

TLP: GREEN

# Operation Dream Magic

부제: Lazarus 조직의 매직라인 취약점 악용에 대한 7개월의 추적과 분석

---

안랩 대응팀

2023. 10. 13

## 문서 등급에 대한 안내

발간물이나 제공되는 콘텐츠는 아래와 같이 문서 등급 별 허가된 범위 내에서만 사용이 가능합니다.

문서 등급	배포 대상	주의 사항
TLP: RED	특정 고객(사)에 한정하여 제공되는 보고서	보고서 수신자 혹은 수신 부서 만 접근이 허가된 문서. 수신자 외 복제 및 배포 불가
TLP: AMBER	제한된 고객(사)에 한정하여 제공되는 보고서	보고서 수신 조직(회사) 내부에서는 복제 및 배포 가능. 다만, 조직 외 교육 목적 등을 위해 사용될 경우에는 안랩의 허락 필수
TLP: GREEN	해당 서비스 내 누구나 이용 가능 보고서	해당 업종 등에서는 자유로운 사용이 가능하며 출처만 밝히면 내부 교육, 동종 업계, 보안 담당자 교육 자료로 활용 가능 다만, 일반인 대상 발표자료에는 엄격히 제한
TLP: WHITE	자유 이용 가능 보고서	출처표시 상업적, 비상업적 이용 가능 변형 등 2 차적 저작물 작성 가능

### [중요] 참고사항

본 보고서에는 현재까지 확인한 내용을 기반으로 분석가 의견이 다수 포함되어 있습니다. 분석가들마다 의견이 다를 수 있으며 새로운 근거가 확인되면, 본 보고서 내용도 사전 고지 없이 변경될 수 있습니다.

보고서에 통계와 지표가 포함되어 있는 경우 일부 데이터는 반올림되어 세부 항목의 합과 전체 합계가 일치하지 않을 수도 있습니다.

본 보고서는 저작권법에 의해 보호를 받는 저작물로서 어떤 경우에도 무단전재와 무단복제를 금지하며, 보고서 내용의 전부 또는 일부를 이용하고자 하는 경우에는 안랩의 사전 동의 받아야 합니다.

만약 안랩의 동의 없이 전재 또는 복제를 하는 경우 저작권 관계법령에 의하여 민사 또는 형사 책임을 지게 되므로 주의가 필요합니다.

## 목 차

1. 프롤로그 .....	4
2. 매직라인 취약점 탐지 현황 .....	4
(1) 분야별 탐지 현황 .....	5
(2) 버전별 탐지 현황 .....	15
3. 안랩의 대응현황 .....	16
(1) 악성코드 진단현황 .....	16
(2) C2 정보 .....	18
(3) 권장사항 .....	19
4. 매직라인 점검 및 조치 .....	21
5. Case Study .....	23
(1) 매직라인 취약점 악용 과정 .....	23
1) Case Study 1: 2023 년 02 월 제조 A 기업 PC 3 .....	24
2) Case Study 2: 2023 년 04 월 언론 B 사 PC 1 .....	25
3) Case Study 3: 2023 년 07 월 금융 A 기업 PC 1 .....	26
4) Case Study 에서 공통점 .....	26
(2) 악성코드 분석 .....	27
1) DLL-Side Loading .....	27
2) 악성코드와 인자값 .....	29
3) 파일리스(Fileless) 악성코드 .....	30
(3) Lazarus 조직으로 판단하는 근거 .....	34
1) 악성코드 재구성 코드 동일 .....	35
2) *.SYS 파일 생성 과정 동일 .....	36
6. 에필로그 .....	36

## 1. 프롤로그

Lazarus 조직은 국가가 배후인 것으로 알려진 해킹 조직으로 금전적인 이득, 자료 탈취 등의 목적으로 전 세계를 대상으로 꾸준히 해킹하고 있습니다.

Lazarus 조직의 이니세이프 취약점을 악용한 워터링 홀을 간단하게 정리하면 언론사의 특정 기사에 악성 링크 삽입, 해당 기사를 클릭하는 기업, 기관이 해킹 대상, 국내 취약한 홈페이지를 C2 로 악용 그리고 제한된 범위의 해킹을 위해서 IP 필터링 등을 사용했습니다. 이번 워터링 홀에서 악용하는 프로그램의 취약점이 매직라인으로 변경됐을 뿐 워터링 홀 과정은 과거 이니세이프 사례와 동일합니다.

안랩은 Lazarus 조직의 매직라인 취약점을 악용한 워터링 홀을 대응하기 위해서 여러 팀의 협업이 있었습니다. 매직라인 취약점을 탐지하기 위한 조건을 연구하고 백신에 업데이트해준 분석팀, 탐지된 PC 가 고객일 경우 로그 및 샘플 수집 등 고객 대응을 해준 기술지원팀, 수집한 로그 분석 및 국가기관과의 소통을 담당한 대응팀 등 여러 팀의 협업이 있었습니다.

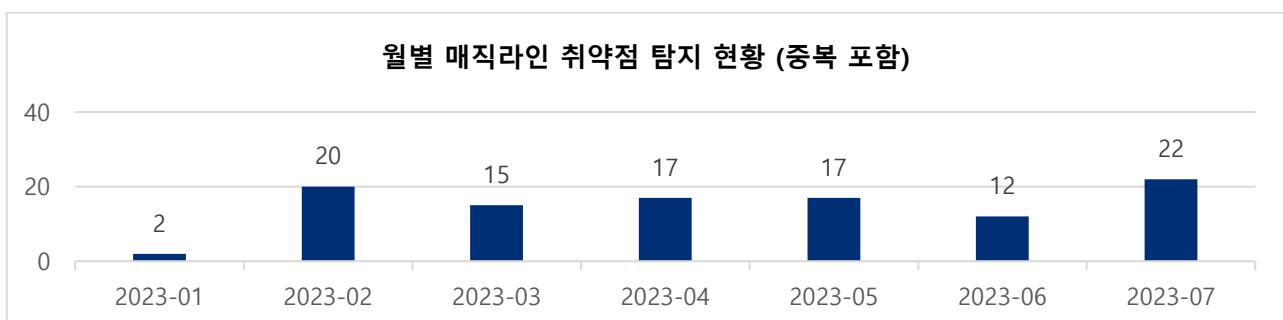
또한 안랩은 국가기관과도 정보 공유 및 협업을 통해서 Lazarus 조직의 매직라인 취약점을 악용한 워터링 홀을 추적하고 분석을 진행했으며, 매직라인 제조 기업의 이름 일부와 매직라인의 이름 일부를 조합하여 이번 작전을 **“Operation Dream Magic”**으로 명명했습니다.

본 보고서에서는 악성코드 분석 내용, 탐지 현황, 일부 기업의 협조로 수집한 로그 분석, 국가 기관과의 정보 공유 및 협업 등을 바탕으로 해석한 내용을 포함했으며, 이번 작전을 Lazarus 조직의 소행으로 판단한 근거에 대해서도 설명했습니다.

**본 보고서는 안랩 대응팀, 분석팀, 기술지원팀, 일부 탐지 기업 그리고 국가기관의 협업으로 탄생한 결과물입니다.**

## 2. 매직라인 취약점 탐지 현황

안랩은 Lazarus 조직의 매직라인 취약점에 의한 악성 행위를 탐지하고 대응할 수 있는 조건을 제품에 반영한 후 모니터링한 결과 2023.01 월부터 2023.07 월까지 7 개월 동안 총 40 곳의 기업, 기관에서 105 건을 탐지했으며, 탐지한 105 건도 추가 분석을 통해서 모두 정탐임을 검증했습니다. (아래 [통계 1] 참고)



[통계 1] 월별 매직라인 취약점 탐지 현황

안랩이 탐지한 105 건은 단순히 수치상으로만 보면 매우 작은 수치이므로 심각하지 않다고 판단할 수 있지만 해킹 주체가 Lazarus 조직이며, 해킹 대상은 개인이 아닌 기업이나 기관이라는 점은 매우 중요한 대목입니다. 그 이유는 만약 Lazarus 조직의 해킹으로 중요한 자료가 유출됐다면 해당 해킹 대상뿐만 아니라 국가적으로도 큰 손실이나 안보 위협이 발생할 수 있기 때문입니다.

