

악성코드 유사 및 변종 유형 예측방법 연구

A study on the prediction method for similar types
and variants of malware.

수탁기관 : 성균관대학교 산학협력단

2011. 11. 27

제 출 문

한국인터넷진흥원 원장 귀하

본 보고서를 “악성코드 유사 및 변종 유형 예측방법 연구”의 최종
연구개발 결과보고서로 제출합니다.

2011 년 11월 27일

수탁 기관 : 성균관대학교 산학협력단

연구책임자 : 교 수 정태명 (성균관대학교 정보통신공학부)

참여연구원 : 연 구 원 박민우 (성균관대학교 전자전기컴퓨터공학과)

연 구 원 강동민 (성균관대학교 전자전기컴퓨터공학과)

연 구 원 장성수 (성균관대학교 전자전기컴퓨터공학과)

연 구 원 민재원 (성균관대학교 전자전기컴퓨터공학과)

요 약 문

1. 제목

악성코드 유사 및 변종 유형 예측방법 연구

2. 연구개발의 목적 및 중요성

각종 산업들과 IT 산업이 융합되면서 사회를 구성하는 많은 부분들이 전산에 의존적으로 변화하였다. 그에 따라 악성코드의 공격 표적이 다양해지고 그 규모 또한 증가하면서 악성코드의 탐지와 치료를 위한 연구가 활발하게 이루어지고 있다. 하지만 악성코드에 대한 연구는 정부, 기업, 연구기관 각각 다른 목적을 갖고 진행되어 왔으며, 기업 내에서도 분석을 위한 명확한 기준이 없어 독자적인 형태로 진행되고 있다. 그 결과 악성코드 분석 결과들 사이에 접점이 존재하지 않아 연구 공조가 어려운 상황이다. 각 기관의 분석 결과 항목이 서로 상이하며 동일한 항목에 대해서도 표기 방법이 서로 달라 쉽게 혼란이 일어난다.

악성코드의 생성과 악성코드 전파로 인한 피해 확산을 막고 효과적인 연구 공조를 위해서는 통일된 악성코드 분류 기준이 필요하다. 따라서 본 보고서에서는 악성코드를 정의·분류할 수 있는 연관성 정보 구조를 만든다. 그 결과 연관성 정보 구조를 이용하여 효과적인 악성코드 분석

환경을 만들기 위해 필요한 요소들을 도출할 수 있으며, 악성코드 분석 환경을 통해 신속하고 정확한 악성코드 분석 체계를 수립을 통해 보다 안전하고 건전한 인터넷 환경 구축이 가능하다.

3. 연구개발의 내용 및 범위

- 악성코드 진화 동향 조사
 - 악성코드 전망 예측 및 대응을 위해 기존 악성코드의 동향 파악
 - 시만텍 연구 보고서 분석
 - 카스퍼스키 랩 연구 보고서 분석
 - 안철수 연구소 연구 보고서 분석
 - 악성코드 공격기술의 진화
 - 악성코드 전파방법의 진화
- 악성코드 기능에 따른 연관성 정보 추출 방법 모색
 - 악성코드의 공격기술 및 전파방법 분석을 통해 악성코드를 정의·분류할 수 있는 기준 도출
 - 악성코드 정의·분류 기준 간의 연관성 정보 추출
- 연관성 정보 구조 구축
 - 악성코드 정의·분류를 위한 연관성 정보 구조 구축
 - 연관성 정보를 통해 악성코드의 위험 수준 평가
 - 새로운 매체에 대한 확장을 위한 방법론 제시
- 악성코드 유사 및 변종 유형 예측방법 연구
 - 연관성 정보 및 최근 기술 정보를 기반으로 기존 악성코드 및 신·변종 악성코드 모델링 방안 연구

4. 연구결과

- 악성코드 개요
 - 연관성 정보 구조 구축을 위해 악성코드의 정의, 역사, 종류 등을 조사함으로써 악성코드에 대해 바르게 이해하며 분석을 통해 연관성 정보 구조를 위한 기준 도출
 - 악성코드 동향 등 조사 및 분석을 통해 국내외 현황 파악 및 위험 수준을 정의하기 위한 정보 수집
- 악성코드 공격 방법
 - 악성코드의 공격 기술 및 그에 대한 대응전략을 분석함으로써 악성코드 연관성 정보 구조의 기준 도출
- 악성코드 전파 방법
 - 악성코드의 전파 기술 및 그에 대한 대응전략을 분석함으로써 악성코드 연관성 정보 구조의 기준 도출
 -
- 연관성 정보 구조
 - 악성코드의 동향, 공격기술·전파기술 분석 등을 통해 연관성 정보 구조 구축을 위한 기준 정의
 - 기준에 따라 연관성 정보 구조 형성
 - 각 기준의 배타적 성격과 종속적 성격 분석
 - 연관성 정보 구조 정의
- 연관성 정보 구조 확장 및 활용
 - 새로운 매체에 대한 확장을 위해 매체특징 도출을 위한 분석 수행
 - 연관성 정보 구조 확장 방안 제시
 - 악성코드의 위험 수준 판단 기준 제시

5. 활용에 대한 건의

- 악성코드 분석을 위한 기반 자료로 활용 가능

- 악성코드 관련 대응 정책을 수립하기 위한 지침
- 악성코드 탐지 및 예방 도구 제작에 적용할 수 있는 방법론으로 활용 가능

6. 기대효과

- 악성코드 연관성 정보 구조는 악성코드 분석 결과를 정리하여 모든 악성코드들에 대해 동일한 형태의 분석 결과를 유도할 수 있다.
- 악성코드 분석 결과의 통일화는 악성코드 분석들 간에 분석자료 공유를 용이하게 만들 수 있으며, 그 결과 연구 과정의 동조가 가능해진다.
- 그 결과 악성코드를 이용한 각종 사이버 범죄를 줄일 수 있으며, 이는 IT 산업 뿐만 아니라 IT와 융합된 많은 산업들의 발전에 긍정적인 효과를 보일 것으로 기대됨

SUMMARY

1. Title

A study on the prediction method for similar types and variants of malware.

2. Purpose of the study

First we grasp the industrial trend and standardization trend of Attack technique and Propagation technique of the malwares. Then we suggest the requirements for the criteria of the malwares. Finally we propose The Relationship Information of Malicious Codes what is used for defining and classifying the malwares.

3. Contents and scope

- o Introduction
 - Background and Need of the study
 - Purpose of the study

- o The Malware
 - Outline of the malwares
 - Definition of the malwares
 - The brief history of the malwares
 - Kind of the malwares
 - Status of the malwares
 - Attack technique of the malwares
 - Attack technique of the malwares
 - Status of Attack technique of the malwares
 - Coping strategy of the malwares' attack technique
 - Propagation technique of the malwares
 - Propagation technique of the malwares
 - Coping strategy of the malwares' Propagation technique

- o The Architecture of The Relationship Information of Malicious Codes
 - Outline of The Relationship Information of Malicious Codes
 - Definition of The Relationship Information of Malicious Codes
 - Need of The Relationship Information of Malicious Codes
 - The Relationship Information of Malicious Codes
 - Expandability of The Relationship Information of Malicious Codes
 - Utilization of The Relationship Information of Malicious Codes

Conclusion

4. Results of the study

The Relationship Information of Malicious Codes can be used for the criteria for sharing studies of the malwares among governments, businesses and research institutes.

5. Expected effects and applications

- o We can analyze malicious codes using same architecture.
- o We can easily share the result about malicious codes among governments, businesses and research institutes.
- o As a result, We can reduce cyber crimes through collaborative research about malicious codes among governments, businesses and research institutes.

목 차

제 1 장 서론	1
제 1 절 연구 배경 및 필요성	1
제 2 절 연구 목표 및 내용	2
1. 연구 목표	2
2. 연구 내용	2
제 2 장 악성코드	4
제 1 절 악성코드 개요	4
1. 악성코드 정의	4
2. 악성코드의 역사	6
3. 악성코드 종류	16
4. 악성코드 동향	20
제 2 절 악성코드 공격 방법	30
1. 악성코드 공격기술	30
2. 최근 악성코드 공격 기술 동향	37
3. 악성코드 대응전략	50
제 3 절 악성코드 전파 방법	55
1. 악성코드 전파경로	55
2. 전파경로별 대응전략	72
제 4 절 레지스트리 악용 방법	74
1. 레지스트리 개요	74
2. 레지스트리 악용 방법	77

제 3 장 연관성 정보 구조	15
제 1 절 연관성 정보 구조 개요	105
1. 연관성 정보 구조 정의	105
2. 연관성 정보 구조 필요성	105
제 2 절 연관성 정보 구조	106
1. 감염경로	108
2. 실행주체	118
3. 공격대상	130
4. 공격행위	150
5. 가중치	176
제 3 절 연관성 정보 구조 확장성	195
1. 신종·변종 악성코드에 대한 확장성 검토 및 적용	195
2. 새로운 매체 분석 및 적용	197
제 4 절 연관성 정보 구조 활용방안	217
1. 연관성 정보 구조 위험 지수	217
2. 악성코드 그룹 별 위험 지수 분포	229
3. 변종 악성코드 예측	241
제 5 절 연관성 정보 구조 예시	250
1. 특정 온라인 게임계정 탈취 악성코드(트로이목마)	250
2. 악성봇(웜)	251
3. 3.4 DDoS 악성코드	252
4. 악성코드 위험지수 도출 결과	253
 제 4 장 결론	 256
참고문헌	258
부록	260

그림 목차

(그림 2-1) Elk Cloner가 출력하는 텍스트	7
(그림 2-2) 1984-2010년 동안 발견된 누적 악성코드 수	20
(그림 2-3) 맥아피가 보유한 시그니처 수	21
(그림 2-4) 국내 인터넷 이용률 및 이용자 수 현황	22
(그림 2-5) 2009-2010년 월별 국내 악성코드 발생건수 추이	22
(그림 2-6) 2009-2010년 월별 국외 악성코드 발생건수 추이	23
(그림 2-7) 2009-2010년 국외 웹 기반 공격 평균 횟수	26
(그림 2-8) 2009-2010년 국내 악성코드 감염 도메인 수	27
(그림 2-9) 국내 악성 봇 감염 PC 수	28
(그림 2-10) 컨픽커(Conficker) 악성코드	31
(그림 2-11) 웨일택(Waledac) 악성코드	32
(그림 2-12) IRC봇(IRCBot)	33
(그림 2-13) 네이트온(NateOn) 악성코드	33
(그림 2-14) 조커(Joker) 악성코드	34
(그림 2-15) 클램피(Clampi) 악성코드	34
(그림 2-16) 델프(Delf) 악성코드	35
(그림 2-17) 지봇(Zbot) 악성코드	35
(그림 2-18) 허위 보안툴(SecurityTool)	36
(그림 2-19) AntiVirus XP 2010	36
(그림 2-20) 스텍스넷(Stuxnet)	37
(그림 2-21) Palevo	37
(그림 2-22) 일반적인 컴파일과 난독화 컴파일 과정	39
(그림 2-23) 악성코드 대량삽입 흐름도	45
(그림 2-24) 기능별로 모듈화를 사용하는 쿵페이스(Koobface) 구성	

도	46
(그림 2-25) Ring 보안 모델	47
(그림 2-26) 현재 AV industry의 모습 (출처 : IKARUS)	55
(그림 2-27) 악성코드 전파 방법의 유형	57
(그림 2-28) 레지스트리 구조	75
(그림 3-1) 악성코드 동작 과정	106
(그림 3-2) 새로운 매체 발생할 경우 분석 및 적용 과정	198
(그림 3-3) 감염대상의 연관성 정보 구조 적용과정	200
(그림 3-4) 악성코드 전파의 연관성 정보 구조 적용과정	201
(그림 3-5) 서비스에 의해 발생하는 정보의 연관성 정보 구조 적용 과정	203
(그림 3-6) 서비스와 직접연관이 없는 사용자 재산의	205
(그림 3-7) 서비스 방해의 연관성 정보 구조 적용과정	206
(그림 3-8) 혼란 초래의 연관성 정보 구조 적용과정	207
(그림 3-9) 위험 지수 기본 공식	218
(그림 3-10) 악성코드 위험지수 $f(x)$	218
(그림 3-11) 악성행위별 위험지수 계산 식]	218
(그림 3-12) 연관성 정보구조 변수	219
(그림 3-13) 확산력	219
(그림 3-14) 악성코드 그룹 별 위험 지수 분포	229
(그림 3-15) 변종 악성코드의 목적 그룹 일치률	243
(그림 3-16) 악성코드 변종 예측	246
(그림 3-17) 70% 이상의 일치률을 가지는 경우 변종 예측	247
(그림 A-1) MTAS: 메인 이미지	260
(그림 A-2) MTAS: 감염경로	262
(그림 A-3) MTAS: 실행주체	262

(그림 A-4) MTAS: 공격대상	263
(그림 A-5) MTAS: 공격행위	263
(그림 A-6) MTAS: 전파경로	264
(그림 A-7) MTAS: 자가보호	264
(그림 A-8) MTAS: 목표대상	265
(그림 A-9) MTAS: 출력결과 1	265
(그림 A-10) MTAS: 출력결과 2	266

표 목차

[표 2-1] 악성코드에 의한 증상	5
[표 2-2] 누적 악성코드 수 추정치	9
[표 2-3] 악성코드의 주요 사건	10
[표 2-5] 코드 난독화 기법	40
[표 2-6] 패킹 툴 목록	41
[표 2-7] 악성코드 탐지 및 대응 기술 구분	50
[표 2-8] 저장매체를 이용하여 전파하는 사례	68
[표 2-9] 네트워크를 이용하여 전파하는 사례	68
[표 2-10] 다운로드를 통해 전파하는 사례	69
[표 2-11] 전자우편을 이용하여 전파하는 사례	69
[표 2-12] 취약성을 이용하여 전파하는 사례	70
[표 2-13] 인스턴트 메시징 프로그램을 이용하여 전파하는 사례	71
[표 2-14] 새로운 방식을 이용하여 전파하는 사례	71
[표 2-15] 서비스 등록을 위해 악용되는 레지스트리	78
[표 2-16] 자동 실행을 위해 악용되는 레지스트리	85
[표 2-17] 파일 확장자를 연결하는 레지스트리	92
[표 2-18] 윈도우 탐색기 관련 행위에 악용되는 레지스트리	94
[표 2-19] 인터넷 익스플로러 관련 설정을 악용하는 레지스트리	98
[표 2-20] 윈도우 설정을 악용하는 레지스트리	100
[표 2-21] 네트워크 설정을 악용하는 레지스트리	102
[표 2-22] 악성코드가 존재 여부를 확인하는 레지스트리	103
[표 2-23] 메일주소와 서버 주소를 가져오는 레지스트리	103
[표 2-24] 키값을 참조하는 레지스트리	104
[표 3-1] 연관성 정보 구조 1 단계 정의	107

[표 3-2] 감염경로의 2 단계	109
[표 3-3] [감염경로]의 감염 이용 매체	110
[표 3-4] 감염경로의 감염 유형	116
[표 3-5] 감염경로의 사용자 의존도	117
[표 3-6] 실행주체의 2 단계	119
[표 3-7] 사용자 실행의 의미	120
[표 3-8] 운영체제 실행의 의미	122
[표 3-9] 운영체제 자동 실행의 하위 단계	124
[표 3-10] 운영체제 실행파일 감염의 하위 단계	125
[표 3-11] 웹브라우저 실행의 의미	125
[표 3-12] 다른 응용프로그램 실행의 의미	128
[표 3-13] 공격대상의 하위 분류	131
[표 3-14] 시스템의 하위 단계	132
[표 3-15] 시스템 장치 제어의 하위 단계	132
[표 3-16] 시스템 서비스 제어의 하위 단계	134
[표 3-17] 시스템 설정 제어의 하위 단계	135
[표 3-18] 사용자 정보의 하위 단계	137
[표 3-19] 사용자 입력 정보의 하위 단계	138
[표 3-20] 사용자 저장 정보의 하위 단계	139
[표 3-21] 사용자 시스템 정보의 하위 단계	141
[표 3-22] 사용자 시스템 사용 이력의 하위 단계	143
[표 3-23] 시스템 사용자의 하위 단계	145
[표 3-24] 이용 방해의 하위 단계	146
[표 3-25] 혼란 초래의 하위 단계	147
[표 3-26] 외부 장치의 하위 단계	148
[표 3-27] 서비스 서버의 하위 단계	149

[표 3-28] 로컬 네트워크의 하위 단계	150
[표 3-29] 공격행위의 2 단계	151
[표 3-30] 공격행위의 네트워크	152
[표 3-31] 유출 정보의 종류	153
[표 3-32] 서버 접속의 목적	157
[표 3-33] 대량 트래픽의 생성 목적	158
[표 3-34] 스팸의 종류	160
[표 3-35] 공격행위의 시스템	161
[표 3-36] 강제 시스템 제어	162
[표 3-37] 자원 관리 하위 단계	164
[표 3-38] 네트워크 설정 변경 하위 단계	165
[표 3-39] 시스템 설정 변경 하위 단계	167
[표 3-40] 디스플레이 설정 변경 하위 단계	170
[표 3-41] 파일시스템 하위 단계	171
[표 3-42] 파일생성 종류	171
[표 3-43] 파일파괴 종류	172
[표 3-44] 파일변조 종류	173
[표 3-45] 입출력장치 오작동 종류	174
[표 3-46] 프로세스 하위 단계	175
[표 3-47] 전파경로의 2 단계	177
[표 3-48] [전파경로]의 전파 이용 매체	178
[표 3-49] 전파경로의 사용자 의존도	179
[표 3-50] 악성코드 자가보호 종류	179
[표 3-51] 난독화의 종류	180
[표 3-52] 패킹의 종류	181
[표 3-53] 분석도구 종류	182

[표 3-54] 분석도구 탐지에 사용 가능한 API	183
[표 3-55] 분석 도구 탐지를 위한 프로세스 상태 확인	184
[표 3-56] 분석 도구 탐지를 위한 프로세스 상태 확인	185
[표 3-57] 루트킷의 종류	186
[표 3-58] Hooking의 종류	187
[표 3-59] 시스템 변조 종류	189
[표 3-60] Injection의 종류	190
[표 3-61] 목표대상 공격목표	190
[표 3-62] 잠재위험 위험지수 가중치	193
[표 3-63] 신·변종 악성코드 확장성 검토 및 적용	196
[표 3-64] 감염 대상 및 예시	199
[표 3-65] 감염 목적 및 예시	200
[표 3-66] 악성코드 전파의 일반화	201
[표 3-67] 서비스에 의해 발생하는 정보에서 정보보관의 일반화	202
[표 3-68] 서비스에 의해 발생하는 정보에서 접근경로의 일반화 과정	203
[표 3-69] 서비스와 직접 연관이 없는 사용자 재산의 일반화	204
[표 3-70] 서비스 방해의 일반화	206
[표 3-71] 혼란 초래의 일반화	207
[표 3-72] 전자투표에서 가능한 악성행위	209
[표 3-73] 전자투표에서 매체의 일반화 적용	210
[표 3-74] U-health에서 가능한 악성행위	211
[표 3-75] U-health에서 매체의 일반화 적용	212
[표 3-76] VANET을 이용하여 가능한 악성행위	213
[표 3-77] VANET에서 매체의 일반화 적용	214

[표 3-78] VANET을 이용하여 가능한 악성행위	215
[표 3-79] ATM에서 매체의 일반화 적용	216
[표 3-80] 연관성 정보 구조 : 감염경로	220
[표 3-81] 연관성 정보 구조 : 실행주체	222
[표 3-82] 연관성 정보 구조 : 공격대상	223
[표 3-83] 연관성 정보 구조 : 공격행위	224
[표 3-84] 연관성 정보 구조 : 전파경로	225
[표 3-85] 연관성 정보 구조 : 자가보호	226
[표 3-86] 목표대상 위험지수 가중치	227
[표 3-87] 잠재위험 위험지수 가중치	228
[표 3-88] 악성코드 그룹	229
[표 3-89] 연관성 구조 제외 항목 : 트로이 목마	231
[표 3-90] 연관성 구조 포함 항목 : 트로이 목마	231
[표 3-91] 위험 지수 : 트로이 목마	232
[표 3-92] 연관성 구조 제외 항목 : 바이러스	232
[표 3-93] 연관성 정보 구조 포함 항목 : 바이러스	233
[표 3-94] 위험 지수 : 바이러스	233
[표 3-95] 연관성 정보 구조 포함 항목 : 웜	233
[표 3-96] 위험 지수 : 웜	233
[표 3-97] 연관성 구조 제외 항목 : 웜	234
[표 3-98] 연관성 구조 제외 항목 : 조크, 혹스	235
[표 3-99] 연관성 정보 구조 포함 항목 : 조크, 혹스	236
[표 3-100] 위험 지수 : 조크, 혹스	236
[표 3-101] 연관성 정보 구조 제외 항목 : 스파이웨어, 애드웨어	236
[표 3-102] 연관성 구조 포함 항목 : 스파이웨어, 애드웨어	238

[표 3-103] 위험 지수 : 스파이웨어, 애드웨어	238
[표 3-104] 연관성 정보 구조 제외 항목 : DDoS 봇	238
[표 3-105] 연관성 정보 구조 포함 항목 : DDoS 봇	239
[표 3-106] 위험 지수 : DDoS 봇	239
[표 3-107] 연관성 정보 구조 제외 항목 : APT	240
[표 3-108] 연관성 정보 구조 포함 항목 : APT	241
[표 3-109] 위험 지수 : APT	241
[표 3-110] 악성코드 분류 그룹	242
[표 3-111] 연관성 정보 구조와 분류 그룹 연관관계 : 공격대상	244
[표 3-112] 연관성 정보 구조와 분류 그룹 연관관계 : 공격행위	244
[표 3-113] 3.4 DDoS 악성코드	248
[표 3-114] 목적 그룹과의 일치률	249
[표 3-115] 시스템파괴형 악성행위 항목	249
[표 3-116] 악성코드 위험지수 도출 예시 : 트로이목마	251
[표 3-117] 악성코드 위험지수 도출 예시 : IRC봇(웜)	252
[표 3-118] 악성코드 위험지수 도출 예시 : 3.4 DDoS	253
[표 3-119] 악성코드 위험지수 도출 결과	254
[표 3-120] 변종 악성행위 목록	254
[표 A-1] mtas.php 소스코드	267
[표 A-2] create_xml.php 소스코드	275

Contents

Chapter 1 Introduction	1
Section 1 Background and Need of the study	1
Section 2 Purpose of the study	2
 Chapter 2 The Malware	4
Section 1 Outline of The Malwares	4
1. Definition of Malware	4
2. The Brief History of Malware	6
3. The Sorts of Malware	16
4. The Status of Malware	20
Section 2 Attack Technique of The Malwares	30
1. Attack Technique of The Malwares	30
2. Status of Attack Technique of The Malwares	37
3. Coping Strategy for the Attack Techniques of Malware	50
Section 3 Propagation Technique of The Malwares	55
1. Propagation Technique of The Malwares	55
2. Coping Strategy for Propagation Techniques of Malware	72
Section 4 Abusing Techniques of The Registries	74
1. Outline of the Registries	74
2. Abusing Techniques of the Registries	77

Chapter 3 The Architecture of Relationship Information of Malicious Codes	105
Section 1 Outline of Relationship Information of Malicious Codes	105
1. Definition of Relationship Information of Malicious Codes	105
2. The Necessity of Relationship Information of Malicious Codes	105
Section 2 Relationship Information of Malicious Codes	106
1. The Infection Route	108
2. The Practice Subject	118
3. The Target of Attack	130
4. The Offensive Action	150
5. The Weight	176
Section 3 Expandability of Relationship Information of Malicious Codes	195
1. New types and Variants of the Malware	195
2. Analysis and Implementation about the New Medium ·	197
Section 4 Utilization of Relationship Information of Malicious Codes	217
1. Threat Index of The Relationship Information Malicious Codes	217
2. Distribution of Threat Index by Malicious Codes Group	229
3. Prediction of The Mutant Malware	241

Section 5 Example of The Relationship Information of Malicious Codes	250
1. Game Account Hijacked Malware(Trojan Horse)	250
2. Malicious Bot(Worm)	251
3. 3.4 DDoS Malware	252
4. The Result of Threat Index	253
Chapter 4 Conclusion	256
References	258
Appendix	256

제 1 장 서론

제 1 절 연구 배경 및 필요성

자가복제가 가능한 바이러스가 처음 발견된 이후 다양한 악성행위와 전파경로를 갖는 악성코드들이 발견되고 있다. 악성코드를 만들 수 있는 툴킷이 발달하면서 전문적인 지식이 없는 일반인도 쉽게 복잡한 악성코드들을 만들 수 있게 되면서 최근에는 하루에도 수천 개의 악성코드들이 만들어져 네트워크에 쏟아져 나오고 있다. 각종 산업들이 IT 산업과 융합되면서 많은 종류의 이로운 서비스들이 가능해졌지만 반대로 악성코드에 의한 피해대상이 IT 산업과 융합된 다른 산업으로 확대되면서 악성코드에 의한 피해규모가 크게 증가하였다. 악성코드로 인한 피해를 최소화하기 위해 정부와 기업과 연구기관 사이에서 악성코드를 분석하기 위한 많은 활동이 이루어지고 있다.

하지만 세 기관의 연구결과가 제대로 공유되지 않아 동일한 악성코드를 다르게 정의하거나, 중복된 연구 수행을 통해 연구효율이 크게 떨어지고 있다. 이러한 현상의 원인은 각 기관들의 연구목적 및 연구방법의 차이에서 나온다. 또한, 동일한 목적으로 악성코드에 대해 연구하는 기관들 사이에서도 분석결과물의 형태는 크게 다른 모습을 보이고 있다. 이는 현재 악성코드의 정의 및 분류에 대한 명확한 기준점이 없어 분류 기준 및 정의가 즉흥적으로 이루어지고 있기 때문이다.

최근 악성코드에 감염되어 피해를 입는 사례가 늘어나고 있다. 올해 상반기에만 발견된 악성코드 수가 15만개에 달하며, 악성코드의 위협은 해가 갈수록 증가하고 있고, 기술적으로도 치명적이고 위협적으로 변하여 사용자 혹은 사회에 불안요소를 가지게 만든다. 악성코드의 목적이 비교적 단순하고 제작되는 수가 적었던 과거 시절과는 달리 하루에도 수천개의 악성코드가 쏟아져 나오는 오늘 각 연구기관의 독립적인 연구활

동으로는 시간 내에 충분한 연구결과를 얻어내기 어려워졌다. 이러한 문제점을 타개하기 위해서는 연구기관 간에 연구결과와 활발한 공유와 공동연구가 필요하게 되었다. 이를 위해서는 현재 판이하게 다른 악성코드 정의 및 분류 과정에 대해 하나의 새로운 기준이 필요하다.

또한, 악성코드를 효과적으로 방어하기 위해서는 기존의 악성코드들을 바탕으로 방어만 하는 방법을 고수하기 보다는 기존에 존재하던 악성코드를 통해 신종 악성코드를 효율적으로 방어할 수 있는 악성코드 유사 및 변종 유형을 예측하는 방법을 연구하는 것이 도움이 될 것으로 기대된다.

제 2 절 연구 목표 및 내용

1. 연구 목표

악성코드 기능에 따른 연관성 정보 및 최근 기술 정보를 기반으로 기존 악성코드에서 파생 가능한 신·변종 악성코드 모델링 방법을 제시

2. 연구 내용

가. 악성코드 진화에 따른 공격 기술 및 전파 방법 동향 조사

악성코드의 향후 전망을 예측하고 이에 대응하기 위해서는 기존의 악성코드의 동향 파악에 대한 조사 분석이 이루어져야 한다.

악성코드의 진화에 따른 공격 기술 및 전파 방법 동향 파악을 위해서는 기존 악성코드들을 조사·분석한다. 조사·분석 대상으로는 시만텍(Symantec), 맥아피(McAfee), 트렌드 마이크로(Trend Micro), 카스퍼스키랩(Kaspersky Lab) 등의 인지도가 높은 주요 세계 안티 바이러스 및 보

안 기업의 보안 보고서를 기반으로 동향 조사·분석한다. 기존의 악성코드와 이슈가 되는 악성코드에 악용 가능한 최근 기술까지 대상으로 한다.

나. 악성코드 기능에 따른 연관성 정보 추출 방법 연구

악성코드의 공격 기술 및 전파 방법을 기준으로 악성코드 간의 연관성 정보를 추출하는 방법을 도출하기 위해 악성코드들의 특성을 분류하여 각 악성코드들 간 유사한 특징들을 고려하여 취합하여 악성코드를 기능에 따라 분류하여 연관성 정보를 추출할 수 있는 방법을 연구한다.

다. 악성코드 유사 및 변종 유형 예측방법 연구

지속적으로 발전하고 변형되는 악성코드의 기법들로부터 보유 자원을 안전하게 지키기 위해 악성코드 연관성 정보 및 최근 기술 정보를 기반으로 기존 악성코드 및 신·변종 악성코드 모델링 방법을 연구한다. 또한 신규 악성코드가 발생했을 경우 모델 검증을 하고, 악성코드의 위험성을 평가하는 방법에 대해 연구한다. 추가로 악성코드 유사 및 변종 유형 예측 방법에 대한 연구결과를 검증할 수 있는 소프트웨어를 설계하고 구현한다.

제 2 장 악성코드

제 1 절 악성코드 개요

1. 악성코드 정의

멀웨어(Malware)는 악성 소프트웨어(Malicious Software)의 줄임말로, 악의적인 목적을 가지고 제작되어 컴퓨터에 악영향을 끼치는 모든 소프트웨어를 칭한다. 이러한 소프트웨어는 사용자의 명령이나 승인 없이 설치되거나 실행되며 시스템의 성능 저하 또는 개인 정보 유출 등의 악의적인 행위를 수행한다. 국내에서는 멀웨어를 “악성코드”로 총칭하고 있으며 컴퓨터 바이러스(Virus), 웜(Worm), 트로이목마(Trojan Horse), 스파이웨어(Spyware), 루트킷(Rootkit) 등이 모두 악성코드에 속한다.

악성코드로 인한 피해가 사회적인 문제로 대두되면서 이를 해소하기 위한 연구가 활발히 진행되고 있다. 이를 위해 수많은 회사들이 샘플들을 수집하여 분석하고 안티바이러스 제품들을 출시하고 있으며 악성코드 동향 보고서를 주기적으로 배포하고 있다. 세계적인 보안 회사 시만텍은 악성코드의 증가율이 정상 프로그램의 증가율보다 높을 수도 있다고 예상을 했고, 2010년에는 중국의 사오잉(Shaoxing)을 세계 악성코드의 중심지로 선정하기도 했다.

악성코드의 전파는 초기 플로피 디스크와 같은 저장매체를 통해 이루어졌으나 최근 인터넷 사용이 일반화되면서 네트워크를 통한 악성코드 전파가 보다 보편적인 요인이 되었다. 이는 보안이 취약한 사이트접속, 전자우편, 메신저 등 네트워크를 통해 서비스되는 다양한 매체들을 통해 악성코드가 전파됨을 의미한다. 또한 다양한 산업들이 IT 산업과 융합되고 인터넷을 통해 많은 컴퓨팅 장치들이 연결되면서 악성코드의 제작 및 유포 목적이 점점 구체화되고 있다. 이에 따라 악성코드에 의한 피해규

모가 크게 증가하였으며, 피해대상이 다양해졌다. 현재의 악성코드는 단순히 시스템을 파괴하는 것뿐만 아니라 사용자의 의도와는 상관없는 스팸 메일 발송, 원격제어, 그리고 광고 팝업 등의 피해를 주기도 한다. 컴퓨터가 악성코드에 감염되었다면 다음과 같은 증상이 있을 수 있다.

[표 2-1] 악성코드에 의한 증상

증상	세부내용
시스템 정보 변경	레지스트리 키 값 변경
FAT 파괴	FAT 유형 파일 시스템 파괴
CMOS 변경	CMOS 변경으로 인한 부팅 장애 유발
기본 메모리 감소	시스템의 기본 메모리 감소로 인한 성능 저하
시스템 속도 저하	시스템의 처리 속도 저하
프로그램 자동 실행	부팅 시 악성코드가 실행되도록 설정 변경
프로세스 종료	특정 프로세스를 강제 종료
시스템 재부팅	시스템을 강제로 재부팅
전자우편 발송	특정 사용자들에게 전자우편을 발송
정보 유출	사용자 정보를 네트워크를 통해서 유출
네트워크 속도 저하	감염된 PC의 네트워크 속도가 감소
메시지 전송	다른 PC로 메시지를 전송
특정 포트 오픈	특정 백도어 포트를 개방
하드 드라이브	저장장치 포맷 또는 접근 장애 유발
파일 생성	사용이력, 정보저장, 쓰레기파일 생성 등 다양
파일 삭제	시스템파일, 사용자파일 삭제
파일 손상	파일의 일부 변경을 통한 파일 손상
화면 출력	특정 내용이 파일에 출력
특정음 발생	비트음이나 스피커를 통해 소리 발생
메시지 출력	화면에 특정 메시지를 출력
실행 가능한 코드	매크로나 스크립터 언어로 짜인 코드 실행
시스템사용이력 저장	방문한 홈페이지, 실행한 프로세스 등 사용자가 시스템을 사용하면서 발생하는 정보 저장
네트워크 트래픽 저장	네트워크 장치를 통해 이동된 트래픽 저장

2. 악성코드의 역사

가. 악성코드 발전 과정

컴퓨터 바이러스의 가능성은 컴퓨터 초창기부터 대두 되었다. 폰노이만(Von Neumann)은 1949년 자신의 논문인 “Theory and Organization of Complicated Automata”를 통해 자기복제 코드의 이론적인 존재 가능성을 언급하였다.

1950년대 후반, 영국 수학자 Lionel Penrose는 “Self-Reproducing Machines”라는 리포트를 발표하였다. 리포트는 자신을 복제하고 변화하며, 컴퓨터 시스템을 공격하는 단순한 2차원 모델에 대한 개략적인 내용이었다. 해당 모델은 Frederick G. Stahl에 의해서 IBM 650 시스템에 포팅 되었다. 당시 과학자들과 연구원들은 인공 지능과 로봇 공학에 관심을 가지고 있었다.

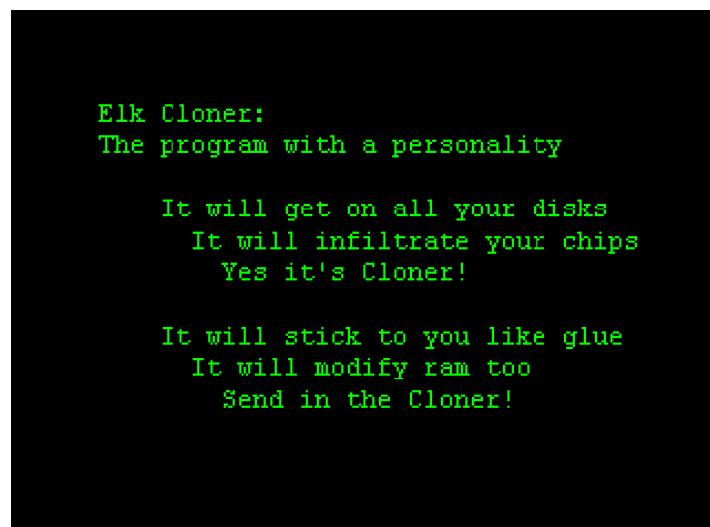
몇 년 후, 벨 연구소에서 Darwin이라는 게임을 통해 다음과 같은 실험을 하였다. 게임 내용은 다음과 같다. 각각 플레이어는 게임 안에서 정의된 함수들을 이용해서 프로그램을 만든다. 만들어진 프로그램은 메모리 영역을 검사하여 다른 플레이어의 프로그램이 사용하고 있는 메모리 영역을 찾으며, 이를 발견하면 해당 프로그램을 종료시키고 자신의 프로그램을 복사해서 최종적으로 자신의 프로그램만 메모리에 존재하도록 만든다. 게임은 악의적인 목적으로 작성된 것은 아니었지만 기존과는 다른 새로운 목적의 자기 복제 소프트웨어의 시초로 여겨진다.

1970년대 초반, ARPANET에 TENEX 시스템을 겨냥한 최초의 크리퍼웜(Creeper worm)이 등장하게 된다. 이 웜은 모뎀을 통해서 원격 시스템을 감염시켰다. 크리퍼는 ARPANET 내부에서 만들어졌고 실험 목적이었기 때문에 악의적인 행동은 하지 않았다. 하지만 단시간 내에 모든 TENEX 네트워크를 감염시켰다. 크리퍼를 삭제하기 위해 익명의 프로그래머가 리퍼(Reaper)라는 바이러스를 만들었는데, 리퍼는 크리퍼와 마찬가지로 네트워크 내부를 이동하면서 크리퍼를 찾아서 삭제를 했다. 리퍼

는 최초로 발견된 바이러스이며, 최초의 안티바이러스 프로그램으로서 악성코드 역사에 있어서 중요한 사건으로 여겨진다.

1974년 래빗(Rabbit) 이라는 새로운 바이러스가 등장했다. 래빗은 빠른 속도로 자신을 복제하고 전파되는 특징을 가지고 있었고 과거 발견된 악성코드와는 다르게 시스템의 성능을 크게 감소시켰고 결과적으로 시스템을 다운시켰다. 현재 이 바이러스가 실제로 악의적인 행동을 위해서 만들어졌는지 아니면 실험 용도로 만들었다가 통제를 벗어난 것인지 알려지지 않았다.

1981년 Apple II 컴퓨터에서 개인용 컴퓨터에선 최초로 악성코드가 발견되었다. 이는 엘크 클로너(Elk Cloner)란 부트 섹터 바이러스로 당시 고등학생이었던 리처드 스크렌타(Richard Skrenta)가 제작한 바이러스이다. 이 바이러스는 플로피 디스크를 통해 전파되는데 엘크 클로너에 감염된 컴퓨터에 플로피 디스크를 삽입 할 경우 해당 플로피 디스크에 바이러스의 복사본을 만드는 형태로 전파된다.



```
Elk Cloner:
The program with a personality

It will get on all your disks
It will infiltrate your chips
Yes it's Cloner!

It will stick to you like glue
It will modify ram too
Send in the Cloner!
```

(그림 2-1) Elk Cloner가 출력하는 텍스트

1980년대 중반부터 최초의 MS-DOS 바이러스인 브레인 바이러스

(Brain virus)를 시작으로 변종 바이러스의 모방이 유행하기 시작했다. 본격적인 악성코드들이 제작, 전파되어 세계각지에서 악성코드에 의한 피해가 속출하기 시작했다. 그와 함께 바이러스를 진단하는 프로그램도 제작되기 시작했는데, 초기에는 대부분 공개 소프트웨어나 셰어웨어(Shareware)로 제공됐다.

1990년대에는 안티바이러스 프로그램이 만들어지면서 바이러스를 탐지·치료하기 위한 움직임이 활발해지면서 이에 대응하여 악성코드 작성자들은 자신들만의 커뮤니티를 형성하기 시작하였다. 안티바이러스 프로그램으로부터 악성코드들을 지키기 위해 코드의 일부를 암호화하는 기술이 가장 처음으로 제안되었다. 이는 당시 문자열 검색을 기반으로 악성코드를 탐지하는 안티바이러스 프로그램의 탐지 과정을 무력화 할 수 있는 방법이었다. 이 후 안티바이러스 프로그램들이 암호·복호화 알고리즘 자체를 탐지 대상으로 이용하면서 새로운 자가보호 기법이 필요하게 되었다. 이에 따라 1990년에 최초로 다형성 기법을 사용한 바이러스가 등장하였다. 최초의 다형성 기법을 이용한 바이러스는 V2PX이다. 다형성 기법을 사용한 바이러스들은 시스템을 감염시킬 때마다 자신의 암호화 부분을 변경하여 백신 프로그램의 탐지를 피해간다. 이후 다형성 바이러스들은 지금까지도 악성코드 제작자들이 백신프로그램의 탐지를 피하기 위해 사용하는 방법이며 이 밖에도 여러 가지 방법이 동원되고 있다.

1994년에 광학디스크의 일종인 CD의 사용이 증가하면서 저장매체를 이용한 악성코드 전파가 활발히 진행되었다. 당시 비교적 큰 용량을 가진 이동식 저장매체인 CD는 시스템 간 데이터를 이동에 적극 사용되었다. 이에 따라 악성코드에 감염된 파일들 또한 CD를 통해 전파되기 시작하였는데, 읽기전용 저장매체인 CD의 특성상 악성코드가 발견되더라도 치료할 수 없다는 점 때문에 크게 문제가 되었다.

2005년 무렵부터는 악성코드 제작 목적이 불순해져 금전적인 이익을 위해 악성코드가 제작되기 시작한다. 이에 따라 시스템을 손상시키거나 데이터를 삭제하는 등 악성행위가 발생하는 즉시 사용자가 인식할 수 있는 유형의 악성코드들의 비중이 줄어들고, 시스템내에 숨어들어 사용자

가 인식하지 못하도록 정보 유출을 수행하는 종류의 악성코드들이 늘어나기 시작했다.

안철수 연구소에서 조사한 1986년 이후 누적 악성코드 수의 추정치를 그래프로 나타내면 다음 [표 2-2]와 같다.

[표 2-2] 누적 악성코드 수 추정치

구분	누적 악성코드 수	구분	누적 악성코드 수
1986년	1개	1998년	1만6천~4만개
1987년	3개	1999년	4만4천개
1988년	6~10개	2000년	4만8천개
1989년	30~140개	2001년	5만5천개
1990년	100~500개	2002년	5만5천~6만2천개
1991년	360~1,100개	2003년	7만개
1992년	1,100~2,400개	2004년	12만개
1993년	2,400~3,600개	2005년	14만~20만개
1994년	3,600~5,600개	2006년	22만~36만개
1995년	5,500~7,700개	2007년	50만~600만개
1996년	7,700~11,000개	2008년	150만~1500만개
1997년	1만~1만6천개	2009년	300만~3000만개

나. 악성코드 주요 사건

브레인 바이러스의 출현 이후, 악성코드의 증가 및 진화와 함께 사람들의 큰 주목을 모았던 사건들이 존재해왔다. 이 중에는 새로운 유형의 악성코드가 발견된 사건도 있고, 슬래머(SQL Slammer) 웜과 같이 전 세계적으로 막대한 피해를 끼친 사건도 있다. 다음의 [표 2-3]은 1986년 이후의 악성코드 관련 주요 사건들을 시간 순서대로 나타낸 것이다.

[표 2-3] 악성코드의 주요 사건

연도	주요 사건
1986년	최초의 MS-DOS 바이러스 '브레인(Brain)' 출현
1987년	비엔나 바이러스 등 초기 바이러스 등장
1988년	브레인 바이러스가 전 세계에 전파 최초의 인터넷 웜인 모리스(Morris) 웜 출현
1990년	최초의 다형성 기법 사용 바이러스 'V2PX' 발견
1991년	연결형 바이러스 Dir-II 바이러스 발견
1992년	미켈란젤로 바이러스 발견
1995년	MS 오피스를 이용한 매크로 바이러스 발견
1998년	윈도우 운영체제의 백도어인 백 오리피스 (Back Orifice) 등장
1999년	I-Worm/Happy 등 메일로 전파되는 웜 등장
2000년	Visual Basic 스크립트를 이용한 VBS/Love_Letter 바이러스 전파
2001년	코드레드 웜의 전파
2003년	슬래머 웜에 의한 1.25 인터넷 대란
2004년	악성 IRC봇 변형의 대량 양산

(1) 1986년 최초의 MS-DOS 바이러스 '브레인(Brain)' 출현

파키스탄의 엠자드 알비, 배시트 알비 형제가 자신들이 만든 프로그램이 불법복제 되는 것을 막기 위해 제작한 최초의 MS-DOS 바이러스, 브레인 바이러스가 출현하였다. 이 바이러스는 알비 형제가 만든 소프트웨어를 불법 복제하는 경우 5.25인치 플로피 디스크의 볼륨라벨을 '(c)Brain'으로 바꿔버리는 기능이 있다. 브레인 바이러스는 한국에서도 최초로 발견된 바이러스이다.

(2) 1987년 세계 각지에서 바이러스가 발견되기 시작

비엔나(Vienna) 바이러스, 예루살렘(Jerusalem) 바이러스 등이 세계 각

지에서 발견 되었다. 이 당시 도스에서 활동하는 바이러스만 10여종이었다.

(3) 1988년 Brain 바이러스의 전 세계 전파, 모리스(Morris) 웜 출현

브레인 바이러스가 전 세계로 퍼지고, 일반 사용자에게도 바이러스의 존재가 알려졌다. 그와 함께 여러 나라에서 바이러스를 탐지하고 치료할 수 있는 프로그램들을 제작하기 시작했다.

11월에는 인터넷을 통해 퍼진 최초의 웜인 모리스 웜이 출현하였다. 이 웜의 제작자인 로버트 모리스(Robert T. Morris Jr)는 코넬대학에서 이 웜을 만들어 MIT에서 11월 2일에 배포하였다. 모리스 웜은 자동으로 정부의 ARPANET에 연결된 모든 시스템에 복제되는 기능을 가지고 있었다. 약 6000대의 유닉스 시스템이 이 웜에 감염 되었으며 이 웜에 감염되면 기계가 느려져 결국 사용할 수 없는 상태가 되었다. 이 웜의 영향으로 컴퓨터 통신망을 해커로부터 보호하기 위해 결성된 통신보안 전문가 그룹인 CERT가 조직되었고 로버트 모리스는 법적인 처분을 당하게 된다. 현재 로버트 모리스는 MIT에서 교수로 재직 중이다.

(4) 1990년 다형성 기법의 바이러스 등장

1990년 1월 최초로 다형성 바이러스인 V2PX 바이러스가 발견되었다. 다형성 바이러스는 감염될 때마다 자신의 모습을 변형시키는 바이러스로 이전의 단순한 암호화 바이러스의 단점을 보완하여 악성코드의 탐지를 보다 어렵게 만든 악성코드이다. 기존의 암호화 바이러스는 암호화된 코드를 복호화 하기 위한 암·복호화 알고리즘이 일정한 형태를 보여 이를 이용한 악성코드 탐지가 가능하였다. 하지만 다형성 기법을 적용한 악성코드의 경우 암·복호화 알고리즘 또한 감염 시마다 변형시켜 안티바이러스 프로그램으로 진단하기 어렵다.

(5) 1991년 연결형 바이러스 Dir-II 발견

불가리아에서 제작한 최초의 연결형 바이러스인 Dir-II 바이러스가 발견되었다. 연결형 바이러스는 컴퓨터 내의 프로그램의 위치 정보를 수정하여 바이러스에 감염된 실행파일의 경로로 교체하여 사용자가 원하는 프로그램을 실행하더라도 바이러스에 감염된 실행파일을 실행하도록 유도하는 유형의 바이러스를 의미한다.

(6) 1992년 미켈란젤로(Michelangelo) 바이러스 발견

1992년 1월 미켈란젤로 바이러스가 처음 발견되었다. 이 바이러스는 3월 6일이 동작하도록 설계되어 있으며 이 날이 되면 해당 바이러스에 감염된 시스템의 하드디스크를 파괴하는 유형의 악성코드이다. 미켈란젤로 바이러스에 의한 파일 파괴에 대한 우려가 확산되면서 언론을 통해 대대적으로 알려져 ‘미켈란젤로 바이러스 신드롬’이라는 말이 생기기도 하였다. 적극적인 홍보와 이를 탐지·치료 가능한 안티바이러스 프로그램의 배포로 실제 미켈란젤로 바이러스에 의한 피해는 크지 않았다. 하지만 바이러스에 의해 개인이나 기업이 지닌 정보들의 파괴로 인해 위험성에 대해 사람들이 인지하게 되는 계기가 되었다

(7) 1995년 매크로 바이러스 출현

매크로 바이러스 또는 스크립트 바이러스는 실행 가능한 스크립트 언어를 이용하여 만들어진 바이러스이다. 스크립트 언어로 만들어진 바이러스는 주로 이를 실행할 수 있는 다른 응용프로그램을 이용하여 실행되는데, 대표적인 응용프로그램이 마이크로소프트사의 오피스 프로그램들이다. 이러한 매크로 바이러스는 응용프로그램의 데이터 파일에 저장되어 있으며 이를 실행하는 순간 자동적으로 실행된다. 이러한 매크로 바이러스는 1997년부터 전 세계적으로 유포되었다.

(8) 1996년 최초의 엑셀 매크로 바이러스 ‘라루 바이러스(Laroux virus)’ 발견

1996년 7월, 윈도우 환경에서 동작하는 최초의 엑셀 매크로 바이러스인 ‘라루 바이러스(Laroux virus)’가 발견되었다. 라루 바이러스는 워드 베이직이 아닌 엑셀의 VBA(Visual Basic for Application)로 작성되어 전 세계에 매크로 바이러스의 심각성을 인식시키게 되었다. 워드 베이직은 각 나라 언어별로 명령어가 달랐다. 그래서 스페인어로 작성된 바이러스는 우리나라에서 동작하지 않는 특징이 있었다. 하지만 VBA로 작성된 바이러스는 언어가 다른 오피스에서도 활동이 가능했기 때문에 전 세계로 퍼져나가게 되었다.

(9) 1998년 백 오리피스(Back Orifice) 등장

백 오리피스는 백도어 악성코드의 대표적인 사례이다. 해커그룹 Cult Of The Dead Cow가 만든 백 오리피스는 윈도우 운영체제를 공격 대상으로 한다. 백 오리피스는 마이크로소프트사의 윈도우 운영체제 시스템을 원격에서 조절할 수 있게 만드는 악성코드이다. 이 프로그램을 이용하면 윈도우 운영체제 환경의 시스템의 중요 정보를 조작하거나 파괴할 수 있게 된다.

(10) 1999년 전자우편으로 전파되는 웜 등장

I-Worm/Happy, I-Worm/ExploreZip 등의 웜이 1999년 등장하였는데, 이 들은 전자우편을 통해 전파될 수 있었다. 악성코드가 전자우편을 통해 전파되면서 기존의 악성코드에 비해 전파시간이 혁신적으로 짧아졌으며 전파 범위도 광범위해졌다. 연말에 보고가 된 버블보이 웜은 전자우편을 읽는 것만으로도 시스템을 감염시킬 수가 있었다. 이 경우 사용자

가 전자우편에 첨부된 파일을 실행시키는 등 2 차적인 행위를 요구하지 않기 때문에 높은 전파력을 갖는다.

(11) 2000년 비주얼 베이직 스크립트를 이용한 바이러스 전파

5월 4일, 일명 Love Bug로 알려진 VBS/Love_Letter 바이러스가 전 세계에 퍼졌다. 이 바이러스는 비주얼 베이직 스크립트로 만들어졌으며, 전자우편을 통해 전파되는 웜의 일종으로 전자우편 주소록에 등록된 모든 사용자에게 바이러스가 감염된 전자우편을 보내기 때문에 전파 속도가 빠르다. 이 바이러스는 빠른 시간에 전 세계에 퍼졌고, 이후 비주얼 베이직 스크립트로 작성된 수많은 변형 바이러스가 등장하였다.

(12) 2001년 코드레드(Codered) 웜의 전파

코드레드 웜은 마이크로소프트사의 윈도우 운영체제를 사용하는 인터넷 서버를 감염대상으로 한다. 코드레드 웜은 목표 시스템을 감염시킨 이후 일정 시간동안 잠복해 있다. 이 후 정해진 시간이 되면 공격행위를 수행한다. 이 때 감염된 서버는 속도가 느려지거나 작동이 멈추는 등 비정상적으로 동작한다. 2001년 7월 첫 발생 후 8시간 만에 25만대 이상의 컴퓨터를 감염시켰다. 공격 대상 된 백악관은 2001년 7월 19일 인터넷 주소를 변경하고, 사이트를 폐쇄하는 조치를 취하기도 했다.

(13) 2003년 슬래머(SQL Slammer) 웜에 의한 1.25 인터넷 대란

윈도우 2000의 SQL 서버 취약점을 이용한 슬래머 웜은 2003년 1월 25일 국내를 비롯하여 전 세계의 인터넷을 마비시켜 엄청난 피해를 입혔다. 슬래머 웜은 감염된 호스트를 이용하여 초당 1만 ~ 5만개의 UDP 패킷을 생성하여 인터넷으로 전송하였다. 또한 적극적으로 악성코드를 전파하여 추가적인 감염 시스템을 양산하였다. 그 과정에서 슬래머 웜에

감염된 호스트들은 과부하를 일으켰으며, 그로 인해 결과적으로 서비스 방해(DoS, Denial of Service) 현상을 겪게 된다.

(14) 2004년 악성 IRC봇 변형의 대량 양산

IRC봇은 IRC 서비스를 이용한 악성코드로 악성코드의 행위가 정해진 알고리즘에 따라 정적으로 이루어지기 보다는 공격자의 입력에 따라 동적으로 동작하는 유형의 악성코드이다. 악성코드가 설치될 경우 해당 악성코드는 미리 정해진 IRC 서버의 특정 대화방에 접속하여 공격자의 명령을 기다린다. 공격자는 IRC 서버를 이용하여 개별 IRC봇에게 명령을 내려 각종 악성행위를 수행한다. 주로 애드웨어 설치와 스팸메일 발송, 그리고 분산 서비스 방해 공격(DDos)등을 수행한다.

(15) 2005년 소니 BMG의 DRM 관련 루트킷 이용한 브레프리트봇(Win-Trojan/Breplibot) 트로이목마 발견

소니 BMG는 불법 복제를 방지하기 위해 음반 CD에 DRM(Digital Rights Management)을 함께 담아 판매했다. 해당 DRM은 커널모드 루트킷을 사용하여 자신이 실행중인 것을 숨기는 기능을 사용했다. 사용자도 모르게 사용자의 시스템에서 수행되는 DRM 기능이 알려지면서 소니는 큰 논쟁이 휩싸이게 된다. 11월, 소니 음반 DRM에서 보호하는 파일 명으로 위장한 브레프리트봇(Win-Trojan/Breplibot) 트로이 목마가 소니 DRM 루트킷 사건이 알려진지 얼마 되지 않은 시점에서 발견됐다.

(16) 2008년 MBR에 감염되는 새로운 형태의 악성코드인 트로이목마 Win-Trojan/Mbrookit이 발견

2007년 12월과 2008년 1월에 마스터 부트 레코드(Master Boot Record)를 감염시키는 새로운 형태의 트로이 목마인 Win-Trojan/Mbrookit이 발

전됐다.

(17) 2010년 스텍스넷(Stuxnet)이 알려짐

특정 원격감시제어(SCADA, Supervisory Control and Data Acquisition) 시스템을 대상으로 공격을 수행하는 스텍스넷이 출현하면서 발전소 등 국가기반시설에 대한 사이버위협이 현실화 되었다. 스텍스넷은 독일 지멘스사의 PCS7 시스템상의 PLC 코드를 변경하여 오작동을 유도함으로써 관련 원격감시제어 시스템에 대한 공격과 파괴를 유발하는 악성코드이다. 스텍스넷의 등장은 악성코드의 새로운 패러다임 시대의 개막을 의미한다. 즉, 자기 과시나 금전적인 이득을 목적으로 악성코드가 제작되었던 지금까지의 패러다임에서 이제는 사회 핵심 시설의 파괴만을 목표로 제작하는 패러다임으로 바뀐 첫 사례로 볼 수 있다.

3. 악성코드 종류

악성코드는 자기 복제 능력과 감염 방법, 혹은 증상에 따라 바이러스, 웜, 트로이목마 등으로 분류된다. 악성코드의 종류는 아래와 같다.

가. 컴퓨터 바이러스(Computer virus)

컴퓨터 바이러스는 컴퓨터 시스템에 침투하여 숙주 컴퓨터의 프로그램이나 파일을 변형시키고, 자기 자신 또는 자신의 변형을 복사하여 또 다른 대상을 감염시킴으로써 컴퓨터 시스템과 파일을 파괴하는 프로그램이다. 주로 전자메일, 매크로, 인터넷 웹페이지, 그리고 USB 메모리 등을 통해서 전파된다.

컴퓨터 바이러스는 컴퓨터의 비정상적인 동작을 유발하고, 시스템의 성능 저하 등에 영향을 미친다. 대표적인 사례로는 브레인 바이러스, 미켈란젤로 바이러스 등이 있다.

나. 웜(Worm)

프로그램 코드 자체를 스스로 복제할 수 있는 컴퓨터 프로그램이다. 자기복제가 가능하다는 점에서는 바이러스와 비슷하지만 바이러스가 다른 파일을 공격해서 거기에 붙어 다니는데 비하여 웜은 파일과는 독립적으로 실행되며 다른 프로그램을 필요로 하지 않는다는 점에 다르다. 또한 웜은 사용자와의 상호작용 없이 컴퓨터 시스템의 취약점을 공격하여 그 자체만으로도 네트워크를 통해 전파될 수 있다. 따라서 웜은 거의 30분 내로 전 세계의 컴퓨터를 감염시킬 수 있다.

컴퓨터 바이러스와 마찬가지로 컴퓨터 자체의 성능에도 영향을 미치지만 네트워크를 손상시킬 수도 있다. 웜의 대표적인 사례로는 모리스 웜, 코드레드 웜, 그리고 슬래머 웜 등이 있다.

다. 트로이목마(Trojan horse)

그리스 신화의 트로이 전쟁에서 사용된 목마처럼 트로이 목마 프로그램은 겉보기에는 유용한 프로그램처럼 보이지만 실제로는 해킹 기능을 가진 악성 프로그램을 말한다. 트로이 목마는 사용자 몰래 컴퓨터의 정보를 외부로 유출하거나 원격제어가 가능하도록 만든다. 트로이 목마는 바이러스나 웜처럼 복사 기능은 없기 때문에 스스로 다른 파일이나 컴퓨터를 감염시키지는 못하고 주로 사용자가 인터넷에서 다운로드한 파일을 통해 전파되어 사용자가 실행시키도록 유도한다.

사용자의 개인 정보 유출 등의 피해뿐 아니라 시스템 파일을 변경하거나 파괴할 수 있다. 심각한 경우 시스템이 마비되기도 한다. 대표적인 트로이목마 프로그램으로는 Setiri, Hydan 등이 있다.

라. 백도어(Backdoor)

백도어는 원래 시스템의 유지 보수나 유사시의 문제 해결을 위하여 시스템 관리자가 보안설정을 우회하여 시스템에 접근할 수 있도록 만든 도구이다. 하지만 최근에는 악의적인 목적을 갖는 공격자들이 시스템에 재침입이 용이하도록 이용하는 도구를 의미한다. 백도어 프로그램은 비(非)인가된 접근을 허용하는 프로그램으로 공격자가 이후 사용자 인증 과정 등 정상적인 절차를 거치지 않고 프로그램이나 시스템에 접근할 수 있도록 지원한다. 공격자는 시스템에 침입한 이후 재접속을 위해 백도어를 설치하기도 하지만, 프로그래머가 관리 목적으로 만들었다가 제거하지 않은 백도어를 찾아 악용하기도 한다.

백도어는 특정 포트를 오픈하여 공격자가 침입할 수 있도록 백그라운드로 실행되며 트로이목마 같은 악성코드를 통해 감염될 수 있다. 대표적으로 백 오리피스가 있다.

마. 스파이웨어/애드웨어(Spyware/Adware)

원래는 미국의 광고회사인 라디에이트(Radiate)가 광고를 보고 있는지를 알아보기 위해 개발한 도구였다. 하지만 최근에는 그 목적이 변질되어 사용자를 특정 사이트에 접속하도록 유도하거나 사용자의 개인 정보를 유출하기 위한 목적으로 설치된다. 보통 인터넷에서 무료로 공개되는 소프트웨어를 다운받거나 특정 웹사이트를 접속할 때 함께 설치된다.

주기적으로 특정 사이트에 접속하게 하거나 홍보 배너 출력하며 심한 경우에는 사용자 컴퓨터의 입출력 내용까지도 수집하여 전송하기도 한다. 다른 악성코드들에 비해 시스템에 치명적인 영향을 주지 않지만 악의적인 목적으로 악용될 가능성이 존재한다.

바. 악성 봇(Malicious Bot)

악성 봇은 감염된 컴퓨터에서 일반 프로세스처럼 존재하지만 스스로 움직이지 않고 공격자가 원격으로 제어할 수 있게 만드는 악성코드이다.

공격자의 명령에 따라 스팸 메일을 전송하게 하거나 분산 서비스 방해 공격(DDoS), 또는 정보 유출을 수행한다.

전자메일이나 웹페이지, 악성코드를 통해 감염이 되며, 대표적인 악성 봇으로는 Bagle, MyDoom 등이 있다.

사. 루트킷 (Rootkit)

루트킷은 전통적인 UNIX 시스템에서 관리자 계정을 뜻하는 root와 소프트웨어 컴포넌트를 뜻하는 kit의 합성어로서, 컴퓨터의 관리자 권한을 유지하고 자신의 존재를 운영체제 또는 다른 프로그램으로부터 숨기는 악성코드 유형을 의미한다.

공격자는 대상 시스템의 관리자 권한을 획득한 이 후 루트킷 설치를 통해 해당 시스템에 악성코드가 관리자 권한을 유지할 수 있도록 만든다. 루트킷은 펌웨어 또는 커널 등을 목표로 하여 운영체제의 구동 전에 특정 기능이 활성화 되는 특징을 갖는다. 즉, 루트킷의 경우 운영체제 보다 먼저 실행되거나 운영체제의 기능을 일부 제한할 수 있어 안티바이러스 프로그램이나 기타 다른 탐지 도구를 통해 자기 자신을 검사하는 것을 저지할 수 있다. 특히 루트킷이 커널 내부에 위치하고 있는 경우, 루트킷의 제거가 거의 불가능하다. 따라서 치료가 불가능하여 운영체제의 재설치가 필요하다.[5]

아. 크라임웨어 (Crimeware)

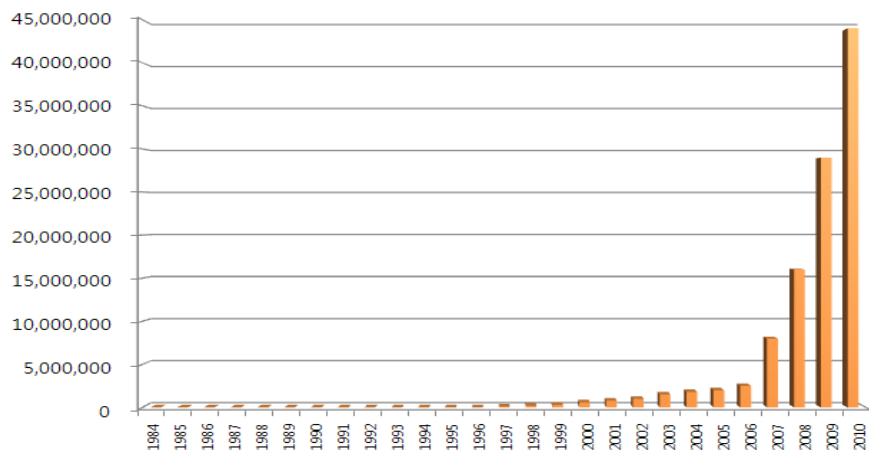
크라임웨어는 사이버범죄를 목적으로 만들어진 악성코드를 뜻한다. 크라임웨어는 사용자의 금융정보를 유출하여 현금 계좌 인출 또는 계좌 이체 등을 수행하거나 메신저 등을 이용한 피싱 행위를 위한 악성코드이다. 크라임웨어는 공격자의 금전적 이익을 위해 동작한다는 점에서 스파이웨어, 애드웨어, 등 다른 악성코드와 차이가 있다.

