K-Shield Jr. 13기 단기 집중 과정

OpenAI 기반 공격 시나리오 생성 CALDERA 플러그인 개발

명장이 이끄는 명팀

박찬욱 서가연 이상현 정원빈 좌민서

목차

명짱이 이끄는 명팀

박찬욱
모델 학습

이상현
모델 학습

좌민서
플러그인 개발

서가연
플러그인 개발

정원빈
플러그인 개발

TTPs

TTPs(Tactic, Technique, Procedure)

- 공격자가 목표를 달성하기 위해 사용하는 전술과 기법, 그 세부 과정을 의미
- 공격자의 행동을 구조화하고 체계적으로 이해하는 데 중요함

ex)

Initial Access(초기 접근)

Phishing(피싱)

공격자는 타겟 조직의 직원을 대상으로 이메일 피싱 캠페인을 진행하여 악성 링크나 파일을 클릭하도록 유도

TTPs

TTPs(Tactic, Technique, Procedure)

- 공격자가 목표를 달성하기 위해 사용하는 전술과 기법, 그 세부 과정을 의미
- 공격자의 행동을 구조화하고 체계적으로 이해하는 데 중요함

ex)

Execution(실행)

Command and Scripting Interpreter
(명령 및 스크립트 인터프리터)

공격자는 PowerShell을 이용하여 원격 서버에 악성 스크립트를 실행

TTPs

TTPs(Tactic, Technique, Procedure)

- 공격자가 목표를 달성하기 위해 사용하는 전술과 기법, 그 세부 과정을 의미
- 공격자의 행동을 구조화하고 체계적으로 이해하는 데 중요함

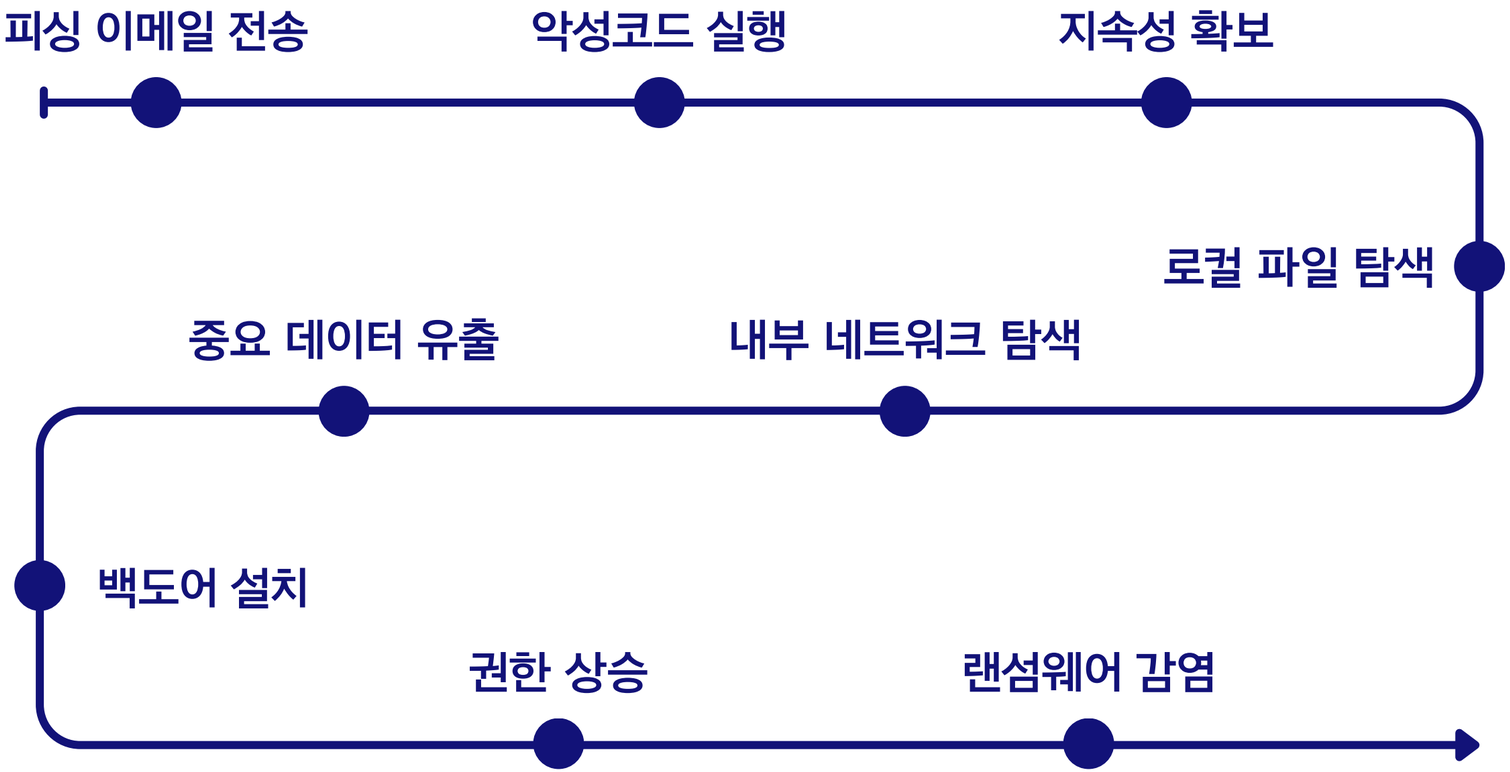
ex)

Privilege Escalation
(권한 상승)

Process Injection
(프로세스 인젝션)

공격자는 정상적인 시스템
프로세스에 악성 코드를
주입하여 높은 권한으로 실행

공격 시뮬레이션



<공격 시나리오 예시>

공격 시뮬레이션

TTPs 이해를 통해 효과적인 방어 전략 수립

TTPs를 활용한 공격 시뮬레이션 수요 증가

침투 테스트 프레임워크와 BAS 제품의 부상

공격 시뮬레이션

TTPs 이해를 통해 효과적인 방어 전략 수립

TTPs를 활용한 공격 시뮬레이션 수요 증가

침투 테스트 프레임워크와 BAS 제품의 부상

실제 위협 시나리오에 맞춰 보안 체계 평가 및 취약점 발견

보안 팀의 대응 능력을 강화해 현실적인 방어 전략을 수립

Caldera

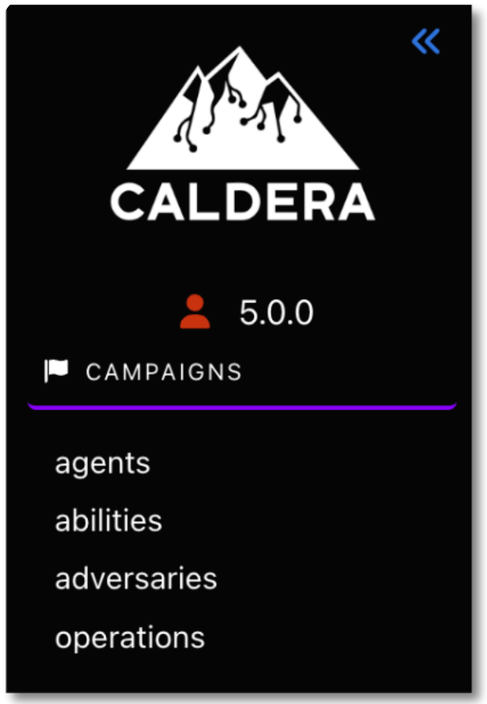
MITRE Caldera

- 공격 시뮬레이션을 쉽게 실행하도록 설계된 레드팀을 위한 플랫폼
- 조직의 사이버 방어 체계를 테스트하고 강화하는 데 유용함
- 직관적인 웹 UI를 통해 손쉽게 설정, 시나리오 실행 및 결과 분석