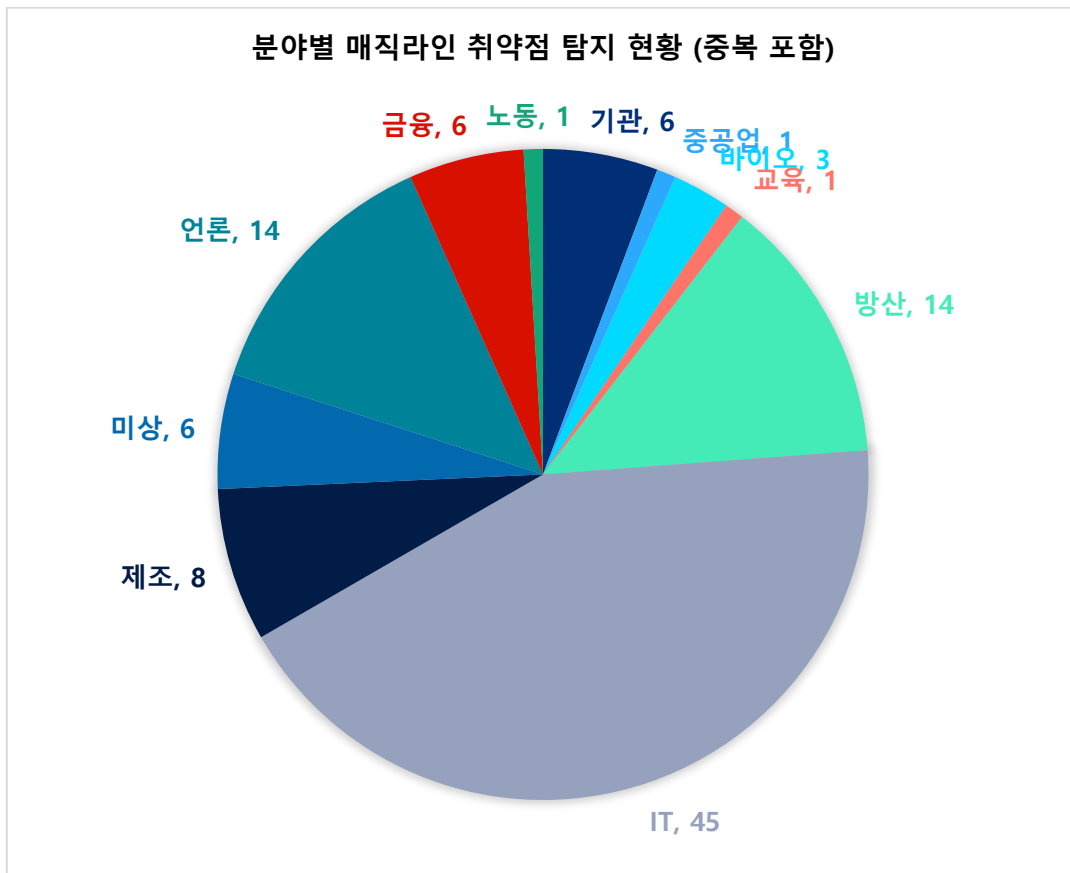
안랩이 탐지한 105 건은 아래 2 가지 항목으로 설명할 수 있습니다.

- 분야별 탐지 현황
- 버전별 탐지 현황

**[참고]** 해킹 대상을 특정할 수 있는 구체적인 정보는 여러가지 상황을 고려하여 언급하지 않았지만 언론을 통해서 이미 기사화된 경우 해킹 대상을 대략적으로 언급했습니다.

### (1) 분야별 탐지 현황

105건의 탐지를 분야별로 살펴보면 IT가 45건으로 가장 많으며, 방산 14건, 언론 14건 순입니다. 그리고 주요 분야를 중심으로 중요하다고 판단되는 사례만 설명했습니다. (아래 [통계 2] 참고)



[통계 2] 분야별 매직라인 취약점 탐지 현황

## 1) IT 분야

45건으로 가장 많은 탐지 건수를 차지하고 있는 IT 분야에는 IT 솔루션 제조 기업, 계열사 IT 인프라 관리 기업 등이 포함되어 있습니다.

**IT 솔루션 제조 기업**은 과거 해킹 조직의 해킹 사례에 비취보면 초기 침투나 내부 전파가 목적일 경우 해킹 대상이 사용하는 IT 솔루션을 악용(ex, 악성코드 유포 및 감염)했습니다. 그 이유는 해킹 대상을 직접 해킹하는 것 보다 IT 솔루션 제조 기업을 해킹하여 습득한 정보(IT 솔루션의 취약점, 운영 방식, 소스코드 등)로 해킹 대상을 해킹하면 효과적이고, 성공률을 높일 수 있기 때문입니다.

아래 기사는 과거 해킹 조직이 중앙 관리 솔루션의 취약점을 악용하여 국내 기업을 해킹한 사례입니다.

### [보안뉴스] 2018.10.24일자 기사, 대규모 사이버 해킹 루트, 중앙관리 솔루션의 취약점

➤ <https://www.boannews.com/media/view.asp?id=73952>

아래 [표 1]은 IT 솔루션 IT A 기업의 매직라인 취약점 탐지 로그로 2023-05-23일 최초 탐지를 시작으로 07월까지 총 5대의 PC에서 8건이 탐지됐으며, 악성코드 감염과 그로 인해 일부 PC는 C2와 통신했음을 확인했습니다.

탐지 시간	탐지 PC	탐지 IP	해킹 대상	탐지 분야	매직라인 버전
2023-07-25 11:43:54	PC 5	***.***.***.178 (KR)	IT A	IT	1.0.0.14
2023-07-21 09:46:45	PC 4	***.***.***.178 (KR)	IT A	IT	1.0.0.14
2023-07-20 12:43:58	PC 4	***.***.***.178 (KR)	IT A	IT	1.0.0.14
2023-07-20 10:01:15	PC 3	***.***.***.178 (KR)	IT A	IT	1.0.0.20
2023-07-19 13:16:17	PC 3	***.***.***.178 (KR)	IT A	IT	1.0.0.20
2023-07-10 22:41:54	PC 2	***.***.***.178 (KR)	IT A	IT	1.0.0.9
2023-07-07 13:39:14	PC 2	***.***.***.178 (KR)	IT A	IT	1.0.0.9
2023-05-23 16:33:48	PC 1	***.***.***.178 (KR)	IT A	IT	1.0.0.20

[표 1] IT 솔루션 제조 A 기업의 매직라인 취약점 탐지 로그

아래 [표 2]에서 금융 보안 솔루션 제조 기업 IT B, C는 각각 2016년 Andariel 조직의 해킹으로 내부 정보 유출, 2020년 Lazarus 조직의 공급망 공격에 악용 피해가 발생했던 기업입니다. 추가로 Lazarus 조직은 2021년부터

IT B의 이니셔티브 취약점을 해킹에 악용했습니다. 그리고 IT D는 DLP 솔루션, IT E는 DRM 솔루션 제조 기업으로 두 기업의 홈페이지에 공개된 고객 레퍼런스에는 주요 기업, 기관이 포함되어 있습니다.

탐지 시간	탐지 PC	탐지 IP	해킹 대상	탐지 분야	매직라인 버전
2023-06-29 15:25:17	PC 1	***.***.***.9(KR)	IT B	IT	1.0.0.14
2023-06-27 13:23:32	PC 1	***.***.***.78(KR)	IT C	IT	확인 불가
2023-06-13 10:04:30	PC 1	***.***.***.78(KR)	IT C	IT	확인 불가
2023-03-20 09:22:31	PC 1	***.***.***.78(KR)	IT C	IT	확인 불가
2023-06-26 11:39:03	PC 2	***.***.***.140(KR)	IT D	IT	1.0.0.9
2023-06-19 17:08:00	PC 1	***.***.***.140(KR)	IT D	IT	1.0.0.14
2023-05-31 10:25:05	PC 1	***.***.***.140(KR)	IT D	IT	1.0.0.14
2023-07-13 10:42:50	PC 1	***.***.***.129(KR)	IT E	IT	1.0.0.20

[표 2] 금융, 데이터 보안 솔루션 제조 기업의 매직라인 취약점 탐지 로그

위 [표 1, 2]를 종합하면 Lazarus 조직이 IT 솔루션 제조 기업을 해킹하는 것은 해당 기업에 대한 정보(IT 솔루션의 취약점, 운영 방식, 소스코드 등) 수집과 함께 수집한 정보를 고객 해킹에 악용할 목적이라고 판단했습니다.

이번 사례와 관련없지만 Lazarus 조직은 2022년 5월 국내 기업을 해킹할 목적으로 해당 기업에서 사용하는 특정 솔루션의 운영 방식을 정확하게 파악하여 악성코드 유포에 사용할 수 있는 또 다른 악성코드를 제작했으며, 아래 [그림 1]은 해당 악성코드의 내부 코드입니다.

특정 솔루션 구축 문제 등 제약 사항이 있어서 아래 악성코드의 정확한 동작 여부를 확인하지 못했지만 위에서 설명한 IT 솔루션 제조 기업과 고객에 대한 잠재적인 보안 위협을 증명하는 예시론 충분하다고 판단합니다.

```
DataRow dataRow = (DataRow)obj;
SqlParameterCollection spParamCollection = new SqlParameterCollection("SP_GL_EQUIPINFO_S");
spParamCollection.Add("PORG", DbType.String, "D");
spParamCollection.Add("PVAL", DbType.String, "ROOT");
spParamCollection.Add("PADMIN", DbType.Int32, dataRow["ADMIN_ID"]);
DataTable dataTable = CommonInfo.rmtch.GetRmBasic().SpExecuteWrappedDataSet(spParamCollection).DataSet.Tables[0];
int admin_ID = int.Parse(dataRow["ADMIN_ID"].ToString());
foreach (object obj2 in dataTable.Rows)
```

[그림 1] 특정 솔루션을 악용하는 악성코드의 코드

매직라인 취약점이 탐지된 IT 솔루션 제조 기업 5곳은 자사 홈페이지에 고객 레퍼런스를 공개하고 있습니다.

