



악성코드 자동화 분석을 통한 인텔리전스

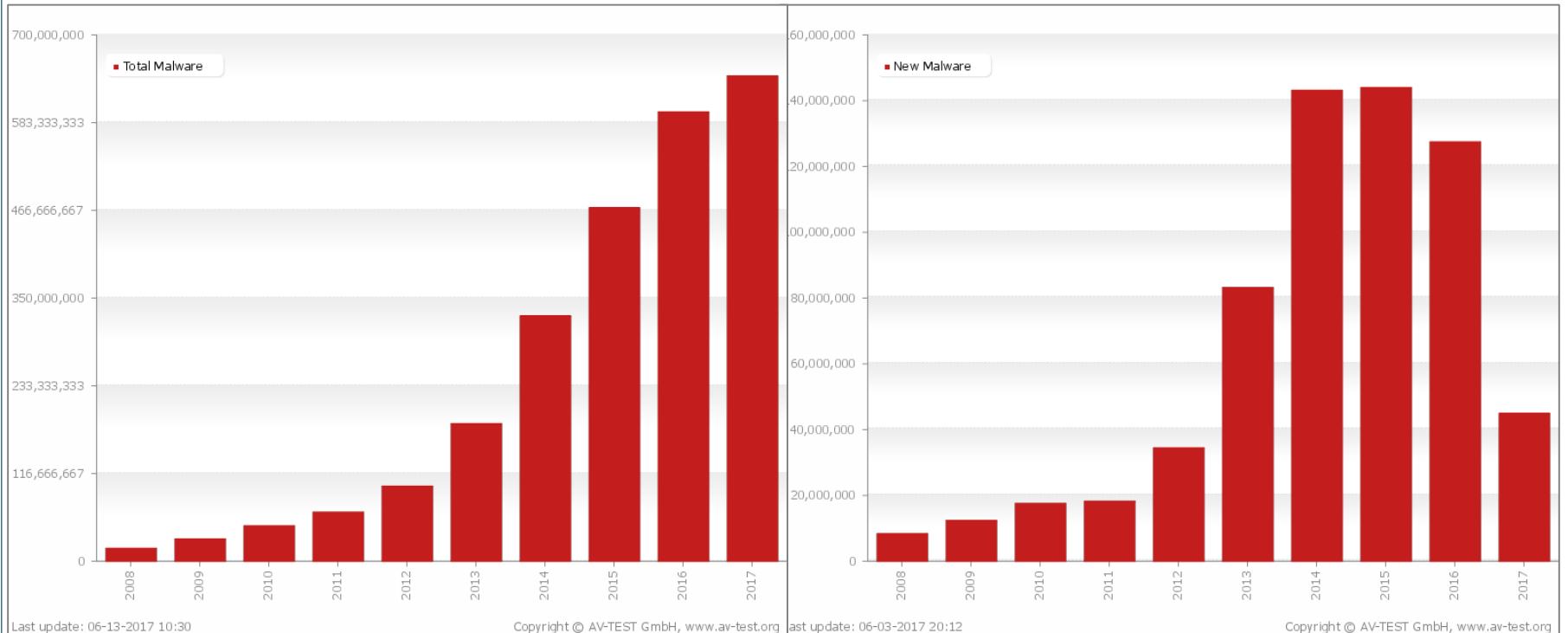
목차

- 악성코드 및 분석 방법론 동향
- 오픈소스 라이선스
- 하이퍼바이저
- 허니팟 프로젝트
- 쿡쿡 샌드박스
- 다른 바이너리 자동화 분석 인텔리전스 서비스
- 바이너리 인텔리전스의 한계
- 결론



