virus는 Document가 Malware의 실행 경로가 될 수 있음을 보여주는 사례이다.

3-4. (참고) Macro Virus: Execution and Payload

Word Document에는 VBA Macro Code가 포함될 수 있다.

다음은 Document를 열었을 때 특정 File을 삭제하는 교육용 예시이다.

Example: VBA Macro

```
1
2  Private Sub Document_Open()
3
4      Dim filePath As String
5      filePath = Environ("TEMP") & "\macro_test.txt"
6
7      If Dir(filePath) <> "" Then
8          Kill filePath
9      End If
10
11 End Sub
```

- *Document_Open()*: 문서 열기 Event Handler
- *Dir()*: File 존재 여부 확인
- *Kill*: 지정한 File 삭제

핵심

Macro 실행이 허용되면 Document에 포함된 Code가 사용자 권한으로 실행되어 File System에 영향을 줄 수 있다.

3-5. (참고) Macro Virus: Self-Replication

앞의 Macro는 File을 삭제하지만, 자기 자신을 복제하지는 않는다.

Virus가 되려면 자신의 Code를 다른 Host File에 복제하는 기능이 필요하다.

예를 들어 감염된 Word Document D_1에 Virus Code V가 포함되어 있다고 하자.

$$D_1' = D_1 + V$$

Virus는 다른 Document D_2에도 자신의 Code를 삽입하려고 시도한다.

$$D_2 \longrightarrow D_2' = D_2 + V$$

이때 D_2'가 실행되면 동일한 Virus Code가 다시 동작할 수 있다.

핵심

Virus의 핵심 특징은 Malicious Payload 자체가 아니라 다른 Host에 자신의 Code를 복제하는 능력이다.

3-6. (참고) Macro Virus: Combined Behavior

Macro Virus는 Self-Replication과 Malicious Payload를 하나의 Program에 결합할 수 있다.

다음은 동작 원리를 나타내는 Pseudocode이다.

Conceptual Macro Virus

```
ON Document_Open():

    target = TEMP + "\macro_test.txt"

    IF target exists:
        DELETE target

    FOR each document in TestDocuments:
        IF document is not infected:
            COPY virus code into document
```

Payload

- 특정 File을 삭제한다.

Self-Replication

- 다른 Document에 자신의 Code를 복제한다.

핵심

Macro Virus는 Document를 실행 경로로 사용하며, Payload 실행과 Self-Replication을 함께 수행할 수 있다.

3-7. (참고) Macro Virus: Infection and Defense

앞의 예시를 실제 공격 흐름으로 생각해보자.

Infected Word Document (D'_1)

⇓

User Opens Document

⇓

Macro Execution Permitted

⇓

Payload | **Replication**

File Deletion | Infect Other Documents

⇓

Additional Infected Documents ($D'_2, D'_3, \ldots$)

Modern Defense

- Internet-origin VBA Macro Blocking
- Protected View / Macro Security Policies
- Endpoint Security Monitoring
- User Awareness and Attachment Verification

핵심

Macro Virus는 Document의 자동화 기능을 악용한다. 현대 Office에서는 Macro 실행 제한이 강화되었지만, 신뢰하지 않는 Document의 Code 실행은 여전히 주의해야 한다.

4. What Is a Worm?

Worm은 독립적으로 실행될 수 있으며, 다른 시스템으로 자신의 복제본을 확산시킬 수 있는 Malware이다.

Worm의 대표적인 특징은 다음과 같다.

- 독립적인 Program으로 실행될 수 있다.
- Network를 통해 다른 Host로 전파될 수 있다.
- Software Vulnerability를 악용하는 경우가 많다.
- 자동으로 여러 시스템을 감염시킬 수 있다.

Virus와 달리 기존 Executable File을 감염시켜야만 복제되는 것은 아니다.

핵심

Worm의 대표적인 특징은 독립적인 실행과 자기 복제를 통한 Network Propagation이다.