MITRE ATT&CK에서 정의된 TTPs를 바탕으로
실제 공격 시나리오 시뮬레이션 가능

Caldera

Caldera 주요 기능



Agent
: 공격 시뮬레이션을 수행하기 위해 타겟 시스템에 설치되는 소프트웨어

Abilities
: Caldera에서 사용 가능한 technique 모음

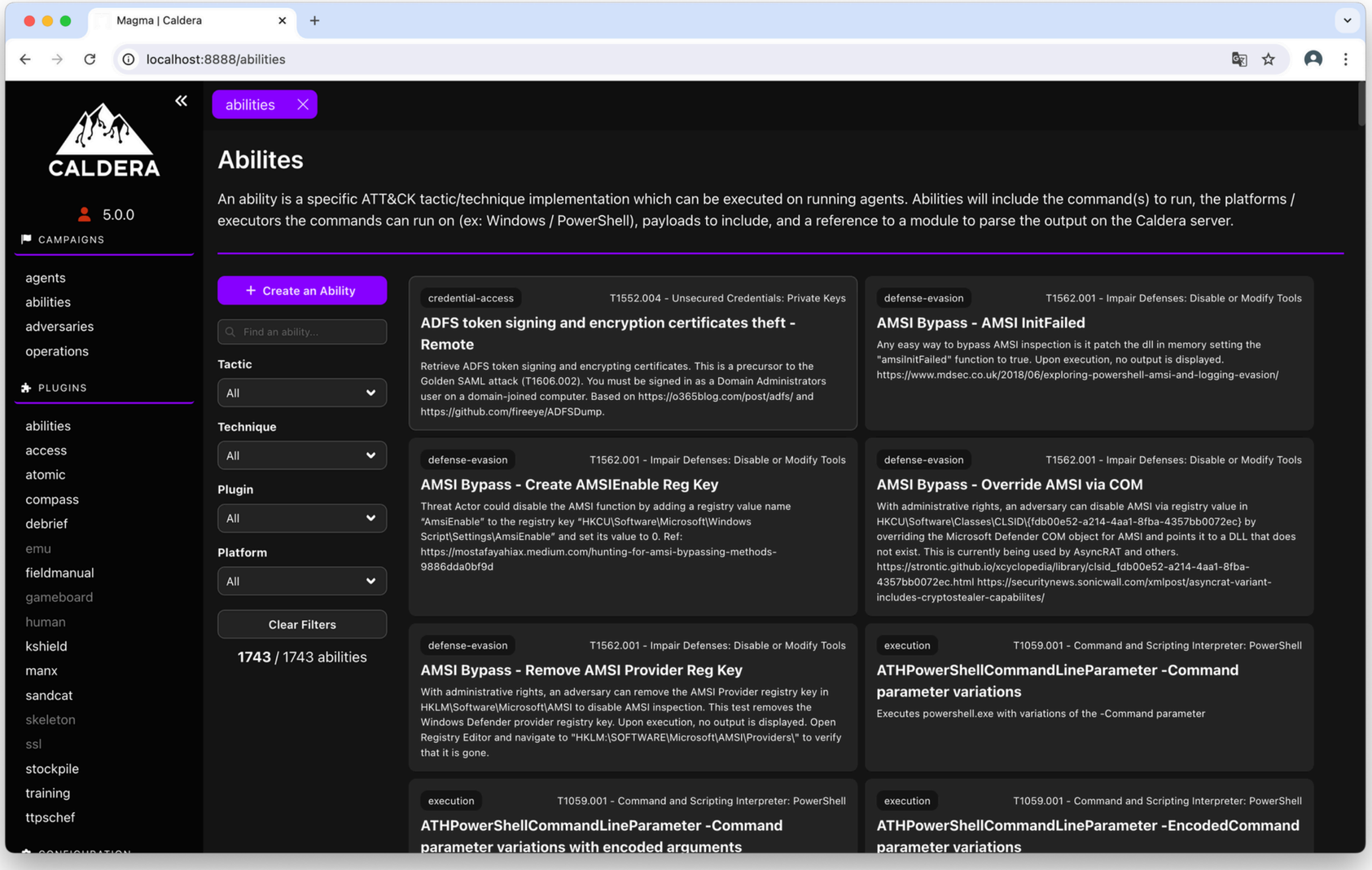
Adversary
: 여러 개의 technique으로 구성된 하나의 시나리오

Operation
: 생성된 시나리오를 실행

K-Shield Jr.
취약점 분석 과정

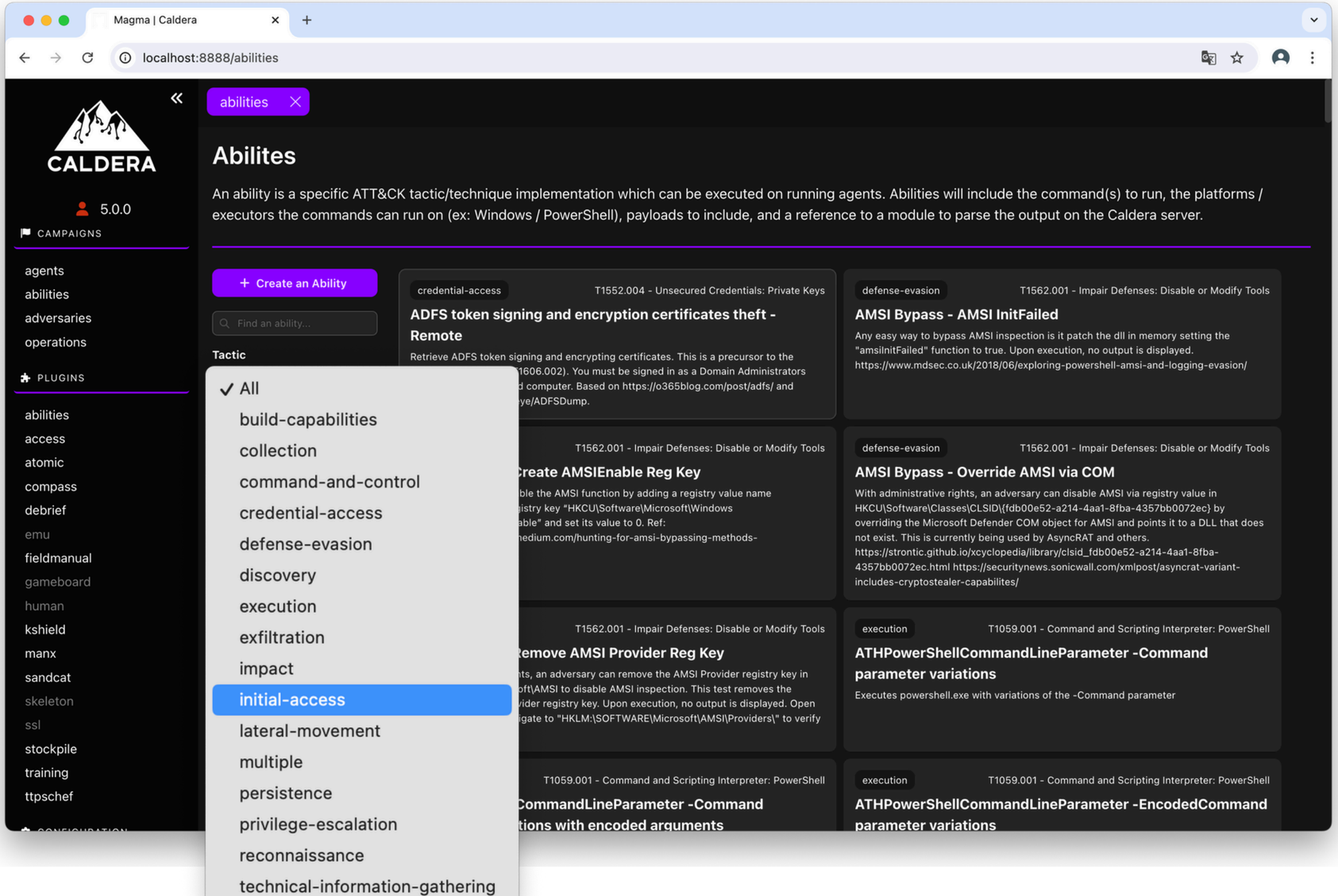
- 팀 소개
- 프로젝트 배경
- 프로젝트 목표
- 프로젝트 진행
- 향후 계획