기업은 신규 고객 확보를 위해서 고객 레퍼런스를 공개할 수 있지만 보안 측면에선 IT 솔루션 제조 기업과 고객 모두에게 잠재적으로 보안 위협이 될 수 있으며, 해킹 조직은 고객 레퍼런스를 고객 현황 파악, 해킹 대상 선정, 해킹 계획 수립 및 실행에 매우 유용한 정보로 사용할 수 있습니다. 따라서 IT 솔루션 제조 기업도 어려운 숙제이지만 고객 레퍼런스를 최소한으로 노출시킬 수 있는 정보 공개 방법에 대해서 고민이 필요합니다.

본 보고서에서는 위 [표 1, 2]의 IT 제조 기업 5곳의 고객 레퍼런스는 인용하지 않았습니다.

**계열사 IT 인프라 관리 기업**도 위에서 설명한 IT 솔루션 제조 기업과 비슷한 맥락으로 해킹 대상을 직접 해킹하기보다 해킹 대상에게 IT 인프라 구축, 운영 등 전반적인 IT 서비스를 제공하며, 해킹 대상의 IT 인프라에 대해 잘 알고 있는 계열사 IT 인프라 관리 기업을 해킹하는 것이 효과적이며, 성공률을 높일 수 있습니다. 경우에 따라서 해킹한 계열사 IT 인프라 관리 기업을 경유하여 모든 계열사의 IT 인프라에 접근할 수 있는 상황이 발생할 수 있습니다.

아래 기사는 위 설명을 뒷받침하는 유사한 사례의 예시입니다.

**[연합뉴스] 2016-06-13일자 기사: 북한, 방산업체 등 대기업 계열사 해킹...군사기밀 노린 듯**

➤ <https://yonhapnewstv.co.kr/news/MYH20160613008100038>

아래 [표 3]은 계열사 IT 인프라 관리 기업의 매직라인 취약점 탐지 로그입니다. 두 기업의 공통점은 계열사에 IT 서비스 제공 경험이 있으며, 계열사 중에 방산 기업이 있거나 자체적으로 방산 사업을 진행하는 등 방산 분야 와도 매우 밀접하게 관련이 있습니다. Lazarus 조직이 매직라인 취약점을 악용하여 두 기업을 해킹하는 것도 이와 무관하지 않으며, 두 기업에 대한 정보 수집과 함께 수집한 정보를 계열사 해킹에 사용할 목적이 라고 판단했습니다.

탐지 시간	탐지 PC	탐지 IP	해킹 대상	탐지 분야	매직라인 버전
2023-04-18 10:40:56	PC 1	***.***.***.4(KR)	IT F	IT	확인 불가
2023-05-16 10:52:51	PC 1	***.***.***.35(KR)	IT F	IT	확인 불가
2023-05-17 15:34:54	PC 2	***.***.***.8(KR)	IT F	IT	확인 불가
2023-05-29 16:01:46	PC 3	***.***.***.8(KR)	IT F	IT	1.0.0.20
2023-06-12 10:31:56	PC 1	***.***.***.8(KR)	IT F	IT	확인 불가
2023-06-12 11:26:02	PC 4	***.***.***.8(KR)	IT F	IT	1.0.0.9
2023-03-20 10:33:04	PC 1	***.***.***.23(KR)	IT G	IT	확인 불가



2023-04-26 09:42:29	PC 1	***.***.***.23(KR)	IT G	IT	확인 불가
2023-04-26 16:10:10	PC 2	***.***.***.165(KR)	IT G	IT	1.0.0.20
2023-05-03 09:59:38	PC 3	***.***.***.22(KR)	IT G	IT	확인 불가
2023-05-05 10:46:34	PC 4	***.***.***.28(KR)	IT G	IT	확인 불가

[표 3] IT 인프라 관리 기업의 매직라인 취약점 탐지 로그

## 2) 방산 분야

방산은 전 세계적으로 국가의 핵심 분야 중 하나로 해킹 조직에 의해서 꾸준히 해킹이 발생하고 있으며, 그 중 일부는 해킹 성공으로 중요한 자료가 유출되는 피해가 발생했습니다. 그리고 해킹 사례 중 일부가 언론을 통해 보도되면서 이슈가 됐던 적도 있습니다. 이처럼 해킹 조직이 방산 기업을 해킹하는 이유는 탈취한 자료 (ex, 무기 설계 자료)를 자국의 방산 기술 개발에 사용할 목적이 가장 크기 때문으로 피해 방산 기업과 국가에게 해킹이라는 사이버 안보 위협이 물리적인 안보 위협으로까지 발전할 수 있음을 증명해주고 있습니다.

[보안뉴스] 2021년 10월 06일자 기사: 방산업체 5년간 해킹 14건... 주요 기밀 빠져나갔다,

➤ <https://www.boannews.com/media/view.asp?idx=101298>

방산에서 매직라인 취약점은 통신 분야 4개 기업, 배터리 분야 1개 기업 등 총 5개 기업의 14대 PC에서 탐지 됐으며, 5개 기업에서 감염이 확인된 Lazarus 조직의 악성코드는 해당 기업의 민감한 자료 탈취에 목적이 있으므로 만약이라는 가정이지만 5개의 기업에서 취급하는 중요한 자료가 해킹으로 유출됐다면 피해 기업뿐만 아니라 국가에게 안보 위협이 될 수 있습니다.

탐지 시간	탐지 PC	탐지 IP	해킹 대상	탐지 분야	매직라인 버전
2023-03-20 15:55:17	PC 1	***.***.***.211(KR)	방산 A	방산	1.0.0.20
2023-03-24 15:48:55	PC 2	***.***.***.211(KR)	방산 A	방산	확인 불가
2023-03-29 10:54:45	PC 3	***.***.***.211(KR)	방산 A	방산	1.0.0.14
2023-03-30 15:26:06	PC 4	***.***.***.211(KR)	방산 A	방산	확인 불가
2023-04-14 15:47:02	PC 5	***.***.***.211(KR)	방산 A	방산	1.0.0.20
2023-02-13 16:36:23	PC 1	***.***.***.35(KR)	방산 B	방산	1.0.0.14
2023-02-15 11:15:33	PC 2	***.***.***.35(KR)	방산 B	방산	1.0.0.20

2023-02-16 16:50:52	PC 3	***.***.***.35(KR)	방산 B	방산	1.0.0.9
2023-03-13 12:41:46	PC 1	***.***.***.43(KR)	방산 C	방산	확인 불가
2023-03-21 16:41:26	PC 2	***.***.***.43(KR)	방산 C	방산	1.0.0.14
2023-03-22 16:16:29	PC 3	***.***.***.44(KR)	방산 C	방산	확인 불가
2023-03-28 11:21:11	PC 4	***.***.***.44(KR)	방산 C	방산	1.0.0.14
2023-04-07 16:11:55	PC 5	***.***.***.43(KR)	방산 C	방산	1.0.0.20
2023-03-14 16:19:19	PC 1	***.***.***.194(KR)	방산 D	방산	1.0.0.20

[표 4] 방산 기업의 매직라인 취약점 탐지 로그

해킹으로 탈취한 코인을 미사일 개발 자금으로 사용하는 것뿐만 아니라 미사일 개발에 필요한 동향 수집 및 자국의 방산 기술 발전을 위해서 타국의 방산 기술 탈취를 위해서 방산 기업을 지속적으로 해킹할 수 있습니다.

아래 기사는 위 설명을 뒷받침하는 유사한 사례의 예시입니다.

[VOA] 2023.5.11일자 기사: 백악관 NSC 부보좌관 "북한, 사이버 활동으로 미사일 자금 절반 총당"

➤ <https://www.voakorea.com/a/7087509.html>

### 3) 언론 분야

Lazarus 조직은 이니세이프 취약점 때와 마찬가지로 매직라인 취약점도 국내 일부 언론사 홈페이지를 워터링 홀에 악용했습니다. 이번 이슈를 추적하고 분석하기 위해서 언론사와의 협업이 매우 중요했지만 언론이라는 특수성 때문에 자칫 불필요한 오해를 낳을 수 있기 때문에 신중하게 접근해야 하는 분야이기도 합니다. 아래 [표 5]에서 일부 언론사는 협업을 위해서 연락을 취했으나 해당 언론사로부터 응답은 없었습니다.