4-1. How Does a Worm Spread?

Worm은 감염된 시스템에서 다른 취약한 시스템을 찾아 확산할 수 있다.

일반적인 과정을 생각해보자.

1. **Find Target**
   Network에서 접근 가능한 다른 Host를 찾는다.
2. **Exploit Vulnerability**
   Target의 Software Vulnerability를 악용한다.
3. **Replicate**
   Target에 자신의 복제본을 전달하거나 실행한다.
4. **Repeat**
   새로 감염된 Host에서도 동일한 과정이 반복된다.

핵심

Worm은 감염된 Host가 다시 새로운 Propagation Source가 될 수 있기 때문에 빠른 확산이 가능하다.

4-2. Worm Propagation Example

다음과 같이 네 개의 Computer가 있다고 하자.

$$A,\ B,\ C,\ D$$

Computer A는 이미 Worm에 감염되었으며, 나머지 Computer는 동일한 Network에 연결되어 있다.

Step 1. Initial Infection

A (Infected) B, C, D (Not Infected)

Step 2. Vulnerability Exploitation

Worm은 Network를 통해 다른 Computer에 접근하고, 취약점이 존재하는 Computer B를 감염시킨다.

$$A \longrightarrow B$$

Step 3. Self-Propagation

감염된 B도 Worm을 실행하여 다른 Computer로 확산을 시도할 수 있다.

$$A \longrightarrow B \longrightarrow C$$

핵심

Worm은 감염된 Computer가 새로운 전파 주체가 될 수 있으므로, Network 전체로 빠르게 확산될 수 있다.

4-3. Why Can Worms Spread Quickly?

Worm의 중요한 특징은 **Self-Propagation**이다.

Virus는 일반적으로 감염된 File의 실행과 전달에 의존하지만,
Worm은 취약한 System을 찾아 독립적으로 전파를 시도할 수 있다.

예를 들어 감염된 Computer가 각각 두 개의 새로운 Computer를 감염시킨다고 하자.

$$1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 8 \longrightarrow 16$$

단순화된 모델에서는 전파 단계가 t일 때 새로운 감염 수를

$$N_t = 2^t$$

와 같이 표현할 수 있다.

하지만 실제 감염 속도는 다음에 따라 달라진다.

- 취약한 Computer의 수
- Network Connectivity
- Firewall 및 Security Configuration
- Patch 적용 여부

핵심

Worm의 전파는 여러 감염된 Host에서 동시에 진행될 수 있다. 다만 실제 확산이 항상 지수적으로 증가하는 것은 아니다.

4-4. Real Case: WannaCry (2017)

2017년 발생한 **WannaCry**는 Worm-like Propagation과 Ransomware Payload를 결합한 대표적인 Malware이다.

WannaCry는 Windows의 SMBv1 관련 취약점 (MS17-010)을 악용하여 다른 Computer로 확산할 수 있었다.

Vulnerable Windows Computer

⇓

SMB Vulnerability Exploitation

⇓

Worm-like Propagation

⇓

File Encryption

⇓

Ransom Demand

- **Propagation**: Network를 통한 자동 확산
- **Payload**: File Encryption
- **Objective**: 금전 요구

핵심

WannaCry는 하나의 Malware가 Worm의 전파 특성과 Ransomware의 악성 기능을 동시에 가질 수 있음을 보여준다.

4-5. What Does WannaCry Teach Us?

WannaCry 사례에서 중요한 것은 Malware의 이름 자체가 아니다.

공격이 성공한 이유와 어떤 Security Failure가 발생했는지 분석해야 한다.

Security Analysis of WannaCry

Concept	WannaCry Example
Asset	Windows Computer, User Files
Vulnerability	SMBv1 관련 취약점
Attack	취약점 악용을 통한 Malware 전파
Payload	File Encryption
Security Impact	Availability 침해
Countermeasure	Security Patch, Network Segmentation, Backup

특히 Microsoft는 WannaCry가 대규모로 확산되기 전인 2017년 3월에 관련 보안 업데이트를 제공하였다.