4. 악성코드 동향

가. 악성코드 피해 동향

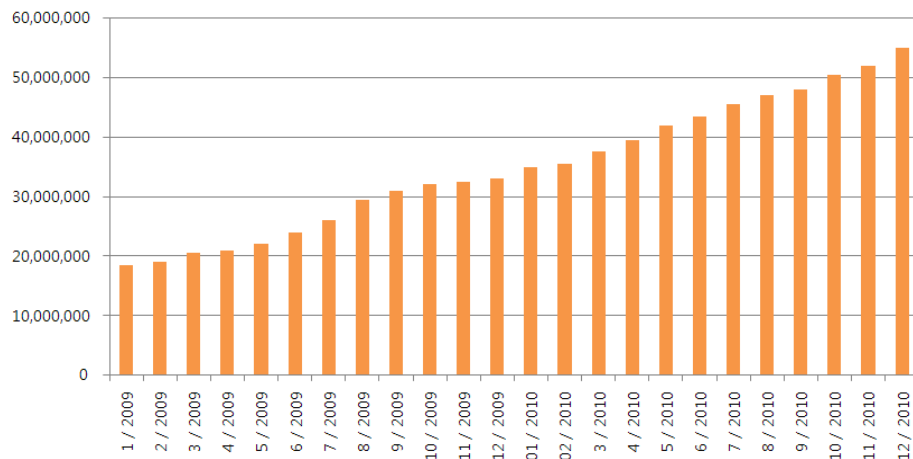
최근 악성코드는 공격 기술과 전파 기술이 고도화, 지능화, 자동화되어 인터넷 사용자들을 위협하고 있다. 예전에는 악성코드 개발자가 호기심 혹은 자기 과시를 목적으로 악성코드를 제작하였으나 최근에는 금전적 이득 및 정치적인 목적을 달성하기 위해 악성코드를 제작하는 경우가 많아 졌다. 그 결과 악성코드로 인한 피해의 규모가 광범위해지고 있다. 악성코드의 개체 수는 시간이 지남에 따라 기하급수적으로 늘어나고 있으며, 다양한 기술들을 접목하여 악성코드 탐지를 우회하거나 악성코드 분석을 방해 혹은 지연할 수 있는 악성코드가 늘어나고 있다.

안티바이러스 프로그램들을 테스트하는 독일의 AV-test는 자사의 2010년 보고서를 통해 지난 1984년부터 2010년까지의 발견된 악성코드 개수가 약 4,400만 개라고 발표하였다. AV-test에서는 지난 2009년 7월 보고서에 누적 악성코드의 수는 약 2천100만 개라고 보고하였으며, 이듬해인 2010년 4월에는 누적 악성코드가 3천만 개를 돌파하였다고 밝혔다[26].



(그림 2-2) 1984-2010년 동안 발견된 누적 악성코드 수

급격히 증가하는 새로운 악성코드로 인해 악성코드 검출을 위한 시그니처의 수도 크게 증가하고 있다. 미국의 안티바이러스 회사인 맥아피의 분기 보고서 “McAfee Threats Report : Fourth Quarter 2010”에 따르면 맥아피가 소유한 악성코드 시그니처의 수는 약 5,500만 개라고 발표하였다[10]. 안티바이러스 회사인 시만텍은 이러한 악성코드가 급증하는 원인으로 악성코드 제작을 위한 사전지식이 없는 일반 사용자들도 악성코드 제작 툴을 사용하여 변종의 악성코드를 손쉽게 제작할 수 있기 때문이라고 분석하고 있다. 아래 (그림 2-3)는 맥아피가 보유하고 있는 악성코드 판별을 위한 시그니처의 수를 나타내고 있다.

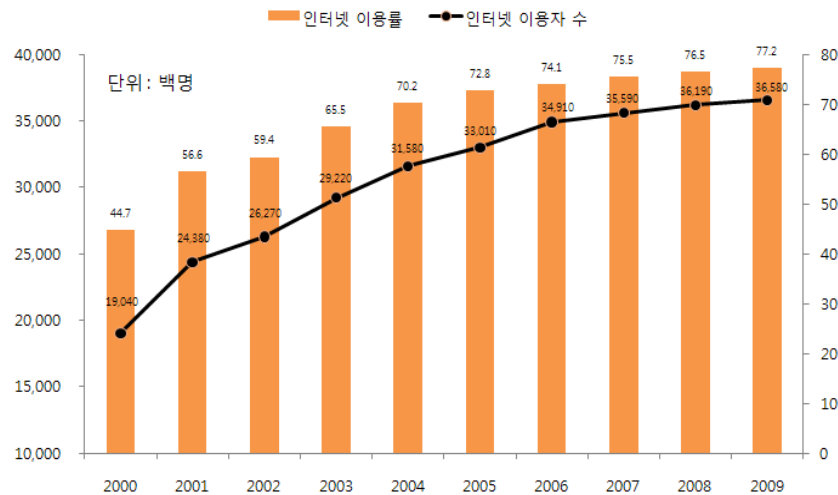


(그림 2-3) 맥아피가 보유한 시그니처 수

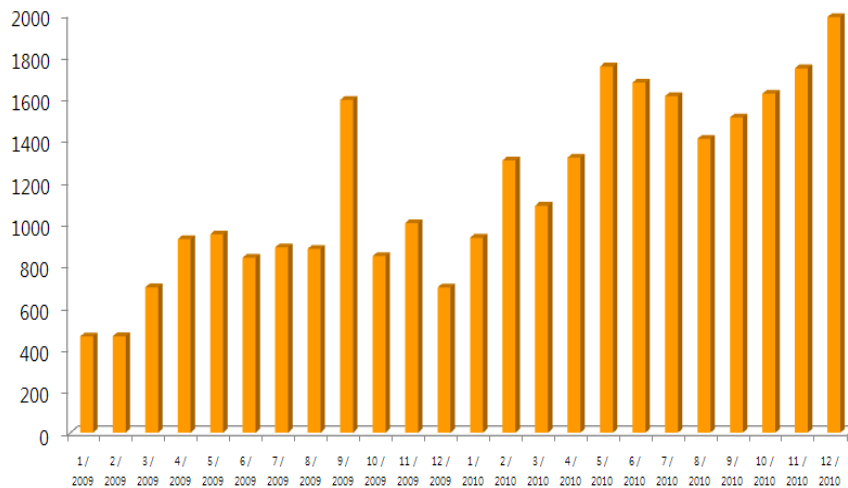
(1) 국내 악성코드 피해 동향

국내 인터넷은 1982년 SDN(TCP/IP)구축을 시작으로 다양한 정책을 통하여 빠른 성장을 거듭해 왔다. 1994년 사용 ISP(Internet Service Provider)의 등장과 1997년 전용회선 서비스의 시작으로 1999년 인터넷 이용자수가 1,000만 명이 넘었으며, 2009년에는 ITU 정보통신발전지수

(ICT-Development Index)가 세계 2위를 기록하였다. “2010 국가정보화백서”에 따르면 만 3세 이상 인터넷 이용자 수는 약 3,600만 명으로 이는 만 3세 이상 국내 인구의 77.2%에 달하는 수치이다.



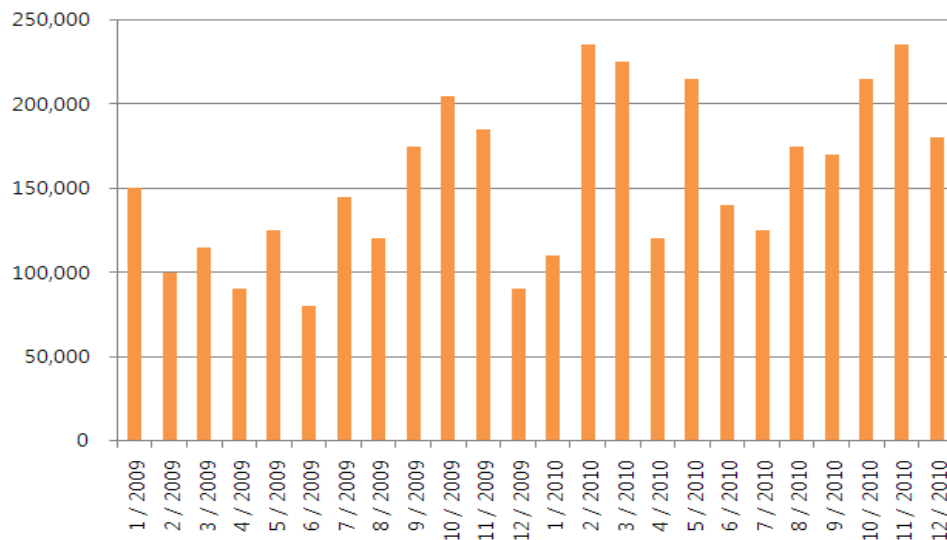
(그림 2-4) 국내 인터넷 이용률 및 이용자 수 현황



(그림 2-5) 2009-2010년 월별 국내 악성코드 발생건수 추이

국내 인터넷 이용률이 증가함에 따라 악성코드의 위협 또한 비례하게 증가하고 있다. 한국인터넷진흥원(KISA)의 “인터넷 침해사고 동향 및 분석월보”에 따르면 2010년 웹·바이러스의 신고 건수는 17,930건으로 전년 대비(10,395) 약 72%의 증가율을 보였다. 아래 (그림 2-5)는 2009년, 2010년 동안 국내에서 발생한 악성코드 발생건수를 나타내고 있다.

국내의 경우 인터넷 이용자들의 인터넷 사용목적인 ‘인터넷 구매 및 판매’, ‘게임’, ‘커뮤니티’와 같이 사용빈도가 높은 환경에서 작용하는 악성코드들이 대량으로 유포되고 있다. 국내 안티바이러스 회사 안철수 연구소의 보고서 “ASEC report”에 따르면 국내에서 많은 사용률을 보이는 ActiveX의 취약점을 이용한 악성코드인 JS/Agent가 2010년 감염보고 건수가 가장 많았으며 USB 저장매체를 통해 전파되는 악성코드인 TextImage/Autorun이 감염보고가 2위, 그리고 ‘게임’의 정보를 탈취하기 위한 악성코드인 Win-Trojan/Onlinegamehack이 감염보고 건수가 3위를 기록했다[26].



(그림 2-6) 2009-2010년 월별 국외 악성코드 발생건수 추이

(2) 국외 악성코드 피해 동향

국외 안티바이러스 업체 G Data의 보고서 “Half-yearly report July-December 2010”에 따르면 2010년 하반기동안 새로 발견된 악성코드의 수는 약 100만개로 일평균 약 5,800개씩 발견되고 있다고 보고하고 있다. 또한 2010년 새로 발견된 악성코드의 수는 약 200만 개로 2009년 대비 32%증가하였으며 해당 수치는 2006년 대비 약 52배 증가한 악성코드의 발견이라고 보고하고 있다. 새로운 악성코드의 유포에 따라 악성코드의 위협 또한 증가하는 모습을 볼 수 있다. 2010년 하반기 악성코드 감염 비율은 2009년 2분기에 비하여 16% 증가한 수치이며 2010년 상반기에 비해서는 6% 증가한 수치이다.

이처럼 국외 악성코드의 증가 수치는 변종 악성코드의 유포 속도가 증가하고 있음을 나타내고 있으며, 악성코드의 위협이 국내에서만 국한된 것이 아니라 세계적인 문제인 것으로 나타난다. 이에 따라 국제간 악성코드 분석을 위한 협력이 거론되고 있으나 악성코드 분석에 관한 표준안이 없어 협력 환경 구축이 아직 미흡한 단계에 머물러 있는 수준이다.

나. 악성코드 관련 동향

근래의 악성코드들은 다양한 사회공학적 기법을 사용하여 보안이 취약한 시스템을 감염시키고 있다. 또한 사용자들이 다수 사용하는 웹하드, 금융 웹사이트 등을 해킹하여 악성코드를 업로드 하는 방식으로 사용자들은 URL을 통해 접근하는 것만으로도 악성코드에 쉽게 감염되고 있다. 또한 악성코드의 제작목적이 금전적 이득의 목적을 넘어 정부기관, 금융기관등 공신력 있는 홈페이지에 대해 서비스 방해 공격(DDoS) 공격 및 악성코드 삽입, 표적 공격(targeted attack) 등을 수행하여 사회적 혼란을 야기하는 목적으로 변형되고 있다. 최근에는 스마트폰이 발달하면서 모바일 악성코드가 증가하고 있다[4].

(1) 사회공학적 기법을 이용한 공격

사회적인 이슈, 가십이 되는 사회공학적 기법을 이용한 공격이 증가하고 있다. 2009년 마이클 잭슨 죽음 관련, 2010년 폴란드 대통령 탄핵 비행기 추락사고 관련, 유명 스포츠인 김연아 동영상 관련 등 악성코드 유포 시점을 기준으로 최신 이슈 및 가십을 이용한 공격이 증가하고 있다. 이러한 이슈들은 악성코드를 전파시키는 전파경로로 활용될 수 있으며 단시간에 많은 사용자들을 감염시킬 수 있어 빠른 서비스 방해 공격(DDoS)이 가능해 진다. 최근에는 SNS(Social Network Service)의 이용이 증가하면서 페이스북, 트위터 등의 서비스를 통해서 사회공학적 기법을 사용하고 있는 추세이다.

(2) 사용자 유도 방식

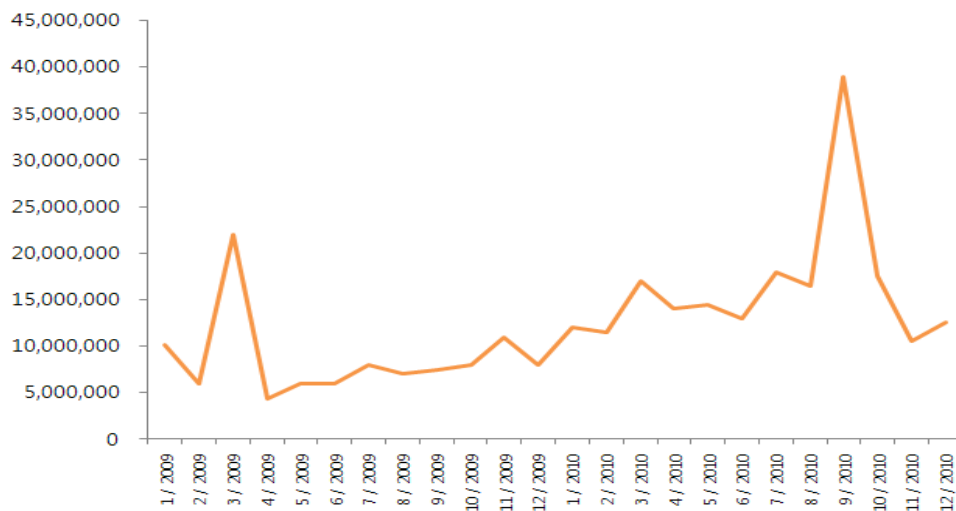
정보보안의 관심이 높아지는 가운데, 이를 악용하는 방식이 늘고 있다. 허위 안티바이러스 프로그램을 설치 시, 정상 PC를 감염 PC로 허위 경고 창을 출력하여 결제를 유도하는 방식, 시스템을 진단하는 시스템 진단 유틸리티를 가장하여 해당 시스템이 현재 문제점이 있는 것으로 위장하여 결제를 유도하는 프로그램 등이 대표적이다. 2011년에는 허위 안티바이러스 프로그램을 대량으로 유포하여 사용자에게 월 자동결제를 유도하여 40억 원을 가로챈 일당이 검거되기도 하였다. 이처럼 시스템의 취약점이 아닌 사용자들의 관심 및 신뢰를 이용한 공격방식이 증가하고 있다.

(3) 웹사이트를 통한 악성코드 삽입

정상적인 서비스를 제공하는 웹사이트가 해킹에 의해 악성코드 유포 사이트로 변질될 수 있다. 이용자가 많은 웹사이트를 공격할 경우 손쉽게 대량의 사용자에게 악성코드를 유포할 수 있어 단시간에 수많은 감염

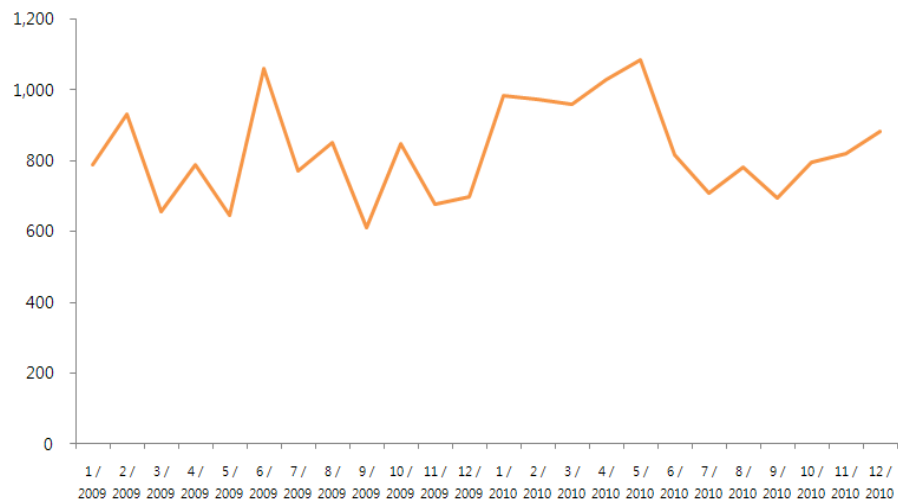
PC를 양산할 수 있다. 국외 안티바이러스 회사인 시만텍의 보고서 “Symantec Internet Security Threat Report Trend for 2010”에 따르면 웹 기반 공격 평균 횟수는 2009년에 비교하여 93%증가하였다고 발표하였다[3].

국내 안티바이러스 업체 안철수 연구소에서 발간하는 “ASEC report”에 따르면 2009-2010년 월별 악성코드가 발견된 도메인의 경우 아래 (그림 2-8)과 같다. 2010년 악성코드의 감염 도메인의 수는 2009년 9336건 대비 113% 수준인 10,528건으로 집계되었다.



(그림 2-7) 2009-2010년 국외 웹 기반 공격 평균 횟수

위의 두 보고서에 따르면 해킹당하여 해당 수정이 가해진 웹사이트는 악성코드를 유포하는 주요 경로로 사용되고 있음을 알 수 있다. 이처럼 악성코드의 전파경로가 접속이 많은 웹사이트를 통하여 전파된다면 단시간에 많은 감염PC를 만들 수 있으며, 또한 감염PC는 악성코드 전파에 중간노드 역할을 하기 때문에 피해가 상당할 것으로 추측된다.



(그림 2-8) 2009-2010년 국내 악성코드 감염 도메인 수

(4) DDoS(Distributed Denial of Service) 공격

[표 2-4] 7·7 DDoS와 3·4 DDoS의 피해 규모

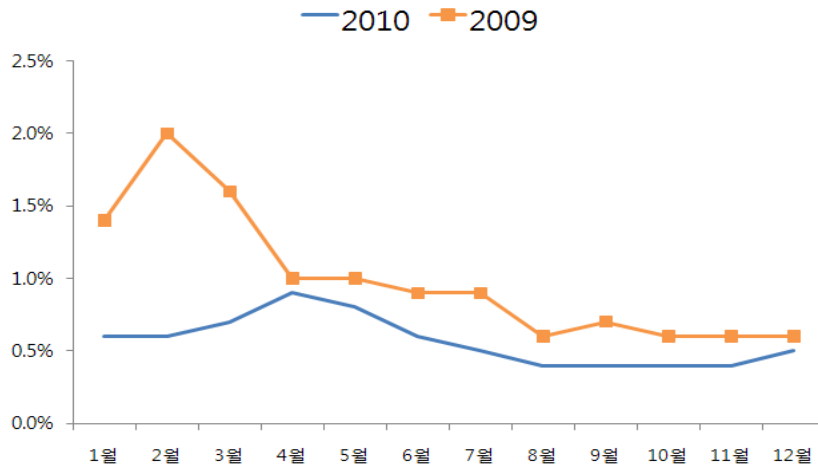
구분	7·7 DDoS	3·4 DDoS
악성코드 유포	웹하드 업체	웹하드 업체
악성코드 기능	DDoS, 파괴	DDoS, 파괴
추정 공격 목적	사회 혼란 야기	사회 혼란 야기
하드디스크 파괴 규모	1,466건	756건
유포, 명령사이트 차단 건수	538건	749건
악성 봇 규모	115,044대	116,299대

2009년 7월 7일에 발생한 7·7 DDoS 공격과 2011년 3·4 DDoS 공격 등 국내 유명 포털 회사 및 정부기관·금융기관을 대상으로 이루어진 동시

다발적인 DDoS 공격으로 인하여 사회적인 불안감을 조성하는 사태가 일고 있다. 또한 DDoS에 사용된 악성코드들은 점점 진화하여 탐지가 어려워지며 악성 행위가 복잡해지고 있다.

(5) 대량의 악성 봇(bot)감염

악성코드들은 때론 PC를 감염시켜 악성 봇을 만들어 제 2의 공격에 악용하고 있다. 악성 봇이 된 PC들은 좀비 PC라고 불린다. 좀비 PC는 시스템의 정보를 외부로 유출하거나 DDoS 공격과 같은 2차적인 공격에 참여한다. 아래 (그림 2-9)는 전 세계 PC중 악성 봇이라고 추정되는 PC 대비 국내 감염률을 나타내고 있다.



(그림 2-9) 국내 악성 봇 감염 PC 수

(6) 제로데이(Zero-day)공격

프로그램의 취약점이 발견된 이 후, 해당 프로그램에 대한 보안 패치가 이루어지기 이전에 해당 취약점을 이용하여 공격을 수행하는 악성코드들 제로데이 취약점을 이용한 악성코드로 분류한다. 과거 제로데이 공

격은 국내 웹사이트에서 사용하는 ActiveX와 인터넷 익스플로러가 주요 타겟이었다면, 최근 제로데이 공격은 대중적으로 사용하는 마이크로소프트사의 오피스 제품군과 플래시 플레이어 등으로 옮겨져 악성코드의 감염경로가 더욱 다양해 졌다.

(7) 표적 공격(Targeted Attack)

불특정 다수를 대상으로 하는 일반 악성코드와 다르게 표적 공격은 특정 기관의 정해진 정보 탈취 및 제어를 목적으로 하기 때문에 불필요한 시스템 파괴나 이상행위를 일으키지 않는다. 따라서 사용자는 자신의 시스템의 악성코드 감염 사실을 알아차리기 어렵다. 기밀 정보 탈취하거나 물리적으로 단절된 네트워크 내에서 동작하기 위해 표적 공격에 사용되는 악성코드는 높은 수준의 컴퓨터 지식을 가지고 제작될 가능성이 높다. 표적 공격이 성공적으로 이루어지기 위해서는 제로데이 취약점을 이용하거나 알려지지 않은 보안 취약점을 이용해야 하기 때문이다. 대표적인 예로 스틱스넷 같은 경우는 5 가지 제로데이 취약점을 이용한 공격 기술이 사용된 것으로 보고되었다. 이처럼 정교하고 공개되지 않은 공격 기술을 사용하기 위해선 보통 장기간의 시간이 필요하다.

(8) 모바일 악성코드

모바일 악성코드는 전파 방법 그리고 행위에서 기존 PC 기반 악성코드와 차이점이 있다. 주로 블루투스(Bluetooth)나 멀티미디어 메시지(MMS) 등을 통하여 감염되고 감염된 모바일 디바이스에서는 시스템 파괴, 가용성 저하, 금전적 피해 또는 개인 정보 유출 등이 일어날 수 있다.

(9) Advanced Persistent Threat (APT)

악성코드를 사용해서 정보를 훔치는 그룹을 Advanced Persistent Threat 라고 한다. 보통 해커 개인을 APT라고 지칭하지 않는다. APT 공격을 수행하기 위해서는 높은 수준의 공격 기술과 지속적인 공격행위를 진행할 수 있는 행동력이 필요하기 때문이다. APT 공격은 주로 특정 목표를 지속적으로 공격하는 형태로 이루어진다. 따라서 정치적 목적을 지니고 정부 기관이나 대형 기업을 대상으로 기밀문서 유출, 기간산업 방해 등을 수행한다[6].

APT는 특정 목적을 수행하기 위해 복합적인 공격 형태를 갖는다. 공격 대상이 되는 조직의 네트워크 접근을 위해 사회공학적인 공격을 통해 이용 가능한 공격 대상을 선정하고 일차적으로 대상 시스템을 감염 시켜 대상의 저장매체나 전자우편 계정, 홈페이지 등 각종 매체를 이용하여 목표한 표적에 접근한다. 이처럼 APT 공격은 전반적인 과정에서 정형화된 정적인 프로세스에 의지하기 보다는 다양한 기술들을 바탕으로 순간에 필요한 행위를 수행하는 동적인 공격 방법이다.

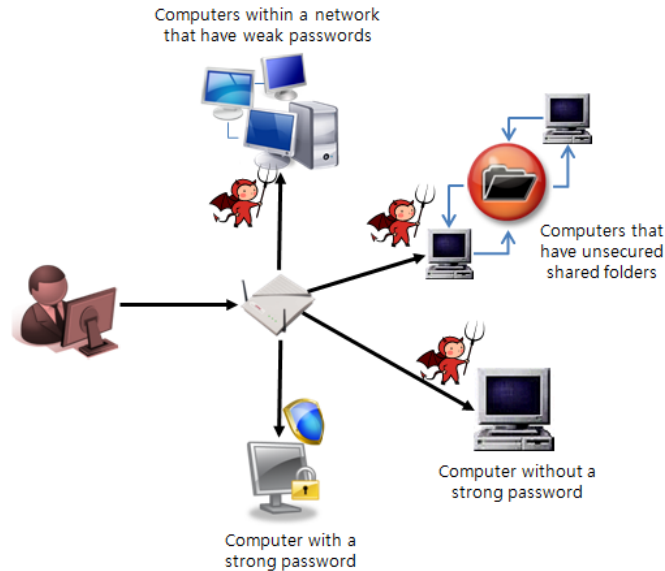
제 2 절 악성코드 공격 방법

1. 악성코드 공격기술

개발자의功名심, 호기심에 의해 제작되던 과거의 악성코드와 달리 최근 악성코드가 경제적·정치적 목적으로 제작·유포되면서 사회적 혼란을 초래하고 경제적 피해를 유발하면서 악성코드에 대한 사회 관심이 고조되고 있다. 2009년 발생한 7·7 DDoS 공격과 3·3 DDoS 공격, 허위 안티 바이러스 프로그램을 위장한 결제 유도 악성코드, 금융 사이트를 위장한 피싱사이트, 증가하는 홈페이지 은닉형 악성코드, 제로데이 취약점 이용한 악성코드의 증가 등 점점 발달하는 악성코드 공격 기술과 그로 인한 위협으로 인해 일반 사용자뿐만 아니라, 보안업체, 정부 또한 악성코드

탐지 기술에 많은 관심을 쏟게 되었다[22].

가. 컨픽커(Conficker)



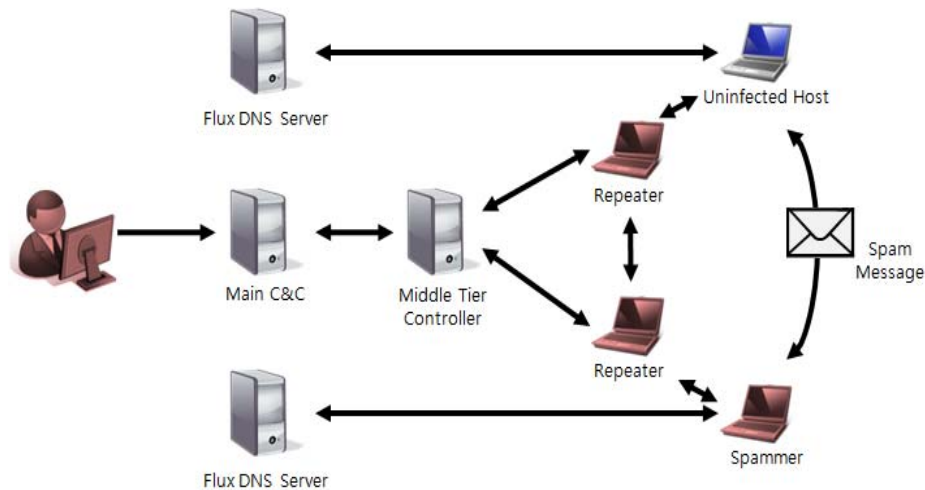
(그림 2-10) 컨픽커(Conficker) 악성코드

컨픽커(Conficker)는 2008년에 최초로 알려졌으며 다양한 형태의 변종 악성코드가 존재하는 악성코드이다. 윈도우 보안취약점, USB와 같은 이동식 저장매체, 취약한 암호를 가진 공유 폴더 등을 통하여 전파되며, 감염 플랫폼에서 특정 웹사이트를 접속하도록 만드는 악성코드이다. 컨픽커는 감염 플랫폼이 자신을 치료하지 못하도록 만들기 위해 DNS 정보 중 안티바이러스 업체 또는 마이크로소프트와 같은 특정 문자열이 들어간 홈페이지 접속을 차단하여 치료를 방해한다.

나. 웨일택(Waledac)

전자우편을 통해 전파되며, 감염된 시스템에서 스팸메일을 대량으로

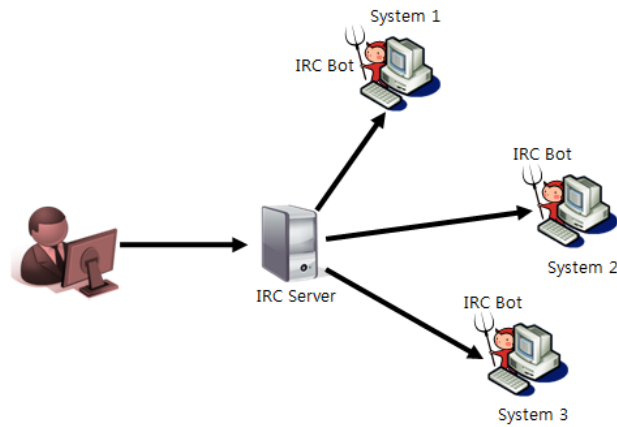
발송하여 네트워크 트래픽을 증가시키는 악성코드이다. 2월 14일 발견되기 시작했으며, 밸런타인데이에 관련 문구가 삽입된 메일제목을 사용하여 클릭을 유도 후 설치되는 특성이 있어 ‘밸런타인데이 웜’이라고도 불린다. 변종 웨일텍(Waledac) 웜의 경우 밸런타인데이뿐만 아니라 다양한 사회적 이슈와 관련된 문구를 이용하여 메일 클릭을 유도한다.



(그림 2-11) 웨일텍(Waledac) 악성코드

다. IRC봇(IRCBot)

윈도우 취약점, 네트워크 공유 폴더, 이동식 저장매체 등을 통하여 전파되며 시스템 날짜를 변경하는 악성코드이다. 시스템의 날짜를 변경함으로써 정상적인 서비스를 제공받을 수 없게 되며, 특정 IRC서버에 접속하여 백도어를 설치하기도 한다. 시스템 날짜를 2090년으로 바꾸는 ‘2090 바이러스’의 변종으로, 감염 시스템의 날짜를 2070년으로 변경해서 ‘2070 바이러스’라고도 불린다.



(그림 2-12) IRC봇(IRCBot)

라. 네이트온(NateOn)



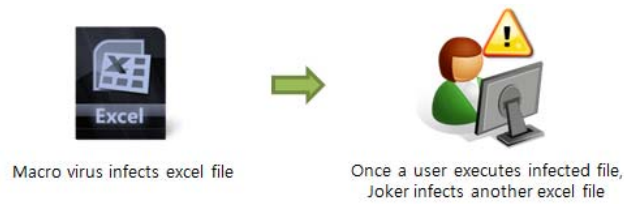
(그림 2-13) 네이트온(NateOn) 악성코드

2001년 유행하였던 'MSN 바이러스'와 유사한 형태의 악성코드로서, 국내에서 많이 사용되는 'NateOn' 메신저를 통해 전파되는 악성코드이다. 변종 유형에 따라 여러 종류의 사진을 보여주거나, 유명 온라인 게임의 게임 정보를 외부로 유출시킨다.

마. 조커(Joker)

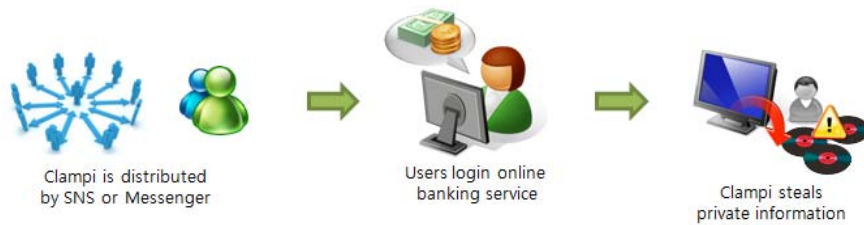
엑셀 파일을 실행할 때마다, 다른 엑셀 문서를 감염시키는 악성코드로, 특정 시간에 파일을 삭제한 것처럼 속이는 메시지를 송출, 사용자가 메시지 클릭할 경우 '뽕임'이라는 문자를 송출하는 악성코드이다. 변종 유

형에 따라 문서를 삭제하기도 한다.



(그림 2-14) 조커(Joker) 악성코드

바. 클램피(Clampi)



(그림 2-15) 클램피(Clampi) 악성코드

주로 SNS나 메신저를 통해 전파되는 악성코드로서 영어권 사용 국가의 은행을 타겟으로 하는 악성코드이다. 감염 플랫폼이 금융 사이트에 접속하여 금융정보 입력 시 해당 금융정보를 외부로 전송하는 기능을 한다.

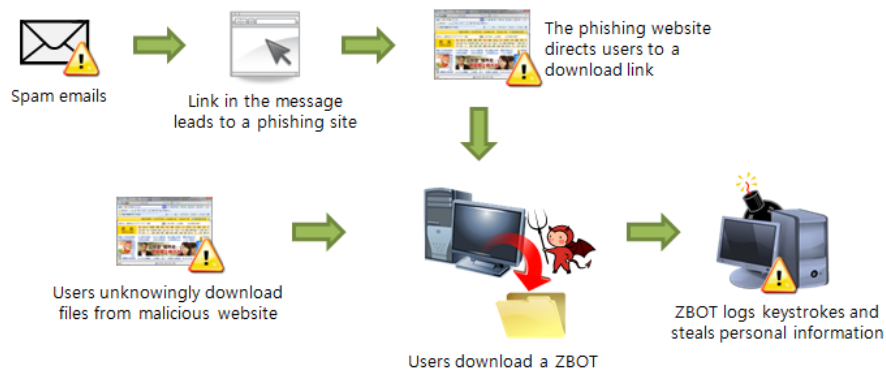
사. 델프(Delf)

어도비(Adobe) 제품의 취약점을 이용한 악성코드로, 허위 PDF파일을 만들어 감염시킨다. 감염된 PC는 검은 화면에 마우스만 나타나는 등 부팅 장애를 일으킨다. 다오놀(Daonol)로 불리기도 하며, 계속해서 변종 바이러스가 발생하고 있다.



(그림 2-16) 델프(Delf) 악성코드

아. 지봇(Zbot)



(그림 2-17) 지봇(Zbot) 악성코드

스팸메일 또는 해킹 사이트를 통하여 전파되는 악성코드로 어도비(Adobe)사 제품의 취약점을 통해 허위 PDF파일을 만들어 PC를 감염시킨다. 또한 추가로 다른 악성코드를 다운로드 한다.

자. 허위 보안툴(SecurityTool)

허위 안티바이러스 프로그램으로 위장한 악성코드로 시스템의 주요 파일을 패치한 뒤 사용자에게 요금 결제를 유도하는 악성코드이다. 변종에 따라 바탕화면의 아이콘을 보이지 않게 하거나, 블루스크린과 비슷한 화면을 보여주고 재부팅시키는 경우도 있다.



(그림 2-18) 허위 보안툴(SecurityTool)

차. AntiVirus XP 2010

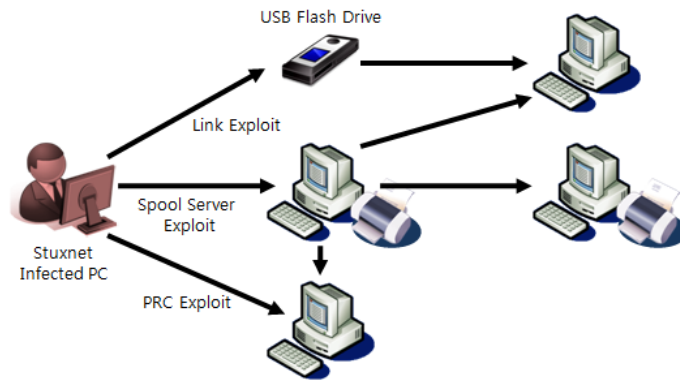


(그림 2-19) AntiVirus XP 2010

안티바이러스 프로그램으로 위장한 악성코드로 실행 시 윈도우업데이트를 가장한 허위 업데이트를 실시하며 지속적인 트레이창을 팝업 하여 치명적인 악성코드에 감염된 것처럼 사용자를 속여 결제를 유도한다.

카. 스텍스넷(Stuxnet)

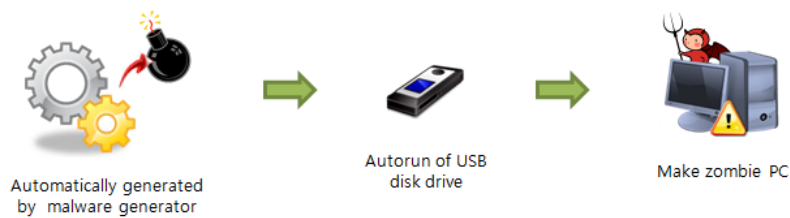
스텍스넷은 특정 프로그램을 타겟으로 하는 악성코드로서 독일 지멘스의 산업 자동화 시스템인 WinCC SCADA 시스템에서 작동하는 나타나는 악성코드 이다. 과거 악성코드의 목표가 개인 PC 또는 특정 서버에 국한되었지만, 최근에는 스텍스넷처럼 특정 프로그램을 겨냥한 악성코드가 발견되고 있다.



(그림 2-20) 스텍스넷(Stuxnet)

타. 팔레보(Palevo)

팔레보는 좀비PC를 만들어내는 대표적인 워의 한가지로 꼽히는 워으로서, 버터플라이(ButterFly)라는 악성코드 생성 툴에 의해 자동으로 만들어진다. 주로 USB 등의 이동식 저장장치를 이용하여 감염이 되며, 메모리 진단 및 치료를 하지 않으면 계속해서 사용자 시스템에 피해를 끼치는 특성이 있다.



(그림 2-21) Palevo

2. 최근 악성코드 공격 기술 동향

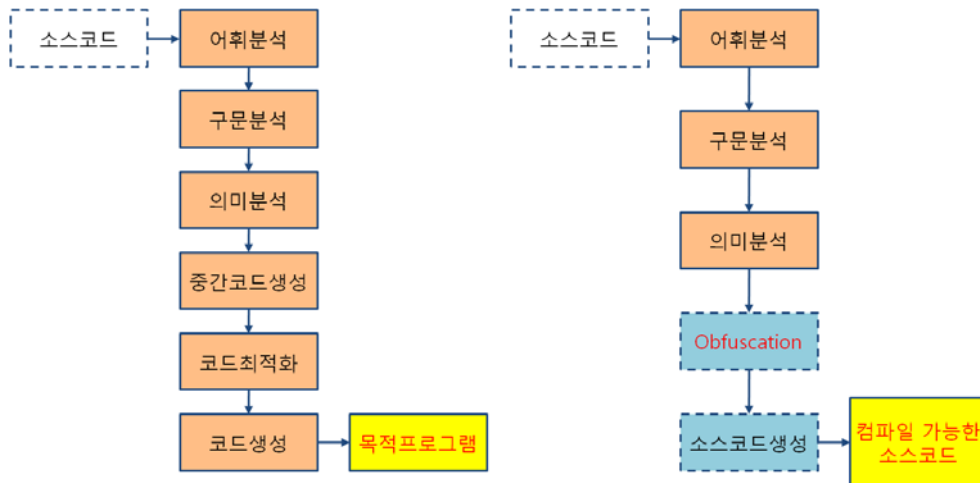
가. 분석 방해·지연 기술

악성코드들은 스팸메일 발송, 개인정보 탈취, 감염 플랫폼 손상, 서비스 방해 공격(DDoS) 등의 목적으로 다양한 전파방식을 통하여 네트워크 전역으로 퍼져나가, 보안이 취약한 플랫폼을 감염시킨다. 이후 악성코드들은 감염사실을 숨기기 위하여 루트킷과 같은 다양한 분석 방해·지연 기법을 사용하여 감염 플랫폼에서의 잔존 시간을 늘려 더욱 많은 정보를 외부로 유출시킨다. 대표적인 분석 방해·지연 기법으로는 생성되는 프로세스의 정보를 숨기는 방식이 있다. 악성코드는 자신의 프로세스 정보를 숨기기 위하여 프로세스 정보를 조회하기 위한 결과 값에서 자신의 프로세스 정보를 숨기거나 커널 레벨 구조체의 정보를 수정하는 방식 등을 사용하여 악성코드의 프로세스 정보를 숨길 수 있다.

다양한 은닉화 기술을 통해 악성코드 감염여부를 진단하지 못한다면, 감염된 플랫폼은 사용자의 개인정보를 지속적으로 악성코드 유포자에게 전송할 수 있으며 DDoS의 좀비 PC가 되어 제 2차 공격에 악용될 수 있다. 또한 다른 플랫폼을 감염시킬 수 있는 중간 노드의 역할을 수행하게 되므로 악성코드의 분석 방해·지연 기법을 우회하거나 무력화시키는 다양한 분석 기술들이 요구되고 있다. 악성코드 분석 방해·지연(은닉화) 기술로는 난독화, 패킹(Packing), 안티-디버거(Anti-Debugger), 안티-가상화(Anti-VM), 시한폭탄 등이 존재한다[24].

(1) 난독화(Obfuscation)

코드 난독화란 프로그램 코드의 일부 또는 전체를 변경하는 방법 중 하나로, 프로그래밍 언어로 작성된 코드에 대해 가독성을 떨어뜨려 읽기 어렵게 만드는 작업을 말한다. 일반적으로 소프트웨어를 분석하려는 역공학에 대한 대비책으로 프로그램에서 사용되는 알고리즘이나 아이디어를 숨기기 위해 사용된다. 코드 난독화는 기술을 무단으로 복제하는 것을 방지하는데 사용되거나 불법으로 침입하는 프로그램을 막기 위해 사용된다[19].



(그림 2-22) 일반적인 컴파일과 난독화 컴파일 과정

난독화 기술은 코드를 완벽하게 보호하지는 못하지만 원래 코드가 더 복잡한 형태를 갖도록 만든다. 난독화 작업은 들어가는 비용에 비해 코드를 보호하는 효과가 크기 때문에 최근 악성코드 제작자들이 안티바이러스 프로그램에 의해 악성코드가 탐지되지 않게 하기 위해 난독화 기술을 사용한다.

[표 2-5]는 코드 난독화 기법을 분류해 놓은 것이다.

[표 2-5] 코드 난독화 기법

구분	설명	
Layout Obfuscation	소스 코드의 포맷이나, 변수 이름, comments와 같은 응용의 layout을 대상으로 하는 방법	
Data Obfuscation	프로그램이 사용하는 데이터 구조를 대상으로 하는 방법	
	Storage	메모리에 데이터가 저장되는 방법을 변경 예) 지역 변수를 전역 변수로 변경
	Encoding	저장된 데이터가 해석되는 방법을 변경 예) 변수 i를 $c1*i+c2$ 로 변경
	Aggregation	데이터의 그룹을 변경 예) 하나의 배열을 여러 개의 하위 배열들로 분할
	Ordering	데이터의 순서를 변경 예) 배열을 reordering 하는 것, i번째 값이 특정한 함수 f에 의해 f(i)번째로 변경
Control Obfuscation	프로그램의 제어 순서를 대상으로 하는 방법	
	Aggregation	문장들의 그룹을 변경하는 방법 예) Inlining의 경우 함수 콜을 함수 코드 자체로 변경
	Ordering	문장들의 실행 순서를 변경 예) 루프의 진행 순서를 반대로 변경
	Computation	프로그램의 제어 흐름을 변경 예) 실행되지 않는 코드를 추가하거나 불필요한 코드를 추가
Preventive Transformation	deobfuscator들이 코드 자체를 break하기 어렵게 함	
	Targeted	자동 obfuscation 기술 적용을 어렵게 함
	Inherent	deobfuscator들의 약점을 이용

(2) 패킹(Packing, 실행파일 압축)

패킹(Packing)이란 PE(Portable executable)형식으로 배포되는 프로그램들을 프로그래머가 자신의 프로그램이 리버서로부터 리버싱이 어렵게 하거나, 프로그램 용량을 줄이기 위해 사용하는 방식을 말한다. 악성코드

개발자는 자신이 개발한 악성코드가 안티바이러스 프로그램으로부터 쉽게 탐지되는 것을 방지하기 위해 악성코드를 패킹하여 유포할 수 있다. 패킹된 악성코드를 분석하기 위해서는 압축된 PE파일을 본래의 PE파일로 되돌리는 언패커(unpacker)가 필요하다. 언패커는 패킹되는 방식에 따라서 알맞은 언패킹 과정을 수행하게 되는데, 만약 악성코드를 패킹 때 알려지지 않은 방식으로 패킹하거나 여러 패킹방식을 복합적으로 사용하여 다중 패킹 할 경우, 시그니처 기반의 악성코드 분석기술을 우회하거나 어렵게 만들 수 있다.

[표 2-6]은 현재 개발된 패킹 툴들의 목록이다.

[표 2-6] 패킹 툴 목록

Armadillo	ASPack	ASprotect	BatExe
Bat2ex.BDTmp	Batlite	BitArts.Fusion	CryptFF
CryptFF.b	Crypter	CryptZ	DebugScript
DBPE	DoomPack	Exeshield	Eagle
Embedded CAB	Exe2Dll	ExeStealth	FlySFX
JDPack	MEW	Mmpo	NDrop
PEncrypt	PE_Patch.AvSpoof	PECRC	PCPEC
Pex	PE-Crypt.Negn	PECrc32	PEBundle
PE-Crypt.Moo	PE-Crypt.UC	PE-Crypt.Wonk	PE-Pack
PE_Patch.Aklay	PE_Patch.Ardurik	PE_Patch.Elka	Pingvin
PE_Patch.ZiPack	PE_Patch.Upolyx	Polyene	Stxe
tElock	Teso	Yoda Crypter	ZiPack

(3) 안티-디버거(Anti-Debugger)

악성코드를 분석하는 방식으로는 디버깅이 오랫동안 사용되고 있다. 악성코드들은 안티바이러스 프로그램이 자신을 분석하는 것을 방해하기

위해서 디버깅이 수행되지 못하도록 디버거가 실행될 때 이를 탐지하여 다른 행위를 하거나 디버거의 실행을 종료시키는 등 다양한 방식을 사용할 수 있다. 다음의 함수들은 악성코드가 디버거의 실행여부를 탐지할 수 있는 함수들에 대한 설명이다.

o IsDebuggerPresent()

IsDebuggerPresent() 함수는 디버거의 정보를 확인하여 디버거가 진행 중이면 '1'을, 그렇지 않으면 '0'을 리턴하는 함수이다. 악성코드는 해당 리턴 값을 확인하여 디버깅이 진행 중인지 확인할 수 있다. 하지만 IsDebuggerPresent()는 커널 레벨의 디버거는 탐지하지 못하며 사용자모드 디버거만 탐지 할 수 있다.

o CheckRemoteDebuggerPresent()

CheckRemoteDebuggerPresent() 함수는 디버거가 프로세스에 접근하는 것을 확인할 수 있는 함수이다. 이 함수는 2개의 파라미터를 받아들이는데, 첫 번째 파라미터는 프로세스 핸들이며, 두 번째 파라미터는 부울 변수의 포인터다. 만약 프로세스가 디버그 중일 경우 부울 변수는 참(TRUE) 값을 갖게 된다.

o NtQueryInformationProcess()

NtQueryInformationProcess()는 커널 구조체인 EPROCESS의 디버그포트(DebugPort)의 표식(Flag)을 체크하여 디버거가 수행되는지를 탐지할 수 있다. 이 함수는 5개의 파라미터를 가지는데, 디버거를 탐지하기 위하여 ProcessInformationClass는 ProcessDebugPort(7)을 설정한다. 즉, 유저모드의 디버거가 프로세스를 디버깅 중일 때는 DebugPort 필드에 '0'이 아닌 값이 나타난다. 이 경우에 ProcessInformation의 값은 '0xFFFFFFFF'

가 되고, 그렇지 않은 경우에는 '0'이 된다.

(4) 안티-가상화(Anti-VM)

가상머신 환경은 사용자 또는 관리자가 호스트 운영체제 내에 게스트 운영체제를 추가로 설치할 수 있도록 가상의 환경을 제공한다. 설치된 게스트 운영체제 내에서 악성코드 분석 시 시스템이 손상되더라도 본래의 호스트 운영체제와는 독립적이므로 게스트 OS만을 복구하여도 시스템이 손상되기 이전의 상태로 복원할 수 있다. 이러한 가상머신의 종류로는 VMware, Xen, VirtualPC 등이 있다.

안티-가상화는 악성코드 분석 기법중 하나인 가상환경에서의 악성코드 분석 기술을 우회하는 방식이다. 안티-가상화는 가상머신 환경을 구축함으로써 나타나는 특징을 탐색하여 악성코드가 수행할 플랫폼이 가상머신 환경인지 판단할 수 있다. 다음은 가상머신 환경을 탐지하는 3가지 방법이다.

- 프로세스, 파일시스템, 레지스트리 요소 탐지 : 안티-가상화는 가상머신 환경 구축 시 수반하는 특정 프로세스, 레지스트리, 디렉터리, 파일등을 탐지하여 가상머신 환경을 판단할 수 있다.
- 메모리 요소 탐지 : 가상머신 내에서 구동되는 게스트 운영체제에서 사용하는 메모리 맵은 호스트 운영체제에서 사용하는 일반적인 메모리 맵과의 차이를 보인다. 악성코드는 게스트 운영체제와 호스트 운영체제의 메모리맵 차이를 탐지하여 가상머신 환경을 판단할 수 있다.
- 가상 하드웨어 주변장치 : 가상머신 환경에서는 주변장치(마우스, USB 컨트롤러, VGA 등)를 사용하기 위하여 주변장치도 가상화 한다. 악성코드는 가상화 된 주변장치를 탐지하여 가상머신 환경을 판단할 수 있다.

(5) 시한폭탄 방식

웹사이트를 해킹하여 악성코드의 유포지로 사용할 경우 이에 대한 효

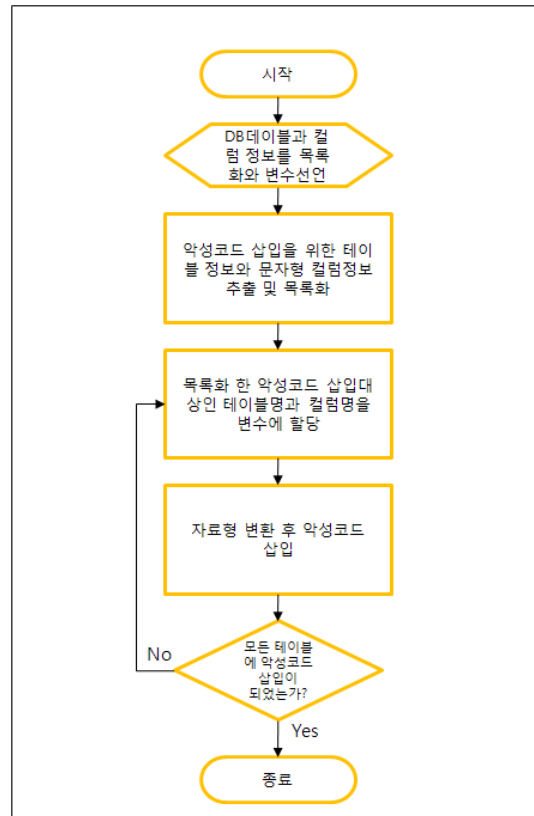
과적인 대처방안은 클라이언트 허니팟(Client Honeypot)과 크롤러(crawler)와 같은 자동화된 악성코드 탐지기술이다. 자동화된 악성코드 탐지기술은 웹브라우저를 통해 의심되는 웹서버를 직접 방문한 후 시스템에 허가되지 않은 변화(파일 시스템의 변화, 레지스트리 변경, 프로세스 생성 등)를 관찰하여 해당 홈페이지의 악성코드 감염 여부를 판단한다. 하지만 클라이언트 허니팟과 웹크롤러와 같이 자동적으로 웹사이트를 방문하여 악성코드 감염 여부를 판단하는 기술들은 검사하는 웹사이트에 특정시간동안 머물러야 하는 단점이 있다. 또한 이런 단점을 이용하는 시한폭탄 기반의 악성코드가 존재한다. 시한폭탄 기반의 악성코드의 경우 자동화된 악성코드 탐지 기술을 회피하기 위하여 웹사이트 방문 시 특정 시간 이후에 악성코드가 실행되게 하여 자동화된 악성코드 수집 프로그램을 우회할 수 있다.

나. 악성코드 대량 삽입 기술

웹사이트의 취약점을 이용하여 웹사이트에 악성코드를 삽입하는 방식은 악성코드를 대량으로 유포하기에 적합한 방법이기때문에 공격자는 악성코드의 전파경로로 웹사이트를 주로 사용한다.

웹사이트에 악성코드를 삽입하는 경우는 대부분 SQL 삽입(Injection) 취약점으로 인해 발생하게 된다. SQL 삽입 공격은 홈페이지와 데이터베이스가 상호 데이터 교환 시 데이터에 대한 검증을 수행하지 않아 공격자가 삽입한 SQL 명령어가 수행되면서 발생하는 문제점이다[23].

최근, 공격자는 SQL 명령어를 특정 웹사이트에 주입하는 것을 넘어 SQL 명령어를 대량으로 삽입하기 위하여 데이터베이스의 문자형 칼럼의 자료값 모두에 악성코드 유포지 URL을 삽입하여 대량의 악성코드를 유포하는 기술을 사용하고 있다. 다음의 (그림 2-23)는 데이터베이스에 악성코드 유포지 URL 스크립트를 삽입하는 과정이다.

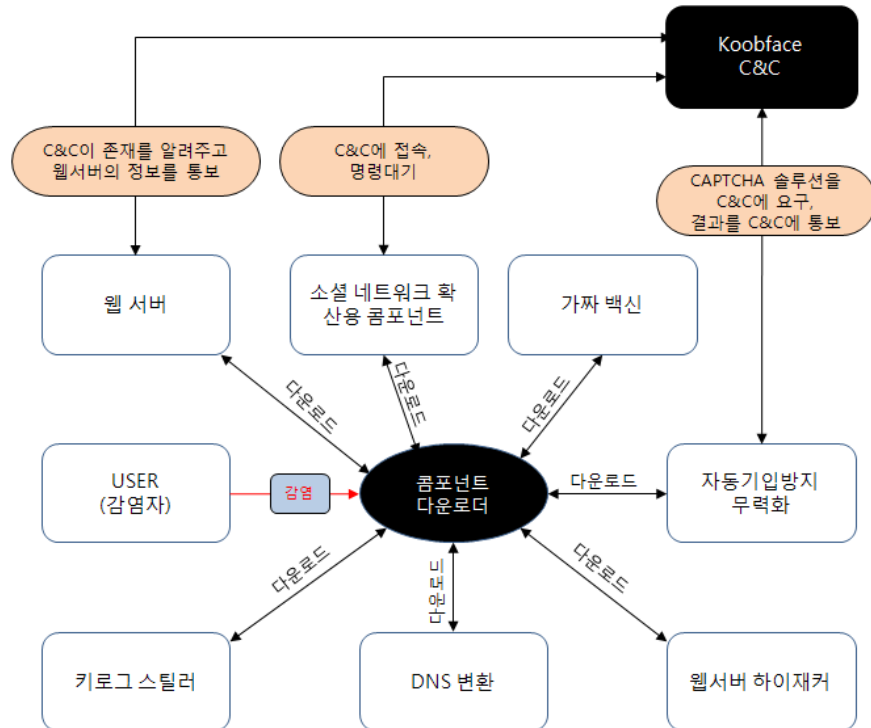


(그림 2-23) 악성코드 대량삽입 흐름도

다. 악성코드의 모듈화

과거 악성코드의 경우 다양한 악성행위를 수행하는 악성코드라도 모든 기능을 하나의 파일로 만들어 유포되었다. 하지만 악성행위의 다양화와 감염 PC에 오랫동안 상주하기 위한 분석방해 기법들을 적용한 악성코드들은 부피가 커져 하나의 파일로 모든 악성행위를 수행하는데 어려움이 있었다. 이에 악성코드 유포자들은 악성행위별로 악성코드들을 모듈화하여 유포하고 있다. 사용자가 악성코드에 감염되었을 시 해당 악성코드는 다른 기능을 수행하는 악성코드들을 다운로드 하는 역할만을 수행한다. 이렇게 악성코드들을 기능별로 모듈화 할 시 수행하는 기능을 제한하여 악성코드로 판단하는 기준을 충족하지 못하여 안티바이러스 프로그램

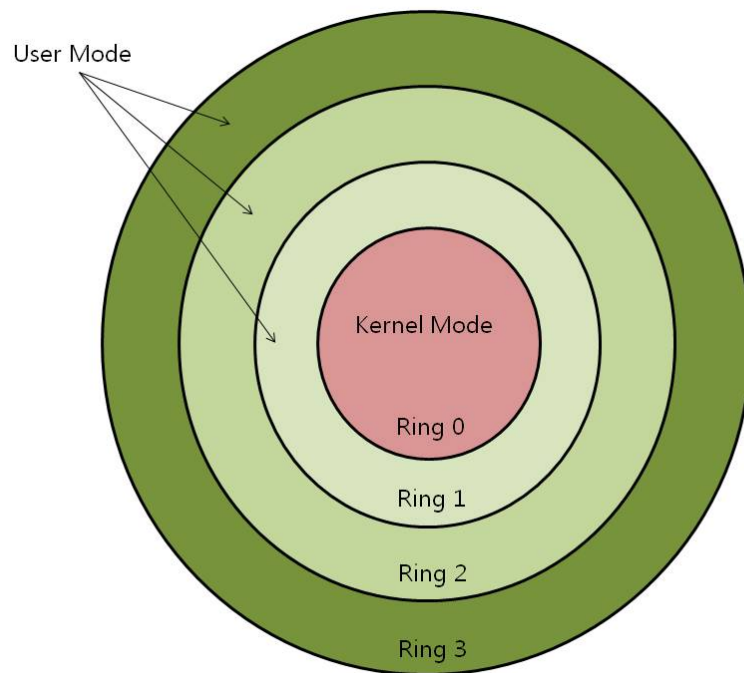
램을 회피할 수 있다[21].



(그림 2-24) 기능별로 모듈화를 사용하는 콤페이스(Koobface) 구성도

라. 악성코드의 은닉화

자기 자신을 은닉시키는 악성코드의 대표적인 예는 루트킷이다. 루트킷은 시스템 내부에 침투해 시스템에게 잘못된 정보를 전달하거나 민감한 데이터를 적에게 유출한다. 다른 악성코드와 다르게 루트킷은 시스템 관리자에게 발각되지 않는 것이 주목적이다. 아래 (그림 2-25)에서 루트킷은 ring 0에 존재한다. Ring 0는 커널 레벨로서 제일 높은 권한을 가지고 있기 때문에, 루트킷 역시 커널과 동일한 권한으로 동작한다[7].



(그림 2-25) Ring 보안 모델

루트킷은 다양한 방법을 통해서 시스템으로부터 은닉한다. 백신 프로그램 역시 시스템 서비스를 사용하기 때문에 루트킷을 탐지하기가 매우 어렵다. 루트킷의 한 종류인 부트킷(bootkit)은 MBR 영역을 감염시키기 때문에 백신 프로그램의 탐지 자체가 애초에 불가능하다. 다음은 대표적인 은닉 기술들이다.

- o Hooking : 원래 함수의 코드를 수정하여 다른 명령어를 수행하게 하거나 함수 테이블을 엔트리를 다른 주소 값으로 변경한다.
- o 시스템 프로그램의 변조 : 시스템 프로그램을 변조하여 시스템이 정상인 것처럼 결과를 반환한다.
- o 커널 데이터 조작 : 커널에서 사용하는 데이터 값을 수정하여 루트킷을 은닉할 수 있다.
- o 디바이스 드라이버 사용 : Layered Device Driver (kernel 모듈이 다른 장치에 접근할 때 사용)을 이용하면 keyboard, file system 또는 network I/O를 intercept 할 수 있다.

- o 레지스터 변조 : debug register를 이용해서 특정 메모리 접근 또는 실행이 특정 부분으로 이동했을 때 실행을 제어할 수 있다. 또 다른 레지스터인 Model Specific Registers (MSRs)를 사용하면 특정 명령어 (예를 들면 SYSENTER)가 실행되었을 때 제어를 할 수 있다.
- o Callback 함수 사용 : 운영체제 API를 이용해서 특정 이벤트 발생 시 호출되는 call back 함수를 등록해서 실행 흐름을 제어한다.

루트킷은 악의적인 목적으로만 사용하는 것은 아니다. 국가 정보기관에서도 루트킷 기술들을 사용한다. 실제로 FBI에서는 매직 랜턴(Magic Lantern)이라는 프로그램을 개발해서 범인을 추적하는데 사용하였다. 세계적인 기업 소니는 자사 제품의 불법 복제를 막기 위해서 루트킷을 사용하였다. 소니의 루트킷은 CD 플레이어에 설치되어 이름이 "\$SYS\$"으로 시작되는 파일, 폴더, 레지스트리 키 등을 숨기고 회사에 플레이어 ID, 사용자 컴퓨터의 IP 주소 등을 전송하였다. 하지만 시스캔(SysScan)의 Mark Russinovich가 자신의 블로그에 이와 같은 내용을 업로드하면서 언론의 주목을 받고 소니는 법정에 서게 되었다.

마. 안티-포렌식(Anti-Forensic)

(1) 데이터 디스트럭션(Data Destruction)

포렌식을 통한 악성코드에 대한 분석을 어렵게 하기 위한 가장 기본적인 방법은 포렌식에 이용되는 정보를 최소화하는 것이다. 즉 안티-포렌식의 데이터 디스트럭션은 분석에 이용될 수 있는 각종 변수값, 메모리 버퍼의 값들을 안전하게 삭제하는 기술을 의미한다. 이러한 방법으로 데이터를 삭제하거나 난수 값으로 채우는 방법이 있다.

(2) 데이터 하이딩(Data Hiding)

데이터 하이딩은 포렌식에 이용될 수 있는 결정적인 정보들을 처음부

터 발견하기 어려운 위치에 저장하여 사용하는 방법을 의미한다. 일반적으로 사용하지 않는 파일 시스템의 메타 데이터나 리저브드(reserved) 디스크 섹터에 정보를 저장함으로써 해당 정보들이 분석에 이용되는 확률을 줄이는 기술이다.

(3) 데이터 변환(Data Transformation)

데이터를 어떤 알고리즘을 통해서 그 의미를 모호하게 만드는 것을 데이터 변환이라고 한다. 스테가노그래피 또는 환자·전치 암호가 대표적인 예이다.

(4) 데이터 칸트러셉션(Data Contraception)

"The grugq"라고 불리는 보안 연구원은 데이터를 분석이 불가능한 곳에 데이터를 저장해서 포렌식 증거물의 양을 줄이는 아이디어를 냈다. 디스크에 데이터를 쓰는 것을 예방하여 포렌식 툴이 분석가에게 노출이 되는 것을 막는다.

(5) 데이터 패브리케이션(Data Fabrication)

데이터 패브리케이션의 목표는 분석가에게 잘못된 정보를 제공해서 분석가들이 잘못된 방향으로 흘러가게 유도하는 것이다. 예를 들어, 만약 분석가가 체크섬을 확인한다면, 최대한 많은 파일을 수정하여 포렌식 분석에 소요되는 시간을 증가시키는 것이다.

(6) 파일 시스템 공격(File System Attack)

파일 시스템이 사용하는 자료 구조를 훼손해서 포렌식 분석을 막는 방법이다. 예를 들어, 파일 시스템의 부트 섹터 또는 마스터 파일 테이블을

훼손하면 포렌식 도구가 파일 시스템을 정상적으로 분석 못하는 결과가 발생할 수 있다.

