

Github를 통해 유포된 VSCode 악용 Contagious Interview 캠페인

enki.co.kr/media-center/blog/contagious-interview-campaign-abusing-vscode-distributed-on-github



요약

- Github에서 VSCode의 자동화 기능을 악용한 악성코드를 확인해 분석하였다.
- 유포된 악성코드는 Contagious Interview 캠페인에 사용되는 Beavertail, InvisibleFerret, OtterCookie로 확인되었다.
- 일부 코드에서 LLM을 활용해 작성된 것으로 추정되는 정황이 확인되었다.
- 공격자는 채용 담당자, 개발자, 가상 기업으로 위장한 계정을 사용하여 신뢰도를 확보하고, 악성코드를 유포한 것으로 확인되었다.
- C&C 서버의 특징을 통해 추가적인 C&C 서버를 확보 및 분석하였다.

1. 개요

최근 Github에서 VSCode(Visual Studio Code)의 자동화 기능을 악용한 악성코드를 다수 확인하였다. 분석 결과 Contagious Interview 캠페인에 사용되는 악성코드로, 최소 2025년 8월부터 캠페인이 시작된 것으로 확인되었다.

Contagious Interview 캠페인은 주로 개발자를 대상으로 악성코드를 유포하는 북한 배후 공격 그룹의 캠페인이다. Contagious Interview 캠페인에서 공격자는 주로 채용 담당자로 위장해 개발자에게 접근하고, 코딩 테스트 혹은 화상 면접 등을 빌미로 Beavertail, InvisibleFerret, OtterCookie 등의 악성코드를 다운로드하도록 유도한다.

이번에 확인된 Github 계정은 실제로 존재하는 기업의 채용 담당자, Web3 개발자, 가상 기업으로 위장한 것으로 확인되었다. 이를 통해 공격자는 정상적인 채용 및 개발 활동으로 위장해 신뢰도를 확보하고, 악성코드 유포를 시도한 것으로 판단된다.

또한 유포에 사용된 악성코드 분석 결과, 사람이 직접 작성한 코드가 아닌 생성형 AI(LLM)를 활용해 작성된 것으로 추정되는 정황이 확인되었다. 이는 공격자가 악성 도구 제작 과정의 효율성을 제고하고, 코드 분석을 어렵게 하기 위해 AI 기술을 적극적으로 활용하고 있음을 시사한다.

본 글에서는 이번 공격에 사용된 악성코드의 동작 방식과 기능을 분석하고, 공격자가 활용한 Github 계정의 활동 내역을 다룬다.

2. 공격자 Github 활동 내역

2.1. 실제 기업 채용 담당자 사칭

Web3 기업인 [Koinos](#)의 채용 담당자를 사칭한 사례로, 코딩 테스트와 관련된 레포지토리에서 악성코드가 확인되었다. 현재 계정 정보를 확인할 수 없지만 [andrew](#)라는 계정을 사용하였고 코딩 테스트를 위해 포크된 레포지토리들이 다수 남아있는 것을 확인할 수 있다.

코딩 테스트 관련 레포지토리

코딩 테스트 관련 레포지토리

모든 레포지토리의 `.vscode/task.json` 파일에 악성코드를 다운로드해 실행하는 명령어가 삽입되어 있다. 삽입된 명령어는 VSCode의 자동화 기능을 악용해 레포지토리를 VSCode로 열었을 때 자동으로 실행된다.

 .vscode/task.json에 삽입된 스크립트

.vscode/task.json에 삽입된 스크립트

현재까지 남아있는 **andrew** 계정이 커밋한 레포지토리는 총 11개로, 커밋 기록 확인 결과 **andrew**는 **andrew_watson**이라는 이름으로도 활동하였고, 이메일은 **andrew@koinos[.]us**를 사용한다.

 커밋 기록 예시

커밋 기록 예시

이메일에 사용된 koinos[.]us 도메인은 사칭한 기업 Koinos의 도메인(koinos.io)과 매우 유사하고, 실제로 레포지토리 이름에 koinos, assessment(평가), dev-test(개발 테스트)와 같은 단어들이 많이 사용된 것으로 보아 실제 악성코드 유포 과정은 알 수 없었지만 채용 절차를 미끼로 기업 채용 담당자를 사칭해 악성코드를 유포한 것으로 판단된다.

2.2. 개발자 위장

개발자로 위장해 공동 작업 중인 레포지토리에 명령어를 삽입한 사례이다. 총 5개의 계정이 확인되었고, 모두 Web3 관련 프로젝트 작업 경험이 있는 풀 스택 개발자로 활동하고 있다. 계정 목록은 아래와 같다.

- ameeetgaikwad
- dannythedawger
- yosket
- nikkhielseath
- jmjles

jmjles 계정 프로필

명령어가 삽입된 레포지토리는 총 10개로, 레포지토리 목록은 아래와 같다.

- Token-Presale-dApp
- Promoting-DApp
- dapp-integration
- Web3-RE-Prototype
- TrustLedger_Fixes
- linkfi
- testtoken
- messageforge
- Token-Presale
- Decentralized-Social

현재 [yosket](#), [nikkhielseath](#) 두 계정은 삭제되었으나 나머지 3개의 계정은 여전히 활성화되어 있다. 명령어는 [.vscode/task.json](#) 파일에 삽입되어 있고, 공동 작업 중인 레포지토리이므로 별도의 감염 유도 없이 악성코드에 감염시킬 수 있다.

 .vscode/task.json에 삽입된 스크립트

.vscode/task.json에 삽입된 스크립트

가장 활발히 활동한 계정은 **yosket**으로 현재는 계정이 삭제되었지만 2026년 1월 16일까지 활동한 기록이 존재한다.

yosket의 2026년 1월 16일 커밋 기록

yosket의 2026년 1월 16일 커밋 기록

또한 **yosket**의 커밋 기록을 통해 해당 계정이 명령어를 삽입한 레포지토리가 현재까지 10개 남아 있는 것으로 확인되었다. 레포지토리 목록은 아래와 같다.

- `hxxps://github[.]com/TrustLedgerLabs/Token-Presale-dApp`
- `hxxps://github[.]com/ryon-business/Promoting-DApp`
- `hxxps://github[.]com/angel-group888/dapp-integration`
- `hxxps://github[.]com/AretaSchmidt/Web3-RE-Prototype`
- `hxxps://github[.]com/QalbeAli/TrustLedger_Fixes`

- `hxxps://github[.]com/trustllabs/Token-Presale-dApp`
- `hxxps://github[.]com/DavidMoura07/linkfi`
- `hxxps://github[.]com/VictorKulagin/testtoken`
- `hxxps://github[.]com/rajaXcodes/Token-Presale-dApp`
- `hxxps://github[.]com/Rochelle128/TokenPresaleDApp`

`yosket` 외에 4개의 계정 모두 `yosket`과 동일한 프로젝트를 진행한 것이 확인되었고,

`messageforge` 레포지토리에서는 2개의 계정이 협력해 악성코드를 유포하려는 모습도 확인되었다.

messageforge 레포지토리, nikkhielseath 계정의 커밋 기록

.vscode/tasks.json 파일에는 악성코드를 다운로드해 실행하는 것 외에도
.vscode/spellright.dict 파일을 nodejs로 실행하는 명령어도 추가되었다.
.vscode/spellright.dict 파일은 obfuscator.io로 난독화된 자바스크립트 파일로 ameeetgaikwad
계정이 추가한 것이 확인되었다.

ameeetgaikwad 계정이 추가한 .vscode/spellright.dict 파일

2.3. 가상 기업 활동

특정 기업의 레포지토리에 명령어가 삽입된 것이 확인되었는데, 확인 결과 홈페이지 구축, 링크드인 계정 생성 등으로 가상 기업이 실제 기업으로 보이도록 위장한 사례이다. 확인된 3개의 레포지토리 모두 명령어가 삽입되어 있다.

가상 기업 veneliteus-dev 계정

3개의 레포지토리 모두 동일한 명령어가 `.vscode/tasks.json` 파일에 삽입되어 있다. 명령어는 긴 공백을 이용해 숨겨져 있고, 모두 2026년 1월 14일에 업로드되었다.