중요

알려진 Vulnerability라도 Patch가 적용되지 않으면 대규모 Security Incident로 이어질 수 있다.

4-6. (참고) How Did EternalBlue Work?

WannaCry는 Windows SMBv1의 Vulnerability를 악용하는 EternalBlue를 이용하여 전파되었다.

공격자는 취약한 Computer에 특수하게 구성된 SMB Request를 전송한다.

이 Request를 처리하는 과정에서 Windows Kernel의 Memory Corruption이 발생할 수 있다.

Attacker

⇓

Malformed SMB Request (TCP 445)

⇓

SMBv1 Vulnerability

⇓

Kernel Memory Corruption

⇓

Remote Code Execution

핵심

EternalBlue는 SMBv1의 Memory Corruption Vulnerability를 악용하여 Remote Code Execution을 가능하게 하는 Exploit이다.

4-7. (참고) From Buffer Overflow to RCE

Memory Corruption이 발생했다고 해서 공격자의 Code가 자동으로 실행되는 것은 아니다.

예를 들어 다음과 같은 Memory Layout을 생각해보자.

Buffer (16 Bytes)
Adjacent Memory (48 Bytes)

16 Bytes만 할당했는데 64 Bytes를 기록하면 48 Bytes가 할당 범위를 벗어난다.

이때 공격자가 중요한 Memory Structure를 조작할 수 있다면 실행 흐름을 변경할 가능성이 있다.

Buffer Overflow ⟶ Memory Corruption ⟶ Control-Flow Hijacking

핵심

Buffer Overflow 자체가 RCE는 아니다. 공격자가 Memory Corruption을 이용하여 실행 흐름을 장악해야 Code Execution으로 이어질 수 있다.

※ 16/64 Bytes는 원리 설명용 가상 예시이며, EternalBlue의 실제 Memory Layout을 나타내지 않는다.

4-8. (참고) Code Injection and WannaCry

그렇다면 공격자가 실행하려는 Malware는 처음부터 피해 Computer에 설치되어 있었을까?

그렇지 않다.

WannaCry의 Worm Propagation 과정에서는 EternalBlue를 통해 초기 공격 Code를 실행시키고, 이후 Malware Payload를 실행하는 방식이 사용되었다.

Step 1. EternalBlue Exploitation

SMBv1 Vulnerability를 이용한 Code Execution

⇓

Step 2. Shellcode Execution

공격자가 전달한 초기 Code 실행

⇓

Step 3. Payload Loading

추가 Malware Code 적재 및 실행

⇓

Step 4. WannaCry Execution

File Encryption 및 추가 Worm Propagation

핵심

WannaCry는 피해 System에 Ransomware가 미리 설치되어 있어야 하는 공격이 아니다. 취약점 악용으로 초기 Code Execution을 확보한 뒤 추가 Payload를 실행할 수 있었다.

※ 실제 EternalBlue의 Kernel Exploitation과 Payload Loading은 단순한 Function Pointer Overwrite보다 복잡하다.

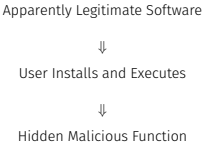
5. What Is a Trojan Horse?

Trojan Horse는 정상적이거나 유용한 Software처럼 보이지만, 숨겨진 악의적인 기능을 수행하는 Malware이다.

예를 들어 사용자가 인터넷에서 무료 Software를 다운로드한다고 생각해보자.

사용자는 정상적인 프로그램이라고 생각하지만, 실제로는 다음과 같은 기능이 포함되어 있을 수 있다.

- 사용자 정보 수집
- Additional Malware 설치
- Backdoor 생성
- 공격자의 Remote Command 실행



핵심

Trojan Horse의 핵심은 정상적인 Software처럼 위장하여 사용자가 악성 기능을 실행하도록 유도하는 것이다.