3. 악성코드 대응전략

가. 대응 기술 구분

[표 2-7] 악성코드 탐지 및 대응 기술 구분

	시그니처 기반	행위 기반
호스트 기반	<ul style="list-style-type: none"> ○ 분석된 악성코드의 패턴을 이용하여 탐지 ○ 알려지지 않은 악성코드 탐지가 어려움 	<ul style="list-style-type: none"> ○ 악성코드의 이상 행위를 기반으로 탐지 ○ 탐지/분석 회피기술을 사용 ○ 오탐의 소지가 있음
네트워크 기반	<ul style="list-style-type: none"> ○ 보유 비정상 트래픽 패턴을 이용 ○ 패턴을 벗어나는 경우 탐지가 불가능 	<ul style="list-style-type: none"> ○ 네트워크 트래픽 분석으로 비정상 봇넷 트래픽 탐지 ○ 오탐의 소지가 있고 탐지율이 매우 낮음

악성코드 탐지 및 대응 기술은 기술을 적용하는 대상에 따라 호스트 기반 탐지 기술과 네트워크 기반 탐지 기술로 구분될 수 있으며, 해당 기술의 동작 특성에 따라 시그니처 기반과 행위 기반으로 구분될 수 있다. 호스트 기반 악성코드 탐지 기술은 각 호스트에 설치된 악성코드 탐지 프로그램을 사용하여 파일 혹은 시스템의 악성코드를 검사하는 방법이다. 이와 달리 네트워크 기반 악성코드 탐지 기술은 네트워크의 경계에서 각 호스트로 전달되는 네트워크 트래픽을 수집하고 검사함으로써 악성코드를 탐지하는 방법이다. 또한 시그니처 기반 악성코드 탐지 기술은 파일의 특정 부분 또는 고유한 부분을 대상으로 하여 이미 알려진 악성코드의 패턴과의 일치 여부를 검사하는 기법이며, 행위 기반 악성코드 탐지 기술은 시스템 내에서 일어나는 다양한 행동을 분석하여 악성코드

의심 파일을 탐지해내는 방법이다. [표 2-7]은 이와 같은 악성코드 탐지 및 대응 기술을 호스트-시그니처 기반, 호스트-행위 기반, 네트워크-시그니처 기반, 네트워크-행위 기반의 4가지로 구분해 놓은 것이다[20].

(1) 호스트-시그니처 기반

호스트-시그니처 기반 악성코드 탐지 기법은 호스트의 파일 시스템을 대상으로 알려진 악성코드의 패턴을 이용하여 탐지하는 기술이다. 대부분의 안티바이러스 소프트웨어는 호스트에서 파일 기반의 탐지 기법을 사용하여 악성코드를 검사한다. 데이터베이스에 악성코드의 시그니처 정보를 최신으로 유지하는 경우 높은 탐지율을 갖는 특징이 있으나, 알려지지 않은 새로운 패턴의 악성코드 또는 시그니처의 일부를 변경한 악성코드의 탐지에는 취약하다는 단점이 있다.

(2) 호스트-행위 기반

호스트-행위 기반 악성코드 탐지 기법은 실행파일이 실행될 때 시스템 내에서 일어나는 행동을 관찰하여 악성코드로 의심되는 파일을 탐지해내는 기법이다. 동적 분석 도구, 샌드박스 등 프로그램의 실행을 모니터링함으로써 악성코드를 검출하는 분석 기법들이 호스트-행위 기반 악성코드 탐지 기법에 포함된다. 전통적인 시그니처 기반 악성코드 탐지 기법과 달리 최근 대두되는 악성코드 분석 기법을 우회하는 신종 악성코드 또한 검출해 낼 수 있다. 그러나, 정상 파일을 악성코드로 잘못 판단하는 오탐(false-positive)의 가능성이 있다는 단점이 있다.

(3) 네트워크-시그니처 기반

네트워크-시그니처 기반 악성코드 탐지 기법은 네트워크 패킷을 감시하여 악성코드로 추정되는 트래픽 패턴을 감지하는 기법이다. 알려진 공

격 시그니처를 감시하고 의심스러운 네트워크 활동을 탐지하며, 침입 탐지 시스템(IDS), 침입 방지 시스템(IPS) 등에서 사용된다. 호스트-시그니처 기반 탐지 기법과 마찬가지로 알려지지 않은 패턴의 악성코드의 탐지에 취약하다는 단점이 있다. 최근에는 미탐(false-negative)률을 낮추기 위한 노력의 일환으로 비정상 행위 탐지 기법과 함께 사용되는 경우가 많다.

(4) 네트워크-행위 기반

네트워크-행위 기반 악성코드 탐지 기법은 네트워크 패킷을 감시하여 미리 정의된 룰셋(rule set)이나 정상 행동을 벗어나는 행위를 하는 트래픽을 공격으로 간주하여 탐지하는 방법이다. 실시간 패킷 처리, 변형 공격의 탐지 등 실시간으로 각 상황에 맞는 대응을 할 수 있는 기술이며, 학습을 통해 정상/비정상 행위의 구분을 스스로 배워간다. 그러나 아직까지는 탐지율이 낮아 오탐의 가능성이 매우 크다.

나. 국내·외 기술 현황

(1) 국내 기술 현황

다음은 안철수 연구소, 하우리 등 국내의 주요 안티바이러스 업체들의 기술 및 제품에 관한 최근 현황이다.

- 넷시큐어: 안전한 인터넷 서비스 제공을 위한 신종 봇넷 대응 기술 개발을 위해 한국정보보호진흥원과 '신종 봇넷 능동형 탐지 및 대응 기술' 개발 공동연구 협약을 체결하였다. (2009)
- 소프트시큐리티: 스마트폰용 통합보안솔루션 '터치앤세이프'를 개발하였다. 터치앤세이프는 한국정보통신기술협회(TTA) 소프트웨어 기능 시험을 통과했다. (2010)
- 시큐아이닷컴: 코스콤 증권분야 통합관제시스템 구축사업에 삼성전

자의 기가급 L2 보안스위치 'iES4200 시리즈'를 공급했다. iES4200 시리즈는 IP 스누핑을 통한 도·감청을 차단하고 웜이나 봇에 의한 유해트래픽 발산을 통제하는 보안 기능을 수행한다. (2010)

- 안철수 연구소: 좀비PC 대응용 네트워크 보안 장비 '트러스와처 (AhnLab TrusWatcher)'를 출시하였다. 트러스와처는 클라우드 기반 악성코드 분석 시스템인 AhnLab Smart Defense를 활용, 좀비PC를 만드는 악성코드를 차단함으로써 내부 PC의 좀비화를 방지한다. (2011)
- 어울림정보기술: 네트워크 통합보안제품 '시큐어웍스' 제품군이 EAL4 국제공통평가기준(CC) 인증을 획득하였다. 고성능 방화벽·VPN 전용 솔루션 기반 위에 통합보안 기술력과 IPv6 환경을 지원함으로써 긴급 대응 인프라가 유기적으로 결합된 네트워크 보안 솔루션을 제공한다. (2011)
- 에어큐브: 스마트워크 보안에 초점을 둔 '패킷 기반 네트워크 접근 제어 솔루션'을 개발하였다. 악성코드에 감염된 단말기의 패킷을 분석해 회사 네트워크와 격리하는 솔루션을 제공한다. (2011)
- 윈스테크넷: 좀비PC 대응시스템 '스나이퍼BPS' 출시와 함께 EAL3 등급의 정보보호제품 공통평가기준(CC)을 획득하였다. 스나이퍼BPS를 통해 DDoS 대응시스템, 침입 방지 시스템, 방화벽 등의 공격 차단 장비와 연동하는 능동적인 DDoS 예방·대응이 가능하다. (2011)
- 이글루시큐리티: 전자우편 첨부파일 악성코드 탐지 및 차단 시스템 제품인 '에스코트(e-Scort)'가 IT보안인증사무국으로부터 CC인증을 획득하였다. (2010)
- 하우리: 통합보안 중앙관리 솔루션인 '바이로봇 매니지먼트 시스템'이 한국정보통신기술협회로부터 GS인증을 획득하였다. 바이로봇 매니지먼트 시스템은 모니터링 센터, 악성코드 의심 파일 자동수집 시스템, NAC(네트워크 접근제어) 기반의 관리 프로그램 설치 유도 기능 등을 제공한다. (2011)
- 한국전자통신연구원: 네트워크를 통해 유입되는 바이러스, 웜, 트로이목마 등 악성 변종 코드를 실시간으로 탐지, 분석하고 관리함으로써 네트워크 공격을 원천 방지할 수 있는 신·변종 악성코드 탐지 기술을 개발하였다. (2008)

(2) 국외 기술 현황

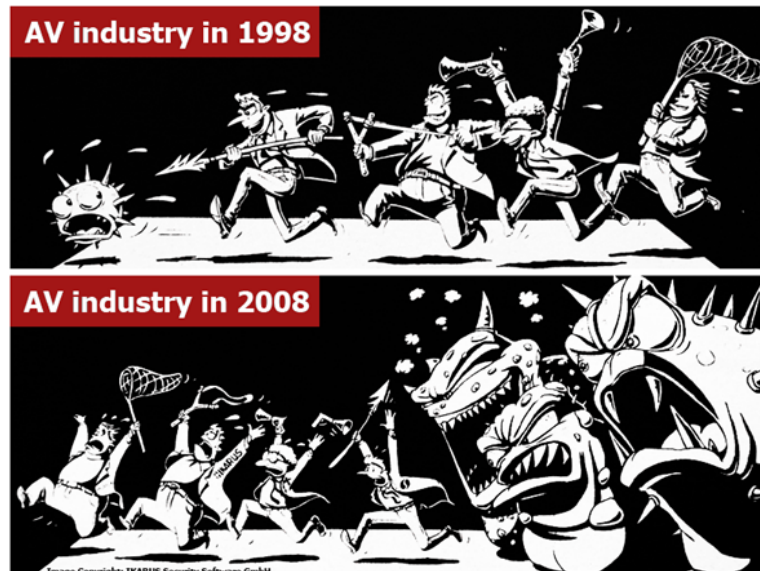
다음은 맥아피, 시만텍 등 국외의 주요 안티바이러스 업체들의 기술 및 제품에 관한 최근 현황이다.

- HP: 네트워크 보안 신제품 '티핑포인트 레퓨테이션 DV'를 출시하였다. 새로운 티핑포인트 솔루션은 악성 IP 및 유해 URL 등 좀비 PC를 만드는 악성코드 유포처로부터 기업을 보호한다. (2011)
- IBM: 20Gbps 처리 성능과 IPv6를 완벽 지원하는 차세대 네트워크 침입방지시스템 'GX7800'를 출시하였다. GX7800은 차세대 데이터센터 클라우드 환경을 고려한 고성능 및 광범위 네트워크 보안 기능을 제공한다. (2011)
- 마이크로소프트: 보안성이 강화되고 빨라진 웹브라우저 '인터넷 익스플로러 9'를 출시하였다. 스마트 스크린 필터, 추적 방지 기능 등을 통해 악성코드의 99% 정도를 차단할 수 있다. (2011)
- 맥아피[10]: 시장조사기관인 가트너가 발간한 '매직 쿼터런트'의 네트워크 IPS(침입 탐지 방지) 어플라이언스 부문에서 리더 그룹으로 선정되었다. (2010)
- 시만텍: 사용자 커뮤니티 기반의 평판 기술을 접목한 차세대 보안 신기술 '유비쿼티(Ubiquity)'를 발표하였다. 1억대 이상의 시만텍 고객 컴퓨터 정보를 익명으로 수집해 소프트웨어 사용 패턴을 파악함으로써 기존의 보안 솔루션으로 탐지가 불가능한 소규모 보안 위협까지 차단이 가능해졌다. (2010)
- 소닉월: 어플리케이션 제어 기능을 포함, 포트 기반의 기존 방화벽을 대체하고, 고도화된 보안을 제공하는 차세대 방화벽 솔루션인 '슈퍼매시브 E10000 시리즈'를 출시하였다. (2011)
- 익시아: 6천여 개의 고유 공격 데이터베이스를 갖춘 네트워크 취약성 테스트 솔루션 '익스로드-어택'을 출시하였다. 노출된 취약성을 악용하는 공격성 트래픽을 재현해 제어 환경에서 분산서비스거부(DDoS) 공격을 생성하고, 클라우드 환경과 기업 및 공공기관, 서비스 제공업체 네트워크의 장치 검증 및 보호 기능을 유지할 수 있다. (2011)
- 팔로알토네트웍스: 최대 20G 방화벽 성능과 10G IPS 성능을 동시에 제공하는 20G 차세대 방화벽 'PA5000 시리즈'를 발표하였다. 4기가의 IPSec VPN성능과 동시 2만 사용자가 접속할 수 있는 SSL VPN을 지원하며 가상화 보안을 위해 대당 225개의 차세대 가상 보안 시스템 구성이 가능하다. (2011)

제 3 절 악성코드 전파 방법

1. 악성코드 전파경로

악성코드가 악성코드 제작자의 목적에 맞게 행동하고, 제작자가 원하는 바를 수행하려면 우선적으로 공격 대상 시스템에 침투하여 시스템을 감염시켜야한다. 즉, 악성코드가 공격 대상 시스템에 들어가지 않으면 해당 시스템에게 아무런 악의적인 행위를 수행 할 수 없다. 그러므로 공격자들은 자신들의 목적대로 행동 할 악성코드를 피해 대상 시스템에 감염시키기 위해 다양한 감염 경로를 이용하며 악성코드는 스스로 자신을 전파하기 위해 다양한 방법을 이용한다.



(그림 2-26) 현재 AV industry의 모습 (출처 : IKARUS)

‘사전에 미리 준비가 되어 있으면 걱정할 것이 없다’ 말이 있듯이 악성코드는 전파경로를 통하지 않고 공격 시스템으로 침투할 수 없기 때문에 웜, 트로이 목마, 바이러스 등의 악성코드 침입 경로를 미리 아는 것

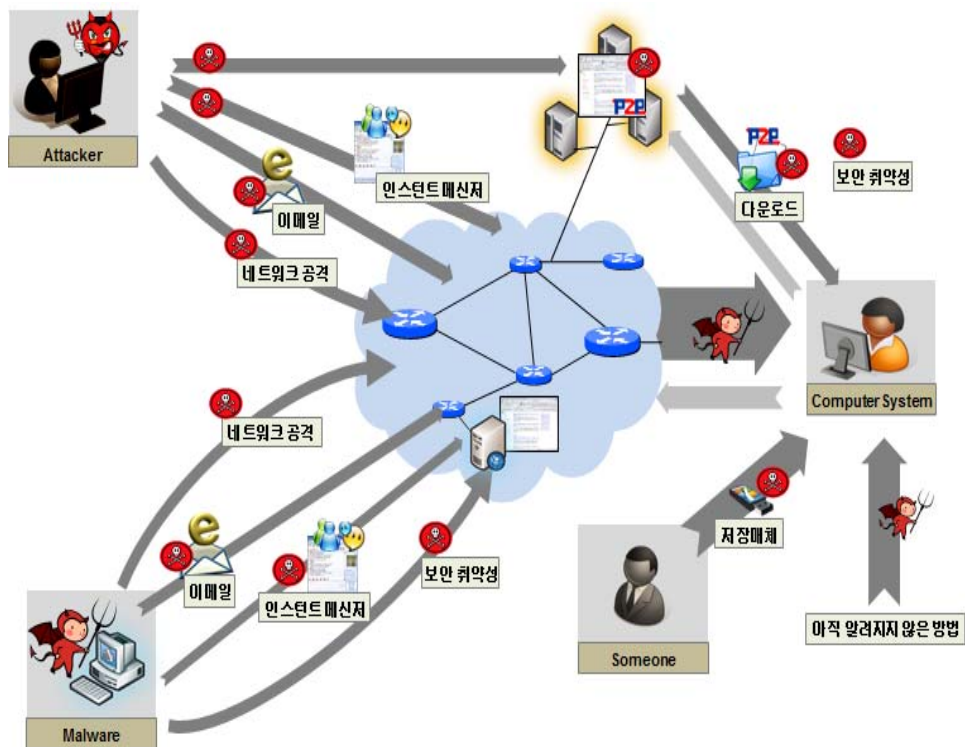
만으로도 어느 정도 악성코드의 위협에 대응할 수 있다.

초기의 악성코드는 단순한 전파 방법을 이용하였다. 과거의 네트워크 인프라는 단순하였고, 공격자들의 공격 수법도 단순하였기 때문에 네트워크의 경계만으로도 충분히 악성코드의 전파를 예방할 수 있었다. 하지만 인터넷과 다양한 웹브라우저 등이 발달함과 스마트폰, 노트북 등 여러 통신 기기들이 발달함, 그리고 인터넷을 통한 활동이 급속히 늘어남에 따라 현대의 네트워크 인프라는 복잡해졌고, 공격자들의 공격 수법도 다양해져서 악성코드가 전파되는 경로 또한 매우 다양해졌다.

이렇게 다양해진 악성코드의 전파 방법은 성격이 비슷한 전파 방법들끼리 묶어 여러 가지 유형으로 나눌 수 있다. 본 절에서는 악성코드 전파 방법을 저장매체, 네트워크, 다운로드, 전자우편, 보안 취약성, 인스턴트 메시징 프로그램, 그 외로 분류하여 설명한다. 또한 각 유형에 속하는 여러 방법들을 설명하고, 각 유형별 피해 사례를 살펴본다.

가. 최근 악성코드 전파 방법

최근 악성코드는 많은 컴퓨터를 감염시키기 위해 매우 다양한 전파 방법들을 이용하고 있다. 본 절에서는 악성코드 전파 방법을 다음 (그림 2-27)에서 볼 수 있듯이 크게 저장매체, 네트워크, 다운로드, 전자메일, 보안 취약성, 인스턴트 메시징 프로그램, 그 외로 나누어 간단하게 설명한다. 대부분의 악성코드 전파방법은 악성코드를 전파하기 위해 공격자에 의해 사용되거나 악성코드가 자신을 전파하기 위해 사용하는 경우가 있다. 즉, 수동적인 전파와 자동적인 전파 두 가지 양상을 보이는 것이다.



(그림 2-27) 악성코드 전파 방법의 유형

(1) 저장매체

저장매체는 텍스트, 이미지, 음성, 영상 등의 디지털 데이터(Digital data)를 저장하는 매체로 데이터를 보관 하거나 한 컴퓨터에서 다른 컴퓨터로 데이터를 옮기려고 할 때 주로 사용된다. 저장매체로는 Compact Disk(CD), 플로피 디스크, 이동식 저장장치 등이 있다.

(가) Compact Disk(CD)

CD는 데이터, 음악, 영상 등의 디지털 데이터를 저장할 수 있는 저장 매체로 플로피 디스크보다 큰 용량의 데이터를 저장할 수 있다. CD에 데이터가 저장된 이후에는 바이러스가 감염될 수 없지만, 데이터를 CD

에 처음 저장하는 과정에서 바이러스가 같이 CD에 저장될 수 있다. 이런 경우 CD 부팅 시마다 바이러스가 자동적으로 실행되어 해당 컴퓨터를 감염 시키게 된다.

(나) 플로피 디스크

플로피 디스크는 디지털 데이터를 기록할 수 있는 원판 모양의 자성 매체이다. 플로피 디스크는 1990년대 중반까지만 해도 많이 사용되었지만 현재는 CD와 같은 대용량 저장매체의 등장으로 그 사용량이 현저하게 줄어들었다. 사용자가 플로피 디스크를 컴퓨터에 꽂아둔 채 컴퓨터 작업을 마치고 이후에 꽂혀있는 상태에서 다시 컴퓨터를 켜는 경우가 많은데, 이때 꽂혀있는 플로피 디스크가 감염된 상태이고, 컴퓨터의 기본설정이 플로피 디스크로 부팅하고 실패 할 경우 하드디스크로 부팅하는 형태이면, 플로피 디스크의 부트 섹터에 저장되어 있는 바이러스가 실행되어 하드웨어를 감염 시키고, 이후 해당 컴퓨터에 사용되는 다른 플로피 디스크를 감염시키는 등의 방식으로 악성코드를 전파할 수 있다.

(다) 이동식저장장치

이동식저장장치는 USB, 휴대용 하드디스크와 같이 휴대 가능한 저장매체를 말하며, 최근 다양한 저장 용량과 다양한 형태를 가진 이동식 저장장치가 판매되고 있어 사용자들에게 편리성을 제공하고 있다. 하지만 휴대하기 편리하다는 장점으로 인해 다양한 장소에서 다수의 컴퓨터에 접속 시키는 것이 가능하여 악성코드 유포의 주요인 중 하나가 되었다. 악성코드는 감염된 해당 컴퓨터에 사용되는 모든 이동식저장장치를 자동으로 감염시키고, 이렇게 감염된 이동식저장장치를 다른 컴퓨터에 꽂아 사용하게 되면, 해당 컴퓨터도 감염되게 되어 급속도로 악성코드가 유포되게 된다.

(2) 네트워크

네트워크는 지속적인 발전을 통해 현재 우리 삶 속 깊숙한 곳까지 들어와 있다. 우리는 하루에 적어도 한번은 네트워크에 접속하여 여러 서비스를 이용하고 있다. 하지만 불특정 익명의 다수가 사용가능한 네트워크는 악성코드를 유포시키는 주범 중 하나가 되고 있다.

네트워크를 이용하여 악성코드를 유포시키는 경우를 세부적으로 나누면 ARP 스푸핑(spoofing), 블로그, 게시판, SNS(Social Network Service) 등이 있다.

(가) ARP 스푸핑(spoofing)

ARP 스푸핑은 공격자가 다른 시스템에게 MAC주소를 속이는 것이다. 즉, 자신의 MAC 주소를 특정 PC의 IP에 연결되는 MAC 주소라고 속여 자신이 속한 네트워크 대역의 모든 시스템이 자신을 게이트웨이로 믿도록 위장함으로써 해당 네트워크 대역의 모든 네트워크 통신을 중간에서 가로채는 해킹 기법을 말한다. 공격자는 해당 네트워크 대역의 모든 네트워크 트래픽을 가로챌 후 악성코드를 삽입하여 다른 시스템을 감염시킬 수 있으며, 사용자가 특정 웹페이지에 접속하려고 접속 요청하는 패킷을 전송하는 경우 공격자가 패킷을 중간에서 가로채어 사용자가 변조된 악성 웹페이지에 접속하도록 패킷을 위조하여 재전송하게 되면 사용자는 그 패킷을 받아 의심 없이 변조된 페이지에 접속하게 되어 악성코드가 감염될 수도 있다. 이외에도 어떠한 시스템을 감염시킨 악성코드가 같은 네트워크 대역의 다른 시스템 또한 감염시키기 위해 자동으로 ARP 스푸핑하는 경우도 있다[9].

(나) 블로그(Blog)

블로그는 자신의 생각과 관심사에 대한 게시글을 자유롭게 올리고, 다

른 사람들과 공유할 수 있는 개인 웹페이지로 볼 수 있다. 최근 우리나라 사람들의 블로그 사용량이 늘어나면서 블로그는 악성코드를 유포시키는 유포지 중 하나로 떠오르고 있다. 블로그는 블로그 주인이 올려놓은 게시글을 다른 사람들이 들어와 공유 할 수 있는 특징이 있어 블로그 주인이 의도적으로 감염된 악성코드를 자신의 블로그에 올려놓아 다른 시스템이 감염시키는 경우가 발생할 수 있다. 블로그에 배너광고를 올리는 경우를 자주 볼 수 있는데, 이러한 배너광고 스크립트 또한 악성코드 유포에 사용되고 있다.

(다) 게시판

게시판은 인터넷 상에서 사람들에게 특정 정보를 알리는 글을 올릴 수 있고, 사람들은 올라온 글들을 볼 수 있는 사이버 상의 공간이다. 게시판은 다양한 사람들이 접속하여 개개인의 글을 올리고, 알림 글을 보는 형태여서 악성코드 유포에 이용되고 있다. 게시판은 웹브라우저 보안 취약점을 이용하여 게시판에 접속하면 자동감염 방식으로 유포되는 방법에도 안전하지 않다.

(라) SNS

SNS는 불특정 다수의 사람이 인터넷 상에서 서로 친구를 맺어 친목을 강화할 수 있게 해주는 서비스이다. 최근 트위터, 페이스북 같은 SNS가 전 세계적으로 호응을 받으면서 SNS 기업들은 수많은 가입자들을 보유하고 있다. 무엇보다 최근에는 스마트폰이 널리 보급되면서 SNS 모바일 어플리케이션을 이용하여 SNS를 하는 등 SNS의 활용범위가 모바일로 까지 넓어지고 있다. 이러한 이유로 SNS가 바이러스 유포자들의 주요한 유포 매개체로 주목받고 있다. 공격자들은 악성코드가 첨부된 메일을 SNS에서 공식적으로 보내는 것처럼 가장하여 보내거나 페이스북 초대 내용처럼 위장한 형태로 악성코드를 유포하고 있다. 이외에도 악성코드

가 스스로 SNS의 친구목록을 검색하여 친구목록에 있는 모든 사용자에게 악성코드가 첨부된 메일을 보냄으로써 자기 자신을 유포시킬 수 있다.

(3) 다운로드

현재 인터넷 사용자들은 필요에 따라 여러 가지 방법으로 각종 프로그램을 다운 받을 수 있다. 프로그램 중 특히 실행파일과 이중 확장자 파일은 특별히 주의해야 할 대상이다. 실행파일은 파일 확장자가 보통 exe인 파일이고, 이중 확장자는 mov, scr, avi, pif등으로 보기에 동영상 파일, 그림 파일 같지만 실제로는 실행파일인 것을 말한다.

(가) 공개 자료실

공개 자료실은 여러 사용자들이 서로에게 필요한 자료를 올리고, 다운 받을 수 있는 곳이다. 1990년 초반에 바이러스 유포의 큰 역할을 한 것이 공개 자료실이다. 바이러스 제작자는 자신이 제작한 바이러스를 유용한 프로그램으로 속여 공개자료실에 올려놓고, 이 프로그램을 많은 사용자들이 다운 받음으로써 바이러스를 유포시킨다.

(나) FTP(File Transfer Protocol)

FTP는 인터넷을 통해 서버와 클라이언트 컴퓨터 사이에서 파일 전송을 하기 위한 프로토콜을 의미하면서 동시에 그런 것을 가능하게 하는 프로그램을 통칭한다. 최근 FTP가 널리 사용되면서, 의도적으로 자신의 FTP에 바이러스 감염 파일을 올리고, 이를 사람들이 다운 받도록 유도함으로써 바이러스를 유포하는 경로가 되었다. 최근 들어 웹이 활성화되면서 사용 빈도가 크게 줄어들었으나 아직까지 큰 위협 요소로 남아 있다.

(다) P2P

P2P는 인터넷을 통해 개인과 다른 개인이 직접 연결되어 파일을 공유하고 교환하는 서비스를 말한다. P2P는 개인 대 개인이 직접 연결되어 파일을 주고받는 형식이어서 중간에서 통제하기가 매우 어려우며, 데이터 파일 뿐만 아니라 실행파일까지도 공유되고 있기 때문에 악성코드의 배포의 주요 전파경로가 되고 있다.

(라) 웹하드

웹하드는 대용량 저장 장소를 사용자에게 제공해주는 서비스로 최근에 웹하드를 제공하는 업체와 이를 사용하는 사용자들이 늘어나고 있는 추세이다. 대용량에다가 너무 많은 사용자에 의해 통제가 불가능하여 악성코드 유포가 빈번하게 일어나고 있다.

(마) 액티브X(ActiveX)

액티브X는 마이크로소프트사에서 제공하는 기술로써 일반 응용프로그램과 웹을 연결시켜주기 위해 제공되는 기술을 말한다. 일반적으로 사용자들이 사이트의 서비스를 이용하기 위해 액티브X를 설치해야 되는 경우가 빈번하기 때문에 주의 깊게 살펴보지 않고 설치 버튼을 누른다. 이러한 특성을 이용하여 악성코드 제작자들은 악성코드를 액티브X처럼 다운로드 설치할 수 있도록 조작하여 유포하고 있다.

(바) 공유 폴더

공유 폴더는 인터넷을 사용하여 외부에서 내 컴퓨터의 파일들을 읽고, 쓰기를 가능하도록 하는 기능이다. 이외에도 여러 사람들이 특정 폴더를

함께 사용하도록 할 수 있어 편리성을 제공한다. 하지만 이러한 편리한 기능 때문에 공유 폴더는 악성코드 유포자에게 주목받고 있다. 여러 사람들이 공유 폴더에 자유롭게 읽고, 수정하고, 쓰기가 가능하므로 공유 폴더의 특정 파일을 감염 시키는 것만으로도 공유하고 있는 모든 사람들의 시스템까지도 쉽게 감염시킬 수 있다.

(4) 전자우편

전자우편은 1999년부터 지금까지 공격자들에게 악성코드 유포 매개체로 이용되고 있다. I-Worm/Happy99와 같이 전자우편을 전파 방법으로 이용한 초기의 바이러스는 단순히 악성코드를 첨부하여 메일을 보내는 형태를 띠었다. 하지만 최근 이러한 수법이 사람들에게 많이 알려지게 되면서 사용자들이 전자우편을 사용할 때 각별히 조심하게 되었다. 이로 인해 초기의 방식으로는 악성코드 유포가 어려워지자 공격자들은 전자우편에 자극적이고 흥미로운 제목이나 내용을 삽입하여 사용자들이 메일을 열어보거나 첨부된 파일을 실행하도록 현혹시키는 사회공학적인 기법을 사용하게 되었다.

이때 사용자들을 현혹시키기 위해 메일 제목이나 메일 내용에 들어가는 소재는 다음과 같이 성적 내용, 유명인이나 스타와 관련된 내용, 바이러스 경고 메일, 농담, 로맨틱한 내용, 불법 소프트웨어, 매우 중요한 내용이나 아무내용이 없거나, 믿을 만한 업체를 가장한 메일, 웹사이트를 가장한 첨부메일 등이

악성코드 유포자는 사회공학기법을 이용한 악성코드 전파 방법 이외에도 보안 취약성을 이용하여 상대방이 단지 메일을 열어보는 행위만으로도 심지어는 미리 보기만 했을 경우에도 악성코드에 감염되도록 하고 있다. 이러한 방식의 대표적인 예가 VBS/Bubbleboy라는 악성코드이다.

또한 악성코드가 자동적으로 동작하여 전자우편 프로그램에서 제공하는 주소록을 참조하여 주소록에 있는 모든 사람들에게 악성코드를 포함하고 있는 메일을 보냄으로써 악성코드를 전파를 위해 이용되기도 한다.

(5) 보안 취약성

최근에 시스템 자체, 어플리케이션의 보안 취약성을 이용하여 유포되는 악성코드가 증가하고 있다. 예를 들어 공격자들은 특정 웹브라우저의 취약성을 이용하여 사용자들이 해당 사이트를 방문하는 것만으로도 사용자의 컴퓨터에 악성코드가 감염되도록 한다.

(가) 웹사이트의 취약성

웹브라우저는 사용자들에게 웹에서의 정보를 검색하거나 볼 수 있게끔 하여 웹 서비스를 이용할 수 있도록 해주는 응용 프로그램이다. 현재 가장 많이 사용되고 있는 웹브라우저로는 익스플로러, 네스케이프, 크롬 등이 있다. 악성코드 유포자들은 웹브라우저들의 보안 취약성을 이용하여 사용자들이 악성 웹사이트에 방문만 해도 악성코드에 감염되도록 하고 있다. 즉, 사용자들이 취약성이 있는 웹브라우저로 악성 웹사이트를 방문하게 되면 악성 웹사이트의 운영자가 상대방의 시스템의 모든 권한을 가질 수 있다.

(나) IIS(Internet Information Services) 취약성

IIS(Internet Information Services)는 마이크로소프트의 윈도우 NT용 인터넷 서버 소프트웨어이다. IIS는 웹 서버, FTP 서버, SMTP 서버 등 여러 서버의 기능을 통합하고 있으며 IIS를 이용하는 사용자들은 마이크로소프트의 프론트페이지(FrontPage) 제품을 사용하여 웹페이지를 만들 수 있다. 또한 사용자들은 ASP 기술을 이용할 수 있는데, 이것은 웹페이지 내에 액티브X를 내장한 응용 프로그램들이 포함될 수 있다는 것을 의미한다.

공격자는 윈도우 NT용 인터넷 서버 서비스를 제공하는 IIS의 보안 취

악성을 이용하여 악성코드가 감염시킬 수 있다. 서버라는 특징상 특정한 서비스를 사용자에게 제공하기 위해 서비스 관련 포트를 계속 열어놓게 되는데 이러한 취약성을 이용하여 외부에서 쉽게 내부로 침투할 수 있으며 웹서버가 감염되면 해당 서버에 접속하는 사용자들도 쉽게 악성코드에 감염될 수 있다.

(다) SQL 서버의 취약성

SQL(Structured Query Language) 서버는 관계데이터베이스용 프로그래밍 언어인 SQL을 기반으로 하는 데이터베이스 서버를 의미한다. SQL 서버에는 웹이나 시스템을 운영하는데 필요한 수백만에서 수천만 건의 정보가 저장된다. 그러므로 SQL 서버가 악성코드로 부터 공격을 받게 되면 인터넷이 마비되는 사태까지 발생 할 수 있다. 이외에도 공격자들은 SQL 삽입(Injection) 공격을 이용하여 웹서버 내에 악성코드 삽입하여 해당 웹사이트를 방문하는 사용자들로 하여금 악성코드를 다운 받도록 유도하는 방법으로 악성코드를 전파하고 있다.

(6) 인스턴트 메시징 프로그램 (또는 채팅 프로그램)

인스턴트 메시징 프로그램(Instant Messenger Program)은 인터넷을 통해 두 명 이상의 사용자들이 실시간으로 텍스트를 주고받는데 사용되는 프로그램이다. 인스턴트 메시징 프로그램은 텍스트를 즉각적으로 주고받을 수 있다는 점이 메일과 다른 점이다.

(가) 메신저

메신저는 인터넷을 통해 자신의 친구 리스트에 등록되어 있는 사람들과 텍스트나 파일을 주고받을 수 있는 소프트웨어이다. 현재 대표적인 메신저로는 네이트온(NateOn), MSN, 다음(Daum) 메신저 등이 있다. 이

런 메신저는 간단한 텍스트 외에도 실행파일을 실시간적으로 주고받을 수 있기 때문에 악성코드 전파에 사용되고 있다. 공격자들은 일반적으로 악성코드 실행파일을 메신저로 직접 전송하거나 악성코드 스스로 메신저를 이용하여 퍼지도록 개발하거나 메신저로 악성코드를 다운로드하는 URL을 전송하는 등의 방법을 사용하고 있다. 안철수 연구소에 의하면 2011년 7월 악성코드를 배포하는 웹사이트의 차단 건수가 전월 4만9317건에 비해 295%가 늘어난 14만5467건으로 늘었다고 밝히고 있다.

(나) IRC 채널

IRC(Internet Relay Chat)는 인터넷을 통해 전 세계 사람들과 대화 할 수 있도록 하는 채팅 프로그램이다. IRC 클라이언트는 IRC 서버에 접속하기 위해 필요하며 윈도우에는 mIRC가 가장 많이 사용된다. IRC 클라이언트는 자체적인 스크립트 기능을 이용하는데 스크립트 중 하나인 DCC(Direct Client to Client) 명령을 사용하여 다른 IRC 클라이언트에게 파일을 전송할 수도 있다. 이러한 IRC 클라이언트의 스크립트 기능을 이용한 악성코드가 많이 존재하고 있다. 또한 악성코드 스스로가 IRC 채널을 이용하여 전파되도록 하거나 공격자가 다른 사용자에게 사회공학적 기법으로 호기심을 유발하여 특정 악성 웹사이트에 접속하게 하여 악성코드에 감염되게 한다.

(7) 그 외

위의 전파 방법 외에도 다른 악성코드를 통해 특정 악성코드가 전파되는 방식과 사회공학기법을 이용하여 악성코드가 전파되는 방식, 검색 엔진 포이즈닝을 이용한 전파, 그 외 신종 전파방법 등이 있다. 이처럼 악성코드는 네트워크를 기반으로 파일을 직접 공유하거나 파일을 공유할 수 있는 URL 을 전달함으로써 악성코드 전파를 수행할 수 있다.

(가) 다른 악성코드

악성코드 전파 방법 중에 악성코드에 의한 전파 방법이 있다. 시스템에 이미 침투한 악성코드가 동작하여 FTP 사이트, 웹사이트 등에서 특정 악성코드나 악성코드가 감염된 프로그램을 다운받아 설치하는 경우가 있다.

(나) 사회공학기법

사회공학기법은 기술적인 방법을 사용하는 것이 아닌 신뢰를 기반으로 사람을 속여 다른 사람들로 하여금 속이는 사람이 원하는 대로 행동하도록 만드는 기법을 일컫는다. 이러한 사회공학기법은 신뢰할 수 있는 발신자로 위장하여 메일을 보내거나 상대방의 흥미를 유발하는 제목이나 메일 본문을 작성하거나 정상파일로 위장된 감염된 첨부파일을 보내는 등의 방식으로 이용되고 있다.

(다) 검색 엔진 포이즈닝(SEP, Search Engine Poisoning)

검색 엔진 포이즈닝은 블루코트(Blue Coat)의 조사에 따르면 2011년 상반기에서 가장 대중적인 악성코드 감염 매개체이다. 악성코드의 40%가 검색 엔진과 포털을 통해서 전파된 것으로 알려지고 있다. 공격자들은 검색 결과에 악성 웹사이트에 접속하는 링크들을 많이 뜨도록 하여 사용자들이 접속하여 악성코드를 다운받도록 하고 있다[27].

(라) 그 외 신종

개인 컴퓨터와 네트워크의 발달은 사용자들에게 많은 편리성을 제공하지만 동시에 악성코드 전파 방법을 매우 다양하게 만들었다. 새로운 서비스의 탄생은 새로운 감염 경로가 작용할 수 있다.

나. 전파 방법 유형별 사례

위에서는 최근 전파 방법에 대해 유형별로 간단하게 살펴보았다. 여기서는 각 유형별로 최근 발생한 사례를 위주로 살펴 볼 것이다.

(1) 저장매체

이동식 저장장치, 플로피 디스크 등의 저장매체를 이용하여 악성코드가 전파되는 사례는 다음의 [표 2-8]과 같다[28].

[표 2-8] 저장매체를 이용하여 전파하는 사례

날 짜	세부내용
2008.06	[USB를 통해 전파되는 트로이 목마류 악성코드] 지난 4월 악성코드 동향에 따르면, USB를 통해 전파되는 악성코드는 autorun.inf 파일을 이용하여 자동 실행하고, 다른 악성코드를 다운로드 하거나 개인정보 유출 등의 악의적인 행동을 유발하며, 급속히 확산되고 있다.

(2) 네트워크

ARP Spoofing, 게시판, SNS 등과 같이 네트워크를 이용하여 악성코드가 전파되는 사례는 다음 표와 같다[29].

[표 2-9] 네트워크를 이용하여 전파하는 사례

날 짜	세부내용
2010.09	[온라인게임계정 짝퉁이 탈취 악성코드 등장] 안철수 연구소에서 최근 온라인 게임 계정을 탈취하는 악성코드가 일부 웹사이트를 통해 확산 중이라고 7일에 경고하였다. 이 악성코드는 ARP Spoofing을 통해 감염된 개인 PC와 해당 PC가 속해있는 네트워크의 다른 PC까지도 전파되고 있어 사용자들의 각별한 주의가 요구되고 있다.

(3) 다운로드

공격자가 P2P, 웹하드, FTP 등에 악성코드가 감염된 파일을 업로드하고, 사용자가 그 파일을 자신의 시스템에 다운로드하여 악성코드가 전파되는 사례는 다음 [표2-10]과 같다[30].

[표 2-10] 다운로드를 통해 전파하는 사례

날 짜	세부내용
2010.09	[아이폰 탈옥 도구 위장 ‘악성코드’ 유포] 아이폰 ‘iOS 4.1’ 버전의 탈옥 도구를 위장한 파일에 내장되어 있는 악성코드가 발견되었다. 이 악성코드는 사용자의 패스워드를 빼내서 특정 서버로 전송하는 역할을 수행한다고 27일 밝혔다. 주로 P2P 파일 공유 프로그램과 블로그 등을 통해 유포되고 있으며, 해당 프로그램을 다운받은 사용자들이 가짜 아이폰 탈옥 프로그램을 실행하게 되면 내포되어 있는 악성코드도 함께 실행된다.

[표 2-11] 전자우편을 이용하여 전파하는 사례

날 짜	세부내용
2008.06	[e메일 통한 ‘고소장접수결과보고.zip’파일] ‘Adobe사 고소장 접수에 대한 안내문’이라는 제목의 전자우편이 특정 메일주소로 대량 발송되었다. 본 메일에는 ‘고소장접수결과보고.zip’라는 제목의 파일이 첨부되어 있다. 압축을 풀면 ‘고소장접수결과보고.exe’라는 실행파일이 있는데 이 파일을 실행하게 되면 nbjs.dll이란 파일이 만들어져서 DDoS공격에 이용되는 것이다. 개인정보 유출과 관련된 집단 소송이 인터넷을 통해 확산되고 있는 것을 악용한 악성코드인 것이다.
2011.06	[신용카드 한도초과 안내메일로 위장한 악성코드 유포] 악성코드를 첨부한 메일을 발송하는 사례가 다시 증가하고 있는 추세이다. 이번에 발견된 악성코드를 첨부한 악성 스팸 메일(Spam Mail)은 신용카드가 한도를 초과하였다는 메일로 위장하고 있다. 해당 악성 스팸 메일에 첨부된 악성코드를 실행하게 될 경우에는 특정 시스템으로부터 금전을 목적으로 한 허위 시스템 복구 프로그램을 다운로드 및 실행하게 되고, 이를 통해 개인 정보가 유출되게 된다.

(4) 전자우편

전자우편을 이용하여 악성코드가 감염된 파일을 첨부하여 보내거나 전자우편의 제목을 수신자의 호기심을 자극하는 문구가 들어가게 하여 사용자가 악성 웹사이트에 접속하도록 유도하는 등의 방법으로 악성코드를 전파하는 사례는 다음 [표 2-11]과 같다[31][32].

(5) 보안 취약성

웹브라우저의 취약성, SQL 서버의 취약성, 소프트웨어 취약성 등의 보안 취약성을 이용하여 악성코드를 전파하는 경우가 있는데, 이러한 방법에 대한 사례는 다음 [표 2-12]와 같다[33].

[표 2-12] 취약성을 이용하여 전파하는 사례

날 짜	세부내용
2008.06	[어도비 플래시 취약점 악용한 악성코드] 어도비 플래시의 취약점을 이용한 악성코드가 유포되고 있다. 이러한 악성코드 때문에 국내 불특정 다수의 웹사이트들이 시달리고 있다. 어도비 플래시는 인터넷상에서 플래시 파일(*.swf)을 볼 수 있게 하는 프로그램으로 대부분의 인터넷 사용자의 PC에 깔려 있다.
2008.06	[MS 어플리케이션 취약점 악용 악성코드] 4월 초부터 계속해서 웹사이트에서 MS 어플리케이션 취약성을 이용한 공격이 급증하고 있다. SQL-Injection 기술을 이용하여 사이트에 악의적인 프레임을 삽입하는 수법을 사용한다. 해당 악성코드에 감염되면 다른 사이트로 연결되어 키로깅 프로그램이 다운되게 된다. 그러면 사용자의 PC에 설치되어 키보드 해킹을 하여 개인 정보가 유출되게 될 가능성이 높다.

(6) 인스턴트 메시징 프로그램

메신저등과 같은 인스턴트 메시징 프로그램을 이용하여 악성코드가 전

파되고 있다. 이와 관련된 사례는 다음 [표 2-13]과 같다[34][35].

[표 2-13] 인스턴트 메시징 프로그램을 이용하여 전파하는 사례

날 짜	세부내용
2010.05	[해외에서 발견된 야후 메신저로 전파되는 악성코드] 시만텍(Symantec)에서 블로그 "New Yahoo! Messenger worm"을 통해서 야후 메신저 프로그램을 이용해 전파되는 악성코드들이 다수 발견되었다고 밝혔다. 이번에 발견된 악성코드는 감염된 시스템에서 야후 메신저를 사용하고 있으면 친구 리스트에 등록되어 있는 모든 사람들에게 특정 웹사이트의 링크를 전달하여 감염을 시도하는 특징을 가지고 있다.
2010.07	[네이트온 메신저로 전파되는 악성코드 대량 유포] 네이트온 메신저로 전파되고 있는 악성코드가 대량 유포되고 있어 각별한 주의가 필요하다. 특히 이번에 발견된 악성코드들은 기존의 악성코드들의 형태와는 다르게 보안 제품의 진단을 우회한 것으로 확인되어 주의가 필요하다.

(7) 그 외

위의 전파 방법 외에도 최근에 새롭게 등장하는 전파 방법이 있다. 이와 관련된 사례는 다음 [표 2-14]와 같다[36][37].

[표 2-14] 새로운 방식을 이용하여 전파하는 사례

날 짜	세부내용
2010.11	[페이스북 채팅 메시지로 유포되는 악성코드] 국내 페이스북(Facebook) 사용자들 사이에서 채팅(Chatting) 메시지를 통해 악성코드가 유포되고 있다는 사실이 밝혀졌다. 이번 악성코드는 채팅 메시지를 통해 악성코드를 다운로드 하는 악성 웹페이지 링크를 전송하여 상대방이 해당 링크를 클릭하여 다운받으면 감염되는 형태를 띠고 있다.
2011.02	[국내서도 페이스북 이용 악성코드 유포] 안철수 연구소의 14일 발표에 따르면, 페이스북을 이용한 악성코드가 유포되기 시작했다고 한다. 페이스북 담벼락에 악성코드 다운로드를 유도하고, 악성 웹사이트 링크가 포함된 게시물들이 유포되기 시작했다고 밝혔다.

2. 전파경로별 대응전략

본 절에서는 앞에서 악성코드 전파 방법을 유형별로 분류한 것에 따라 악성코드 전파에 대한 대응기술을 유형별로 설명한다. 전파 방법은 특별히 전문적인 기술을 필요로 하지 않아 특별히 기술적인 대응방법이 없다. 다음에 나오는 유형별 대응기술 설명을 보면 유형별 대응기술 결국 한 지점으로 모인다는 것을 알 수 있다. 유형별 대응기술은 공통적으로 사용자의 주의와 관심, 그리고 노력이 필요하며 항상 백신 검사는 필수적으로 하고 정기적으로 보안패치를 설치해야 한다는 사항이 포함된다.

가. 저장매체

저장매체를 통해 전파되는 악성코드를 예방하기 위해서는 다음과 같이 대응해야 한다. 저장매체 중 하나인 플로피 디스크에 부트 바이러스가 감염되어 있는 경우 시스템 설정에 들어가서 부팅 우선 순서를 플로피 디스크가 아닌 하드디스크로 설정함으로써 예방할 수 있다. 플로피 디스크는 컴퓨터에 실행되기 전에 먼저 바이러스 감염여부를 검사해야 하며, 컴퓨터의 전원을 켜기 전에 플로피 디스크가 컴퓨터에 꽂혀있지 않은지 확인하여 자동으로 실행되지 않도록 해야 한다.

CD는 불법 CD에 바이러스가 감염되어 있는 경우가 있어 되도록 정품 CD를 사용하도록 하며, 간혹 정품 CD에도 바이러스가 발견되는 경우도 있으므로 CD를 재생하기 전에 바이러스 검사를 하도록 한다. 이외에도 어떤 저장매체를 사용하든 저장매체를 컴퓨터에 꽂거나 파일을 복사하려고 하면 그전에 반드시 바이러스 감염여부를 확인해야 한다.

나. 네트워크

네트워크를 통해 감염되는 악성코드를 막기 위해서는 의심되는 사이트

를 함부로 접속하지 않고, 개인 방화벽이나 침입탐지 시스템과 같은 네트워크 보안 장비를 이용한다. 네트워크 관리자는 포트를 잘 관리하여 악의적인 목적으로 외부에서 내부 시스템으로 침입하지 못하도록 한다. 사용자는 내부에 침투된 악성코드에 대비하기 위해 자신의 컴퓨터에 백신을 설치해야 놓아야 한다.

다. 다운로드

다운로드를 통해 악성코드가 유포되는 것을 막기 위해서는 다음과 같은 대응법이 있다. 공유 폴더는 악성코드 감염 및 보안상으로 위험하므로 되도록 사용을 자제하며 파일을 다운로드 할 때에는 신뢰할 수 있는 경로를 통해 다운로드 받아야 한다. 그리고 다운 받을 때는 반드시 백신으로 검사해야 한다.

라. 전자우편

전자우편을 통해 전파되는 악성코드를 예방하기 위해서는 확인되지 않은 전자우편이나 의심이 가는 첨부파일을 열어보지 말아야 한다. 사용자의 컴퓨터는 해당 운영체제 업체에서 제공하는 업데이트를 통해 항상 최신 버전을 유지해야 한다. 첨부파일을 다운 받아야 하는 상황이면 전자우편 사이트에서 제공하는 백신으로 첨부파일을 검사해야 한다. 이러한 서비스는 전자우편 서버에서 데이터가 사용자에게 전송되기 전에 서버 내의 백신으로 검사해주는 것이다.

마. 보안 취약성

보안 취약성을 통해 전파되는 악성코드를 예방하기 위해서는 다음과 같은 방법이 있다. 모든 보안 취약성의 가장 일반적인 대응 방법은 해당 제품의 최신 버전의 보안 패치를 업데이트 하는 것이다. 즉, 어도비플래

시 취약성을 이용하는 공격을 막기 위해서는 어도비 플래시 플레이어 최신 버전을 업데이트해야 하고, 윈도우 취약성을 막기 위해서는 최신 윈도우 보안패치를 업데이트해야 한다[8].

바. 인스턴트 메시징 프로그램

인스턴트 메시징 프로그램을 이용하여 악성코드가 유포되는 것을 사전에 예방하기 위해서는 다음과 같은 방법이 있다. 메신저는 주기적으로 기능 향상과 해킹이나 악성코드의 공격에 대비하기 위해서 프로그램을 개선하여 업그레이드된 버전을 사이트에 올려놓는다. 이러한 최신 버전을 검증된 경로로 신속하게 업데이트 해주도록 한다. 특히, 메신저를 통해 상대방이 URL 정보나 파일을 전송할 경우 주의해서 보아야 한다.

사. 그 외

악성코드의 종류가 날이 갈수록 많아지고, 악성코드 감염 경로가 매우 다양해지고 있다. 이에 대응하기 위해서는 항상 최신 보안 정보에 관심을 가지고, 자신의 컴퓨터에 설치된 백신이 항상 최신 버전일수 있도록 업데이트를 하며, 최신 패치 등을 생활화해야 한다.

제 4 절 레지스트리 악용 방법

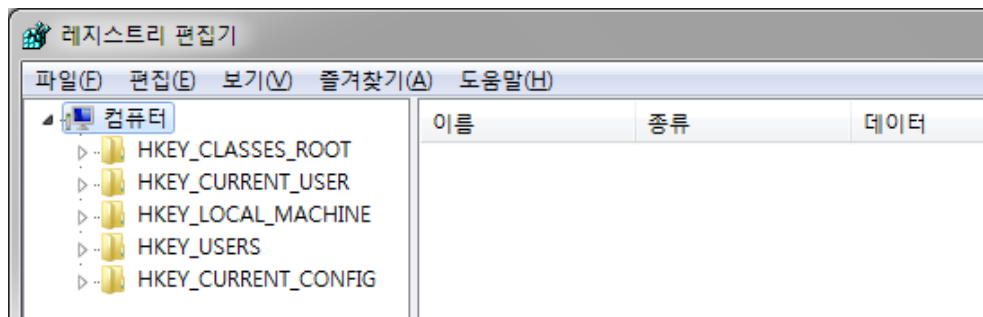
1. 레지스트리 개요

레지스트리는 윈도우를 실행하는데 필요한 환경설정 데이터를 말하며, 윈도우의 모든 설정을 담고 있는 중앙 저장소를 말한다. 시스템에 설치되어 있는 랜카드나 디스플레이 어댑터, 응용프로그램에 대한 정보, 디바

이스, 사용자 설정, 윈도우 자체적인 설정, 인터넷 익스플로러 설정 등의 정보가 담겨 있다. 윈도우를 설치하거나 추가적인 장치 혹은 프로그램을 설치할 때마다 관련된 설정 값들이 레지스트리에 저장되고 그 정보가 추가된다. 따라서 레지스트리를 통해 시스템에 대한 중앙 집중화 관리가 가능하고, 각 사용자에게 대한 설정과 권한에 대한 관리 역시 가능하다. 이러한 시스템의 정보를 담고 있는 레지스트리는 다음과 같이 구성된다.

- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_CONFIG

다섯 개의 루트키 각각에는 하위키가 존재하는데 하위키는 자신이 직접 값을 가지거나 실제 값을 갖는 다른 하위키를 포함하는 용도로 사용된다. 이는 파일시스템의 하위 디렉토리와 같은 맥락으로 사용될 수 있다. 최하위키는 레지스트리 값을 가지는데, 레지스트리 값은 보통 이름, 종류, 데이터의 세 부분으로 구성되어 있다.



(그림 2-28) 레지스트리 구조

가. HKEY_CLASSES_ROOT

HKEY_CLASSES_ROOT는 파일 확장자와 각 파일에 맞는 애플리케이션

선에 대한 정보, 프로그램간의 연결 정보, 마우스 오른쪽 버튼의 등록정보 등에 관련된 정보를 가지고 있다. 윈도우에서 사용되는 모든 형식의 파일 확장자가 디렉토리 형태의 서브키로 구성되어 있다.

나. HKEY_CURRENT_USER

HKEY_CURRENT_USER는 현재 로그인되어 있는 사용자에 대한 정보와 사용자의 초기 설정 파일에 대한 구성정보를 갖고 있다. 사용자의 배경화면, 디스플레이 설정이나 단축아이콘의 정보가 담겨있다. 한 대의 컴퓨터를 여러 사용자가 이용할 경우 각 사용자에 대한 정보는 HKEY_USERS에 저장되고, 현재 로그인한 사용자에 대한 정보는 HKEY_CURRENT_USER에 저장된다. 컴퓨터의 사용자가 한 명이라면 HKEY_USERS/.default와 거의 동일한 내용을 갖는다.

다. HKEY_LOCAL_MACHINE

HKEY_LOCAL_MACHINE은 시스템 하드웨어와 소프트웨어 환경에 대한 전반적인 정보들을 갖고 있다. 윈도우의 실행에 필요한 모든 하드웨어 설정에서부터 그 하드웨어가 사용하고 있는 드라이버 정보가 저장되어 있다. 제어판의 장치관리자 내용 등 가장 중요한 부분을 차지하며, 컴퓨터에 등록된 모든 사용자에게 동일하게 나타난다.

라. HKEY_USERS

HKEY_USERS는 각각의 사용자들에 대한 정보를 갖고 있다. 컴퓨터를 공유하는 각 사용자들의 윈도우 환경에 대한 여러 설정 사항을 저장한다. HKEY_CURRENT_USER 서브트리에 대한 포인터와 .default 프로파일을 갖고 있으며 계정이 하나인 경우 .default 프로파일만 존재한다. 여러 사용자가 사용하는 경우 각 사용자의 정보는 SID 명으로 존재하며

SID 하위 디렉토리의 내용은 HKEY_CURRENT_USER 설정과 동일하다.

마. HKEY_CURRENT_CONFIG

HKEY_CURRENT_CONFIG는 하드웨어 환경설정에 대한 정보들을 갖고 있다. 즉, 디스플레이나 프린터 등에 대한 설정 내용이 저장되어 있다.

2. 레지스트리 악용 방법

악성코드는 실행되거나 사용자 시스템에 설치될 때 레지스트리와 관련된 행위를 한다. 악성코드는 레지스트리의 특정 위치에서 생성, 변경, 삭제 등의 행위를 통해 서비스에 등록시키거나 혹은 윈도우 시작 시 자동으로 악성코드가 실행될 수 있게 한다. 또한 레지스트리의 존재를 확인하고 레지스트리의 값을 참조하는 행위를 수행하기도 한다. 다음은 악성코드가 수행하는 레지스트리 관련 행위의 목록이다.

- 서비스 등록
- 부팅 시 자동 실행
- 파일 확장자 연결
- 윈도우 탐색기(Explorer)와 관련된 행위
- 인터넷 익스플로러 관련 행위
- 윈도우 설정과 관련된 행위
- 네트워크 설정 관련 행위
- 레지스트리 존재 여부 확인
- 레지스트리 값 참조

가. 서비스 등록

악성코드가 특정 레지스트리를 수정하거나 특정 위치에 레지스트리 값을 추가시킴으로써 자기 자신을 서비스로 등록하는 행위를 수행한다. 서

비스로 등록되는 악성코드는 부팅 시 마다 실행된다. 악성코드가 서비스로 등록하는 레지스트리의 공통적인 위치는 다음과 같다.

- o HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services
- o HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services
- o HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

다음 표는 악성코드가 윈도우 서비스로 등록하여 윈도우 시작 시 자동으로 실행되도록 하는 레지스트리의 목록이다.

[표 2-15] 서비스 등록을 위해 악용되는 레지스트리

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\NtmsSvcStart = 0x2
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\NtmsSvcType = 0x110
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsysType = 0x1
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsysErrorControl = 0x1
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsysStart = 0x1
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeListenerType = 0x20
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeListenerStart = 0x2
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeListenerErrorControl = 0x1
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeListenerImagePath = "%systemroot%\system32\svchost.exe -k

homelisten"

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeL
isten\SecuritySecurity = 01 00 14 80 90 00 00 00 ...

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeL
istenObjectName = "localsystem"

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeL
istenDisplayName = "home group listener..."

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\HomeL
isten\ParametersServiceDll = "[악성코드].dll"

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\[실행파
일명]Type = 0x1

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\[실행파
일명]Start = 0x3

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\[실행파
일명]ErrorControl = 0x1

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\[실행파
일명]ImagePath = "\\??\[실행폴더]\[악성코드].exe"

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\[실행파
일명]\Enum0 = "root\legacy_test\0000"

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\resdr32
Type = 0x1

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\resdr32
ErrorControl = 0x1

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\resdr32
Start = 0x1

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FADescription = "43b182fe"

HKEY_CURRENT_USER\SYSTEM\CurrentControlSet\Services\654C

```
A2FADescription = "43b182fe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FADisplayName = "654ca2fa"
HKEY_CURRENT_USER\SYSTEM\CurrentControlSet\Services\654C
A2FADisplayName = "654ca2fa"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FADescription = "43b182fe"
HKEY_CURRENT_USER\SYSTEM\CurrentControlSet\Services\654C
A2FADescription = "43b182fe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FAObjectName = "localsystem"
HKEY_CURRENT_USER\SYSTEM\CurrentControlSet\Services\654C
A2FAObjectName = "localsystem"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FAStart = 0x2
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FAType = 0x10
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\AVPsys
ImagePath = "\\??\c:\windows\system32\drivers\cdaudio.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\sysdrv3
2ImagePath = "\\??\c:\windows\system32\drivers\sysdrv32.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Windo
wseImagePath = "c:\program files\common files\microsoft
shared\msinfo\qweref.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\netsikI
magePath = "\\??\c:\windows\system32\drivers\netsik.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\amd64s
iImagePath = "\\??\ 윈도우 시스템 폴더\drivers\amd64si.sys"
```

```

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\winrare
seImagePath = "윈도우 시스템 폴더\nddend26.bat"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\AVPsys
ImagePath = "\\??\C:\WINDOWS\system32\drivers\cdaudio.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\BackGr
ound switchImagePath = "c:\windows\system32\regedit32.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\awpIm
agePath = "c:\windows\system32\awp.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\BifroseI
magePath = "윈도우 프로그램 파일 폴더\internet
explorer\bifrose.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\explore
rImagePath = "c:\program files\internet explorer\explorer.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsy
sImagePath = "\\??\c:\windows\system32\drivers\klif.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Applica
tion Experience ServerImagePath =
"c:\windows\system32\axsvr.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\wuause
rvImagePath = "%fystemroot%\system32\svchost.exe -k netsvcs"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\BITSIm
agePath = "%fystemroot%\system32\svchost.exe -k netsvcs"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\emxxkl
ImagePath = "%systemroot%\system32\svchost.exe -k netsvcs"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\srumpI
magePath = "\\??\ 윈도우 시스템 폴더\04.tmp"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsy
sImagePath = "\\??\ 윈도우 시스템 폴더\drivers\klif.sys"

```



```

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\AVPsys
ImagePath = "\\??\c:\windows\system32\drivers\cdaudio.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\AVPsys
ImagePath = "\\??\윈도우 시스템 폴더\drivers\cdaudio.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\GmPnS
NImagePath = "윈도우 시스템 폴더\xnpydd.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\httpssli
magePath = "[실행폴더]\hp.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsys
sImagePath = "\\??\c:\windows\system32\drivers\vga.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\ICFIma
gePath = "윈도우 시스템 폴더\svchost.exe:exe.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\biaiefqi
ImagePath = '윈도우 시스템 폴더\jegyssooc.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SYNSI
magePath = "윈도우 폴더\system\svchost.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\System
Locator ServiceDisplayName = "system restore locator"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\System
Locator ServiceObjectName = "localsystem"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\System
Locator ServiceDescription = "provides and enables starting
processes under alternate credentials. if this service is stopped, thi"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\System
Locator ServiceImagePath = "윈도우 시스템 폴더
\drivers\systemdriver.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\654CA2
FAImagePath = "윈도우 시스템 폴더\efef3cb8.exe -k"

```

```
HKEY_CURRENT_USER\SYSTEM\CurrentControlSet\Services\654C
A2FAImagePath = "윈도우 시스템 폴더\efef3cb8.exe -k"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SSLSIm
agePath = "%systemroot%\system32\svchost.exe -k netsvcs"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\brtimty
ImagePath = "윈도우 시스템 폴더\02.tmp"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\asp.netI
magePath = "윈도우 프로그램 파일 폴더\common files\microsoft
shared\msinfo\asp.net"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Networ
k DDBImagePath = "윈도우 시스템 폴더\server.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\NetDD
SImagePath = "윈도우 시스템 폴더\[랜덤한파일명6자리].exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\IrmonI
magePath = "%systemroot%\system32\svchost.exe -k netsvcs"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\System
Locator ServiceImagePath = "윈도우 시스템 폴더
\drives\systemdriver.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Nationa
lrwiImagePath = "윈도우 시스템 폴더\svc[랜덤한문자5자리].exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\IasImag
ePath = "%systemroot%\system32\svchost.exe -k netsvcs"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsy
sImagePath = "\\??\c:\windows\system32\drivers\klif.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\SafeBoo
t\MinimalSVCWINSPOOL\Default = "service"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Shared
Access\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplic
```

```

ations\ListC = success "c:\windows\system\smc.exe*:microsoft
enabled"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsy
sImagePath = "\\??\c:\windows\system32\drivers\klif.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SharedA
ccess\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplic
ations\ListC = success
"c:\windows\system\netmon.exe*:microsoft enabled"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\sysdrv3
2ImagePath = "\\??\c:\windows\system32\drivers\sysdrv32.sys"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Shared
Access\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplic
ations\ListC = success
"\\??\c:\windows\system32\winlogon.exe*:enabled:@shell32.dll,-1"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\NtmsSv
c\ParametersServiceDll = "%systemroot%\system32\ntmssvr.dll"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\CreateP
rocessImagePath = "윈도우 폴더\system\svchost.exe"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Windo
wsDownImagePath = "윈도우 시스템 폴더\servlet.exe"

```

나. 부팅 시 자동 실행

악성코드가 레지스트리의 특정 위치에 레지스트리 값을 새로 등록하거나 기존의 레지스트리 값을 조작하여 다음번 부팅 시 악성코드가 자동으로 실행되도록 한다. 악성코드가 자동 실행을 위해 값을 등록하는 레지스트리의 공통적인 위치는 다음과 같다.

o HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\Current

- Version\Run
- o HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
- o HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

자동 실행을 위해 준비된 레지스트리에 악성코드 실행파일을 연결시키거나 dll 파일을 후킹 하는 등 레지스트리 등록 또는 변조를 통해 다양한 방법으로 악성코드 자동실행을 위한 환경을 구축한다. 다음 표는 부팅 시 자동 실행을 위해 악용되는 악성코드의 레지스트리다.