탐지 시간	탐지 PC	탐지 IP	해킹 대상	탐지 분야	매직라인 버전
2023-04-06 11:39:05	PC 1	***.***.***.219(KR)	언론 A	언론	1.0.0.20
2023-05-02 09:23:17	PC 1	***.***.***.219(KR)	언론 A	언론	1.0.0.20
2023-04-07 17:06:00	PC 1	***.***.***.40(KR)	언론 B	언론	1.0.0.20
2023-05-28 11:41:17	PC 2	***.***.***.51(KR)	언론 B	언론	1.0.0.20

2023-05-09 11:34:20	PC 1	***.***.***.49(KR)	언론 C	언론	1.0.0.14
2023-05-23 10:27:30	PC 1	***.***.***.49(KR)	언론 C	언론	1.0.0.14
2023-05-30 16:46:39	PC 1	***.***.***.81(KR)	언론 D	언론	1.0.0.20
2023-05-30 18:46:10	PC 2	***.***.***.48(KR)	언론 D	언론	1.0.0.20
2023-05-31 11:39:00	PC 3	***.***.***.32(KR)	언론 D	언론	1.0.0.20
2023-07-11 16:27:49	PC 2	***.***.***.48(KR)	언론 D	언론	1.0.0.20
2023-07-21 08:57:25	PC 4	***.***.***.81(KR)	언론 D	언론	확인 불가
2023-06-23 11:16:19	PC 1	***.***.***.2(KR)	언론 E	언론	1.0.0.9
2023-07-05 11:26:47	PC 1	***.***.***.2(KR)	언론 E	언론	1.0.0.9
2023-07-05 15:06:07	PC 1	***.***.***.71(KR)	언론 F	언론	1.0.0.20

[표 5] 언론사의 매직라인 취약점 탐지 로그

위 [표 5]의 각 언론사 PC에서 매직라인 취약점이 탐지된 것은 직접 해킹 대상은 아니며, 자사 홈페이지에 접속했다가 우연히 매직라인 취약점 동작 조건(ex, IP 대역)이 일치하여 탐지된 것으로 의심하고 있습니다. 워터링 홀의 특성상 웹 사이트를 통해 진행되며, 불특정 다수가 접속하므로 해킹 대상의 범위를 명확, 제한하기 위해서 조건(ex, IP 대역)을 설정해도 조건이 일치하여 감염될 가능성은 충분히 있습니다.

안랩은 언론사 B의 협조로 매직라인 취약점이 탐지된 PC에서 로그를 수집하여 분석했지만 타임라인에서 매직라인 취약점 탐지 시점의 흔적은 없었으며, 아래 [그림 2]는 비록 매직라인 취약점 탐지 시점 이후 언론사 홈페이지에 접속한 흔적이지만 매직라인 취약점 탐지 시점에 해당 PC가 위 언론사 B에 접속했을 것으로 의심하는 근거입니다.

날짜	종류	사용자	내용	Extra1	Extra2
2023-04-07 오후 5:32:03	CrmH		https://stats	dn.com/cmsapi/execPurge_open.php?auth=383F36823F3E3...	https://stats.kinxcdn.com/cmsapi/exe
2023-04-07 오후 5:31:54	CrmH		https://www	ws.com/	reload   뉴스 -   신문
2023-04-07 오후 5:31:54	CrmH		https://www	ws.com/	reload   뉴스 -   신문
2023-04-07 오후 5:31:54	CrmH		https://www	ws.com/	reload   뉴스 -   신문
2023-04-07 오후 5:31:53	CrmH		https://www	ws.com/	reload   뉴스 -   신문
2023-04-07 오후 5:31:51	CrmH		https://www	ws.com/	reload   뉴스 -   신문
2023-04-07 오후 5:31:48	CrmH		https://nice.i	rs.com/main/admin/#/manage/makeliveedit/list/youtubeLive	link, redirect : Login
2023-04-07 오후 5:31:40	CrmH		https://nice.i	rs.com/main/admin/#/manage/makeliveedit/list/youtubeLive	link : Login
2023-04-07 오후 5:31:34	CrmH		https://nice.i	rs.com/main/admin/#/manage/makemovieedit/list	link 동영상 배치v2 :
2023-04-07 오후 5:31:32	CrmH		https://nice.i	rs.com/main/admin/#/manage/makenewsedit/list	link 콘텐츠 배치v2 :
2023-04-07 오후 5:31:30	CrmH		https://nice.i	rs.com/main/admin/#/member/authority	link 권한 관리
2023-04-07 오후 5:31:29	CrmH		https://nice.i	rs.com/main#/dashboard	link, redirect 대쉬보드 :
2023-04-07 오후 5:31:28	CrmH		https://nice.i	rs.com/main	form submi... 대쉬보드 :
2023-04-07 오후 5:31:28	CrmH		https://nice.i	rs.com/	form submit 대쉬보드 :
2023-04-07 오후 5:31:26	CrmH		https://nice.i	rs.com/login	typed, redir... : Login
2023-04-07 오후 5:31:26	CrmH		https://nice.i	rs.com/	typed, redir... 대쉬보드
2023-04-07 오후 5:31:26	CrmH		https://nice.i	rs.com/	typed 대쉬보드

[그림 2] 언론사 B의 PC 타임라인

위 [표 5]에서 각 언론사의 PC가 자사 또는 타사 홈페이지에 접속했는지 여부는 확인을 못했지만 위 [그림 2] 언론사 B의 PC와 동일한 패턴이라면 각 언론사의 PC에서 매직라인 취약점이 탐지됐을 시점에 각 언론사 홈페이지는 Lazarus 조직의 매직라인 취약점을 악용한 워터링 홀에 악용됐을 가능성이 매우 높다고 판단하고 있습니다.

언론 분야는 매직라인 취약점을 악용한 워터링 홀의 목적이 불명확합니다. 아래 기사는 과거 Lazarus 조직의 중앙일보 해킹 사건에 대한 기사로 신문 발행에 차질을 발생시켜 피해를 주려는 목적이 명확했습니다.

**[미디어오늘] 2013.01.16일자 기사: 경찰 “지난해 중앙일보 해킹, 북한 체신청 IP”**

➤ <http://www.mediatoday.co.kr/news/articleView.html?idxno=107146>

하지만 Lazarus 조직의 이니세이프, 매직라인 취약점 악용은 과거 중앙일보 해킹의 목적과는 다르며, 위에서 설명한 것처럼 언론사 홈페이지를 워터링 홀에 악용한 것으로 판단했습니다. 각 언론사의 기조는 기사를 통해서 파악이 가능하므로 이를 위해서 언론사 PC를 해킹했을 가능성은 낮으며, 의심해 볼 수 있는 한 가지 가능성은 언론사의 기사 송고 플랫폼에 대한 정보 수집입니다. 만약 언론사 PC 해킹으로 언론사의 기사 송고 플랫폼까지 해킹할 수 있다면 워터링 홀 또는 다른 해킹의 수단으로 악용할 수 있습니다.

**4) 기타: 기관, 금융 분야**

아래 [표 6]에서 기관 A는 시 산하 복지 재단, 기관 B는 공공 기관, 금융 A는 디지털 금융 기업으로 기관 B의 PC 1에서 4회 탐지, 금융 A의 PC 1에서 5회 탐지라는 동일 PC에서 반복 탐지 패턴이 존재합니다.

동일 PC에서 반복 탐지, 해킹 대상의 다수 PC에서 탐지, 3회 이상 탐지 등 이 세 가지 패턴을 다른 분야에 적용하면 동일한 패턴을 가진 해킹 대상은 IT A, C, D, F, G, 방산 A, B, C, 언론 D 등으로 워터링 홀의 특성상 보수적으로 2회까지는 탐지될 수 있다고 해도 그 이상은 Lazarus 조직의 집중 해킹 대상으로 판단했습니다.

탐지 시간	탐지 PC	탐지 IP	해킹 대상	탐지 분야	매직라인 버전
2023-01-30 11:50:15	PC 1	***.***.***.158(KR)	기관 A	기관	1.0.0.14
2023-01-31 11:32:26	PC 2	***.***.***.158(KR)	기관 A	기관	1.0.0.14
2023-06-05 10:58:28	PC 1	***.***.***.58(KR)	기관 B	기관	확인 불가
2023-06-07 10:39:15	PC 1	***.***.***.58(KR)	기관 B	기관	확인 불가
2023-06-08 15:15:04	PC 1	***.***.***.58(KR)	기관 B	기관	확인 불가
2023-06-14 11:27:01	PC 1	***.***.***.58(KR)	기관 B	기관	확인 불가

2023-07-10 15:11:23	PC 1	***.***.***.193(KR)	금융 A	금융	1.0.0.14
2023-07-13 09:50:02	PC 1	***.***.***.193(KR)	금융 A	금융	1.0.0.14
2023-07-18 09:22:57	PC 1	***.***.***.193(KR)	금융 A	금융	1.0.0.14
2023-07-19 15:24:03	PC 1	***.***.***.193(KR)	금융 A	금융	1.0.0.14
2023-07-20 11:20:19	PC 1	***.***.***.193(KR)	금융 A	금융	1.0.0.14
2023-07-31 09:38:30	PC 1	***.***.***.193(KR)	금융 A	금융	확인 불가

[표 6] 기타(기관, 금융 분야)의 매직라인 취약점 탐지 로그

Lazarus 조직의 집중 해킹 대상으로 판단한 근거를 설명하기 위해서 위 [표 6]의 금융 A 기업의 PC 1을 예시로 들었으며, 해당 기업의 매직라인 취약점 최초 탐지 시점인 2023-07-10일부터 2023-08-07일까지 다수의 악성 행위 중 주요 악성 행위만 간추려서 아래 [표 7]로 만들었습니다.

해킹 초기에 정찰 과정을 거칩니다. 그 이유는 정찰 과정에서 정보 수집과 수집한 정보를 분석하여 해킹 대상이 중요한 PC인지를 판단할 수 있으므로 아래 [표 7]에서 1 ~ 3번까지를 정찰 과정으로 판단한다면 4 ~ 12번까지 다수의 악성코드 생성과 실행의 반복, 내부 IP 통신 그리고 C2 통신은 Lazarus 조직이 금융 A 기업의 PC 1을 중요한 PC로 판단하기 때문에 발생하는 과정이라고 판단했습니다.