.vscode/tasks.json에 삽입된 악성 스크립트

veneliteus-dev 계정 프로필에 연결된 웹사이트는 현재 아무런 기능도 존재하지 않고, SNS 링크가 4개 있으나 링크드인을 제외한 나머지 3개(페이스북, 인스타그램, X)는 계정 프로필이 아닌 관련 SNS 메인 페이지로 연결되어 있다.

hxxps://veneliteus[.]com

 Venelitus 링크드인

Venelitus 링크드인

또한 홈페이지에서 이메일 입력란에 이메일을 입력해도 **Message sent!**라는 메시지만 나타날 뿐 메일이 오지 않는다. 실제 악성코드 유포 과정은 알 수 없었지만 "2.1. 실제 기업 채용 담당자 사칭"과 유사한 방식으로 악성코드를 유포한 것으로 판단된다.

3. 공격 분석

 공격 개요도

공격 개요도

이번 Contagious Interview 캠페인의 공격 흐름은 두 개로 나뉜다. 대상 환경이나 공격 단계에 따라 Beavertail과 InvisibleFerret을 유포하거나 별도로 OtterCookie를 유포하였다.

3.1. Beavertail, InvisibleFerret

3.1.1. task.json - Downloader

해당 다운로더는 VSCode에 의해서만 실행된다. runOptions의 runOn 옵션이 folderOpen으로 설정되어 있어 레포지토리에 대한 폴더가 VSCode에서 열릴 경우 자동으로 삽입된 명령어가 실행되는 방식이다.

 .vscode/task.json 긴 공백을 통해 숨겨진 명령어

.vscode/task.json 긴 공백을 통해 숨겨진 명령어

삽입된 명령어는 총 3개로 사용자 운영체제(linux, macOS, windows)에 맞는 명령어가 실행된다. 명령어 행위는 C&C 서버에서 스크립트를 다운로드해 실행한다.

 삽입된 명령어

삽입된 명령어

운영체제 별 다운로드하는 url은 다르지만 linux와 macOS 다운로드 스크립트는 동일하다. 또한 모든 스크립트는 공통적으로 새로운 스크립트를 다운로드해 실행한다.

운영체제가 linux, macOS인 경우 `$HOME/Documents/tokenlinux.npl` 파일로 쉘 스크립트를 다운로드한다. 이후 파일 이름을 `tokenlinux.sh`로 변경하고 실행한다.

 linux, macOS 다운로드 스크립트

linux, macOS 다운로드 스크립트

운영체제가 windows인 경우에는 `%USERPROFILE%/parse`로 cmd 스크립트를 다운로드한다. 이후 파일 이름을 `token.cmd`로 변경하고 실행한다.

 windows 다운로드 스크립트

windows 다운로드 스크립트

그러나 분석 당시 windows에서 다운로드하는 cmd 스크립트는 **Access permanently suspended.** 라는 메시지를 에러로 출력해 확보하지 못하였다.

3.1.2. tokenlinux.sh - Downloader

linux, macOS에서 실행되는 tokenlinux.sh 파일은 다운로더로, LLM으로 작성된 것으로 판단된다.

지금까지 Contagious Interview에서 사용된 스크립트와 달리 주석이 상세하게 작성되어 있고, 주석에서 이모티콘을 사용하는 패턴이 확인되어 LLM을 활용해 작성된 것으로 보인다.

 상세하고 이모티콘이 포함된 주석

상세하고 이모티콘이 포함된 주석

tokenlinux.sh는 현재 경로에 nodejs 디렉토리의 존재 여부를 확인하며, 해당 디렉토리가 존재하지 않을 경우 nodejs를 다운로드한 후 압축을 해제한다. 압축 해제 이후 nodejs를 시스템 PATH에 추가하고, C&C 서버에서 `parser.js` 및 `package.json` 파일을 다운로드하여 `$HOME/Documents` 경로에 저장한다.

이후 다운로드된 nodejs를 이용해 `parser.js` 파일을 실행한다. 이때 함께 다운로드된 `package.json`에는 `parser.js` 실행에 필요한 패키지 정보가 포함되어있다.

3.1.3. - parser.js - Downloader

`parser.js`는 obfuscator.io로 난독화가 적용된 다운로더이다. 자바스크립트로 제작되었으며, 적용된 주요 난독화 방식은 아래와 같다.

- 문자열 추출 및 배열화: 코드 내에서 사용되는 주요 문자열을 별도의 배열에 모아 저장한다.
- 실행 시 배열 셔플: 문자열 배열은 코드 실행 초기에 특정 로직에 의해 순서가 재배치된다.
- 오프셋 기반 문자열 접근: 문자열 위치를 직접 참조하지 않고 오프셋 값을 인자로 받는 별도의 함수를 거쳐 배열 내의 문자열을 호출한다.
- 식별자 변조: 의미 있는 변수명과 함수명을 `a1`, `aI`과 같이 무의미하고 짧은 이름으로 변경한다.

난독화 적용 모습

난독화 적용 모습

난독화 외에도 코드 수정 및 포매팅(정렬)되어 있는 경우 더미 코드가 실행되어 에러가 발생하도록 하는 보호 기법이 적용되어 있다.

 보호 기법 관련 함수

보호 기법 관련 함수

2개의 C&C 서버랑 통신하는데 각각 목적은 아래와 같다.

- 정보 수신용 C&C 서버: Beavertail를 다운로드하기 위한 정보(두 번째 C&C 서버 주소 및 캠페인 ID) 수신

- 악성코드 다운로드용 C&C 서버: Beavertail 다운로드

정보 수신용 C&C 서버 주소는 base64로 인코딩되어 있고, 2개의 주소가 존재한다. 만약 첫 번째 정보 수신용 C&C 서버와 통신에 실패하면 두 번째 정보 수신용 C&C 서버와 통신한다.

hxxp://[중간 서버]:1244/s/6df937fe9011 접속 결과가 ZT3로 시작할 경우, 이후의 값을 base64로 디코딩한다. 디코딩된 값은 ,를 기준으로 앞 부분이 악성코드 다운로드용 C&C 서버 주소, 뒷 부분이 캠페인 ID 값이다. 2026년 1월 26일 기준으로 C&C 서버 주소는 66.235.175[.]117, 캠페인 ID는 knHbMe8이다.

 악성코드 다운로드용 C&C 서버 주소 디코딩 루틴

악성코드 다운로드용 C&C 서버 주소 디코딩 루틴

문자열은 xor로 복호화해 사용하고, 이때 키는 0x70a07948이다.

문자열 복호화 루틴

문자열 복호화 루틴

악성코드 다운로드 및 실행 과정은 다음과 같다.

1. 홈 디렉토리에 `.vscode` 폴더 생성
2. `hxxp://[악성코드 다운로드용 C&C 서버]:1244/key`에 호스트 이름, 유저 이름, 현재 시각 전송
3. `hxxp://[악성코드 다운로드용 C&C 서버]:1244/j/[캠페인 ID]`에서 Beavertail 다운로드

저장 경로: `.vscode/test.js`

4. `hxxp://[악성코드 다운로드용 C&C 서버]:1244/p`에서 `test.js`를 동작하기 위한 패키지 파일 다운로드

저장 경로: `.vscode/package.json`

5. `.vscode/test.js` 실행

악성코드 다운로드 루틴은 약 10분마다 반복해 3번 실행된다.

3.1.4. Beavertail

Beavertail 악성코드는 총 3개의 자바스크립트 파일로, 처음 실행되는 `test.js`에서 나머지 2개의 파일을 다운로드해 실행한다. 또한 모든 Beavertail 악성코드는 `parser.js`와 동일한 방식의 난독화와 보호 기법이 적용되어 있다.

`test.js`

브라우저 정보를 수집하고 `n.js`와 `p.js`를 다운로드해 실행한다. 정보를 수집하는 브라우저 목록은 아래와 같다.