Caldera



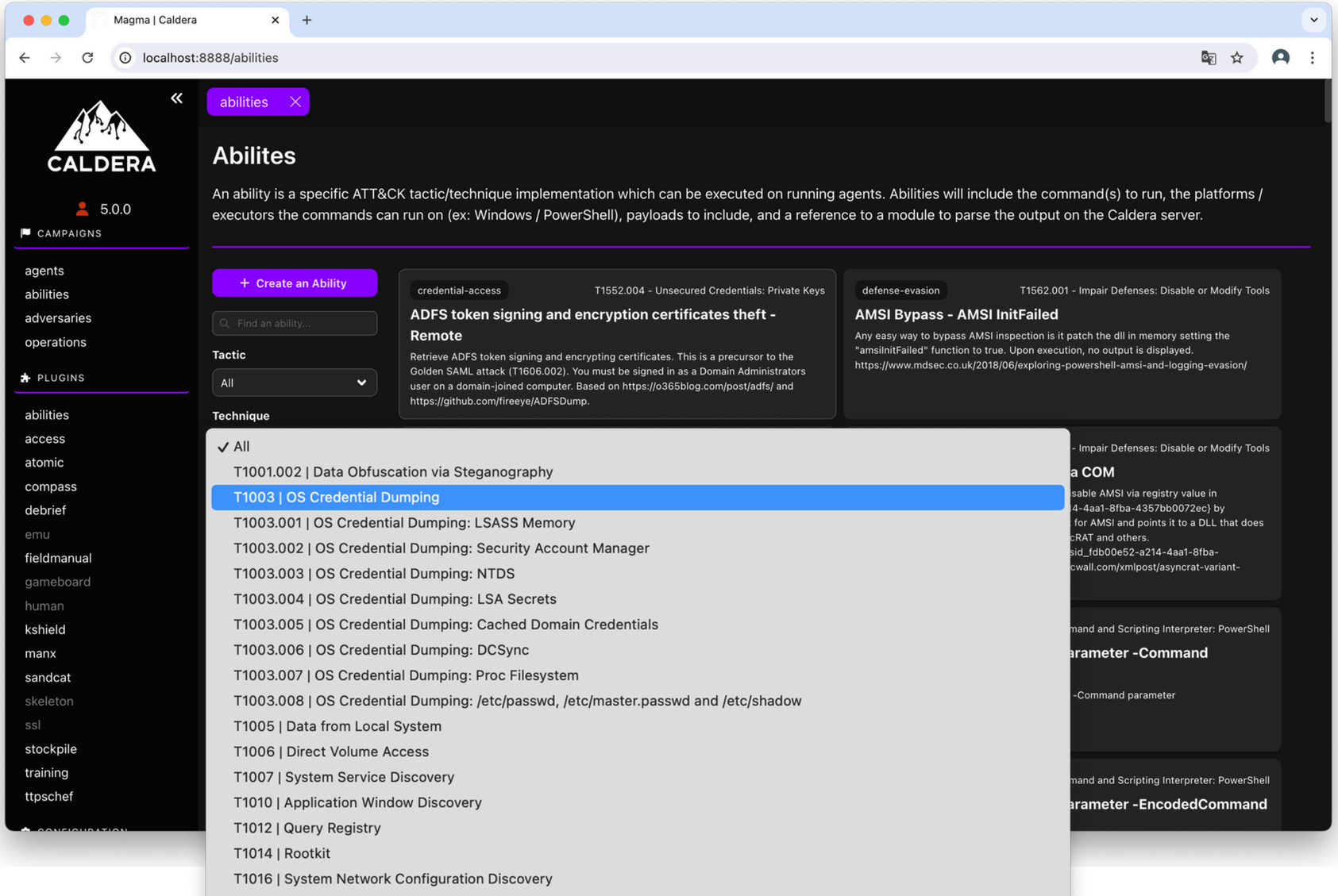
K-Shield Jr.
취약점 분석 과정

Caldera



K-Shield Jr.
취약점 분석 과정

Caldera



프로젝트 배경

1. 보고서 분석의 어려움

- 방대한 데이터와 복잡한 기술 용어
- 상세 내용을 파악하는 데 많은 시간 소요

2. 공격 시나리오 재현의 어려움

- 복잡한 시나리오를 수작업으로 재현하는 데 필요한 인력과 시간의 한계

K-Shield Jr.
취약점 분석 과정

프로젝트 배경

As is common for malware, the GeminiDuke infostealer uses a mutex to ensure that only one instance of itself is running at a time. What is less common is that the name used for the mutex is often a timestamp. We believe these timestamps to be generated during the compilation of GeminiDuke from the local time of the computer being used.

Comparing the GeminiDuke compilation timestamps, which always reference the time in the UTC+0 timezone, with the local time timestamps used as mutex names, and adjusting for the presumed timezone difference, we note that all of the mutex names reference a time and date that is within seconds of the respective sample's compilation timestamp. Additionally, the apparent timezone of the timestamps in all of the GeminiDuke samples compiled during the winter is UTC+3, while for samples compiled during the summer, it is UTC+4.

The observed timezones correspond to the pre-2011 definition of Moscow Standard Time (MSK) [20], which was UTC+3 during the winter and UTC+4 during the summer. In 2011 MSK stopped following Daylight Saving Time (DST) and was set to UTC+4 year-round, then reset to UTC +3 year-round in 2014. Some of the observed GeminiDuke samples that used timestamps as mutex names were compiled while MSK still respected DST and for these samples, the timestamps perfectly align with MSK as it was defined at the time.

Map of timezones in Russia; © Eric Muller[20] Pink: MSK (UTC +3); Orange: UTC +4

However, GeminiDuke samples compiled after MSK was altered still vary the timezone between UTC+3 in the winter and UTC+4 during the summer. While computers using Microsoft Windows automatically adjust for DST, changes in timezone definitions require that an update to Windows be installed. We therefore believe that the Dukes group simply failed to update the computer they were using to compile GeminiDuke samples, so that the timestamps seen in later samples still appear to follow the old definition of Moscow Standard Time.

The GeminiDuke infostealer has occasionally been wrapped with a loader that appears to be unique to GeminiDuke and has never been observed being used with any of the other Duke toolsets. GeminiDuke also occasionally embeds additional executables that attempt to achieve persistence on the victim computer. These persistence components appear to be uniquely customized for use with GeminiDuke, but they use many of the same techniques as CosmicDuke persistence components.

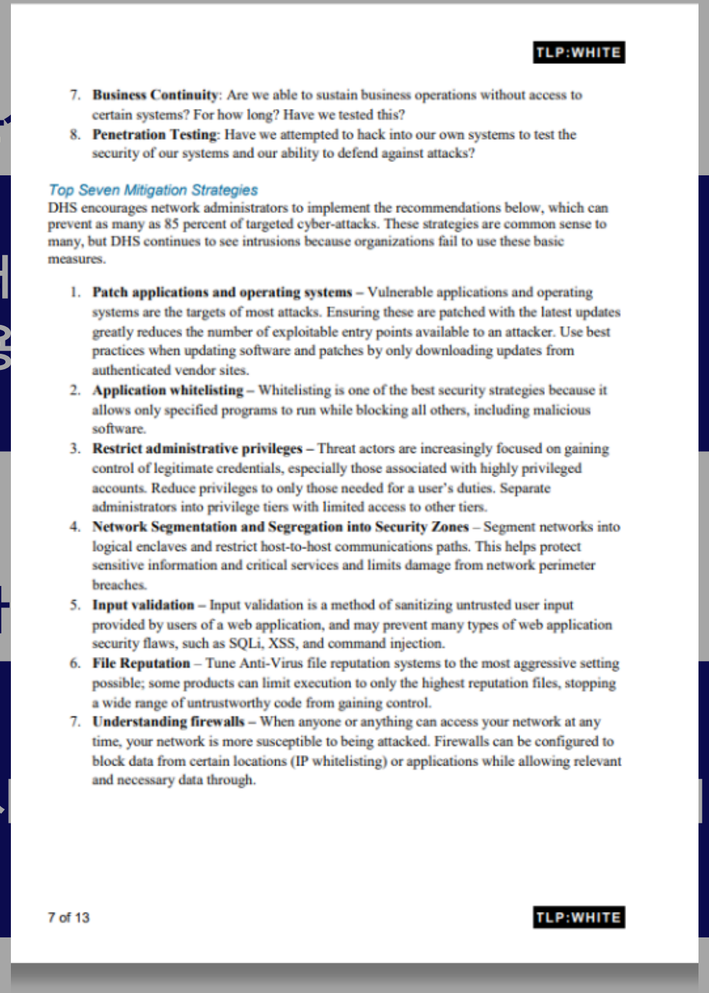
고서 분

대한 더
세 내용

격 시나

잡한 시

필요한



TLP:WHITE

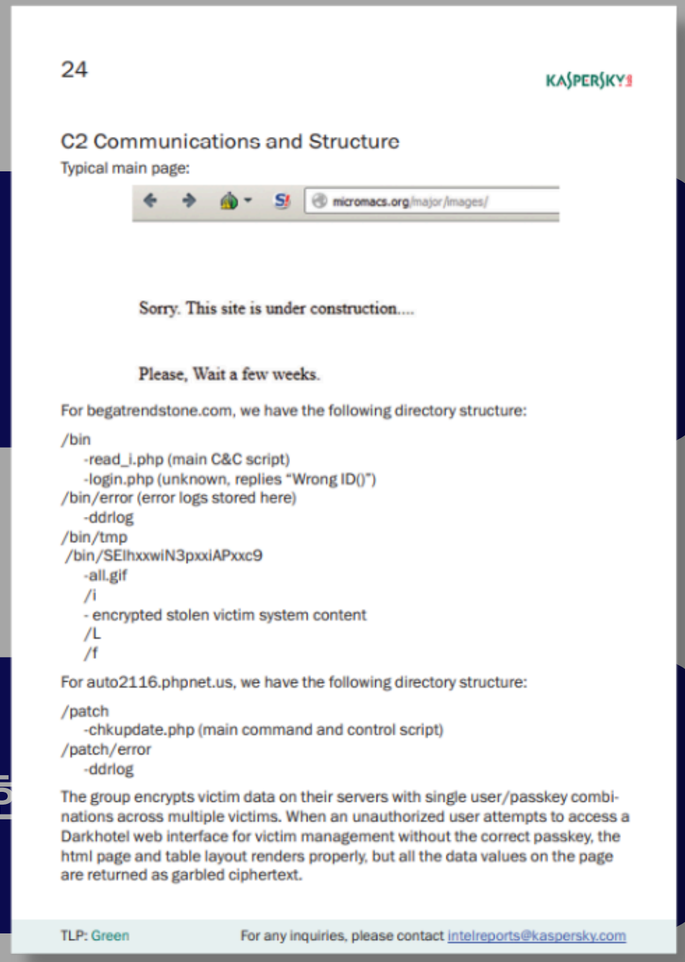
7. **Business Continuity**: Are we able to sustain business operations without access to certain systems? For how long? Have we tested this?
8. **Penetration Testing**: Have we attempted to hack into our own systems to test the security of our systems and our ability to defend against attacks?

Top Seven Mitigation Strategies

DHS encourages network administrators to implement the recommendations below, which can prevent as many as 85 percent of targeted cyber-attacks. These strategies are common sense to many, but DHS continues to see intrusions because organizations fail to use these basic measures.