만약 금융 A 기업의 PC 1이 중요한 PC가 아니었다면 정찰 과정을 거친 후 해킹은 종료되므로 4번부터 행위는 발생하지 않았을 것으로 판단했습니다.

No.	시간	프로세스	행위	데이터
1	2023-07-10 15:11:23	%ProgramFiles% (x86)\dreamsecurity \magicline4nx\magic cline4nx.exe	악성코드 생성 및 실행	%ALLUSERSPROFILE%\citrix\wpvsagen t\wsmprovhost.exe (정상, DLL-Side Loading) %ALLUSERSPROFILE%\citrix\wpvsagen t\mi.dll (악성 DLL 생성)
2	2023-07-13 09:50:02	%SystemRoot%\system32\spoolsv.exe	인젝션 수행	
3	2023-07-18 09:22:57	%ALLUSERSPROFILE%\citrix\mgrmlpac.exe	C2 통신	119.205.211.85:443
4	2023-07-19 15:24:03	%SystemRoot%\system32\spoolsv.exe	인젝션 수행	

5	2023-07-20 16:36:53	%SystemRoot%\system32\spoolsv.exe	악성코드 생성	%ALLUSERSPROFILE%\microsoft\microsoft\msimg32.dll (악성 DLL 생성)
6	2023-07-19 16:41:40	%SystemRoot%\system32\cmd.exe	악성코드 실행	%ALLUSERSPROFILE%\whp\sgrmlpac.exe (정상, DLL-Side Loading)
7	2023-07-20 11:20:19	%SystemRoot%\system32\cmd.exe	악성코드 생성	%ALLUSERSPROFILE%\whp\sgrmlpac.exe → (악성 DLL 생성) %SystemRoot%\temp\rgzsr.dll
8	2023-07-20 11:20:19	%ALLUSERSPROFILE%\whp\sgrmlpac.exe	C2 통신	hxxps://hiwoosung.com/ (221.139.104.54)
9	2023-07-21 15:01:04	%SystemRoot%\system32\spoolsv.exe	내부 IP 통신	***.***.***.***:445
10	2023-07-31 09:38:30	%SystemRoot%\system32\spoolsv.exe	C2 통신	hxxps://www.hds-secu.com/ (203.251.81.11)
11	2023-08-02 14:55:12	%SystemRoot%\system32\spoolsv.exe	악성코드 생성 및 실행	%ALLUSERSPROFILE%\microsoft\windows\servicesetting\tieringengineservice.exe (정상, DLL-Side Loading) %ALLUSERSPROFILE%\microsoft\windows\servicesetting\clusapi.dll (악성 DLL)
12	2023-08-07 09:40:19	%ALLUSERSPROFILE%\microsoft\windows\servicesetting\tieringengineservice.exe	C2 통신	203.251.81.11:443

[표 7] 금융 A 기업의 PC 1에서 확인된 주요 악성 행위

Lazarus 조직의 과거 행적을 고려하면 금융 A 기업의 PC 1을 집중 해킹 대상으로 삼은 이유를 이해할 수 있습니다. Lazarus 조직은 무기 개발에 필요한 자금 조달을 위해서 전 세계 암호화폐 기업, 은행 등 돈과 관련된 기업을 해킹하고 있으며, 최근에 에스토니아의 암호화폐 기업에서 477억원 탈취 사례가 있으므로 금융 A사 해킹도 이와 동일한 목적이라고 판단했습니다.

[MBC] 2023.07.30일자 기사: "북한 Lazarus, 암호화폐 기업 해킹해 477억 원 탈취",

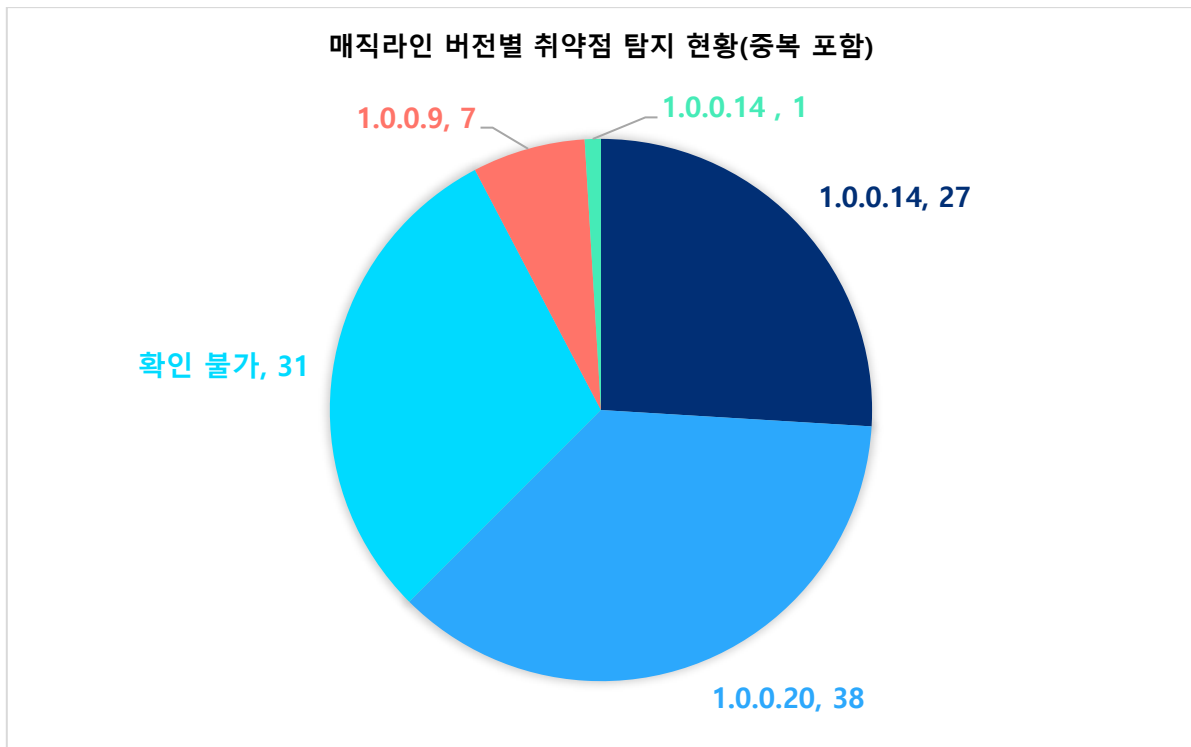
➤ [https://imnews.imbc.com/news/2023/world/article/6509172\\_36133.html](https://imnews.imbc.com/news/2023/world/article/6509172_36133.html)

[경향신문] 2016.03.23일자 기사: 방글라 중앙은행 1000억원대 해킹의 전말은? '

➤ <https://m.khan.co.kr/world/world-general/article/201603230920011#c2b>

## (2) 버전별 탐지 현황

105건의 탐지 중 매직라인 버전이 확인된 탐지 건수 중심으로 살펴보면 1.0.0.20 버전: 38건으로 가장 많고, 그 다음은 1.0.0.14 버전: 27건, 1.0.0.9 버전: 7건 그리고 1.0.0.15버전: 1건 순입니다. 아래 [통계 3]에서 확인 불가가 31건인 이유는 매직라인 취약점뿐만 아니라 추가 악성 행위를 탐지하는 조건을 제품에 반영했기 때문이며, 일부 PC에서 매직라인 취약점 탐지 로그는 없지만 추가 악성 행위만 탐지한 사례가 있었습니다.



[통계 3] 분야별 매직라인 취약점 탐지 현황

위 [통계 3]에서 각 매직라인 버전의 제작 시간 정보를 확인한 결과 아래 [표 8]과 같습니다.

버전	파일 제작 시간
1.0.0.9	Mar 23 2018 17:53:51 GMT+09
1.0.0.14	Feb 12 2019 17:15:41 GMT+09
1.0.0.15	Feb 19 2020 11:05:12 GMT+09
1.0.0.20	Mar 29 2021 19:48:46 GMT+09

[표 8] 매직라인 버전별 제작 시간 정보

위 [표 8]를 보면 각 버전의 파일 제작 시간을 기준으로 최소 2년에서 최대 5년 전에 제작된 것으로 매직라인이 설치되어 있습니다.

지금도 취약한 버전이 설치되어 있는 것은 첫째, 매크라인의 설치 방식이 수동이기 때문에 사용자가 인지하여 최신 버전의 매크라인을 설치하지 않으면 계속 취약한 버전의 매크라인을 사용할 수밖에 없으며, 둘째, 취약한 버전이 설치되어 있어도 홈페이지 사용에 문제가 없는 것이 이유입니다. 이런 이유 때문에 취약한 버전의 매크라인이 설치된 PC는 이번 Lazarus 조직의 해킹 사례처럼 악성코드 감염 가능성이 존재합니다.

사실 일반 사용자에게 매크라인은 홈페이지 사용을 위해 필수로 설치해야 하는 프로그램 중 하나일 뿐 해당 프로그램이 정확히 어떤 프로그램인지 인지하고 있을 가능성은 낮을 것이기 때문에 이점을 고려하면 매크라인의 설치 방식을 변경하는 것에 고민이 필요합니다. 예를 들면 수동 설치 후 업데이트가 발생하면 자동 업데이트 되도록 하거나 만약 이 방식이 어렵다면 홈페이지 접속할 때 설치된 매크라인 버전을 체크한 후 구 버전이면 사용자에게 창을 띄워서 최신 버전을 설치하도록 유도하는 방법을 사용할 수 있을 것입니다.