- Chrome
- Chromium
- Brave
- Opera

브라우저에 저장된 계정 정보와 브라우저 확장 정보를 수집해

[hxxp://66.235.175\[.\]117:1244/uploads](http://66.235.175[.]117:1244/uploads)로 업로드한다. 이때 정보 수집 `대상 브라우저 확장 목록은 아래 표와 같다.

목록	목록
nkbihfbeogaeaoehlefnkodbefgpgknn	ejbalbakoplchlghecdalmeeeeajnimhm
ibnejdfjmmkpcnlpebklmknkoeiohofec	fhbohimaelfbohpbldcngcnapndodjp
hnfanknocfeofbddgcijnmhnfnkdnaad	bfnaelmomeimhlpmgjnphhpkkoljpa
aeachknmefphepccionboohckonoeemg	egjidjbpglichdcondbcdbnbeppgdph
hifafgmccdpekplomjjkcfgodnhcellj	aholpfdialjgfhomihkjbmgiidlcno
mcohilncbfahbmgdjkbpemcciolgcge	pdliaogehgdbhbnmkklieghmmjkipgpa
nphplpgoakhhjchkkhmiggakijnkhfnd	fdjamakpfbddfdjaoaikfcpapjohcfmg
acmacodkjbdgmoleebolmdjonilkdbch	dmkamcknogkgcdfhbbddcghachkejeap
mkpegjkbkkfacfnmkajcmabijhclg	X

정보 수집 대상 브라우저 확장 목록

업로드가 완료되면 [hxxp://66.235.175\[.\]117:1244/client/knHbMe8](http://66.235.175[.]117:1244/client/knHbMe8)에서 파일을 다운로드해 [.np1](#)로 저장한다. 다운로드한 [.np1](#) 파일은 InvisibleFerret으로, 파이썬을 통해 실행된다. 이때 운영체제가 windows인 경우 [hxxp://66.235.11\[.\]117:1244/pdo](http://66.235.11[.]117:1244/pdo)에서 파이썬을 다운로드해 [.np1](#) 파일을 실행한다.

 npl 파일 다운로드 루틴

.npl 파일 다운로드 루틴

InvisibleFerret 실행 이후 나머지 Beavertail 악성코드를

[hxxp://66.235.111.117:1244/n/knHbMe8](http://66.235.111.117:1244/n/knHbMe8), [hxxp://66.235.111.117:1244/z/knHbMe8](http://66.235.111.117:1244/z/knHbMe8)에서 다운로드해 각각 [.vscode/n.js](#), [.vscode/p.js](#)로 저장하고 실행한다.

 추가 Beavertail 다운로드 루틴

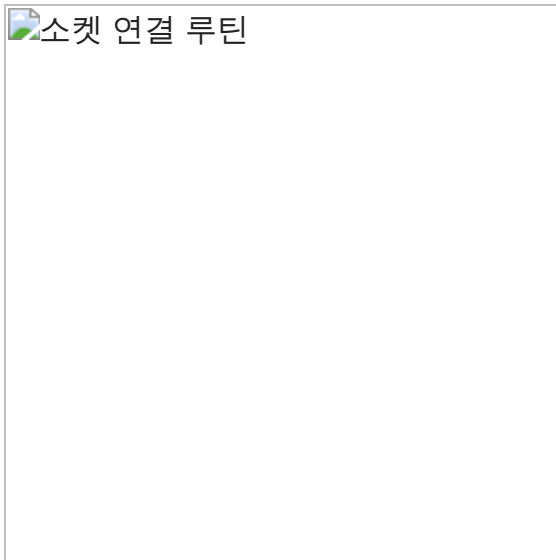
추가 Beavertail 다운로드 루틴

n.js

백도어 악성코드로, 먼저 시스템 정보 수집 이후 [hxxp://66.235.175\[.\]117:1244/keys](http://66.235.175[.]117:1244/keys)에 업로드한다. 수집하는 시스템 정보는 아래와 같다.

- IP를 통해 확인한 위치 정보
- ISP 정보
- 유저 이름을 HMAC으로 다이제스트한 uuid
- 호스트 이름
- 유저 이름
- 운영체제 종류
- 운영체제 릴리스 정보
- 운영체제 버전

업로드 이후 C&C 서버와 소켓 연결을 수립하여 통신한다. 이때 C&C 서버 주소는 216.250.251[.]87이며 포트 번호는 1247이다.



소켓 연결 루틴

연결이 수립되면 소켓을 통해 C&C 서버로부터 code 값을 수신하고, 수신한 code 값에 따라 수행하는 행위는 아래 표와 같다.

code 값	명령 이름	행위
1	ssh_obj	주어진 명령어를 실행한다. (리모트 셸)
2	ssh_cmd	현재 프로세스를 종료한다.
5	ssh_upload	지정된 파일 혹은 디렉토리를 FTP로 업로드한다.
6	ssh_kill	chrome과 brave 프로세스를 강제로 종료한다.
8	ssh_env	조건을 만족하는 파일을 전부 탐색해 FTP로 업로드한다.
10	ssh_mmc	주어진 자바스크립트 파일을 ~/.vscode/c.js로 저장해 실행한다

백도어 행위

파일을 업로드할 때 ftp를 이용하고 사용되는 ftp 서버의 도메인, 유저 이름, 패스워드는 C&C 서버에서 전송한다. 업로드 경로는 /DAknHbMe8/[호스트 이름]+[유저 이름]/[파일 이름]이다.

FTP 업로드 루틴

FTP 업로드 루틴

ssh_env 명령은 경로에 아래 표에 포함된 문자열이 포함된 파일을 탐색해 업로드한다.

목록	목록
.env	.xls
config.js	.xlsx
secret	.doc
metamask	.docx
wallet	.rtf
private	.kbdx
mnemonic	.one
password	.onenote
account	solana
seed	X

탐색 문자열 목록

이때 **tsconfig.json**과 같이 특정 이름 혹은 확장자를 가진 파일은 업로드하지 않고, node_modules와 같은 특정 폴더는 탐색하지 않는다. 자세한 목록은 "부록 C. 탐색 조건" 참고 바란다.

p.js

파일 수집 및 업로드하는 악성코드이다. 아래 표에 포함된 패턴을 만족하는 경로의 파일을 탐색해 업로드한다.

목록	목록
.env	seed
config.js	keys*
secret*	keypair.json
metamask*	1pass*.sqlite
wallet*	notes.txt
private*	hardhat.config.ts
mnemonic*	solana*
password*	.kdbx
account*	X

탐색 패턴 목록

n.js와 동일하게 특정 경로를 탐색하지 않고, 특정 확장자를 가진 파일은 업로드하지 않는다. 자세한 목록은 "부록 C. 탐색 조건" 참고 바란다.

업로드 파일 탐색 루틴

업로드 파일 탐색 루틴

파일 업로드는 `hxxp://66.235.175[.]117:1244/uploads`에 아래와 같은 구조의 POST 요청을 보내 업로드한다.

```
{
  "timestamp": "jzt",
  "type": "knHbMe8",
  "hid": "[호스트 이름]",
  "multi_file": {
    "value": "[파일 스트림]",
    "options" : {
      "filename":
        "env_[현재 시각]_[0~40의 랜덤 숫자]_[파일 인덱스]"
    }
  }
}
```

위 구조에서 파일 인덱스는, 탐색 패턴으로 매칭해 생성한 파일 리스트 내부에서 해당 파일의 순서를 의미한다. 파일 업로드가 완료되면 파일 경로, 호스트 이름을 아래와 같은 구조로 `hxxp://66.235.175[.]117:1244/keys`에 업로드한다.

```
{
  "ts": "jzt",
  "type": "knHbMe8",
  "hid": "[호스트 이름]",
  "ss": "jzt_[탐색 패턴]_[현재 시각]",
  "cc": "env_[현재 시각]_[0~40의 랜덤 숫자]_[파일 인덱스] : [파일 경로]"
}
```

3.1.5. InvisibleFerret

InvisibleFerret은 파이썬으로 작성된 악성코드이며 5개의 파이썬 스크립트로 이루어져 있다. 모든 InvisibleFerret 스크립트는 다음과 같은 방식으로 난독화되어있다.