1. **Patch applications and operating systems** – Vulnerable applications and operating systems are the targets of most attacks. Ensuring these are patched with the latest updates greatly reduces the number of exploitable entry points available to an attacker. Use best practices when updating software and patches by only downloading updates from authenticated vendor sites.
2. **Application whitelisting** – Whitelisting is one of the best security strategies because it allows only specified programs to run while blocking all others, including malicious software.
3. **Restrict administrative privileges** – Threat actors are increasingly focused on gaining control of legitimate credentials, especially those associated with highly privileged accounts. Reduce privileges to only those needed for a user's duties. Separate administrators into privilege tiers with limited access to other tiers.
4. **Network Segmentation and Segregation into Security Zones** – Segment networks into logical enclaves and restrict host-to-host communications paths. This helps protect sensitive information and critical services and limits damage from network perimeter breaches.
5. **Input validation** – Input validation is a method of sanitizing untrusted user input provided by users of a web application, and may prevent many types of web application security flaws, such as SQLi, XSS, and command injection.
6. **File Reputation** – Tune Anti-Virus file reputation systems to the most aggressive setting possible; some products can limit execution to only the highest reputation files, stopping a wide range of untrustworthy code from gaining control.
7. **Understanding firewalls** – When anyone or anything can access your network at any time, your network is more susceptible to being attacked. Firewalls can be configured to block data from certain locations (IP whitelisting) or applications while allowing relevant and necessary data through.

TLP:WHITE



KASPERSKY

C2 Communications and Structure

Typical main page:

micromacs.org/major/images/

Sorry. This site is under construction....

Please, Wait a few weeks.

For begatrendstone.com, we have the following directory structure:

/bin
-read_i.php (main C&C script)
-login.php (unknown, replies "Wrong ID()")
/bin/error (error logs stored here)
-ddrlog
/bin/tmp
/bin/SElhxxwiN3pxxiAPxxc9
-all.gif
/i
- encrypted stolen victim system content
/L
/f

For auto2116.phpnet.us, we have the following directory structure:

/patch
-chkupdate.php (main command and control script)
/patch/error
-ddrlog

The group encrypts victim data on their servers with single user/passkey combinations across multiple victims. When an unauthorized user attempts to access a Darkhotel web interface for victim management without the correct passkey, the html page and table layout renders properly, but all the data values on the page are returned as garbled ciphertext.

시간 소모와 번거로움

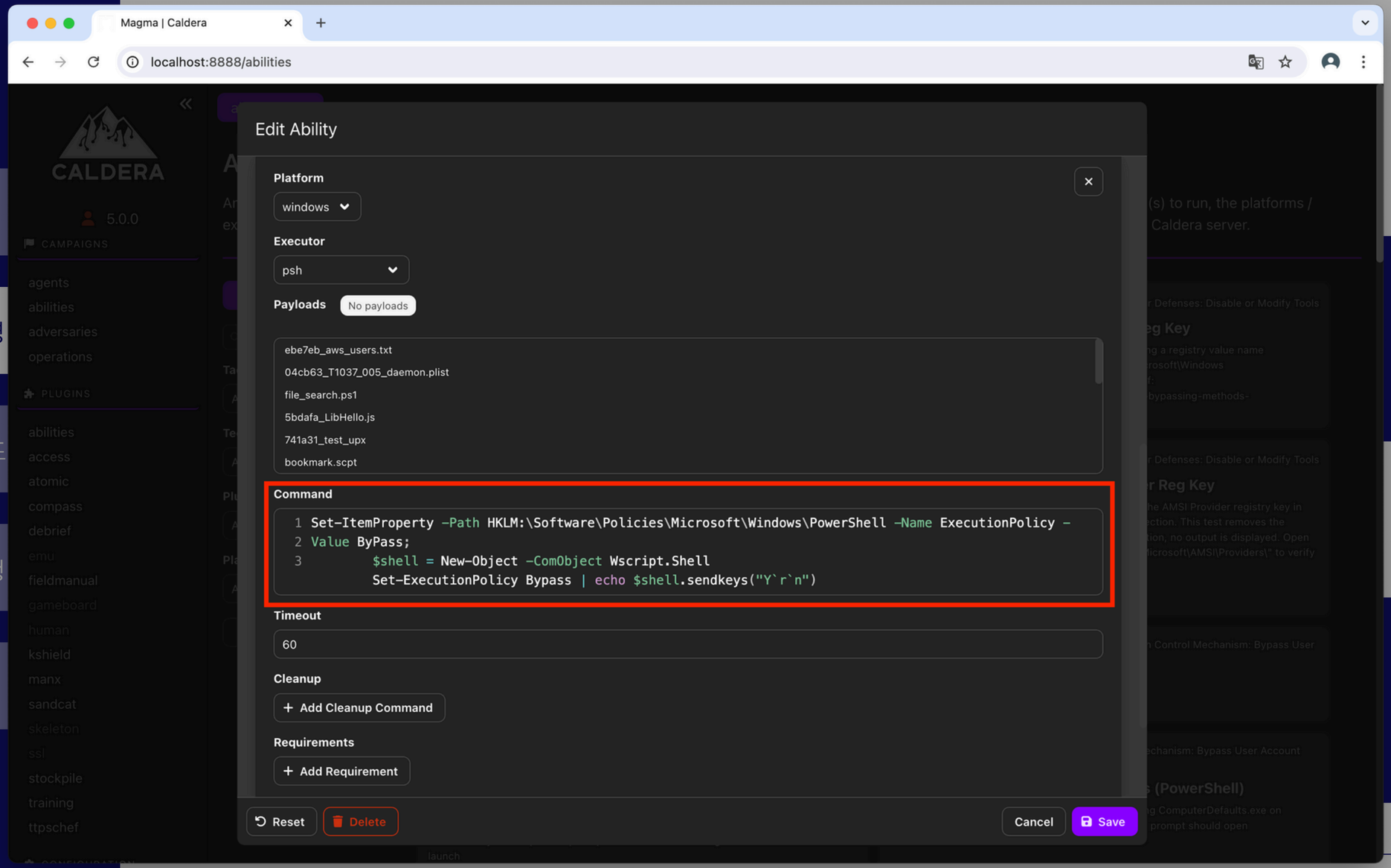
프로젝트 배경

1. 보고서 분석의 어려움

- 방대한 데이터와 복잡한 기술 용어
- 상세 내용을 파악하는 데 많은 시간 소요

2. 공격 시나리오 재현의 어려움

- 복잡한 시나리오를 수작업으로 재현하는 데 필요한 인력과 시간의 한계



사전 정의된 명령어 제공의 필요성

프로젝트 배경

1. 보고서 분석의 어려움

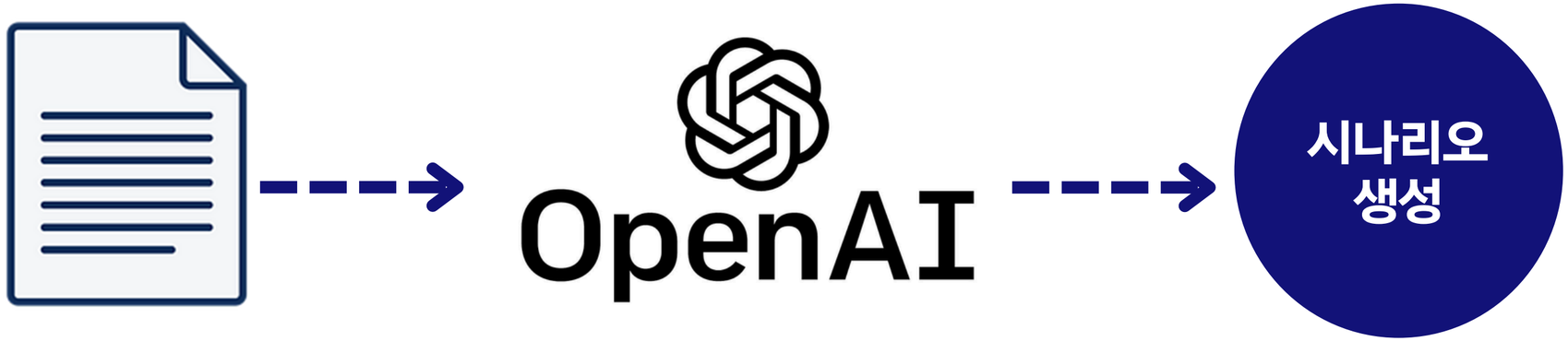
- 방대한 데이터와 복잡한 기술 용어
- 상세 내용을 파악하는 데 많은 시간 소요

2. 공격 시나리오 재현의 어려움

- 복잡한 시나리오를 수작업으로 재현하는 데 필요한 인력과 시간의 한계

프로젝트 목표

공격 그룹들의 실제 시나리오를 분석한 보고서를 입력 받아
OpenAI를 활용해 공격 시나리오를 재현할 수 있는 플러그인을 개발



공격 그룹 보고서

프로젝트 진행 순서

01 프롬프트 엔지니어링

02 Caldera 플러그인 개발

03 테스트 및 검증

04 시연 영상

05 기대효과

1. 프롬프트 엔지니어링

1) TTPs와 UUID를 매칭시킨 json 형식 데이터

- 1634개의 학습 데이터

```
Technique_ID: T1566.001,
UUID : [1afaec09315ab71fdfb167175e8a019a],
Technique_Name: Phishing: Spearphishing Attachment
```

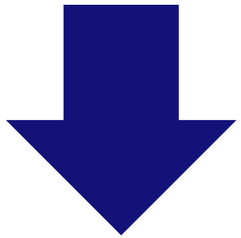
2) GitHub에 공개된 실제 공격 그룹 시뮬레이션 시나리오 데이터 활용