[표 2-16] 자동 실행을 위해 악용되는 레지스트리

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\KAVsys\ImagePath= 윈도우 시스템 폴더\drivers\klif.sys
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Runkvasoft = 윈도우 시스템 폴더\kvosoft.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\tipguarder.exe
HKEY_CLASSES_ROOT\CLSID\{D032570A-5F63-4812-A094-87D007C23012}
HKEY_CLASSES_ROOT\ieguarder.TIEAdv
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Runkvasoft = 윈도우 시스템 폴더\kvosoft.exe
HKEY_CLASSES_ROOT\CLSID\{1F88A6F5-908C-4C28-9A81-829953C5F5C5}
HKEY_CLASSES_ROOT\gs23d1
HKEY_CLASSES_ROOT\Interface\{06360872-0310-49C1-8EDA-953E73941E3E}
HKEY_CLASSES_ROOT\Interface\{419803E0-EBB5-418E-BCDD-8EA63

647EC5E}
HKEY_CLASSES_ROOT\Kioll
HKEY_CLASSES_ROOT\TypeLib\{10026069-7A5F-4531-811E-C8DF20643BEE}
HKEY_CURRENT_USER\SOFTWARE\MicroAV
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ANTIVIRUSDropper
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\WinFix Master
HKEY_CURRENT_USER\SOFTWARE\WinDV
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\WinDVDDownloader
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\Run\SystemManager= 시스템 루트
\work\samples\ec07040700004_lsh\08.exe
HKEY_CURRENT_USER\SOFTWARE\AV2010
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Windows Gamma Display
HKEY_CLASSES_ROOT\.key
HKEY_CURRENT_USER\SOFTWARE\MSa
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ANTIVIRUS
HKEY_CURRENT_USER\SOFTWARE\PCMightyMax
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{94D5AF0F-E6EE-4A75-BE31-9C9C9A87AD45}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\PCMMRealtimeDropper
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current

tVersion\Run\PCMMRealtime
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ThemeManager\SystemID"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\TotalSecure2009
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\MyLibHelper
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\SharedApplication
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Systemhost
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\MyLibHelper
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\SharedApplication
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Systemhost
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\kamssoft = 윈도우 시스템 폴더\ckvo.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ "{***}"="%SYSTEMDRIVE%\Skypei.exe"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ "{***}"="%SYSTEMDRIVE%\Winamp6.exe"
HKEY_CURRENT_USER\SOFTWARE\VirRL2009
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\VirRL2009
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VirRL2009

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Warning CenterClicker

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunMSN = 윈도우 폴더\k8svr.exe

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\BoozizoaImagePath = 윈도우 시스템 폴더\boozizoa.exe

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BoozizoaImagePath = 윈도우 시스템 폴더\boozizoa.exe

HKEY_CLASSES_ROOT\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}\Shell\Secure\Clean

HKEY_CURRENT_USER\SOFTWARE\PrivacyWarrior

HKEY_CURRENT_USER\SOFTWARE\Purchased Products

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\{"***"}="%PFDIR%\PrivacyWarrior\plug\GDCW.exe"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\PrivacyWarrior

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\SalestartDropper

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\tvchost

HKEY_CURRENT_USER\SOFTWARE\AntiMalwareGuard

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\AntiMalwareGuard

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\AntiMalwareGuard

HKEY_CLASSES_ROOT\CLSID\{037C7B8A-151A-49E6-BAED-CC05FCB50328}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\AntiMalwareGuard

tVersion\Run\ "{**}"="%SYSDIR%\ieexplorer32.exe"
HKEY_CURRENT_USER\SOFTWARE\4992934617530433645364505048
2366
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\49929346175304336453645050482366
HKEY_CURRENT_USER\SOFTWARE\4992934617530433645364505048
2366
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\49929346175304336453645050482366
HKEY_CURRENT_USER\SOFTWARE\fc
HKEY_CURRENT_USER\SOFTWARE\LanConfig
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\psbeajtd
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\psbeajtd
HKEY_CLASSES_ROOT\.key
HKEY_CURRENT_USER\SOFTWARE\MicroAV
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ANTIVIRUS
HKEY_CLASSES_ROOT\AllFilesystemObjects\shellex\ContextMenuHandlers\pcprosd.dll
HKEY_CLASSES_ROOT\CLSID\Value* **** Data* ***
HKEY_CLASSES_ROOT\CLSID\{0FEBB44D-9C2A-4C8A-81C1-28F4904895A0}
HKEY_CLASSES_ROOT\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}\shellex\ContextMenuHandlers\pcprosd.dll
HKEY_CURRENT_USER\SOFTWARE\PC Clean Pro
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\49929346175304336453645050482366


```

tVersion\Run\PC Clean Pro
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ShellExtensions\Approved\{***}="pcprosd.dll"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\PCCleanPro
HKEY_CLASSES_ROOT\AllFilesystemObjects\shellex\ContextMenuHandlers\pcprosd.dll
HKEY_CLASSES_ROOT\CLSID\{0FEBB44D-9C2A-4C8A-81C1-28F4904895A0}
HKEY_CLASSES_ROOT\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}\shellex\ContextMenuHandlers\pcprosd.dll
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ShellExtensions\Approved\{0FEBB44D-9C2A-4C8A-81C1-28F4904895A0}
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\kvasoft = 윈도우 시스템 폴더\kvasoft.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunASocksrv = "socksa.exe"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunSVCHOST =
c:\samples\ac07012400021-000002_b070124c_11_mdm.exe
HKEY_CLASSES_ROOT\CLSID\{49E0E0F0-5C30-11D4-945D-000000000000}\InprocServer32(Default) = 윈도우 시스템 폴더\lnkchg.dll
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\RunHFDF = 윈도우 시스템 폴더\hfmd00001.exe
HKEY_CURRENT_USER\SOFTWARE\XP_Antispyware
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\PC Clean Pro

```

```

tVersion\Uninstall\XP_AntiSpyware
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\XP
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Doctor Antivirus 2008
HKEY_CURRENT_USER\SOFTWARE\AntiSpywareExpert
HKEY_CURRENT_USER\SOFTWARE\{5222008A-DD62-49c7-A735-7BD18ECC7350}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\AntiSpywareExpert
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\kvasoft = 윈도우 시스템 폴더\kvasoft.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run= 윈도우 프로그램 파일 폴더\windows
nt\system\wdfmgr.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Ras.exeDebugger
= "c:\program files\common files\microsoft shared\dcayaal.exe"

```

다. 파일 확장자 연결

악성코드가 레지스트리의 특정 위치에 레지스트리 값을 등록하여 악성코드를 특정한 확장자를 가진 파일과 연결함으로써 해당 확장자의 파일이 사용되면 악성코드가 자동으로 실행되도록 한다. 이 때 사용되는 확장자로는 .exe, .bat, .com, .pif, .scr, .hta, .reg, .txt 등이 있다. 파일 확장자 연결에 사용되는 레지스트리의 공통적인 위치는 다음과 같다.

- o HKEY_CLASSES_ROOT\exefile\shell\open\command
- o HKEY_CLASSES_ROOT\batfile\shell\open\command

- o HKEY_CLASSES_ROOT\comfile\shell\open\command
- o HKEY_CLASSES_ROOT\piffile\shell\open\command
- o HKEY_CLASSES_ROOT\scrfile\shell\open\command
- o HKEY_CLASSES_ROOT\htafile\shell\open\command
- o HKEY_CLASSES_ROOT\regfile\shell\open\command
- o HKEY_CLASSES_ROOT\txtfile\shell\open\command

다음 표는 악성코드가 파일 확장자를 연결하여 특정 파일이 실행될 때 악성코드 또한 자동으로 실행되게 하는 레지스트리의 목록이다.

[표 2-17] 파일 확장자를 연결하는 레지스트리

HKEY_CLASSES_ROOT\exefile\shell\open\command	=
C:\WINDOWS\SYSTEM\winsvrc.exe "%1" %*	
HKEY_CLASSES_ROOT\exefile\shell\open\command	=
C:\Winnt\System32\winsvrc.exe "%1" %*	
HKEY_CLASSES_ROOT\exefile\shell\open\command	=
C:\WINDOWS\SYSTEM\wintask.exe "%1" %*	
HKEY_CLASSES_ROOT\exefile\shell\open\command=%1" %*	
HKEY_CLASSES_ROOT\exefile\shell\open\command->c:\recycled\[5자, 또는 그 이상의 랜덤한 파일명]" %1 %*	
HKEY_CLASSES_ROOT\batfile\shell\open\command	= raVe.exe
HKEY_CLASSES_ROOT\exefile\shell\open\command	= raVe.exe
HKEY_CLASSES_ROOT\comfile\shell\open\command	= raVe.exe
HKEY_CLASSES_ROOT\piffile\shell\open\command	= raVe.exe
HKEY_CLASSES_ROOT\scrfile\shell\open\command	= raVe.exe
HKEY_CLASSES_ROOT\batfile\shell\open\command (Default)	= 랜덤한 문자.exe "%1" %*
HKEY_CLASSES_ROOT\comfile\shell\open\command (Default)	= 랜덤한 문자.exe "%1" %*
HKEY_CLASSES_ROOT\exefile\shell\open\command (Default)	= 랜

```

덤한 문자.exe "%1" %
HKEY_CLASSES_ROOT\piffile\shell\open\command (Default) =랜
덤한 문자.exe "%1" %
HKEY_CLASSES_ROOT\regfile\shell\open\command (Default) =랜
덤한 문자.exe "%1" %
HKEY_CLASSES_ROOT\scrfile\shell\open\command (Default) =랜
덤한 문자.exe showerror
HKEY_CLASSES_ROOT\scrfile\shell\open\command (Default) =랜
덤한 문자.exe "%1" /S
HKEY_CLASSES_ROOT\scrfile\shell\open\command (Default) =랜
덤한 문자.exe showerror
HKEY_CLASSES_ROOT\scrfile\shell\open\command (Default) =랜
덤한 문자.exe "%1"
HKEY_CLASSES_ROOT\txtfile\shell\open\command =
\%SystemRoot%\NOTEPAD.EXE %1
HKEY_CLASSES_ROOT\txtfile\shell\open\command = C:\윈도우즈
시스템폴더\notepad.exe
HKEY_CLASSES_ROOT\exefile\shell\open\command = C:\윈도우
즈 시스템폴더\KNREL32.exe
HKEY_CLASSES_ROOT\comfile\shell\open\command = C:\윈도우
즈 시스템폴더\KNREL32.exe

```

라. 윈도우 탐색기(Explorer)와 관련된 행위

악성코드가 레지스트리의 특정 위치에 레지스트리 값을 추가함으로써 윈도우 탐색기의 폴더 옵션 중 '숨김 파일 및 폴더 표시 안함'과 '알려진 파일 형식의 파일 확장명 숨기기'등 보기와 관련된 설정을 변경하여 트로이 목마의 발견을 어렵게 한다. 윈도우 탐색기의 보기 설정을 변경하

는 레지스트리의 공통적인 위치는 다음과 같다.

- o HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
- o HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden

또한 브라우저 도우미 개체(Browser Helper Object, BHO)에 등록된 DLL 형태의 스파이웨어 또는 애드웨어를 윈도우 탐색기와 인터넷 익스플로러에 연결하여 윈도우 시작 시 윈도우 탐색기에 의해 자동으로 실행되게 한다. 이때 변경되거나 새로이 등록되는 레지스트리의 키는 해당하는 악성코드에 따라 다양한 위치에서 악용된다. 다음 표는 윈도우 탐색기와 관련된 행위에 악용되는 레지스트리다.

[표 2-18] 윈도우 탐색기 관련 행위에 악용되는 레지스트리

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Hidden = 0x2
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\AdvancedSuperHidden = 0x0
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ShowSuperHidden = 0x0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALLCheckedValue = 0x0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALLRegPath = "software\\microsoft\\windows\\currentversion\\explorer\\advanced"
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curre

ntVersion\Explorer\Advanced\Folder\Hidden\SHOWALLText =
@shell32.dll,-30500
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALLType =
radio
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALLValueName
= hidden
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALLHelpID =
shell.hlp#51105
HKEY_CLASSES_ROOT\AppID\IEDefender.DLL
HKEY_CLASSES_ROOT\AppID\{3C40236D-990B-443C-90E8-B1C07BCD4A68}
HKEY_CLASSES_ROOT\CLSID\{FC8A493F-D236-4653-9A03-2BF4FD94F643}
HKEY_CLASSES_ROOT\IEDefender.IEDefenderBHO
HKEY_CLASSES_ROOT\IEDefender.IEDefenderBHO.1
HKEY_CLASSES_ROOT\Interface\{7BC7565C-5062-43CE-8797-DC2C271140A9}
HKEY_CLASSES_ROOT\TypeLib\{705FD64B-2B7B-4856-9337-44CA1DA86849}
HKEY_CLASSES_ROOT\CLSID\{F9BA1AA9-CAD4-4C14-BDE6-922DF5F6F38}
HKEY_CLASSES_ROOT\TypeLib\{13539089-270A-449C-8C6F-DEEBDA8C2A2E}
HKEY_CLASSES_ROOT\CLSID\{500BCA15-57A7-4eaf-8143-8C619470B13D}

HKEY_CLASSES_ROOT\TypeLib\{9233C3C0-1472-4091-A505-5580A23BB4AC}
HKEY_CLASSES_ROOT\XML.XMLHKEY_CLASSES_ROOT\XML.XML.1
HKEY_CLASSES_ROOT\CLSID\{FFFFFFFFC-A520-4778-99FE-AACCB002669E}
HKEY_CLASSES_ROOT\DCPQ4.DCPQ4ObjHKEY_CLASSES_ROOT\DCPQ4.DCPQ4Obj.1
HKEY_CLASSES_ROOT\Interface\{528736A2-6FE3-46B0-87FD-D076068779B7}
HKEY_CLASSES_ROOT\TypeLib\{50A4EA11-9855-41A3-9CF6-99CD6CDF3F75}
HKEY_CLASSES_ROOT\CLSID\{59B77AA4-609D-499b-BD27-BC903E75E9EF}
HKEY_CLASSES_ROOT\Interface\{C71A6E3B-110F-4E8C-B1C4-5631E02D626F}
HKEY_CLASSES_ROOT\TypeLib\{8D8900CD-BD56-4983-8642-13F5848D9511}
HKEY_CLASSES_ROOT\winmgmt.winmgmtObjHKEY_CLASSES_ROOT\winmgmt.winmgmtObj.1
HKEY_CLASSES_ROOT\CLSID\{5C78E2DB-5AFC-4A3B-9B9F-6AF136562E6F}
HKEY_CLASSES_ROOT\CLSID\{BE1A344F-9FF5-4024-949B-52205E6DB2D0}
HKEY_CLASSES_ROOT\CLSID\{95325092-62FC-473B-B32A-AE613278855B}
HKEY_CLASSES_ROOT\Interface\{F7D09218-46D7-4D3D-9B7F-315204CD0836}

HKEY_CLASSES_ROOT\TypeLib\{E63648F7-3933-440E-B4F6-A8584D
D7B7EB}

HKEY_CLASSES_ROOT\w123.w123mgrHKEY_CLASSES_ROOT\w12
3.w123mgr.1

HKEY_CLASSES_ROOT\CLSID\{9C3A3CA4-172E-4A1F-AA22-94C639
979252}

HKEY_CLASSES_ROOT\dqktgmrey.XunBHoSwfClicker

HKEY_CLASSES_ROOT\TypeLib\{385AB8C5-FB22-4D17-8834-064E2B
A0A6F0}

HKEY_CLASSES_ROOT\CLSID\{037C7B8A-151A-49E6-BAED-CC05F
CB50328}

HKEY_CLASSES_ROOT\CLSID\{6EDB3B51-076A-4828-92D8-911AD7F
3B3A5}

HKEY_CLASSES_ROOT\CLSID\{285AB8C6-FB22-4D17-8834-064E2BA
0A6F0}

HKEY_CLASSES_ROOT\Interface\{385AB8C4-FB22-4D17-8834-064E2B
A0A6F0}

HKEY_CLASSES_ROOT\TypeLib\{385AB8C5-FB22-4D17-8834-064E2B
A0A6F0}

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Windo
ws MMS ServiceImagePath = 윈도우 프로그램 파일 폴더\common
files\microsoft shared\msinfo\servers.exe

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wi
ndows MMS ServiceImagePath = 윈도우 프로그램 파일 폴더
\common files\microsoft shared\msinfo\servers.exe

HKEY_CLASSES_ROOT\CLSID\{C6039E6C-BDE9-4de5-BB40-768CAA
584FDC}

HKEY_CLASSES_ROOT\CLSID\{030A0F33-5B99-482E-83F5-2EEB8457

878B} HKEY_CLASSES_ROOT\Interface\{F7D09218-46D7-4D3D-9B7F-315204 CD0836} HKEY_CLASSES_ROOT\TypeLib\{E63648F7-3933-440E-B4F6-A8584D D7B7EB} HKEY_CLASSES_ROOT\z444.z444mgr HKEY_CLASSES_ROOT\z444.z444mgr.1

마. 인터넷 익스플로러 관련 행위

악성코드는 레지스트리의 특정 위치에 레지스트리 값을 생성 또는 변경함으로써 인터넷 익스플로러 관련 행위를 수행한다. 브라우저 도우미 개체 레지스트리 값을 추가함으로써 인터넷 익스플로러 시작 시 자동으로 실행되도록 한다. 또한 활성화되는 인터넷 익스플로러 프로세스의 수를 고정시킴으로써 새로운 인터넷 익스플로러 탭을 생성하여도 하나의 프로세스만 활성화되도록 설정한다. 인터넷 익스플로러의 시작 페이지를 변경하고 고정시키는 악성 행위 역시 레지스트리 값의 추가를 통해 이루어지며 팝업 설정, 웹페이지에서의 이미지와 비디오의 디스플레이 등에 관한 설정을 변경한다. 인터넷 익스플로러 관련 행위가 이루어지는 레지스트리의 공통적인 위치는 다음과 같다.

- o HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer
- o HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer

다음 표는 악성코드가 인터넷 익스플로러 관련 설정을 변경함으로써 악성 행위를 수행하는 레지스트리의 목록이다.

[표 2-19] 인터넷 익스플로러 관련 설정을 악용하는 레지스트리

```
HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\MainStart Page = http://www.h**ol*nk.com
HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\MainDisplay Inline Videos = no
noHKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\New WindowsPopupMgr = yes
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersi
on\ExplorerWINID = 1c3eaea5e6ea624
HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\MainDisplay Inline Images = no
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersi
on\RunRestore Operation
HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\MainStart Page = "http://www.1*5*u.***/?6953/"
HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet
Explorer\Control PanelHOMEPAGE = 0x1
```

바. 윈도우 설정과 관련된 행위

악성코드는 레지스트리의 특정 위치에 레지스트리 값을 추가함으로써 레지스트리 편집기가 실행되지 않도록 하거나 도구 메뉴의 폴더 옵션 메뉴 항목을 제거하는 등 사용자 정책을 변경하는 악성 행위를 수행한다. 악성코드는 또한 정책 설정을 통해 드라이버의 자동 실행 설정을 변경하여 트로이 목마의 발견을 방해한다. 레지스트리에 값을 추가해 윈도우가 제공하는 커맨드 창, 레지스트리 에디터, 작업관리자를 사용할 수 없도록 하고 웹 콘텐츠 영역이나 제한된 사이트 웹 콘텐츠 영역에서 첨부 파일에 영역 정보를 보존하지 않도록 설정한다. 또 확장자의 위험수준을 낮

추는 행위 역시 수행한다. 사용자 정책 변경을 위해 악용되는 레지스트리의 공통적인 위치는 다음과 같다.

- o HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows
- o HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies

그 밖에 윈도우 시작 시 autoexe.bat를 실행하도록 하거나 레지스트리에 값을 추가하여 사용자 시스템 숨김 파일 보기 속성을 해제하고 윈도우 SFC 복원 기능을 중지시키는 행위를 수행한다. 뿐만 아니라 트로이 목마에 의해 윈도우 시스템 폴더에 생성되는 특정 파일을 은폐시키는 행위를 수행한다. 다음 표는 악성코드가 윈도우 관련 설정을 변경함으로써 악성 행위를 수행하는 레지스트리의 목록이다.

[표 2-20] 윈도우 설정을 악용하는 레지스트리

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\ExplorerNoDriveTypeAutoRun = 0x91
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\ExplorerNoDriveTypeAutoRun = 0x0
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\SystemDisableRegistryTools = 0x1
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\ExplorerNoFolderOptions = 0x1
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\AssociationsLowRiskFileTypes = "zip;.rar;.cab;.txt;.exe;.reg;.msi;.htm;.html;.gif;.bmp;.jpg;.avi;.mov;.mp3;.wav"
HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\SystemDisableCMD = 1DisableRegistryTools = 1DisableTaskMgr = 1

```

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\
Policies\AssociationsSaveZoneInformation = 0x1
HKEY_CURRENT_USER\Software\Microsoft\Windows
NT\CurrentVersion\WinlogonParseAutoexec = "1"
HKEY_USER\.DEFAULT\Software\Microsoft\Windows
NT\CurrentVersion\WinlogonParseAutoexec = "1"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\WinlogonSFCDisable = 9d, ff, ff, ff
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SVKP\I
magePath = \??\C:\WINNT\System32\SVKP.sys
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SV
KP\ImagePath = \??\C:\WINNT\System32\SVKP.sys
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\dx32cxe
l\ImagePath = \??\C:\WINNT\System32\dx32cxel.sys
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dx
32cxel\ImagePath = \??\C:\WINNT\System32\dx32cxel.sys
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SVKP\I
magePath = \??\C:\WINNT\System32\SVKP.sys
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SV
KP\ImagePath = \??\C:\WINNT\System32\SVKP.sys
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\dx32cxe
l\ImagePath = \??\C:\WINNT\System32\dx32cxel.sys
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dx
32cxel\ImagePath = \??\C:\WINNT\System32\dx32cxel.sys

```

사. 네트워크 설정 관련 행위

악성코드는 특정 위치의 레지스트리를 변경하거나 새로운 레지스트리

키 값을 추가함으로써 TCP/IP 설정을 변경하고 사용가능한 포트의 개수를 변경하고 방화벽을 우회하는 등의 네트워크와 관련된 행위를 수행한다. 또한 특정 폴더를 생성하여 P2P 공유프로그램의 공유폴더로 설정한다. 다음 표는 악성코드가 네트워크 관련 설정을 변경함으로써 악성 행위를 수행하는 레지스트리의 목록이다.

[표 2-21] 네트워크 설정을 악용하는 레지스트리

```
HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Main\TabProcGrowth = 0x0
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcp
ip\Parameters\ "TcpNumConnections" = "(설정값 변경)"- 윈도우 XP
SP2tcpip.sys
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\
ParametersMaxUserPort = 0xffff
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Shared
Access\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplic
ations\List[악성원본파일의 FullPath] = "[악성원본파일의
fullpath]:*.enabled;pino"HKEY_LOCAL_MACHINE\SYSTEM\Control
Set001\Services\SharedAccess\EpochEpoch = [주기적으로 기본값+1
씩 증가]
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcp\
Parameters
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcp
ip\Parameters
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpi
p\Parameters\DatabasePath
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersi
on\Internet Settings
```

HKEY_CURRENT_USER\Software\Microsoft\Windows\KAZAA\LocalContent\Dir0 = 012345:C:\윈도우 시스템 폴더\KaZaaBinaries\

아. 레지스트리 존재 여부 확인

악성코드가 특정 레지스트리의 존재를 확인하여 존재하지 않으면 새로이 생성하고 존재하는 경우 종료하는 행위를 한다. 2001년 8월 발견된 VBS/PeachyPDF 웜이 그 예인데, PDF(Adobe Portable Document Format) 파일 속에 스크립트 기능으로 존재하는 PDF 웜으로서 사용자가 Acrobat 프로그램을 실행시킬 때 VB 스크립트가 실행된다. 특정 레지스트리의 존재를 확인하여 존재하지 않으면 만들고 존재하면 종료한다. VMS/PeachyPDF 웜이 존재를 확인하는 레지스트리의 위치는 [표 2-22]와 같다.

[표 2-22] 악성코드가 존재 여부를 확인하는 레지스트리

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\OUTLOOK.PDF

자. 레지스트리 값 참조

악성코드가 레지스트리의 값을 참조하여 사용자들의 메일주소나 계정 주소, 키값 등을 얻어내는 행위를 한다. 국내에서 2001년 발견된 Win32/Aliz.worm 웜은 레지스트리의 특정 주소에 있는 파일에서 사용자들의 메일주소와 SMTP 서버 주소를 가져온 후, 모든 사용자에게 메일을 보내는 행위를 한다. Win32/Aliz.worm 웜이 메일주소와 서버 주소를 가져오는 레지스트리의 위치는 [표 2-23]과 같다.

[표 2-23] 메일주소와 서버 주소를 가져오는 레지스트리

HKEY_CURRENT_USER\SOFTWARE\Microsoft\WAB\WAB4\Wab

File Name
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Account Manager\Accounts\00000001

Win32/Zerbot.worm.157662 웜은 IRCBot의 일종으로 비주얼 C++로 제작되었으며 별도의 툴을 이용하여 암호화된 형태이다. 실행될 때 마다 암호화된 자신을 실행하기 위해 특정 레지스트리 값을 참조하여 키값을 얻어낸다. Win32/Zerbot.worm.157662 웜이 키값을 참조하는 레지스트리의 위치는 [표 2-24]와 같다.

[표 2-24] 키값을 참조하는 레지스트리

HKEY_LOCAL_MACHINE\SOFTWARE\Krypton

제 3 장 연관성 정보 구조

제 1 절 연관성 정보 구조 개요

1. 연관성 정보 구조 정의

연관성 정보 구조는 악성코드에 감염된 샘플을 확보한 시점부터 악성코드 샘플을 이용한 분석과정, 악성코드 확산 현황 분석 과정에서 얻을 수 있는 종합적인 정보들을 기준으로 악성코드를 정의·분석할 수 있는 구조이다. 연관성 정보 구조는 악성코드의 기능에 따라 기준을 마련함으로써 악성코드 위험 수준에 대한 평가와 변종 악성코드에 대한 예측에 적용 가능한 정보 구조이다.

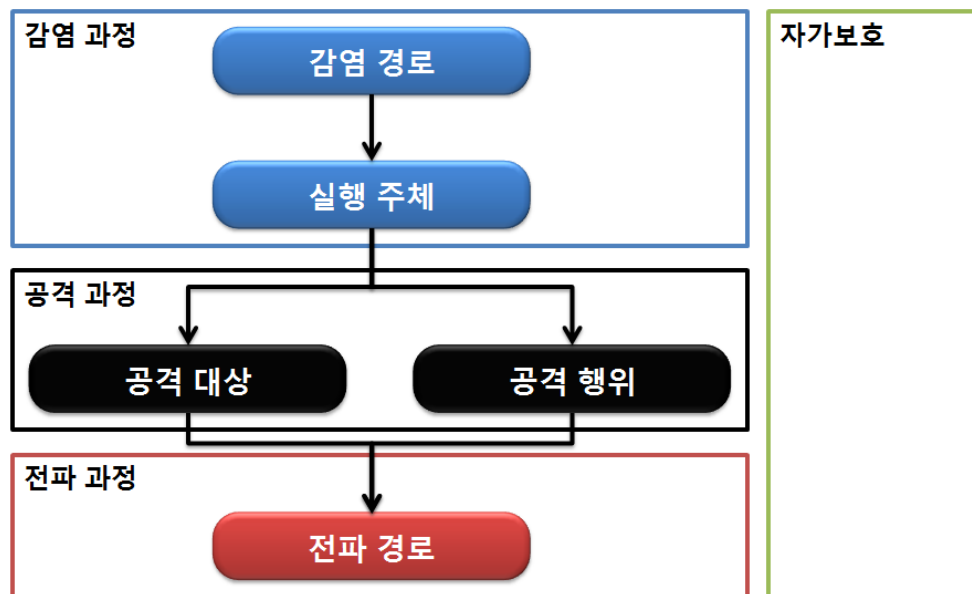
2. 연관성 정보 구조 필요성

각종 산업들과 IT 산업이 융합되면서 사회기반산업들이 전산에 의존적으로 변화하였다. 그에 따라 악성코드의 공격 표적이 사회기반산업들까지 확대되면서 악성코드로 인한 피해규모 또한 크게 증가하였다. 악성코드로 인한 피해규모가 증가하면서 악성코드 분석과 예방 방안의 필요성에 대해 사회가 인식하기 시작하였다. 이에 따라 정부, 기업, 연구기관 등에서 악성코드 분석을 위해 독자적인 활동을 수행 하게 되었다. 하지만 각 기관들은 악성코드 분석 목적에 있어 미묘한 차이를 가진다. 정부는 국가 안보를 지키기 위한 목적으로 악성코드 분석을 수행하는가 하면, 기업은 시그니처 확보를 통한 자사의 안티바이러스 프로그램 판매를 위해 악성코드를 분석하고 있다. 연구기관의 경우 학술적 측면에서 악성코드를 분석한다. 그 결과 악성코드의 분석결과가 기관에 따라 차이가 있으며 특히 이를 표현하는 방법에 있어서 표준화된 기준이 없다보니 기

관에 따라 분석 결과에 대한 표현이 제각각으로 이루어지고 있다. 그러다 보니 동일한 악성코드에 대한 분석결과가 기관에 따라 서로 다르게 나타나고 있어 연구결과물에 대한 공유가 어려운 실정이다. 또한 이러한 이유로 기관들 간에 악성코드 분석을 위한 협력 연구가 어렵게 되었다. 하지만 최근 하루에 제작되는 악성코드가 절대적인 수량이 크게 증가하고 악성코드의 공격 대상이 개인을 넘어서 국가 기간산업 위협하면서 기관별 협력 연구를 통한 악성코드에 대한 빠르고 정확한 분석이 필요하게 되었다. 악성코드에 대한 협동 연구를 위해서는 가장 기본적으로 분석과정 및 분석결과에 대한 통일된 기준이 필수적이다.

본 연구 보고서에서는 악성코드 분석결과에 대한 통일된 기준안 마련을 위한 연관성 정보 구조를 제시한다.

제 2 절 연관성 정보 구조



(그림 3-1) 악성코드 동작 과정

연관성 정보 구조는 악성코드 발생 시에 그 분석 결과를 바탕으로 악성코드를 정의하고 위험 수준을 판별하기 위한 기준 도구이다. 악성코드 정의를 위해 악성코드가 감염되고 공격행위를 수행하는 과정을 구분하고 이를 각각 개별적으로 심화하는 형태로 구성된다. 즉, 연관성 정보 구조는 단계적인 구조를 가지며, 하위 단계는 상위 단계를 세분화하여 악성코드의 행위를 상세히 기술한다. 이를 통해 악성코드들을 행위 및 특징을 바탕으로 악성코드를 분류하고 세부 행위에 따라 위험수준을 측정하여 최종적으로 악성행위의 위험수준을 평가한다.

[표 3-1] 연관성 정보 구조 1 단계 정의

연관성 정보 구조의 1 단계	설명
감염경로	특정 시스템이 악성코드에 감염된 경로에 따라 악성코드를 정의·분류하기 위한 내용이다. 이는 동일한 단계 중 하나인 전파경로와 흡사하나 의미상의 차이가 존재한다.
실행주체	특정 시스템에서 악성코드를 실행시키는 주체에 대해 정의·분류하기 위한 요소이다.
공격대상	악성코드가 공격하는 개념적 대상과 사회적 대상에 대해 정의·분류하기 위한 요소이다.
공격행위	악성코드가 수행하는 사이버 공격 행위들에 대해 정의·분류하기 위한 요소이다.
전파경로	악성코드가 다른 시스템을 감염시키기 이용하는 경로들에 대해 정의·분류하기 위한 요소이다.
자가보호	악성코드가 스스로를 보호하기 위한 각종 기술들에 대해 정의·분류하기 위한 요소이다.

연관성 정보 구조의 1 단계는 악성코드가 감염되어 악성행위를 수행하고 악성코드를 전파하는 전체 악성코드 동작 과정과 연관이 있다. (그림 3-1)은 악성코드의 동작 과정을 나타낸 것이다. 악성코드의 전체 동작 과정은 감염 과정, 공격 과정, 전파 과정으로 단계적으로 이루어져 있으며, 전체 동작 과정과는 별개로 모든 과정에서 그 행위를 수행하는 [자가보호]로 구성된다. 연관성 정보 구조의 1 단계는 악성코드 동작 과정을 구성하는 [감염경로], [실행주체], [공격대상], [공격행위], [전파경로], [자가보호] 등이 존재한다.

1. 감염경로

연관성 정보 구조 1 단계 중 [감염경로]는 특정 시스템이 악성코드에 감염된 경로를 바탕으로 악성코드를 정의·분류하기 위한 부분이다. 이는 연관성 정보 구조 1단계 중 하나인 [전파경로]와 많은 부분 흡사하게 표현되나 의미상의 차이로 구분된다. [감염경로]는 악성코드 샘플을 확보하는 과정에서 이 악성코드 샘플 확보를 위해 해당 악성코드에 시스템이 감염되는 과정과 경로를 나타낸다. 즉, 확보한 악성코드 샘플이 악성코드의 [전파경로]에 따라 자동적인 형태로 감염이 되는 경우와 악성코드 전파를 위해 직접 사람이 개입하여 인위적인 경로를 통해 감염이 이루어지는 모든 경우를 포함한 개념이 감염경로이다.

모든 악성코드들은 최종적으로는 네트워크를 통한 다운로드 명령이나, 저장매체를 통한 복사 명령에 따라 피해자의 시스템에 옮겨진다. 악성코드 정의 및 분류를 위해서는 악성코드의 복사 또는 다운로드가 발생하게 되는 결정적 요인에 대한 정의나 구분이 필요하다. 감염경로는 이를 구체화하기 위해 하위 단계로 세분화된다. 감염경로를 구체화하기 위한 요소로는 [감염 이용 매체], [감염 유형]과 [사용자 의존도]가 있다. 아래 표는 감염경로의 하위 단계를 서술한 표이다.

[표 3-2] 감염경로의 2 단계

감염경로의 하위단계	설명
감염 이용 매체	감염 이용 매체는 특정 시스템에 악성코드가 유입되는데 이용된 매체의 종류를 정의·분류하기 위한 요소이다.
감염 유형	감염 유형은 특정 시스템에 악성코드가 감염되는 요인을 정의·분류하기 위한 요소이다.
사용자 의존도	특정 시스템에 악성코드가 감염되는 과정이 시스템 사용자 행위에 얼마나 의존적인지를 정의·분류하기 위한 요소이다.

가. 감염 이용 매체

[감염 이용 매체]는 [감염경로]의 2 단계 구조 중 하나로 악성코드가 특정 시스템에 감염되는 과정에서 이용된 매체에 대해 정의·분류하기 위한 단계이다. 악성코드의 감염에 이용되는 매체는 새로운 매체가 발생함에 따라 얼마든지 변화할 수 있다. 따라서 이 분류는 충분한 확장성을 요한다. 새로운 매체에 등장에 따라 악성코드 감염에 이용되는 매체가 늘어나는 경우 [감염 이용 매체]의 하위 단계에 새 매체를 등록해야 한다. 연관성 정보 구조의 확장성에 대해서는 제 3 절에서 자세히 기술한다.

[감염 이용 매체]는 [전파경로]의 [전파 이용 매체]와 동일한 세부 구조를 가지고 있으나 의미상에 차이를 갖는다. [감염경로]의 [감염 이용 매체]는 해당 악성코드가 특정 시스템에 감염되는 과정에서 분석된 감염경로이며 이는 악성코드 분석을 통해 얻을 수 있는 [전파경로]의 [전파 이용 매체] 중 하나일 수 있으며 악성코드의 전파경로에 포함되지 않는 매체일 수도 있다.

[표 3-3] [감염경로]의 감염 이용 매체

감염 이용 매체 분류	감염 이용 매체 설명
이동식 저장매체	데이터 저장 및 이동을 위한 장치로 플래시 메모리, 광학디스크, 자기 디스크(외장 하드) 등
모바일 디바이스	데이터 저장 및 이동이 가능하며, 이동성을 제공함과 동시에 컴퓨팅 장치에 연결이 가능한 단말기
하드 디스크	큰 용량 장점으로 데이터 저장/백업이 주목적, 데이터 이동을 위해 사용가능
네트워크	ARP 패킷등을 통해 탐색 및 전송
P2P	서버를 거치지 않고 사용자간 데이터 공유를 위해 사용하는 데이터 공유 기법
웹하드	웹 인터페이스를 통해 접근 가능한 디스크
FTP 클라이언트/서버	FTP를 이용한 데이터 전송
웹서버	HTTP를 이용한 데이터 전송
메신저	네트워크를 통해 대화 및 데이터 공유
게시판	특정 사용자들 간에 정보를 업로드/다운로드 할 수 있는 커뮤니티 공간
블로그	개인의 관심사를 업로드하는 공간
전자우편	전자 메일
ActiveX	웹과 프로그램을 연결하기 위한 기술
BHO	웹 브라우저 이용 편의를 위한 프로그램
클라우드 컴퓨팅	유틸리티 컴퓨팅 개념의 필요한 자원을 서버로부터 대여하여 사용하는 서비스

(1) 이동식 저장매체

플로피 디스크, 광학 디스크 등 데이터를 저장할 수 있는 디스크 또는 비휘발성 메모리를 사용하여 사용자의 데이터를 저장하고 복사, 이동시킬 수 있는 장치들을 뜻한다. 특정 운영체제는 Autorun.inf 등 매체를

시스템에 연결 시 자동 실행하는 기능을 제공하고 있어 시스템 사용자는 매체를 시스템에 연결하는 행위만으로도 악성코드에 감염될 수 있다. 이동식 디스크를 통해 감염되는 악성코드 들은 물리적인 접근을 통해서만 접근이 가능하기 때문에 네트워크를 통해 전파되는 악성코드 보다 낮은 확산력을 갖는다.

(2) 모바일 디바이스

모바일 디바이스는 스마트폰, 태블릿PC와 같이 모빌리티 특성을 가짐과 동시에 컴퓨팅이 가능한 단말기들을 뜻한다. 휴대전화에 인터넷 통신과 정보검색 등 컴퓨터 지원 기능을 추가한 단말기인 모바일 디바이스는 대용량의 저장장치와 웹 접근 가능성, 소액결제, 인터넷 뱅킹 등 다양한 종류의 서비스 이용이 가능하고 물리적으로 다른 시스템과의 연결, 악성코드가 삽입된 웹 사이트 열람, 감염된 이동식 저장매체 연결, 블루투스를 통한 악성코드 전송 등 다양한 [전파경로]를 가지고 있다. 스마트폰이 악성코드에 감염되고 해당 악성코드가 다양한 [전파경로]를 포함하고 있다면 빠른 시간에 악성코드가 확산될 수 있다.

(3) 하드 디스크

하드 디스크는 자기 디스크를 사용하는 대용량 저장매체로 다양한 파일들이 저장된 PC의 하드 디스크는 악성코드가 전파될 수 있는 좋은 매체가 된다. 이동식 디스크와는 다르게 휴대성이 뛰어나지는 않지만 USB 케이블을 통하여 외부 장치로 인식을 하면 손쉽게 데이터를 읽고 쓸 수 있기 때문에 악성코드의 주요 전파 경로가 된다. 특히 하드 디스크는 큰 용량을 이용하여 시스템의 정보들을 백업하는 용도로 사용되어 기존의 시스템이 악성코드에 감염되어 있는 경우 해당 악성코드 또한 함께 백업되기 때문에 이후 시스템을 새로 설치하는 경우에도 백업된 악성코드에 재감염될 수 있는 위험성이 존재한다.

(4) 네트워크

네트워크는 각종 프로토콜에 따라 패킷을 악성코드를 전파에 이용하는 경우를 뜻한다. ARP 스푸핑이 이에 해당하며, 악성코드가 ARP 스푸핑을 이용하면 같은 네트워크 대역에 있는 다른 시스템을 탐지하여 다수의 컴퓨터를 감염시킬 수 있다. 이러한 경우 같은 네트워크의 시스템 중 한 대라도 치료가 되지 않고 감염되어 있으면 자신의 시스템을 치료했더라도 다시 전파 될 위험이 있다.

(5) P2P

P2P는 인터넷을 사용하는 사용자들의 시스템을 직접 연결을 통해 파일을 공유하는 방법이다. 이는 하나의 노드가 일방적인 공급자가 되고 다른 노드가 일방적인 수요자가 되는 기존의 서버-클라이언트의 개념과 차별화된 방법으로 P2P의 경우 공유에 참여하는 모든 사용자들이 공급자인 동시에 수요자가 될 수 있는 파일 공유 구조이다. P2P의 가장 큰 장점은 공유에 참여하는 모든 사용자들이 공급자가 될 수 있기 때문에 빠른 속도로 파일 공유가 가능하다는 점이다. 따라서 P2P를 [전파경로]로 사용할 경우 악성코드의 확산은 삼시간이 이루어지게 된다. 주로 악성코드들은 P2P 프로그램들의 기본 공유폴더에 자기 자신을 복사하거나 공유되는 파일에 악성코드를 삽입하여 전파하고 있다.

(6) 웹하드

웹하드는 대용량 저장 장소를 사용자에게 제공해주는 서비스로 대형 포털사와 웹하드 업체들이 시장에 진출해 있다. 기본적으로 웹하드는 개인에게 저장공간을 제공하지만 특정 서비스업에서는 웹하드를 자료 전송 용으로 사용하고 있다. 많은 자료들이 저장되어 있는 웹하드가 악성코드

에 감염될 경우 전파되는 확산력은 매우 높다.

(7) FTP 클라이언트/서버

FTP는 인터넷을 통해 서버와 클라이언트 간에 파일을 전송하기 위한 프로토콜을 말한다. 특정 FTP서버를 해킹하여 악성코드를 삽입하고 해당 파일을 다운로드 받도록 유도할 수 있으며, FTP에 접속하는 URL을 바탕으로 메시지를 작성하여 악성코드에 감염된 파일을 다운로드 받도록 할 수 있다.

(8) 웹서버

악성코드는 웹서버를 통하여 전파될 수 있다. 웹서버를 이용한 악성코드 전파는 웹 페이지에 스크립트를 삽입하여 웹페이지를 열람하는 것만으로도 악성코드를 자동적으로 다운로드할 수 방법이나, 사용자가 특정 웹페이지를 열람할 경우 공격자가 미리 악성코드를 배포할 수 있도록 설정해둔 다른 웹페이지로 연결시켜 사용자로 하여금 악성코드를 다운로드 하도록 만드는 방법이 있다. 웹서버를 이용한 악성코드 감염은 사용자가 인지하지 못하거나 무의식적으로 동의를 수행하는 과정을 통해 주로 이루어진다.

(9) 메신저

메신저는 단문메시지(IM message) 및 채팅 기능을 통해 악성코드를 감염에 이용 될 수 있다. 메신저를 통해 악성코드가 삽입된 URL을 배포하거나 시장에 서비스되고 있는 메신저들은 대부분 파일 전송기능을 포함하고 있어, 악성코드가 감염된 파일을 직접 전송할 수 있다. 또한 메신저는 모르는 사람들 간에 메시지를 주고받는 것이 아닌 대게 친구, 친인척들을 대상으로 하기 때문에 사회공학적 기법을 병행하여 악성코드를

유포할 경우 많은 시스템을 감염시킬 수 있다.

(10) 게시판

많은 웹서버에서 제공하고 있는 게시판 기능은 파일 업로드, 사회이슈 메시지, URL 등을 게시할 수 있다. 악성코드는 자동적으로 게시글을 남길 수 있는 게시판을 검색하여 글을 작성하는 방법으로 악성코드를 유포하거나 게시글을 직접 작성할 수 없는 게시판인 경우 댓글 기능을 통해 악성코드가 감염된 웹 사이트의 URL을 게시하는 방법으로 악성코드 유포에 이용된다.

(11) 블로그

개인의 관심사에 대한 게시글을 누구나 작성할 수 있는 블로그는 다른 사람의 게시글을 옮겨올 수 있으며 수입의 목적으로 배너형 광고들이 사용되고 있다. 악성코드는 배너광고에 악성스크립트를 삽입할 수 있으며 게시판과 같이 댓글을 통하여 악성코드가 포함된 URL을 유포하는대에 이용될 수 있다. 최근에는 방문자가 많은 유명 블로거들이 생겨나고 있으며 유명 블로거의 블로그가 악성코드에 감염될 시 악성코드는 빠르게 전파될 위험이 존재한다.

(12) 전자우편

메신저의 보급화로 전자우편의 사용률은 많이 떨어졌으나, 동시에 접속해야 메시지를 주고받을 수 있는 메신저와는 달리 전자우편은 주기적인 접속 없이도 메시지를 전송할 수 있는 수단이 된다. 또한 전자우편을 서비스하는 업체들은 대용량의 전송과 장기간의 저장기간을 제공하고 있어 파일전송의 목적으로도 사용되고 있다. 많은 인터넷 사용자들은 복수개의 전자우편 주소를 갖고 있고, 공격자는 스팸메일 리스트를 구비하고

있어 전자우편을 매개체로 한 악성코드 전파방법은 여전히 높은 확산력을 갖고 있다.

(13) 액티브X(ActiveX)

액티브X는 마이크로소프트사에서 개발한 것으로 기존의 응용프로그램으로 작성된 문서 등을 웹과 연동시켜주는 서비스 이다. 하지만 악성코드가 액티브X를 위장하여 설치될 수 있으며 기존의 액티브X의 취약점을 통하여 악성코드를 삽입할 수 있다.

(14) 브라우저 도우미 개체(Browser Helper Object, BHO)

기존의 브라우저 도우미 개체는 브라우저에서 처리할 수 없는 PDF와 같은 문서를 읽을 수 있도록 지원하기 위해 라이브러리 파일을 제공하거나 브라우저 이용 편의를 위해 제공되는 번역, 북마크 기록, 브라우저 화면 캡처, 웹 문서 스크랩, 팝업 페이지 차단 등 다양한 도움 기능을 위해 툴바(toolbar) 형태로 설치되는 보조 프로그램을 의미한다. 브라우저 도우미 개체는 동적 라이브러리를 가지고 있거나 브라우저를 통해 실행 가능한 파일 형태로 존재하기 때문에 악성코드의 감염 대상이 된다. 또한 브라우저 동작을 제어할 수 있어 악성코드가 감염된 웹 페이지로 연결될 수 있으며 유명 브라우저 도우미 개체를 가장하고 악성코드가 포함된 브라우저 도우미 개체가 설치될 수 있다.

(15) 클라우드 컴퓨팅

클라우드 컴퓨팅은 인터넷에 접속가능한 공간이면 언제 어디서나 휴대폰, PDA 등 어떠한 단말기 보다 저렴하고 향상된 IT 서비스와 컴퓨팅 자원을 이용할 수 있게 해주는 기술을 의미한다. 최근 클라우드 컴퓨팅 서비스에 대한 관심이 증가하면서 다양한 유형의 서비스가 이루어지고

있다. 특히 IaaS 서비스의 경우 대용량 데이터의 저장 및 처리를 위한 서비스로 정보 저장 및 접속이 가능하다는 점에서 악성코드 [전파경로]로 사용될 가능성이 있다.

나. 감염 유형

[감염 유형]은 [감염경로]의 2 단계 구조 중 하나로 악성코드가 특정 시스템에 감염되는 과정에서 이용된 결정적 감염 요인을 정의·분류하기 위한 요소이다. 악성코드의 [감염 유형]은 새로운 매체가 발생하더라도 변화의 폭이 좁아 확장성이 크게 요구되지 않는다. 아래 표는 [감염경로]의 [감염 유형]들을 정리한 것이다.

[표 3-4] 감염경로의 감염 유형

감염 유형 분류	감염 유형 설명
악성파일 직접 수신	파일 공유가 가능한 서비스의 경우 악성코드를 직접 타인에게 전달
URL 접근	메시지 전달이 가능한 서비스의 경우 악성코드를 내려 받을 수 있는 URL이나 악성 스크립트가 존재하는 웹 페이지 URL을 타인에게 전달
원격 코드 실행	보안 취약점을 이용하여 원격 코드 실행 URL, 직접전송을 통해 악성코드를 내려 받음
권한 상승(취약점)	보안 취약점을 이용하여 권한 상승 URL, 직접전송을 통해 악성코드를 내려 받음
권한 상승(암호)	암호 분석을 이용하여 권한 상승 URL, 직접전송을 통해 악성코드를 내려 받음
권한 상승(동의)	무분별한 동의에 따른 권한 상승 URL, 직접전송을 통해 악성코드를 내려 받음

[악성코드 직접 수신]은 HTTP, FTP(FTP를 포함한 모든 파일 전송이 가능한 프로토콜), MAIL(SMTP을 포함한 모든 메일 관련 프로토콜) 등 파일 전송이 가능성 서비스를 통해 직접 내려 받는 경우로 가장 기본적인 방식이다. [URL 접근]은 악성코드가 포함되어 있는 웹페이지나 악성코드가 포함된 파일의 URL을 접근하여 악성코드를 내려 받는 경우이다. [원격 코드 실행]은 운영체제나 어플리케이션에 존재하는 보안 취약점으로 인해 악성코드를 다운받도록 설계된 원격 코드를 실행하게 되는 경우이다. [권한 상승]은 보안 취약점이나 높은 권한을 가진 계정의 암호를 분석하여 알아내거나 사용자의 무분별한 동의에 따라 디렉터리의 쓰기 권한을 얻게 되는 경우로 각종 파일전송 프로토콜을 통해 악성코드 감염이 가능하다.

다. 사용자 의존도

[사용자 의존도]는 [감염경로]의 2 단계 구조 중 하나로 악성코드가 특정 시스템에 감염되는 과정이 사용자의 행위에 어느 정도 의존적인지를 정의·분류하기 위한 요소이다. 악성코드의 [사용자 의존도]는 새로운 매체가 발생하더라도 변화의 폭이 좁아 확장성이 크게 요구되지 않는다. 아래 표는 [감염경로]의 [사용자 의존도]들을 정리한 것이다.

[표 3-5] 감염경로의 사용자 의존도

사용자 의존도 분류	사용자 의존도 설명
사용자 직접 명령	파일의 다운로드 또는 복사를 목적으로 사용자가 직접 시스템에 요청하는 경우
사용자 간접 명령	특정 서비스 이용의 위해 파일의 다운로드 또는 복사가 간접적으로 시스템에 요청되는 경우
비(非) 의존적	사용자의 의지와는 무관하게 파일의 다운로드 또는 복사가 이루어지는 경우

[사용자 직접 명령]은 사용자가 직접 다운로드나 복사의 명령을 내렸거나 그에 준하는 행위를 시스템에게 요청한 유형이다. 이는 감염 매체를 통해 직접적으로 사용자가 파일을 시스템으로 이동시키는 유형으로 사용자가 파일 복사, 파일 실행, 파일 다운로드, 다운로드 파일 실행, URL 접근 등을 요청하는 경우가 이에 속한다. [사용자 간접 명령]은 사용자가 서비스를 이용하는 과정에서 간접적으로 파일의 다운로드나 복사를 명령하는 유형이다. 이 유형으로는 무분별한 동의, 자동실행, 자동동의, 자동 업데이트, 자동 다운로드 등 사용자 편의를 위해 사용자의 동의 과정 없이 이루어지는 파일 다운로드나 복사를 의미한다. 웹서비스 이용을 위한 액티브X 설치 동의, 추가적인 응용프로그램 설치 동의, 브라우저 도우미 개체(BHO) 설치 동의, 사이트 인증서 동의 등은 무분별한 동의의 대표적인 예이다. 사용자에게 의존하지 않고 실행되는 경우인 [비의존적]은 보안 취약점에 따른 원격 코드 실행, 보안 취약점에 따른 권한 상승, DNS 서버를 이용한 파밍, ARP 스푸핑과 같이 사이버 공격을 통해 사용자의 의사와는 무관하게 시스템에 접근하여 파일을 다운로드하는 유형이다.

2. 실행주체

연관성 정보 구조 1 단계 중 [실행주체]는 특정 시스템에 다운로드된 악성코드가 실행되는 경로에 따라 악성코드를 정의·분류하기 위한 내용이다.

악성코드가 [감염경로]를 통해 특정 시스템에 복사·다운로드 되더라도 해당 악성코드가 실행되기 이전에 이를 발견하고 조치를 취할 수 있다면, 악성코드 감염에 따른 피해를 입기 전에 이를 치료·격리 할 수 있다. 하지만 악성코드의 복사·다운로드 과정과 복사 또는 다운로드된 악성코드의 실행이 연속적으로 이루어지는 경우가 존재하며, 이러한 경우 그렇지 않은 경우에 비해 높은 위험 수준을 갖는다. 그러므로 악성코드의 [실행주체]에 대한 명확한 정의나 구분이 필요하다. [실행주체]는 [운영체

제], [웹브라우저], [사용자], [다른 응용프로그램]과 같은 하위 단계로 세분화 된다. 아래의 표는 [실행주체]의 하위 단계를 서술한 표이다.

[표 3-6] 실행주체의 2 단계

실행주체 하위단계	설명
사용자	특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 사용자가 직접 관여하는 유형을 정의·분류하기 위한 요소이다.
부트영역	운영체제 이미지가 로드되기 전 악성코드가 먼저 실행되도록 부트영역에 코드를 삽입하는 악성코드 유형을 정의·분류하기 위한 요소이다.
운영체제	특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 운영체제가 관여하는 유형을 정의·분류하기 위한 요소이다.
웹브라우저	특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 [사용자]가 관여하는 유형을 정의·분류하기 위한 요소이다.
다른 응용프로그램	특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 다른 응용프로그램이 관여하는 유형을 정의·분류하기 위한 요소이다.

가. 사용자

[사용자]는 [실행주체]의 2 단계 구조 중 하나로 특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 시스템 [사용자]가 직접 관여하는 유형을 정의·분류하기 위한 요소이다.

[표 3-7] 사용자 실행의 의미

분류	설명
비(非)연속성	특정 시스템에 복사·다운로드 된 악성코드가 실행되는 주체로 사용자의 입력이 필요한 경우는 복사·다운로드와 악성코드 실행 과정이 비연속적으로 연결된 경우이다.

악성코드가 사용자의 행위에 의존적으로 동작하는 경우 [비 연속성]의 의미를 갖는다. [비 연속성]은 기본적인 의미로 악성코드의 감염과정 및 사이버 공격 행위가 충분히 자동화되어있지 않다는 것을 의미한다. [사용자]에 의해 최초 실행되는 악성코드의 경우 악성코드가 복사·다운로드된 이후 실행되기까지 논리적 연속성이 존재하지 않는다. 이는 감염이 악성코드의 자동화된 [전파경로]를 통해 이루어진 것이 아님을 의미한다. 즉, 해당 악성코드가 자동화된 [전파경로]가 존재하지 않거나, 자동화된 [전파경로]를 갖고 있더라도 이를 이용하여 감염이 이루어진 것이 아님을 나타낸다. 이 경우 악성코드가 무사히 시스템에 복사·다운로드 되었다 할지라도 악성코드의 감염이 완전히 이루어진 것이 아니며 악성코드의 공격 행위에 대한 피해가 발생하지 않는다. 따라서 악성코드 실행이 [사용자]에 의존적인 경우 악성코드의 실행 과정에 있어 [비(非)연속성]을 띠고 있으며, 이 경우 악성코드 행위가 발생하기 이전에 탐지 및 치료가 가능하여 악성코드의 감염이 논리적으로 연속성을 지닌 다른 경우에 비해 위험 정도가 낮다. 악성코드의 [실행주체]가 [사용자]일 경우 예상할 수 있는 다른 의미는 악성코드의 행위 목적이 사용자의 동의가 필요한 경우 악성코드의 [실행주체]로 [사용자]가 될 수 있다. 대표적인 예로 [사용자]의 결제 유도가 목적이거나 광고를 목적으로 하는 각종 스파이웨어, 애드웨어의 경우 [사용자]의 입력을 유도하여 악성 행위를 수행한다. 이 경우 최종적으로 [사용자]의 결제를 완료하기 위해서는 [사용자]의 금융정보 입력 및 동의가 필요하기 때문에 악성행위의 [실행주체]가 [사용

자]가 된다.

나. 부트영역

[부트영역]은 [실행주체]의 2 단계 구조 중 하나로 시스템이 부트되는 것을 총괄하는 바이오스(BIOS)가 직접 관여하는 유형을 정의·분류하기 위한 요소이다. [부트영역]은 악성코드가 운영체제 이미지가 로드되기 전 악성코드가 실행되는 형태의 악성코드를 말한다. 다른 악성코드와는 달리 운영체제가 로드되기 전에 악성코드가 먼저 실행되기 때문에 운영체제가 로드된 후 악성코드를 삭제하더라도 시스템이 재부팅 되면 다시 악성코드가 실행된다. 따라서 악성코드의 탐지가 어려우며 완전한 치료가 어려워 악성코드의 위험 수준이 높은 분류이다.

바이오스는 롬(Read Only Memory, ROM)에 기록되어 있어 롬-바이오스(ROM-BIOS)라고도 하며 시스템이 부트 될 때 POST(Power On Self Test)를 수행하고 부팅매체 검색 후 부트로더를 실행하는 역할을 한다.

악성코드는 롬에 작성된 바이오스를 직접 수정할 수는 없지만, 악성코드 자체를 MBR에 삽입한 이후 부트로더가 해당 부팅매체를 사용하여 부트할 때 운영체제보다 먼저 실행되게 만들 수 있다. 이 경우 [운영체제]나 [다른 응용프로그램]보다 하위 레벨에서 악성코드가 실행되기 때문에 안티바이러스 프로그램을 종료 시키거나 디버거 프로그램을 중단시키는 등 다양한 악성행위를 수행할 수 있다. 특히 악성코드가 스스로를 보호하기 위한 보호 수단으로 이용될 가능성이 크다.

다. 운영체제

운영체제는 [실행주체]의 2 단계 구조 중 하나로 특정 시스템에 복사·다운로드된 악성코드를 실행하는 주체로 시스템의 운영체제가 직접 관여하는 유형을 정의·분류하기 위한 요소이다.

[운영체제]가 악성코드 실행의 주체가 되는 경우에는 아래 표와 같은

세 가지 의미로 해석된다.

[표 3-8] 운영체제 실행의 의미

분류	설명
운영체제 취약점에 의한 악성코드 실행	특정 시스템에 복사·다운로드 된 악성코드가 사용자의 동의 없이 운영체제에 의해 실행되는 경우는 복사·다운로드와 악성코드 실행 과정이 운영체제에 의해 연속적으로 연결된 경우이다. 이 경우는 주로 악성코드 감염이 운영체제의 보안 취약점에 의해 이루어지는 형태이다.
운영체제 수준에서 악성코드 자동실행	이미 시스템에 설치된 악성코드가 악성 행위 수행을 위해 부팅 단계에서 운영체제에 의해 실행되는 유형으로 항상 시스템에 상주하며 악성행위를 실행하는 형태이다.
운영체제 파일 실행에 따라 악성코드 실행	이미 시스템에 설치된 악성코드가 운영체제의 실행파일(예, explorer.exe)을 변조하였거나 관련 DLL파일을 변조하여 실행되는 형태이다.

첫째 악성코드의 감염과정 및 사이버 공격 행위가 [운영체제]에 존재하는 보안 취약점을 이용하여 이루어지는 형태이다. 이 경우 악성코드의 감염과정 및 사이버 공격 행위가 충분히 자동화되어 있음을 의미한다. [운영체제]에 의해 최초 실행되는 악성코드의 경우 악성코드의 실행을 전제로 악성코드 복사·다운로드가 이루어지기 때문에 이후 실행과정이 논리적 연속성을 갖는다. 이는 감염이 악성코드의 자동화된 [전파경로]를 통해 이루어진 것을 의미한다. 즉, 해당 악성코드가 자동화된 [전파경로]가 존재하며 감염과정이 이를 통해 이루어진 것을 나타낸다. 이 경우 악성코드가 무사히 시스템에 복사·다운로드 됨과 동시에 악성코드의 감염이 완전히 이루어지기 때문에 즉각 악성코드의 [공격 행위]에 대한 피해가 발생한다. 따라서 악성코드 실행이 [운영체제]에 의존적인 경우 악성

코드의 실행 과정에 있어 연속성을 띠고 있으며, 이 경우 악성코드 행위가 발생하기 이전에 탐지 및 치료가 어려워 악성코드의 감염이 논리적으로 연속적이지 않은 다른 경우에 비해 위험 정도가 높다.

둘째 부팅 후 자동적으로 실행되어 지속적으로 행위 수행을 하는 악성코드의 경우 주로 운영체제를 [실행주체]로 이용한다. 특히 가장 많은 컴퓨터 사용자들이 사용하는 운영체제인 마이크로소프트의 윈도우즈는 부팅 시 자동적으로 실행되며 사용자 인터페이스에 들어나지 않는 프로그램 유형인 윈도우즈 서비스 응용 프로그램(Windows Service Application)이 있다. 이 프로그램은 개인 사용자가 특정한 설정환경에서 각 서비스별로 시작, 정지 등의 명령을 내릴 수 있으며, 시작 명령을 받은 서비스는 시스템이 부팅될 때 운영체제에 의해 자동적으로 실행된다. 만약 악성코드가 윈도우즈 서비스 응용 프로그램 형태로 제작되어 서비스로 등록되거나 서비스로 등록되어 있는 다른 윈도우즈 서비스 응용 프로그램을 감염시킬 경우 항상 자동적으로 실행될 수 있다. 이러한 악성코드들은 시스템 감시를 목적으로 하거나 커맨더(commander)의 명령을 수신하여 명령에 따라 악성 행위를 수행하는 형태를 많이 갖는다. 운영체제를 이용해 자동 실행되는 악성코드의 경우 목표 시스템을 완전히 감염시킨 이후에는 매번 실행되기 때문에 공격 성공률이 다른 [실행주체]를 이용하는 악성코드들에 비해 높은 편이며, 감염되더라도 사용자의 인터페이스에 별다른 징후가 나타나지 않기 때문에 전문적인 툴을 사용하지 않으면 사용자가 감염 사실을 알아차리기가 어렵다. 다음 표는 [실행주체]가 운영체제로 인한 자동실행인 경우에 대한 세부 단계에 대해 기술한 표이다. [실행주체]가 운영체제인 경우 자동실행을 위해 이용하는 도구를 기준으로 세부 단계가 나뉘지며, 그 도구로 [윈도우즈 서비스 등록/변경], [레지스트리 등록/변경], [시작프로그램 등록/변경], [Autorun.inf] 감염 등이 있다.

[표 3-9] 운영체제 자동 실행의 하위 단계

분류	설명
서비스 등록/변경	특정 시스템에 복사·다운로드 된 악성코드가 부팅 시 자동적으로 실행되기 위해 윈도우 서비스에 등록/변경하는 유형이다.
레지스트리 등록/변경	특정 시스템에 복사·다운로드 된 악성코드가 자동적으로 실행되기 위해 관련 레지스트리를 등록/변경하는 유형이다.
시작프로그램 등록/변경	특정 시스템에 복사·다운로드 된 악성코드가 부팅 시 자동적으로 실행되기 위해 윈도우 시작프로그램에 등록/변경하는 유형이다.
autorun.inf 감염	사용자 편의를 위해 장치 연결 시 자동적 실행을 위한 autorun.inf에 악성코드가 삽입된 유형이다.

셋째 운영체제의 실행파일인 익스플로러, 윈도우 탐색기와 같은 사용빈도가 높은 실행파일을 변조시키거나 해당 실행파일이 사용하는 DLL을 변조시킴으로써 사용자 인터페이스에서 실행여부를 파악하기 힘들도록 실행되는 악성코드가 존재한다. 이러한 형태의 악성코드는 치료 시 통상적인 모드에서는 어려우며, 안전모드를 이용해야 하는 경우가 많다. 아래 표는 운영체제 실행파일 감염의 하위 단계로 운영체제 실행파일 자체가 변조되는 유형과 관련 라이브러리 파일이 변조되는 유형이 존재한다.

[표 3-10] 운영체제 실행파일 감염의 하위 단계

분류	설명
운영체제 실행파일 자체의 변조	특정 시스템 운영체제의 주요 실행파일에 악성코드가 감염되면서 해당 실행파일에 의해 실행되는 악성코드 유형이다.
관련 라이브러리 변조	특정 시스템 운영체제의 주요 실행파일이 참조하는 라이브러리의 변조에 따라 해당 실행파일 실행 시 구동되는 악성코드 유형이다.