이는 비단 매크라인의 문제만이 아니며, 유사한 방식으로 설치되는 모든 프로그램이 가지고 있는 공통적인 문제입니다. 사용자의 편의성은 곧 보안과 연결되므로 매크라인의 사례를 타산지석으로 삼아 사용자의 편의성과 보안을 동시에 높일 수 있는 프로그램 설치 방식에 대해서 고민이 필요합니다. 만약 지금의 설치 방식을 고수한다면 Lazarus 조직이 해킹에 악용할 것이며, 피해도 반복될 것입니다. 어쩌면 지금보다 더 큰 피해와 국가 안보를 위협하는 상황이 발생할 수 있습니다.

### 3. 안랩의 대응현황

지금의 해킹은 고도화, 조직화되어 있어서 안랩만으로 대응하는 것은 한계가 있습니다. 해킹 주체가 국가의 지원을 받는 해킹 조직이라면 더욱 그렇습니다.

안랩은 Lazarus 조직의 이니세이프 취약점 악용에 이어 매크라인 취약점 악용도 해킹 대상이 고객이라면 점검에 필요한 정보를 공유하고 있으며, Lazarus 조직처럼 국가가 배후인 것으로 알려진 해킹 조직에 대응하기 위해서 국가기관과 정보 공유 및 협업하고 있습니다. 예를 들어 안랩은 매크라인 취약점과 관련된 악성코드 샘플이 수집되거나 C2가 발견되면 국가기관에 공유하여 피해 확산 방지 및 대응에 기여하고 있습니다.

#### (1) 악성코드 진단현황

본 보고서에서 언급한 악성코드는 아래와 같이 진단 중입니다.

File Name	Hash(MD5)	진단명	진단 버전
java8.exe	08883d753154589a76347fda9e2799c0	Infostealer/Win.Outlook	2023.04.09.03
cryptra.cpl	0ade34f05a150af07e3406b719e18fb8	Trojan/Win.LazarLoader	2023.05.16.00
DevSync.cpl	1883e0ab8e2c8ee3e2dccfd7bf95fcfe	Trojan/Win.Lazardoor	2023.02.17.03
scrapkivsc.dll	19063f1f2136fd32e37371cdb2d871cd	Trojan/Win.LazarLoader	2023.05.17.00
pchealthsvc.dll	1dd5e8ff30df1ed5fc1b303c39a92e65	Trojan/Win.LazarLoader	2023.04.29.00



softoknhelp.dll	1e15d4af016a767456738a311a55415a	Trojan/Win.LazarLoader	2023.05.17.00
Adobe.dat	1f58b5a891188e8455baf734829b2b08	Trojan/Win.Lazardoor	2023.04.14.03
vuproc.sys	1fdd194c628cc841289be38d2188c502	Trojan/Win.Akdoor	2023.01.21.00
SharmIpcsvc.dll	25d93e899a13bd24d1e3701941b036c0	Trojan/Win.LazarLoader	2023.04.18.01
DUI70.dll	2b0bb56e4115a4c435b37dcad7ab7310	Trojan/Win.LazarLoader	2023.05.31.03
rasgreeng.dll	2d32c64bf80090c8f9b29e1f49426161	Trojan/Win.Lazardoor	2023.04.18.03
sgrmlpac.exe	4c3628722d46bfaa80af2d741c6d7657	정상	
pagefile.sys	4d7e1de48b2392344f6c1eddf898801a	Trojan/Win.Lazardoor	2023.04.01.00
DIFxdgsvc.dll	53fe57f78f41be54a7ca9cd0909882ff	Trojan/Win.LazarLoader	2023.04.18.01
ARM.dat	5c196692a1808f3f4c3be1ac9363d55f	Trojan/Win.Lazardoor	2023.03.15.04
VPWalletSvc.dll	612a469370873160d0844d01f614f176	Trojan/Win.LazarLoader	2023.05.17.00
dnproc.sys	678224aff0209d1f84d78a8d4f75206f	Trojan/Win.LazarLoader	2023.04.01.00
TieringEngineService.exe	6d5326ffb6c208ff0766eb2666755a6c	정상	
DevSync.cpl	704629d0660259a7818a7af580c416db	Trojan/Win.Lazardoor	2023.02.17.03
SgrmLpac.exe	7e4f5ee531a80e80b2ab7cc2f5621eec	정상	
kqproc.sys	858cf11d43fc34a8dcfc503cf776a5e8	Trojan/Win.Lazardoor	2023.02.08.01
dfrgui.exe	87db68334feab8038a7e147296bc56c6	정상	
FastCopy.exe	8a7eb3d1faddb2b9ff04e882044d1b97	Trojan/Win.Agent	2023.06.09.02
NHK.dll	9a2df4f0757ece108004bb36f5c9e620	Trojan/Win.Lazardoor	2023.04.14.03
DRM.dll	9d8b31971292bc0edc2057b0b5fc19e6	Trojan/Win.Lazardoor	2023.02.17.03
CameraSettingsUIHost.exe	9e98636523a653c7a648f37be229cf69	정상	
cryptini.cpl	ac2de04a2c272acacbac232cf657f0a6	Trojan/Win.LazarLoader	2023.05.16.00
smartaudios.dll	b04db343a683a8f3e3baad35679e3cb9	Trojan/Win.LazarAgent	2023.06.01.03
usosh.cpl	b085f5131ec7d83248f8be00f9c6143c	Trojan/Win.Lazardoor	2023.05.16.00
dnproc.sys	b0e8073bbb56d16535a05f8afece8860	Trojan/Win.Lazardoor	2023.02.08.01
sxshared.dll	b383ec8e053995b210d275c1e5cdaf91	Trojan/Win.Injector	2023.07.22.00
ESENT.dll	b855099cf41aa24da1812a3ca3c2d4d1	Trojan/Win.Lazardoor	2023.08.10.02
sasIGSSAPIsvc.dll	ba098924fbfbe949582ee390e4158b49	Trojan/Win.LazarLoader	2023.05.17.00
Rendering.dll	bb7cba29c83409d1a264b96670f06663	Trojan/Win.LazarLoader	2023.03.29.03
ARM.cpl	bbef84ae91c1e0a5f5032fced2faf4ba	Trojan/Win.LazarLoader	2023.04.01.00
Acrobat.tmp	bd444b440c96524dce8c941fc0d54713	Infostealer/Win.Browser	2023.03.15.04
Replicate.dll	c0f12fb677bbaaf2be09ef30a96885c6	Trojan/Win.LazarLoader	2023.04.01.00
winhttp.dll	c6498bd362b4d56b73bc8eee2342f3c6	Trojan/Win.LazarLoader	2023.07.21.00
DevSync.cpl	cfbb7cb382a71ad377a64cc9d16238e6	Trojan/Win.Lazardoor	2023.02.20.03
pick.dat	d2eb518419bc9baceb70c24216c7fa22	Trojan/Win.LazarLoader	2023.02.08.03
Adobe.dll	da5fc6ae361affa50ed473f845f26eb6	Trojan/Win.Lazardoor	2023.04.14.03
SFBAPPSDK.DLL	f18fe45ba93e3116ba989d5f591dfdda	Trojan/Win.LazarLoader	2023.04.01.00
nsldapsvc.dll	f30fee6aefc4fed05c68f9077154a10a	Trojan/Win.LazarLoader	2023.05.17.00
Pester.dll	f6b286f5457e1b22bbe96abfe99e80e4	Trojan/Win.LazarLoader	2023.04.22.01
NHK.dll	fa499a7ee3c13c88973d7550b332de54	Trojan/Win.LazarLoader	2023.05.16.00

[표 9] Lazarus 조직의 악성코드 진단 현황

## (2) C2 정보

[참고] 안랩이 인지한 C2 는 대부분 호스팅 IP 또는 도메인이므로 차단할 경우 정상 접속도 차단되는 부작용이 발생할 수 있기 때문에 차단은 권장하지 않습니다. 다만 점검 목적으로 사용할 수 있지만 이 경우에도 악성코드 진단 정보와 함께 사용하는 것을 권장합니다.