- Step별로 구분하여 학습 효과 강화

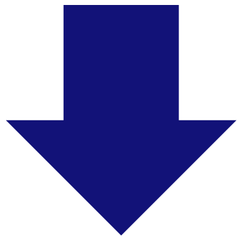
```
Step 1 - Initial Breach
The scenario begins with an initial breach, where a legitimate user clicks (T1204 / T1204.002) an
executable payload (screensaver executable) masquerading as a benign word document (T1036 / T1036.002).
Once executed, the payload creates a C2 connection over port 1234 (T1065) using the RC4 cryptographic
cipher. The attacker then uses the active C2 connection to spawn interactive cmd.exe (T1059 / T1059.003)
and powershell.exe (T1086 / T1059.001).
```

1. 프롬프트 엔지니어링

TTPs와 UUID를 매칭시킨 json 형식 데이터 제공

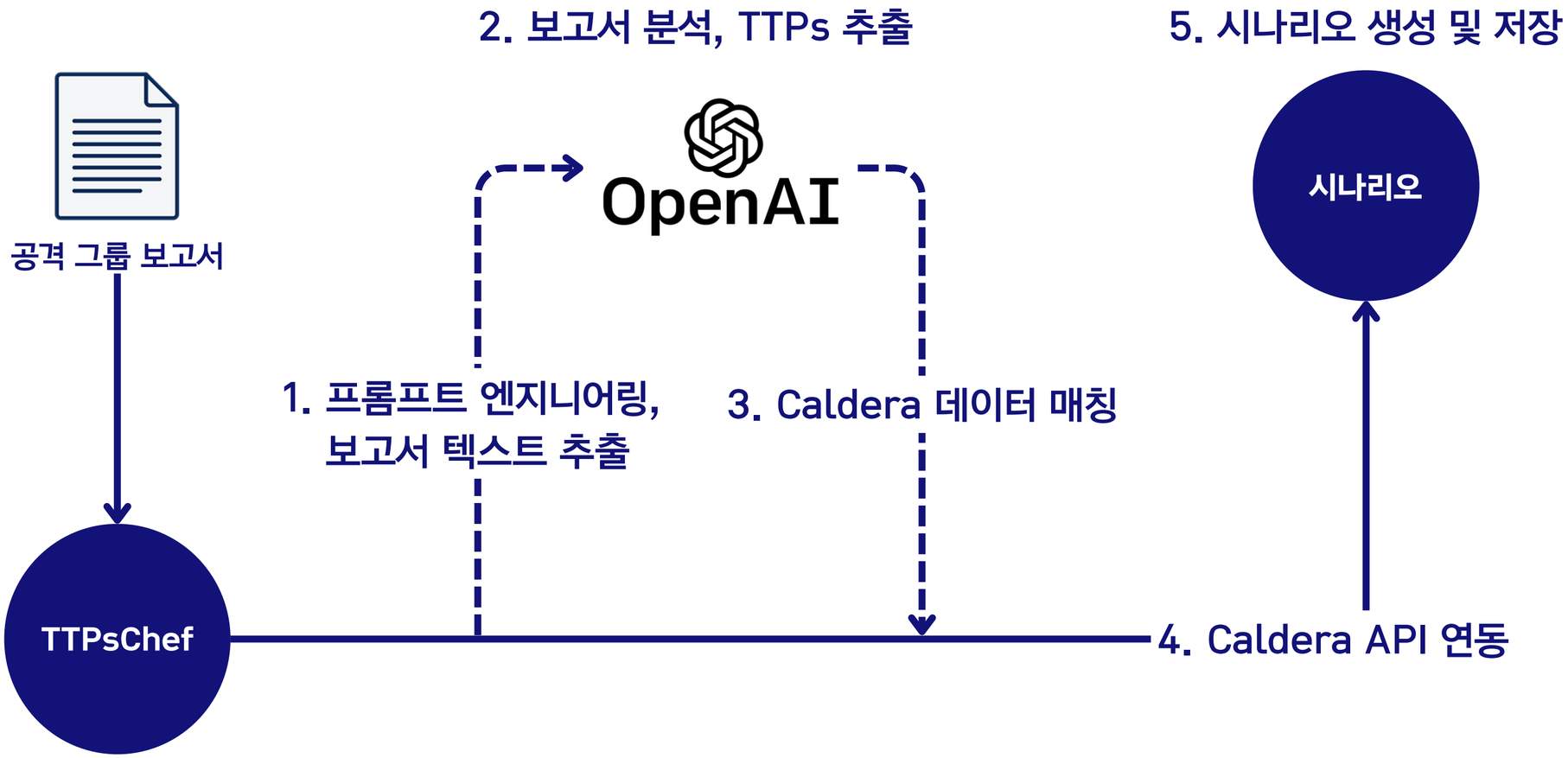


Zero-Shot Prompting으로 공격 시나리오 학습



Python list 형식으로 분석한 Technique 및 UUID 반환

2. Caldera 플러그인 개발



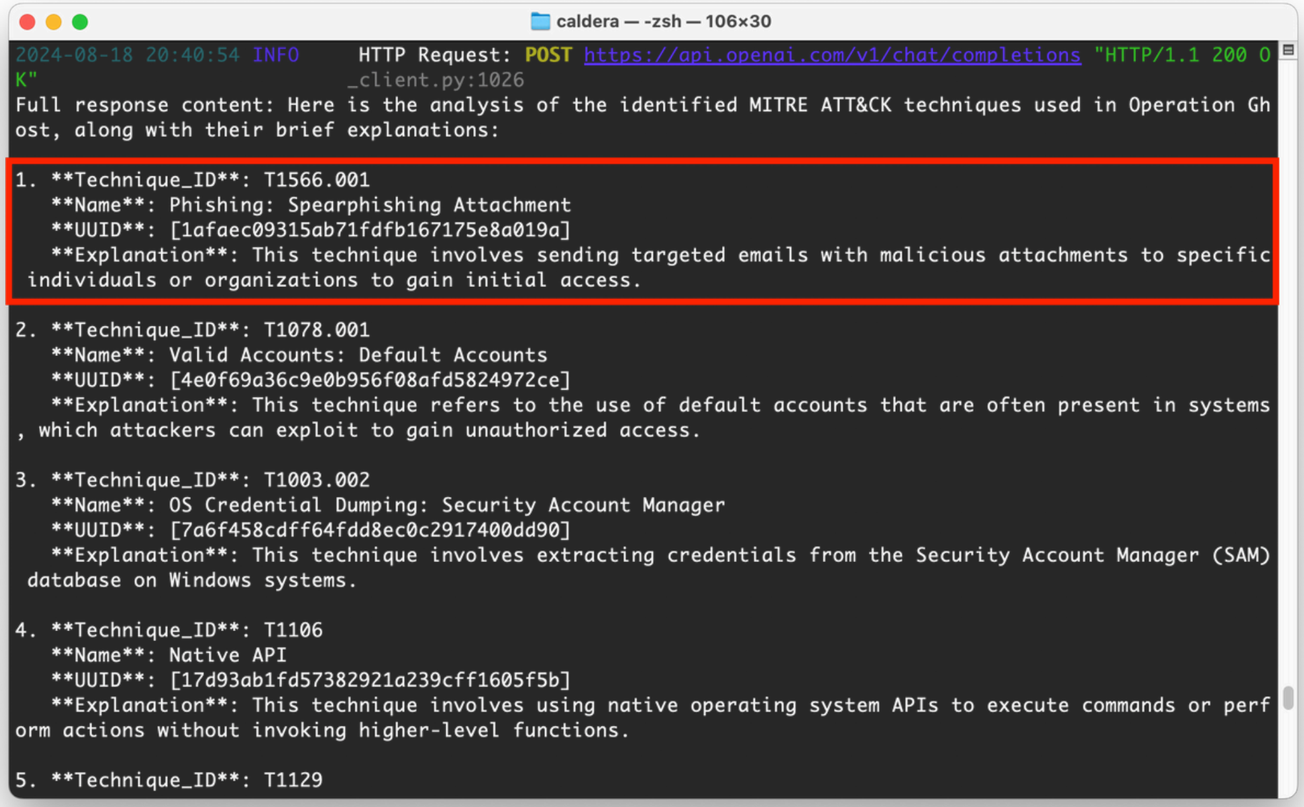
2. Caldera 플러그인 개발

1) 보고서 파일 업로드
Python 라이브러리를 통해 텍스트 추출

```python
@staticmethod
def extract_text_from_pdf(pdf_path):
    reader = pypdf.PdfReader(pdf_path)
    text = ""
    for page in reader.pages:
        text += page.extract_text()
    print(text)

    return text
```