라. 웹브라우저

[표 3-11] 웹브라우저 실행의 의미

분류	설명
웹브라우저 보안 취약점 이용 실행	웹브라우저의 보안 취약점으로 인해 악성코드의 복사·다운로드와 악성코드의 실행이 연속적으로 이루어지는 형태이다.
악성코드가 포함된 액티브X 설치/실행	사용자가 설치 및 실행에 동의한 액티브X가 악성코드를 포함하고 있거나, 이 후 포함된 경우 액티브X의 로드와 함께 악성코드가 실행되는 형태이다.
브라우저 도우미 개체(BHO)를 이용한 악성코드 실행	브라우저의 추가적인 기능을 위해 플러그인 형태로 설치되는 dll 모듈 중 악성행위를 위한 라이브러리가 포함되어 있는 경우 브라우저 이용에 의해 악성코드가 실행되는 형태이다.

[웹브라우저]는 [실행주체]의 2 단계 구조 중 하나로 특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 [웹브라우저]가 직접 관여하는 유형을 정의·분류하기 위한 요소이다.

[웹브라우저]가 악성코드 실행의 주체가 되는 경우에는 [표 3-11]과 같

은 세 가지 의미로 해석된다.

첫째 악성코드의 감염과정 및 사이버 공격 행위가 [웹브라우저에 존재하는 보안 취약점]을 이용하여 이루어지는 형태이다. 이 경우 악성코드의 감염과정 및 사이버 공격 행위가 충분히 자동화되어 있음을 의미한다. 웹브라우저에 의해 최초 실행되는 악성코드의 경우 악성코드의 실행을 전제로 악성코드 복사·다운로드가 이루어지기 때문에 이후 실행과정이 논리적 연속성을 갖는다. 이는 감염이 악성코드의 자동화된 [전파경로]를 통해 이루어진 것을 의미한다. 즉, 해당 악성코드가 자동화된 [전파경로]가 존재하며 감염과정이 이를 통해 이루어진 것을 나타낸다. 이 경우 악성코드가 무사히 시스템에 복사·다운로드 됨과 동시에 악성코드의 감염이 완전히 이루어지기 때문에 즉각 악성코드의 공격 행위에 대한 피해가 발생한다. 따라서 악성코드 실행이 웹브라우저에 의존적인 경우 악성코드의 실행 과정에 있어 연속성을 띠고 있으며, 이 경우 악성코드 행위가 발생하기 이전에 탐지 및 치료가 어려워 악성코드의 감염이 논리적으로 연속적이지 않은 다른 경우에 비해 위험 정도가 높다.

둘째 [악성코드가 포함된 액티브X 설치 및 구동]에 따른 악성코드 실행 형태가 있다. 이 경우 두 가지 시나리오로 구분되는데 첫째 처음부터 악성코드가 포함된 액티브X를 사용자가 설치·구동에 대해 동의하는 경우와 둘째 초기에는 악성코드가 포함되지 않은 정상적인 액티브X를 설치·구동하였으나 액티브X가 수정되어 악성코드가 포함한 경우에도 보안 설정에 따라 사용자 동의 없이 자동 설치 및 실행되는 경우이다.

셋째 모듈형의 DLL파일 설치를 통해 웹브라우저 기능을 확장하는 [BHO를 이용한 실행] 형태가 있다. 기존의 BHO는 웹브라우저에서 처리할 수 없는 문서(예, PDF)를 인식하기 위해 라이브러리 파일을 설치하거나, 웹브라우저의 사용 편의를 위해 툴바(ToolBar) 형태의 확장 기능을 사용 가능하도록 지원하는 서비스를 의미한다. 이는 추가적인 라이브러리 설치가 가능하기 때문에 그 과정에서 악성행위를 수행할 수 있는 악성코드를 복사·다운로드하며, 최종적으로는 이를 실행하는데 이용될 수 있다. BHO에 설치된 악성코드는 웹브라우저를 통해 실행되기 때문에 사

용자 인터페이스에 악성코드의 실행여부가 들어나지 않으며, 많은 경우 치료를 위해서는 안전모드로 부팅이 요구된다. 따라서 악성코드 탐지가 어려우며 탐지된 이후 치료가 어렵다는 점에서 다른 악성코드들에 비해 위험 수준이 높다.

마. 다른 응용프로그램

[다른 응용프로그램]은 [실행주체]의 2 단계 구조 중 하나로 특정 시스템에 복사·다운로드 된 악성코드를 실행하는 주체로 [다른 응용프로그램]이 직접 관여하는 유형을 정의·분류하기 위한 요소이다.

[다른 응용프로그램]이 악성코드 실행의 주체가 되는 경우에는 아래 [표 3-12]와 같은 다섯 가지 의미로 해석된다.

첫째 악성코드의 감염과정 및 사이버 공격 행위가 통신 포트를 이용하는 응용프로그램에 존재하는 보안 취약점을 이용하여 이루어지는 형태이다. 이 경우 악성코드의 감염과정 및 사이버 공격 행위가 충분히 자동화되어 있음을 의미한다. 다른 응용프로그램의 취약점에 의해 최초 실행되는 악성코드의 경우 악성코드의 실행을 전제로 악성코드 복사·다운로드가 이루어지기 때문에 이후 실행과정이 논리적 연속성을 갖는다. 이는 감염이 악성코드의 자동화된 [전과경로]를 통해 이루어진 것을 의미한다. 즉, 해당 악성코드가 자동화된 [전과경로]가 존재하며 [감염과정]이 이를 통해 이루어진 것을 나타낸다. 이 경우 악성코드가 무사히 시스템에 복사·다운로드 됨과 동시에 악성코드의 감염이 완전히 이루어지기 때문에 즉각 악성코드의 공격 행위에 대한 피해가 발생한다. 따라서 악성코드 실행이 [다른 응용프로그램의 취약점]에 의존적인 경우 악성코드의 실행 과정에 있어 연속성을 띠고 있으며, 이 경우 악성코드 행위가 발생하기 이전에 탐지 및 치료가 어려워 악성코드의 감염이 논리적으로 연속적이지 않은 다른 경우에 비해 위험 정도가 높다.

[표 3-12] 다른 응용프로그램 실행의 의미

분류	설명
다른 응용프로그램 보안 취약점 이용 실행	통신 포트를 이용하는 응용 프로그램의 보안 취약점으로 인해 악성코드의 복사·다운로드와 악성코드의 실행이 연속적으로 이루어지는 형태이다.
스크립트 포함 파일 열람	특정 응용프로그램은 실행 가능한 스크립트 파일을 제공한다. 이 경우 악성코드의 복사·다운로드 및 실행을 위한 스크립트가 해당 응용 프로그램의 파일에 포함되어 있는 경우 해당 파일을 열람함으로써 스크립트가 실행되는 형태이다. (예, 매크로 파일)
작업 스케줄러	주기적으로 특정 응용프로그램을 실행시켜 주기 위한 작업 스케줄러를 이용하여 악성코드를 실행하는 형태이다.
자동 업데이트 프로그램	보편적으로 자동적인 수행이 허가된 업데이트 프로그램의 경우 업데이트 서버의 감염 등으로 인해 자동적으로 악성코드가 감염된 악성코드가 복사·다운로드 및 실행되는 형태이다.
다른 악성코드	악성코드 복사·다운로드를 목적으로 제작된 [악성코드 1]에 의해 다른 악성행위 수행을 위한 [악성코드 2]가 복사·다운로드 및 실행되는 형태이다.

둘째 응용프로그램에 의한 악성코드가 포함된 [스크립트 실행에 따른 악성코드 실행] 형태가 있다. 이 형태의 대표적인 예로는 매크로 바이러스 종류나 과거 마이크로소프트사의 아웃룩에서 메일에 포함된 비주얼 베이직(visual basic) 구문을 자동으로 실행시키는 기능을 이용한 웜이 있다. 이처럼 스크립트를 실행 가능한 응용프로그램이나 운영체제에서

실행 가능한 스크립트를 자동적으로 실행할 수 있는 응용프로그램을 통해 스크립트가 실행되면서 악성코드의 복사·다운로드 및 실행이 이루어지는 경우가 이 형태에 속한다.

셋째 [작업 스케줄러]에 등록함으로써 악성코드를 자동적으로 실행하는 형태가 있다. 응용프로그램의 실행여부를 모니터하고 주기적으로 실행시켜주는 응용프로그램 중 하나인 스케줄러는 윈도우즈의 관리도구에 포함된 작업 스케줄러와 리눅스환경에서 많이 사용되는 크론탭(crontab)이 있다. 스케줄러는 서버 시스템에서 특정 데몬(daemon)을 지속적으로 구동하기 위해 사용하는 응용프로그램이다. 특정 프로그램의 실행시키기 위한 일정을 지정하면 스케줄러는 주어진 일정에 맞춰서 해당 프로그램을 실행시킨다. 악성코드의 실행 일정이 스케줄러에 등록되면 해당 스케줄러를 통해서 악성코드가 주기적으로 구동된다. 윈도우즈 운영체제의 서비스 응용프로그램도 스케줄러와 유사한 행위를 한다. 스케줄러를 통해 이용되는 악성코드는 사실 드물다. 그 이유는 자주 이용되는 응용프로그램이 아닌 관계로 사용자가 이를 이용한 악성코드의 동작을 탐지하기 용이하며 운영체제의 동작과 직접적인 관계가 없어 치료가 쉬운 편이다.

넷째 자동적인 수행이 허가된 [업데이트 프로그램]의 경우 업데이트 서버의 감염 등으로 인해 악성코드 전파 및 악성코드 자동 [실행주체]로 이용된다. 대표적인 사건으로 2011년 7월 26일에 발생한 SK커뮤니케이션즈의 네이트 개인정보 유출 사건이 있다. 알툴즈의 업데이트 서버의 제어권이 악의적인 사용자에게 의해 침해되면서 알툴즈의 업데이트 서버에 악성코드가 업로드 되었으며 이를 통해 SK커뮤니케이션즈 내부 네트워크 PC에 악성코드가 전파가 성공하면서 대규모 개인정보 유출이 발생한 사건이다. 이처럼 자동적으로 파일을 다운로드 및 설치하는 권한을 가진 [업데이트 프로그램] 및 그에 준하는 프로그램이 악성코드의 복사·다운로드 및 [실행주체]로 작용하는 형태이다.

다섯째 악성코드 복사·다운로드 및 실행을 목적으로 제작된 악성코드에 의해 새로운 악성코드의 복사·다운로드 및 실행이 이루어지는 형태이

다. 흔히 이러한 역할을 가진 악성코드를 드롭퍼(drooper)라 부르며, 제어권을 탈취할 목적으로 제작된 악성코드나 백도어, 트로이목마 등 많은 종류의 악성코드들이 악성코드 복사·다운로드 및 [실행주체]로 동작한다. 이 경우 또한 악성코드의 복사·다운로드 및 실행 과정이 연속적으로 이루어지기 때문에 대응이 어렵다.

3. 공격대상

연관성 정보 구조 1 단계 중 [공격대상]은 특정 시스템에서 동작하는 악성코드의 공격 대상을 정의·분류하기 위한 내용이다.

모든 악성코드들은 하나 이상의 악성행위를 수행하게 되는데, 이때 발생하는 악성행위에 따라 피해를 입는 대상이 달라진다. 악성코드가 수행하는 [공격행위]와 악성코드의 목표가 되는 [공격대상]은 서로를 한정할 수 있다. [공격행위]에 따라 [공격대상]이 한정되기도 하고, [공격대상]에 따라 [공격행위]가 한정되기도 한다. 또한 [공격행위]가 악성코드의 위험한 정도를 판단하기 위한 가장 주요한 요소이며, [공격대상]은 위험한 정도를 판단하기 위한 가장 주요한 가중치 요소이다. 따라서 [공격대상]과 [공격행위]가 악성코드를 정의·분류하기 위한 가장 핵심적인 요소이며, 이에 대한 명확한 기준과 방안이 필요하다. 다음은 [공격대상]을 구분하는 2 단계 구조를 나타내는 표이다.

[표 3-13] 공격대상의 하위 분류

공격대상 하위 분류	설명
시스템	악성코드의 공격대상이 되는 유형이 악성코드가 감염된 시스템 자체일 경우 이 유형에 속한다.
사용자 정보	악성코드의 공격대상이 되는 유형이 악성코드가 감염된 시스템의 저장 정보일 경우 이 유형에 속한다.
시스템 사용자	악성코드의 공격대상이 되는 유형이 악성코드가 감염된 시스템의 사용자일 경우 이 유형에 속한다.
외부 장치	악성코드의 공격대상이 되는 유형이 악성코드가 감염된 시스템 이 외의 다른 장치일 경우 이 유형에 속한다.

가. 시스템

악성코드에 의해 감염이 이루어진 [시스템] 자체가 악성코드의 공격 대상인 경우 이 유형에 속한다. [시스템]의 경우 공격대상의 유형을 구체화하기 위해 하위 단계로 구분된다. [시스템]의 하위 단계로는 [시스템 장치 제어], [시스템 서비스 제어], [시스템 설정 제어]가 있다. 아래 표는 시스템의 하위 단계이다.

[표 3-14] 시스템의 하위 단계

시스템 하위단계	설명
시스템 장치 제어	악성코드의 공격대상이 시스템을 구성하는 장치들의 제어와 직접적인 관련이 있을 경우 이 세부 유형에 속한다.
시스템 서비스 제어	악성코드의 공격대상이 시스템을 통해 제공되는 서비스 제어와 직접적인 관련이 있을 경우 이 세부 유형에 속한다.
시스템 설정 제어	악성코드의 공격대상이 시스템의 설정 정보와 직접적인 관련이 있을 경우 이 세부 유형에 속한다.

(가) 시스템 장치 제어

시스템의 하위 단계 중 [시스템 장치 제어]는 시스템을 구성하는 장치들을 기준으로 다시 세분화 된다. [시스템 장치 제어]의 하위 단계는 [저장 장치], [입출력 장치], [네트워크 장치]가 존재한다.

[표 3-15] 시스템 장치 제어의 하위 단계

시스템 장치 제어 하위단계	설명
저장 장치	악성코드의 공격대상이 시스템에 연결된 저장 장치의 제어와 관련이 있을 경우 이 세부 유형에 속한다.
입출력 장치	악성코드의 공격대상이 시스템에 연결된 입출력 장치의 제어와 관련이 있을 경우 이 세부 유형에 속한다.
네트워크 장치	악성코드의 공격대상이 시스템에 연결된 네트워크 장치의 제어와 관련이 있을 경우 이 세부 유형에 속한다.

[저장 장치]는 시스템에 연결된 주기억장치, 보조 기억장치, 기타 이동식 디스크, 저장 기능을 가진 기타 장치 등을 나타낸다. 이 유형은 저장 장치의 제어를 방해하여 인식 가능한 메모리의 양을 줄이는 등 저장 장치가 비정상적으로 활동하도록 만드는 유형이다. [공격대상] 중 [시스템]에 속하는 저장 장치 유형의 경우 [저장 장치]에 저장되어 있는 정보 자체를 [공격대상]으로 하는 것이 아님에 유의한다. 이는 [공격대상] 중 [사용자 정보]에 속한다.

[입출력 장치]는 시스템에 연결된 모니터, 프린터, 마우스, 키보드, 스피커 등 사용자와 직접 커뮤니케이션하는 장치들을 말한다. 이 유형은 입출력 장치의 제어를 방해하거나 악성코드에 의해 동작하도록 조작하는 등 입출력 장치가 비정상적인 활동하도록 만드는 유형이다.

[네트워크 장치]는 외부의 다른 사용자나 시스템과 통신을 수행할 수 있도록 지원하는 장치이다. [네트워크 장치]의 제어를 방해하거나 악성코드에 의해 임의의 메시지를 송신/수신 하도록 조작하는 등 [네트워크 장치]가 비정상적인 활동을 하도록 만드는 유형이다. 외부에 존재하는 장치나 서비스에 대한 공격 또한 네트워크 장치를 통해 이루어지지만 [시스템]의 [네트워크 장치]를 [공격대상]으로 하는 것이 아님에 유의한다. [공격대상]을 분류함에 있어 이러한 대상을 공격하는 경우에는 [공격대상] 중 [외부 장치]에 속한다.

(나) 시스템 서비스 제어

[시스템]의 하위 단계 중 [시스템 서비스 제어]는 [시스템]을 구성하는 서비스 유형을 기준으로 다시 세분화 된다. [시스템 서비스 제어]의 하위 단계는 부팅 제어, 인터넷 서비스 제어, 네트워크 공유 제어, SQL 서비스 제어가 존재한다. 시스템 서비스 제어는 새로운 매체가 발생할 경우 이에 따라 크게 변동될 수 있는 유형이다. 따라서 충분한 확장성 고려가 요구된다.

[표 3-16] 시스템 서비스 제어의 하위 단계

시스템 서비스 제어 하위단계	설명
부팅 제어	악성코드의 공격대상이 시스템 부팅과 직접 연관이 있을 경우 이 세부 유형에 속한다.
인터넷 서비스 제어	악성코드의 공격대상이 인터넷 서비스와 직접 연관이 있을 경우 이 세부 유형에 속한다.
네트워크 공유 제어	악성코드의 공격대상이 네트워크 공유 서비스와 직접 연관이 있을 경우 이 세부 유형에 속한다.
SQL 서비스 제어	악성코드의 공격대상이 SQL 서비스와 직접 연관이 있을 경우 이 세부 유형에 속한다.

[부팅 제어]는 시스템의 가장 기본적인 서비스로 컴퓨터의 전원 공급이 이루어진 이후 정상적으로 운영체제 구동되기까지의 과정을 의미한다. 악성코드가 부팅 제어를 공격대상으로 삼는 경우에는 시스템의 강제적 종료, 재시작, 안전모드 시작, 로그오프 등의 결과가 발생한다. 특히 강제적 종료 및 재시작의 경우 주로 악성코드의 감염이 완료된 이후 악성코드가 정상적으로 동작하기 위해 수행되거나 사용자의 컴퓨터 이용을 방해하기 위한 목적으로 이루어진다.

[인터넷 서비스 제어]는 시스템을 통해 인터넷 서비스가 이루어지고 있는 경우 해당 서비스를 악용하거나 방해하는 것을 의미한다. 악성코드가 감염된 경우 인터넷 서비스의 동작을 방해할 수 있으며 해당 인터넷 서비스를 이용하는 사용자들에게 악성코드 전파 경로로 악용될 수 있다. 단, 인터넷 서비스를 통해 수집된 각종 저장 정보에 대한 공격은 인터넷 서비스 제어 유형에 속하지 않음에 유의한다.

[네트워크 공유 제어]는 시스템의 저장 정보를 동일한 네트워크 안에

존재하는 다른 시스템이나 원격지에 존재하는 시스템과 공유하기 위한 서비스를 의미한다. 악성코드가 감염된 경우 네트워크 공유 서비스의 동작을 방해할 수 있으며 해당 네트워크 공유 서비스를 이용하는 사용자들에게 악성코드 전파 경로로 악용될 수 있다. 단, 네트워크 공유를 통해 저장 및 공유되는 각종 저장 정보에 대한 공격은 [네트워크 공유 제어] 유형에 속하지 않음에 유의한다.

[SQL 서비스 제어]는 대량의 데이터 처리를 위한 SQL 서버 서비스를 의미한다. 악성코드가 감염된 경우 SQL 서비스 자체를 방해하거나 해당 SQL 서비스를 이용해 구성된 다른 서비스를 방해할 수 있다. SQL 서비스를 통해 저장 및 공유되는 각종 저장 정보에 대한 공격은 [SQL 서비스 제어] 유형에 속하지 않음에 유의한다.

(다) 시스템 설정 제어

[시스템]의 하위 단계 중 [시스템 설정 제어]는 제어하는 설정 정보를 기준으로 다시 세분화 된다. 시스템 설정 제어의 하위 단계로는 시스템을 구성하는 장치의 설정이나 시스템 운용 정책, 시스템 보안 설정으로 다시 세분화 된다.

[표 3-17] 시스템 설정 제어의 하위 단계

시스템 설정 제어 하위단계	설명
장치 설정 정보	악성코드의 공격대상이 시스템의 장치 설정 정보 변경에 있을 경우 이 세부 유형에 속한다.
시스템 운영 정책	악성코드의 공격대상이 시스템의 운영 정책 변경에 있을 경우 이 세부 유형에 속한다.
시스템 보안 설정	악성코드의 공격대상이 시스템의 보안 설정 정보 변경에 있을 경우 이 세부 유형에 속한다.

[장치 설정 정보]는 시스템을 구성하는 저장 장치, 입출력 장치, 네트워크 장치와 관련된 설정 정보들을 의미한다. 대표적으로 네트워크 설정 정보 중 IP, DNS, GATEWAY, .hosts 파일 등 다양한 설정 정보와 이를 관리하기 위한 파일들이 존재한다. 악성코드의 공격행위가 [장치 설정 정보]를 대상으로 하는 경우 이 유형에 속한다.

[시스템 운영 정책]은 시스템의 구동에서 시작하여 시스템 운영에 관련된 모든 정보를 의미한다. 시스템 부팅과 관련된 바이오스 설정, 윈도우즈 운영체제의 경우 서비스, 시작프로그램, 레지스트리이나 운영체제 업데이트 주기 등 각종 시스템 운영과 연관된 정책이 [시스템 운영 정책]에 속한다. 악성코드의 공격행위가 시스템 운영 정책을 대상으로 하는 경우 이 유형에 속한다.

[시스템 보안 설정]은 시스템의 보안 수준을 나타내는 설정을 의미한다. 대표적인 [시스템 보안 설정] 요소로는 웹 브라우저의 보안 설정, 네트워크 방화벽 설정, 안티바이러스 프로그램 사용 여부 등이 있다. 악성코드의 공격행위가 시스템 보안 설정을 대상으로 하는 경우 이 유형에 속한다.

나. 사용자 정보

악성코드에 의해 감염이 이루어진 시스템에 저장된 [사용자 정보]가 악성코드의 공격 대상인 경우 이 유형에 속한다. [사용자 정보]의 경우 [공격대상]의 유형을 구체화하기 위해 하위 단계로 구분된다. [사용자 정보]의 하위 단계로는 [사용자 입력 정보], [사용자 저장 정보], [사용자 시스템 정보], [사용자 시스템 사용 이력]이 있다. 아래 표는 사용자 정보의 하위 단계이다.

[표 3-18] 사용자 정보의 하위 단계

사용자 정보 하위단계	설명
사용자 입력 정보	악성코드가 시스템 사용자 입력 정보를 공격 대상으로 삼는 경우 이 세부 유형에 속한다.
사용자 저장 정보	악성코드가 시스템 사용자 저장 정보를 공격 대상으로 삼는 경우 이 세부 유형에 속한다.
사용자 시스템 정보	악성코드가 사용자의 시스템 정보를 공격 대상으로 삼는 경우 이 세부 유형에 속한다.
사용자 시스템 사용 이력	악성코드가 사용자 시스템 사용 이력을 공격 대상으로 삼는 경우 이 세부 유형에 속한다.

(1) 사용자 입력 정보

[사용자 입력 정보]는 사용자가 [시스템]에 정보를 입력하게 되면 일시적으로 존재하였다가 이내 사라지는 정보를 의미한다. [사용자 입력 정보]는 정보의 종류를 기준으로 다시 세분화 된다. [사용자 입력 정보]의 하위 단계는 [식별 정보], [금융 정보], [사회 활동 정보]가 존재한다. 이 하위 단계는 새로운 매체가 발생하여 취급되는 정보가 늘어날 경우 하위 단계가 세분화 될 여지가 존재하기 때문에 충분한 확장성을 필요하다.

[표 3-19] 사용자 입력 정보의 하위 단계

사용자 입력 정보 하위단계	설명
식별 정보	악성코드의 공격대상이 시스템에 입력되는 사용자의 식별 정보일 경우 이 세부 유형에 속한다.
금융 정보	악성코드의 공격대상이 시스템에 입력되는 사용자의 금융 정보일 경우 이 세부 유형에 속한다.
사회 활동 정보	악성코드의 공격대상이 시스템에 입력되는 사용자의 사회 활동 정보일 경우 이 세부 유형에 속한다.

[식별 정보]는 인터넷 상에서 사용자가 서비스 제공자로부터 자기가 자신이 맞음을 보이기 위해 사용하는 정보를 의미한다. 이 유형은 사용자가 입력하는 식별 정보 유출을 통해 타인이 사용자의 행세를 하며 사용자의 자산을 위협하거나 2 차적인 피해를 유발할 수 있는 유형이다. 대표적인 [식별 정보]로는 흔히 사용되는 서비스 이용 계정과 그 계정에 해당하는 비밀번호나 소프트웨어 설치를 위해 필요한 라이선스 코드, 그 외 식별에 사용되는 전산화된 정보들이 존재한다.

금융과 관계된 서비스가 많은 부분 전산화가 이루어지면서 인터넷이나 스마트폰을 이용한 비대면 서비스가 늘어나고 있다. 최근 들어 컴퓨팅 장치를 통해 처리되는 금융 정보의 빈도가 과거에 비해 월등히 높아지면서 금융 정보를 수집하는 것을 목적으로 하는 악성코드들이 생겨나기 했다. [금융 정보]는 안전한 비대면 서비스를 위해 만들어진 금융 목적의 식별 정보인 공인인증서 비밀번호, 보안카드 번호, 일회성 암호(one-time password)의 값 등이나, 계좌번호, 신용카드 번호 및 비밀번호와 CVV번호, 안심클릭 비밀번호 등 금융과 직결되는 정보들을 의미한다. 이러한 [금융 정보]는 사용자의 금품을 훔치거나 불법적인 대출에 이용하는 등 악용 가능성이 높은 개인정보이다. 따라서 이 유형의 정보는 유출될 경

우 사용자의 경제활동에 큰 피해를 줄 수 있다.

[사회 활동 정보]는 인터넷을 통해 사용자가 만들어가는 다양한 인간 관계에 의해 발생하는 정보이다. 대표적인 사회 활동의 사례로 인스턴트 메시지 서비스나 전자우편 서비스, 각종 온라인 커뮤니티 활동 등이 있으며 사회 활동 정보는 이러한 사회 활동을 통해 발생하는 정보들로 인스턴트 메시지의 대화내용, 전자우편의 본문 내용과 같은 정보들을 의미한다. [사회 활동 정보]는 앞서 정의한 식별 정보, 금융 정보와 같이 직접적인 이용 목적이 존재하는 정보는 아니지만, 이를 이용하면 쉽게 타인을 훔쳐 낼 수 있으며 그 결과 사회공학적 방법을 통한 사이버 공격 행위가 쉬워진다.

(2) 사용자 저장 정보

[표 3-20] 사용자 저장 정보의 하위 단계

사용자 저장 정보 하위단계	설명
주소록	악성코드의 공격대상이 시스템에 저장된 각종 주소록일 경우 이 세부 유형에 속한다.
저장파일	악성코드의 공격대상이 시스템에 저장된 각종 저장파일일 경우 이 세부 유형에 속한다.
프로그램	악성코드의 공격대상이 시스템에 저장된 개발한 소스파일, 유료 응용프로그램 등 일 경우 이 세부 유형에 속한다.

[사용자 저장 정보]는 사용자가 시스템에 저장하게 되는 각종 정보들을 의미한다. [사용자 정보]의 하위 단계 중 [사용자 저장 정보]는 정보의 종류를 기준으로 다시 세분화 된다. [사용자 저장 정보]의 하위 단계는 [주소록], [저장파일], [프로그램] 등이 존재한다. 이 하위 단계는 새로

운 매체가 발생하여 취급되는 정보가 늘어날 경우 하위 단계가 세분화될 여지가 존재하기 때문에 충분한 확장성을 필요하다.

[주소록]은 사용자 기억을 돕고 사용 편의를 위해 각종 응용프로그램이 제공하는 사용자와 관계된 타인들의 정보를 의미한다. [주소록]은 그 자체가 치명적인 비밀정보는 아니지만 악성코드 배포를 용이하게 하기 위한 수단으로 이용되거나 기타 사회공학적 사이버 공격을 용이하게 하기 위한 정보로 이용된다. 대표적인 예로 전자우편으로 전파되는 악성코드의 경우 감염된 시스템에 존재하는 마이크로소프트사의 전자우편 프로그램인 아웃룩의 주소록을 검색하여 악성코드를 첨부 후 이를 실행하도록 하는 간단한 문구를 추가하여 보내는 방법을 사용한다. 또한 메신저를 이용한 피싱이 극성을 부리던 시기에는 인스턴트 메시지 프로그램의 주소록 정보를 이용하여 입수한 사용자의 지인들을 공격 대상으로 삼는 등 사회공학적 사이버 범죄에 악용한 사례가 많이 존재한다.

[저장파일]은 응용프로그램을 통해 이루어지는 비즈니스, 엔터테인먼트, 창작 활동 등 다양한 서비스를 이용하면서 발생하게 되는 각종 파일들을 의미한다. 대표적인 예로 각종 문서파일(PDF, DOC, HWP, PPT, XLS 등), 오디오·비디오 파일, 그림파일, 캐드(CAD)파일 등 있다. 단, 지적재산권을 가진 파일 중 소스코드나 응용프로그램 자체는 [사용자 정보]의 하위 단계인 [프로그램]으로 분류되는 것에 유의한다.

[프로그램]은 비즈니스, 엔터테인먼트, 창작 활동 등 컴퓨터를 이용한 다양한 활동이 가능하도록 만드는 응용프로그램 자체를 의미한다. 이는 응용프로그램의 실행 파일 및 실행 권한(라이선스)를 의미하기도 하며, 경우에 따라 응용프로그램의 소스코드를 의미하기도 한다. 최근 들어 스마트 장치들이 늘어나면서 유사 컴퓨터 장치에서 동작하는 많은 종류의 응용프로그램이 개발·배포되면서 응용프로그램 재산권이 이슈화 되고 있다.

(3) 사용자 시스템 정보

[사용자 시스템 정보]는 악성코드가 감염된 시스템의 상태 정보를 의미한다. [사용자 정보]의 하위 단계 중 [사용자 시스템 정보]는 정보의 종류를 기준으로 다시 세분화 된다. [사용자 시스템 정보]의 하위 단계는 [가상화 정보], [운영체제 종류 및 버전], [시스템 구성 정보], [동작 프로세스 정보] 등이 존재한다. 이 하위 단계는 새로운 매체가 발생하여 취급되는 정보가 늘어날 경우 하위 단계가 세분화 될 여지가 존재하기 때문에 충분한 확장성을 필요하다.

[표 3-21] 사용자 시스템 정보의 하위 단계

사용자 시스템 정보 하위단계	설명
가상화 정보	시스템의 가상화 정보 추출이 공격목표인 경우 이 분류에 속한다.
운영체제 종류 및 버전	운영체제의 종류 및 버전 정보 추출이 공격목표인 경우 이 분류에 속한다.
시스템 구성 정보	시스템을 구성하는 장치 정보 추출이 공격목표인 경우 이 분류에 속한다.
동작 프로세스 정보	동작 프로세스 정보 추출이 공격목표인 경우 이 분류에 속한다.

가상화는 하나의 시스템에서 둘 이상의 운영체제를 동시에 서비스하기 위한 기술로 하이퍼바이저를 통해 각 개별 운영체제들이 운영된다. 이 경우 가상화된 운영체제는 직접적으로 시스템 자원이나 장치에 접근하는 것이 아니라 하이퍼바이저를 통해 접근이 이루어지며, 각각의 가상화된 운영체제는 자신만의 영역 내에서 동작하며 그 영역을 벗어나서는 접근하는 것이 개념적으로 불가능하다. 따라서 가상화된 운영체제에 악성코드가 감염되더라도 다른 운영체제에 영향을 입히지 못하거나 시스템 전체에 영향을 미치지 못한다. 이러한 특징을 이용하여 악성코드를 동적으로 분석하거나 샘플 코드 수집을 위해 가상화 시스템을 이용한다. [가상화 정보]는 현재 접근하고자 하는 대상 시스템의 가상화 여부와 가상화

에 이용된 하이퍼바이저의 정보를 의미한다. [가상화 정보]는 그 자체로 치명적인 정보는 아니지만, 악성코드의 행동 규정을 통해 취약점을 가진 하이퍼바이저에 대해 선택적으로 행위를 발현하도록 설계하는 데 이용할 수 있다.

[운영체제 종류 및 버전] 정보는 시스템을 구동하는 운영체제와 그에 버전에 대한 정보를 의미한다. 운영체제는 기능추가에 따라 새로운 취약점이 발생하거나, 밝혀진 취약점에 대한 패치가 이루어지는 등 꾸준히 변화하는 요소이다. [운영체제 종류 및 버전] 정보는 그 자체로 치명적인 비밀정보는 아니지만 이를 이용하여 악성코드가 공격 가능한 대상을 선정하거나 공격 행위가 선택적으로 발현하도록 구현하기 위해 이용될 수 있다.

[시스템 구성 정보]는 현재 시스템에 연결된 각종 장치들에 대한 정보를 의미한다. [시스템 구성 정보]는 그 자체로 치명적인 비밀정보는 아니지만, 악성코드의 접근 가능한 장치를 찾고 그에 대한 공격 수행 가능하도록 근거를 제공할 수 있다.

[동작 프로세스 정보]는 현재 시스템에서 구동중인 각종 프로세스에 대한 정보를 의미한다. [동작 프로세스 정보]는 그 자체로 치명적인 비밀정보는 아니지만, 동작 프로세스 정보를 바탕으로 취약점을 지닌 프로세스를 탐색하거나 안티바이러스 툴의 동작여부를 확인하기 위한 목적으로 사용될 수 있다.

(4) 사용자 시스템 사용 이력

[사용자 시스템 사용 이력]은 사용자가 시스템을 사용하면서 생성되는 각종 부가 정보를 의미한다. 사용자 정보의 하위 단계 중 [사용자 시스템 사용 이력]은 이력의 종류를 기준으로 다시 세분화 된다. 사용자 시스템 사용 이력의 하위 단계는 [인터넷 접속 정보], [서버 접속 정보], [트래픽 이력], [웹 히스토리], [프로세스 사용 이력] 등이 존재한다. 이 하위 단계는 새로운 매체가 발생하여 취급되는 정보가 늘어날 경우 하위

단계가 세분화 될 여지가 존재하기 때문에 충분한 확장성을 필요하다.

[표 3-22] 사용자 시스템 사용 이력의 하위 단계

사용자 시스템 사용 이력 하위단계	설명
인터넷 접속 정보	시스템 사용자가 인터넷을 접속 시 발생하는 일시적인 접속정보가 공격대상인 경우 이 유형에 속한다.
서버 접속 정보	시스템 사용자가 서버 접속 이력이 공격대상인 경우 이 유형에 속한다.
트래픽 이력	시스템 사용자가 트래픽 이력이 공격대상인 경우 이 유형에 속한다.
웹 히스토리	시스템 사용자가 방문한 웹 히스토리가 공격대상인 경우 이 유형에 속한다.
프로세스 사용 이력	시스템 사용자가 사용한 프로세스 히스토리가 공격대상인 경우 이 유형에 속한다.

[인터넷 접속 정보]는 연결성이 없는 HTTP 프로토콜의 한계를 보완하기 대응책으로 나온 저장정보로 쿠키나 세션이 이에 해당한다. 웹 서비스 중 사용자 식별이 필요한 경우 인증과정을 거친 이후 쿠키나 세션을 이용하여 그 정보를 일시적으로 저장할 수 있다. 쿠키의 경우 클라이언트 측에 저장되는 정보로 쿠키 정보가 유출되는 경우 악의적인 사용자가 쿠키 정보를 이용하여 다른 사용자의 행세를 할 수 있는 위험이 존재한다.

[서버 접속 정보]는 시스템 사용자가 외부에 존재하는 다른 시스템에 원격으로 접속한 경험에 대한 이력을 의미한다. 이 정보는 그 자체만으로 치명적인 비밀정보는 아니지만 사용자의 서버 접속 이력을 바탕으로 새로운 공격대상을 선정하는 대에 악용될 수 있다. 심각한 경우 서버의 주소 및 포트 번호 뿐만 아니라 접근 계정과 비밀번호가 저장된 경우 손

쉽게 서버에 접근이 가능해진다.

[트래픽 이력]은 악성코드가 감염된 시스템에서 발생한 트래픽의 이력이나 해당 시스템이 연결된 인접한 네트워크에서 발생한 트래픽에 대한 이력을 의미한다. 이 정보는 그 자체로 치명적인 비밀정보는 아니지만 네트워크 트래픽 분석을 통해 다양한 결론을 이끌어낼 수 있으며, 서버 접속 정보와 마찬가지로 다른 공격대상에 대한 탐색이 가능하다.

[웹 히스토리]는 사용자가 접근한 웹에 대한 이력을 의미한다. 이 정보는 그 자체로 치명적인 비밀정보는 아니지만 사용자의 관심 사항을 분석하거나 사용자가 자주 접근하는 취약한 사이트에 대한 탐색이 가능하다. 사용자의 취향 분석은 사회공학적 사이버 공격의 성공률을 높이는데 이용될 수 있으며, 사용자가 자주 접근하는 취약한 사이트에 대한 정보는 추가적인 악성코드 전파를 유도하는데 이용될 수 있다.

[프로세스 사용 이력]은 사용자가 사용한 프로세스들에 대한 최근 목록을 의미한다. 대표적인 예로 유닉스 계열의 운영체제의 경우 history 커맨드를 이용하여 사용자가 사용한 커맨드의 기록들을 열람할 수 있다. 이 정보는 그 자체로 치명적인 비밀정보는 아니지만 사용자의 평소 시스템 이용 습관을 분석하거나 사용자가 자주 사용하는 취약한 프로그램에 대한 탐색이 가능하도록 만드는 정보이다. 시스템 이용 습관의 분석이 가능할 경우 다른 사용자의 계정을 이용하여 시스템에 접근한 공격자는 보다 완벽하게 자신을 다른 사용자로 속일 수 있다. 최근 시그니처에 기반을 둔 악성코드 탐기 기술의 한계점이 논의되면서 행위 기반의 침입탐지 프로그램을 사용하는 시스템이 늘어났다. 하지만 공격자가 프로세스 사용 이력을 바탕으로 정상 범주 내에서 행위를 모사할 수 있게 될 경우 행위 기반의 침입탐지 프로그램을 무력화 할 수 있어 보다 완벽한 침입 행위를 수행할 수 있다.

(다) 시스템 사용자

악성코드에 의해 감염이 이루어진 시스템을 사용하는 사용자가 악성코

드의 공격 대상인 경우 이 유형에 속한다. [시스템 사용자]의 경우 공격 대상의 목적을 구체화하기 위해 하위 단계로 구분된다. [시스템 사용자]의 하위 단계로는 [경제적 손실], [이용 방해], [혼란 초래]가 있다. 아래 표는 시스템 사용자의 하위 단계이다. [시스템 사용자]가 공격 대상이 되는 공격 행위들의 경우 [시스템] 장치나 [사용자 정보]를 공격대상으로 하는 다른 유형과 달리 감염된 시스템과 직접적인 관련이 있는 공격으로 보기 어렵다.

[표 3-23] 시스템 사용자의 하위 단계

시스템 사용자 하위단계	설명
경제적 손실	악성코드가 시스템 사용자의 경제적 손실이 목적인 경우 이 세부 유형에 속한다.
이용 방해	악성코드가 시스템 사용자의 시스템 이용을 방해가 목적인 경우 이 세부 유형에 속한다.
혼란 초래	악성코드가 시스템 사용자를 혼란스럽게 할 목적인 경우 이 세부 유형에 속한다.

(1) 경제적 손실

악성코드가 [시스템 사용자]의 [경제적 손실]을 일으킬 목적으로 공격 행위를 수행하는 경우 이 유형에 속한다. 대표적인 경제적 손실 유형은 허위 바이러스 보고 등으로 인해 유료의 안티바이러스 서비스 결제를 유도하는 시스템이나 전화 연결 등을 통해 비용이 발생하도록 만드는 악성코드가 존재한다.

(2) 이용 방해

악성코드가 시스템을 사용하는 사용자의 시스템 이용을 방해할 목적으

로 공격행위를 수행하는 경우 이 유형에 속한다. [시스템 사용자]의 하위 단계 중 [이용 방해]는 공격 수단을 기준으로 다시 세분화 된다. 이용 방해의 하위 단계는 [시스템 자원], [시스템 UI], [프로세스] 등이 있다.

[표 3-24] 이용 방해의 하위 단계

이용 방해 하위단계	설명
시스템 자원	악성코드가 시스템 자원 고갈을 통해 사용자의 이용 불편을 초래하는 경우 이 유형에 속한다.
시스템 UI	악성코드가 시스템 UI 조작을 통해 사용자의 이용 불편을 초래하는 경우 이 유형에 속한다.
프로세스	악성코드가 프로세스 제어를 통해 사용자의 이용 불편을 초래하는 경우 이 유형에 속한다.

[시스템 자원]은 시스템을 구성하는 각종 장치 및 그로 인해 확보되는 시스템 자원들을 악성코드의 공격대상으로 이용하여 사용자의 이용 불편을 초래하는 유형이 이에 속한다. 대표적인 방법으로 시스템 자원 중 프로세스 처리 속도에 직접적인 영향을 미치는 CPU 자원, 메모리 자원 등을 고갈시킴으로써 사용자의 시스템 이용을 방해하는 방법의 공격 유형이 존재한다.

[시스템 UI]는 시스템 사용을 위해 제공되는 사용자의 UI 환경을 악성코드의 공격대상으로 이용하여 사용자의 이용 불편을 초래하는 유형이 이에 속한다. 대표적인 방법으로 입력장치인 키보드나 마우스의 비정상적인 동작을 유도하거나 출력장치의 비정상적인 동작을 유도함으로써 사용자의 시스템 이용을 방해하는 방법의 공격 유형이 존재한다.

[프로세스]는 사용자가 이용하는 프로세스를 악성코드의 공격대상으로 이용하여 사용자의 이용 불편을 초래하는 유형이다. 대표적인 방법으로 프로세스의 제어에 관여하여 사용자가 구동시키고자 하는 프로세스를 강제적으로 종료시키거나 비정상적으로 작동하도록 방해하는 종류의 공격 유형이 존재한다.

(3) 혼란 초래

악성코드가 시스템을 사용하는 사용자에게 혼란을 초래하는 것을 목적으로 공격행위를 수행하는 경우 이 유형에 속한다. 시스템 사용자의 하위 단계 중 [혼란 초래]는 행위를 기준으로 다시 세분화 된다. [혼란 초래]의 하위 단계는 [거짓 메시지]와 [허위 정보]가 있다.

[표 3-25] 혼란 초래의 하위 단계

혼란 초래 하위단계	설명
거짓 메시지	악성코드가 사용자에게 거짓 메시지를 전달하여 혼란을 초래하는 경우 이 유형에 속한다.
허위 정보	악성코드가 사용자에게 허위 정보를 전달하여 혼란을 초래하는 경우 이 유형에 속한다.

[거짓 메시지]는 메시지 전달이 가능한 서비스를 통해 악성코드가 생성한 메시지가 마치 다른 사용자가 보내는 메시지인 것처럼 보이도록 만드는 행위를 의미한다. 메시지 전달이 가능한 서비스로는 메신저, 전자메일 및 기타 휴먼 네트워크 서비스가 있다. [거짓 메시지]는 주로 악성코드 전파에 이용되거나 사용자의 결제 유도에 쓰인다. 그 외에 단순히 사용자혼란을 초래하기 위해 사용하기도 한다.

[허위 정보]는 시스템이나 응용프로그램에 의해 실제 상태와 다른 정보를 사용자에게 제공함으로써 혼란을 야기하는 경우이다. [허위 정보]의 예로는 악성코드에 감염되지 않았으나 마치 감염이 이루어진 것처럼 사용자에게 보고하는 등 사용자를 속이고 혼란에 빠트리는 것을 목적으로 하는 공격 유형이 이에 속한다.

나. 외부 장치

악성코드에 의해 감염이 이루어진 시스템이 자신 이외의 다른 장치를 공격하는 도구로 사용되는 경우 이 유형에 속한다. [외부 장치]의 경우 공격대상을 구체화하기 위해 하위 단계로 구분된다. [외부 장치]의 하위 단계로는 [서비스 서버], [로컬 네트워크]가 있다. 아래 표는 [외부 장치]의 하위 단계이다. 과거 사회적 이슈로 크게 대두되었던 분산 서비스 방해 공격을 수행하기 위해서는 [외부 장치]를 공격 대상으로 하는 악성코드의 감염이 필수적이다. 이는 [외부 장치]를 공격 대상으로 하는 악성코드들에 대해 사회적인 경각심을 가지고 충분한 노력을 기울여 탐지·치료에 힘쓴다면 분산 서비스 방해 공격을 어느 정도 예방할 수 있음을 의미한다.

[표 3-26] 외부 장치의 하위 단계

외부 장치 하위단계	설명
서비스 서버	악성코드 공격목적이 외부 장치가 제공하는 서비스인 경우 이 세부 유형에 속한다.
로컬 네트워크	악성코드가 공격목적이 시스템이 연결된 로컬 네트워크인 경우 이 세부 유형에 속한다.

(1) 서비스 서버

악성코드의 공격대상이 악성코드에 감염된 시스템 자체가 아닌 외부에 존재하는 장치의 서비스인 경우 이 유형에 속한다. 악성코드가 특정 서비스를 제공하는 외부 서버에 대해 공격을 수행하는 경우 [서비스 서버 동작 제어]와 [서비스 서버 제공 서비스]로 나뉘며 이는 아래 [표 3-27]과 같다.

[표 3-27] 서비스 서버의 하위 단계

서비스 서버의 하위단계	설명
서비스 서버 동작 제어	악성코드 공격목적이 외부 장치의 동작과 관련된 경우 이 세부 유형에 속한다.
서비스 서버 제공 서비스	악성코드가 공격목적이 외부 장치가 제공하는 서비스인 이 세부 유형에 속한다.

악성코드에 감염된 시스템이 [외부 장치]의 보안 취약점을 이용하여 시스템 동작을 제어할 수 있는 메시지를 생성, 전달하는 경우 이 유형에 속한다.

악성코드에 감염된 시스템이 [외부 장치] 제공하는 서비스를 방해하기 위해 특정 행위를 수행하는 경우 이 유형에 속한다. [외부 장치]가 제공하는 서비스를 방해하기 위한 방법으로 대량의 트래픽을 생성·전송 하는 방법이 대표적이다.

(2) 로컬 네트워크

악성코드의 공격대상이 악성코드에 감염된 시스템에 인접한 네트워크 또는 네트워크에 존재하는 다른 장치를 [공격대상]으로 하는 경우 이 유형에 속한다. 악성코드는 감염에 성공한 장치를 이용하여 물리적으로 단절된 네트워크에 접근하거나 로컬 네트워크의 구성 정보를 수집하는데 이용할 수 있다. 로컬 네트워크는 네트워크 접근 경로, 네트워크 구성 정보로 나뉜다.

[표 3-28] 로컬 네트워크의 하위 단계

로컬 네트워크 하위단계	설명
네트워크 접근 경로	악성코드 공격목적이 로컬 네트워크 안에 존재하는 다른 시스템인 경우 이 세부 유형에 속한다.
네트워크 구성 정보	악성코드가 공격목적이 로컬 네트워크를 구성하는 시스템의 정보인 경우 이 세부 유형에 속한다.

[네트워크 접근 경로]는 특정 시스템의 저장 정보나 해당 시스템을 사용하는 대상에 대한 공격을 수행하기 보다는 해당 시스템이 연결된 인근 네트워크 내에 존재하는 또 다른 시스템을 공격하기 위한 접근 경로로 이용하는 경우 이 유형에 속한다. 악성코드가 물리적 또는 논리적인 형태로 단절된 네트워크에 대해 악성코드 전파 및 특수한 목적의 악성행위 수행을 위해서는 물리적인 저장 매체를 이용한 침입이 필요하다. 이러한 공격대상을 이용한 악성코드 사례로 스텝스넷이 있다.

[네트워크 구성 정보]는 특정 시스템의 저장 정보나 해당 시스템을 사용하는 사용자에게 대한 공격을 수행하기 보다는 해당 시스템이 연결된 인근 네트워크 내에 존재하는 다른 시스템들의 정보 수집을 목적으로 하는 경우 이 유형에 속한다.

4. 공격행위

연관성 정보 구조 1 단계 중 [공격행위]는 특정 시스템에서 동작하는 악성코드의 사이버 공격 행위를 정의·분류하기 위한 내용이다.

모든 악성코드들은 하나 이상의 악성행위를 수행한다. 앞서 서술한 것처럼 악성코드 [공격대상]은 일부 악성코드의 공격 행위를 한정할 수 있다. 따라서 본 보고서에서는 악성코드를 정의·분류하기 위한 [공격행위]

를 정리함과 동시에 [공격대상]과 [공격행위] 간에 연관성에 대해서도 부분적으로 서술한다. 공격행위는 악성코드의 위험 수준을 정의하는 가장 주요한 요소로 정확한 정의·분류가 필요하다. [공격행위]를 구체화하기 위해 [공격행위]에 이용된 수단을 기준하여 하위 단계로 세분화된다. [공격행위]를 구체화하기 위한 요소로는 [네트워크], [시스템], [프로세스], [입출력장치 오작동], [파일시스템]이 있다. 아래 표는 [공격행위]의 하위 단계를 서술한 표이다.

[표 3-29] 공격행위의 2 단계

공격행위의 하위단계	설명
네트워크	악성코드의 공격행위가 네트워크를 이용하여 이루어지는 경우 이 분류에 속한다.
시스템	악성코드의 공격행위가 시스템을 이용하여 이루어지는 경우 이 분류에 속한다.
파일시스템	악성코드의 공격행위가 파일시스템을 이용하여 이루어지는 경우 이 분류에 속한다.
입출력장치 오작동	악성코드의 공격행위가 입출력장치의 오작동을 이용하여 이루어지는 경우 이 분류에 속한다.
프로세스	악성코드의 공격행위가 프로세스를 이용하여 이루어지는 경우 이 분류에 속한다.

가. 네트워크

네트워크는 [공격행위]의 2 단계 구조 중 하나로 네트워크를 이용한 공격행위를 수행하는 악성코드를 정의·분류하기 위한 분류이다. 악성코드의 네트워크 [공격행위]는 아래 표와 같은 여섯 가지 하위 단계를 갖는다.

[표 3-30] 공격행위의 네트워크

네트워크 하위단계	설명
정보 유출	악성코드의 공격행위 결과 사용자 정보가 외부로 유출되는 경우 이 분류에 속한다.
서버 접속	악성코드의 공격행위가 서버 접속을 통해 이루어지는 경우 이 분류에 속한다.
대량 트래픽 전송	악성코드의 공격행위가 대량의 트래픽 생성 및 전송이 필요한 경우 이 분류에 속한다.
가로채기	악성코드의 공격행위 결과 사용자가 전송하는 정보를 공격자 또한 수신할 수 있는 경우 이 분류에 속한다.
전송지연	악성코드의 공격행위 결과 사용자가 외부로 정보를 전송할 때 시간을 지연시키는 경우 이 분류에 속한다.
스팸	악성코드의 공격행위가 스팸메시지 생성 및 전송이 필요한 경우 이 분류에 속한다.

(1) 정보 유출

악성코드의 [공격행위] 결과 사용자 정보가 외부로 유출되는 경우 이 유형에 속한다. 네트워크의 정[보 유출]의 경우 유출되는 정보 종류에 따라 하위 단계로 구분된다. [정보 유출]의 하위 단계로는 [사용자 식별 정보], [사용자 금융 정보], [사용자 사회 활동 정보], [주소록], [저장파일], [프로그램], [시스템 정보], [시스템 사용 이력]이 있다. 아래 표는 [정보 유출]의 하위 단계이다. 정보 유출의 하위 단계는 앞서 기술한 [공격대상]의 [공격대상] 중 사용자 정보의 항목들과 동일한 항목들로 구성된다. 앞서 [공격대상]에서는 악성 행위의 대상에 대해 정의 하였으며, 이후 [공격행위]에서는 동일한 [공격대상]에 대한 악성 행위에 대해 정의한다.

[표 3-31] 유출 정보의 종류

유출 정보의 종류	설명
사용자 식별 정보	악성코드의 공격행위 결과 유출되는 정보가 사용자 식별 정보인 경우 이 분류에 속한다.
사용자 금융 정보	악성코드의 공격행위 결과 유출되는 정보가 사용자 금융 정보인 경우 이 분류에 속한다.
사용자 사회 활동 정보	악성코드의 공격행위 결과 유출되는 정보가 사용자 사회 활동 정보인 경우 이 분류에 속한다.
주소록	악성코드의 공격행위 결과 유출되는 정보가 주소록인 경우 이 분류에 속한다.
저장파일	악성코드의 공격행위 결과 유출되는 정보가 저장 파일인 경우 이 분류에 속한다.
프로그램	악성코드의 공격행위 결과 유출되는 정보가 프로그램인 경우 이 분류에 속한다.
시스템 정보	악성코드의 공격행위 결과 유출되는 정보가 시스템 정보인 경우 이 분류에 속한다.
시스템 사용 이력	악성코드의 공격행위 결과 유출되는 정보가 시스템 사용 이력인 경우 이 분류에 속한다.

(가) 사용자 식별 정보

악성코드가 [사용자 식별 정보]를 수집하여 외부로 전송하는 경우 이 유형에 속한다. [사용자 식별 정보]는 [사용자 입력 정보] 중 하나로 서비스 요청 시 올바른 접근 권한을 가진 사용자가 맞는지 확인하기 위한 용도로 사용되는 정보이다. [사용자 식별 정보]는 기본적으로 사용자가 기억하고 있는 정보를 사용하는 것을 전제로 하기 때문에 이러한 정보는 주로 시스템에 저장되기 보다는 식별이 필요한 경우 일시적으로 사용자에게 입력받는 형태로 쓰인다. 하지만 최근 사용자의 편의를 위해 사용

자의 계정 및 암호를 저장하는 기능을 가진 응용프로그램이 많아지면서 [사용자 식별 정보]는 경우에 따라 시스템 내에 저장되기도 한다.

악성코드가 [사용자 식별 정보]를 유출하기 위해서는 사전에 [사용자 식별 정보]를 수집하는 과정이 필요하다. 이 과정은 키 로거(key logger)를 사용하여 일시적인 입력정보를 저장하거나 시스템에 식별 정보가 저장된 파일을 찾아 추출하는 과정이 필요하다. 또한 수집 대상이 시스템의 정보에 국한되지 않을 경우 통신 프로토콜의 취약점을 이용하여 외부 시스템의 식별 정보를 수집하여 저장하기도 한다.

(나) 사용자 금융 정보

악성코드가 [사용자 금융 정보]를 수집하여 외부로 전송하는 경우 이 유형에 속한다. 컴퓨터를 이용한 인터넷 뱅킹이나 스마트폰 앱(app, application의 약어; 스마트폰 응용프로그램을 지칭)을 이용한 뱅킹 서비스 등 비대면 서비스의 이용이 늘어나면서 컴퓨터나 스마트 기기에 설치된 악성코드를 이용하여 [사용자 금융 정보]를 유출하려는 시도가 늘어나고 있다.

악성코드가 사용자 금융 정보를 유출하기 위해서는 사전에 [사용자 금융 정보]를 수집하는 과정이 필요하다. 이 과정은 키 로거(key logger)를 사용하여 일시적인 입력정보를 저장하거나 시스템에 금융 정보가 저장된 파일을 찾아 추출하는 과정이 필요하다. 또한 수집 대상이 시스템의 정보에 국한되지 않을 경우 통신 프로토콜의 취약점을 이용하여 외부 시스템의 금융 정보를 수집하여 저장하기도 한다.

특히 금융 서비스에서 인증을 위해 사용되는 공인인증서의 경우 해독이 어려운 공개키 알고리즘을 통해 만들어지지만 실제로 공인인증서를 이용할 때에는 사용자가 입력한 암호를 이용하여 사용자를 식별하기 때문에 공인인증서의 악용할 위협이 존재한다.

금융 서비스의 경우 비교적 보안 체계가 잘 이루어져 있기 때문에 식별에 이용할 수 있는 정보 유출이 성공하더라도 대개 사용할 수 있는 시

간이 짧고, 오류가 발생 시에 비대면 서비스가 정지되는 등 다양한 보안 기술이 포함되어 있다.

(다) 사용자 사회 활동 정보

악성코드가 [사용자 사회 활동 정보]를 외부로 전송하는 경우 이 유형에 속한다. 메신저, 전자메일, 기타 휴먼 네트워크 서비스 등을 통해 송/수신 되는 메시지는 주로 개인적인 내용으로 대화에 참여하는 사용자들만이 볼 수 있다.

악성코드가 [사용자 사회 활동 정보]를 유출하기 위해서는 사전에 [사용자 사회 활동 정보]를 수집하는 과정이 필요하다. 이 과정은 키 로거(key logger)를 사용하여 일시적인 입력정보를 저장하거나 시스템에 사회 활동 정보를 저장한 파일을 찾아 추출하는 과정이 필요하다. 또한 수집 대상이 시스템의 정보에 국한되지 않을 경우 통신 프로토콜의 취약점을 이용하여 외부 시스템의 사회 활동 정보를 수집하여 저장하기도 한다.

[사용자 사회 활동 정보]는 다른 사용자 입력 정보에 비해 비교적 낮은 위험 수준을 갖는다. 이는 다른 경우에 비해 정보 유출 시 악용되는 유형이 적으며 직접적인 피해도 미미한 수준이기 때문이다.

(라) 주소록

악성코드가 사용자 [주소록]을 외부로 전송하는 경우 이 유형에 속한다. 사용자의 [주소록]은 그 자체로 가치가 높은 정보가 아니기 때문에 정보 유출 행위의 대상이 되는 경우는 적으며, 간혹 개인정보 수집 및 다른 사회공학적 사이버 공격을 위한 정보로 이용하기 위해 수집하는 경우가 있다.

악성코드가 주소록 유출을 위해서는 시스템에 저장된 주소록의 탐색 및 해당 파일에 대한 접근 과정이 필요하다.

(마) 저장파일

악성코드가 시스템 [저장파일]을 외부로 전송하는 경우 이 유형에 속한다. [저장파일]은 공격대상유형에 대한 정의 시 “응용프로그램을 통해 이루어지는 비즈니스, 엔터테인먼트, 창작 활동 등 다양한 서비스를 이용하면서 발생하게 되는 각종 파일”로 정의하였다. 즉 각종 문서파일, 오디오·비디오 파일, 그림파일, 캐드파일 등 파일에 기록된 정보 자체만으로도 큰 가치를 갖는 대상으로 악성코드에 의한 정보유출 중 가장 위험 수준이 높은 [공격대상]이다. [저장파일] 유출 결과 군사 작전 정보가 적군의 손에 넘어가거나 등 큰 규모의 피해가 발생할 위험이 존재한다.

악성코드가 [저장정보] 유출을 위해서는 시스템에 저장된 저장정보의 탐색 및 해당 파일에 대한 접근 과정이 필요하다.

(바) 프로그램

악성코드가 시스템에 저장된 프로그램과 관련된 파일들을 외부로 전송하는 경우 이 유형에 속한다. [프로그램]은 실행 파일 또는 실행 권한(라이선스)과 응용프로그램의 소스코드를 의미한다.

악성코드가 프로그램 유출을 위해서는 시스템에 저장된 프로그램의 탐색 및 해당 파일에 대한 접근 과정이 필요하다.

(사) 시스템 정보

악성코드가 시스템의 정보를 외부로 전송하는 경우 이 유형에 속한다. [시스템 정보]는 가상화 정보, 운영체제 종류 및 버전, 시스템 구성 정보, 동작 프로세스 정보 등 시스템의 주요 상태 정보를 의미한다. [시스템 정보]는 시스템의 보안 취약점을 분석하기 위한 목적으로 수집될 수 있다. 하지만 그 자체만으로 악용이 불가능한 정보로 유출 대상으로 보기

어렵다.

(아) 시스템 사용 이력

악성코드가 [시스템 사용 이력]을 외부로 전송하는 경우 이 유형에 속한다. 시스템 사용 이력은 인터넷 접속 정보, 서버 접속 정보, 트래픽 이력, 웹 히스토리, 프로세스 사용 이력과 같은 시스템 동작 과정에서 발생하는 패턴들에 대한 정보를 의미한다. [시스템 사용 이력] 또한 그 자체로 악용이 불가능한 정보로 유출 대상으로 보기 어렵다.

(2) 서버 접속

악성코드의 [공격행위] 결과 외부 시스템에 접속하여 공격 행위를 수행하는 경우 이 유형에 속한다. [네트워크]의 [서버 접속]은 접속 목적에 따라 하위 단계로 구분된다. [서버 접속]의 하위 단계로는 [악성코드 다운로드], [명령 수신]이 있다. 아래 표는 서버 접속의 하위 단계이다.

[표 3-32] 서버 접속의 목적

서버 접속 목적	설명
악성코드 다운로드	악성코드가 추가적인 악성코드 설치를 위해 서버로부터 악성코드를 수신하는 경우 이 분류에 속한다.
명령 수신	악성코드가 공격자의 명령 수신을 위해 서버에 접속하는 경우 이 분류에 속한다.

(가) 악성코드 다운로드

시스템에 설치된 추가적인 악성행위 수행을 위해 악성코드 배포 서버

로부터 악성코드를 다운로드 및 설치과정을 시도하는 경우 이 분류에 속한다.

(나) 명령 수신

특정 악성코드는 악성코드가 설치된 시스템에 상주하면서 이를 조정할 수 있는 커맨드 서버를 이용하여 다양한 악성행위를 명령한다. 이때 커맨드 서버의 명령을 수신하기 위해 특정 서버에 접속하는 경우 이 분류에 속한다. 예를 들어 IRC 서버를 이용한 IRC 봇(bot)의 경우 악성코드 감염 시 악성코드는 사용자들 몰래 IRC 서버에 접속하여 대화내용을 바탕으로 악성행위를 수행한다.

(3) 대량 트래픽 전송

악성코드의 [공격행위] 결과 대량의 트래픽이 생성 및 전송 되는 경우가 유형에 속한다. [네트워크]의 [대량 트래픽 생성]은 생성 목적에 따라 하위 단계로 구분된다. [대량 트래픽 생성]의 하위 단계로는 [DoS 공격], [로컬 네트워크 마비] 등이 있다. 아래 표는 대량 트래픽 생성의 하위 단계이다.

[표 3-33] 대량 트래픽의 생성 목적

대량 트래픽 생성 목적	설명
DoS 공격	악성코드가 DoS 공격을 위해 대량 트래픽을 생성·발송하는 경우 이 분류에 속한다.
로컬 네트워크 마비	악성코드가 로컬 네트워크 마비를 위해 대량 트래픽을 생성·발송하는 경우 이 분류에 속한다.

(가) DoS 공격

악성코드가 DoS 공격을 위해 대량 트래픽을 생성하는 경우 이 분류에 속한다. [DoS 공격]은 단순한 형태로 동작하나 뚜렷한 대응 방안이 존재하지 않아 공격이 발생하는 경우 피해를 입기 쉽다.

(나) 로컬 네트워크 마비

악성코드가 로컬 네트워크를 마비시킬 목적으로 대량 트래픽을 생성하여 로컬 네트워크를 공격하는 경우 이 분류에 속한다. [로컬 네트워크 마비] 공격은 단순한 형태로 동작하나 문제가 되는 시스템을 즉각 차단하는 조치를 취할 수 없는 경우 공격이 발생한 이후 대응이 어렵다.

(4) 가로채기

악성코드의 [공격행위] 결과 사용자가 전송하는 정보가 공격자 또한 수신 및 확인할 수 있는 경우 이 유형에 속한다. 악성코드에 의해 감염된 후 사용자가 전송하는 정보가 변조되어 공격자에게로 전송 될 수 있다. 이에 더하여, 시스템 사용자가 전송하는 정보를 수신자와 함께 공격자가 수신할 수도 있다. [가로채기]를 수행하는 악성코드의 경우 [정보 유출] 악성코드 보다 상대적으로 유출하는 정보가 제한적이며 위협적이지는 않지만 전송되는 정보 자체가 기밀을 요하는 정보일 경우 [가로채기]를 수행하는 악성코드 또한 매우 높은 위협이 될 수 있다.

(5) 전송지연

악성코드의 [공격행위] 결과 사용자가 전송하는 정보가 지연되는 경우가 유형에 속한다. [전송지연]은 시스템 사용자가 정보를 외부로 송신하는 경우 해당 정보를 공격자가 연기시킬 수 있다. 공격자가 지연시키는

정보가 시급을 요하는 정보일 경우 [전송지연]을 수행하는 악성코드는 높은 위협요인이 된다.