1. 난독화된 문자열에서 첫 8 글자 이후의 문자열 base85로 디코딩한다.
2. 디코딩된 결과를 사용하지 않은 첫 8 글자와 xor한 후 exec으로 실행한다.
3. exec으로 실행된 문자열은 다시 새로운 문자열을 뒤집고 base64 디코딩, zlib 압축 후 exec으로 실행한다.
4. 3번 과정을 64번 가량 반복하면 실제 악성코드 스크립트가 실행된다.

.npl

Beavertail에서 실행한 InvisibleFerret 스크립트로 2개의 추가 스크립트를 다운로드해 실행한다.

스크립트 다운로드 및 실행 루틴

스크립트 다운로드 및 실행 루틴

[hxxp://66.235\[.\]175.117:1244/pay1/knHbMe8](http://66.235[.]175.117:1244/pay1/knHbMe8)에서 파일을 다운로드해 [.vscode/pay](#)로 저장한 뒤 실행하고, [hxxp://66.235\[.\]175.117:1244/bro/knHbMe8](http://66.235[.]175.117:1244/bro/knHbMe8)에서 파일을 다운로드해 [.vscode/bow](#)로 저장한 뒤 실행한다. 이때 운영체제가 Darwin인 경우 bow 파일은 다운로드하지 않는다.

pay

C&C 서버에서 소켓 연결을 통해 수신한 code 값에 따라 악성 행위를 수행하는 백도어 악성코드이다. C&C 서버 주소는 216.250.251[.]87 이고 포트 번호는 1245이다.

 소켓 연결 루틴

소켓 연결 루틴

수신한 code 값에 따라 수행하는 행위는 아래 표와 같다.

code	이름	행위
1	ssh_obj	주어진 명령어를 실행한다.
2	ssh_cmd	자가 삭제를 실행한다.
3	ssh_clip	키로깅/클립보드 로깅 데이터를 전송한다.
4	ssh_run	hxxp://66.235.175[.]117:1244/bro/knHbMe8에서 ~/.n2/bow로 파일을 다운로드하여 실행한다.
5	ssh_upload	주어진 경로의 파일 업로드한다.
6	ssh_kill	Chrome, Brave 브라우저의 프로세스를 종료한다.
7	ssh_any	hxxp://66.235.175[.]117:1244/adc/knHbMe8에서 ~/.n2/adc로 파일을 다운로드하여 실행한다.
8	ssh_env	조건을 만족하는 파일을 전부 탐색하여 FTP로 업로드한다.
9	ssh_zcp	주어진 경로의 파일을 암호화된 압축파일로 압축하고 텔레그램과 FTP로 업로드한다.
10	ssh_mmc	hxxp://66.235.175[.]117:1244/dd/knHbMe8에서 ~/.n2/mc로 파일을 다운로드하여 실행한다.

백도어 행위

ssh_env 명령은 경로에 아래 표에 포함된 문자열이 포함된 파일을 탐색해 업로드한다.

목록	목록
env	.xls
config.js	.xlsx
secret	.doc
metamask	.docx
wallet	.rtf
private	.kidx
mnemonic	.one

password .onenote

account X

자세한 목록은 "부록 C. 탐색 조건" 참고 바란다.

bow

TsunamiInjector 스크립트를 난독화한 후 startup 폴더에 저장하여 스크립트가 지속적으로 실행되도록 설정한다.

 스크립트 저장 루틴

스크립트 저장 루틴

TsunamiInjector 스크립트는 약 1000 여 개의 암호화된 url을 복호화해 방문하고, 내용이 존재하는 url을 탐색한다. 복호화 과정은 아래와 같다.

- 16 진수로 인코딩된 문자열을 디코딩한다.
- 디코딩된 문자열을 **!!!HappyPenguin1950!!!**로 xor한다.
- 복호화된 문자열을 base64로 디코딩한 후 순서를 뒤집는다.

URL 복호화 루틴

URL 복호화 루틴

복호화된 url은 모두 텍스트 저장 사이트인 pastebin의 url이며 내용이 남은 pastebin url을 발견한 경우 해당 페이지의 내용을 위에서 한 것과 같은 방식으로 복호화한다. 복호화된 결과는 또 다른 url로 해당 url에서 파일을 다운로드해 %APPDATA%\Microsoft\Windows\Applications\Runtime Broker.exe로 압축 해제한다

파일 다운로드 루틴

파일 다운로드 루틴

파일 다운로드가 완료되면 해당 파일 경로를 windows 디펜더 검사 경로에서 제외하고, 유저가 로그인할 때마다 실행되도록 작업을 생성한다.

작업 생성 루틴

작업 생성 루틴

adc

[hxxp://66.235.175\[.\]117:1244/and](http://66.235.175[.]117:1244/and)에서 원격 데스크톱 제어 프로그램인 AnyDesk를 다운로드한다. 이후 다운로드한 AnyDesk의 설정을 업데이트해 공격자의 계정으로 언제든지 원격 접속이 가능하도록 변경한다.

mc

브라우저 환경을 조작해 최종적으로 암호화폐 지갑 브라우저 확장을 설치한다. 실행 시 먼저 자가 삭제를 한다.



자가 삭제 루틴

이후 Chrome 및 Brave 브라우저의 확장 프로그램 디렉토리를 탐색해, 특정 확장 프로그램의 파일을 모두 삭제한다. 삭제 대상 확장 프로그램의 ID 목록은 아래와 같다.

- nkbihfbeogaeaoehlefnkodbefgpgknn
- acmacodkjbdgmoleebolmdjonilkdbch

확장 파일 삭제가 완료되면, [http://45.59.163\[.\]55:1244/mmz/\[브라우저 확장 ID\]_knHbMe8](http://45.59.163[.]55:1244/mmz/[브라우저 확장 ID]_knHbMe8)에서 파일을 다운로드해 삭제한 기존 확장 프로그램 디렉토리에 압축을 해제한다. 이 과정에 변경되는 브라우저 확장은 Rabby Wallet 및 MetaMask로, 모두 암호화폐 지갑과 관련된 확장 프로그램이다.

브라우저 확장 파일 변경 루틴

브라우저 확장 파일 변경 루틴

이후 악성코드는 Chrome 브라우저의 프로필 설정 파일을 수집해 `hxxp://45.59.163[.]55:1244/h`로 전송한다. 서버에서 정상적인 응답을 수신할 경우, 응답 값을 이용해 Chrome의 Secure Preferences 설정을 조작한다.

추가적으로 macOS 환경에서 Chrome의 메이저 버전이 140 이상인 경우 `hxxp://45.59.163[.]55:1244/ddo`에서 파일을 다운로드해 `/Applications/Google Chrome.app` 경로에 덮어쓰는 방식으로 Chrome 다운그레이드를 시도한다

크롬 다운그레이드 루틴

크롬 다운그레이드 루틴

설정 변경이 완료되면, 변경 사항을 즉시 반영하기 위해 브라우저에 저장된 쿠키, 캐시, 세션 데이터 등을 모두 삭제하고 성공 여부를 `hxxp://66.235.175[.]117:1244/t`로 전송한다.

3.2. OtterCookie

3.2.1. task.json - Downloader

해당 다운로더는 VSCode에 의해서만 실행된다. runOptions의 runOn 옵션이 folderOpen으로 설정되어 있어 레포지토리에 대한 폴더가 VSCode에서 열릴 경우 자동으로 삽입된 명령어가 실행되는 방식이다. Beavertail, InvisibleFerret 유포에 사용된 task.json 파일과 다르게 명령어가 숨겨져 있지 않다.

 .vscode/tasks.json에 삽입된 명령어

.vscode/tasks.json에 삽입된 명령어

운영체제 별 다운로드하는 url은 다르지만 linux와 macOS 다운로드 스크립트는 동일하다. 또한 모든 스크립트는 공통적으로 새로운 스크립트를 다운로드해 실행한다.