2. Caldera 플러그인 개발

2) Technique 분석
학습 데이터를 바탕으로 사용된 Technique 분석



2. Caldera 플러그인 개발

3) 매칭되는 UUID 출력

분석한 Technique과 매칭되는 ability UUID 출력

```
This analysis provides a comprehensive overview of the techniques employed in Operation Ghost, highlighting the sophisticated methods used by the Duk
es for cyber espionage.
Extracted UUIDs: ['1afaec09315ab71fdfb167175e8a019a', '4e0f69a36c9e0b956f08afd5824972ce', '7a6f458cdff64fdd8ec0c2917400dd90', '17d93ab1fd57382921a239
cff1605f5b', 'bbe3f4e9185839deffef4886b10f137a', 'c0177717b47f2cd07949186523fa3c6b', 'd860912345435c53b73d2d309b2c2b9a', 'e4c51df716410dc7baccead922f
9d9a4', '1f15ab22c39a9b6bb2bb0d77276dfcb3', '73fed1f32224461748c3630217b7d300', '7a84be471d1df7e676a97f7b6286aa13', 'a55beed84485deba9e9c98f0dc0e990a
', 'c275ffb52331397b42ebc52338be3c8c', '9995c62a6263a14ae3b60fe2bb52e67a', '7b6d0accaab6330d701dea8f4d7d96d4', 'eb5568c299ec2de8091eeefb3bc347ec', 'd
65c4ebfab4fcd52f49e10ab1d6b4d2d', '8cd933afe764c4159000cadea55f8ca5', '2f5cf8c3e8a3a3e6e5621e8f77c5bf65', 'eba68ea18bc9ac93808fffbdcf7c3a3f', '7ade88
54cf5f27a38e6b9d9aba15e22b', '638fb6bb-ba39-4285-93d1-7e4775b033a8', '65048ec1-f7ca-49d3-9410-10813e472b30', '720a3356-eee1-4015-9135-0fc08f7eb2d5',
'2edd98c78521bde1e05cd1c10c269fab', '548f5aa3ec2fcfc6872ee10975480f29', '129a792378b8ae67926bb2d043c1a069', '72784d12700b219ec134aa42cec5603e', 'bb0d
f721f4a4defa743efe9e61837c44']
```

2. Caldera 플러그인 개발

4) Adversary 생성

시나리오 생성을 위해 출력된 UUID와 기존 Adversary 플러그인 API 연동

```python
@staticmethod
def create_adversary_api(name, description, atomic_ordering):
  url = 'http://localhost:8888/api/v2/adversaries'
  data = {
    "name": name,
    "description": description,
    "atomic_ordering": atomic_ordering
  }

  response = requests.post(url, data=json.dumps(data))

...

create_adversary_api(filename, description, uuids)
```

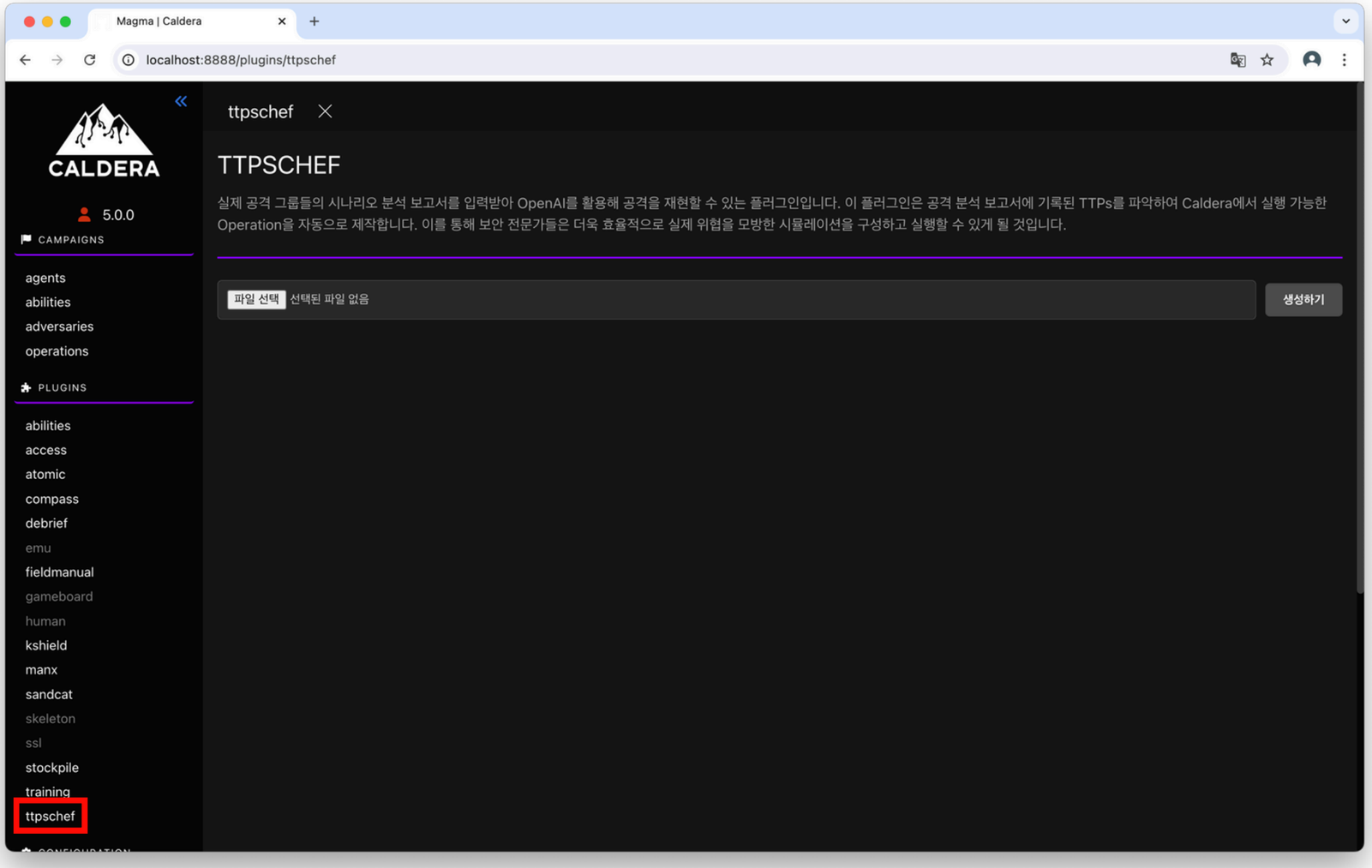
2. Caldera 플러그인 개발

4) Adversary 생성

입력 받은 파일의 이름, 반환된 UUID 전달

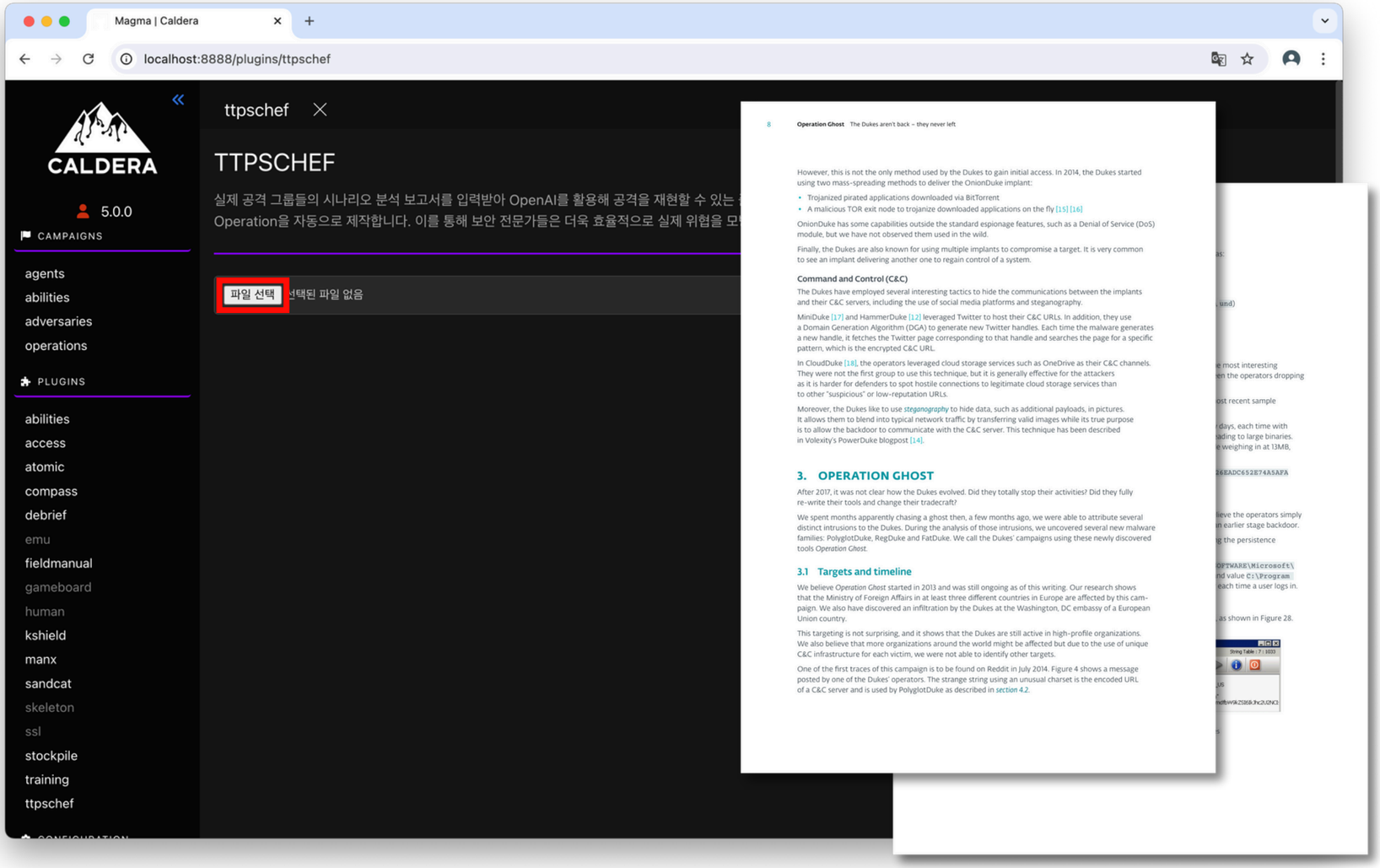
```python
@staticmethod
def create_adversary_api(name, description, atomic_ordering):
  url = 'http://localhost:8888/api/v2/adversaries'
  data = {
    "name": name,
    "description": description,
    "atomic_ordering": atomic_ordering
  }

  response = requests.post(url, data=json.dumps(data))

...

create_adversary_api(filename, description, uuids)
```