(6) 스팸

[스팸]은 악성코드의 목적이 감염된 시스템의 자원을 사용하여 스팸 메시지를 생성하고 전송하는 악성코드 분류 요소이다. 스팸 메시지는 시스템 자원을 강제로 사용할 뿐만 아니라 스마트폰과 같은 모바일 기기에서 과금이 부과되는 SMS 또는 MMS를 전송할 수도 있다. 다음은 스팸에 속하는 종류를 나타내는 표이다.

[표 3-34] 스팸의 종류

스팸 하위 단계	설명
전자우편	악성코드가 감염된 해당 시스템의 자원을 사용하여 스팸메일을 보내는 악성코드가 이 분류에 속한다.
SMS/MMS	E-mail이 아닌 SMS 및 MMS의 스팸 메시지를 전송하는 악성코드가 해당 분류에 속한다.
댓글	광고 목적으로 불특정 다수의 웹 사이트에 광고성 댓글을 작성하는 악성코드가 해당 분류에 속한다.

(가) 전자우편

악성코드가 해당 시스템의 자원을 악용하여 스팸메일을 전송하는 악성코드를 말한다. 본 악성코드는 기존에 내제하고 있던 전자우편 목록을 사용하여 스팸 메일을 전송하거나 사용자 정보를 탈취하여 해당 정보를 바탕으로 스팸메일을 전송할 수 있다.

(나) SMS/MMS

광고 목적의 SMS/MMS를 전송하는 악성코드를 뜻한다. 악성코드에 감염된 시스템이 스마트폰과 같은 모바일 단말기인 경우 해당 악성행위로 인하여 요금이 발생될 수 있다. 전자우편과 같이 내포하고 있는 주소록을 통하여 SMS/MMS를 전송하거나 해당 시스템의 주소록을 통하여 SMS/MMS를 전송할 수 있다.

(다) 댓글

광고가 목적인 악성코드로 다양한 웹 사이트에 광고성 댓글 및 사회 불안감 조성 문구를 작성하는 악성코드를 뜻한다.

나. 시스템

[표 3-35] 공격행위의 시스템

시스템 하위단계	설명
강제 시스템 제어	악성코드의 공격행위 결과 시스템의 전원과 관련하여 강제적인 제어가 발생할 경우 이 분류에 속한다.
시스템 자원 제어	악성코드의 공격행위 결과 시스템의 자원과 관련하여 강제적인 제어가 발생할 경우 이 분류에 속한다.
네트워크 설정 변경	악성코드의 공격행위 결과 시스템의 네트워크 설정이 변경되는 경우 이 분류에 속한다.
시스템 설정 변경	악성코드의 공격행위 결과 시스템의 설정이 변경되는 경우 이 분류에 속한다.

시스템은 [공격행위]의 2 단계 구조 중 하나로 시스템에 대한 [공격행

위]를 수행하는 악성코드를 정의·분류하기 위한 분류이다. 악성코드의 시스템 [공격행위]는 위 [표 3-35]와 같은 네 가지 하위 단계를 갖는다.

(1) 강제 시스템 제어

[표 3-36] 강제 시스템 제어

강제 시스템 제어의 하위단계	설명
시스템 종료/정지	악성코드의 공격행위가 시스템 종료/정지 등의 부팅 이상인 경우 이 분류에 속한다.
사용자 권한 변경	악성코드의 공격행위가 사용자 권한 변경인 경우 이 분류에 속한다.
원격제어	악성코드의 공격행위가 원격제어인 경우 이 분류에 속한다.
기구류 작동	악성코드의 공격행위가 기구류 작동인 경우 이 분류에 속한다.

[강제 시스템 제어]는 시스템의 권한을 탈취하여 시스템의 설정 등을 변경하여 사용자가 원치 않는 행위를 수행하는 요소이다. [강제 시스템 제어]는 크게 [시스템 종료], [사용자 권한 변경], [원격 제어], [기구류 작동]으로 나뉜다. [기구류 작동]은 CD-ROM 열기/닫기가 아닌 실린더와 같은 FA(Factory Automation)의 실린더 작동과 같은 자동화 시스템의 오작동을 말함에 주의한다. 위 [표 3-36]은 강제 시스템 제어의 하위 단계를 나타내는 표이다.

(가) 시스템 종료/정지

사용자의 의지와는 무관하게 시스템의 작동을 종료하거나 정지·안전모드로 부팅 등 정상적인 종료 및 실행을 방해하는 분류가 해당 분류에 속

한다.

(나) 시스템 권한 변경

관리자를 제외한 다른 사용자들의 시스템 사용 권한을 조종하는 행위가 해당 분류에 속한다. 해당 공격 행위는 공격자가 관리자 권한과 동일한 권한을 갖는 사용자를 생성하여 악성행위를 수행할 수 있도록 한다.

(다) 원격제어

강제로 시스템 제어권을 탈취하여 원격에서 시스템을 사용하는 악성코드가 해당 분류에 속한다.

(라) 기구류 작동

단순 PC의 CD-ROM 열기/닫기가 아닌 실린더, 모터와 같은 자동화된 FA시스템에서 사용되는 기계 기구류의 강제 실행이 해당 분류에 속한다. 본 요소를 포함하는 악성코드는 자동화된 산업현장에서 기기의 오작동을 통하여 금전적인 피해를 입히거나 인명피해를 입힐 수 있다.

(2) 시스템 자원 제어

시스템 자원 제어는 시스템의 잔여 자원을 고갈시켜 원활한 실행을 방해하거나 물리적 장치 인식 장애를 일으켜 물리적 자원의 사용을 발해하는 악성코드가 속한다. 다음 표는 시스템 자원 제어의 하위 단계를 나타내는 표이다.

[표 3-37] 자원 관리 하위 단계

자원관리 하위단계	설명
잔여 자원 고갈	악성코드의 공격행위가 시스템 잔여 자원 고갈인 경우 이 분류에 속한다.
시스템 인식 장애	악성코드의 공격행위가 시스템 인식 장애인 경우 이 분류에 속한다.

(가) 잔여 자원 고갈

네트워크와 연결된 상태가 아니더라도 과도한 메모리를 사용하는 프로세스를 동작시키거나 동일한 프로세스를 여러 번 실행시켜 시스템의 자원을 고갈시키는 악성코드가 이 분류에 속한다.

(나) 시스템 인식 장애

메모리, 하드디스크, 이동식 디스크와 같은 물리적인 매체의 인식을 일으켜 해당 매체를 사용할 수 없도록 하는 악성코드가 해당 분류에 속한다.

(3) 네트워크 설정 변경

네트워크 설정 변경은 네트워크와 관련된 설정 정보들을 강제로 변경시켜 악성코드가 원활한 악성행위를 수행할 수 있도록 하는 악성코드가 이 분류에 속한다. 단순 시작사이트 변경에서부터 제 3의 악성행위를 위한 네트워크 설정 상태를 변경할 수 있다. 다음 표는 네트워크 설정을 변경할 수 있는 파일 및 변경 가능한 악성행위를 나타낸다.

[표 3-38] 네트워크 설정 변경 하위 단계

네트워크 설정 변경 파일 및 가능한 악성행위	설명
.hosts파일	네트워크 설정 변경 대상이 .host파일인 경우 이 분류에 속한다.
DNS 정보 변경	네트워크 설정 변경 대상이 DNS 정보 변경인 경우 이 분류에 속한다.
신뢰할 수 있는 사이트	네트워크 설정 변경 대상이 신뢰할 수 있는 사이트인 경우 이 분류에 속한다.
인터넷 보안 수준 변경	네트워크 설정 변경 대상이 인터넷 보안 수준 변경인 경우 이 분류에 속한다.
공유 폴더 설정 변경	네트워크 설정 변경 대상이 공유 폴더 설정 변경인 경우 이 분류에 속한다.
홈페이지 고정	네트워크 설정 변경 대상이 홈페이지 고정인 경우 이 분류에 속한다.
특정 포트 오픈	네트워크 설정 변경 대상이 특정 포트 오픈인 경우 이 분류에 속한다.

(가) .hosts 파일

도메인과 IP로 변경 시켜주는 호스트 파일을 변경하여 사용자가 접속하려고 하는 웹 사이트와는 다른 웹 사이트로 연결하는 경우 이 분류에 속한다. .host파일을 변경하는 경우 인지도 있는 웹 사이트와 동일한 웹 페이지를 생성 후 정보를 유출하거나 금전적 손해를 입히는 웹 피싱과 함께 발생할 수 있다.

(나) DNS 정보 변경

DNS를 변경하여 .host 파일 변경과 같이 특정 도메인에 대한 웹 피싱

을 수행하거나 안티 바이러스 업체 홈페이지 및 업데이트 서버 접속 자체를 방해하는 악성코드가 이 분류에 속한다.

(다) 신뢰할 수 있는 사이트

인터넷 익스플로러의 신뢰할 수 있는 사이트에 강제로 등록시키는 악성코드로 이후 특정 사이트에 접속할 때 보안검사를 수행하지 않도록 설정하여 원활한 악성행위를 목표로 하는 악성코드가 해당 분류에 속한다.

(라) 인터넷 보안 수준 변경

인터넷 보안 수준을 변경하여 악성코드가 감염된 사이트 열람 시 경고 메시지를 출력하지 않게 하는 악성코드가 이 분류에 속한다.

(마) 공유 폴더 설정 변경

공유 폴더 설정을 변경하여 보안에 취약한 환경으로 변경하는 악성코드가 이 분류에 속한다.

(바) 홈페이지 고정

사용자 모르게 시작 홈페이지를 강제로 변경하여 악성코드에 감염되게 하거나 홈페이지 광고를 목적으로 하는 악성코드가 이 분류에 속한다.

(사) 특정 포트 오픈

제 3의 공격을 위하여 외부에서 접속할 수 있는 특정 포트를 개방하는 악성코드가 이 분류에 속한다.

(4) 시스템 설정 변경

[시스템 설정 변경]은 [시스템 설정 언어], [서비스 등록/변경], [사용자 계정정보 변경] 등 시스템에 전반적인 설정 정보들을 변경하는 악성코드를 뜻한다. 시스템 설정 변경은 사용자로 하여금 시스템 접근 자체를 거부당하거나 UI 강제 변경으로 불편함을 초래한다.

[표 3-39] 시스템 설정 변경 하위 단계

시스템 설정 변경 하위 단계	설명
사용자 계정 변경	관리자 및 시스템 사용자의 비밀번호를 변경하거나 특정 사용자를 추가하는 악성행위가 이 분류에 속한다.
언어 및 글꼴	시스템에 설정된 기본 언어를 변경하거나 글꼴 등 문자 관련된 설정 정보를 변경하는 악성코드가 이 분류에 속한다.
레지스트리 변경	악성코드의 자동 실행 또는 악성코드가 필요로 하는 응용프로그램의 실행을 위하여 레지스트리를 변경하는 악성코드가 이 분류에 속한다.
서비스 등록/변경	악성코드를 자동실행하거나 프로세스가 종료될 때 자동실행을 위하여 서비스를 등록하는 악성코드가 이 분류에 속한다.
시작 프로그램 등록/변경	시스템이 부트될 때 자동실행을 위해 시작 프로그램에 등록하는 악성코드가 해당 분류에 속한다.
UI 변경	운영체제에서 제공하는 UI를 변경하여 사용자가 원하는 작업을 수행할 수 없게 하거나 방해하는 목적의 악성코드가 이 분류에 속한다.

(가) 사용자 계정 변경

관리자 및 시스템 사용자의 비밀번호를 변경하거나 특정 사용자를 추가하는 악성행위를 나타낸다. [사용자 계정 변경]은 사용자의 불편을 초래하기 위한 단순한 목적으로 수행되거나 악성코드가 감염된 시스템에 공격자가 접근할 수 있는 백도어 설치를 위한 설정으로 이용되기도 한다. 백도어 설정을 위해 사용자 계정이 변경되는 경우에는 최상위 권한의 사용자 계정을 추가적으로 생성함으로써 사용자가 인지하지 못하는 접근 계정을 만들거나 실제로 존재하는 사용자 계정 중 일부의 권한을 최상위 권한으로 상승시킨 후 이를 이용하는 유형이 있다.

(나) 언어 및 글꼴

시스템에 설정된 기본 언어를 변경하거나 글꼴 등 문자 관련된 설정 정보를 변경하여 사용자로 하여금 불편을 초래하는 악성코드를 일컫는다. 운영체제의 기본 언어가 변경될 시에 정상적으로 응용프로그램이 동작하지 않아 불편이 발생하거나 운영체제의 메뉴를 알아볼 수 없어 다시 환경을 복구하는 데에 어려움이 발생한다.

(다) 레지스트리 변경

악성코드의 자동 실행 또는 악성코드가 필요로 하는 응용프로그램의 실행을 위하여 레지스트리를 변경하는 악성코드를 나타낸다. 레지스트리가 악용되는 경우는 서비스 등록, 부팅 시 자동 실행, 파일 확장자 연결, 윈도우 탐색기(Explorer)와 관련된 행위, 인터넷 익스플로러 관련 행위, 윈도우 설정과 관련된 행위, 네트워크 설정 관련 행위, 레지스트리 존재 여부 확인, 레지스트리 값 참조 등이 있다. 이는 2 장에서 서술하였다.

(라) 서비스 등록/변경

시스템이 부트 될 때 악성코드를 자동실행하거나 악성행위를 수행하는 프로세스가 종료될 때 자동실행을 위하여 서비스를 등록하는 악성코드를 나타낸다. 서비스 응용프로그램의 경우 윈도우 부팅 시 사용자 로그인 일어나기 전에 시작되는 응용프로그램으로 시작 프로그램을 이용한 자동 실행보다 높은 위험을 갖는다.

(마) 시작 프로그램 등록/변경

시스템이 부트될 때 악성코드의 자동실행을 위해 시작 프로그램에 등록하는 악성코드가 해당 분류에 속한다.

(바) UI변경

운영체제에서 제공하는 UI를 변경하여 사용자가 원하는 작업을 수행할 수 없게 하거나 방해하는 목적의 악성코드가 이 분류에 속한다. UI를 변경하는 악성코드는 데스크바 감추기, 작업관리자 감추기, 단축키 사용 불가 등 시스템을 사용하는데 문제는 없지만 사용자로 하여금 불편을 초래한다.

(5) 디스플레이 설정 변경

디스플레이 설정 변경은 출력장치에 대한 오작동을 일으키는 것이 아닌 운영체제 상에서 디스플레이에 관한 설정 정보를 수정하는 것을 말한다. PC와 같은 시스템이 아닌 모바일 디바이스의 경우 단순 장난을 위하여 디스플레이 설정을 변경하는 악성코드가 유포되고 있다. 다음은 디스플레이 설정을 변경하는 악성코드의 하위 단계를 나타내는 표이다.

[표 3-40] 디스플레이 설정 변경 하위 단계

디스플레이 설정 변경 하위 단계	설명
바탕화면 변경	디스플레이 설정을 변경하는 악성행위 중 바탕화면을 변경하는 악성코드가 이 분류에 속한다.
스크린세이버 변경	디스플레이 설정을 변경하는 악성행위 중 스크린세이버를 변경하는 악성코드가 이 분 류에 속한다.
색 변경	디스플레이 설정을 변경하는 악성행위 중 색을 변경하는 악성코드가 이 분류에 속한 다.
해상도 변경	디스플레이 설정을 변경하는 악성행위 중 디스플레이 해상도를 변경하는 악성코드가 이 분류에 속한다.
창 설정 변경	디스플레이 설정을 변경하는 악성행위 중 윈도우의 타이틀을 변경하거나 크기, 색상을 변경하는 악성코드가 이 분류에 속한다.

다. 파일시스템

파일시스템은 악성코드가 파일생성, 파괴, 변조 등 파일관련 악성행위를 수행하는 악성코드들이 해당 분류에 속한다. 또한 시스템 장치 파괴를 위한 부트섹터 삭제, 파일 저장 시 오류를 유발하는 악성행위 등 또한 파일시스템에 속한다. 다음 표는 파일시스템 하위 단계를 나타내는 표이다.

[표 3-41] 파일시스템 하위 단계

파일시스템 하위 단계	설명
파일생성	악성코드의 공격행위가 파일생성이면 이 분류에 속한다.
파일파괴	악성코드의 공격행위가 파일파괴면 이 분류에 속한다.
파일변조	악성코드의 공격행위가 파일변조면 이 분류에 속한다.

(가) 파일생성

사용자의 동의 없이 파일을 생성하는 악성코드들을 뜻한다. 생성되는 파일은 바로가기 아이콘, 즐겨찾기가 될 수 있으며 해당 파일을 통하여 다른 악성코드를 다운로드 될 수 있다. 또한 파일 생성의 경우 시스템 정보를 외부로 유출시키기 위하여 생성하는 로그파일, 특정악성행위를 실행하거나 과금 유발을 위해 생성된 실행파일 등이 있다. 다음 표는 [파일생성]으로 수행할 수 있는 악성행위를 나타낸다.

[표 3-42] 파일생성 종류

종류	설명
아이콘 생성	즐거찾기 아이콘, 바탕화면에 바로가기 아이콘을 생성하는 악성코드를 나타낸다.
로그 파일 생성	외부로 유출시킬 정보를 로그화하는 악성코드를 나타낸다.
실행파일	다른 악성행위를 수행하기 위하여 실행파일을 생성하는 악성코드를 나타낸다.

(나) 파일파괴

시스템이 부트되지 못하게 하거나 중요 파일의 단순 삭제를 수행하는 악성코드이다. 파일 파괴로 수행할 수 있는 악성행위는 강제로 디스크를 포맷하거나 다음 부팅 시 부트되지 못하도록 부트섹터를 훼손하는 행위, 시스템의 오작동을 일으키거나 실행할 수 없도록 시스템 파일을 삭제하는 행위 등이 있다. 다음 표는 파일파괴의 종류를 나타내는 표이다.

[표 3-43] 파일파괴 종류

종류	설명
파일삭제	시스템파일, 바로가기 아이콘, 문서파일 등 시스템에 저장된 파일을 삭제하는 악성행위를 말한다. 디스크 포맷, 시스템 파일 삭제 등 위협이 높은 악성행위가 포함된다.
부트섹터삭제	부트섹터가 저장된 MBR의 값을 수정하여 하드디스크를 통해 부트하는 것을 방해하는 악성행위이다. 하드디스크를 쓸 수 없도록 할 수 있기 때문에 여타 다른 악성행위보다 높은 위협행위라고 할 수 있다.

(다) 파일변조

악성코드의 실행을 위하여 DLL과 같은 시스템 파일을 변조하는 악성코드를 나타낸다. 시스템 파일의 변조를 통하여 프로세스에 의해 로드되는 라이브러리 파일 등에 악성코드를 삽입하여 변조된 라이브러리 파일이 수행될 때 악성코드가 실행되게 할 수 있다. 정상작동 되던 특정 파일을 변조하여 악성코드가 실행되게 하는 악성코드를 [파일변조]라 할 수 있다.

[표 3-44] 파일변조 종류

종류	설명
실행파일 변조	실행파일 후위에 자신을 인젝션하여 실행파일(숙주)이 실행될 때 동반 실행되는 형태를 취하는 악성코드를 말한다.
시스템파일 변조	시스템파일을 수정하여 정상 프로세스가 악성행위를 수행할 때 동반 실행될 수 있도록 할 수 있다. 또한 시스템파일을 변조할 경우 정상적인 부팅을 할 수 없게 된다.
매크로파일 변조	많은 어플리케이션들이 매크로파일을 사용하여 복잡한 명령을 수행한다. 악성코드는 어플리케이션이 사용하는 매크로를 통하여 악성행위를 수행할 수 있다. 매크로파일 변조는 악성코드가 매크로파일을 변조하여 악성행위를 수행하는 것을 나타낸다.

라. 입출력장치 오작동

입출력장치 오작동은 시스템에 연결되어 있는 입력장치, 출력장치에 대하여 오작동을 일으키는 악성행위를 말한다. 주로 사용자를 놀라게 하는 조크 바이러스가 입출력장치 오작동을 수행한다. 하지만 차량에 입출력장치 오작동을 일으키는 악성코드가 설치된다면 위험한 상황을 발생시킬 수 있다. 현재 차량 엔터테인먼트 기술 중 차량의 속도에 따라서 음악 재생 볼륨을 서서히 올리는 기술이 도입되어 있다. 하지만 이런 시스템을 악용하여 갑자기 음악을 크게 틀게 되면 사용자의 혼란은 곧 인명피해로 변질 수 있다. 다음은 입출력장치 오작동과 관련된 악성행위를 나타내는 표이다.

[표 3-45] 입출력장치 오작동 종류

입출력장치 오작동 하위 단계	설명
미디어 제어	악성코드가 미디어 장치에 대해 오작동을 일으키는 경우 이 분류에 속한다.
입력장치 제어	악성코드가 입력장치에 대해 오작동을 일으키는 경우 이 분류에 속한다.
출력장치 제어	악성코드가 출력장치에 대해 오작동을 일으키는 경우 이 분류에 속한다.
기타 장치 제어	악성코드가 기타 장치에 대해 오작동을 일으키는 경우 이 분류에 속한다.

(가) 미디어 제어

미디어 제어는 단순히 시스템에 포함된 스피커를 통하여 비프음을 재생, 미디어(영상, 음악)를 사용자 동의 없이 재생시키거나 귀신소리, 자극적인 소리와 같은 혐오스러운 소리를 재생시키는 악성코드가 해당 분류에 속한다.

(나) 입력장치 제어

입력장치 제어는 시스템에 입력장치로 연결된 장치들에 대하여 악성행위를 수행하는 악성코드를 나타낸다. 입력장치가 인식되지 않게 하지 않고 단순 입력장치에 대하여 제대로 된 입력이 되지 않게 하거나 방해하는 행위이다. 입력장치 제어는 키 입력 방해, 마우스 반전, 마우스 커서 사라짐과 같은 악성행위를 수행할 수 있다.

(다) 출력장치 제어

입력장치와 마찬가지로 시스템에 연결된 출력장치에 대하여 악성행위를 수행하는 악성코드를 나타낸다. 출력장치 제어를 수행하는 악성코드는 프린터 동작, 디스플레이 이상동작(화면 떨림, 절전모드) 등을 수행할 수 있다.

(라) 기타 장치 제어

기타 장치 제어는 입출력장치가 아닌 단순 하드웨어에 대하여 악성행위를 수행하는 악성코드를 말한다. 시스템에 장착된 LED를 점등시키거나 CD-ROM을 작동 시키는 등 사용자로 하여금 혼란을 유발하는 악성코드를 나타낸다.

마. 프로세스

프로세스에 대한 악성행위 수행은 프로세스에 대하여 직접적으로 프로세스를 실행하게 하거나 종료시키는 [제어]와 간접적으로 단순 프로세스의 목록, 상태, 정보 등을 관찰하는 [모니터링]으로 나뉜다. 프로세스에 대한 악성행위를 수행하는 악성코드는 프로세스를 주기적으로 실행시키거나 실행되지 않도록 할 수 있다. 다음은 프로세스와 관련된 악성행위를 나타낸 표이다.

[표 3-46] 프로세스 하위 단계

프로세스 하위 단계	설명
제어	악성코드가 프로세스에 대한 제어를 수행하는 경우 이 분류에 속한다.
모니터링	악성코드가 프로세스에 대해 모니터링을 수행하는 경우 이 분류에 속한다.

(1) 제어

제어는 프로세스들에 대해 관리자 권한으로 제어하는 악성코드를 의미한다. 프로세스를 제어하는 악성코드의 경우 프로세스에 대하여 실행을 방해하거나 프로세스를 직접적으로 종료하거나 실행시킬 수 있으며 주기적인 실행에 대해서도 관여할 수 있다.

(2) 모니터링

모니터링은 프로세스들에 대하여 직접적인 악성행위는 수행하지 않지만 프로세스의 목록을 스캔하거나 프로세스들의 상태를 지속적으로 모니터링하는 악성코드를 뜻한다. 또한 모니터링의 경우 특정 프로세스가 실행중인지 스캔할 수 있어 탈취하려는 정보와 관련된 프로세스의 실행여부, 안티 바이러스 소프트웨어의 실행여부 등을 확인할 수 있다.

5. 가중치

연관성 정보 구조는 해당 악성코드의 위험 정보를 판단하기 위하여 [전파경로], [목표대상], [자가보호], [잠재위험]이라는 가중치를 적용한다. 이러한 가중치는 동일한 악성행위를 수행하는 악성코드라도 전파경로의 다양화, 목표대상이 사회에 미치는 영향력, 분석을 방해하는 기술, 악성코드에 감염 후 제 3의 공격에 악용될 수 있는 잠재위험에 따라서 악성코드의 위험 정도가 다르다는 것을 나타낸다. 이후 연관성 정보 구조는 악성코드의 악성행위와 가중치를 바탕으로 위험지수 도출 및 악성코드 목적, 변종 악성코드를 나타낼 수 있다.

가. 전파경로

연관성 정보 구조 중 [전파경로]는 시스템이 감염된 이후 다른 시스템

을 감염시키기 위하여 전파되는 경로를 나타낸다. 이는 동일한 단계 중 하나인 [감염경로]와 많은 부분 흡사하나 의미상의 차이가 존재한다. [감염경로]는 확보한 악성코드 샘플이 악성코드의 [전파경로]에 따라 자동적인 형태로 감염이 되는 경우와 다양한 사람들의 개입으로 인해 인위적인 경로를 통해 감염되는 모든 경우를 포함하는 개념이다.

[전파경로]는 이를 구체화하기 위해 하위 단계로 세분화된다. [전파경로]를 구체화하기 위한 요소로는 [전파 이용 매체], [사용자 의존도]가 있다. 아래 표는 [전파경로]의 하위 단계를 서술한 표이다.

[표 3-47] 전파경로의 2 단계

전파경로의 하위단계	설명
전파 이용 매체	전파 이용 매체는 특정 시스템에 악성코드가 유입되는데 이용된 매체의 종류를 정의·분류하기 위한 요소이다.
사용자 의존도	특정 시스템에 악성코드가 전파되는 과정이 시스템 사용자 행위에 얼마나 의존적인지를 정의·분류하기 위한 요소이다.

(1) 전파 이용 매체

[전파 이용 매체]는 [전파경로]의 2 단계 구조 중 하나로 악성코드가 특정 시스템에 감염되는 과정에서 이용된 매체에 대해 정의·분류하기 위한 분류이다. 악성코드의 전파에 이용되는 매체는 새로운 매체가 발생함에 따라 얼마든지 변화할 수 있다. 따라서 이 분류는 충분한 확장성을 요한다. 전파 이용 매체는 [감염경로]의 [감염 이용 매체]와 동일한 세부 구조를 가지고 있으나 의미상에 차이를 갖는다.

[표 3-48] [전파경로]의 전파 이용 매체

전파 이용 매체 분류	전파 이용 매체 설명
이동식 저장매체	데이터 저장 및 이동을 위한 장치로 플래시 메모리, 광학디스크, 자기 디스크(외장 하드) 등
모바일 디바이스	데이터 저장 및 이동이 가능하며, 이동성을 제공함과 동시에 컴퓨팅 장치에 연결이 가능한 단말기
하드 디스크	큰 용량 장점으로 데이터 저장/백업이 주목적, 데이터 이동을 위해 사용가능
P2P	서버를 거치지 않고 사용자간 데이터 공유를 위해 사용하는 데이터 공유 기법
웹하드	웹 인터페이스를 통해 접근 가능한 디스크
FTP 클라이언트/서버	FTP를 이용한 데이터 전송
웹서버	HTTP를 이용한 데이터 전송
메신저	네트워크를 통해 대화 및 데이터 공유
게시판	특정 사용자들 간에 정보를 업로드/다운로드 할 수 있는 커뮤니티 공간
블로그	개인의 관심사를 업로드하는 공간
전자우편	전자 메일
액티브X	웹과 프로그램을 연결하기 위한 기술
BHO	웹 브라우저 이용 편의를 위한 프로그램
클라우드 컴퓨팅	유틸리티 컴퓨팅 개념의 필요한 자원을 서버로부터 대여하여 사용하는 서비스

(2) 사용자 의존도

[사용자 의존도]는 [전파경로]의 2 단계 구조 중 하나로 악성코드가 특정 시스템에 전파되는 과정이 사용자의 행위에 어느 정도 의존적인지를 정의·분류하기 위한 요소이다. 악성코드의 [사용자 의존도]는 새로운 매체가 발생하더라도 변화의 폭이 좁아 확장성이 크게 요구되지 않는다.

아래 표는 [전파경로]의 [사용자 의존도]들을 정리한 것이다.

[표 3-49] 전파경로의 사용자 의존도

사용자 의존도 분류	사용자 의존도 설명
사용자 직접 명령	파일의 다운로드 또는 복사를 목적으로 사용자가 직접 시스템에 요청하는 경우
사용자 간접 명령	특정 서비스 이용을 위해 파일의 다운로드 또는 복사가 간접적으로 시스템에 요청되는 경우
비(非) 의존적	사용자의 의지와는 무관하게 파일의 다운로드 또는 복사가 이루어지는 경우

나. 자가보호

[표 3-50] 악성코드 자가보호 종류

자가보호 종류	설명
난독화	난독화를 수행하여 분석가로부터 분석되는 것을 예방 및 지연 하는 악성코드가 해당 분류에 속한다.
분석도구 탐지	악성코드가 분석되는 것을 방지하기 위하여 가상화 환경, 특정 프로세스 동작 확인 등을 통하여 분석하는 악성코드가 해당 분류에 속하게 된다..
루트킷	시스템 권한을 획득하는 방법들과 안티 바이러스 제품으로부터 스캔당하는 것을 방지하는 악성코드가 해당 유형에 속하게 된다.

[자가보호]는 악성코드가 분석가 혹은 안티바이러스 소프트웨어로부터 자기 자신을 보호하는 기능들을 일컫는다. [자가보호]는 분석가가 악성코드 자체를 분석할 수 없도록 하는 분석도구 탐지와 악성코드가 분석될 때 알아볼 수 없도록 소스코드를 숨기는 난독화, 그리고 시스템 내부에서 악성코드 자체를 은닉시키기 위한 루트킷 기술로 나뉜다. 위 [표

3-50]은 [자가보호] 종류를 나타내는 표이다.

(1) 난독화

난독화는 기존 상업 소프트웨어에 사용된 소스코드가 외부로 유출되는 것을 방지하기 위한 방법이었으나 악성코드 제작자들 사이에서는 악성코드의 분석을 방해하는 기술로 사용되는 기술이다. 난독화가 적용된 악성코드의 경우 새로운 악성코드가 나타났을 시 이를 분석하고 대응하는데 높은 시간과 비용이 소요된다.

기존의 난독화는 소스코드의 흐름에는 영향을 미치지 않는 범위에서 코드를 수정하거나 제어 흐름을 바꾸는 것을 통해 디컴파일과 리버싱 과정을 어렵게 하는 소스코드 레벨의 난독화를 포함한다. 하지만 소스코드 레벨의 난독화는 패킹과 다형성 기법 보다 낮은 분석시간을 요하기 때문에 본 보고서에서는 다형성 기법, 패킹을 의미하는 것으로 간주한다.

다음 표는 난독화의 종류와 설명을 나타낸 표이다.

[표 3-51] 난독화의 종류

난독화 종류	설명
다형성 기법	특정 파일이 악성코드에 감염될 때 마다 악성코드의 형태가 바뀌는 악성코드를 의미한다. 기본적으로 복호화 루틴이 다형성을 띠고 있어 감염될 때 마다 악성코드 형태가 달라진다.
패킹	실행파일을 압축하여도 실행될 수 있도록 한 방식으로 다양한 패킹을 적용함으로써 분석을 방해할 수 있다.

(가) 다형성 기법

다형성 기법은 악성코드가 감염될 때 마다 형태가 변하여 감염 여부를

판단하기 어렵도록 할 수 있다. 많은 안티 바이러스 업체들은 바이러스 코드의 특징을 기준으로 악성코드 감염 여부를 판단하기 때문에 이를 우회할 수 있는 방식인 다형성 기법이 적용된 악성코드가 제작 및 유포되고 있다.

(나) 패킹

패킹은 악성코드를 실행할 수 있는 파일형식으로 압축하여 내부 소스 코드를 숨기는 역할을 한다. 기존의 상업용 혹은 평가판 패킹툴이 제공되고 있다. 패킹을 사용함으로써 악성코드의 소스코드를 은닉하여 신종 및 변종 악성코드에 대해 유기적인 대응을 할 수 없도록 한다.

[표 3-52] 패킹의 종류

패킹의 하위 종류	설명
다중패킹	기존에 존재하는 패킹방식을 사용하여 다중 패킹을 수행하는 것을 뜻한다.
커스텀패킹	커스텀 패킹은 기존에 존재하는 패킹방식을 사용하는 것이 아니라 악성코드 제작자가 새로운 패킹형식을 제작하여 악성코드에 적용한 것을 뜻함

악성코드 제작자는 여러 번 패킹을 수행할 수 있다. 분석가는 패킹된 악성코드를 언패킹하기 위하여 적용된 패킹형식(패커)을 알아내는 분석작업이 선행되어야 한다. 또한 악성코드 제작자는 여러 개의 패커를 사용할 수 있기 때문에 다수(적게는 두 개 이상, 많게는 10개 이상)의 패커가 사용될 경우 분석가들의 업무효율은 급격히 떨어진다.

o 다중패킹

다중패킹이 적용된 파일과 커스텀패킹이 적용된 파일을 직접 실행하면

서 악성코드라고 판단하고 시그니처를 제작할 수 있다. 하지만 악성코드를 분석하여 해당 악성코드의 특징과 목표 등 다양한 정보를 수집하기 위해서는 패킹된 악성코드를 언패킹하는 작업이 필수적이다.

o 커스텀패킹

커스텀패킹은 기존에 존재하는 패킹방식이 아니므로 이를 언패킹하기 위해서는 패킹형태를 분석하고 언패커를 새로 만들어야 한다. 다중 패킹의 경우 기존에 존재하는 패커들에 맞는 언패커를 사용하면서 패킹된 악성코드를 복원할 수 있지만 악성코드 제작자가 직접 제작한 커스텀 패커의 경우 패킹된 형태를 보고 직접 언패커를 제작해야 하기 때문에 다중 패킹보다 많은 분석시간이 소요된다.

(2) 분석도구 탐지

악성코드는 자신이 분석되는 것을 방지하기 위하여 분석도구가 실행되면서 나타나는 특징들을 검색하여 분석도구를 무력화 시키거나 우회할 수 있다. 다음 표는 분석 도구를 탐지하는 레벨에 관한 표이다.

[표 3-53] 분석도구 종류

분석도구 종류	설명
API	시스템에서 제공하는 디버깅관련 API 함수나 플러그인을 사용하여 현재 시스템에서 분석도구가 실행 중인지 판단하는 방식
Process	Process분류는 디버거가 수행중일 때 나타나는 프로세스의 특징을 이용하여 분석도구를 탐지하는 방식
CPU Instruction	CPU Instruction은 CPU레벨에서 디버거가 실행될 때 나타나는 특징을 바탕으로 현재 분석도구가 실행 중인지 판단하는 방식

(가) API

악성코드가 시스템 내부에서 구동될 때 악성코드는 API함수나 플래그 값을 사용하여 현재 분석 도구가 실행중인지 판단할 수 있다. 다음 표는 디버거 관련 API와 설명을 나타낸다. API를 제공하는 시스템에서 악성코드는 API를 통하여 변수 값으로 현재 분석 도구가 실행 중인지 알 수 있으며 분석 도구 탐지 시 해당 프로세스를 종료하거나 우회하는 방법을 사용할 수 있다.

[표 3-54] 분석도구 탐지에 사용 가능한 API

API 종류	설명
ProcessDebugFlags	ProcessDebugFlags는 호출되었을 때, 만약 디버거가 실행 중이라면 FALSE를 반환한다
HideThreadFromDebugger	Process분류는 디버거가 수행중일 때 나타나는 프로세스의 특징을 이용하여 분석도구를 탐지하는 방식이다.
OutputDebugString	디버거는 HideThreadFromDebugger 클래스가 호출된 쓰레드로부터 어떠한 이벤트 발생도 받을 수 없다. 이벤트의 예로는 breakpoint, 프로그램 종료 등이 있다.
IsDebuggerPresent	OutputDebugString은 현재 활성화된 디버거가 없는 경우에 에러를 발생시킨다.
IsDebugged	만약 디버거가 실행 중이라면, 1을 반환하고 반대의 경우라면 0을 반환한다. 이 API는 BeingDebugged 바이트 Flag를 확인한다.
NtGlobalFlags	프로세스가 실행되면 시스템이 flag를 설정하는데 프로세스가 디버깅 중일 때 설정하는 flag가 다르다. 디버깅 중이라면, ntdll안에 heap 조작 루틴을 제어하는 플래그들이 설정된다.

(나) Process

[표 3-55] 분석 도구 탐지를 위한 프로세스 상태 확인

Process Exploitation	설명
Open Process	디버거에 의해 프로세스가 실행 중일 때, 프로세스의 실행 권한이 올바르게 재설정되지 않았다면 프로세스는 SeDebugPrivilege 권한을 얻게 된다. 이 권한이 있으면 특정 프로세스의 핸들을 얻을 수가 있다.
Parent Process	보통 경우에 사용자는 windows shell이 만든 window로부터 프로세스를 시작한다. 이 때, 새로 만든 프로세스의 부모 프로세스는 Explorer.exe이다. 따라서 현재 프로세스의 부모 프로세스의 pid와 Explorer.exe의 pid와 비교를 해서 디버거가 실행을 했는지 Explorer.exe가 실행했는지 알 수 있다.
Self Debugging	메인 프로세스가 자식 프로세스를 만들어서 자기 자신을 디버깅하게 만든다. 그러면, 다른 디버거가 attach될 수가 없다.
UnhandledExceptionFilter	UnhandledExceptionFilter는 exception을 처리할 알맞은 핸들러가 있지 않을 때 호출되는 기본 핸들러이다. 만약 디버거가 실행 중이라면, 프로세스가 종료되고 실행 중이 아니라면, 프로세스는 다시 실행된다.
NtQueryObject	NtQueryObject 함수는 호출되면 호스트 시스템과 현재 프로세스에 관한 정보를 출력한다. 그 중에는 DebugObject 항목이 있다. 이 항목으로 디버거 실행 여부를 파악할 수 있다.

Process(Exploitation)는 디버거가 실행 중일 때 프로세스가 가지는 특징을 이용하여 분석 도구를 탐지한다. 프로세스 상태를 확인하면 악성코

드는 현재 시스템이 악성코드 분석을 위한 가상머신 상태인지, 안티 바이러스 프로세스가 구동중인지 판단할 수 있다. [표 3-55]는 프로세스와 관련된 Process Exploitation들이다.

(다) CPU Instruction

악성코드는 CPU 레벨에서 CPU instruction을 통하여 분석 도구 및 가상화 환경을 탐지할 수 있다. CPU instruction을 통하여 디버거가 실행 중인지 판단하는 방식은 다음과 같다.

[표 3-56] 분석 도구 탐지를 위한 프로세스 상태 확인

CPU Instruction	설명
Interrupt 2D	만약 디버거가 실행 중이 아니면, INT 2D 명령어를 실행하면 exception이 발생한다. 하지만 만약 디버거가 실행 중이라면, exception이 발생하지 않고 디버거의 종류에 따라 다른 일이 발생한다.
Stack Segment	PUSH SS, POP SS 명령어를 사용해서 스택 세그먼트를 조작하면, 디버거의 오작동을 유발할 수 있다.
Instruction Prefix	디버거가 명령어 prefix를 핸들하는 방법을 이용하는 것이다. 특정 코드를 실행했을 때, 디버거가 prefix를 건너뛰게 할 수 있다. 반면에 디버거가 실행 중이지 않다면, SEH가 exception을 catch 한다.
Int3	INT3 명령어를 코드 중간에 삽입하고 실행하면 디버깅하지 않는 경우에는 exception 핸들러가 실행되고 원래 코드로 다시 실행 흐름이 바뀐다.

(3) 루트킷

루트킷의 기존 의미는 관리자가 관리의 목적으로 시스템 내부에 정상적인 경로를 거치지 않고 바로 접속 할 수 있도록 생성한 경로를 기존에 설정한 보안 경로를 우회하여 접속하는 것을 뜻하였다. 하지만 오늘날에는 루트권한을 획득하기 구현된 기능들을 루트킷이라 한다.

루트킷은 악성코드 자체를 은닉시키기 위하여 Hooking, DLL 또는 코드 Injection, 시스템 변조를 수행한다. 악성코드는 루트킷을 통하여 안티 바이러스 소프트웨어의 업데이트를 무력화 시키거나 안티 바이러스 프로세스를 강제로 종료시키고 악성코드의 흔적을 남기지 않는 MBR에 악성코드를 물리적으로 기록할 수 있다.

다음은 루트킷이 수행하는 악성 행위를 나타낸다.

[표 3-57] 루트킷의 종류

루트킷의 종류	설명
Hooking	어플리케이션과 OS간의 메시지 전송을 중간에 확인하는 후킹은 콜 IAT, IDT, SSDT 등의 후킹을 통하여 사용자 모르게 악성행위를 수행할 수 있다.
시스템 변조	시스템 파일, DLL 파일등의 변조를 통하여 악성코드가 안티 바이러스 소프트웨어보다 하위에서 구동되어 자신을 숨기는 방식을 말한다.
Injection	강제로 코드를 인젝션 하여 공격자가 원하는 행위를 수행하게 하는 악성코드가 해당 분류에 속한다.

(가) Hooking

다음 표는 후킹 방식들은 루트킷이 악성행위를 수행하기 위해 필요한 정보들을 수집하거나 명령어를 수행하기 위해 수행되는 후킹들을 나타낸다.

[표 3-58] Hooking의 종류

Hooking 종류	설명
Hooking IAT	IAT는 특정 DLL이 export하는 함수들의 주소를 저장하는 콜 테이블이다. 각각 DLL 마다 IAT가 따로 있다. 후킹을 하려는 프로세스의 메모리 공간을 알아낸 다음, IAT의 위치를 알아내서 IAT 값을 원하는 메모리 주소 값으로 수정할 수가 있다.
Hooking IDT	IDT는 디스크립터의 배열이다. 각각의 디스크립터는 크기가 8 bytes이다. Interrupt 핸들러를 후킹하기 위해서는 먼저 IDT의 위치를 알아내야 한다. IDT의 베이스 주소와 사이즈는 IDTR 레지스터에 저장되어 있다. SIDT 명령어와 LIDT 명령어로 IDTR 레지스터 값을 저장하고 수정할 수 있다. 명령어를 사용해서 IDT의 주소를 알아내고 후킹하기 원하는 디스크립터를 테이블에서 찾아서 수정을 하면 된다.

Hooking Processor MSRs	보통 커널 모드로 jump를 하기 위해서 SYSENTER 명령어를 사용하는데, 이 명령어는 3가지 MSR(Machine Specific Registers)를 사용한다. 3가지 레지스터 중에서 실행할 메모리 주소를 가지고 있는 IA32_SYSENTER_EIP 값을 후킹하면, SYSENTER가 호출되었을 때 원하는 명령어를 실행 할 수 있다.
Hooking SSDT	SSDT는 System Service Dispatch Table를 뜻하는데, SSDT를 후킹하면 Ring 0의 권한을 얻을 수 있고, 시스템 콜을 필터링 할 수 있다. 예를 들어, ZwQuerySystemInformation()을 후킹하면, 프로세스를 사용자로부터 숨길 수 있다.
IRP Handler hooking	IRP 후킹을 하는 방법은 2가지가 있다. 함수 포인터를 후킹하거나 attached device로 등록하는 것이다.
Hooking GDT	GDT에 있는 디스크립터들은 보통 코드나 데이터 메모리 세그먼트를 나타낸다. Call gate라는 것은 시스템 디스크립터 라고도 불리는데, 특별한 타입의 GDT 디스크립터이다. Call gate는 낮은 권한으로 실행되는 코드가 높은 권한의 루틴을 호출할 때 사용된다. 임의로 GDT에 call gate를 삽입시켜 원하는 루틴을 높은 권한으로 실행시킬 수 있다.
프로세스 탐지	접근하고 싶은 프로세스를 탐지하여 변형을 가한다. 예를 들어, 백신 프로그램을 탐지하여 악성코드를 치료하지 못하게 강제 종료를 한다.

(나) 시스템 변조

시스템 변조는 호스트 파일을 변조하여 악성코드를 실행, 특정 네트워크에 접속하여 백신 프로그램이 업데이트 하는 것을 방해하거나 access token을 방해함으로써 프로세스가 접근하는 것을 방해하여 다른 프로그램이 실행하는 것을 방해하는 것을 말한다. 다음은 시스템 변조를 통한 루트킷의 악성행위를 나타낸다.

[표 3-59] 시스템 변조 종류

시스템 변조 종류	설명
MBR	컴퓨터를 켜면, BIOS가 MBR을 로드하고, MBR은 VBR을, VBR이 운영체제를 실행시킨다. 루트 키트는 master boot record를 변조하여 코드를 삽입하여 운영체제 이미지를 바꿀 수 있다.
EPROCESS	EPROCESS는 프로세스를 내부적으로 표현하는 구조체이다. PsGetCurrentProcess() 함수를 통해서 접근할 수 있다. 구조체 안에는 UniqueProcessID, ActiveProcessLinks, Token, ImageFileName이 있다.
DRIVER_SECTION	DRIVER_SECTION 커널 모드 구조체는 시스템이 로드된 드라이버를 추적하는데 사용된다. 이 구조체는 KMD의 메모리 이미지를 표현한다.
TOKEN	윈도우에서는 각각 프로세스는 access token이 주어진다. 이 토큰은 authorization을 하는데 큰 역할을 한다. 그렇기 때문에 루트킷의 목표가 되기도 한다.
호스트 파일	시스템의 호스트 파일 엔트리를 변조하여 백신 프로그램이 업데이트 서버로부터 시그니처를 업데이트하는 것을 방해한다.

(다) Injection

인젝션은 DLL 파일 혹은 코드에 직접적으로 악성코드를 삽입하여 명령을 수행하거나 실행흐름을 바꿀 수 있다. 다음은 인젝션의 종류를 나타낸 표이다.

[표 3-60] Injection의 종류

Injection 종류	설명
DLL Injection	강제로 프로세스에 DLL을 로드시켜서 다른 프로세스 메모리 공간의 코드를 실행하게 하는 기술이다.
코드 Injection	컴퓨터 버그를 악용해서 프로그램에 코드를 삽입해서 실행 흐름을 바꾸는 기술이다.

다. 목표대상

[표 3-61] 목표대상 공격목표

목표대상 분류	설명
군사시설	악성코드의 공격대상이 군사관련 시설을 공격목표로 삼는 경우 이 분류에 속한다.
사회기반시설	악성코드의 공격대상이 사회기반시설을 공격목표로 삼는 경우 이 분류에 속한다.
행정기관	악성코드의 공격대상이 행정기관 관련 시설을 공격목표로 삼는 경우 이 분류에 속한다.
기업	악성코드의 공격대상이 기업을 공격목표로 삼는 경우 이 분류에 속한다.
개인	악성코드의 공격대상이 개인을 공격목표로 삼는 경우 이 분류에 속한다.

악성코드의 표적이 되는 시설에 따라 악성코드를 정의·분류하기 위한 분류이다. 목표대상은 새로운 매체가 발생하더라도 변화가 어렵기 때문에 비교적 확장성에 대한 고려가 불필요한 요소이다.

악성코드의 공격 목표대상이 되는 경우는 위 [표 3-61]과 같은 다섯 가지가 존재한다.

군 관련 홈페이지, 군사 관련 시설 등을 공격목표로 하는 경우 [군사시설] 분류에 속한다. 이는 공격목표 중 위험 수준이 가장 높은 분류로 [공격행위]의 위험수준에 대해 높은 가중치를 갖는다. [군사시설]에 대한 공격이 성공적으로 수행될 경우 그 어떤 표적에 대한 공격보다 큰 규모의 피해가 발생할 수 있다. 실제 악성코드가 공격목표로 군사 관련 시설을 삼은 것은 아니지만 과거 미공군 네트워크에 악성코드가 전파된 경험 있으며, 최근에는 미국 공군 무인폭격기 드론(drone)이 컴퓨터 바이러스에 감염되어 운항 정보 등 중요한 군사 기밀의 유출 가능성이 생기면서 사회적인 문제가 되기도 하였다[38]. 최근 많은 무기들이 컴퓨터를 통해 제어되면서 군사 관련 시설 등에 이를 제어할 목적으로 악성코드가 침투, 감염이 이루어질 경우 제어체계 마비에 의한 오작동부터 시작하여 심각한 경우 악의적인 사용자가 무기 제어권을 가질 경우 심각한 많은 인명피해나 국가 간 긴장 상태를 유발할 수 있는 위험성이 존재한다. 따라서 본 연관성 정보 구조에서는 [군사시설]이 목표대상이 되는 경우 가장 높은 가중치를 적용한다.

원자력발전소, AMI 네트워크, 도로교통제어, 항공교통제어 등 사회기반시설을 공격목표로 하는 경우 [사회기반시설] 분류에 속한다. 이는 공격목표 중 위험 수준이 두 번째로 높은 분류이다. [사회기반시설]에 대한 공격이 성공적으로 수행될 경우 사회기반시설이 마비되어 혼란이 발생하거나 심각한 경우 원자력발전소 등의 오동작으로 큰 재해가 발생할 위험이 있다. [사회기반시설]을 공격목표로 하는 대표적인 사례로 스텝스넷(stuxnet)이 있다. 스텝스넷은 2009년 전파되어 물리적으로 격리되어 있던 원자력발전소 제어를 위한 시스템에 접근한 후 오작동을 유발한 악성

코드로 악성코드에 대한 사회적으로 긴장감을 고조 시켰던 대표적인 사례이다. 특히 관제 시스템에서 제공하는 정보에 매우 의존적인 비행기의 항공 교통 관제 시스템 등이 악성코드의 공격을 받아 오작동할 경우 항공 사고의 발생으로 인해 큰 인명피해가 발생할 우려가 존재한다.

국내 행정을 담당하는 각종 부처의 홈페이지나 각종 부처의 내부 네트워크를 공격목표로 하는 경우 [행정기관] 분류에 속한다. 행정기관이 공격목표가 될 경우 전산화된 행정 시스템이 마비되어 행정 처리가 지연되거나 민감한 대량의 개인정보나 기밀자료들이 유출될 위협이 존재하여 비교적 높은 위험 수준을 갖는 분류이다. 2009년 7월 7일 발생한 대규모 분산 서비스방해(DDoS) 공격이 이루어지면서 국내 행정기관전산망에서 사용하는 '아래아 한글 확장자(.kwp)'를 파괴하는 등 정치적 목적을 가진 공격이 감행된 사건이 있다[39].

기업의 서비스 네트워크나 기업 내부 네트워크를 공격목표로 하는 경우 [기업] 분류에 속한다. 이 분류가 공격 목표가 될 경우 주로 대규모 경제적 피해가 발생한다. 기업의 서비스 네트워크가 마비될 경우 마비된 시간에 비례하여 경제적 손실이 발생할 수 있으며, 특히 생산라인을 제어하는 기업의 제어시스템이 마비 될 경우 천문학적인 단위의 경제 피해가 발생할 위협이 존재한다.

악성코드의 공격 목표가 정해지지 않거나 불특정 다수의 개인 사용자 정보 수집을 위한 악성코드일 경우 [개인]분류에 속한다. 본 악성코드들은 개인 사용자를 타깃으로 금전적 이익을 취하는 악성코드들이 속한다. [개인]을 목표로 하는 악성코드지만, 제 3의 공격을 위해 좀비PC로 만들기 위한 악성코드인 경우 공격 대상이 하드코딩되어 악성코드 내부에 나타나 있다면 해당 악성코드는 타깃의 [목표대상]이 된다. 하지만 공격자로부터 실시간 명령을 받는 악성코드의 경우 [잠재위험]의 가중치가 적용되어 해당 악성코드의 위협 정도를 구분한다. [개인]을 목표대상으로 하는 악성코드에 대해서는 가장 낮은 위험 수준을 갖는 것으로 판단하며 본 연관성 정보 구조에서는 가장 낮은 가중치를 적용한다.

라. 잠재위험

[잠재위험]은 해당 악성코드가 수행하는 악성행위 자체로는 시스템에 큰 영향을 미치지 않지만 향후 공격자가 직접 시스템으로 침투하도록 할 수 있거나 특정 서버에 접속하여 공격자가 내리는 명령에 따라서 다양한 악성코드를 수행할 수 있을 경우 악성코드가 수행한 수 있는 잠재적인 악성행위가 있다고 판단하여 정의된 가중치 항목이다. [잠재위험]은 [전파경로], [자가보호], [목표대상]처럼 연관성 정보 구조 중 [감염경로], [실행주체], [공격대상], [공격행위]와는 별도의 구조를 갖지 않고 [공격행위] 항목에 특정 악성행위를 나타낸다. 다음은 해당하는 [공격행위] 중 [잠재위험]을 나타내는 악성행위를 나타내는 표이다.

[표 3-62] 잠재위험 위험지수 가중치

공격행위 하위 단계	악성행위(레벨 3)
네트워크	서버접속
시스템	강제 시스템 제어
	네트워크 설정 변경
	시스템 설정 변경

(1) 서버접속

서버 접속은 해당하는 악성코드가 특정 서버에 접속하는 것을 나타내는 악성행위이다. 악성코드가 특정 서버에 접속할 경우 다른 악성코드를 다운로드하거나 공격자의 커맨드 수신과 같은 악성행위를 수행할 수 있다. 이는 해당하는 악성코드가 다른 악성코드를 다운로드하는 단순 모듈형 악성코드나 외부명령을 수신하여 다른 시스템을 공격하는 악성코드를 뜻한다. 모듈형 악성코드의 경우 악성코드의 목적이 다른 악성행위를 수행하는 악성코드 다운로드이기 때문에 해당하는 악성코드가 수행하는 악

성행위는 여타 일반 프로그램과는 차이가 없을 수 있다. 따라서 [서버접속]은 해당 악성코드가 다른 악성코드보다 수행하는 악성행위가 위협적이지 않더라도 다른 악성코드들과 구분 짓기 위한 요소이다.

(2) 강제 시스템 제어

[강제 시스템 제어]는 시스템을 강제로 종료 및 사용자 권한을 변경할 수 있다. 해당하는 악성코드가 특정 사용자의 권한을 상승시킬 경우 해당하는 사용자를 통하여 공격자는 시스템의 다른 설정을 변경하거나 위험도 높은 악성행위를 수행할 수 있게 되므로 [강제 시스템 제어]는 다른 악성코드와 위험정도를 구분하기 위한 요소가 된다.

(3) 네트워크 설정 변경

[네트워크 설정 변경]은 백도어의 기능을 수행하는 악성코드를 구분하기 위한 요소로써 외부에서 공격자가 악성코드에 감염된 시스템에 침투하기 위하여 특정 포트를 개방하여 접근을 용이토록 하거나 DNS등을 변경하여 피싱과 함께 악용될 수 있어 [네트워크 설정 변경]은 여타 다른 악성행위보다 위험도가 높다고 할 수 있다.

(4) 시스템 설정 변경

[시스템 설정 변경]은 [강제 시스템 제어]와는 달리 특정 사용자를 추가하거나 관리자의 비밀번호를 변경시키는 악성행위를 뜻한다. [시스템 설정 변경]과 [강제 시스템 제어]를 혼용하여 악성코드는 해당하는 시스템의 모든 권한을 획득할 수 있다.

제 3 절 연관성 정보 구조 확장성

현재 연관성 정보 구조는 확장성을 고려하여 설계되었으므로 구조에 있어서 앞으로 큰 변화는 없을 것으로 예상된다. 특히 연관성 정보 구조의 1단계, 2단계에 있어서는 현재 모든 요소를 고려하여 같은 단계의 분류가 서로 배타성을 가지도록 설계하여 큰 변동이 없을 것으로 보인다. 그러나 연관성 정보 구조의 3단계의 일부분의 경우는 2단계의 각 분류에 대한 구체적인 요소로써 완전히 새로운 매체가 등장할 경우 확장이 필요할 수 있다. 특히 전파경로 및 감염경로에서 매체분류의 요소는 새로운 매체가 발생할 경우 새로운 매체 자체가 경로가 되는 경우가 있어서 확장될 가능성이 다른 분류보다 높다. 이런 경우를 위해 연관성 정보 구조의 확장성을 고려할 필요가 있다.

본 절에서는 신종 악성코드 또는 기존 악성코드 변종이 발생하는 경우 연관성 정보 구조의 확장성 검토를 위한 과정과 새로운 매체가 발생할 경우 연관성 정보 구조 확장을 위한 매체 분석 및 적용 과정에 대해 설명한다.

1. 신종·변종 악성코드에 대한 확장성 검토 및 적용

신종 악성코드 및 변종 악성코드가 발생하는 경우, 악성코드 분석과 동시에 연관성 정보 구조가 확장성이 필요한지 검토 할 필요가 있다. 앞서 말했듯이 현재 연관성 정보 구조는 확장성을 고려하여 설계되었으므로 1단계, 2단계에서는 확장성이 크게 요구되지 않을 것이다. 그러므로 신·변종 악성코드 분석 시 확장성이 필요한 경우에는 3단계를 확장하는 것을 원칙으로 하며 부득이하게 2단계에서의 확장이 필요한 경우에는 여러 번의 토의를 거쳐 확장 되어야 할 것이다.

다음 슈도코드(pseudo code)는 신·변종 악성코드를 분석하고 연관성 정보 구조에 적용하기 위한 과정을 보여준다. 가독성을 위해 전체 과정

을 간결한 슈도코드(pseudo code)로 표현하였다.

[표 3-63] 신·변종 악성코드 확장성 검토 및 적용

신·변종 악성코드 확장성 검토 및 적용	
<pre> void extend(malware m) { //설치 수단 if(check_install_level2(m)) { if(!check_install_level3(m)) { add_level3(analyze(m)); } } else { add_level2(analyze(m)); } //공격 대상 if(check_victim_level2(m)) { if(!check_victim_level3(m)) { add_level3(analyze(m)); } } else { add_level2(analyze(m)); } //실행 주체 if(check_execution_level2(m)) { if(!check_execution_level3(m)) { add_level3(analyze(m)); } } else { add_level2(analyze(m)); } } </pre>	<pre> //공격 행위 if(check_attack_level2(m)) { if(!check_attack_level3(m)) { add_level3(analyze(m)); } } else { add_level2(analyze(m)); } //감염 경로 if(!check_infection(m)) add_infection(m); //자가 보호 if(!check_protection(m)) add_protection(m); } </pre>

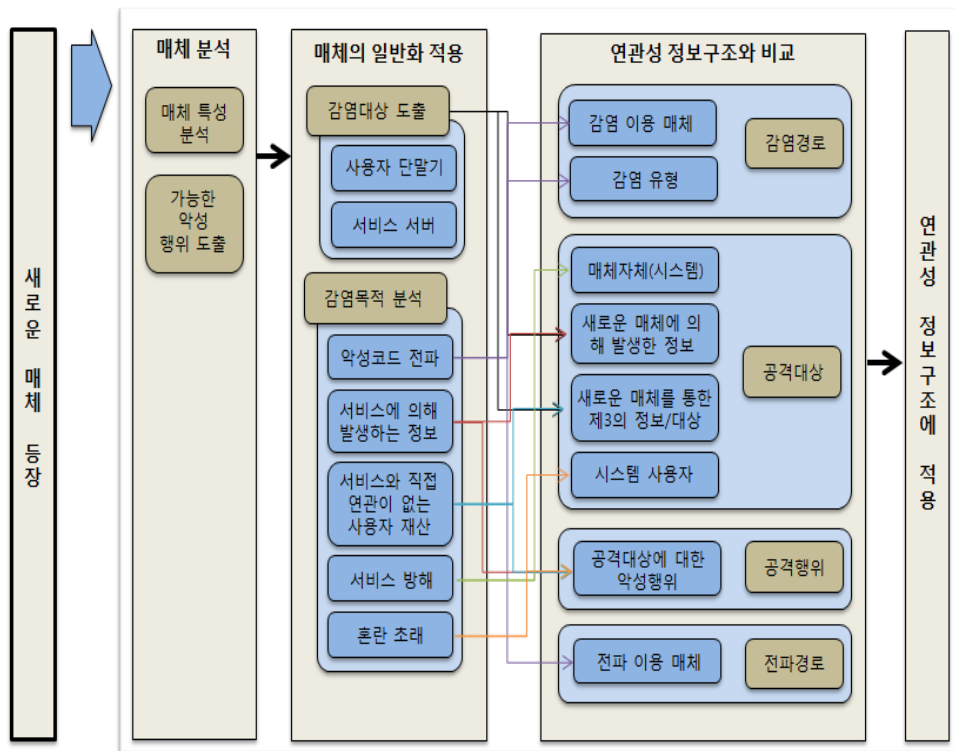
연관성 정보 구조의 1단계는 모든 악성코드에 해당하는 분류이므로 슈도코드에서 제외하였다. 검토과정은 2단계에서 3단계로 검토를 하며 확장성이 필요한 경우 3단계에서 값을 추가하도록 한다. 신·변종 악성코드를 분석하는 과정에서 새로운 매체가 등장하거나 새로운 매체로 인해 확장성이 필요한 경우에는 다음에서 설명하는 새로운 매체 분석 및 적용 과정을 통해 확장하도록 한다.

2. 새로운 매체 분석 및 적용

새로운 매체가 발생하는 경우에 분석가는 해당 매체를 분석하고 분석 결과에 따른 연관성 정보 구조 확장 여부를 판단한다. 본 절에서는 새로운 매체가 발생하는 경우에 매체의 일반화를 이용하여 연관성 정보 구조를 확장하는 방법을 제시한다. 매체의 일반화는 최근에 발생한 여러 매체의 특성과 매체에서 가능한 보안 위협 분석 등의 과정을 통해 이루어졌다. 또한 제시하는 방법을 적용한 사례를 추가하였다.

가. 새로운 매체 발생할 경우 분석 및 적용 과정

분석가는 도출된 매체의 일반화를 이용하여 새로운 매체가 발생할 경우 아래의 그림과 같은 과정으로 확장성을 적용할 수 있다.



(그림 3-2) 새로운 매체 발생할 경우 분석 및 적용 과정

구체적인 분석 및 적용과정은 다음과 같다. 분석가는 새로운 매체가 발생하면 매체를 분석하는 단계부터 시작하게 된다. 매체를 분석하는 단계에서 새로운 매체의 특성을 분석하고 매체를 이용하여 수행할 수 있는 모든 악성행위를 도출한다.

그런 다음에 매체의 일반화 단계에 새로운 매체를 일반화에 적용하기 위해 감염대상 및 감염목적(악성코드 전파, 서비스에 의해 발생하는 정보, 서비스와 직접 연관이 없는 사용자 재산, 서비스 방해, 혼란초래)에서 도출된 요소를 연관성 정보 구조에서 각각 감염경로, 공격대상, 공격행위, 전파 경로와 비교하여 확장 여부를 결정하고 연관성 정보 구조에 적용한다. 연관성 정보 구조에 새로운 요소를 추가할 경우에 분석가가 판단하여 추가되는 요소에 위험지수를 매기도록 한다.

나. 매체의 일반화

새로운 매체가 발생할 경우 연관성 정보 구조 확장성 정보를 도출하기 위해 매체의 일반화를 하였다. 매체의 일반화는 최근에 발생한 여러 매체의 특성을 분석하고 해당 매체를 이용하여 가능하게 되는 모든 보안 위협 요소를 도출하는 등의 과정을 통해 이루어졌다. 일반화는 감염 대상과 감염 목적으로 구성된다.