운영체제가 linux, macOS인 경우 `$HOME/.vscode/vscode-bootstrap.sh` 파일로 쉘 스크립트를 다운로드하고 실행한다.

 linux, macOS에서 실행되는 셸 스크립트

linux, macOS에서 실행되는 셸 스크립트

운영체제가 windows인 경우 %USERPROFILE%\vscode\vscode-bootstrap.cmd 파일로 cmd 스크립트를 다운로드하고 실행한다.

 windows에서 실행되는 cmd 스크립트

windows에서 실행되는 cmd 스크립트

그러나 분석 당시 windows에서 다운로드하는 cmd 스크립트는 `Access permanently suspended.` 라는 메시지를 에러로 출력해 확보하지 못하였다.

3.2.2. vscode-bootstrap.sh - Downloader

linux, macOS에서 실행되는 `vscode-bootstrap.sh` 파일은 다운로더로, `tokenlinux.sh`와 동일하게 LLM으로 작성된 것으로 판단된다. 다른 점은 이모티콘이 사용되지 않았다.

 상세한 주석이 작성된 쉘 스크립트

상세한 주석이 작성된 쉘 스크립트

`vscode-bootstrap.sh`는 현재 경로에 `nodejs` 디렉토리의 존재 여부를 확인하며, 해당 디렉토리가 존재하지 않을 경우 `nodejs`를 다운로드한 후 압축을 해제한다. 압축 해제 이후 C&C 서버에서 `env-setup.js` 및 `package.json` 파일을 다운로드해 `$HOME/.vscode` 경로에 저장한다. 이후 다운로드된 `nodejs`를 이용해 `env-setup.js`를 실행한다.

3.2.3. env-setup.js - Downloader

`env-setup.js`는 `https://y-lilac-sigma.vercel[.]app/api/ipcheck-encrypted/608`에 접속해 응답을 `eval`로 실행하는 다운로더 이다. 특이한 것은 정상 응답이 왔을 때 실행하지 않고, 에러가 발생하였을 때 실행한다.

 env-setup.js 코드

env-setup.js 코드

3.2.4. OtterCookie

`env-setup.js`를 통해 다운로드 및 실행되고, `Beavertail`과 동일한 방식의 난독화와 보호 기법이 적용되어 있다.

OtterCookie는 문자열로 저장된 3개의 스크립트를 각각 독립된 프로세스에서 실행한다. 이때 lock 파일에 { pid: [프로세스 PID], startedAt: [프로세스 시작 시각] } 내용을 작성해 스크립트가 중복 실행되지 않도록 한다. lock 파일 이름과 스크립트 정보는 아래 표와 같다.

lock 파일 이름	스크립트 이름	행위
pid.6.1.lock	ldbScript	브라우저에 저장된 정보 탈취한다.
pid.6.2.lock	autoUploadScript	특정 파일을 찾아 업로드한다.
pid.6.3.lock	socketScript	C&C 서버에 접속하여 명령에 따라 행위를 수행한다

lock 및 스크립트 정보

ldbScript

브라우저에 저장된 계정 정보와 브라우저 확장 정보를 수집하는 스크립트이다. 정보 수집 대상 브라우저 목록은 아래 표와 같다.

목록	목록
Chrome	Kiwi
Brave	Yandex
AVG Browser	Iridium
Edge	Comodo
Opera	SRWare
Opera GX	Chromium
Vivaldi	X

정보 수집 대상 브라우저 목록

정보 수집 대상 브라우저 확장 목록은 아래 표와 같다.

목록	목록
1nkbihfbeogaeaoehlefnkodbefgpgknn	fhkbpkphfeanlhnlfkologfoccekhic
ejbalbakoplchlghecdalmeeeajnimhm	fhmfendgdocmcbmfikdcogofphimnkno
acmacodkjbdgmoleebolmdjonilkbdbch	fldfpgipfncgndfolcbkdeeknbbsnhcc
bfnaelmomeimhlpmgjnjojphhpkkoljpa	gjnckgkfmgmibbkoficdidcljeaaaheg

ibnejdfjmmkpcnlpebklnkoeiohofec	hifafgmccdpekplomjjkcfgodnhcellj
egjidjbpglichdcondbcdbdnbeppgdph	hmeobnfnfcmkdcmlblgagmfpfboieaf
nphplpgoakhhjchkkhmiggakijnkhfnd	hnfanknocfeofbddgcijnmhnfnkdnaad
omaabbefbmijjedngplfjmnooppbclkk	jiidiaalihmmhddjgbnbgsdffeolcpak
bhhhlbepdkbapadjdnnojkbgioiodbic	jbIndlipeogpafnldhgmapagccccfchpi
aeachknmefphepccionboohckonoeemg	jmbkjchcobfffnmjboflnchcbljiljdk
aflkmhkijdbfcmhplgifokgdeclgpoi	jnjpmcgfcfeffkfgcnjefkbkgcpnkpab
agoakfejjabomempkjlepdlaleeobhb	kpkmbkboifcfpapmleipncofdbjdvice
aholpfdialjgjfhomihkjbmgiidlcno	khpkpbbcccdmmclmpigdgddabeilkdpd
afbcbjpbpfadlkmhmcLhkeeodmamcflc	ldinpeekobnhjjdofggfgjlcehhmanaj
cgbogdmdefihhljhfeffkljbghamglni	lgmpcpglpngdoalbgeoldeaifclnhafa
dmkamcknogkgcdfhhbddcghachkejeap	mcohilncbfahbmgdjkbpemcciiolgce
dlcobpjigpikoobohmabehhmhfoodbb	mopnmbcafieddcaagdcbnhejhlodfdd
efbglgofoippbgcjepnhiblaibcnclgk	nkklfkfpelghghbidbnpdfhblphpfjmbo
ejjladinnckdgjemekebdpeokbikhfci	penjlddjkgjpnkllboccdgccekpkcbin
fhbohimaelbohpbjbbldcngcnapndodjp	ppbibelpcjmhbdihakflkdcoccbgbkpo

정보 수집 대상 브라우저 확장 목록
autoUploadScript

모든 드라이브 대상으로 특정 문자열이 포함된 파일을 탐색한 뒤

[hxxp://172.86.73\[.\]198:8086/upload](http://172.86.73[.]198:8086/upload)로 업로드한다. 탐색에 사용되는 문자열은 "부록 C. 탐색 조건" 참고 바란다.

파일 업로드 루틴

파일 업로드 루틴

이때 `ldbScript`의 파일 업로드 방식과 달리 파일 메타데이터가 존재하지 않고, HMAC 토큰 생성에 파일 경로를 사용한다.

`autoUploadScript` 또한 주석이 상세하게 적혀있고, 사용자에게 보여주지 않는 콘솔 출력을 자주 한다는 점 등의 특징이 LLM을 통해 작성된 것으로 보인다.

socketScript

시스템 정보를 수집하고 `hxxp://172.86.73[.]198:8087/api/notify`로 전송한다. 수집하는 시스템 정보와 전송 형식은 아래와 같다.

```
{
  ukey: 608,
  t: 6,
  host: "608_[호스트 이름]",
  os: "[OS 종류]",
  username: "[유저 이름]",
  timestamp: "[현재 시간]"
}
```

이후 C&C 서버로 소켓 연결을 수립한다. C&C 서버 주소는 172.86.73[.]198 이고 포트 번호는 8087 이다. 소켓을 통해 C&C 서버에서 수신한 메시지에 따라 수행하는 행위는 아래 표와 같다.

메시지	행위
whour	앞서 수집한 시스템 정보를 전송한다.
command	주어진 명령어를 실행한다.
disconnect	소켓 서버와 연결 종료하고 대기한다.
reconnect	소켓 서버와 다시 연결 수립한다.
processControl	OtterCookie의 다른 프로세스를 조작한다

메세지에 따라 수행하는 행위

`processControl` 기능은 `ldbScript`, `autoUploadScript`, `socketScript`를 멈추거나 다시 시작할 수 있도록 설계되어 있으나 멈추는 기능만 구현되어 있다.

미완성된 processControl 기능

미완성된 processControl 기능

`socketScript` 또한 주석이 상세하게 적혀있고, 사용자에게 보여주지 않는 콘솔 출력을 자주 한다는 점 등의 특징이 LLM을 통해 작성된 것으로 보인다.