2. Caldera 플러그인 개발



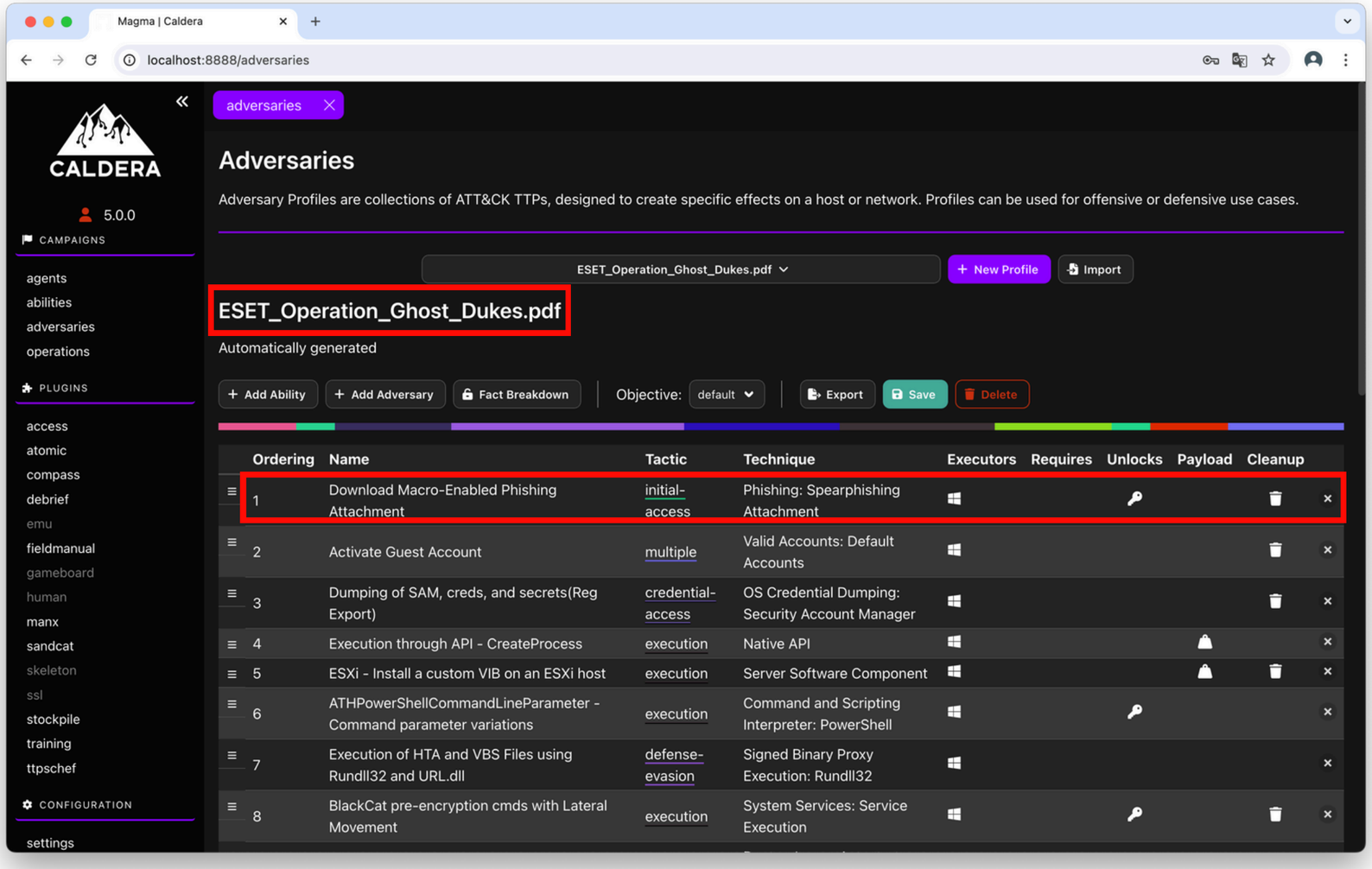
TTPsChef 플러그인 선택

2. Caldera 플러그인 개발



분석 보고서 PDF 업로드

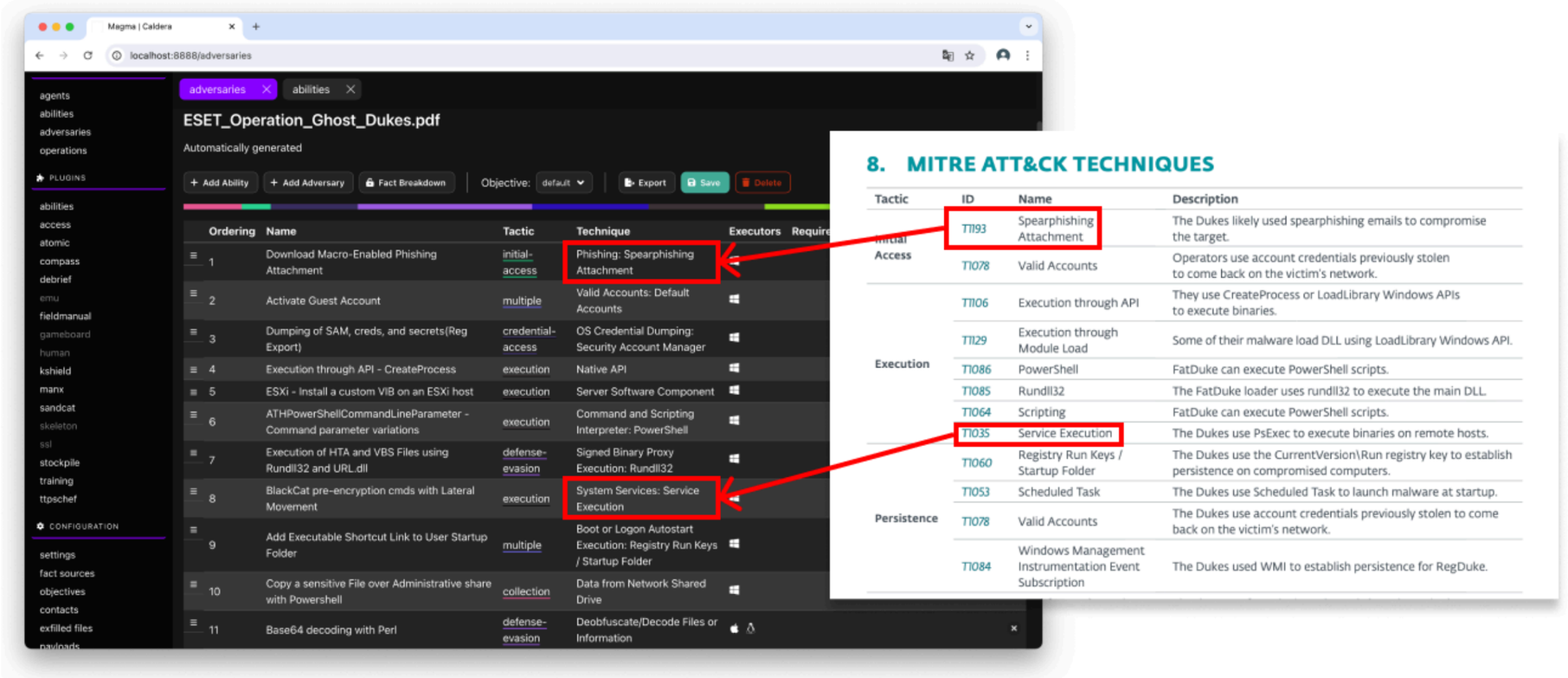
2. Caldera 플러그인 개발



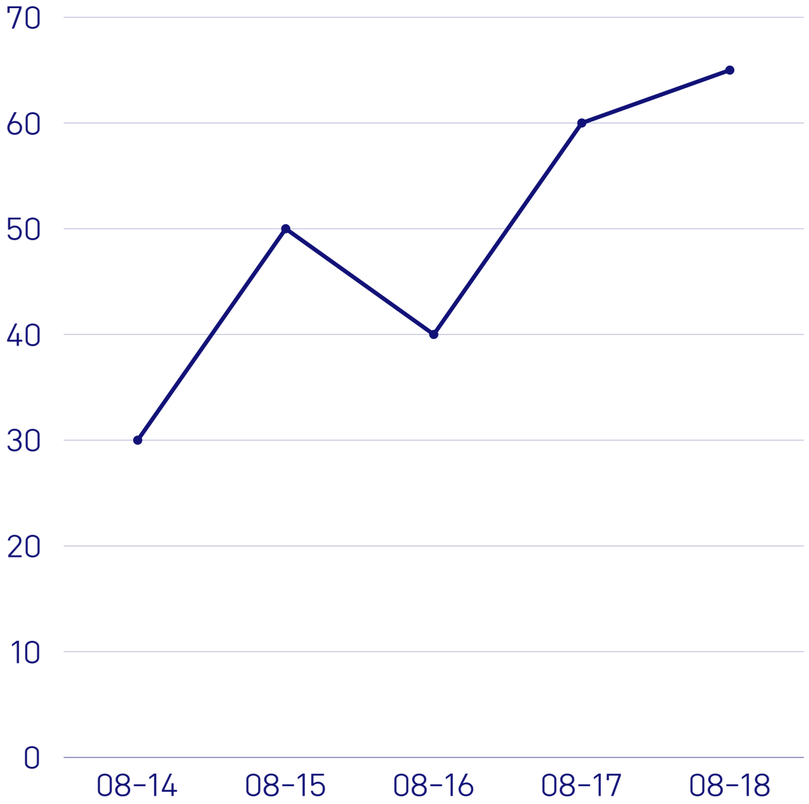
Adversary 시나리오 생성 완료

3. 테스트 및 검증

정확도 검증



3. 테스트 및 검증



<모델 정확도 추이>

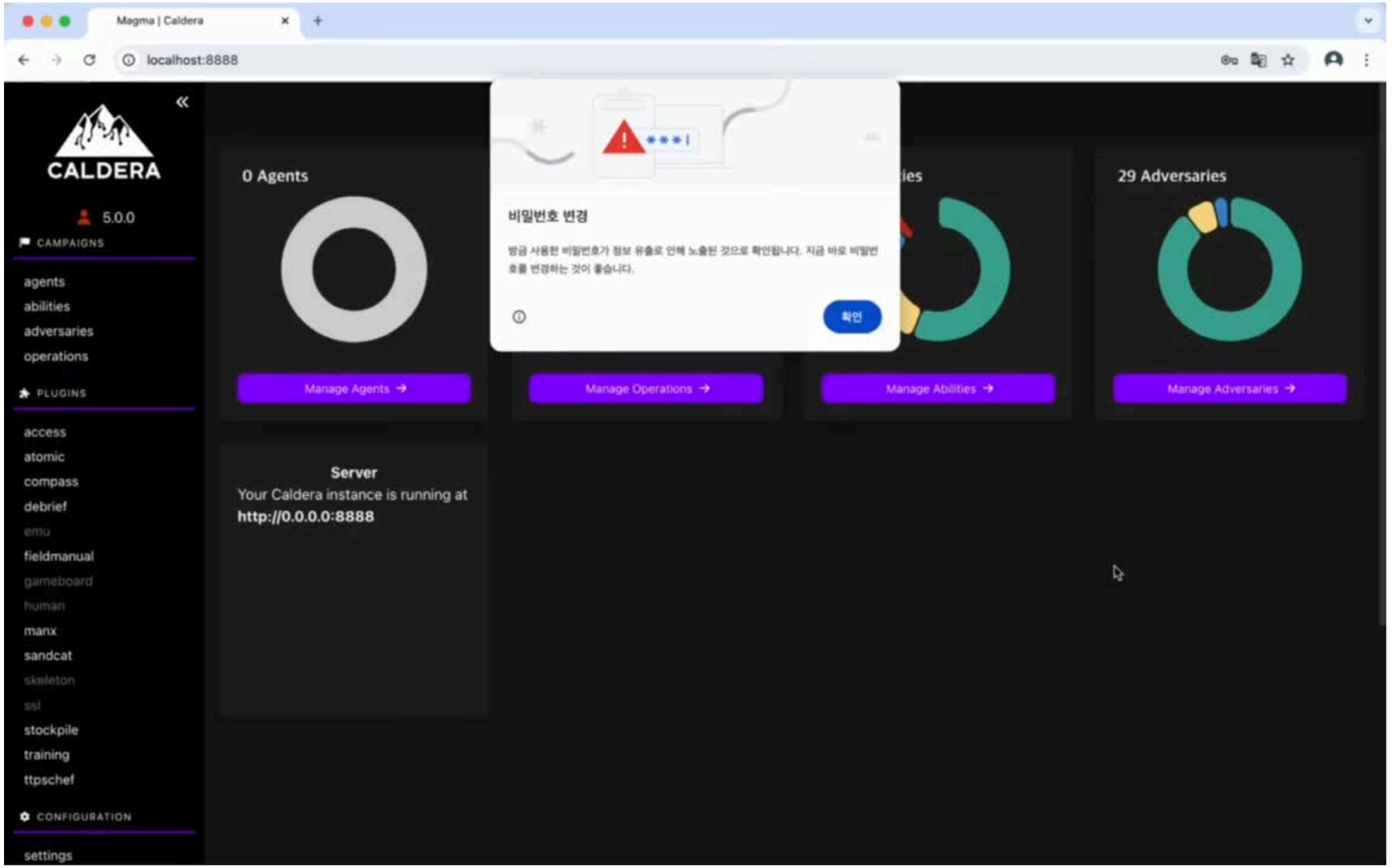
라벨링된 데이터셋 활용

플러그인을 통해 추출한 Technique과