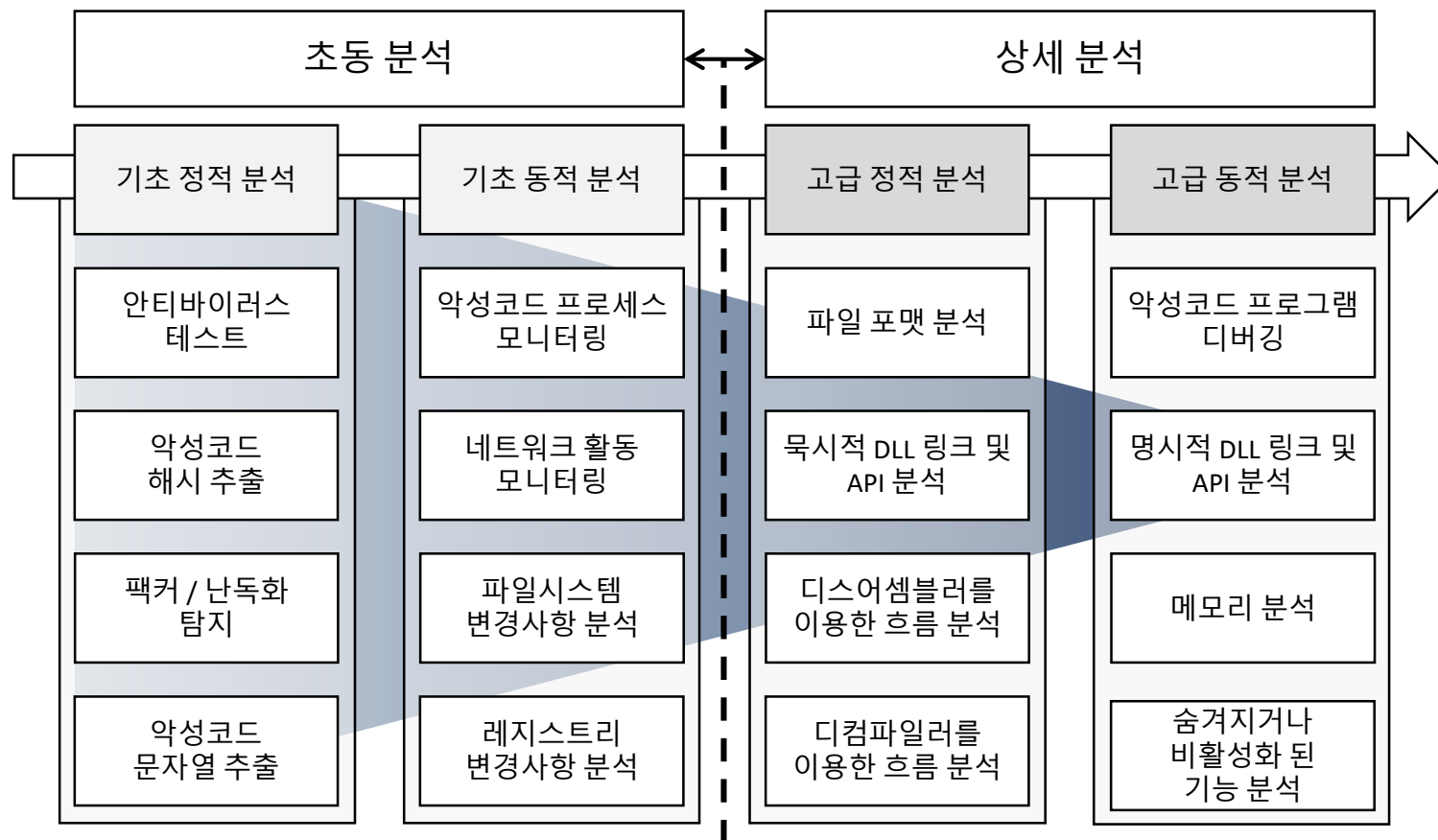
● 악성코드 동향

➔ 악성코드 발생의 고도화



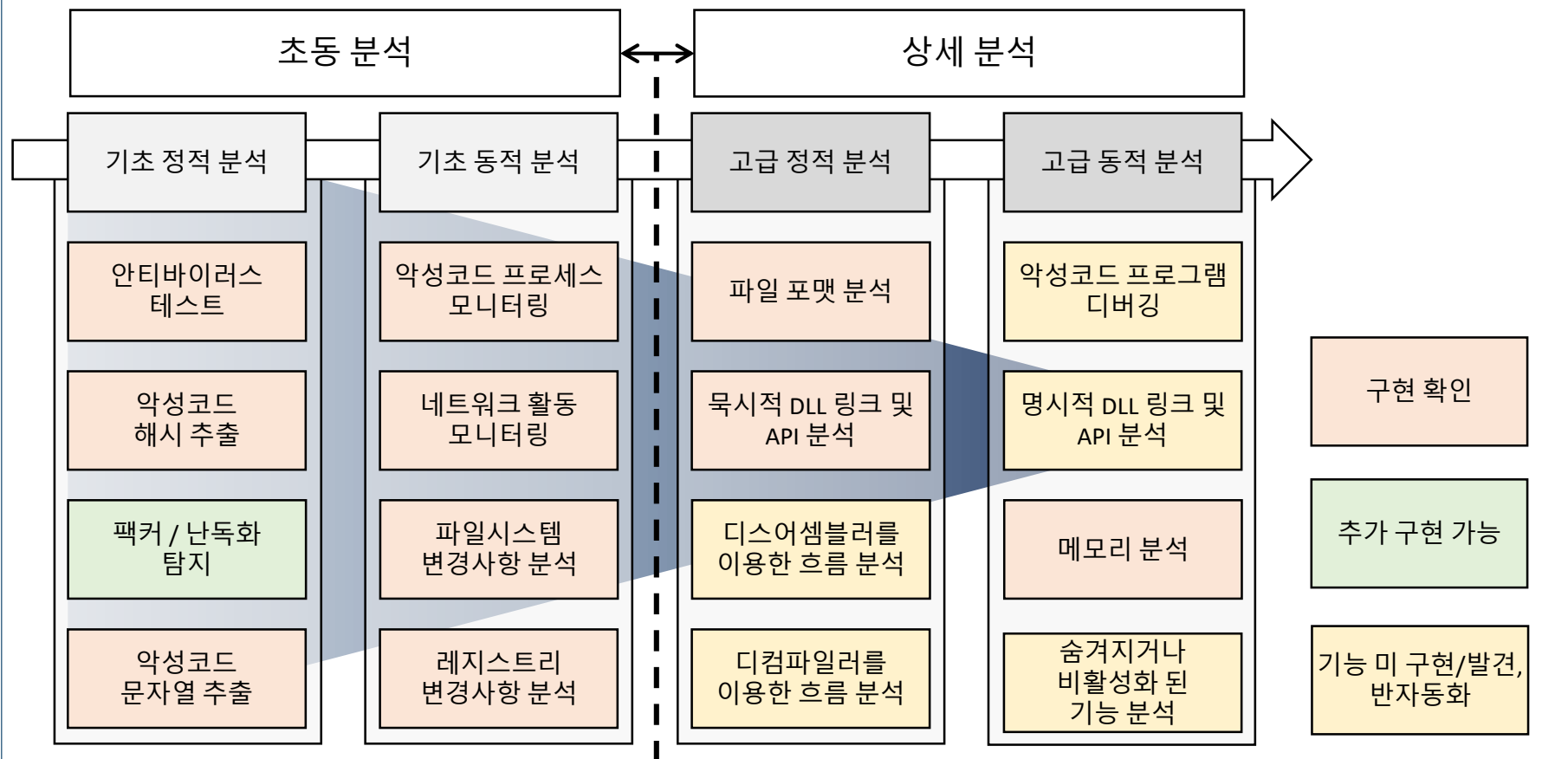
● 악성코드 분석 방법론 동향

→ 악성코드 분석 방법론 - 기존



● 악성코드 분석 방법론 동향

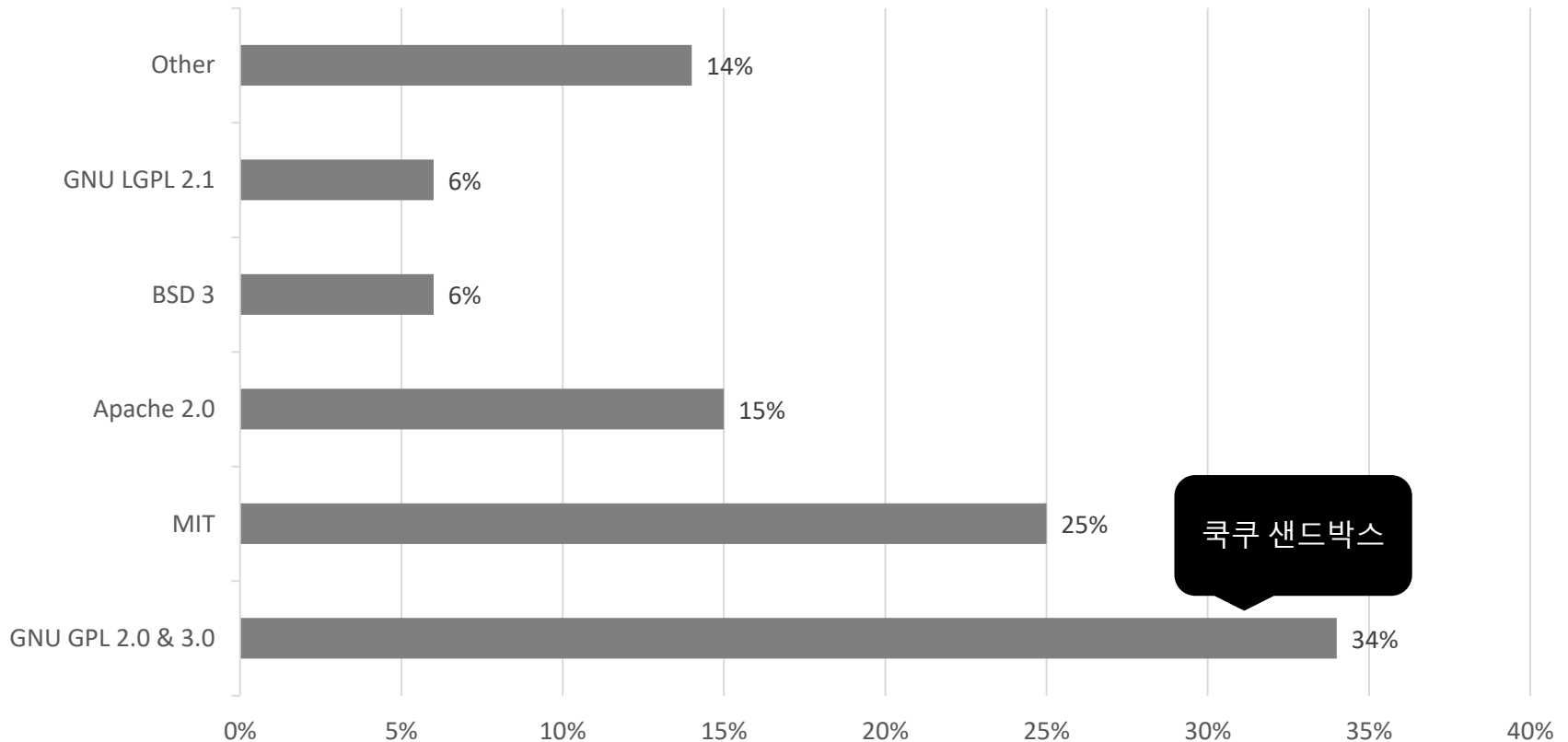
➔ 악성코드 분석 방법론 - 자동화



● 오픈소스 라이선스

→ 라이선스 동향

2016 오픈소스 라이선스 Top 5



● 오픈소스 라이선스

→ 오픈소스 라이선스 비교

라이선스	링크	배포	수정	특허	비공개 사용	라이선스 재설정	상표
GNU GPL	자기 자신	비상업적 가능	비상업적 가능	가능	가능	비상업적 가능	가능