5-1. How Does a Trojan Work?

공격자가 정상적인 Utility Program처럼 보이는 악성 설치 파일을 배포한다고 하자.

Step 1. Distribution

공격자는 악성 Software를 정상적인 프로그램처럼 위장하여 배포한다.

Step 2. User Execution

사용자는 해당 프로그램을 신뢰하고 직접 설치하거나 실행한다.

Step 3. Malicious Behavior

프로그램은 정상 기능을 제공하는 것처럼 보이면서 숨겨진 악성 기능을 실행한다.

Fake Utility

⇓

User Execution

⇓

Malware Execution

⇓

Information Theft / Backdoor

핵심

Trojan은 반드시 Software Vulnerability를 악용해야만 실행되는 것은 아니다. 사용자의 신뢰와 실행 행위를 악용할 수 있다.

5-2. Virus vs. Worm vs. Trojan

Virus, Worm, Trojan은 모두 Malware이지만, 분류 기준이 완전히 동일하지는 않다.

Comparing Malware Types

	Virus	Worm	Trojan
Main Feature	File Infection	Self-Propagation	Deception
Replication	Yes	Yes	Not Required
Host Program	Usually Required	Not Required	Not Required
Typical Spread	Infected Files	Network	User Installation

주의할 점은 이 분류들이 반드시 서로 배타적이지 않다는 것이다.

예를 들어 Trojan으로 설치된 Malware가 Worm 기능을 통해 다른 Computer로 확산할 수도 있다.

핵심

Malware는 하나의 이름으로만 분류하기보다 Infection, Propagation, Payload 등 여러 관점에서 분석해야 한다.

5-3. (참고) Modern Trojan-Based Attacks

최근에도 Trojan 방식은 Malware Delivery에 널리 사용된다.

다만 과거의 단순한 악성 실행 파일뿐 아니라 다양한 Software Distribution 경로가 악용될 수 있다.

대표적인 예시는 다음과 같다.

- **Fake Software Installer**
  정상적인 프로그램 설치 파일처럼 위장한다.
- **Fake Software Update**
  보안 업데이트 또는 프로그램 업데이트로 위장한다.
- **Malicious Package**
  정상적인 개발용 Package처럼 보이는 악성 Dependency를 배포한다.
- **Trojanized Application**
  정상 Application에 악성 기능을 추가하여 배포한다.

핵심

Trojan의 본질은 특정 File Format이 아니라, 신뢰할 수 있는 Software처럼 보이도록 사용자를 속이는 데 있다.

6. Malware Payloads

Malware가 Target System에 도달하면 다양한 악성 기능을 수행할 수 있다.

이러한 기능을 **Payload**라고 한다.

교재에서는 대표적인 Payload를 다음과 같이 분류한다.

Malware Payload Categories

Payload	Description
Corruption	System 또는 Data를 변경하거나 파괴
Theft of Service	감염된 System Resource를 공격에 악용
Information Theft	Password, Personal Data 등 탈취
Stealthing	Malware의 존재와 악성 행위 은폐

하나의 Malware는 여러 Payload를 동시에 수행할 수 있다.

핵심

Payload는 Malware가 수행하는 악성 기능이며, Propagation은 Malware가 확산되는 방식이다.

6-1. Information Theft

Information Theft는 감염된 System에서 중요한 정보를 수집하고 공격자에게 전달하는 악성 행위이다.

대표적인 정보 탈취 대상은 다음과 같다.

- Login Credentials
- Passwords
- Browser Cookies
- Personal Information
- Confidential Documents

대표적인 Malware 유형에는 Spyware와 Keylogger가 있다.

Compromised Computer

⇓

Collect Sensitive Information

⇓

Exfiltration to Attacker

핵심

Information Theft Malware는 시스템을 즉시 파괴하지 않더라도 Confidentiality를 심각하게 침해할 수 있다.