### ● C2 도메인

yoohattanet.kr (111.92.189.42)  
www.starsportsmall.co.kr (210.116.114.30)  
www.siriuskorea.co.kr (119.205.211.243)  
www.samdb.or.kr (210.116.111.163)  
www.oci.co.kr (52.78.224.225:443)  
www.mujae.com (218.232.94.40)  
www.medric.or.kr (14.52.99.233)  
www.kyungrok.com (121.78.190.61)  
www.imyonggosi.com (14.0.82.6)  
www.hds-secu.com (203.251.81.11)  
www.hanlasangjo.com (121.189.14.244)  
www.foodcutter.co.kr (115.68.53.77)  
www.droof.kr (111.92.188.166)  
www.circulation.or.kr (117.52.153.164)  
www.bi-nex.com (211.239.157.44)  
www.bcdm.or.kr (125.141.204.43)  
www.asps.co.kr (110.45.156.101)  
www.artbeads.co.kr (211.111.33.60)  
warevalley.com (20.41.99.17)  
www.scoutpeople.co.kr (121.254.168.87)  
swt-keystonevalve.com (111.92.189.45)  
safemotors.co.kr (112.175.29.157)  
safemotors.co.kr (112.175.29.151)  
ps-s.kr (222.122.213.155)  
pms.nninc.co.kr (114.207.112.19)  
online-plchat.com (13.125.103.101)  
oneline.kr (182.162.70.209)  
mot.korea.ac.kr (210.116.92.175)  
mainbiz.or.kr (14.63.217.197)  
koreaerb.or.kr (119.205.211.85)  
hiwoosung.com (221.139.104.54)  
himacsdesign.com (112.175.29.152)  
www.kiwiman.co.kr (219.253.141.142)

## Operation Dream Magic 보고서

lifelong.shu.ac.kr (220.68.211.111)  
www.apluslife.co.kr (112.175.115.116)  
hicar.kalo.kr (211.43.14.75)  
grandgolf.co.kr (211.231.58.34)  
fric.kangwon.ac.kr (192.203.144.135)  
celemics.com (203.245.24.18)  
busangcc.or.kr (219.253.141.135)

### ● C2 IP

183.111.138.158:443  
61.111.3.33:443  
211.233.63.138:443  
218.232.94.226:443  
210.118.194.131:443  
119.205.211.85:443  
13.125.42.157:443  
14.63.217.197:443  
222.122.213.155:443  
211.119.189.71:22  
210.109.101.16:80  
200.200.0.21:3000  
112.175.29.157:443

### ● C2 Full URL

www.happinesscc.com/mobile/include/func.asp (211.107.159.74)  
www.hankooktop.com/ko/company/info.asp (210.109.101.16)  
yoohannet.kr/min/tmp/process/proc.php (111.92.189.42)  
ps-s.kr/mall/policy/privacy\_data.asp (222.122.213.155)  
hiwoosung.com/mall/board\_data/notice/img\_view.php (221.139.104.54)

## (3) 권장사항

### 1) 행위 기반 탐지 기능 사용

V3의 행위 기반 탐지 기능은 실행 중인 프로세스가 기준에 부합하는 악성 의심 행위를 발생시킬 경우 해당 프로세스를 탐지 및 종료하여 피해를 최소화할 수 있으며, 백신의 시그니처 진단법 한계도 보완할 수 있습니다.

예를 들어 랜섬웨어에 감염되면 파일이 암호화되는 피해가 발생하지만 V3의 행위 기반 탐지 기능을 사용 중 이라면 랜섬웨어의 파일 암호화 기능이 동작할 시점에 해당 프로세스를 탐지 및 종료하여 파일 암호화를 차단할 수 있습니다.

안랩은 랜섬웨어 감염 PC에서 수집한 로그를 분석한 결과 V3의 시그니처 진단법에서 랜섬웨어를 진단하지 못했지만 행위 기반 탐지 기능을 사용했다면 랜섬웨어 실행으로 인한 파일 암호화 피해를 차단할 수 있었던 수 많은 사례를 경험했습니다.

하지만 V3의 행위 기반 탐지 기능에서 사용하는 기준도 분석가가 꾸준히 관리 및 업데이트해줘야 하는 한계는 있습니다. 바꿔 말하면 악성코드 실행으로 발생하는 의심 행위를 100% 탐지할 수 없다는 의미이기도 합니다. 또한 해당 기능은 PC에서 발생하는 행위를 모니터링 하다가 기준에 부합되는 행위를 탐지하므로 오탐이 발생할 수 있으며, 이로 인해 불편이 발생할 수도 있지만 오탐이 발생한 경우 안랩으로 신고하여 오탐이 발생한 기준을 제외하거나 좀더 정교하게 설정할 수도 있습니다.

고객마다 사정이 다 다르기 때문에 강요할 순 없지만 특별한 이유가 없다면 V3의 행위 기반 탐지 기능을 사용함으로써 얻을 수 있는 이득이 더 크므로 가급적이면 해당 기능을 사용하는 것을 권장합니다. 뿐만 아니라 침해 사고 발생 시 사용할 수 있는 로그를 기록하므로 이 또한 해당 기능의 장점이라고 할 수 있습니다.



[그림 3] V3 행위 기반 진단 사용

이번 Lazarus 조직의 매직라인 취약점에 의한 악성 행위도 V3의 행위 기반 탐지 기능을 통해 인지하고 대응할 수 있었습니다.

## 2) 로그 저장 기간 설정 필요

안랩은 이번 매직라인 취약점 악용 사례를 추적하고 분석하면서 일부 기업의 협조로 수집한 로그를 분석한 결과 로그 보관 기간이 4일로 설정되어 있어서 소량의 유의미한 정보만 저장되어 있었기 때문에 분석에 어려움이 있었습니다. 위에서 설명한 것처럼 고객마다 사정이 다 다르기 때문에 강요할 순 없지만 특별한 이유가 없다면 로그 보관 기간을 60 ~ 90일로 설정하는 것을 권장합니다. 로그 보관 기간을 길게 설정하면 침해사고 발생으로 로그를 분석할 때 많은 도움이 됩니다.



[그림 4] V3의 로그 보관 기간 설정

## 4. 매직라인 점검 및 조치

매직라인은 자동 업데이트를 통한 취약점 해결을 지원하지 않으므로 번거롭지만 사용자가 아래 단계를 수행하여 점검 및 조치할 수 있습니다.


- Step 1. 매직라인 설치 여부 및 버전 확인
- Step 2. 매직라인 삭제
- Step 3. 매직라인 설치

### Step 1. 매직라인 설치 여부 및 버전 확인

매직라인 설치 여부 및 버전은 [제어판 → 프로그램 삭제 또는 변경]에서 확인할 수 있습니다. 아래 [그림 5]의 예시에서 PC에 설치된 매직라인의 버전은 v1.0.0.20으로 v1.0.0.28보다 낮으므로 Lazarus 조직에 의한 해킹 가능성이 있기 때문에 최신 버전의 매직라인 설치가 필요합니다.

#### 프로그램 제거 또는 변경

프로그램을 제거하려면 목록에서 선택한 후 [제거], [변경] 또는 [복구]를 클릭하십시오.

구성 ▾	이름	게시자	설치 날짜	크기	버전
	 MagicLine4NX	Dreamsecurity, Inc.	2023-05-03	17.0MB	1.0.0.20

[그림 5] 매직라인 버전 확인

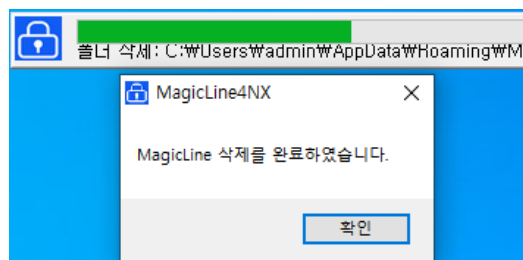
### Step 2. 매직라인 삭제

해킹 가능성이 있는 취약한 버전의 매직라인을 삭제 방법은 아래 두 가지입니다.

- 매직라인 삭제 프로그램 사용
- [제어판 → 프로그램 삭제 또는 변경]에서 삭제

#### 1) 매직라인 삭제 프로그램 사용

드림시큐리티는 자사 홈페이지를 통해서 매직라인 삭제 프로그램 제공하고 있으므로 다운로드한 후 실행하면 아래 [그림 6]처럼 매직라인을 삭제한 후 완료 팝업을 띄웁니다.



[그림 6] 매직라인 삭제

매직라인 삭제 프로그램은 아래 URL 에서 다운로드 가능합니다.


- 매직라인 삭제 프로그램 다운로드
- [https://www.dreamsecurity.com/download/MagicLineNX\\_Uninstall.exe](https://www.dreamsecurity.com/download/MagicLineNX_Uninstall.exe)

## 2) [제어판 → [프로그램 삭제 또는 변경]에서 매직라인 삭제

드림시큐리티에서 제공하는 매직라인 삭제 프로그램을 사용한 삭제는 몇 단계를 수행해야 하므로 번거로울 수 있으므로 [제어판 → 프로그램 삭제 또는 변경]에서 악성코드 감염 가능성이 있는 취약한 버전의 매직라인을 삭제할 수 있습니다. (아래 [그림 7] 참고)

### 프로그램 제거 또는 변경

프로그램을 제거하려면 목록에서 선택한 후 [제거], [변경] 또는 [복구]를 클릭하십시오.

구성 ▾	제거/변경				
이름	게시자	설치 날짜	크기	버전	
 MagicLine4NX	Dreamsecurity, Inc.	2023-05-03	17.0MB	1.0.0.20	

[그림 7] 매직라인 삭제

## Step 3. 매직라인 설치


악성코드 감염에 취약한 버전의 매직라인을 삭제한 후 최신 버전의 매직라인을 설치해야 하며, 아래 URL 에서 다운로드 및 설치할 수 있습니다.

- 매직라인 설치 프로그램 다운로드
- [https://www.dreamsecurity.com/download/magicline4nx\\_v1.0.0.28\\_setup.exe](https://www.dreamsecurity.com/download/magicline4nx_v1.0.0.28_setup.exe)

매직라인 설치가 정상 완료되면 [제어판 → 프로그램 삭제 또는 변경]에서 매직라인의 버전을 확인할 수 있으며, 예시로 아래 [그림 8]에서 설치한 매직라인 버전은 1.0.0.28 이므로 Lazarus 조직의 매직라인 취약점을 악용한 워터링 홀로부터 안전합니다.