MIT	가능	가능	가능	불가능	가능	가능	불가능
Apache	가능	가능	가능	가능	가능	가능	불가능
BSD	가능	가능	가능	불가능	가능	가능	불가능
GNU LGPL	제한	비상업적 가능	비상업적 가능	가능	가능	비상업적 가능	가능

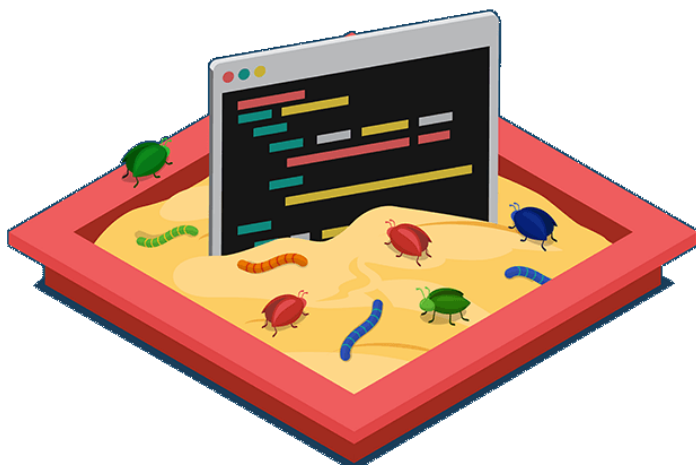
● 하이퍼바이저

→ 샌드박스

- ▶ 컴퓨터 보안에서 샌드박스는 실행되는 프로그램을 분리하기 위한 보안 메커니즘
- ▶ 디스크, 메모리 공간에 엄격한 자원 제어를 통해 서비스를 제공
- ▶ 보통 가상화를 통해 구현

→ 가상머신(하이퍼바이저)

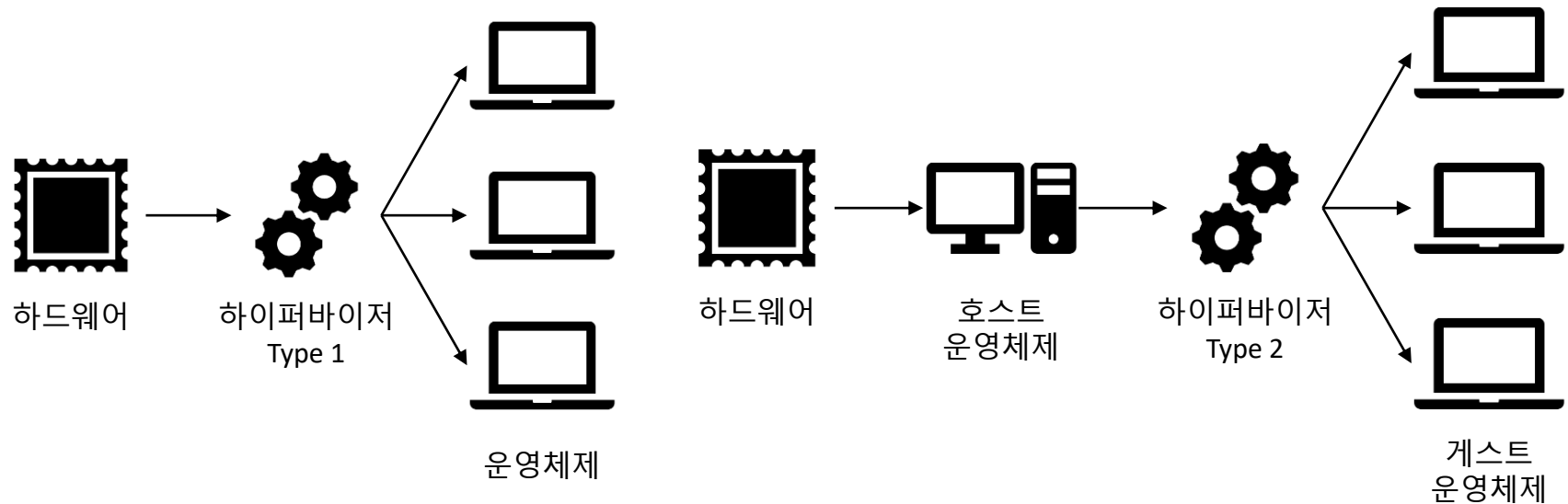
- ▶ 샌드박스는 가상머신으로 많이 구현
- ▶ 가상머신을 구현하도록 도와주는 소프트웨어를 하이퍼바이저로 불림