4. 추가 C&C 서버 확보

Beavertail, InvisibleFerret의 C&C 서버는 공통적으로 아래와 같은 특징을 가지고 있다.

1. 1244번 포트에 HTTP 서버가 존재한다.
2. 21(FTP), 3389(RDP) 포트가 활성화되었다.
3. 주로 VPS 호스팅 업체를 사용해 서버를 운영한다.
4. `hxxp://[C&C 서버]:1244/p`에 접속하면 `package.json` 파일을 응답한다.

위 특징 1, 2, 3을 이용해 첫 번째로 [Censys](#)에서 관련 IP 18개를 찾을 수 있었다.

Censys 검색 결과

두 번째로 확보한 IP에 대해 `hxxp://[IP]:1224/p` 접속 결과를 확인하였고, 최종적으로 10개의 C&C 서버 주소를 식별하였다. C&C 서버 주소 목록은 아래 표와 같다.

서버 IP	캠페인 ID	호스팅 정보
216.250.251[.]211	X	인도의 호스팅 업체 Majestic Technosoft의 VPS 서버
45.59.163[.]23	X	미국의 호스팅 업체 Tier.Net의 VPS 서버
38.92.47[.]152	X	미국의 호스팅 업체 Tier.Net의 VPS 서버
103.65.230[.]100	X	불가리아의 호스팅 업체 EuroHoster의 VPS 서버
147.124.213[.]19	X	인도의 호스팅 업체 Majestic Technosoft의 VPS 서버
66.235.175[.]117	knHbMe8 ryGnMe8	미국의 호스팅 업체 Tier.Net의 VPS 서버
66.235.175[.]109	ryGnMe8	미국의 호스팅 업체 Tier.Net의 VPS 서버
147.124.202[.]225	X	미국의 호스팅 업체 Tier.Net의 VPS 서버
67.203.7[.]205	ZRj2s0k9	미국의 호스팅 업체 Colocation America의 VPS 서버
147.124.213[.]232	kmHgMq7	인도의 호스팅 업체 Majestic Technosoft의 VPS 서버

추가 확보 C&C 서버 목록

C&C 서버는 모두 VPS 호스팅을 이용하였고, 호스팅 업체가 달라도 IP 위치 정보는 미국으로 확인되었다. 또한 130.65.230[.]100, 66.235.175[.]117, 66.235.175[.]109를 제외한 7개의 IP에는 445번 포트(SMB)가 활성화되어 있다.

이외에 대부분의 C&C 서버는 잘못된 캠페인 ID를 이용해 요청을 전송하면 응답이 오지 않았지만 66.235.175[.]109, 38.92.47[.]152, 45.59.163[.]23 3개의 C&C 서버에서는 응답이 왔다.

38.92.47[.]152는 에러 메시지가 응답으로 오고, 66.235.175[.]109, 45.59.163[.]23는 요청으로 보낸 잘못된 캠페인 ID를 응답으로 보냈다.

OtterCookie의 C&C 서버는 방탄 호스팅을 사용해 보호받고 있어 추가 C&C 서버 확보를 못하였다

5. 마치며

본 글에서는 Github에서 VSCode의 자동화 기능을 악용해 유포된 Contagious Interview 캠페인을 분석하였다. 최종 단계에서 실행되는 악성코드는 Beavertail, InvisibleFerret, OtterCookie로 확인되었으며, 감염된 시스템으로부터 브라우저 계정 정보 및 암호화폐 지갑 관련 파일 등 민감 정보를 탈취하고, C&C 서버로부터 명령을 수신하여 원격 제어, 추가 악성코드 다운로드 등의 악성 행위가 확인되었다.

VSCode 기능을 악용한 공격은 프로젝트를 여는 것만으로도 일부 개발 도구가 설정에 따라 코드를 자동 실행할 수 있으므로 지속적인 주의가 필요하다. 특히 VSCode 사용 시 Workspace Trust 기능이 활성화되어 있는지 확인하고 (`security.workspace.trust.enabled`, 기본값 `true`), 신뢰 부여 전 편집기 설정과 워크스페이스 구성을 사전에 점검하는 것이 권고된다. 이때 `.vscode` 디렉터리, 사용 중인 툴체인, 확장(Extension) 설정을 우선적으로 감사하여 자동 실행 경로 및 예기치 않은 동작 가능성을 최소화할 필요가 있다.

특히 이번 캠페인은 생성형 AI(LLM)를 활용해 악성 스크립트를 제작함으로써 공격 도구 개발의 효율성을 높였으며, 실제 기업의 채용 담당자 사칭과 일반 개발자로 위장한 오픈소스 프로젝트 기여 등 사회공학적 기법이 확인되었다. 아울러 분석 과정에서 공격자가 운영하는 다수의 추가 C&C 서버를 식별하였으며, 이를 통해 공격 인프라가 지속적으로 확장 및 운영되고 있는 정황이 확인되었다.

현재까지도 Contagious Interview 캠페인이 관측되고 있으며, 공격자는 Github를 중심으로 한 오픈소스 생태계와 개발자를 공격 대상으로 삼고 있다. 특히 채용 과제, 인터뷰용 프로젝트, 테스트 코드 등으로 위장한 저장소에 개발 도구의 자동화 기능을 악용하는 공격 방식은, 캠페인 공격 방식이 지속적으로 발전하고 있는 사실을 뒷받침한다.

6. 부록

부록 A. MITRE ATT&CK

Tactics	Techniques
Initial Access	T1566: Phishing T1195.001: Supply Chain Compromise: Compromise Software Dependencies and Development Tools
Execution	T1059: Command and Scripting Interpreter
Defense Evasion	T1027: Obfuscated Files or Information T1562.001: Disable or Modify Tools T1070.004: Indicator Removal: File Deletion
Persistence	T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder T1053.005: Scheduled Task/Job: Scheduled Task

Discovery	T1082: System Information Discovery T1083: File and Directory Discovery T1518: Software Discovery T1057: Process Discovery
Credential Access	T1555.003: Credentials from Password Stores: Credentials from Web Browsers T1056.001: Input Capture: Keylogging
Collection	T1115: Clipboard Data
Command and Control	T1071.001: Application Layer Protocol: Web Protocols T1071.002: Application Layer Protocol: File Transfer Protocols T1219.002: Remote Access Software: Remote Desktop Software T1102.001: Web Service: Dead Drop Resolver T1571: Non-Standard Port
Exfiltration	T1048: Exfiltration Over Alternative Protocol T1041: Exfiltration Over C2 Channel

MITRE ATT&CK

부록 B. IOCs

sha256

sha256	파일명
748b044a62787bf8341ac46208ac5c7703b0cf6d0ffdd2c76160244247cd930	tokenlinux.sh
2f94573c15bfa1f945a0e1116b2373542d7a1dbc12dd58eb612078ddafb54b	parser.js
a51b6b30d871acf43539ef4f19bfd1954a5db3f8422ed91fc14519ef60431296	test.js
7e6ae216310d938218ff4fdf8bfe2e9a73d7684188705b97232e968022cc6b46	n.js
50b14f2ea38cb7a388917394e8906712caf67acaf1c60ac6fab71072ef1eca3b	p.js
0f0b20e88511ae2014b0c90ed2dbbcaad7e5d9593b50fdd43081f24cb0d05534	pay
9310f180e417a6cd5b505cdec0a726a9f0c39519687491d22bd98d917e0139f8	bow
8e142c55f0af4961b59cb0dae027eeb256d173ba094afdf66d8fbf047f014169	adc
572a75f932aec2bcbe5eb4685e399e04688c94dd44a75db92d51618467a0d18	mc
964885a574aa6287a55bfe6fdb1512d531b03c9e15609fe6782afbb0116f6ba	vscode-bootstrap.sh
1f97f7a0614685f2e8cd8347bf10cd7832b5dd9dce96e195e15da9b9b2be2e3e	env-setup.js

d7c36881f91c45b66b32ec0c80f2d49ff90d76d2d147f2d208c6eea48842618c
sha256

OtterCookie

ip

- 216.250.251[.]211
- 216.250.251[.]87
- 45.59.163[.]23
- 45.59.163[.]55
- 38.92.47[.]152
- 103.65.230[.]100
- 147.124.213[.]19
- 66.235.175[.]117
- 66.235.175[.]109
- 147.124.202[.]225
- 67.203.7[.]205
- 147.124.213[.]232
- 172.86.73[.]198