6-2. How Does a Keylogger Work?

Keylogger는 사용자의 Keyboard Input을 기록하는 Software 또는 Hardware이다.

악성 Keylogger는 사용자가 입력하는 민감한 정보를 수집하는 데 사용될 수 있다.

예를 들어 사용자가 다음과 같이 로그인한다고 하자.

Username: alice

Password: ********

악성 Keylogger는 Keyboard Event를 관찰하여 입력된 문자열을 수집할 수 있다.

Keyboard Input

⇓

Keylogger

⇓

Captured Keystrokes

⇓

Attacker

핵심

Keylogger는 사용자의 입력을 수집하므로, 정상적인 Encryption을 사용하더라도 End-point에서 정보가 유출될 수 있다.

6-3. What Is a Backdoor?

Backdoor는 정상적인 Security Check를 우회하여 System에 접근할 수 있도록 하는 Mechanism이다.

일반적인 System에서는 사용자가 Authentication을 통과해야 한다.

User

⇓

Authentication

⇓

Access Control

⇓

Protected Resource

하지만 Backdoor가 존재하면 공격자가 정상적인 Security Check를 우회할 수 있다.

Attacker ⟶ Backdoor ⟶ System

핵심

Backdoor는 공격자가 이후에도 Unauthorized Access를 수행할 수 있도록 만드는 접근 경로가 될 수 있다.

6-4. What Is a Rootkit?

Rootkit은 Malware나 공격자의 활동을 System에서 숨기기 위한 기술 또는 도구 모음이다.

예를 들어 공격자가 악성 Process를 실행하고 있다고 하자.

정상적인 상황에서는 Operating System의 Process List에서 해당 Process를 관찰할 수 있다.

하지만 Rootkit은 System의 동작이나 관찰 결과를 조작하여 악성 Process를 숨길 수 있다.

Malicious Process Running

⇓

Rootkit Manipulates Visibility

⇓

Security Tool May Not See Process

Rootkit은 User Mode, Kernel Mode 등 다양한 수준에서 동작할 수 있다.

핵심

Rootkit의 중요한 목적은 Malware를 직접 확산시키는 것이 아니라, 악성 행위나 공격자의 존재를 숨기는 것이다.

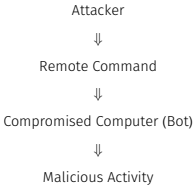
6-5. Bot and Zombie

공격자는 Malware를 이용하여 다른 사람의 Computer를 원격으로 제어할 수 있다.

이렇게 공격자의 명령을 받아 동작하는 감염된 System을 **Bot 또는 Zombie**라고 부른다.

Bot은 다음과 같은 악성 행위를 수행할 수 있다.

- Distributed Denial-of-Service (DDoS)
- Spam Message 전송
- Additional Malware 다운로드
- Information Collection



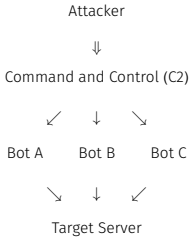
핵심

Bot은 공격자가 감염된 System의 Computing Resource와 Network Connection을 자신의 목적에 이용할 수 있도록 한다.

6-6. Botnet Architecture

Botnet은 공격자가 통제하는 다수의 감염된 Device로 구성된 Network이다.

공격자는 Command and Control (C2) Mechanism을 통해 여러 Bot에 명령을 전달할 수 있다.



예를 들어 공격자가 모든 Bot에 동일한 Target으로 Traffic을 보내도록 명령하면, DDoS Attack이 발생할 수 있다.

핵심

Botnet은 다수의 감염된 System을 하나의 공격 자원처럼 활용할 수 있게 한다.

6-7. Real Case: Mirai Botnet (2016)

2016년 등장한 **Mirai**는 IoT Device를 감염시켜 대규모 Botnet을 형성한 대표적인 Malware이다.

주요 Target은 다음과 같았다.