● 하이퍼바이저

→ 가상머신(하이퍼바이저)

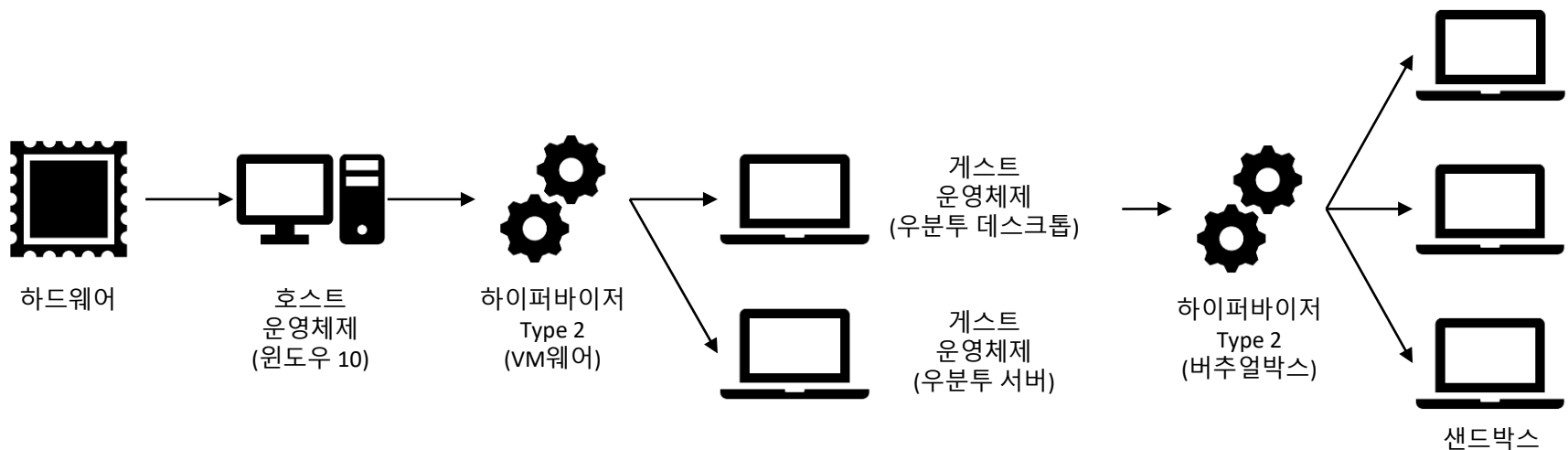
▶ 하이퍼바이저는 Type 1과 Type 2 방식이 있음



● 하이퍼바이저

→ 가상머신(하이퍼바이저)

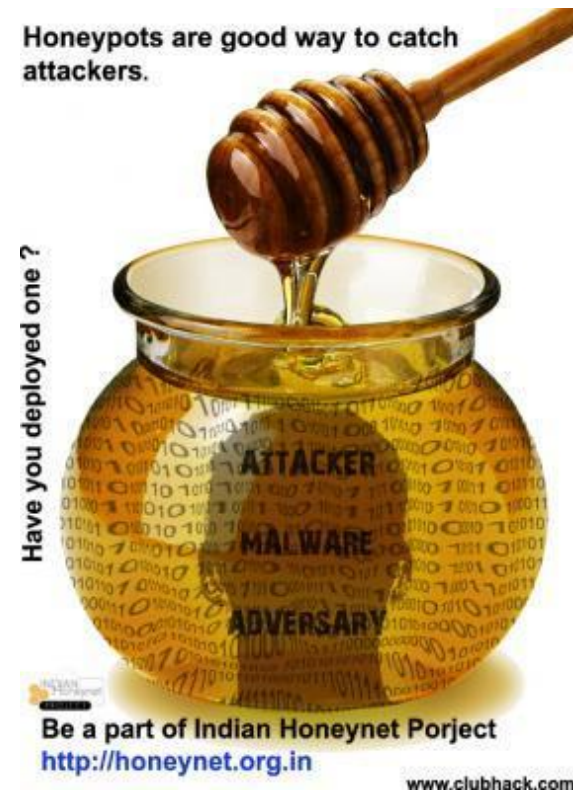
- ▶ 쿡쿠 샌드박스는 물리적 시스템으로 구축하려면 많은 비용이 필요함
- ▶ 호스트 운영체제를 리눅스 계열로 쓰기 힘들
 - Type 1을 고려할 수 없으며, Type 1과 Type 2 호환성이 좋지 않음 (예, 하이퍼v)
- ▶ 이를 해결하기 위해 중첩 가상화(Nested Virtualization) 기술을 이용 - Intel VT-x 지원(o)
 - 이 기술은 Type 2 중에 VM웨어가 잘 지원함



● 허니팟 프로젝트

→ 허니팟 프로젝트 요건

- ▶ 쉽게 해커에게 노출
 - 취약한 소프트웨어들로 구성,
 - 다양한 상황에 맞는 구성 요소
- ▶ 시스템에 접근하는 모든 정보를 감시 및 기록 가능
- ▶ 프로젝트
 - 프로젝트 단위로 운영 (허니넷, 허니팟)
- ▶ 클라이언트 허니팟
 - 허니클라이언트(HoneyClient)로 불리기도 함
 - 클라이언트 단위에서 동작
 - 주로 인터넷 보안에 특화되어 있음