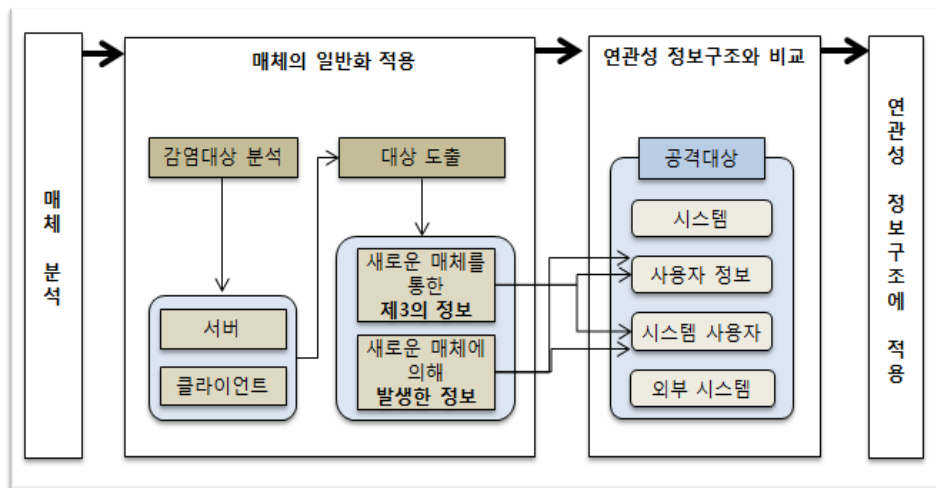
(1) 감염 대상

새로운 매체가 발생할 경우 새로운 매체를 통한 악성코드 감염 대상은 클라이언트[PC, 스마트폰, 차량 장치 등]와 서버[클라우드 컴퓨팅 서버, 스마트 그리드 망, 기반시설 제어망 등]로 나뉘며 이 외에 새로운 감염 대상은 고려하기 어렵다.

[표 3-64] 감염 대상 및 예시

감염대상	예시
클라이언트	PC, 스마트폰, 차량 장치 등
서버	클라우드 컴퓨팅 서버, 스마트 그리드 망, 기반시설 제어망 등

감염 대상을 도출하면 새로운 매체에 의해 발생한 정보와 새로운 매체를 통한 제 3의 정보를 분석할 수 있다. 즉 공격대상이 되는 정보를 예측할 수 있다. 분석된 정보는 연관성 정보 구조에서 연관성 정보 구조 1 단계인 공격대상에 적용된다.



(그림 3-3) 감염대상의 연관성 정보 구조 적용과정

(2) 감염 목적

새로운 매체가 발생할 경우 새로운 매체를 통한 악성코드 감염 목적은 악성코드 전파[SNS], 서비스에 의해 발생하는 정보 탈취[클라우드 컴퓨팅], 서비스와 직접연관이 없는 사용자 재산[결재 유도], 서비스 방해[기반시설 제어망 마비 - stuxnet], 혼란 초래[허위정보 유포]로 나눌 수 있으며, 이 외에는 새로운 감염 목적을 고려하기 어렵다.

[표 3-65] 감염 목적 및 예시

감염대상	예시
악성코드 전파	SNS 등
서비스에 의해 발생하는 정보	클라우드 컴퓨팅 등
서비스와 직접연관이 없는 사용자 재산	결재 유도 등
서비스 방해	기반시설 제어망마비(stuxnet) 등
혼란 초래	허위정보 유포(VANET) 등

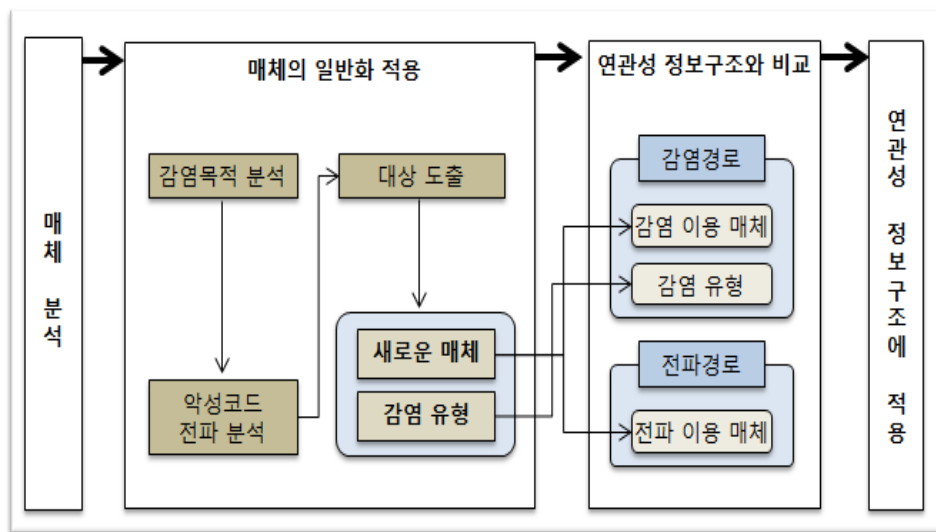
(가) 악성코드 전파

새로운 매체는 악성코드 전파 경로로 이용되기 쉽다. 대표적인 예로 SNS가 있다. 악성코드 전파 경로로 이용되는 경우 파일 직접 공유, URL 전달, 취약점을 이용한 파일 복사 및 제어, 권한상승에 따른 파일 복사 및 제어 등으로 나뉘며 이 외에는 새로운 악성코드 전파 방안을 고려하기 어렵다.

[표 3-66] 악성코드 전파의 일반화

악성코드 전파의 일반화
○ [새로운 매체]를 이용하여 [감염 유형]으로 악성코드를 전파

감염 목적에서 악성코드 전파는 위의 표와 같은 문장으로 일반화 할 수 있으며 연관성 정보 구조 1단계인 감염경로와 전파경로, 구체적으로 감염·전파의 이용 매체와 유형에 적용한다.



(그림 3-4) 악성코드 전파의 연관성 정보 구조 적용과정

(나) 서비스에 의해 발생하는 정보

새로운 매체의 생성에 의해 수많은 부가적인 정보들이 생산된다. 예를 들어 금융서비스가 컴퓨터를 통해 이루어지면서 금융과 관계된 정보들이 공격대상이 되기도 하며, 클라우드 컴퓨팅과 같이 다양한 서비스들이 제공되면서 여기에 저장된 정보들이 공격대상이 되기도 한다. 이처럼 새로운 매체가 발생할 경우 해당 매체로 인해 생성되는 정보들이 공격 대상이 된다. 아래에서 감염 대상에 정보가 보관되어 있는 경우인 정보 보관과 감염 대상이 정보에 접근하는 경로가 되는 경우인 접근 경로로 나누어 보도록 한다.

○ 정보 보관

새로운 매체를 통해 감염 대상에 저장된 정보들을 공격 대상으로 삼아 공격하는 행위는 다음과 같이 일반화 할 수 있다.

[표 3-67] 서비스에 의해 발생하는 정보에서 정보보관의 일반화

서비스에 의해 발생하는 정보에서 정보보관의 일반화
○ 사용자의 단말기[감염 대상 1]에 저장된 [공격대상]을 [공격행위]함
○ 사용자의 단말기[감염 대상 1]에 의해 생성된 [공격대상]을 [공격행위]함
○ 서비스 서버[감염 대상 2]에 저장된 [공격대상]을 [공격행위]함
○ 통신매체 및 프로토콜의 취약점에 의한 [공격대상]을 [공격행위]함

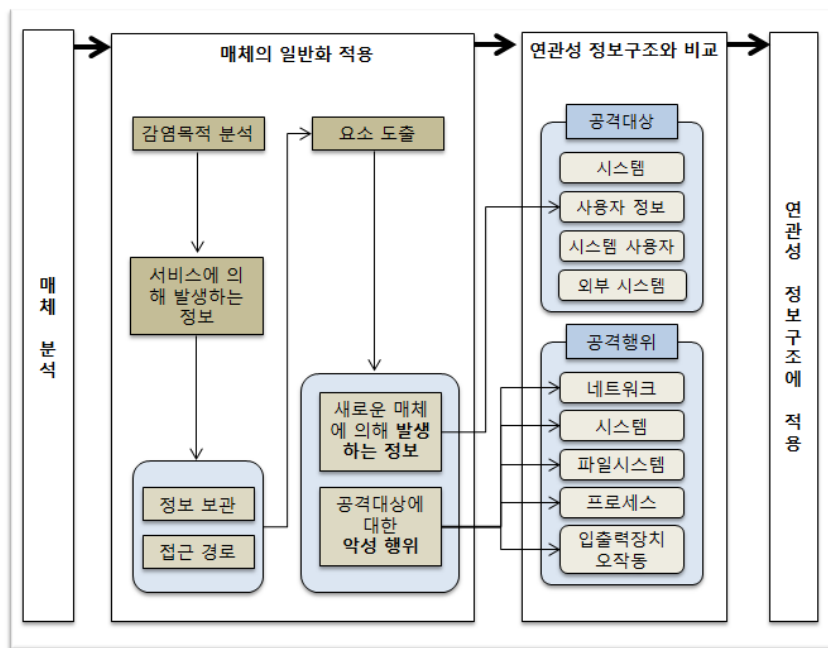
○ 접근 경로

감염 대상을 통해 새로운 매체에 접근하여 저장된 정보를 공격대상으로 삼는 경우는 다음과 같다.

[표 3-68] 서비스에 의해 발생하는 정보에서 접근경로의 일반화

서비스에 의해 발생하는 정보에서 접근경로의 일반화
○ 사용자의 단말기[감염 대상 1]를 통해 서비스 서버 접근, 그 결과 또 다른 사용자의 [공격대상]을 [공격행위]함
○ 사용자의 단말기[감염 대상 1]를 통해 서비스 서버 접근, 그 결과 서비스 서버[감염 대상 2] 자체의 [공격대상]을 [공격행위]함

감염 목적의 서비스에 의해 발생하는 정보에서 정보 보관과 접근 경로는 위의 표들과 같은 문장으로 일반화 할 수 있다. 여기서 얻을 수 있는 요소 중 공격 대상은 연관성 정보 구조 1단계에서 공격대상에 적용된다. 공격 대상에 대해 예측되는 공격 행위 또한 연관성 정보 구조 1단계인 공격행위에 적용될 것이다.



(그림 3-5) 서비스에 의해 발생하는 정보의 연관성 정보 구조 적용과정

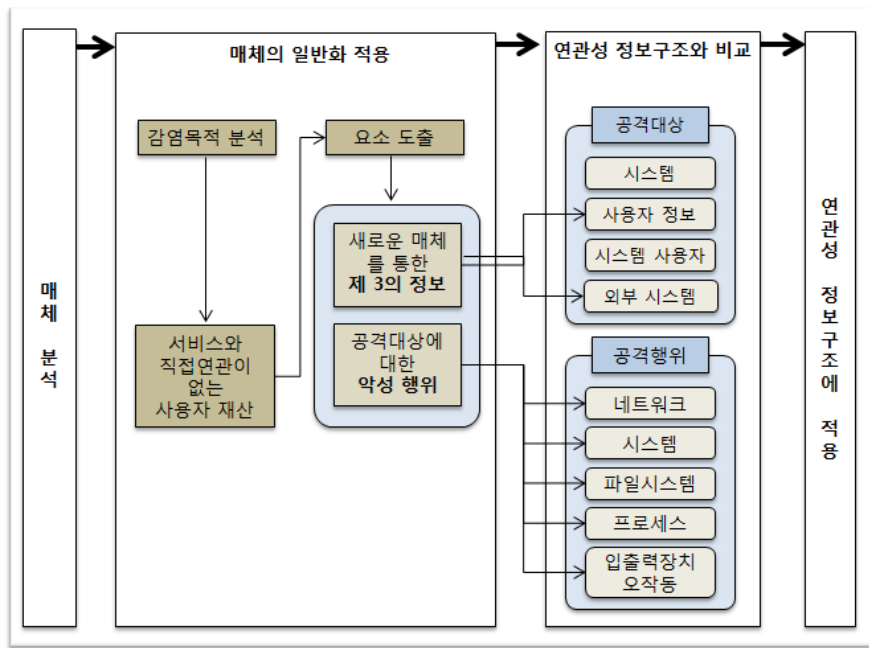
(다) 서비스와 직접연관이 없는 사용자 재산

새로운 매체를 통해 사용자의 단말기[감염 대상 1]에 접근이 용이한 경우 사용자의 단말기[감염 대상 1]에 존재하는 매체와 무관한 재원이 공격받을 위험이 있다. 사용자의 단말기[감염 대상 1]에 연결되는 휴대용 장치나, 다른 컴퓨팅 장치들이 악성코드에 의해 공격받거나, 사용자의 신용카드, 금융정보 유출로 인한 경제적 손실이 일어날 수 있다. 또한 사용자의 단말기[감염 대상 1]로 이루어지는 사회관계에 피해를 주기도 하며, 새로운 매체와 무관한 다른 정보들의 유출이 가능하다.

[표 3-69] 서비스와 직접 연관이 없는 사용자 재산의 일반화

서비스와 직접 연관이 없는 사용자 재산의 일반화
○ [감염 대상 1]에 새로운 매체와 무관한 [공격대상]을 [공격행위]함

감염 목적에서 서비스와 직접 연관이 없는 사용자 재산을 목적으로 하는 경우 위의 표와 같은 문장으로 일반화 할 수 있다. 새로운 매체를 통한 서비스와 무관한 정보들을 도출하여 연관성 정보 구조 1단계에 공격 대상에 추가할 수 있으며 해당 대상에 대한 악성 행위를 도출함으로 연관성 정보 구조 1단계에 공격행위를 확장 할 수 있다.



(그림 3-6) 서비스와 직접연관이 없는 사용자 재산의
연관성 정보 구조 적용과정

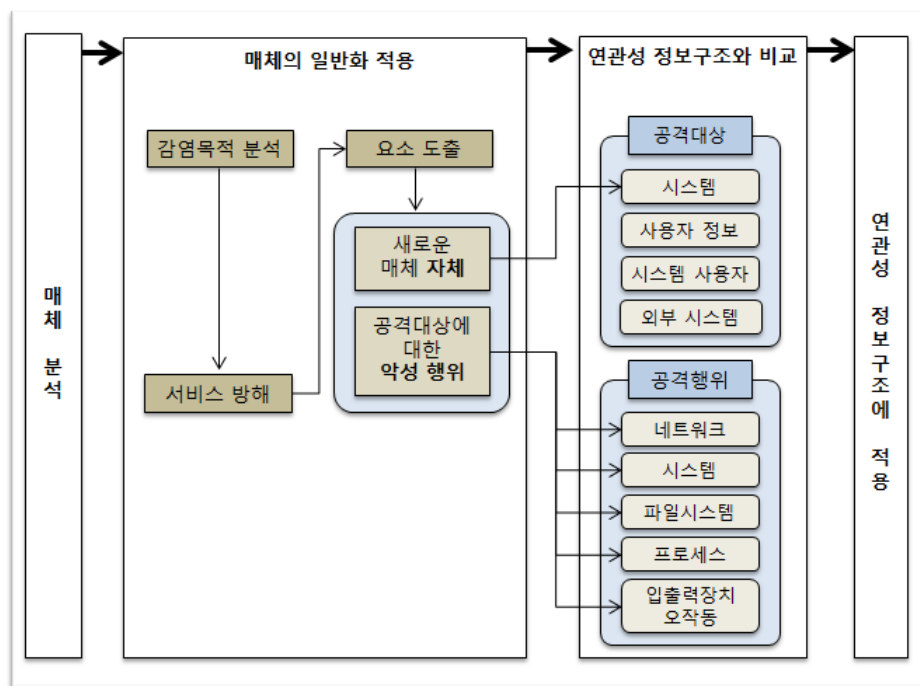
(라) 서비스 방해

새로운 매체가 직접적인 공격 대상이 됨에 따라 새로운 매체를 통한 서비스 자체가 위협받기도 한다. 컴퓨팅 장치를 이용한 산업 기반시설 제어가 가능해지면서 제어를 담당한 매체를 공격 마비시킴으로써 산업 기반시설 자체를 공격한 사례가 2010년에 있었다. 이와 같이 새로운 매체가 발생할 경우 트래픽 생성을 통해 통신을 방해하거나 시스템 자체를 공격 또는 시스템에 설치된 취약점을 이용하여 서비스 자체를 무력화 시킬 위험이 존재한다.

[표 3-70] 서비스 방해의 일반화

서비스 방해의 일반화
○ [새로운 매체]를 [공격행위]함

감염 목적에서 서비스 방해는 위의 표와 같은 문장으로 일반화 할 수 있다. 새로운 매체는 연관성 정보 구조 1단계 공격대상에 적용되며 새로운 매체에 대한 공격 행위를 도출하여 연관성 정보 구조 1단계에서 공격 행위를 확장한다.



(그림 3-7) 서비스 방해의 연관성 정보 구조 적용과정

(마) 혼란 초래

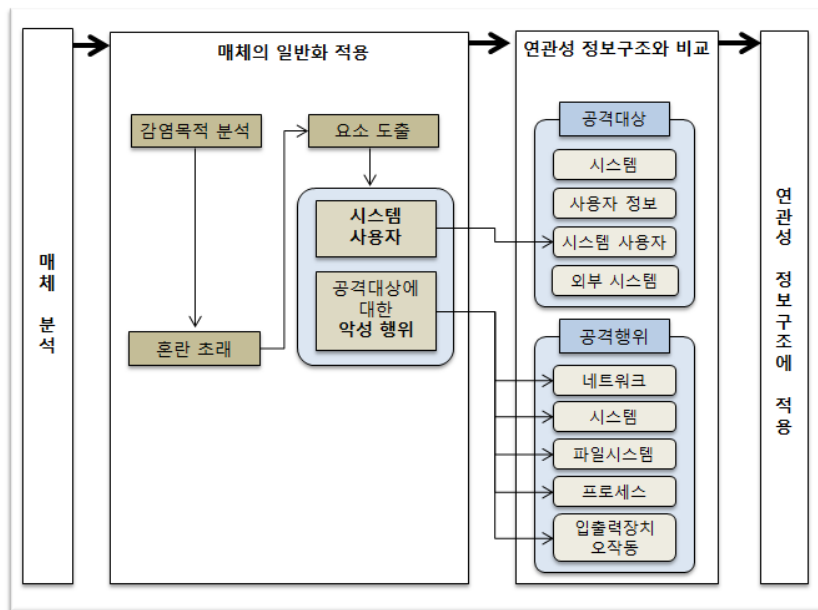
새로운 매체를 통해서 [감염 대상1]의 사용자에게 혼란을 유발하는 경우도 있다. 유비쿼터스 컴퓨팅 기술과 융합기술이 고도화되면서 개인에

생활에 밀접하게 영향을 줄 수 있는 새로운 매체가 등장하고 있다. 이러한 매체를 이용하여 서비스를 이용하는 사용자들에게 허위 정보를 퍼트리거나 서비스 이용에 방해 등의 혼란을 초래할 수 있다. 경우에 따라 혼란 초래는 단순한 혼란으로 끝나지 않고 시스템 사용자에게 직접적인 피해를 줄 가능성도 존재한다.

[표 3-71] 혼란 초래의 일반화

혼란 초래의 일반화
○ [공격대상]을 [공격행위]함

감염 목적에서 혼란 초래는 위의 표와 같은 문장으로 일반화 할 수 있다. 혼란 초래를 분석한 결과는 연관성 정보 구조 1단계 공격대상에 적용되며 새로운 매체에 대한 공격 행위를 도출하여 연관성 정보 구조 1단계에서 공격행위를 확장한다.



(그림 3-8) 혼란 초래의 연관성 정보 구조 적용과정

나. 예시

연관성 정보 구조 확장성에 대해 예시를 보여주기 위해 최근에 새로 등장한 몇 가지 매체들을 선정하여 각 매체들의 특징과 매체를 이용하여 가능한 악성행위를 분석하였다. 아래에서 분석한 매체는 전자투표, U-health, VANET이다.

(1) 전자투표

전자투표는 유권자 등록, 개표, 검표 등의 선거 과정에서 전체 혹은 일부를 전자화하는 것을 말한다.

(가) 특성

전자투표는 인터넷을 통해서 하는 원격투표방식(Remote Voting)과 투표소를 전자화한 투표소투표(Pool Site Voting)방식이 존재한다. 전자투표의 해외 사례를 보면 2000년부터 15개 주에서 사용하고 있고, 현재 해외 35개국에서 전자투표를 사용하고 있다. 우리나라는 아직 적극적으로 반영되고 있지 않지만 전자선거 및 전자투표 사업 등을 추진하기 위한 ‘전자선거추진협의회’가 구성되는 등 전자 선거를 제공하기 위한 사업이 추진 중에 있다.

전자투표는 안전성이 보장되고 편리하고 신속한 투표를 목표로 하며 몇 가지 특성을 만족해야 한다. 완결성(Completeness), 편리성(Convenience), 익명성(Anonymity), 효율성(Efficiency), 부정행위 불가능성(Non-cheating), 강건성(Robustness), 검증성(Verifiability), 공정성(Fairness), 선거의 일반성(General election), 반강제성(Uncoercibility), 이동성(Mobility)이 특성에 해당한다. 전자투표는 전자투표의 내용은 완벽하게 유지 되어야 하며, 투표의 편리성 제공하고, 인증기관을 포함한 어

는 누구도 투표내용과 투표를 한 유권자를 연결시킬 수 없어야 하며, 투표를 위한 모든 과정이 효율적으로 수행해야 한다. 또한 부정한 행위가 방지되도록 설계하며 전자투표 시스템에 대한 제 3자의 공격에 방어할 수 있어야 하며 유권자의 투표결과를 정확히 알 수 있게 설계하고, 투표 집계가 끝나고 결과 발표 전에는 어느 누구도 중간 투표 결과 값을 알 수 없어야 하며, 다양한 선거 방식에 적용 가능해야 하며, 투표는 매매할 수 없고, 어디서나 투표가 가능해야 한다.

(나) 가능한 악성행위

전자투표 서비스에서 악성코드가 수행할 수 있는 악성행위는 다음 표와 같다.

[표 3-72] 전자투표에서 가능한 악성행위

가능한 악성행위
○ 유권자의 투표 값 혹은 전체 투표 결과 값 등을 정보 탈취/유출 (투표비밀침해)
○ 온라인 투표 시, 유권자의 투표용지를 중간에서 가로채기, 전송지연, 수신 경로를 임의로 공격자에게로 수정
○ 전자투표시행 당일, 중앙관리시설 혹은 전자투표시스템을 DDoS공격
○ 투표결과 값 등을 투표소에 시스템 조작을 통한 정보 변조

(다) 매체의 일반화 적용

위에서 도출한 악성행위를 기반으로 매체의 일반화에 적용하여 연관성 정보 구조에 매칭하기 위한 요소를 도출한다. 다음 표가 매체의 일반화를 이용하여 요소들을 도출한 것이다. 이 과정이 끝나면 연관성 정보 구

조와 비교하여 연관성 정보 구조에 실제로 적용할지 여부를 결정한다.

[표 3-73] 전자투표에서 매체의 일반화 적용

가능한 악성행위	요소
(서비스에 의해 발생하는 정보) ○ [유권자의 투표 값] 혹은 [전체 투표 결과 값] 등을 [정보 탈취/유출]	[공격대상] 유권자의 투표 전체 투표 결과 [공격행위] 정보 탈취/유출
(서비스에 의해 발생하는 정보) ○ 온라인 투표 시, [유권자의 투표용지]를 중간에서 [가로채기], [전송지연], 임의로 공격자에게로 [수신 경로를 수정]	[공격대상] 유권자의 투표용지 [공격행위] 가로채기 전송지연 수신경로 수정
(서비스 방해) ○ 전자투표시행 당일, [중앙관리시설] 혹은 [전자투표시스템]을 [DDoS공격]	[공격대상] 중앙관리시설 전자투표시스템 [공격행위] DDoS공격
(서비스에 의해 발생하는 정보) ○ [투표결과 값] 등을 투표소에 [시스템 조작]을 통한 [정보 변조]	[공격대상] 투표결과 값 [공격행위] 시스템 조작 정보변조

(2) U-health

U-health는 유무선 네트워킹 기술을 기반으로 언제 어디서나 사용자들

이 이용 가능한 건강관리 및 의료 서비스를 말하며 새로운 의료패러다임 시대를 도래시켰다.

(가) 특성

U-health가 실현되면 보건소, 진료소와 같은 공공보건의료기관과 국립대병원 등 공공의료기관이 직접 연결되어 환자의 의료정보를 공유하게 되며, 구급차와 응급의료센터 및 의료기관과의 통신망이 연동되어 응급상황 시에 신속한 조치가 가능해 질 것이다. 환자들은 원격으로 의료 서비스를 받을 수 있게 될 것이며, u-모바일, RFID, PAN, BAN 등의 활용으로 정확한 생체정보 측정이 가능해 질 것이다[41][42].

(나) 가능한 악성행위

U-health 서비스를 통해 악성코드의 수행 가능한 악성행위는 다음 표와 같다.

[표 3-74] U-health에서 가능한 악성행위

가능한 악성행위
<ul style="list-style-type: none"> ○ DB화 되어있는 환자의 의료정보를 유출 (개인프라이버시침해) ○ 치료용 나노로봇 등의 의료시스템을 오작동하게 하여 환자의 생명을 위협 ○ 수집되는 생체정보를 위조하여 허위로 위급 상황 연출 가능 ○ BAN을 통해 인체 내 장치(인공심박조율기 등)를 구동시킬 시, 오작동하게 하여 환자의 생명을 위협

(다) 매체의 일반화 적용

위에서 도출한 악성행위를 기반으로 매체의 일반화에 적용하여 연관성 정보 구조에 매칭하기 위한 요소를 도출한다. 다음 표가 매체의 일반화를 이용하여 요소들을 도출한 것이다. 이 과정이 끝나면 연관성 정보 구조와 비교하여 연관성 정보 구조에 실제로 적용할지 여부를 결정한다.

[표 3-75] U-health에서 매체의 일반화 적용

가능한 악성행위	요소
(서비스에 의해 발생하는 정보) ○ DB화 되어있는 환자의 [의료정보]를 [유출]	[공격대상] 의료정보 [공격행위] 유출
(서비스 방해) ○ 치료용 나노로봇 등의 [의료시스템]을 [오작동]하게 하여 환자의 생명을 위협	[공격대상] 의료시스템 [공격행위] 시스템 오작동
(서비스에 의해 발생하는 정보) ○ 수집되는 [생체정보]를 [위조]하여 허위로 위급 상황 연출 가능	[공격대상] 생체정보 [공격행위] 위조
(서비스 방해) ○ [BAN]을 통해 인체 내 [장치](인공심박조율기 등)를 구동시킬 시, [오작동]하게 하여 환자의 생명을 위협	[매체] BAN [공격대상] 장치 [공격행위] 오작동

(3) VANET/지능형자동차

지능형자동차는 기계, 전자, 통신, 제어 등 여러 기술을 융합 하여 차

량의 안정성과 편의성을 획기적으로 향상한 자동차를 말한다.

(가) 특성

이러한 지능형차량이 안정성과 편의성을 제공하기 위해 사용하는 기반 네트워크가 VANET이다. VANET은 Vehicular Ad Hoc NETwork의 약자로 차량의 통신을 의미한다.

지능형자동차는 자동차에 마이크로 컴퓨터와 센서 등을 장착하며 차량 내부 통신, 주변장치와 통신 혹은 차량 간의 통신을 통해서 작업을 수행한다.

(나) 가능한 악성행위

VANET을 통해서 악성코드가 수행할 수 있는 악성 행위는 다음 표와 같다[43][44][45].

[표 3-76] VANET을 이용하여 가능한 악성행위

가능한 악성행위
○ 일정 네트워크 영역 내에서 차량의 통신에 장애가 발생되도록 DoS 공격
○ 교통사고, 고속도로 정체, 차량통제 등에 관한 정보를 다른 차량에게 허위로 유포하여 혼란초래
○ 시간, 자동차의 위치, 차량의 ID, 차량이동정보 등의 정보 탈취/유출(개인프라이버시침해)
○ 속도, 위치, 차량전장부분의 상태, 각종센싱정보 등의 차량 내부 정보에 대한 위변조 공격
○ 각종 내부 센서의 오작동 유발
○ RSU와의 통신 차단 (서비스 방해)

(다) 매체의 일반화 적용

위에서 도출한 악성행위를 기반으로 매체의 일반화에 적용하여 연관성 정보 구조에 매칭하기 위한 요소를 도출한다. 다음 표가 매체의 일반화를 이용하여 요소들을 도출한 것이다. 이 과정이 끝나면 연관성 정보 구조와 비교하여 연관성 정보 구조에 실제로 적용할지 여부를 결정한다.

[표 3-77] VANET에서 매체의 일반화 적용

가능한 악성행위	요소
(서비스 방해) ○ 일정 네트워크 영역 내에서 [차량의 통신]에 장애가 발생되도록 [DoS 공격]	[공격대상] 차량 통신(VANET) [공격행위] DoS공격
(혼란 초래) ○ 교통사고, 고속도로 정체, 차량통제 등에 관한 정보를 다른 차량에게 [허위로 유포]하여 혼란초래	[공격행위] 허위 유포
(서비스에 의해 발생하는 정보) ○ [시간, 자동차의 위치, 차량의 ID, 차량이동정보] 등의 [정보 탈취/유출]	[공격대상] 시간정보 자동차 위치정보 차량 ID, 차량이동정보 [공격행위] 정보탈취/유출
(서비스에 의해 발생하는 정보) ○ 속도, 위치, 차량전장부분의 상태, 각종 센싱정보 등의 [차량 내부 정보]에 대한 [위변조] 공격	[공격대상] 차량 내부 정보 (속도, 위치, 차량전장부분 상태, 각종 센싱 정보) [공격행위] 위변조
(서비스 방해) ○ 각종 [내부 센서]의 [오작동] 유발	[공격대상] 내부 센서 [공격행위] 오작동
(서비스 방해) ○ RSU와의 [통신 차단]	[공격행위] 통신 차단

(4) Automated Teller Machine(ATM)

ATM은 현금 자동 입출금기로 은행원을 직접 통하지 않고 컴퓨터를 이용하여 직접 금융거래를 할 수 있도록 고안된 기기이다.

(가) 특성

ATM은 전철역, 터미널, 쇼핑 센터 등 다양한 장소에 설치되며 아침 일찍부터 저녁까지 혹은 24시간 내내 운영되기도 한다. 또한 무인기기라는 특징 때문에 물리적인 공격 등의 위협에 노출되기 쉽다.

ATM은 전자기기로서 일반 PC와 비슷한 내부 장치를 가지고 있으며 인터넷을 통해 해당 은행의 네트워크와 직접 연결되어 중앙서버와 입·출금 정보를 실시간적으로 공유한다. 은행 네트워크 즉 금융결제원의 통합 전산망에 연결되는 기기의 경우 다른 은행과의 거래도 가능하다[46].

(나) 가능한 악성행위

악성코드가 ATM 현금 서비스를 통해서 수행할 수 있는 악성 행위는 다음 표와 같다.

[표 3-78] VANET을 이용하여 가능한 악성행위

가능한 악성행위
○ ATM에 악성코드를 감염시키고 해당 악성코드가 중앙서버 감염
○ ATM과 연결된 중앙서버에 접속하여 금융 정보를 탈취
○ 사용자들의 카드 정보와 ATM에 입력하는 비밀번호를 탈취
○ USB 포트에 꽂으면 자동 실행되어 ATM에 감염시켜 ATM를 통제하여 원하는 만큼의 현금을 강제로 인출

(다) 매체의 일반화 적용

위에서 도출한 악성행위를 기반으로 매체의 일반화에 적용하여 연관성 정보 구조에 매칭하기 위한 요소를 도출한다. 다음 표가 매체의 일반화를 이용하여 요소들을 도출한 것이다. 이 과정이 끝나면 연관성 정보 구조와 비교하여 연관성 정보 구조에 실제로 적용할지 여부를 결정한다.

[표 3-79] ATM에서 매체의 일반화 적용

가능한 악성행위	요소
(악성코드 전파) ○ [ATM]에 악성 코드를 감염시키고 해당 악성코드가 중앙 서버를 감염시킴	[새로운 매체] ATM
(서비스에 의해 발생하는 정보) ○ ATM과 연결된 중앙서버에 접속하여 [금융 정보]를 [탈취]	[공격대상] 금융정보 [공격행위] 정보탈취
(서비스에 의해 발생하는 정보) ○ 사용자들의 [카드 정보]와 ATM에 입력하는 [비밀번호]를 [탈취]	[공격대상] 카드 정보 비밀번호 [공격행위] 정보탈취
(서비스 방해) ○ [USB 포트]에 꽂으면 자동 실행되어 ATM에 감염시켜 [ATM]를 통제하여 원하는 만큼의 현금을 [강제로 인출]	[감염매체] USB 포트 [공격대상] ATM 시스템 [공격행위] 강제 인출

제 4 절 연관성 정보 구조 활용방안

연관성 정보 구조를 활용하면 각각의 항목에 점수를 배점하고 특정 상황에 따라 가중치를 부여하는 방법으로 악성코드의 위험도를 수치화할 수 있다. 또한 악성코드가 공격 가능한 대상, 행위 그리고 감염과 전파와 관련된 매체, 주체 등에 대해 쉽게 확인할 수 있어 변종 악성코드의 유형에 대한 어느 정도의 예측이 가능하다. 그 결과 연관성 정보 구조를 이용할 경우 악성코드의 위험 지수를 이용하여 악성코드간 비교 분석이 용이하며, 악성 행위를 기반으로 변종 악성코드를 예측이 가능하다.

1. 연관성 정보 구조 위험 지수

연관성 정보 구조는 악성코드가 필수적으로 구성하고 있는 네 가지 행위인 [감염경로], [실행주체], [공격대상], [공격행위]와 선택적으로 적용할 수 있는 [전파경로], [자가보호]로 구성된다. 연관성 정보 구조를 바탕으로 악성코드에 대한 위험 지수를 도출 할 수 있으며 도출될 위험지수는 보다 사용자들에게 직관적으로 악성코드가 미칠 피해를 예측할 수 있다.

연관성 정보 구조의 위험지수는 필수 구성요소인 [감염경로], [실행주체], [공격대상], [공격행위]는 악성코드가 공격하는 대상, 정보의 수준, 악성코드 자체를 보호하기 위한 안티 디버깅 기능, 빠른 확산을 위한 다양한 전파 채널의 보유 정도 및 해당 악성코드의 잠재적인 악성행위에 따라서 가중치가 서로 다르게 부여된다. 악성코드의 실질적인 악성 행위는 레벨 3으로 분류되며 연관성 정보 구조 중 레벨 2 는 레벨 3의 구분을 위한 분류로 행위 및 대상에 따른 구분으로 특정 가중치는 부여되지 않는다. 연관성 정보 구조는 악성코드의 위험지수의 편차를 결정하는 요인을 구분하기 위하여 다음과 같은 비중으로 악성코드의 위험 지수를 적용한다. 전체 위험 지수를 10으로 가정하고 [감염경로], [실행주체], [공격대상], [공격행위]를 각 1.5:1.5:3:4로 구분한다.

위험 지수 기본적으로 다음과 같은 형태를 갖는다.

$$\text{위험 지수: } \alpha \times f(x)$$

(그림 3-9) 위험 지수 기본 공식

함수 $f(x)$ 는 [감염경로], [실행주체], [공격대상], [공격행위]를 나타낸다. 다음 수식은 확산력 α 와 $f(x)$ 를 나타낼 수 있다.

$$f(x) = [f_{route}(x) + f_{execution}(x) + \beta_{target}\{f_{target}(x) + \gamma_{behavior} \times f_{behavior}(x)\}]$$

(그림 3-10) 악성코드 위험지수 $f(x)$

$f(x)$ 는 [감염경로], [실행주체], [공격대상], [공격행위]의 합으로 결정되며 [공격대상]과 [공격행위]는 [목표대상]의 가중치가 적용되고 [공격행위]는 특정 악성행위에 대하여 [잠재위험] 가중치가 적용된다. 다음 수식은 연관성 정보 구조의 레벨 1의 위험지수 도출식을 나타낸다.

X_n : 레벨 n 점수

$$X_2 = 1$$

$$f_{route}(x) = X_1(route) \sum X_2(x) \times X_3(x)$$

$$f_{execution}(x) = X_1(execution) \sum X_2(x) \times X_3(x)$$

$$f_{target}(x) = X_1(target) \sum X_2(x) \times X_3(x)$$

$$f_{behavior}(x) = X_1(behavior) \sum X_2(x) \times X_3(x)$$

(그림 3-11) 악성행위별 위험지수 계산 식]

다음은 연관성 정보 구조 함수와 변수를 나타낸다.

$f_{route}(x)$: 감염경로
$f_{execution}(x)$: 실행주체
$f_{target}(x)$: 공격대상
$f_{behavior}(x)$: 공격행위
β_{target}	: 목표대상가중치
$\gamma_{behavior}$: 잠재력
k	: 전파력
n	: 자가보호
α	: 확산력

(그림 3-12) 연관성 정보구조 변수

α 는 다른 악성코드보다 위험도가 높은 악성코드를 구분하기 위한 확산력을 나타내며 확산력은 [전파경로]와 [자가보호]를 변수로 한다.

$$\alpha = \frac{\lfloor 10 \times \sqrt[2]{k(1.1)^n} \rfloor}{10}$$

(그림 3-13) 확산력 α

가. 감염경로

[감염경로]는 악성코드가 시스템에 설치되는 방식을 나타낸다. 다음 표는 [감염경로]에 대한 연관성 정보 구조를 나타낸다.

[감염경로]는 전체 위험 지수 중 1.5의 비중을 차지한다. 레벨 2는 시스템이 악성코드에 감염되는 경로를 세분화 하여 감염되는 매체와 매체를 통해 감염될 때의 유형, 사용자의 의존도로 구분한다.

[표 3-80] 연관성 정보 구조 : 감염경로

레벨-1	레벨-2	레벨-3	위험지수
감염경로 (1.5)	감염이용 매체	이동식 디스크	2
		모바일 디바이스	3
		하드 디스크	1
		P2P	4
		웹하드	4
		FTP클라이언트/서버	3
		웹서버	5
		메신저	2
		게시판	2
		블로그	2
		전자우편	2
		액티브X	3
		BHO	3
		클라우드 컴퓨팅	3
	감염유형	악성파일 직접 수신	2
		URL 접근	3
		원격 코드 실행	5
		권한 상승(취약점)	5
		권한 상승(암호)	4
		권한 상승(동의)	2
	사용자 의존도	사용자 직접 명령	2
		사용자 간접 명령	4
		비 의존적	5

[감염 이용 매체]의 경우 기본적으로 물리적인 매체를 통하여 접근할 경우만 악성코드로부터 감염될 경우 낮은 위험지수를 갖는다. [이동식 저장매체]의 경우 autorun.inf처럼 시스템이 제공하는 자동실행 기능으로

물리적인 매체의 접근을 허용할 경우 시스템이 빠르게 감염될 수 있으나 네트워크를 통해 자발적으로 전파될 수 없으므로 웹하드, 웹서버 등의 네트워크를 통해 전파되는 악성코드보다 낮은 위험지수를 갖는다. 또한 시스템에 악영향을 미칠 수 있는 웹서버에 접속하거나 무분별한 BHO, 액티브X를 설치하는 등 사용자의 보안의식이 미약하여 발생할 수 있는 경우 또한 낮은 위험지수를 갖는다. 예외적으로 많은 사용자가 접속할 수 있는 [웹서버]와 [웹하드]의 경우 짧은 시간동안 높은 전파속도를 보일 수 있으므로 다른 [감염경로]와 구분된 위험지수를 갖는다.

나. 실행주체

[실행주체]는 악성코드가 설치된 후 해당 악성코드를 실행하는 주체를 나타낸다. [실행주체]는 크게 사용자의 실수로 실행하는 [사용자]와 사용자의 개입이 없어도 자동적으로 실행되는 부분으로 나눌 수 있다. 레벨 2는 악성코드를 실행하는 형태에 따라서 구분 되었으며 레벨 3은 악성코드가 로드되는 시점과 실행되는 빈도에 따라서 가중치가 1-5까지 부여되었다.

[실행주체]는 전체 위험지수 중 [감염경로]와 동일한 1.5의 비율을 갖는다. 정의한 위험지수는 사용자가 자각하지 못하게 실행될 경우 다른 실행주체보다 높은 점수로 정의하였으며 취약점을 이용하여 악성코드를 실행할 경우 아직 해당 취약점에 대한 응용프로그램 및 운영체제의 업데이트가 수행되지 않았음을 의미하여 높은 위험지수가 적용된다. 또한 레벨 3의 위험지수를 정의한 기준으로는 악성코드가 감염된 이후 실행빈도를 고려하였다. [스크립트 포함 파일 열람]의 경우 부팅된 후 악성코드가 자동 실행되는 경우보다 낮은 위험지수를 갖는다.

[부트영역]의 경우 [실행주체]의 여타 다른 레벨 2의 주체보다 높은 10의 위험지수로 부여하였다. 부트영역에서부터 악성코드가 로드될 경우 여타 다른 [실행주체]보다 많은 악성행위를 수행할 수 있기 때문에 높은 위험지수를 부여하였다.

다음 표는 연관성 정보 구조의 실행주체를 나타낸 표이다.

[표 3-81] 연관성 정보 구조 : 실행주체

레벨-1	레벨-2	레벨-3	위험지수
실행주체 (1.5)	사용자	비 연속성	3
	운영체제	운영체제 취약점에 의한 악성코드 실행	5
		운영체제 수준에서 악성코드 자동 실행	3
		운영체제 파일 실행에 따른 악성코드 실행	4
	웹 브라우저	웹 브라우저 보안 취약점 이용 실행	5
		악성코드가 포함된 ActiveX 설치/실행	3
		BHO를 이용한 악성코드 실행	3
	응용 프로그램	다른 응용프로그램 보안 취약점 이용 실행	5
		스크립트 포함 파일 열람	2
		작업 스케줄러	3
		자동 업데이트 프로그램	4
		다른 악성코드	5
	부트영역	BIOS	10

다. 공격대상

[공격대상]은 악성코드가 악성 행위를 수행하는 대상을 나타낸다. 악성코드가 수집하는 정보, 정보가 변조되는 파일 등이 속하며 나아가 시스템을 직접 사용하는 사용자까지 공격 대상에 포함된다. 악성코드의 [공격대상]에 따라 유출되거나 손상되는 정보의 가치가 달라지기 때문에 전

체 연관성 정보 구조 중 [공격대상]은 3의 비중을 갖는다.
 다음 표는 [공격대상]에 관한 위험지수를 나타낸 표이다.

[표 3-82] 연관성 정보 구조 : 공격대상

레벨-1	레벨-2	레벨-3	위험지수
공격대상 (3)	시스템	시스템 장치 제어	4
		시스템 서비스 제어	5
		시스템 설정 제어	3
	사용자 정보	사용자 입력 정보	5
		사용자 저장 정보	3
		사용자 시스템 정보	2
		사용자 시스템 사용 이력	2
	시스템 사용자	경제적 손실	5
		이용 방해	3
		혼란 초래	1
	외부 장치	서비스 서버	4
		로컬 네트워크	3

[공격대상]의 정의된 위험지수는 대상이 공격당함에 따라 발생될 금전적 손해 및 다양한 악성행위를 수행할 수 있는지에 따라서 정의되었다. 단순 시스템 사용자를 불편하게 하는 [혼란 초래]의 경우 사용자의 시스템의 원활한 사용을 방해하는 [시스템 서비스 제어]보다 상대적으로 낮은 위험지수를 갖는다.

라. 공격행위

[공격행위]는 위험지수를 결정하는데 [공격대상]과 더불어 중요요인으로 작용한다. 악성코드가 유출하는 정보, 파일의 삭제/변조/생성은 시스템과 사용자에게 치명적인 피해를 입히게 된다.

공격행위는 공격행위의 대상에 따라서 [네트워크], [시스템], [파일시스

템], [프로세스], [입출력장치 오작동]으로 나뉜다. 다음은 [공격행위]의 하위 레벨과 위험지수를 나타내는 표이다.

[표 3-83] 연관성 정보 구조 : 공격행위

레벨-1	레벨-2	레벨-3	위험지수
공격행위 (4)	네트워크	정보유출	5
		서버 접속	5
		대량 트래픽 전송	5
		가로채기	3
		전송지연	2
		스팸	2
	시스템	강제 시스템 제어	5
		자원 관리	4
		네트워크 설정 변경	3
		시스템 설정 변경	3
		디스플레이 설정 변경	1
	파일 시스템	파일생성	3
		파일파괴	5
		파일변조	4
	프로세스	제어	3
		모니터링	3
	입출력장치 오작동	미디어 제어	2
		입력장치 제어	3
		출력장치 제어	2
		기타 장치 제어	1

악성코드는 수행하는 [공격행위]에 따라 시스템 및 시스템 사용자에게 미치는 피해가 높기 때문에 전체 연관성 정보 구조 중 4의 비중을 갖는다. [공격행위]의 위험지수를 정의하는 기준은 악성행위가 수행된 뒤 미치는 영향력과 피해규모에 따라 정의되었다. 단순 시스템의 디스플레이 되는 환경을 변경하는 [디스플레이 설정 변경]의 경우 여타 많은 악성행위를 수행할 수 있는 [강제 시스템 제어]보다 낮은 위험지수가 부여되었다.

마. 가중치

가중치는 연관성 정보 구조를 통해 나타나는 위험지수에 대하여 악성행위가 높은 악성코드에 대해 가중치를 적용하여 구분을 짓는 기준이 된다. 가중치는 [전파경로], [자가보호], [목표대상], [잠재위험]으로 구분되며 동일한 악성행위를 수행하더라도 높은 위협을 가하는 악성코드에 대해 높은 위험지수를 부여하고 사용자로 하여금 위험지수의 편차를 통해 위험 정도를 가늠할 수 있도록 한다.

(1) 전파경로 가중치

전파되는 채널이 많다는 것은 악성코드의 확산력을 나타낸다. 동일한 악성행위를 수행하더라도 전파되는 채널을 많이 갖는 악성코드는 위험도가 더 높다고 할 수 있다.

전파경로는 크게 이동식 저장매체와 네트워크로 구분할 수 있다. 이동식 저장매체만을 전파경로로 사용하는 악성코드는 물리적으로 시스템에 접근해야 하기 때문에 네트워크를 통해 전파되는 악성코드보다 낮은 확산력을 갖는다. 따라서 이동식 저장매체만을 전파경로로 사용하는 악성코드는 다른 전파경로를 갖는 악성코드보다 낮은 위험지수를 갖는다.

전파경로는 각 전파경로마다 가중치를 부여하는 것 보다 기존 악성코드의 전파경로 개수를 조사하고 보통, 많음, 매우 많음의 의미를 갖도록 구분하였다. 다음 표는 전파경로에 대한 가중치를 나타낸다.

[표 3-84] 연관성 정보 구조 : 전파경로

전파경로	가중치
이동식 저장 매체	0.8
1~2개 채널	1
3~5개 채널	1.2
6개 이상의 채널	1.5

(2) 자가보호 가중치

[자가보호]는 악성코드가 분석가로부터 분석되는 것을 방해하거나 안티 바이러스 소프트웨어가 악성코드를 탐지하지 못하도록 더욱 하위 레벨로 시스템에 침투하여 자신을 숨기거나 분석당하는 시간을 지연시켜 더욱 많은 시스템을 감염시키는 악성행위를 말한다. 많은 [자가보호] 기능이 적용된 악성코드는 분석하는데 시간이 더욱 걸리기 때문에 악성코드에 대한 분석을 통해 특징 및 원인, 제작자, 목적 등을 파악하는데 어려움을 준다.

다음은 악성코드가 수행할 수 있는 [자가보호] 종류를 나타낸 표이다.

[표 3-85] 연관성 정보 구조 : 자가보호

자가 보호	분석도구탐지	API
		프로세스
		CPU
	루트킷	후킹
		인젝션
		변조
	난독화	패킹
		다형성 기법

[자가보호]는 크게 분석도구 탐지, 루트킷, 난독화로 나뉜다. [자가보호]는 분석가가 해당 악성코드를 분석하는데 걸리는 시간을 지연시킨다. 분석가가 악성코드를 분석하고 해당 악성코드에 알맞은 백신을 제공하는 시간이 지연된다면 해당 악성코드는 더욱 많은 시스템을 감염시킬 수 있기 때문에 [자가보호]는 확산력을 계산하는데 적용된다.

(3) 목표대상 가중치

[목표대상] 가중치는 악성코드가 목표로 하는 대상에 따라 피해 규모 편차가 크기 때문에 이를 구분하기 위한 가중치다. 다음은 [목표대상]에 따른 가중치를 나타낸 표이다.

[표 3-86] 목표대상 위험지수 가중치

대상	가중치
군사	2
사회기반시설	1.7
행정기관	1.5
기업	1.3
개인	1

[군사]를 목표로 할 경우 인명피해가 발생할 수 있으며 악성코드가 인접한 국가를 목표로 군사 시설을 가동할 경우 국가간 대치상태를 야기하기 때문에 가장 높은 가중치인 2로 적용된다.

[사회기반시설]을 목표로 하는 악성코드는 국가를 대상으로 한 테러를 의미한다. 상수도시설 및 발전소와 같은 [사회기반시설]이 악성코드에 의하여 오작동을 일으킬 경우 인명피해와 막대한 경제적 손실, 대규모 공황을 수반하므로 1.7의 가중치가 적용된다.

[행정기관]은 자국민의 중요한 정보들을 저장하고 있으며 행정업무의 장애가 발생할 경우 사회적인 혼란을 야기할 수 있어 가중치 1.5가 적용되었다.

[기업]을 대상으로 하는 악성코드는 경쟁사의 이미지 실추, 영업방해, 금전적 손실을 목표로 하거나 국제적 범죄조직이 기업을 대상으로 하여 금전을 편취하기 위한 악성코드이다. 악성코드의 대상이 되는 [기업]이 사회에 막대한 영향을 미치는 대기업의 경우 천문학적 금전적 피해가 발생할 수 있으며 중소기업인 경우 발빠른 대처가 불가능 하기 때문에

1.3의 가중치가 적용되었다.

악성코드의 목표가 [개인]인 경우 위험지수가 변동되지 않는 1의 가중치를 갖는다. 하지만 [개인]시스템을 좀비PC로 만들어 제 3의 공격에 악용될 경우가 있다. 이 경우 악성코드는 [잠재위험]에 영향을 받기 때문에 적정 위험지수를 도출할 수 있다.

만약 악성코드가 동시다발적으로 기업/행정기관/사회기반시설 등 복수개의 목표를 가질 경우 가장 위협수준이 높은 가중치가 적용된다.

(4) 잠재위험 가중치

[공격행위]의 경우 수행하는 [공격행위]에 따라 제 3의 공격에 악용될 수 있기 때문에 [잠재위험]이라는 [공격행위]에만 적용되는 가중치가 부여된다. [잠재위험]은 해당 악성행위 수행 후 향후 공격자가 해당 시스템을 제 3의 공격에 악용하거나 다른 악성행위를 수행할 수 있는 여지를 남겨두는 악성행위들을 뜻한다. 다음은 [잠재위험]에 적용되는 [공격행위]를 나타내며 해당 가중치는 [공격행위]에만 적용된다.

[표 3-87] 잠재위험 위험지수 가중치

악성행위	가중치
서버접속	0.075
강제 시스템 제어	0.075
네트워크 설정 변경	0.075
시스템 설정 변경	0.075

[잠재위험]은 C&C서버에 접속하여 실시간으로 공격자로부터 명령을 하달 받아 악성행위를 수행할 수 있는 [서버접속]과 원격에서 시스템을 제어할 수 있는 [강제 시스템 제어], 외부에서 접속을 허용하도록 설정을 변경할 수 있는 [네트워크 설정 변경], 사용자의 권한상승을 통하여 다양한 악성행위를 수행하도록 하는 [시스템 설정 변경]이 속한다.

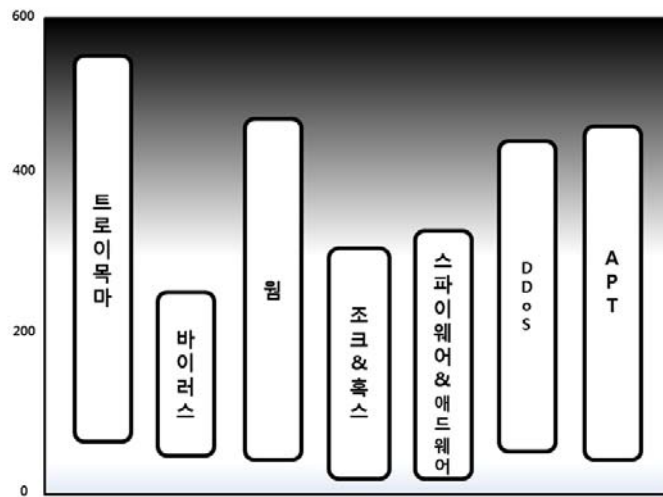
[잠재위험]의 가중치는 기본 값 1에 더하여 덧셈형태를 취한다. 각 악성행위마다 0.075의 고정된 값을 가지며 네 가지 악성행위를 모두 수행할 경우 1.3의 [잠재위험]가중치를 갖는다.

2. 악성코드 그룹 별 위험 지수 분포

악성코드의 특징에 따른 위험 지수의 분포를 확인하기 위해서 악성코드 그룹을 다음 표와 같이 나누었다.

[표 3-88] 악성코드 그룹

트로이 목마
바이러스
웜
스파이웨어, 애드웨어
조크, 혹스
DDoS 봇
APT (Advanced Persistent Threat)



(그림 3-14) 악성코드 그룹 별 위험 지수 분포

각 악성코드 그룹 별로 연관성 구조에서 필수적으로 가지는 특성들의 위험 지수는 포함시키고 절대 가질 수 없는 특성들의 위험 지수는 제외시킨 다음, 해당 그룹의 악성코드가 가질 수 있는 위험 지수의 최소값과 최대값을 계산했다. 결과는 (그림 3-14)와 같다.

트로이 목마와 웜은 다양한 악성 행위가 가능하기 때문에, 점수 또한 넓은 영역에 분포한다. 조크와 후스 그리고 스파이웨어 애드웨어는 수행하는 악성 행위들이 다른 악성코드 그룹에 비해 점수 배점이 낮기 때문에 다른 그룹에 비해 낮은 영역에 분포를 한다. APT의 경우, 수행하는 악성 행위의 종류는 다양하지 않지만, 각각의 행위가 점수 배점이 높아 트로이 목마와 웜 다음으로 넓은 영역에 분포를 한다. 바이러스의 경우, 특징이 뚜렷해서 수행하는 악성행위들이 한정적이기 때문에 상대적으로 좁은 영역에 분포를 한다. 악성코드 그룹 별 특징은 다음과 같다.

본 악성코드 그룹 별 위험지수 분포는 최대값 산정 시 모든 가중치를 적용하므로 단순히 분포 범위가 높은 트로이목마, 웜이 DDoS, APT악성코드보다 위험도가 높다고 판단할 수 없음에 유의한다.

가. 트로이 목마

트로이 목마는 주로 사용자의 실수 또는 사회 공학적 기법으로 인해 설치가 되며, 정보 유출과 시스템 제어 기능을 가지고 있다. 따라서 위험 지수의 범위를 계산할 때 정보 유출과 시스템 제어 관련 항목들을 포함시키고 트로이 목마와 관련이 없는 항목들은 제외시켰다.

[표 3-89] 연관성 구조 제외 항목 : 트로이 목마

레벨-1	레벨-2	레벨-3
감염 경로	감염 유형	원격 코드 실행
		권한 상승(취약점)
		권한 상승(암호)
		권한 상승(동의)
	사용자 의존도	비 의존적
실행 주체	운영체제	운영체제 취약점에 의한 악성코드 실행
	응용 프로그램	다른 응용프로그램 보안 취약점 이용 실행
		다른 악성코드
	웹 브라우저	악성코드가 포함된 ActiveX 설치/실행
	부트 영역	BIOS
공격 대상	시스템 사용자	경제적 손실
		이용 방해
		혼란 초래
	외부 장치	서비스 서버
		로컬 네트워크
공격 행위	네트워크	서버 접속
		대량 트래픽 전송
		전송 지연
		스팸
	시스템	디스플레이 설정 변경
	파일 시스템	파일 생성
		파일 파괴
		파일 변조
	프로세스	제어
		모니터링
	입출력 장치 오작동	미디어제어
		입력장치 제어
		출력장치 제어
		기타 장치

[표 3-90] 연관성 구조 포함 항목 : 트로이 목마

레벨-1	레벨-2	레벨-3
공격 행위	네트워크	정보 유출
	시스템	강제 시스템 제어

[표 3-91] 위험 지수 : 트로이 목마

최대값	552.86
최소값	47.6

[표 3-92] 연관성 구조 제외 항목 : 바이러스

레벨-1	레벨-2	레벨-3
공격 대상	사용자 정보	사용자 입력 정보
		사용자 저장 정보
		사용자 시스템 정보
	사용자 정보	사용자 시스템 사용 이력
		사용자 저장 정보
		시스템 사용 이력
	시스템 사용자	경제적 손실
		이용 방해
		혼란 초래
	외부 장치	서비스 서버
		로컬 네트워크
공격 행위	네트워크	정보 유출
		서버 접속
		대량 트래픽 전송
		가로채기
		전송지연
		스팸
	시스템	강제 시스템 제어
		자원 관리
		네트워크 설정 변경
		시스템 설정 변경
		디스플레이 설정 변경
	프로세스	제어
		모니터링
	입출력 장치 오작동	미디어 제어
		입력장치 제어
		출력장치 제어
		기타 장치

나. 바이러스

숙주를 통해서 전파를 하고 사용자 컴퓨터에 피해를 입힌다. 주로 시스템 파괴 항목들을 위험 지수 계산에 필수적으로 포함시켰다.

[표 3-93] 연관성 정보 구조 포함 항목 : 바이러스

레벨-1	레벨-2	레벨-3
공격 행위	파일 시스템	파일 파괴

[표 3-94] 위험 지수 : 바이러스

최대값	228.9
최소값	31.6

다. 웜

자기 복제 기능이 있으며, 네트워크를 통해서 자기 자신을 전파한다. 따라서 네트워크와 관련된 행위들을 위험 지수 계산에 포함시켰다. 그리고 바이러스와 비슷하게 시스템에 피해를 입힐 수 있기 때문에 파일 시스템 항목들도 계산에 포함시켰다.

[표 3-95] 연관성 정보 구조 포함 항목 : 웜

레벨-1	레벨-2	레벨-3
공격 대상	외부 장치	서비스 서버
	외부 장치	로컬 네트워크

[표 3-96] 위험 지수 : 웜

최대값	430.22
최소값	37.6

[표 3-97] 연관성 구조 제외 항목 : 웹

레벨-1	레벨-2	레벨-3
감염 경로	감염 유형	악성파일 직접 수신
		URL 접근
		권한 상승(동의)
	사용자 의존도	사용자 직접 명령
		사용자 간접 명령
실행 주체	사용자	비 연속성
공격 대상	시스템	시스템 장치 제어
		시스템 서비스 제어
		시스템 설정 제어
	사용자 정보	사용자 입력 정보
		사용자 저장 정보
		사용자 시스템 정보
		사용자 시스템 사용 이력
	시스템 사용자	이용 방해
		혼란 초래
공격 행위	네트워크	정보 유출
		서버 접속
		대량 트래픽 전송
		가로채기
		전송지연
	시스템	강제 시스템 제어
		자원 관리
		시스템 설정 변경
		디스플레이 설정 변경
	입출력 장치 오작동	미디어 제어
		입력장치 제어
		출력장치 제어
		기타 장치

라. 조크, 혹스

사용자에게 큰 피해는 입히지 않고 단순히 장난을 치거나 유언비어를 퍼트리는 종류의 악성코드이다. 사용자에게 혼란을 주는 등의 단순한 행위를 포함시키고 큰 피해를 입힐 가능성이 있는 항목들은 위험 지수 계산에서 제외시켰다.

[표 3-98] 연관성 구조 제외 항목 : 조크, 혹스

레벨-1	레벨-2	레벨-3
감염 경로	감염 유형	원격 코드 실행
		권한 상승(취약점)
		권한 상승(암호)
	사용자 의존도	비 의존적
실행 주체	운영체제	운영체제 취약점에 의한 악성코드 실행
		운영체제 수준에서 악성코드 자동 실행
		운영체제 파일 실행에 따라 악성코드 실행
	응용 프로그램	다른 응용프로그램 보안 취약점 이용 실행
		스크립트 포함 파일 열람
		작업 스케줄러
		자동 업데이트 프로그램
		다른 악성코드
	웹 브라우저	웹브라우저 보안 취약점 이용 실행
	부트 영역	BIOS
공격 대상	시스템	시스템 장치 제어
		시스템 서비스 제어
		시스템 설정 제어
	사용자 정보	사용자 입력 정보
		사용자 저장 정보
		사용자 시스템 정보

공격 행위	시스템 사용자 외부 장치	사용자 시스템 사용 이력
		경제적 손실
		서비스 서버
	네트워크	로컬 네트워크
		정보 유출
		서버 접속
		대량 트래픽 전송
		가로채기
		전송 지연
	시스템	강제 시스템 제어
		자원 관리
		네트워크 설정 변경
	파일 시스템	파일 생성
		파일 파괴
		파일 변조

[표 3-99] 연관성 정보 구조 포함 항목 : 조크, 흑스

레벨-1	레벨-2	레벨-3
공격 대상	시스템 사용자	이용 방해
		혼란 초래

[표 3-100] 위험 지수 : 조크, 흑스

최대값	310.1
최소값	22.4

마. 스파이웨어, 애드웨어

스파이웨어는 사용자 몰래 컴퓨터에 설치가 되어 정보를 수집하거나 모니터링하는 악성코드이며, 애드웨어는 사용자의 동의없이 광고를 띄우는 악성코드이다. 사용자 UI에 변경을 가하거나 정보를 수집하는 항목을 포함시켰다.

[표 3-101] 연관성 정보 구조 제외 항목 : 스파이웨어, 애드웨어

레벨-1	레벨-2	레벨-3
감염 경로	감염 유형	원격 코드 실행
		권한 상승(취약점)
		권한 상승(암호)
		권한 상승(동의)
	사용자 의존도	사용자 간접 명령 비 의존적
실행 주체	운영체제	운영체제 취약점에 의한 악성코드 실행
		운영체제 수준에서 악성코드 자동실행
		운영체제 파일 실행에 따라 악성코드 실행
	응용 프로그램	다른 응용프로그램 보안 취약점 이용 실행
		스크립트 포함 파일 열람
		작업 스케줄러
		자동 업데이트 프로그램
		다른 악성코드
	부트 영역	BIOS
공격 대상	시스템	시스템 장치 제어
		시스템 서비스 제어
		시스템 설정 제어
	외부 장치	서비스 서버 로컬 네트워크
공격 행위	네트워크	서버 접속
		대량 트래픽 전송
		가로 채기
		전송 지연
	시스템	강제 시스템 제어
		자원 관리
		네트워크 설정 변경
		시스템 설정 변경 디스플레이 설정 변경

	파일 시스템	파일 파괴
		파일 변조
	입출력장치 오작동	미디어 제어
		입력장치 제어
		출력장치 제어
		기타 장치

[표 3-102] 연관성 구조 포함 항목 : 스파이웨어, 애드웨어

레벨-1	레벨-2	레벨-3
공격 행위	네트워크	정보 유출

[표 3-103] 위험 지수 : 스파이웨어, 애드웨어

최대값	387.1
최소값	28

바. DDoS 봇

시스템을 감염하고 C&C 서버로부터 명령을 받아 다른 시스템을 DDoS 공격한다. 따라서 시스템을 제어하거나 DDoS 공격을 하는 항목을 포함시키고 관련 없는 항목들을 제외시켰다.

[표 3-104] 연관성 정보 구조 제외 항목 : DDoS 봇

레벨-1	레벨-2	레벨-3
감염 경로	감염 유형	원격 코드실행
		권한 상승(취약점)
		권한 상승(암호)
	사용자 의존도	권한 상승(동의)
실행 주체	운영체제	사용자 직접 명령
		운영체제 약점에 의한
		악성코드 실행
		운영체제 수준에서 악성코드 자동 실행
		운영체제 파일 실행에 따라

		악성코드 실행
	부트 영역	BIOS
공격 대상	사용자 정보	시스템 장치 제어
		시스템 서비스 제어
		시스템 설정 제어
	시스템 사용자	경제적 손실
		이용 방해
		혼란 초래
공격 행위	네트워크	정보 유출
		가로채기
		전송 지연
		스팸
	시스템	디스플레이 설정 변경
	파일 시스템	파일 생성
		파일 파괴
		파일 변조
	프로세스	제어
		모니터링
	입출력 장치 오작동	미디어 제어
		입력장치 제어
		출력 장치 제어
		기타 장치

[표 3-105] 연관성 정보 구조 포함 항목 : DDoS 봇

레벨-1	레벨-2	레벨-3
공격 대상	외부 장치	서비스 서버
		로컬 네트워크
공격 행위	네트워크	서버 접속
		대량 트래픽 전송

[표 3-106] 위험 지수 : DDoS 봇

최대값	439.6
최소값	50

사. APT

제로 데이 또는 고급 기술을 사용하며, 상대방에게 노출되지 않고 정보를 유출하거나 시스템을 제어하는 것이 목적이다. 사용자가 공격을 인지할 수 있는 항목이나 정보 유출과 상관이 없는 항목들은 제외시켰다.