- IP Camera
- DVR
- Network Router
- 기타 Internet-connected IoT Devices

Mirai의 대표적인 초기 감염 방식은 Telnet Service와 Default 또는 Weak Credentials를 악용하는 것이었다.

Internet-connected IoT Device

⇓

Weak / Default Credentials

⇓

Unauthorized Login

⇓

Mirai Infection

⇓

Botnet Participation

핵심

Mirai는 복잡한 Zero-day Exploit 없이도 취약한 Authentication과 Device Configuration이 대규모 공격으로 이어질 수 있음을 보여준다.

6-8. How Does Mirai Enable DDoS?

Mirai에 감염된 IoT Device는 공격자의 C2 Infrastructure를 통해 공격 명령을 받을 수 있다.

예를 들어 공격자가 특정 Server를 공격 대상으로 지정한다고 하자.

Step 1. Botnet Formation

다수의 IoT Device가 감염되어 공격자의 통제를 받는다.

Step 2. Attack Command

공격자가 Bot에 DDoS Attack 명령을 전달한다.

Step 3. Traffic Flooding

감염된 Device들이 동시에 Target으로 대량의 Traffic을 전송한다.

$$\text{Many Bots} \longrightarrow \text{Excessive Traffic} \longrightarrow \text{Service Disruption}$$

핵심

개별 IoT Device의 성능이 높지 않더라도 많은 Device가 동시에 공격하면 큰 Availability 문제를 발생시킬 수 있다.

6-9. (최근의 공격 동향) Ransomware

Ransomware는 피해자의 Data나 System에 대한 접근을 제한하거나 정보를 탈취하고, 금전을 요구하는 Malware 또는 공격 유형이다.

전통적인 Ransomware는 주로 File Encryption을 이용하였다.

User Files

⇓

Malware Execution

⇓

File Encryption

⇓

Files Become Inaccessible

⇓

Ransom Demand

예를 들어 중요한 Document와 Database가 암호화되면 정상적인 업무가 중단될 수 있다.

핵심

Ransomware는 Data를 암호화하여 Availability를 침해할 수 있으며, 정보 탈취를 결합하면 Confidentiality도 침해할 수 있다.

6-10. (최근의 공격 동향) Double Extortion

최근의 Ransomware Attack에서는 단순히 File을 암호화하는 것뿐 아니라 Data Exfiltration을 결합하는 경우도 있다.

이를 **Double Extortion**이라고 부른다.

Compromised System

⇓

Sensitive Data Exfiltration

⇓

File Encryption

⇓

Ransom Demand

공격자는 피해자에게 두 가지 압박을 가할 수 있다.

- File을 복구하려면 금전을 지불하라고 요구
- 금전을 지불하지 않으면 탈취한 정보를 공개하겠다고 위협

중요

Backup은 암호화된 File의 복구에 도움이 되지만, 이미 유출된 Confidential Information을 되돌릴 수는 없다.

6-11. (참고) What Is a Wiper?

Wiper는 Data나 System을 삭제하거나 파괴하는 것을 주요 목적으로 하는 Malware이다.

Ransomware와 유사하게 System을 사용할 수 없게 만들 수 있지만, 공격 목적은 다를 수 있다.

Ransomware vs. Wiper

	Ransomware	Wiper
Primary Objective	Extortion	Destruction
Typical Payload	Encryption / Data Theft	Data Destruction
Recovery	Backup 등이 중요	Backup 등이 중요
Security Impact	Availability / Confidentiality	Availability / Integrity

일부 Wiper는 Ransomware처럼 금전을 요구하는 화면을 표시하여 공격 목적을 위장하기도 한다.

핵심

Ransomware와 Wiper는 모두 System에 피해를 주지만, 전형적인 공격 목적은 각각 Extortion과 Destruction이다.

7. What Is Blended Malware?

초기의 Malware는 하나의 Propagation Mechanism과 하나의 Payload를 사용하는 경우가 많았다.