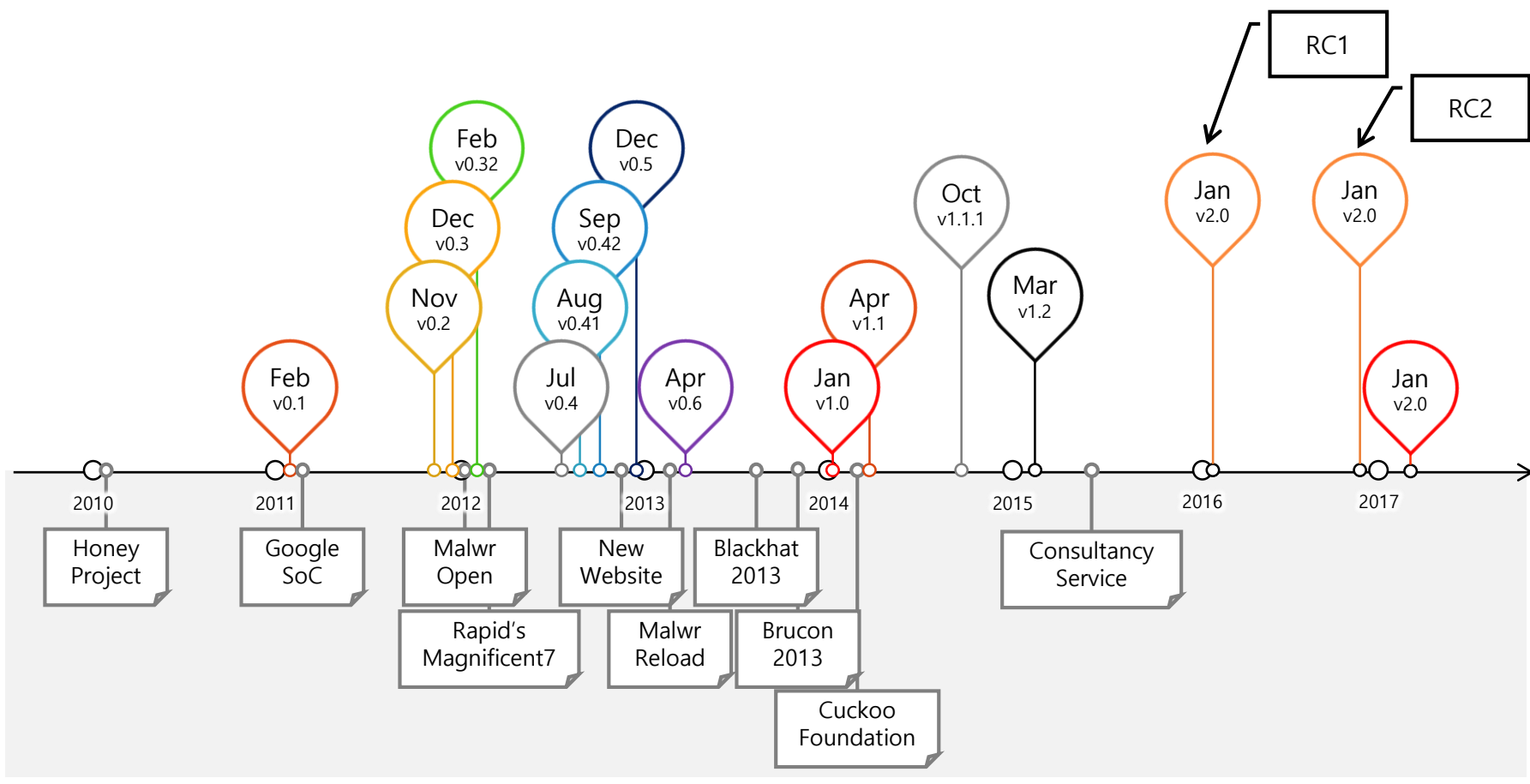
● 허니팟 프로젝트

→ 클라이언트 허니팟 종류

- ▶ 높은 상호작용 클라이언트 허니팟
 - 실제 환경과 비슷한 가상 환경을 구축하여 운영
 - 알려지지 않은 공격을 탐지 할 수 있음
 - 성능 편차가 심함, 라이선스 비용이 큼, 구축 시간의 소모가 큼
- ▶ 낮은 상호작용 클라이언트 허니팟
 - 시뮬레이션 형태로 운영
 - 탐지 및 차단이 빠름, 구축 시간이 빠름
 - 알려지지 않은 공격을 탐지할 수 없음
- ▶ 하이브리드 클라이언트 허니팟
 - 높은, 낮은 상호작용의 단점을 극복하기 위해 장점들만을 이용

● 쿡쿠 샌드박스

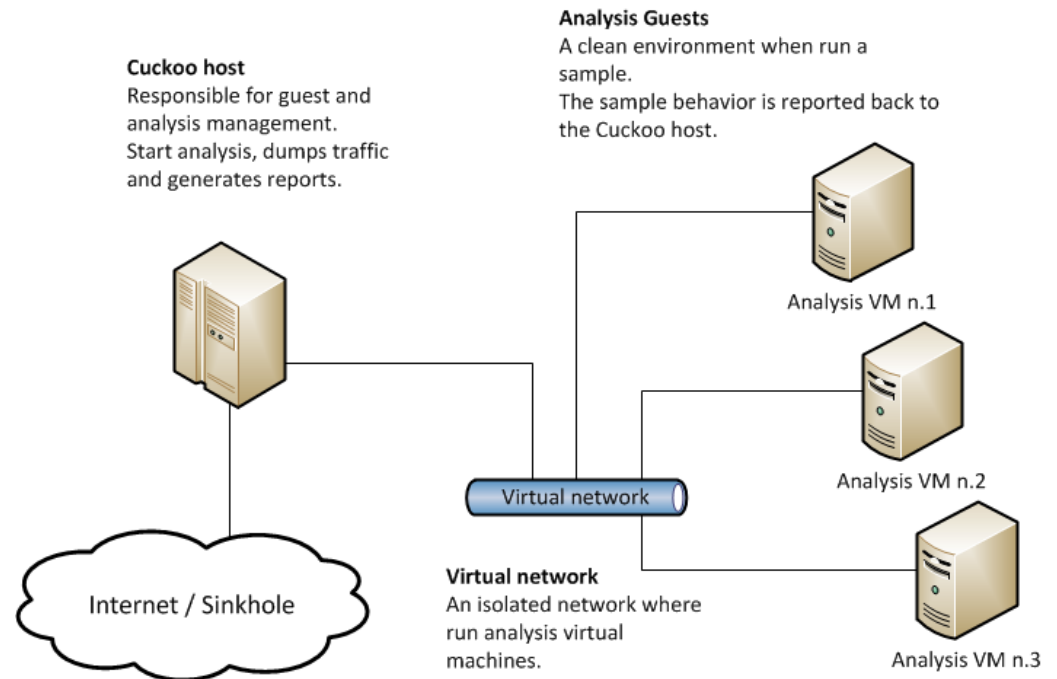
→ 발전 역사



● 쿡쿠 샌드박스

→ 분석 가능한 파일

- ▶ PE+, MSI, DLL files
- ▶ PDF documents
- ▶ Microsoft Office documents
- ▶ URLs and HTML files
- ▶ PHP scripts, CPL files
- ▶ Visual Basic (VB) scripts
- ▶ ZIP files, Java JAR, applet
- ▶ Python files
- ▶ 기타 등등



악성코드 자동화 분석을 통한 인텔리전스

● 쿡쿠 샌드박스

➔ 동작 원리



● 쿡쿠 샌드박스

→ 장점

- ▶ 소스코드가 공개
- ▶ 파이썬으로 제작되어 접합성이 뛰어남
- ▶ CLI와 GUI모드 둘 다 제공하고 있기 때문에, 독립적인 시스템으로 운영이 가능하고, 기능을 추가 혹은 변형 가능
- ▶ api.py 라는 유틸리티로 REST API를 제공받아 어떤 언어를 사용하든 Cuckoo의 기능을 붙일 수 있도록 제공
- ▶ 현재까지 지속적인 버그 패치 및 버전 업그레이드를 제공
- ▶ 사용자와 지속적인 커뮤니케이션을 유지하여 새로운 방향성, 버그 패치, 시그니처 업데이트 등 제공
- ▶ 샌드박스를 다수 개로 구성할 수 있으며, 각각의 시스템 환경을 다르게 할 수 있음