기존 공격 그룹 분석 보고서에 라벨링된 Technique 비교

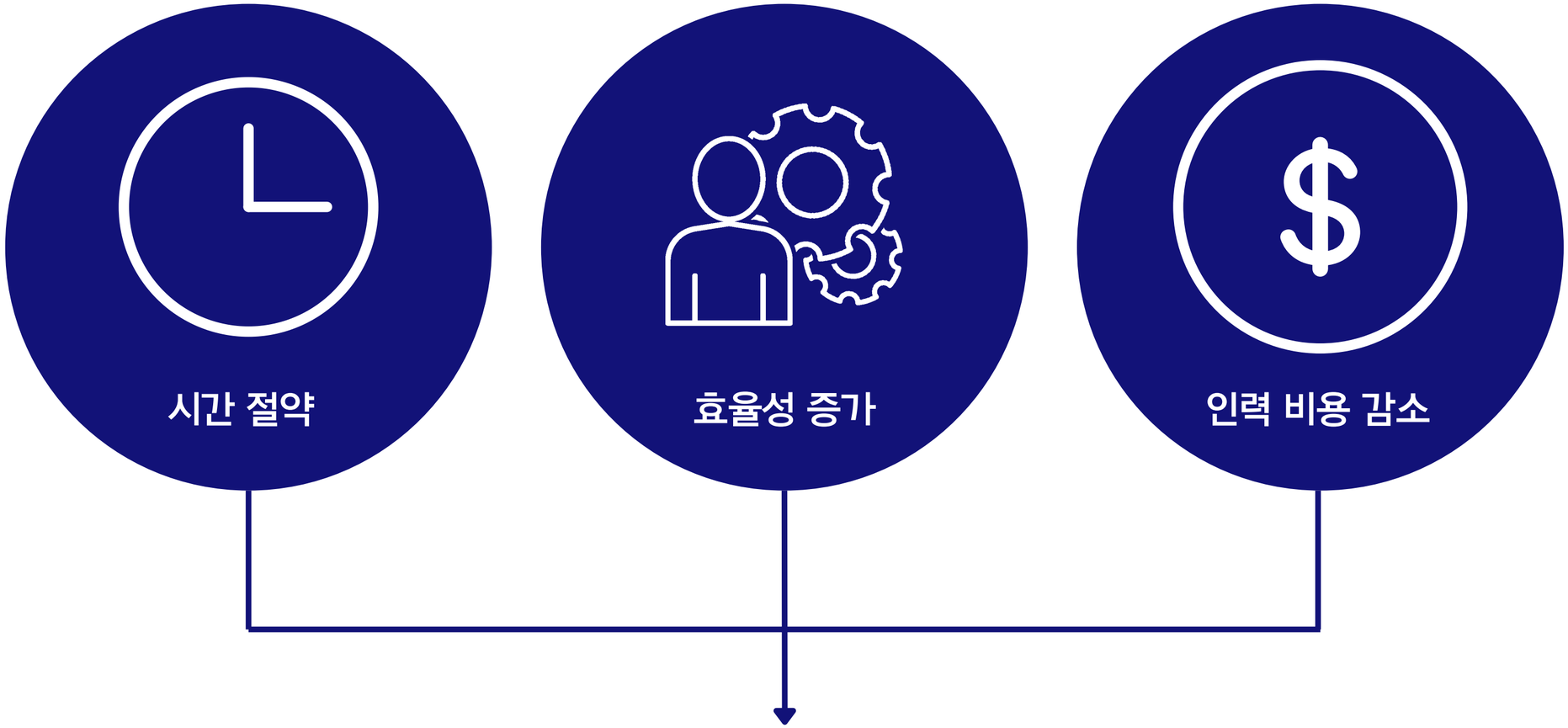
K-Shield Jr.
취약점 분석 과정

- 팀 소개
- 프로젝트 배경
- 프로젝트 목표
- **프로젝트 진행**
- 향후 계획

4. 시연 영상



5. 기대 효과



사이버 위협을 보다 신속하게 대응,
보안 강화에 더 많은 자원 투자

향후 계획

Operation 연동

생성된 시나리오를
실행하는 과정 자동화

과정 간소화

파인튜닝 학습

기존 프롬프트 엔지니어링
학습 방식을
파인튜닝 방식으로 변경

특화 모델 구축

새로운 시나리오 생성

학습 데이터 기반으로
새로운 공격 시나리오 생성

위협의 다양성 대응

감사합니다