하지만 현대 Malware는 여러 Mechanism을 결합할 수 있다.

이를 **Blended Malware**라고 한다.

예를 들어 다음과 같은 Malware를 생각해보자.

Phishing

⇓

Trojan Execution

⇓

Downloader

⇓

Backdoor Installation

⇓

Information Theft

이 공격은 하나의 Malware Campaign에서 여러 공격 기능을 결합한 개념적 예시이다.

핵심

Blended Malware는 여러 Propagation Mechanism과 Payload를 결합하여 공격 효과를 높일 수 있다.

7-1. Malware Categories Can Overlap

Malware를 분류할 때 하나의 Malware가 반드시 하나의 Category에만 속한다고 생각해서는 안 된다.

예를 들어 어떤 Malware가 다음과 같이 동작한다고 하자.

1. 정상적인 Software처럼 위장하여 설치된다.
2. Network를 통해 다른 Computer로 전파된다.
3. 감염된 Computer의 File을 암호화한다.
4. 공격자의 Remote Command를 실행한다.

이 Malware는 다음과 같은 특징을 동시에 가진다.

- Trojan: Deceptive Installation
- Worm: Self-Propagation
- Ransomware: File Encryption
- Bot / Backdoor: Remote Control

핵심

Malware를 이해하려면 이름 하나를 암기하기보다 Propagation, Payload, Objective를 각각 분석해야 한다.

8. What Are Attack Kits?

과거에는 Malware를 직접 개발하고 배포하려면 상당한 Programming과 Security Knowledge가 필요했다.

하지만 **Attack Kit**의 등장으로 공격에 필요한 여러 기능을 하나의 도구로 제공할 수 있게 되었다.

Attack Kit은 다음과 같은 기능을 포함할 수 있다.

- Malware Generation
- Exploit Modules
- Payload Configuration
- Delivery Mechanisms
- Attack Management

공격자는 모든 기능을 직접 개발하지 않고도 기존 Module을 조합하여 공격을 수행할 수 있다.

핵심

Attack Kit은 공격 기술의 재사용과 자동화를 통해 공격 수행에 필요한 기술적 장벽을 낮출 수 있다.

8-1. Historical Examples of Attack Kits

교재에서는 과거의 대표적인 Attack Kit과 Crimeware를 소개한다.

Historical Malware Toolkits

Name	Main Characteristic
Zeus	Banking Credential Theft에 사용된 Crimeware
Blackhole	Web-based Exploit Kit
Phoenix	Web-based Exploit Kit
Sakura	Web-based Exploit Kit

이러한 도구들은 Malware와 Exploit의 개발 및 배포를 보다 쉽게 만들었다.

하지만 위 사례들은 주로 2000년대 후반에서 2010년대 초반의 역사적인 사례이다.

핵심

Attack Kit이라는 개념은 현재도 중요하지만, 교재의 대표 사례는 역사적 배경으로 이해하는 것이 좋다.

8-2. (최근의 공격 동향) Malware-as-a-Service

최근의 Cybercrime에서는 Malware와 공격 Infrastructure를 서비스 형태로 제공하는 모델도 등장하였다.

대표적인 개념은 다음과 같다.

- **Malware-as-a-Service (MaaS)**
  Malware 또는 관련 공격 기능을 다른 공격자에게 제공하는 모델
- **Ransomware-as-a-Service (RaaS)**
  Ransomware 운영자가 Malware와 공격 Infrastructure를 제공하는 모델
- **Initial Access Broker (IAB)**
  이미 침해된 System에 대한 접근 권한을 거래하는 공격 주체

이러한 모델에서는 Malware Developer, Access Broker, Attack Operator가 서로 다른 역할을 수행할 수 있다.

핵심

현대 Cybercrime에서는 공격에 필요한 기술과 Infrastructure가 분업화될 수 있다.

Thank You

Questions?