● 쿡쿠 샌드박스

→ 단점

- ▶ 핵심 기능은 문서화가 잘되어 있으나 자세하지 않음
- ▶ 부가 기능은 문서화가 되어 있지 않음
- ▶ 버전 명시가 안되어 있어 버전에 따라 달라지는 운영을 파악해야함 - 개선
- ▶ 문제가 발생하면 레포팅하거나 직접 소스코드를 수정
 - 서비스 제작자가 문제 해결 의지는 있으나 제작자 선택사항임
- ▶ 문제가 되는 서비스나 의미 없는 추가 기능은 동의 없이 사라질 수 있음



● 쿡쿠 샌드박스

→ 설치 환경

▶ Cuckoo Core Server

- Ubuntu 16.04 LTS Desktop 64 bit
- CPU 4 Core, Memory 4 GB

▶ Cuckoo DB Server

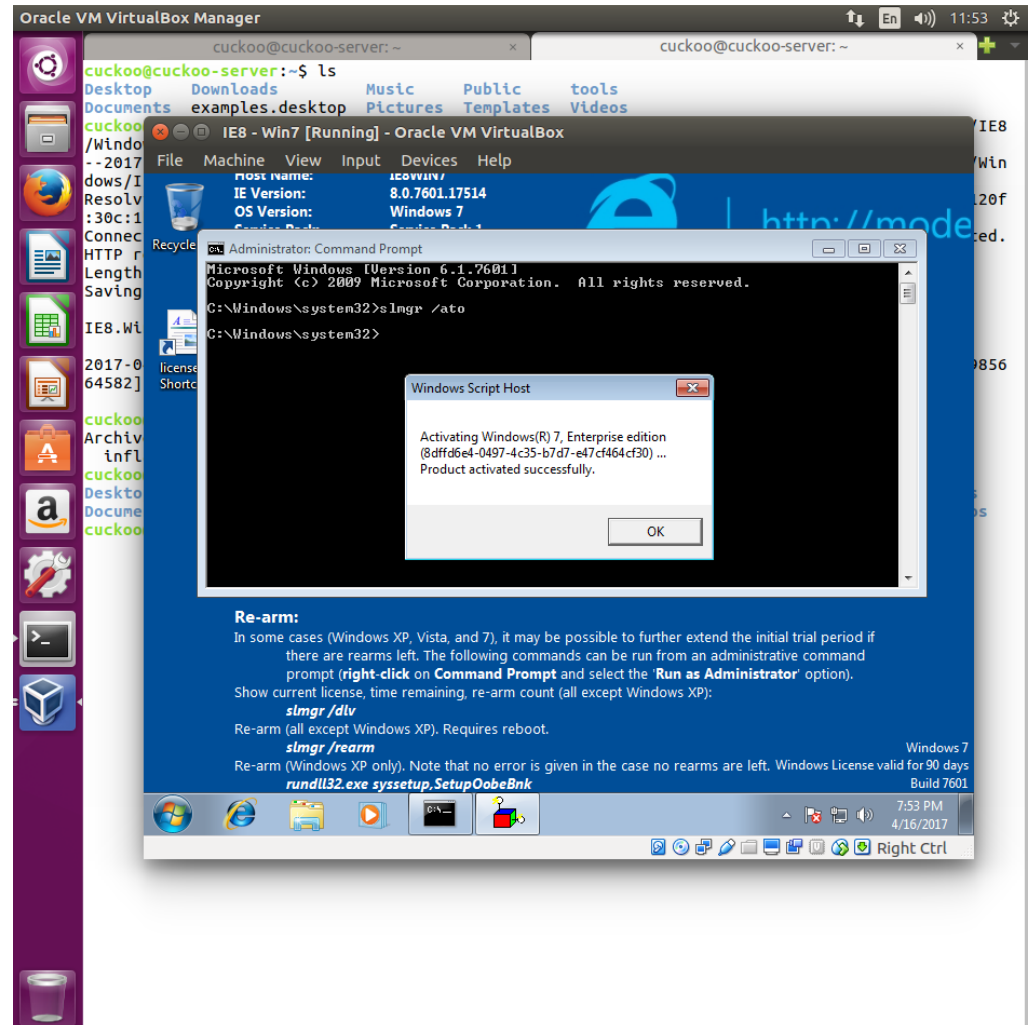
- Ubuntu 16.04 LTS Server 64 bit
- CPU 4 Core, Memory 4GB