URL

- [https://vscode-load-config.vercel\[.\]app/settings/mac?flag=4](https://vscode-load-config.vercel[.]app/settings/mac?flag=4)
- [https://vscode-settings-config.vercel\[.\]app/settings/windows?flag=8](https://vscode-settings-config.vercel[.]app/settings/windows?flag=8)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/linux?flag=306](https://vscode-settings-bootstrap.vercel[.]app/settings/linux?flag=306)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/windows?flag=4](https://vscode-helper171-ruby.vercel[.]app/settings/windows?flag=4)
- [https://www.vscodeconfig\[.\]com/settings/mac?flag=1](https://www.vscodeconfig[.]com/settings/mac?flag=1)
- [https://www.vscodeconfig\[.\]com/settings/mac?flag=3](https://www.vscodeconfig[.]com/settings/mac?flag=3)
- [https://vscodeconfig03kui.vercel\[.\]app/api/settings/mac](https://vscodeconfig03kui.vercel[.]app/api/settings/mac)
- [https://www.vscodeconfig\[.\]com/settings/windows?flag=3](https://www.vscodeconfig[.]com/settings/windows?flag=3)

- [https://vscode-helper171.vercel\[.\]app/settings/linux?flag=4](https://vscode-helper171.vercel[.]app/settings/linux?flag=4)
- [https://vscode-load-config.vercel\[.\]app/settings/linux?flag=4](https://vscode-load-config.vercel[.]app/settings/linux?flag=4)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/linux?flag=301](https://vscode-settings-bootstrap.vercel[.]app/settings/linux?flag=301)
- [https://www.vscodeconfig\[.\]com/settings/mac?flag=4](https://www.vscodeconfig[.]com/settings/mac?flag=4)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/mac?flag=302](https://vscode-settings-bootstrap.vercel[.]app/settings/mac?flag=302)
- [https://www.vscodeconfig\[.\]com/settings/linux?flag=1](https://www.vscodeconfig[.]com/settings/linux?flag=1)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/linux?flag=6](https://vscode-helper171-ruby.vercel[.]app/settings/linux?flag=6)
- [https://vscode-toolkit-bootstrap.vercel\[.\]app/settings/windows?flag=306](https://vscode-toolkit-bootstrap.vercel[.]app/settings/windows?flag=306)
- [https://vscode-settings-config.vercel\[.\]app/settings/linux?flag=606](https://vscode-settings-config.vercel[.]app/settings/linux?flag=606)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/windows?flag=306](https://vscode-settings-bootstrap.vercel[.]app/settings/windows?flag=306)
- [https://vscode-load.onrender\[.\]com/settings/linux?flag=5](https://vscode-load.onrender[.]com/settings/linux?flag=5)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/windows?flag=302](https://vscode-settings-bootstrap.vercel[.]app/settings/windows?flag=302)
- [https://vscode-toolkit-bootstrap.vercel\[.\]app/settings/mac?flag=306](https://vscode-toolkit-bootstrap.vercel[.]app/settings/mac?flag=306)
- [https://vscode-settings-config.vercel\[.\]app/settings/mac?flag=8](https://vscode-settings-config.vercel[.]app/settings/mac?flag=8)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/mac?flag=3](https://vscode-helper171-ruby.vercel[.]app/settings/mac?flag=3)
- [https://vscode-load-config.vercel\[.\]app/settings/linux?flag=1](https://vscode-load-config.vercel[.]app/settings/linux?flag=1)
- [https://www.vscodeconfig\[.\]com/settings/windows?flag=4](https://www.vscodeconfig[.]com/settings/windows?flag=4)
- [https://vscode-helper-132.vercel\[.\]app/settings/windows?flag=4](https://vscode-helper-132.vercel[.]app/settings/windows?flag=4)
- [https://vscode-helper-132.vercel\[.\]app/settings/linux?flag=4](https://vscode-helper-132.vercel[.]app/settings/linux?flag=4)
- [https://vscode-settingstask.vercel\[.\]app/api/settings/windows](https://vscode-settingstask.vercel[.]app/api/settings/windows)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/mac?flag=306](https://vscode-settings-bootstrap.vercel[.]app/settings/mac?flag=306)
- [https://vscode-load-config.vercel\[.\]app/settings/windows?flag=1](https://vscode-load-config.vercel[.]app/settings/windows?flag=1)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/linux?flag=302](https://vscode-settings-bootstrap.vercel[.]app/settings/linux?flag=302)
- [https://vscode-load.onrender\[.\]com/settings/windows?flag=5](https://vscode-load.onrender[.]com/settings/windows?flag=5)

- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/mac?flag=308](https://vscode-settings-bootstrap.vercel[.]app/settings/mac?flag=308)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/windows?flag=6](https://vscode-helper171-ruby.vercel[.]app/settings/windows?flag=6)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/windows?flag=301](https://vscode-settings-bootstrap.vercel[.]app/settings/windows?flag=301)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/windows?flag=305](https://vscode-settings-bootstrap.vercel[.]app/settings/windows?flag=305)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/mac?flag=305](https://vscode-settings-bootstrap.vercel[.]app/settings/mac?flag=305)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/linux?flag=308](https://vscode-settings-bootstrap.vercel[.]app/settings/linux?flag=308)
- [https://codeviewer-three.vercel\[.\]app/task/windows?token=6df937fe9011](https://codeviewer-three.vercel[.]app/task/windows?token=6df937fe9011)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/linux?flag=4](https://vscode-helper171-ruby.vercel[.]app/settings/linux?flag=4)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/windows?flag=3](https://vscode-helper171-ruby.vercel[.]app/settings/windows?flag=3)
- [https://vscode-settingstask.vercel\[.\]app/api/settings/linux](https://vscode-settingstask.vercel[.]app/api/settings/linux)
- [https://vscode-helper171.vercel\[.\]app/settings/windows?flag=4](https://vscode-helper171.vercel[.]app/settings/windows?flag=4)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/linux?flag=3](https://vscode-helper171-ruby.vercel[.]app/settings/linux?flag=3)
- [https://codeviewer-three.vercel\[.\]app/task/windows?token=f93a80304111](https://codeviewer-three.vercel[.]app/task/windows?token=f93a80304111)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/mac?flag=301](https://vscode-settings-bootstrap.vercel[.]app/settings/mac?flag=301)
- [https://codeviewer-three.vercel\[.\]app/task/mac?token=2a643f1b401f](https://codeviewer-three.vercel[.]app/task/mac?token=2a643f1b401f)
- [https://vscode-load.onrender\[.\]com/settings/mac?flag=5](https://vscode-load.onrender[.]com/settings/mac?flag=5)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/windows?flag=308](https://vscode-settings-bootstrap.vercel[.]app/settings/windows?flag=308)
- [https://codeviewer-three.vercel\[.\]app/task/linux?token=f93a80304111](https://codeviewer-three.vercel[.]app/task/linux?token=f93a80304111)
- [https://vscode-load-config.vercel\[.\]app/settings/windows?flag=4](https://vscode-load-config.vercel[.]app/settings/windows?flag=4)
- [https://vscode-helper-132.vercel\[.\]app/settings/mac?flag=4](https://vscode-helper-132.vercel[.]app/settings/mac?flag=4)
- [https://codeviewer-three.vercel\[.\]app/task/mac?token=6df937fe9011](https://codeviewer-three.vercel[.]app/task/mac?token=6df937fe9011)
- [https://codeviewer-three.vercel\[.\]app/task/linux?token=6df937fe9011](https://codeviewer-three.vercel[.]app/task/linux?token=6df937fe9011)
- [https://vscode-settingstask.vercel\[.\]app/api/settings/bootstraplinux](https://vscode-settingstask.vercel[.]app/api/settings/bootstraplinux)
- [https://vscode-settings-config.vercel\[.\]app/settings/mac?flag=9](https://vscode-settings-config.vercel[.]app/settings/mac?flag=9)