[표 3-107] 연관성 정보 구조 제외 항목 : APT

레벨-1	레벨-2	레벨-3
감염 경로	감염 유형	악성파일 직접 수신
		URL 접근
		권한 상승(취약점)
		권한 상승(동의)
	사용자 의존도	사용자 직접 명령 사용자 간접 명령
실행 주체	사용자	비연속성
	운영체제	운영체제 수준에서 악성코드 자동실행
		운영체제 파일 실행에 따라 악성코드 실행
		스크립트 포함 파일 열람
	응용프로그램	작업 스케줄러
		자동 업데이트 프로그램
		다른 악성코드
	웹 브라우저	악성코드가 포함된 ActiveX 설치/실행
		BHO를 이용한 악성코드 실행
공격 대상	시스템 사용자	경제적 손실
		이용 방해
		혼란 초래
	외부 장치	서비스 서버
		로컬 네트워크
공격 행위	네트워크	대량 트래픽 발송

		가로채기
		전송 지연
		스팸
	시스템	자원관리
		네트워크 설정 변경
		시스템 설정 변경
		디스플레이 설정 변경
	파일시스템	파일생성
		파일파괴
		파일 변조
	프로세스	제어
		모니터링
	입출력장치 오작동	미디어 제어
		입력장치 제어
		출력장치 제어
		기타 장치

[표 3-108] 연관성 정보 구조 포함 항목 : APT

레벨-1	레벨-2	레벨-3
감염 경로	사용자 의존도	비 의존적
공격 행위	네트워크	정보 유출
	시스템	강제 시스템 제어

[표 3-109] 위험 지수 : APT

최대값	447.3
최소값	40

3. 변종 악성코드 예측

연관성 정보 구조는 악성코드 발생 시 분석 결과를 바탕으로 악성코드를 정의 할 수 있다. 더 나아가, 연관성 정보 구조의 [종속성]과 [배타성] 특징을 활용하여 분석된 악성코드의 변종을 예측하여 잠재적인 위험과 변종 악성코드를 도출할 수 있다. 이는 악성코드가 수행할 수 있는 악성 행위를 정의·분류하여 그룹화 하고 분석된 악성코드를 바탕으로 해당 악

성코드의 그룹을 도출한다. 악성코드 그룹은 앞서 정의한 목적행위를 기반으로 한 7가지의 악성코드 그룹을 사용한다. 이후 특정 그룹에 속하는 악성코드는 앞서 정의한 그룹의 악성행위를 수행할 수 있기 때문에 악성코드의 변종을 예측할 수 있다.

연관성 정보 구조의 [종속성] 특징은 분석된 악성코드 정보를 바탕으로 동시에 일어날 수 있는 악성코드의 행위를 의미한다. [배타성]은 해당 악성코드와 동시에 일어날 수 없는 악성코드의 행위를 의미한다.

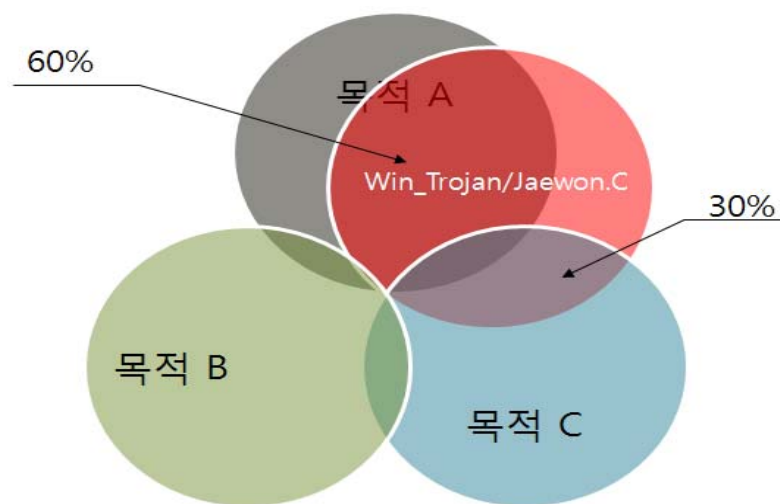
연관성 정보 구조를 7개의 그룹에 대하여 해당 그룹에 속할 수 있는 악성행위를 구분 짓는다. 이 후 분석된 악성코드에 대한 정보를 바탕으로 해당 악성코드의 목적을 도출할 수 있다.

[표 3-110] 악성코드 분류 그룹

그룹	분류 기준	예
정보탈취형	- 사용자 정보를 외부로 유출	트로이목마, 스파이웨어
과금유발형	- 금전적 이익을 위해 개발 - 광고 팝업, 결제 유도	허위 백신, 애드웨어
시스템파괴형	- 시스템 자원이 목적	DDoS, 로직 폭탄
모듈형	- 자체적인 악성행위가 없음 - 악성코드 다운로드 및 생성	드롭퍼, 다운로더
원격제어형	- 외부로부터의 침입 - 시스템 권한을 획득하고 제어	악성 봇
유해가능형	- 세어웨어, 정상 프로그램 - 악용 가능한 행위를 수행하는 프로그램	휴리스틱스 탐지
혼란야기형	- 사용자가 불편함을 느끼게 함 - 공포사진 팝업, 화면보호기 변경	조크, 혹스

연관성 정보 구조를 바탕으로 변종 악성코드를 예측하는 과정은 다음과 같다.

- 분석된 악성코드를 연관성 정보 구조에 대입한다.
- 메타데이터를 바탕으로 해당 악성코드의 목적을 도출한다. 연관성 정보 구조는 7 가지의 목적 그룹에 대한 메타데이터가 된다.
- 목적이 도출된 악성코드는 목적 그룹과의 일치률을 계산한다. 악성코드의 목적이 4가지 이상일 경우 일치률이 높은 세 그룹을 보고한다.
- 해당 악성코드가 특정 그룹과 60% 일치한다면, 해당 그룹의 나머지 40%의 악성 행위 중 일부를 추가적으로 수행하는 변종 악성코드가 출현할 가능성을 예측할 수 있다. 그리고 목적이 두 가지 이상일 경우는 해당 악성코드의 제작 목적이 여러 가지인 것을 뜻한다.
- [배타성]은 해당 악성코드에 대하여 변종을 유추할 경우 정확한 목적을 구분하기 위해 사용된다. 분석된 악성코드가 수행하는 악성 행위에 대한 배타적 속성을 갖는 악성행위를 배제하면 악성코드에 대해 명확한 분류가 가능할 뿐만 아니라 변종 악성코드를 유추할 시 정확도를 높일 수 있다.



(그림 3-15) 변종 악성코드의 목적 그룹 일치률

다음은 연관성 정보 구조와 목적그룹에 대한 연관관계를 나타낸 표이다.

[표 3-111] 연관성 정보 구조와 분류 그룹 연관관계 : 공격대상

레벨-1	레벨-2	레벨-3	목적그룹
공격 대상	시스템	시스템장치제어	원격제어형 혼란야기형
		시스템 서비스 제어	원격제어형 모듈형
		시스템 설정 제어	유해가능형 모듈형 원격제어형
	사용자정보	사용자 입력정보	정보탈취형
		사용자 저장 정보	정보탈취형
		사용자 시스템 정보	정보탈취형
		사용자 시스템 사용이력	정보탈취형
	시스템사용자	경제적 손실	과금유발형
		이용방해	시스템파괴형
		혼란초래	혼란야기형
	외부 장치	서비스 서버	시스템파괴형
		로컬네트워크	시스템파괴형

[표 3-112] 연관성 정보 구조와 분류 그룹 연관관계 : 공격행위

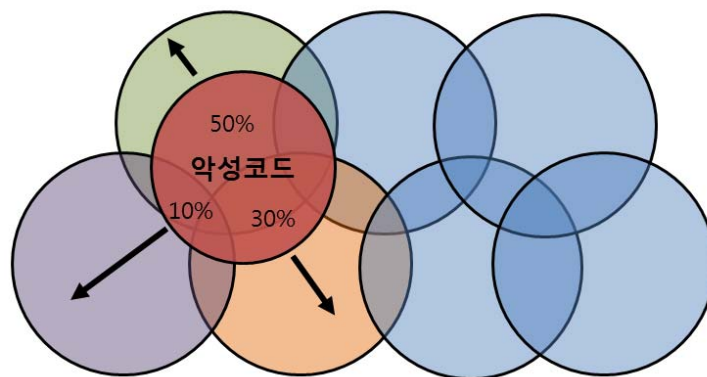
레벨-1	레벨-2	레벨-3	목적그룹
공격 행위	네트워크	정보유출	정보탈취형
		서버접속(웹서버,FTP,IRC등)	정보탈취형 시스템파괴형 모듈형 원격제어형
			시스템파괴형
			시스템파괴형
		대량트래픽전송	정보탈취형

		전송지연	혼란유발형
		스팸	과금유발형
	시스템	강제시스템제어	원격제어형 혼란야기형
		자원관리	시스템파괴형
		네트워크설정변경	모듈형 과금유발형 유해가능형
		시스템설정변경	정보탈취형 과금유발형 시스템파괴형 모듈형 원격제어형 유해가능형 혼란야기형
		디스플레이설정변경	혼란야기형
	파일시스템	파일생성	정보탈취형 과금유발형 시스템파괴형 모듈형 원격제어형 유해가능형 혼란야기형
		파일파괴	시스템파괴형
		파일 변조	모듈형 혼란야기형
	프로세스	제어	모듈형 유해가능형 혼란야기형 과금유발형

		모니터링	모듈형 유해가능형
	입출력장치 오작동	미디어제어	혼란야기형
		입력장치제어	정보탈취형 혼란야기형
		출력장치제어	혼란야기형
		기타장치	혼란야기형

위의 표 중 [감염경로]와 [실행주체]의 경우 모든 목적행위 그룹에 속할 수 있기 때문에 변종 악성코드 탐지를 위한 연관성 정보 구조와 목적행위별 분류그룹에서는 제외되었다.

어떤 임의의 악성코드를 분석해본 결과 [정보유출형]과 50%, [시스템 파괴형]과 30% 그리고 [혼란야기형]과 10% 일치한다면, 나머지 하위 네 가지 목적 그룹을 제외한 상위 세 가지 목적 그룹을 분석가에게 보고한다.

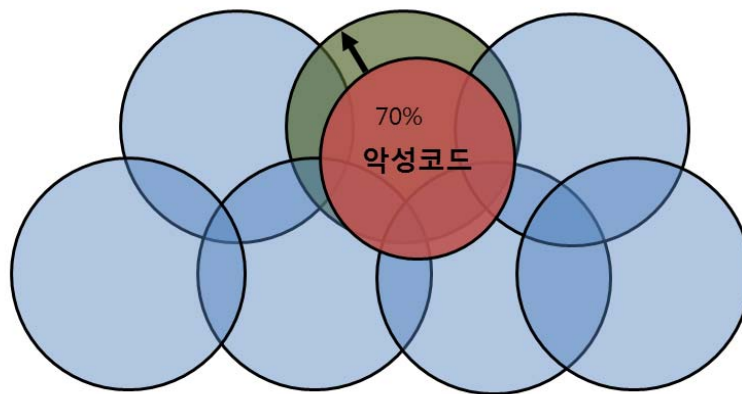


(그림 3-16) 악성코드 변종 예측

위 그림은 각 목적 그룹과의 일치률을 시각적으로 표현하고 있다. 초

록색, 보라색, 주황색은 차례대로 정보 유출형, 시스템 파괴형 그리고 혼란 야기형을 표현한다. 변종이라는 것은 기본적으로 그 종류에 들면서 일부분이 다른 것을 뜻하기 때문에 일치률이 낮은 다른 목적 그룹보다는 화살표 방향으로 변종이 일어날 가능성이 높다.

아래 그림의 경우와 같이, 만약 한 가지의 목적 그룹과 70% 이상의 일치률을 보이면, 그 목적 그룹만 보고를 한다. 이런 악성코드의 경우, 낮은 일치률을 보이는 다른 그룹으로 변종이 일어나지 않고 화살표 방향으로 변종이 일어날 가능성이 매우 높다는 것을 예측할 수 있다.



(그림 3-17) 70% 이상의 일치률을 가지는 경우 변종 예측

가. 변종 예측 예시

실제로 변종을 예측해보기 위해 다음과 같이 하나의 악성코드에 관련하여 연관성 정보 구조에 대입 후 해당 악성코드의 목적과 변종 악성행위의 가능성을 나타내었다. 변종 예측에 사용된 악성코드는 2011년 3월 3~4일에 발생한 DDoS 악성코드로 3.4 DDoS로 알려진 악성코드이다.

[표 3-113] 3.4 DDoS 악성코드

레벨-1	레벨-2	레벨-3	목적 그룹
공격대상	사용자 정보	사용자 입력정보	정보탈취형
	시스템 사용자	경제적 손실	과금유발형
	외부 장치	서비스 서버	시스템파괴형
공격행위	네트워크	서버 접속	모듈형 원격제어형 정보탈취형 시스템파괴형
		대량 트래픽 전송	시스템파괴형
	시스템	시스템 설정 변경	정보탈취형 원격제어형 시스템파괴형 과금유발형 모듈형 유해가능형 혼란야기형
	파일시스템	파일생성	정보탈취형 원격제어형 시스템파괴형 과금유발형 모듈형 유해가능형 혼란야기형
		파일파괴	시스템파괴형 유해가능

이 악성코드가 가지고 있는 8 가지 항목과 목적 그룹간의 일치률은 다음 표와 같다.

[표 3-114] 목적 그룹과의 일치률

목적 그룹	일치률
시스템 파괴형	75%
과금유발형	50%
정보탈취형	44%
원격제어형	42%
유해가능형	42%
모듈형	33%
혼란야기형	22%

3.4 DDoS 악성코드는 [시스템파괴형] 악성코드와 가장 높은 일치률을 나타낸다. [시스템파괴형] 악성코드는 시스템의 자원을 고갈하거나 파일 삭제 등으로 시스템 및 서비스에 대한 공격을 수행하는 악성코드를 나타낸다. 연관성 정보 구조는 위와 같이 해당 악성코드의 목적을 정확히 도출할 수 있으며 또한 악성코드의 목적을 바탕으로 [시스템파괴형] 악성코드가 수행할 수 있는 악성행위를 나타낼 수 있어 변종을 예측할 수 있다. [시스템파괴형] 목적 그룹에 해당하는 항목들은 다음 표와 같으며 해당하는 악성행위를 수행하는 변종의 3.4 DDoS 악성코드가 나타날 수 있음을 의미한다.

[표 3-115] 시스템파괴형 악성행위 항목

레벨-1	레벨-2	레벨-3
공격대상	외부 장치	서비스 서버
		로컬 네트워크
공격 행위	시스템 사용자	이용방해
	네트워크	대량 트래픽 전송
	시스템	자원 관리
		시스템 설정 변경
	파일 시스템	파일 생성
		파일 파괴

제 5 절 연관성 정보 구조 예시

연관성 정보 구조를 통하여 실제 악성코드의 분석보고서를 바탕으로 악성코드의 위험지수를 계산해 볼 수 있다. 악성코드 위험지수 도출에 관해 사용된 분석 정보는 국내 안티 바이러스 소프트웨어 업체인 ‘안철수 연구소’의 악성코드 분석정보를 바탕으로 하였다. 해당하는 악성코드 분석 보고서가 연관성 정보구조를 바탕으로 제작되지 않다. 따라서 악성코드의 구조를 이루는 악성행위에 대해서 언급이 없으면 수행 가능한 악성행위를 임의로 적용하여 정확한 위험지수를 표현할 순 없지만 실제 악성코드 종류의 따라서 나타나는 위험지수의 변화를 관측할 수 있다. 악성코드 위험지수 도출 예시에 사용된 악성코드 정보는 국내에서 많이 발견되는 ‘특정 온라인 게임계정 탈취(트로이 목마)’ 악성코드와 워의 형태로 전파되어 시스템 감염 후 IRC 서버에 접속, 오픈가 하달하는 명령을 수행하는 악성코드(워), 그리고 사회적 이슈로 떠올랐던 3.4 DDoS 악성코드를 예시로 한다.

1. 특정 온라인 게임계정 탈취 악성코드(트로이목마)

안철수 연구소의 진단명 ‘Win-Trojan/Onlinegamehack.115224’이 적용된 온라인 게임 계정 탈취 악성코드로 해당 악성코드를 연관성 정보 구조에 적용한 경우 다음 표와 같은 결과를 얻을 수 있다.

본 악성코드는 정상파일을 가장한 악성코드를 공격자가 특정 게시판에 업로드하고 특정 시스템 사용자가 게임을 실행하여 사용자 입력 정보를 입력 시 해당 정보를 외부로 유출시키는 악성코드다.

[표 3-116] 악성코드 위험지수 도출 예시 : 트로이목마

레벨-1(비율)	레벨-2	레벨-3	위험지수
감염 경로(1.5)	감염이용 매체	게시판	2
	감염 유형	악성파일 직접 수신	2
	사용자 의존도	사용자 직접 명령	2
실행 주체(1.5)	운영체제	운영체제 수준에서 악성코드 자동실행	3
공격 대상(3)	사용자 정보	사용자 입력정보	5
	시스템 사용자	경제적 손실	5
공격 행위(4)	네트워크	정보유출	5
	시스템	시스템 설정 변경	3/0.075
	파일시스템	파일생성	3
	프로세스	모니터링	3

2. 악성봇(웜)

안철수 연구소의 진단명 ‘Win32/IRCBot.worm.86016.AL’이 부여된 악성코드로 IRC서버에 접속하여 공격자가 하달하는 명령에 따라 악성행위를 수행한다. 해당 악성코드를 연관성 정보 구조에 적용한 경우 다음 표와 같은 결과를 얻을 수 있다.

[표 3-117] 악성코드 위험지수 도출 예시 : IRC봇(웜)

레벨-1(비율)	레벨-2	레벨-3	위험지수
감염 경로(1.5)	감염이용 매체	네트워크	5
	감염 유형	권한상승(취약점)	5
		권한상승(암호)	4
	사용자 의존도	사용자 직접 명령	2
실행 주체(1.5)	운영체제	운영체제 수준에서 악성코드 자동실행	3
공격 대상(3)	사용자 정보	사용자 입력정보	5
	시스템 사용자	경제적 손실	5
	외부 장치	서비스 서버	4
공격 행위(4)	네트워크	서버 접속	5/0.075
	시스템	시스템 설정 변경	3/0.075
		강제 시스템 제어	5/0.075
	파일시스템	파일생성	3

본 악성코드는 운영체제의 취약점을 통하여 네트워크로 자체적으로 전파되는 악성코드이다. 전파된 후 보안이 취약한 암호리스트를 바탕으로 시스템의 권한을 획득한다. 이후 악성코드는 IRC서버에 접속하여 공격자가 명령을 하달할 때 까지 대기한다.

3. 3.4 DDoS 악성코드

2011년 3월 4일에 발생한 행정기관 및 기업 40여 곳을 목표로 한 DDoS악성코드는 2009년 7월 7일 발생한 DDoS 공격보다 낮은 피해를 입었지만 수행한 악성행위는 더욱 위협적이었다. 다음은 해당 악성코드가 수행한 악성행위를 연관성 정보 구조에 기반하여 정리된 표를 나타낸다.

[표 3-118] 악성코드 위험지수 도출 예시 : 3.4 DDoS

레벨-1(비율)	레벨-2	레벨-3	위험지수
감염 경로(1.5)	감염이용 매체	웹하드	5
	감염 유형	악성파일 직접 수신	2
	사용자 의존도	사용자 직접 명령	2
실행 주체(1.5)	부트영역	BIOS	10
	운영체제	운영체제 수준에서 악성코드 자동실행	3
공격 대상(3)	사용자 정보	사용자 입력정보	5
	시스템 사용자	경제적 손실	5
	외부 장치	서비스 서버	4
공격 행위(4)	네트워크	서버 접속	5/0.075
		대량 트래픽 전송	5
	시스템	시스템 설정 변경	3/0.075
	파일시스템	파일생성	3
		파일파괴	5

해당 악성코드는 DDoS공격 모듈, 하드디스크 파괴 모듈, C&C 접속 모듈과 같이 세 모듈로 모듈화되어 전파되었지만 세 악성코드 모듈이 연계되어 특정 목적을 위해 제작되었기에 하나의 악성코드로 간주하였다.

4. 악성코드 위험지수 도출 결과

위의 예시들을 통하여 실제 악성코드의 분석결과를 연관성 정보구조에 반영하였다. 다음은 반영된 결과를 바탕으로 계산된 위험지수와 해당 악성코드의 목적그룹 및 변종 가능 악성행위를 나타낸다. 변종가능 악성행위의 경우 해당 악성코드가 수행하지는 않지만 수행 가능한 악성행위만을 나타낸다.

[표 3-119] 악성코드 위험지수 도출 결과

악성코드 종류	위험지수	목적그룹
1. 트로이목마	168	정보유출형: 55% 과금유발형: 50% 유해가능형: 42%
2. IRC봇(웜)	178.6	원격제어형: 57% 정보탈취형: 44%
3. 3.4 DDoS	254.7	시스템파괴형 : 75% 과금유발형: 50% 정보탈취형: 44%

[표 3-120] 변종 악성행위 목록

악성코드 종류	변종 악성행위		
트로이목마	공격대상	사용자 정보	사용자 저장 정보 사용자 시스템 정보 사용자 시스템 사용 이력
	공격행위	네트워크	가로채기
		입출력장치	입력장치 제어
IRC봇(웜)	공격대상	시스템	시스템 장치 제어 시스템서비스 제어 시스템 설정 제어
	공격행위	-없음	
3.4DDoS	공격대상	외부 장치	로컬 네트워크
		시스템 사용자	이용방해
	공격행위	시스템	자원관리

위의 위험지수는 [전파경로]와 [자가보호] 내용의 부재로 확산력은 [자가보호]가 두 가지 방법이 적용된 악성코드로 임의정의 하였다.

위험지수 및 악성코드의 목적, 변종 악성행위를 예측한 결과 연관성

정보 구조에 정확히 부합하지 않는 악성코드 분석 보고서라도 해당 악성코드의 목적그룹과 다양한 변종행위를 정확히 나타낼 수 있음을 보인다.

분석 보고서가 연관성 정보 구조를 바탕으로 제작될 경우 보다 정확한 위험지수를 도출하여 사용자로 하여금 악성코드의 위협정도를 쉽게 확인할 수 있으며 정형화된 분석 보고서 방식을 제공하여 분석가들에게 높은 업무효과를 얻을 수 있다.

제 4 장 결론

본 연구 보고서에서는 악성코드의 공격기술, 전파경로를 바탕으로 악성코드를 정의·분류하기 위한 기준들을 도출하였다. 그 결과 감염경로, 실행주체, 공격대상, 공격행위, 전파경로, 자가보호와 같은 6가지 정의 기준을 도출하였으며 각각의 기준들을 이용하여 공통된 형태의 악성코드 연관성 정보 구조를 만들었다. 연관성 정보 구조는 악성코드의 감염이 이루어진 샘플에 대한 확보부터 악성코드 샘플을 이용한 분석과정, 악성코드 확산 현황 분석 과정에서 얻을 수 있는 종합적인 정보들을 기준으로 악성코드를 정의·분석할 수 있는 구조이다. 따라서 현재 악성코드 분석과정이 주로 소스코드 수준에서 이루어지고 있는 한계점을 보완할 수 있는 정보 구조로 악성코드 분석 환경 개선의 초석이 될 것이다.

또한 연구기관에 악성코드 연관성 정보 구조가 적용되면, 현재 판이하게 다른 구조를 갖는 연구 기관별 분석 과정 및 분석 결과에 대한 공통된 기준 설립이 가능하다. 그 결과 현재 각 연구 기관별로 독립적으로 이루어지던 악성코드 분석이 연구 기관들의 협력을 통해 보다 효과적으로 이루어질 수 있을 것으로 예상된다.

연관성 정보 구조는 새로운 매체의 탄생에 따라 새로운 요소들이 필요로 하게 된다. 연관성 정보 구조가 충분한 확장성을 지니지 못한 경우에는 연관성 정보 구조를 새롭게 구성하기 위해 동일한 연구를 반복적으로 수행할 필요가 있다. 따라서 연관성 정보 구조는 충분한 확장성을 바탕으로 제작되어야 한다. 따라서 본 연구에서는 연관성 정보 구조의 확장성 확보를 위해 매체들을 분석하고 그 특징을 일반화 하는 과정을 거쳐 새로운 매체가 탄생 시에 연관성 정보 구조에 적용하기 위한 방법론을 제시하였다.

연관성 정보 구조는 악성코드를 정의·분류 할 수 있는 기준으로 각 의미를 바탕으로 위험 수준을 정의함으로써 전체적인 악성코드의 위험도를

도출할 수 있다. 또한 연관성 정보 구조는 그 기준들의 조합으로 신종 악성코드나 변종 악성코드의 형태에 대해 예측할 수 있어 악성코드 분석 활동에 큰 도움이 될 것으로 예상된다.

참고문헌

- [1] Bitdefender, "Malware History", 2010.
- [2] Symantec, "Symantec Internet Security Threat Report Trends for 2010", 2010.
- [3] G Data MalwareReport, "Half-yearly report July-December 2010", 2011
- [4] 심재홍, 이석래, "모바일 인터넷 정보보호를 위한 모바일 악성코드 동향 분석", 정보보호학회지, 2009. 11
- [5] Jamie Butler, et al, "R^2: The Exponential Growth of Rootkit Techniques", BlackHat USA, 2006.
- [6] HBGray, "Advanced Persistent Threat", 2010.
- [7] B. Blunden, "The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System", Wordware, 2009.
- [8] 교육사이버안전센터, "교육·연구기관 웹서버 보안취약점 대응가이드", 2009.
- [9] 인터넷침해사고대응지원센터, "ARP Poisoning[Spoofing] 악성코드 감염사고 분석", 2008.
- [10] McAfee, "McAfee Threats Report: Fourth Quate 2010", 2011.
- [11] 카스퍼스키랩, <http://www.kaspersky.com/>, 2011.
- [12] 시만텍, <http://www.symantec.com/index.jsp>, 2011.
- [13] 안철수 연구소, <http://www.ahnlab.com/kr/site/main/main.do>, 2011.
- [14] CARO, <http://www.caro.org/>, 2011.
- [15] N. FitzGerald, "A virus by any other name: The revised CARO naming convention," , 2002.
- [16] M. Heidari, "Malicious codes in depth," Securitydocs.com, 2004
- [17] J. Lin, "On malicious software classification," , pp.368-371, 2008.
- [18] 서희석, 최중섭, and 주필환, "윈도우 악성코드 분류 방법론의 설계," 정보보호학회 논문지, 2009.
- [19] 이병용, 최용수, "Obfuscation 기술의 현황 및 분석과 향후 개발 방향", 보안공학연구논문지, 제 5권 제 3호, 2008.
- [20] 임채태, "봇넷 기술 동향 및 대응 방안", 한국정보보호진흥원, 2008.
- [21] 임채태, 오주형, 정현철, "최신 악성코드 기술동향 및 분석 방안 연구", 정보과학회지, 2010.
- [22] 한국정보화진흥원, "2010 국가정보화백서", 2010.
- [23] 한국정보보호진흥원, "자동화된 SQL Injection 공격을 통한 악성코드 대량 삽입 수법 분석", 2008.
- [24] 한국정보보호진흥원, "악성코드 유형에 따른 자동화 분석 방법론 연구", 2009.
- [25] 안철수 연구소, ASEC 이슈진단, 2011.
- [26] AV-test, <http://www.av-test.org/>, 2011.
- [27] Blue Coat, <http://www.bluecoat.com/>, 2011.

- [28] 보안뉴스, "USB를 통해 전파되는 트로이 목마류 악성코드",
<http://www.boannews.com/media/view.asp?idx=10126&kind=1>, 2008. 6. 5.
- [29] 전자신문, "온라인게임계정 싹쓸이 탈취 악성코드 등장",
<http://www.etnews.com/news/detail.html?id=201009070091>, 2010. 9. 7.
- [30] ZDNet Korea IT/과학, "아이폰 탈옥 도구 위장 '악성코드' 유포",
http://www.zdnet.co.kr/news/news_view.asp?article_id=20100927184256&type=det, 2010. 9. 27.
- [31] 보안뉴스, "[악성코드③] 사건으로 본 전파경로",
<http://www.boannews.com/media/view.asp?idx=10126&kind=1>, 2008. 6. 5.
- [32] 안철수 연구소, "신용카드 한도초과 안내메일로 위장한 악성코드 유포",
<http://blog.ahnlab.com/asec/559?category=5>, 2011. 6. 29.
- [33] 보안뉴스, "[악성코드③] 사건으로 본 전파경로",
<http://www.boannews.com/media/view.asp?idx=10126&kind=1>, 2008. 6. 5.
- [34] 안철수 연구소, "해외에서 발견된 야후 메신저로 전파되는 악성코드",
<http://blog.ahnlab.com/asec/318>, 2010. 5. 11.
- [35] 안철수 연구소, "네이트온 메신저로 전파되는 악성코드 대량 유포",
<http://asec.ahnlab.com/620>, 2011. 10. 27.
- [36] 안철수 연구소, "페이스북 채팅 메시지로 유포되는 악성코드",
<http://blog.ahnlab.com/asec/443>, 2010. 11.19.
- [37] 디지털타임스, "국내서도 페이스북 이용 악성코드 유포",
http://www.dt.co.kr/contents.html?article_no=2011021502011160746003, 2011. 2. 14.
- [38] 전자신문, <장길수의 IT인사이드>(268)무인폭격기(드론), 바이러스에 감염되다,
<http://www.etnews.com/201110090020>, 2011. 10. 09
- [39] 아이뉴스, "악성코드에서 '정부용 확장자' 발견 ...사이버 전쟁 의혹 커져",
http://news.inews24.com/php/news_view.php?g_serial=427934&g_menu=020300, 2009. 07. 10.
- [40] 조희정, "해외의 전자투표 추진 현황 연구", 사회연구 통권 13호, 2007.
- [41] 송지은, 김신호, 정명애, "u-헬스케어 서비스에서의 의료정보보호", 정보보호학회지 제 17권, 2007.
- [42] 남홍순, 이형수, 김재영, "WBAN 응용서비스 동향", 전자통신동향분석 제 24권 제 5호, 2009.
- [43] 최병철, 한승완, 정병호, 김정녀, "지능형 차량 보안 기술 동향", 전자통신동향분석 제 22권 1호, 2007.
- [44] 최병철, 김정녀, "차량 통신 보안 및 프라이버시 주요 이슈", ETRI 정보보호연구본부, 2008. 05.
- [45] 신재용, "사용자 경험 기반 U-헬스케어 서비스", 정보통신산업진흥원 주간기술동향 통권 1462호, 2010.
- [46] 박찬암, "ATM 해킹 위협과 보안 전망", 소프트웨어 보안기술분석팀, 2010.

부록

1. MTAS 개요

MTAS는 Malicious Threat Analysis System의 두문자어로 악성코드 분석가가 분석된 보고서를 바탕으로 연관성 정보 구조에 대입시키는 시스템을 말한다. MTAS의 전체적인 이미지는 다음과 같다.



(그림 A-1) MTAS: 메인 이미지

MTAS는 분석가가 악성코드의 기본 정보인 [감염경로], [실행주체], [공격대상], [공격행위]와 가중치 적용 악성행위인 [자가보호], [전파경로], [목표대상]을 직접 입력하게 된다. 또한 악성코드 분석 후 명명된 악성코드명과 해당 악성코드를 분석한 분석가의 이름을 입력받는다.

MTAS의 기능은 다음과 같다.

1. XML 출력.

MTAS는 연관성 정보 구조를 악성행위별로 체크리스트 테이블을 제공한다. 분석가는 레벨 1 수준을 선택하고 분석된 악성코드 정보에 대하여 체크박스에 체크해 나간다. 모든 악성코드 정보를 입력하면 XML로 출력하는 기능을 갖고 있다. 유연하고 개방적인 표준 기반의 형식을 사용하는 XML을 통하여 연관성 정보 구조의 결과물을 확장성 있게 사용할 수 있도록 하였다.

2. 악성코드의 목적 리포팅

분석가가 연관성 정보 구조에 대한 모든 정보를 입력하면 입력정보를 바탕으로 해당하는 악성코드의 목적을 리포팅한다. 악성코드의 목적 정보는 연관성 정보 구조의 결과물인 XML에 리포팅된다.

3. 변종 예측

연관성 정보 구조의 결과물을 바탕으로 악성코드의 목적과 함께 해당 목적을 위해 수반될 수 있는 다른 악성행위를 나타낸다. MTAS의 결과물로 나타나는 악성행위는 현재 분석된 악성코드에는 수행하지 않는 악성행위지만 향후 해당 악성행위를 함께 수행하는 변종 악성코드가 출현할 수 있음을 나타낸다.

다음은 MTAS의 전체적인 구현물과 각 페이지 화면, 출력되는 XML화면을 나타내는 그림이다.

감염경로	실행주체	공격대상	공격행위	자가보호	전파경로	목표대상
<input type="checkbox"/>	감염경로		감염이용매체			이동식저장매체
<input type="checkbox"/>	감염경로		감염이용매체			모바일디바이스
<input type="checkbox"/>	감염경로		감염이용매체			하드디스크
<input type="checkbox"/>	감염경로		감염이용매체			P2P
<input type="checkbox"/>	감염경로		감염이용매체			웹하드
<input type="checkbox"/>	감염경로		감염이용매체			FTP클라이언트/서버
<input type="checkbox"/>	감염경로		감염이용매체			웹서버
<input type="checkbox"/>	감염경로		감염이용매체			메신저
<input type="checkbox"/>	감염경로		감염이용매체			게시판
<input type="checkbox"/>	감염경로		감염이용매체			블로그
<input type="checkbox"/>	감염경로		감염이용매체			전자우편
<input type="checkbox"/>	감염경로		감염이용매체			ActiveX
<input type="checkbox"/>	감염경로		감염이용매체			BHO
<input type="checkbox"/>	감염경로		감염이용매체			IaaS
<input type="checkbox"/>	감염경로		감염 유형			악성파일직접수신
<input type="checkbox"/>	감염경로		감염 유형			URL접근
<input type="checkbox"/>	감염경로		감염 유형			원격코드실행
<input type="checkbox"/>	감염경로		감염 유형			권한상승(취약점)
<input type="checkbox"/>	감염경로		감염 유형			권한상승(암호)

악성 코드명: 이름: XML로컬:

Designed by Internet Management Technology Lab.

(그림 A-2) MTAS: 감염경로

감염경로	실행주체	공격대상	공격행위	자가보호	전파경로	목표대상
<input type="checkbox"/>	실행주체	사용자				비연속성
<input type="checkbox"/>	실행주체	운영체제				운영체제취약점에 의한 악성코드실행
<input type="checkbox"/>	실행주체	운영체제				운영체제수준에서악성코드자동실행
<input type="checkbox"/>	실행주체	운영체제				운영체제파일실행에따라악성코드실행
<input type="checkbox"/>	실행주체	웹브라우저				웹브라우저보안취약점이용실행
<input type="checkbox"/>	실행주체	웹브라우저				악성코드가포함된ActiveX설치/실행
<input type="checkbox"/>	실행주체	웹브라우저				BHO를이용한악성코드실행
<input type="checkbox"/>	실행주체	응용프로그램				다른응용프로그램보안취약점이용실행
<input type="checkbox"/>	실행주체	응용프로그램				스크립트포함파일열람
<input type="checkbox"/>	실행주체	응용프로그램				작업스케줄러
<input type="checkbox"/>	실행주체	응용프로그램				자동업데이트프로그램
<input type="checkbox"/>	실행주체	응용프로그램				다른악성코드
<input type="checkbox"/>	실행주체					BIOS

악성 코드명: 이름: XML로컬:

Designed by Internet Management Technology Lab.

(그림 A-3) MTAS: 실행주체

감염경로	실행주체	공격대상	공격행위	자가보호	전파경로	목표대상
<input type="checkbox"/>	공격대상	시스템	시스템			시스템장치제어
<input type="checkbox"/>	공격대상	시스템	시스템			시스템서비스제어
<input type="checkbox"/>	공격대상	시스템	시스템			시스템설정제어
<input type="checkbox"/>	공격대상	사용자정보	사용자정보			사용자저장정보
<input type="checkbox"/>	공격대상	사용자정보	사용자정보			사용자입력정보
<input type="checkbox"/>	공격대상	사용자정보	사용자정보			사용자시스템정보
<input type="checkbox"/>	공격대상	사용자정보	사용자정보			사용자시스템사용이력
<input type="checkbox"/>	공격대상	시스템 사용자	시스템 사용자			경제적 손실
<input type="checkbox"/>	공격대상	시스템 사용자	시스템 사용자			이용방해
<input type="checkbox"/>	공격대상	시스템 사용자	시스템 사용자			혼란초래
<input type="checkbox"/>	공격대상	외부시스템	외부시스템			외부시스템서비스
<input type="checkbox"/>	공격대상	외부시스템	외부시스템			로컬네트워크

Designed by Internet Management Technology Lab.

(그림 A-4) MTAS: 공격대상

감염경로	실행주체	공격대상	공격행위	자가보호	전파경로	목표대상
<input type="checkbox"/>	공격행위	네트워크	네트워크			정보유출
<input type="checkbox"/>	공격행위	네트워크	네트워크			서버접속
<input type="checkbox"/>	공격행위	네트워크	네트워크			대량트래픽전송
<input type="checkbox"/>	공격행위	네트워크	네트워크			가로채기
<input type="checkbox"/>	공격행위	네트워크	네트워크			전송지연
<input type="checkbox"/>	공격행위	네트워크	네트워크			스팸
<input type="checkbox"/>	공격행위	시스템	시스템			강제시스템제어
<input type="checkbox"/>	공격행위	시스템	시스템			자원관리
<input type="checkbox"/>	공격행위	시스템	시스템			네트워크설정변경
<input type="checkbox"/>	공격행위	시스템	시스템			시스템설정변경
<input type="checkbox"/>	공격행위	시스템	시스템			디스플레이설정변경
<input type="checkbox"/>	공격행위	파일시스템	파일시스템			파일생성
<input type="checkbox"/>	공격행위	파일시스템	파일시스템			파일파괴
<input type="checkbox"/>	공격행위	파일시스템	파일시스템			파일변조
<input type="checkbox"/>	공격행위	프로세스	프로세스			제어
<input type="checkbox"/>	공격행위	프로세스	프로세스			모니터링
<input type="checkbox"/>	공격행위	입출력장치오작동	입출력장치오작동			미디어제어
<input type="checkbox"/>	공격행위	입출력장치오작동	입출력장치오작동			입력장치제어
<input type="checkbox"/>	공격행위	입출력장치오작동	입출력장치오작동			출력장치제어

Designed by Internet Management Technology Lab.

(그림 A-5) MTAS: 공격행위

감염경로	실행주체	공격대상	공격행위	자가보호	전파경로	목표대상
<input type="checkbox"/>	전파경로		감염이용매체			이동식저장매체
<input type="checkbox"/>	전파경로		감염이용매체			모바일디바이스
<input type="checkbox"/>	전파경로		감염이용매체			하드디스크
<input type="checkbox"/>	전파경로		감염이용매체			P2P
<input type="checkbox"/>	전파경로		감염이용매체			웹하드
<input type="checkbox"/>	전파경로		감염이용매체			FTP클라이언트/서버
<input type="checkbox"/>	전파경로		감염이용매체			웹서버
<input type="checkbox"/>	전파경로		감염이용매체			메신저
<input type="checkbox"/>	전파경로		감염이용매체			게시판
<input type="checkbox"/>	전파경로		감염이용매체			블로그
<input type="checkbox"/>	전파경로		감염이용매체			전자우편
<input type="checkbox"/>	전파경로		감염이용매체			ActiveX
<input type="checkbox"/>	전파경로		감염이용매체			BHO
<input type="checkbox"/>	전파경로		감염이용매체			IaaS

악성코드명: 이름: XML출력

Designed by Internet Management Technology Lab.

(그림 A-6) MTAS: 전파경로

감염경로	실행주체	공격대상	공격행위	자가보호	전파경로	목표대상
<input type="checkbox"/>		자가보호				분석도구탐지
<input type="checkbox"/>		자가보호				루트킷
<input type="checkbox"/>		자가보호				난독화

악성코드명: 이름: XML출력

Designed by Internet Management Technology Lab.

(그림 A-7) MTAS: 자가보호

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
- <malware>
  <name>Worm.TEST.A</name>
  <analyst>홍길동 </analyst>
- <record>
  <level1>공격행위 </level1>
  <level2>네트워크 </level2>
  <level3>정보유출 </level3>
  <score>5</score>
</record>
+ <record>
+ <record>
- <record>
  <level1>공격행위 </level1>
  <level2>입출력장치오작동 </level2>
  <level3>미디어제어 </level3>
  <score>2</score>
</record>
+ <record>
- <record>
  <level1>실행주제 </level1>
  <level2>부동소수점 </level2>
  <level3>BIOS </level3>
  <score>10</score>
</record>
+ <record>
+ <record>
+ <record>
+ <record>
+ <record>
+ <record>
+ <record>
+ <record>
+ <record>
  <dbg>1</dbg>
  <result>193.8</result>
  <type>정보발취형 </type>
+ <mutant>
+ <malware>
```


는 mtas.php로부터 수신한 정보를 바탕으로 악성코드의 위험지수 계산과 목적 도출, 변종 가능 악성행위를 XML로 리포팅하는 역할을 한다.

[표 A-1] mtas.php 소스코드

```
<?
    $level1 = array("감염경로" => 'infect', "실행주체" => 'exec',
"공격대상" => 'victim', "공격행위" => 'attack', "자가보호" => 'protect
ion', "전파경로" => 'potential',"목표대상"=>'who');

    $mysql = mysql_connect("localhost", "root", "apmsetup");
    mysql_select_db("mtas");
    mysql_query("SET NAMES UTF8");
    $test_sql = "SELECT * FROM `mats` ORDER BY `level1`, `Id`
";

    //echo $test_sql;
    $test_res = mysql_query($test_sql, $mysql);
    $cnt = mysql_num_rows($test_res);
    //echo $cnt;

    $group_header = "";
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "ht
tp://www.w3.org/TR/html4/loose.dtd">
<HTML>
  <HEAD>
    <TITLE> MTAS </TITLE>
<META HTTP-EQUIV="content-type" CONTENT="text/html; CHARSET=utf-8
">
<META NAME="Generator" CONTENT="EditPlus">
    <META NAME="Author" CONTENT="">
    <META NAME="Keywords" CONTENT="">
    <META NAME="Description" CONTENT="">
<style type="text/css">
```

```
.tbl_header{
width:960px;
height: 45px;
border: 0px solid black;
position: fixed;
top: 148px;
left: 160px;
}

.tbl_header_entity{
width: 130px;
height: 40px;
display: inline-block;
text-align: center;
padding-top: 10px;
}

#test{
    position:absolute;
    top: 635px;
    left: 350px;
}

body{
    background: url("../MTAS.png");
    background-attachment: fixed;
    background-attachment: scroll;;
    background-repeat: no-repeat;
    background-position:center top;
}

.main_table{
    width: 960px;
    height: 420px;
    position: fixed;
    top: 198px;
```

```

        left: 160px;
        overflow-x: hidden;
        overflow-y: auto;
    }
    .main_table table{

        width: 100%;
        border-collapse: collapse;

    }
    .main_table table tbody tr td{
        border: 1px solid #cccccc;
        text-align:right;
    }
</style>
<script>
    function init(){
        var hide=document.getElementById("mtas_table_infect");

        hide.style.display='none';
        hide=document.getElementById("mtas_table_attack");
        hide.style.display='none';
        hide=document.getElementById("mtas_table_victim");
        hide.style.display='none';
        hide=document.getElementById("mtas_table_exec");
        hide.style.display='none';
        hide=document.getElementById("mtas_table_protection");

        hide.style.display='none';
        hide=document.getElementById("mtas_table_potential");

        hide.style.display='none';
        hide=document.getElementById("mtas_table_who");
        hide.style.display='none';
    }

```

```

function create_xml(){
    sum_chk();
    document.forms[0].mname.value=document.getElementById(
yId("mname").value;
    document.forms[0].analyst.value=document.getElementen
tById("analyst").value;
    document.forms[0].action='create_xml.php';
    document.forms[0].submit();
    document.forms[0].action='';
}
function sum_chk(){
    var i = 0;
    var sel = new Array();

    var table;
    //alert(table.innerHTML);

    table=document.getElementById("mtas_table_infect
").children[0];
    for(i = 0; i < table.children.length; i++){
        if(table.children[i].children[0].children
[0].checked){
            //alert(table.children[i].children
[0].children[0].value);
            sel.push(table.children[i].children
[0].children[0].value);
            //alert(1);
        }
    }
    table=document.getElementById("mtas_table_attack
").children[0];
    for(i = 0; i < table.children.length; i++){
        if(table.children[i].children[0].children
[0].checked){
            //alert(table.children[i].children

```

```

[0].children[0].value);
                                sel.push(table.children[i].children
[0].children[0].value);
                                //alert(1);
                                }
                                }
                                table=document.getElementById("mtas_table_exec").c
hildren[0];
                                for(i = 0; i < table.children.length; i++){
                                    if(table.children[i].children[0].children
[0].checked){
                                        //alert(table.children[i].children
[0].children[0].value);
                                        sel.push(table.children[i].children
[0].children[0].value);
                                        //alert(1);
                                    }
                                }
                                table=document.getElementById("mtas_table_victim
").children[0];
                                for(i = 0; i < table.children.length; i++){
                                    if(table.children[i].children[0].children
[0].checked){
                                        //alert(table.children[i].children
[0].children[0].value);
                                        sel.push(table.children[i].children
[0].children[0].value);
                                        //alert(1);
                                    }
                                }
                                table=document.getElementById("mtas_table_potentia
l").children[0];
                                for(i = 0; i < table.children.length; i++){
                                    if(table.children[i].children[0].children
[0].checked){

```

```

//alert(table.children[i].children
[0].children[0].value);

sel.push(table.children[i].children
[0].children[0].value);

//alert(1);

}

}

table=document.getElementById("mtas_table_protecti
on").children[0];

for(i = 0; i < table.children.length; i++){
    if(table.children[i].children[0].children
[0].checked){

//alert(table.children[i].children
[0].children[0].value);

sel.push(table.children[i].children
[0].children[0].value);

//alert(1);

}

}

table=document.getElementById("mtas_table_who").ch
ildren[0];

for(i = 0; i < table.children.length; i++){
    if(table.children[i].children[0].children
[0].checked){

//alert(table.children[i].children
[0].children[0].value);

sel.push(table.children[i].children
[0].children[0].value);

//alert(1);

}

}

document.forms[0].sel_list.value = sel.join("/");
}

function view_group(str){
    init();

```

```

        var test = document.getElementById("mtas_table_"+str);

        test.style.display='block';

    }
</script>
</HEAD>

<BODY onload="init()">
    <div class="tbl_header">
        <span class="tbl_header_entity"><input type='button' value="감염경로" onclick="view_group('infect');"/></span>
        <span class="tbl_header_entity"><input type='button' value="실행주체" onclick="view_group('exec');"/></span>
        <span class="tbl_header_entity"><input type='button' value="공격대상" onclick="view_group('victim');"/></span>
        <span class="tbl_header_entity"><input type='button' value="공격행위" onclick="view_group('attack');"/></span>
        <span class="tbl_header_entity"><input type='button' value="자가보호" onclick="view_group('protection');"/></span>
        <span class="tbl_header_entity"><input type='button' value="전파경로" onclick="view_group('potential');"/></span>
        <span class="tbl_header_entity"><input type='button' value="목표대상" onclick="view_group('who');"/></span>
    </div>
    <div class="main_table">
        <form name="mats" method="post">
            <input type="hidden" name="group" />
            <input type="hidden" name="type" />
            <input type="hidden" name="sel_list" />
            <input type="hidden" name='analyst'>
            <input type="hidden" name='mname'>
        </form>
        <?
        for($i = 0; $i < $cnt; $i++){
            $data = mysql_fetch_assoc($test_res);
            if($group_header == ""){

```



```

                                echo "<table id=\"mtas_table_\".$level1[$data[\"level1\"]].\">";
                                echo "<tbody>";
                                $group_header = $data["level1"];
                                }else if($group_header != $data["level1"]){
                                echo "</tbody>";
                                echo "</table>";
                                echo "<table id=\"mtas_table_\".$level1[$data[\"level1\"]].\">";
                                echo "<tbody>";
                                $group_header = $data["level1"];
                                }
                                echo "<tr>\n";
                                echo "<td ><input type='checkbox' name='chkbox' value='\".$data[\"Id\"].\"' /></td>\n";
                                echo "<td ><font face=sans>\".$data[\"level1\"].\"</font></td>\n";
                                echo "<td >\".$data[\"level2\"].\"</td>\n";
                                echo "<td >\".$data[\"level3\"].\"</td>\n";
                                echo "</tr>\n";
                                }
?>

                                </tbody>
                                </table>
                                </form>
                                </div>
                                <div id="test">

                                <div>
                                악성코드명:<input type="text" id='mname'>

                                이름:<input type="text" id='analyst'>

```

```

                                <input type="button" value="XML출력" onclick
="create_xml();" >
                                </div>
                                </div>
                                </BODY>
                                </HTML>

```

[표 A-2] create_xml.php 소스코드

```

<?
    header("Content-type: text/xml; charset=utf-8");
    header("Content-Disposition:attachment; filename=test.xml;");

    $mysql = mysql_connect("localhost", "root", "apmsetup");
    mysql_select_db("mtas");
    mysql_query("SET NAMES UTF8");
    $test_sql = "SELECT * FROM `mats` WHERE `level1` =
'".$level1[$_POST["type"]]."'";

    $test_res = mysql_query($test_sql, $mysql);
    $cnt = mysql_num_rows($test_res);

    //print_r($_POST);
    $test_sql = "SELECT * FROM `mats` WHERE `Id` =
'".$_str_replace("/", " OR `Id` = ", $_POST["sel_list"])."'";
    //echo $test_sql;

    // variables
    $infect_sum=0; //감염경로 합
    $attack_sum=0; //공격행위 합
    $victim_sum=0; //공격대상 합

```

```

$exec_sum=0; //실행주체 합
$potential=1; //잠재위험 기본 점수
$propagation=0; //전파경로 개수에 따른 점수
$protection=0; //자가보호 가중치
$who=1; //목표대상

// mutant variables
$num_info=0;
$num_remote=0;
$num_money=0;
$num_destory=0;
$num_module=0;
$num_fear=0;
$num_potential=0;

$result=mysql_query($test_sql,$mysql);
$cnt=mysql_num_rows($result);

echo "<?xml version=\"1.0\" encoding=\"utf-8\"
standalone=\"yes\"?>";

echo "<malware>";
echo "<name>".$_POST['mname']. "</name>";
echo "<analyst>".$_POST['analyst']. "</analyst>";

for($i=0;$i<$cnt;$i++)
{

    $data = mysql_fetch_assoc($result);

    // xml print
    echo "<record>";
    echo "<level1>".$data['level1']. "</level1>";
    echo "<level2>".$data['level2']. "</level2>";

```

```

echo "<level3>".$data['level3']. "</level3>";
echo "<score>".$data['score']. "</score>";
echo "</record>";

// 변종 계산
$key = $data['Id'];
$mutant_search_query = "select * from mutant where
info=$key or remote=$key or money=$key or destroy=$key or
module=$key or fear=$key or potential=$key";

$mutant_result=mysql_query($mutant_search_query,$mysql);
$mcnt=mysql_num_rows($mutant_result);

for($j=0;$j<$mcnt;$j++)
{
    $mdata=mysql_fetch_assoc($mutant_result);

    if($mdata['info']==$key)
        $num_info++;
    if($mdata['remote']==$key)
        $num_remote++;
    if($mdata['money']==$key)
        $num_money++;
    if($mdata['destroy']==$key)
        $num_destroy++;
    if($mdata['module']==$key)
        $num_module++;
    if($mdata['fear']==$key)
        $num_fear++;
    if($mdata['potential']==$key)
        $num_potential++;
}

// 구조 점수
if($data['level1']=="감염경로")

```

```

        $infect_sum+=$data['score'];
    if($data['level1']=="실행주체")
        $exec_sum+=$data['score'];
    if($data['level1']=="공격대상")
        $victim_sum+=$data['score'];
    if($data['level1']=="공격행위")
        $attack_sum+=$data['score'];

//전파 경로 개수

if($data['level1']=="전파경로")
{
    $propagation+=1;
    echo "<dbg>$propagation</dbg>";
}

//잠재위험 항목 계산
if($data['level3']=="네트워크설정변경")
    $potential+=0.075;
if($data['level3']=="시스템설정변경")
    $potential+=0.075;
if($data['level3']=="서버접속")
    $potential+=0.075;
if($data['level3']=="강제시스템제어")
    $potential+=0.075;

// 자가보호 개수 저장
if($data['level2']=="분석도구탐지")
    $protection+=1;
if($data['level2']=="루트킷")
    $protection+=1;
if($data['level2']=="난독화")
    $protection+=1;

```

```

// 목표대상 -> 복수 경우 높은 목표대상 가중치 적용
if($data['level3']=="군사시설")
{
    $who=2;
}
if($data['level3']=="사회기반시설")
{
    if($who<1.8)
        $who=1.8;
}
if($data['level3']=="행정기관")
{
    if($who<1.5)
        $who=1.5;
}
if($data['level3']=="기업")
{
    if($who<1.3)
        $who=1.3;
}
if($data['level3']=="개인")
{
    $who=1;
}

}

$infect_sum=$infect_sum*1.5;
$exec_sum=$exec_sum*1.5;
$victim_sum=$victim_sum*3;
$attack_sum=$attack_sum*4;

```

/ / e c h o

```

"<dat>$infect_sum,$exec_sum,$victim_sum,$attack_sum</dat>";

// 전파경로 개수 에 따른 점수 계산
if($propagation==0)
    $propagation= 0.8;
else if($propagation>0 && $propagation <3)
    $propagation=1;
else if($propagation>2 && $propagation <6)
    $propagation=1.2;
else if($propagation>5)
    $propagation=1.5;

// 목적그룹 구하기
$malware_type="";
$max =
max($num_info,$num_remote,$num_destroy,$num_module,$num_fear,$num
_potential);
//echo $num_info;
//echo $num_destroy;

//echo"<da>$num_info,$num_remote,$num_destroy,$num_module,$num_fe
ar,$num_potential</da>";
if($max==$num_info)
{
    $malware_type=$malware_type."정보탈취형/";
    //echo $malware_type;
}
if($max==$num_remote)
    $malware_type=$malware_type."원격제어형/";
if($max==$num_money)
    $malware_type=$malware_type."과금유발형/";
if($max==$num_destroy)
{
    $malware_type=$malware_type."시스템파괴형/";
}

```

```

        //echo $malware_type;
    }
    if ($max==$num_module)
        $malware_type=$malware_type." 모듈형/";
    if ($max==$num_fear)
        $malware_type=$malware_type." 혼란야기형/";
    if ($max==$num_potential)
        $malware_type=$malware_type." 유해가능형/";

    //echo "<da>$propagation, $protection</da>";

$alpha=floor((10*sqrt($propagation*((1.1)^$protection)))/10;

    //echo "<da>$alpha</da>";
    $result
=
($infect_sum+$who*($attack_sum*$potential+$victim_sum)+$exec_sum)
*$alpha;
    echo "<result>$result</result>";
    echo "<type>$malware_type</type>";

    //변종 생성

    $pieces = explode("/", $malware_type);
    //echo    "<mutant>$malware_type    -->    $pieces[0]    -->
$pieces[1]</mutant>";
    echo "<mutant>";
    for ($k=0;$k<7;$k++)
    {
        if ($pieces[$k]=="정보탈취형")
        {
            echo "<type>";
            echo "<name>정보탈취형</name>";
            $test_sql = "SELECT info FROM mutant WHERE
info <> ".str_replace("/", " and info <>", $_POST["sel_list"]).";";
            //echo "<test>$test_sql</test>";

```



```

        $result=mysql_query($test_sql,$mysql);
        //echo $result;
        $cnt = mysql_num_rows($result);

        for($l=0;$l<$cnt;$l++)
        {
            $mdata=mysql_fetch_assoc($result);
            $key=$mdata['info'];
            $query = "SELECT * from mats where
Id = $key";

            $get = mysql_query($query, $mysql);
            //echo $get;
            $get1 = mysql_fetch_assoc($get);
            $get2=$get1['level1'];
            echo "<record>";
            echo "<level1>$get2</level1>";
            $get2=$get1['level2'];
            echo "<level2>$get2</level2>";
            $get2=$get1['level3'];
            echo "<level3>$get2</level3>";
            echo "</record>";

        }

        echo "</type>";
    }

    if($pieces[$k]=="과금유발형")
    {
        echo "<type>";
        echo "<name>과금유발형</name>";
        $test_sql = "SELECT money FROM mutant WHERE
money      <>      ".str_replace("/",      "      and      money      <> ",
$_POST["sel_list"]).";

```

```

//echo "<test>$test_sql</test>";

$result=mysql_query($test_sql,$mysql);
//echo $result;
$cnt = mysql_num_rows($result);

for($l=0;$l<$cnt;$l++)
{
    $mdata=mysql_fetch_assoc($result);
    $key=$mdata['money'];
    $query = "SELECT * from mats where
money = $key";

    $get = mysql_query($query, $mysql);
    //echo $get;
    $get1 = mysql_fetch_assoc($get);
    $get2=$get1['level1'];
    echo "<record>";
    echo "<level1>$get2</level1>";
    $get2=$get1['level2'];
    echo "<level2>$get2</level2>";
    $get2=$get1['level3'];
    echo "<level3>$get2</level3>";
    echo "</record>";
}

echo "</type>";
}

if($pieces[$k]=="시스템 파괴형")
{
    echo "<type>";
    echo "<name>시스템 파괴형</name>";
    $test_sql = "SELECT destroy FROM mutant
WHERE destroy <> ".str_replace("/", " " and destroy <> ",

```

```

$_POST["sel_list"])." ";

        //echo "<test>$test_sql</test>";

        $result=mysql_query($test_sql,$mysql);
        //echo $result;
        $cnt = mysql_num_rows($result);

        for($l=0;$l<$cnt;$l++)
        {
            $mdata=mysql_fetch_assoc($result);
            $key=$mdata['destroy'];
            $query = "SELECT * from mats where
Id = $key";

            $get = mysql_query($query, $mysql);
            //echo $get;
            $get1 = mysql_fetch_assoc($get);
            $get2=$get1['level1'];
            echo "<record>";
            echo "<level1>$get2</level1>";
            $get2=$get1['level2'];
            echo "<level2>$get2</level2>";
            $get2=$get1['level3'];
            echo "<level3>$get2</level3>";
            echo "</record>";

        }

        echo "</type>";
    }

    if($pieces[$k]=="모듈형")
    {
        echo "<type>";
        echo "<name>모듈형</name>";
        $test_sql = "SELECT module FROM mutant WHERE

```

```

module    <>    ".str_replace("/",    "    and    module    <>",
$_POST["sel_list"])."";

        //echo "<test>$test_sql</test>";

        $result=mysql_query($test_sql,$mysql);
        //echo $result;
        $cnt = mysql_num_rows($result);

        for($l=0;$l<$cnt;$l++)
        {
                $mdata=mysql_fetch_assoc($result);
                $key=$mdata['module'];
                $query = "SELECT * from mats where
Id = $key";

                $get = mysql_query($query, $mysql);
                //echo $get;
                $get1 = mysql_fetch_assoc($get);
                $get2=$get1['level1'];
                echo "<record>";
                echo "<level1>$get2</level1>";
                $get2=$get1['level2'];
                echo "<level2>$get2</level2>";
                $get2=$get1['level3'];
                echo "<level3>$get2</level3>";
                echo "</record>";

        }

        echo "</type>";
}

if($pieces[$k]=="원격제어형")
{
        echo "<type>";
        echo "<name>원격제어형</name>";
}

```

```

        $test_sql = "SELECT remote FROM mutant WHERE
remote    <>    ".str_replace("/", "    and    remote    <>",
$_POST["sel_list"]).";

        //echo "<test>$test_sql</test>";

        $result=mysql_query($test_sql,$mysql);
        //echo $result;
        $cnt = mysql_num_rows($result);

        for($l=0;$l<$cnt;$l++)
        {
            $mdata=mysql_fetch_assoc($result);
            $key=$mdata['remote'];
            $query = "SELECT * from mats where
Id = $key";

            $get = mysql_query($query, $mysql);
            //echo $get;
            $get1 = mysql_fetch_assoc($get);
            $get2=$get1['level1'];
            echo "<record>";
            echo "<level1>$get2</level1>";
            $get2=$get1['level2'];
            echo "<level2>$get2</level2>";
            $get2=$get1['level3'];
            echo "<level3>$get2</level3>";
            echo "</record>";

        }

        echo "</type>";
    }

    if($pieces[$k]=="유해가능형")
    {
        echo "<type>";
    }

```

```

        echo "<name>유해가능형</name>";
        $test_sql = "SELECT potential FROM mutant
WHERE potential <> ".str_replace("/", " and potential <>",
$_POST["sel_list"])."";

        //echo "<test>$test_sql</test>";

        $result=mysql_query($test_sql,$mysql);
        //echo $result;
        $cnt = mysql_num_rows($result);

        for($l=0;$l<$cnt;$l++)
        {
            $mdata=mysql_fetch_assoc($result);
            $key=$mdata['potential'];
            $query = "SELECT * from mats where
Id = $key";

            $get = mysql_query($query, $mysql);
            //echo $get;
            $get1 = mysql_fetch_assoc($get);
            $get2=$get1['level1'];
            echo "<record>";
            echo "<level1>$get2</level1>";
            $get2=$get1['level2'];
            echo "<level2>$get2</level2>";
            $get2=$get1['level3'];
            echo "<level3>$get2</level3>";
            echo "</record>";
        }

        echo "</type>";
    }
    if($pieces[$k]=="혼란야기형")
    {
        echo "<type>";
    }

```

```

        echo "<name>혼란야기형</name>";

        $test_sql = "SELECT fear FROM mutant WHERE
fear <> ".str_replace("/", " and fear <>", $_POST["sel_list"]).";
        //echo "<test>$test_sql</test>";

        $result=mysql_query($test_sql,$mysql);
        //echo $result;
        $cnt = mysql_num_rows($result);

        for($l=0;$l<$cnt;$l++)
        {
            $mdata=mysql_fetch_assoc($result);
            $key=$mdata['fear'];
            $query = "SELECT * from mats where
Id = $key";

            $get = mysql_query($query, $mysql);
            //echo $get;
            $get1 = mysql_fetch_assoc($get);
            $get2=$get1['level1'];
            echo "<record>";
            echo "<level1>$get2</level1>";
            $get2=$get1['level2'];
            echo "<level2>$get2</level2>";
            $get2=$get1['level3'];
            echo "<level3>$get2</level3>";
            echo "</record>";

        }

        echo "</type>";

    }

}
echo "</mutant>";
echo "</malware>";

?>

```

악성코드 유사 및 변종유형 예측방법 연구

인 쇄 : 2011 년 11 월

발 행 : 2011 년 11 월

발행인 : 서 종 렬

발행처 : 한국인터넷진흥원(KISA, Korea Internet&Security Agency)

서울시 송파구 가락동 79-3 대동빌딩

Tel: (02) 4055-114

인쇄처 : R&B 기획

Tel: (031) 293-6179

<비매품>

1. 본 보고서는 방송통신위원회의 출연금으로 수행한 정보보호 강화 사업의 결과입니다.
2. 본 보고서의 내용을 발표할 때에는 반드시 한국인터넷진흥원 정보보호 강화사업의 결과임을 밝혀야 합니다.
3. 본 보고서의 판권은 한국인터넷진흥원이 소유하고 있으며, 당 진흥원의 허가 없이 무단 전재 및 복사를 금합니다.