▶ Sandbox

- Win7 SP1 32 bit
- CPU 1 Core, Memory 1024 MB

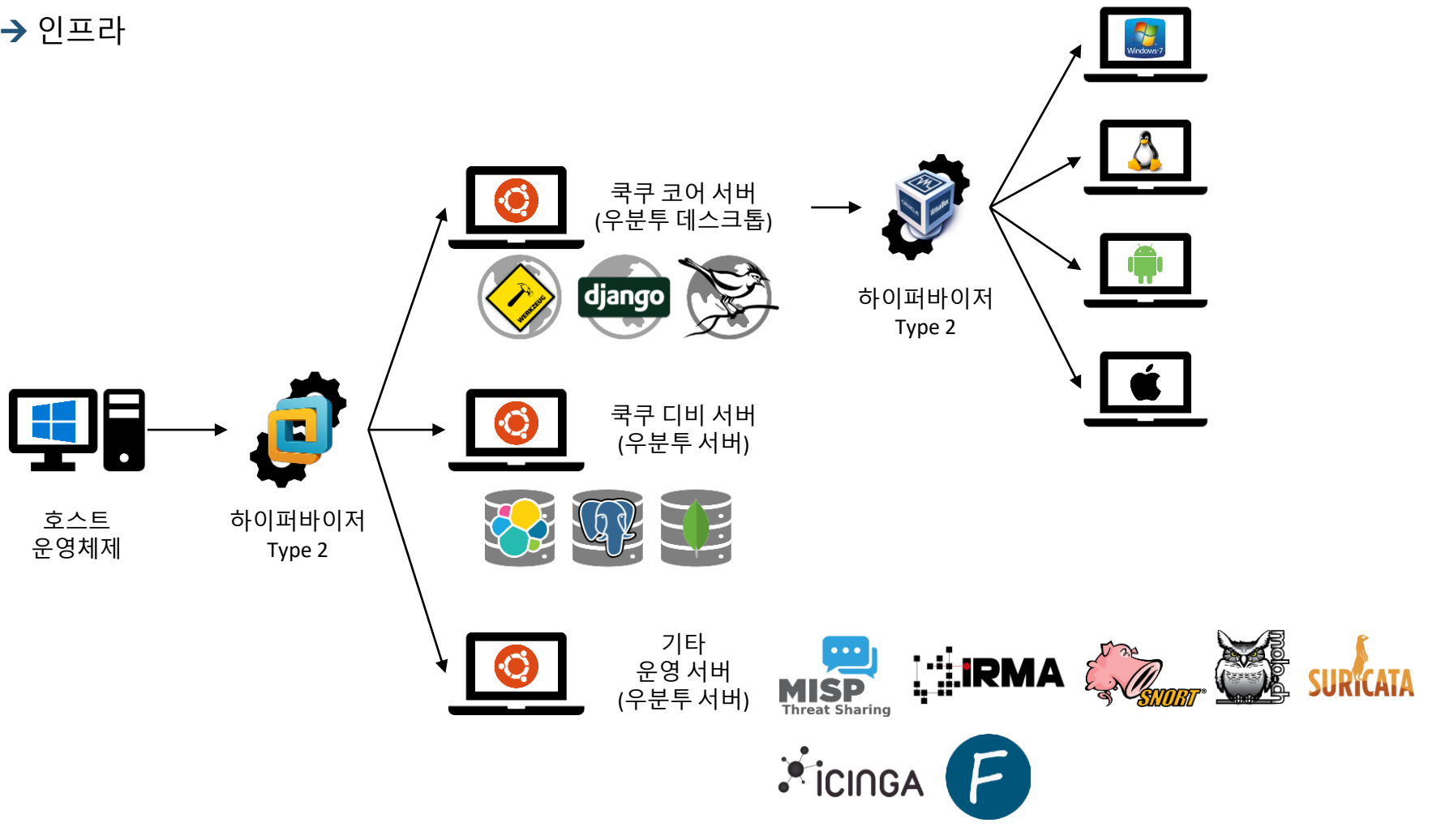
▶ Cuckoo Sandbox Application

- v 2.0.X



● 쿡루 샌드박스

→ 인프라



● 쿡쿡 샌드박스

→ 쿡쿡 샌드박스에서 사용하는 기술

▶ 데이터베이스

- 스케줄링 - RDBMS(SQLite, PostgreSQL, MySQL 등)
- 웹 데이터 - NoSQL(MongoDB)
- 검색 엔진 - ElasticSearch

▶ 지원하는 하이퍼바이저

- XenServer, VMware(ESX, vSphere, Workstation), KVM, Qemu, Virtualbox

▶ 물리 샌드박스 - Fog Project

▶ 안드로이드 샌드박스 - AVD

▶ 메모리 분석 - 볼라티리티(Volatility)

▶ 네트워크 관련 - 스노트(Snort), 수리카타(Suricata), 모로크(Moloch)

▶ HTTP 서버 - 장고(Django), Werkzeug

● 쿡쿠 샌드박스

→ 쿡쿠 샌드박스에서 사용하는 기술

- ▶ 행위 분석 - 자체 개발한 트레이싱과 후킹을 이용한 시스템 분석 에이전트
- ▶ 보고서 - HTML, PDF, JSON, 웹 사이트
- ▶ 인텔리전스 연동 - MISP
- ▶ 쿡쿠 시스템 관리 도구 - ICINGA
- ▶ 안드로이드 관련 - GooglePlay, Droidmon, ApkInfo,
- ▶ 바이러스토탈, Certification Pinning at PE

● 쿡쿠 샌드박스

→ 쿡쿠 샌드박스에서 사용하는 기능

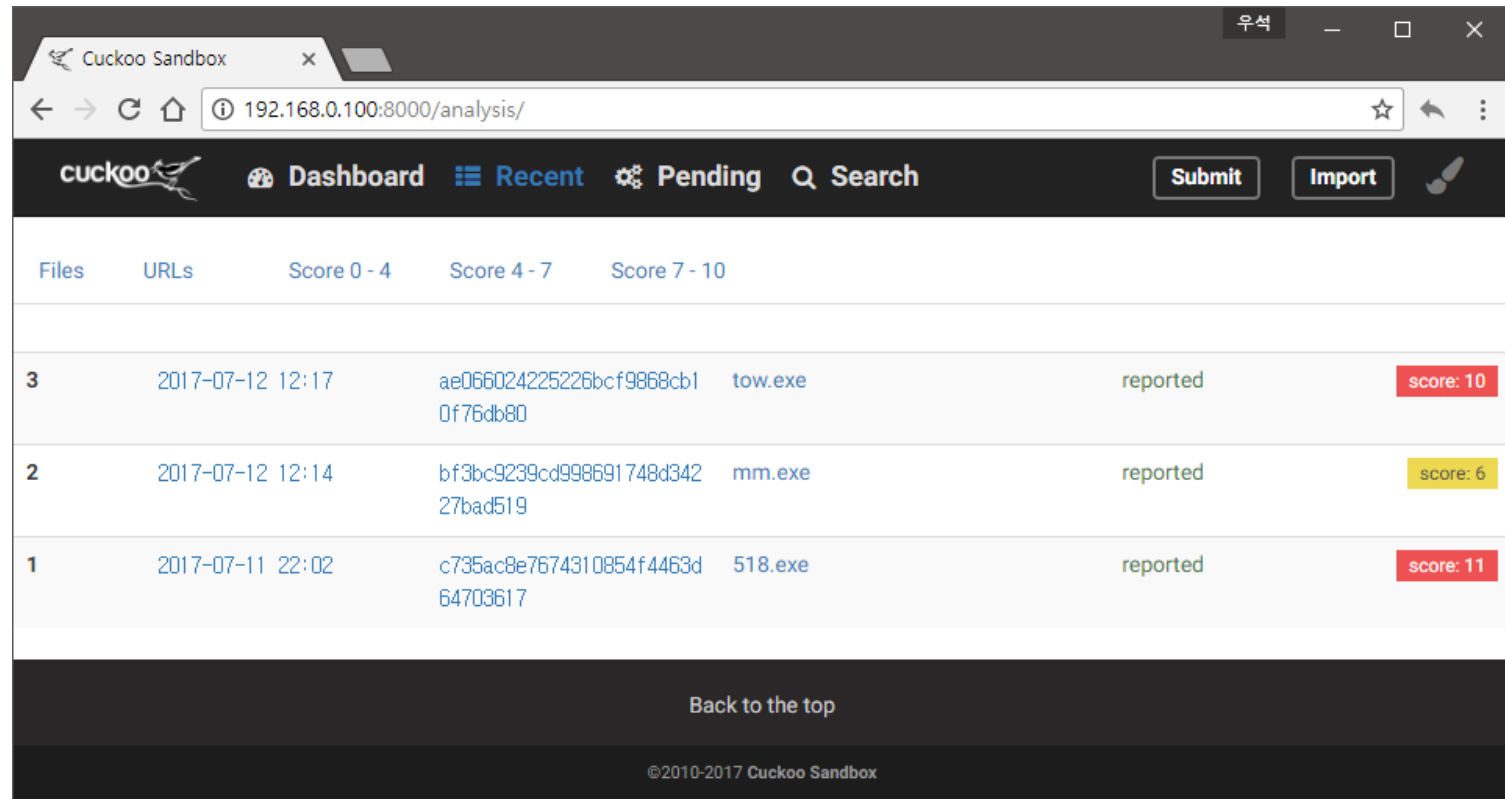
- ▶ 다양한 악성코드용 해시 지원 - ssdeep, Imphash
- ▶ 전자메일 첨부파일 분석 기능 - 2.0-RC2에서 언급 / 구현중(?)
- ▶ 레터럴 무브먼트 분석 지원 - 다수의 샌드박스 동시 운영 2.0-RC1에서 언급 / 구현중(?)
- ▶ 네트워크 라우팅 지원 - no 라우팅, drop 라우팅, dirty line, InetSim, Tor, VPN 등) / 구현되어 있으나 문서화되어 있지 않음
- ▶ TLS/HTTPS 트래픽 수집 및 복호화 - MITMProxy & HTTPReplay를 이용 / 구현되어 있으나 문서화되어 있지 않음
- ▶ 메모리 베이스라인 분석 지원 - 구현되어 있으나 문서화되어 있지 않음
- ▶ 프로세스 메모리에서 URL 추출 - 구현되어 있으나 잘 동작하지 않음
- ▶ 정적 분석 시그니처와 동적 분석 시그니처 제작 - YARA와 쿡쿠 함수를 이용 / 구현 완료
- ▶ 악성 행위 수치화 - 동적 분석 시그니처를 이용 / 구현 완료
- ▶ 악성코드 비교 기능 - API 호출 순서에 의한 비교 / 구현 완료