- [https://vscode-settings-config.vercel\[.\]app/settings/linux?flag=9](https://vscode-settings-config.vercel[.]app/settings/linux?flag=9)
- [https://vscode-helper171.vercel\[.\]app/settings/mac?flag=4](https://vscode-helper171.vercel[.]app/settings/mac?flag=4)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/mac?flag=6](https://vscode-helper171-ruby.vercel[.]app/settings/mac?flag=6)
- [https://vscode-settings03kui.vercel\[.\]app/api/settings/windows](https://vscode-settings03kui.vercel[.]app/api/settings/windows)
- [https://vscode-load-config.vercel\[.\]app/settings/mac?flag=1](https://vscode-load-config.vercel[.]app/settings/mac?flag=1)
- [https://codeviewer-three.vercel\[.\]app/task/windows?token=2a643f1b401f](https://codeviewer-three.vercel[.]app/task/windows?token=2a643f1b401f)
- [https://codeviewer-three.vercel\[.\]app/task/mac?token=f93a80304111](https://codeviewer-three.vercel[.]app/task/mac?token=f93a80304111)
- [https://vscode-settings-config.vercel\[.\]app/settings/windows?flag=9](https://vscode-settings-config.vercel[.]app/settings/windows?flag=9)
- [https://www.vscodeconfig\[.\]com/settings/windows?flag=1](https://www.vscodeconfig[.]com/settings/windows?flag=1)
- [https://vscode-settings-config.vercel\[.\]app/settings/mac?flag=606](https://vscode-settings-config.vercel[.]app/settings/mac?flag=606)
- [https://vscode-toolkit-bootstrap.vercel\[.\]app/settings/linux?flag=306](https://vscode-toolkit-bootstrap.vercel[.]app/settings/linux?flag=306)
- [https://vscode-settings03kui.vercel\[.\]app/api/settings/linux](https://vscode-settings03kui.vercel[.]app/api/settings/linux)
- [https://www.vscodeconfig\[.\]com/settings/linux?flag=4](https://www.vscodeconfig[.]com/settings/linux?flag=4)
- [https://vscode-settings-config.vercel\[.\]app/settings/windows?flag=606](https://vscode-settings-config.vercel[.]app/settings/windows?flag=606)
- [https://vscode-settings-bootstrap.vercel\[.\]app/settings/linux?flag=305](https://vscode-settings-bootstrap.vercel[.]app/settings/linux?flag=305)
- [https://www.vscodeconfig\[.\]com/settings/linux?flag=3](https://www.vscodeconfig[.]com/settings/linux?flag=3)
- [https://vscode-helper171-ruby.vercel\[.\]app/settings/mac?flag=4](https://vscode-helper171-ruby.vercel[.]app/settings/mac?flag=4)
- [https://vscode-settingstask.vercel\[.\]app/api/settings/mac](https://vscode-settingstask.vercel[.]app/api/settings/mac)
- [https://codeviewer-three.vercel\[.\]app/task/linux?token=2a643f1b401f](https://codeviewer-three.vercel[.]app/task/linux?token=2a643f1b401f)
- [https://vscode-settings-config.vercel\[.\]app/settings/linux?flag=8](https://vscode-settings-config.vercel[.]app/settings/linux?flag=8)

Github repository

- [https://github\[.\]com/veneliteus-dev/casino-game/](https://github[.]com/veneliteus-dev/casino-game/)
- [https://github\[.\]com/brahmabit/be_challenge_blockchain/](https://github[.]com/brahmabit/be_challenge_blockchain/)
- [https://github\[.\]com/vnvstore/funtico-labs-assessment-15/](https://github[.]com/vnvstore/funtico-labs-assessment-15/)
- [https://github\[.\]com/samuelmeadowbiankah/felina/](https://github[.]com/samuelmeadowbiankah/felina/)

- <https://github.com/nhonlvsoict/skill-test-main/>
- <https://github.com/goldendragon68/Bullana/>
- <https://github.com/ivanwassaf/skill-test/>
- https://github.com/SettleMint-Tech-Hub5/SettleMint_Platform/
- <https://github.com/veneliteus-dev/exchange-backend/>
- <https://github.com/0x9x-sketch/Oasis361/>
- <https://github.com/eastmade/web3project-momo-token/>

부록 C. 탐색 조건

탐색하지 않는 폴더

목록	목록
node_modules	.svg
static	.idea
license	.lock
site-packages	.bin
robots	.vscode
vendor	.p2
Pods	__MACOSX
.git	.angular
.github	.yarn
.node-gyp	android
.nvm .local	cocoapods
.cache	homebrew
.pyp	xcuserdata
.pyi	release
.pyenv	debug

.qt	x86
.dex	Python
__pycache__	.svn
yarn	.android
.gradle	cache
.cursor	Qt
.py	X

탐색하지 않는 폴더 목록

업로드하지 않는 확장자

목록	목록
.exe	.mpeg
.dll	.m4p
.msi	.m4a
.dmg	.m4v
.vmdk	.aac
.iso	.flac
.pkg	.aiff
.apk	.qt
.xapk	.flv
.aar	.swf
.ap_	.pyc
.aab	.lock
.dex	.psd
.class	.pack
.rpm	.old
.deb	.ppt

.ipa	.pptx
.dsym	.virtualization
.mp4	.indd
.avi	.eps
.mp3	.ai
.wmv	.a
.wma	.jar
.mov	.so
.webm	.o
.avchd	.wt
.mkv	.lib
.ogg	.dylib
.mpe	.bin
.mpv	.ffx
.svg	.gem
.css	.html
.scss	X

업로드하지 않는 확장자 목록

autoUploadScript 탐색 문자열

목록	목록
.keystore	phone
database	bank
financ	.env
env	environment
config	configuration
configure	.conf

.cfg	.ini
.properties	.yaml
.yml	.toml
metamask	phantom
bitcoin	ethereum
eth	trust
wallet	coinbase
exodus	ledger
trezor	keystore
keyring	keychain
atomic	electrum
mycelium	blockchain
bravewallet	rabby
coin98	backpack
core	mathwallet
solflare	glow
keplr	argent
martian	petra
binance	okx
crypto	cryptocurrency
hardhat	truffle
private	privatekey
private_key	private-key
privkey	priv_key
key	keypair
key_pair	key-pair

.pem	.p12
.pfx	.jks
keystore	.keys
keys	.p8
.p7b	.p7c
.cer	.crt
.cert	cert
.der	id_rsa
id_dsa	id_ecdsa
id_ed25519	.pub
.priv	seed
seedphrase	seed_phrase
seed-phrase	mnemonic
phrase	passphrase
pass_phrase	pass-phrase
recovery	recoveryphrase
recovery_phrase	recovery-phrase
backup	backupphrase
backup_phrase	backup-phrase
12words	12_words
12-words	24words
24_words	24-words
bip39	bip44
password	passwd
pass	pwd
credential	credentials

auth	authentication
token	access_token
refresh_token	api_key
apikey	api-key
apisecret	api_secret
api-secret	secret
secrets	secretkey
secret_key	secret-key
masterkey	master_key
master-key	masterpassword
master_password	master-password
account	accounts
profile	profiles
user	username
user_name	user ame
login	signin
sign_in	sign-in
address	addresses
tx	transaction
transactions	.db
.sqlite	.sqlite3
.sql	.mdb
.accdb	.dbf
.doc	.docx
.pdf	.md
.markdown	.rtf

.odt	.xls
.xlsx	.txt
text	note
notes	memo
memos	screenshot
screen	snapshot
capture	.png
.jpg	.jpeg
.bmp	.json
.js	.ts
.jsx	.tsx
.csv	.xml
.lock	.log
.bak	backup
.old	.orig
.save	.swp
.tmp	tmp
my	personal
vault	safe
secure	lock
encrypt	decrypt
signature	sign
certificate	cert
identity	session
cookie	X

autoUploadScript 탐색 문자열 목록



엔키화이트햇

ENKI Whitehat

오펜시브 시큐리티 전문 기업, 공격자 관점으로 깊이가 다른 보안을 제시합니다.