● 쿡쿠 샌드박스

→ 분석 데이터

▶ 시연

- <http://192.168.0.100:8000>



The screenshot shows the Cuckoo Sandbox web interface in a browser window. The address bar displays '192.168.0.100:8000/analysis/'. The interface includes a navigation bar with 'Dashboard', 'Recent', 'Pending', and 'Search' links, along with 'Submit' and 'Import' buttons. Below the navigation bar, there is a table of analysis results with columns for 'Files', 'URLs', 'Score 0 - 4', 'Score 4 - 7', and 'Score 7 - 10'. The table lists three analysis entries, each with a rank, timestamp, file hash, filename, status, and score.

	Files	URLs	Score 0 - 4	Score 4 - 7	Score 7 - 10
3		2017-07-12 12:17	ae066024225226bcf9868cb1 0f76db80	tow.exe	reported score: 10
2		2017-07-12 12:14	bf3bc9239cd998691748d342 27bad519	mm.exe	reported score: 6
1		2017-07-11 22:02	c735ac8e7674310854f4463d 64703617	518.exe	reported score: 11

Back to the top

©2010-2017 Cuckoo Sandbox

● 다른 바이너리 자동화 분석 인텔리전스 서비스

→ 조 샌드박스 - <https://joesecurity.org/joe-security-products>

- 무료 - 6가지 유형을 분석하는 기능을 제공
 - File Analyzer, Document Analyzer, APK Analyzer, URL Analyzer, Yara Rule Generator, Mail Analyzer
- 유료 - 무료의 모든 기능을 통합, 추가 기능(iOS, Mac 등) 지원
 - JOESandbox Cloud
- 보고서 샘플 - <https://www.joesecurity.org/joe-sandbox-reports>

▶ 하이브리드 아날리시스 - <https://www.hybrid-analysis.com/>

- API를 제공, OSINT로 사용하기 좋으나 개인 또는 독립형은 구매 필요
- 로그인 후 샘플 다운로드 가능
- 보고서 샘플 - <https://www.hybrid-analysis.com/sample/37462679766ade74c154a103f34df496760bff59e659ff5e1c9b18017806fd3d?environmentId=100>

● 바이너리 인텔리전스의 한계

→ 인텔리전스 목적

- ▶ 적을 파악 (선별)
- ▶ 과거를 통해 미래를 예측 (대응)

→ 바이너리 분석 인텔리전스

- ▶ 바이너리에서 사용하는 다양한 공격 및 응용 기법을 통계적으로 다룰 수 있음
 - 데이터 마이닝을 통해 근본에 가까운 기법을 파악할 수 있음
- ▶ 체계적으로 악성코드를 관리, 많은 데이터가 모이면 연관관계 분석 가능

→ 바이너리 분석 인텔리전스의 한계

- ▶ 모든 데이터는 변조 / 위장 가능하다는 전제를 인식
- ▶ 한계를 해결하기 위해 최대한 많은 데이터를 수집하는 것이 중요
 - 서버, 클라이언트 전체를 다루어야 더욱 선명한 정보를 얻을 수 있음 - 바이너리의 한계
 - 하지만 서버의 침해를 분석하거나 정보를 얻기 위한 권한이나 수사권이 없음(π.π)

● 결론

- 악성코드 초동 분석은 자동화 분석 이용을 추천
 - ▶ 500건의 악성코드 > 자동화 분석 > 10건의 특별한 샘플 추출 > 상세 분석 = 효율성

- 악성 여부를 파악하기 힘들 때 자동화 분석을 추천
 - ▶ 분석 요청 후 간략한 정보만으로 쉽게 추측이 가능
 - 예, 오토런, 파일 시스템 사용 흔적, C2 통신 이력, 바이러스 토탈 탐지 내역 등 한눈에
 - ▶ 분석에 대한 기초지식을 크게 요구하지 않음 (도구의 편리성)

- 오픈소스이기에 기능의 알고리즘을 파악하여 활용
- API를 이용하여 기존 시스템에 접목하여 활용
- 악성코드 관련 신기술을 파악하기 좋음
 - ▶ 개선 및 기능 강화할 때 마다 다양한 기술을 접목시켜 활용
 - ▶ 여러 참여자가 다양한 형태로 변형하여 공개

● 결론

두산백과

지피지기백전불태

[知彼知己百戰不殆]

요약 상대를 알고 나를 알면 백 번 싸워도 위태롭지 않다는 뜻으로, 상대편과 나의 약점과 강점을 충분히 알고 승산이 있을 때 싸움에 임하면 이길 수 있다는 말.

Q & A

The End