### 프로그램 제거 또는 변경

프로그램을 제거하려면 목록에서 선택한 후 [제거], [변경] 또는 [복구]를 클릭하십시오.

구성 ▾	제거/변경				
이름	게시자	설치 날짜	크기	버전	
 MagicLine4NX	Dreamsecurity, Inc.	2023-07-27	13.3MB	1.0.0.28	

[그림 8] 매직라인 설치

지금까지 매직라인 점거 및 조치에 대해서 설명했지만 매직라인 삭제 프로그램과 설치 프로그램이 분리되어 있어 사용자가 삭제와 설치 과정을 진행하기엔 번거로움이 있습니다. 사용자의 편의성을 위해서 삭제와 설치를 하나의 프로그램으로 통합하여 제공할 필요가 있습니다.

## 5. Case Study

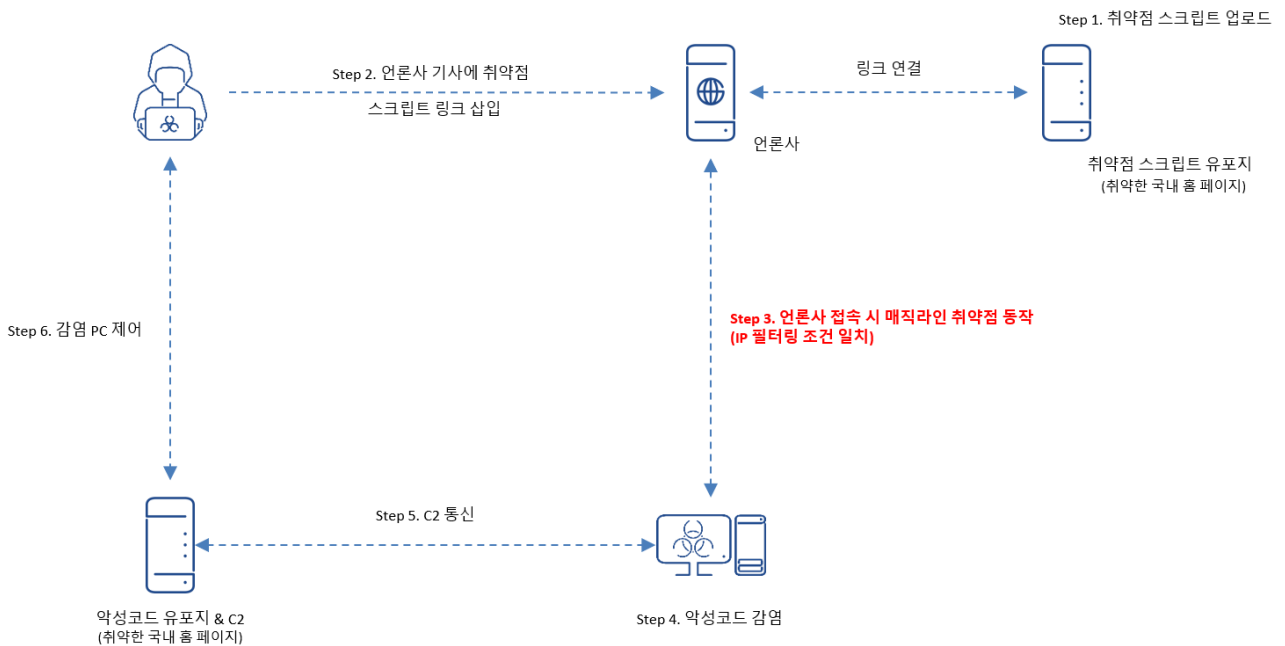
안랩이 일부 기업의 협조로 수집한 로그, 악성코드 샘플, 자사 ASD(AhnLab Smart Defense)에 수집된 정보 그리고 국가기관과의 정보 공유 및 협업을 통해서 인지한 정보 등 다양한 정보를 분석한 결과를 설명했습니다.

- 매직라인 취약점 악용 과정
- 악성코드 상세 분석
- Lazarus 조직으로 판단하는 근거

### (1) 매직라인 취약점 악용 과정

매직라인 취약점을 악용한 워터링 홀은 아래 [그림 9]의 총 6 단계로 정리할 수 있으며, 국내 취약한 홈페이지가 악성코드 유포지, C2 로 악용됐습니다. 그리고 언론사 기사 클릭할 때 악성코드에 감염되는 조건은 IP 필터링과 매직라인 취약점이 존재해야 합니다.

안랩은 일부 기업의 협조로 로그를 수집하여 분석했지만 아래 [그림 9]을 그리는 과정에서 초기 유입 단계의 흔적(ex, 언론사 접속)이 삭제 의심으로 존재하지 않았거나, 로그 보관 기간이 4 일로 설정되어 있어 과거 로그는 삭제, 안랩이 매직라인 취약점을 탐지한 후 기업에 알려서 로그를 수집할 시점에 이미 악성코드 흔적 삭제 등의 문제로 완벽하게 도식화하지 못한 점은 한계입니다.



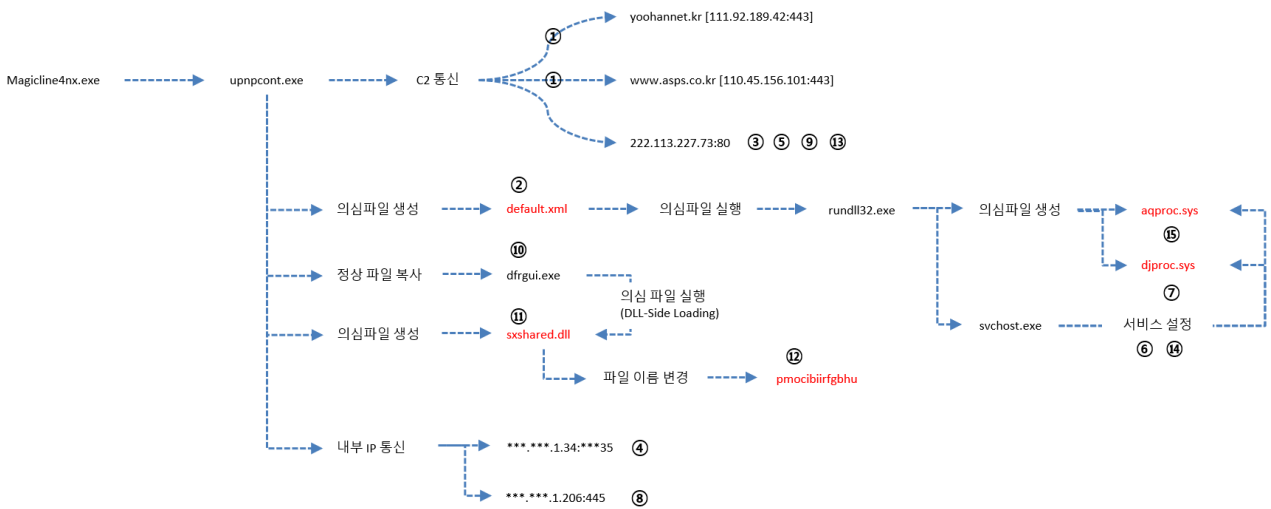
[그림 9] Lazarus 조직의 매직라인 취약점을 악용한 워터링 홀 과정

### 1) Case Study 1: 2023년 02월 제조 A기업 PC 3

아래 [그림 10]은 제조 A 기업의 PC 3 에서 수집한 로그를 분석하여 매직라인 동작 시점의 악성 행위를 도식화한 그림으로 ① ~ ⑮의 의미는 수집한 로그를 분석하여 주요 악성 행위의 순서를 넘버링한 것으로 ⑥, ⑦, ⑭, ⑮는 이니세이프 취약점을 악용한 해킹과 동일합니다. 이니세이프 취약점 사례에서 파일명의 맨 처음 2 문자만 다르며, DLL 이지만 확장자는 SYS 인 악성코드(\*\*proc.sys)가 사용됐는데 이 특징은 한국인터넷진흥원(KISA)에서 발간한 보고서 TTPs #10: Operation GoldGoblin 의 21 페이지에 설명돼 있습니다.

#### [KISA] TTPs #10: Operation GoldGoblin - 제로데이 취약점을 사용해 선별적으로 침투하는 공격전략 분석

➤ <https://thorcet.notion.site/TTPs-10-Operation-GoldGoblin-bab695345e984edbb8fe5e16e36face6?pvs=4>



[그림 10] 제조 A 기업 PC 3 의 악성 행위 도식화

위 [그림 10]에는 Lazarus 조직이 의도한 기능 수행 완료 후 C2(222.113.227.73:80)와 통신하는 일정한 패턴이 존재하며, 해당 패턴을 정리하면 아래와 같습니다.

- ② 의심파일(default.xml) 생성 → ③ C2(222.113.227.73:80) 통신
- ④ 내부 IP 통신(\*\*\*.\*\*\*.1.34:\*\*\*35) → ⑤ C2(222.113.227.73:80) 통신
- ⑥ 서비스 설정 → ⑦ 의심파일(djproc.sys) 생성 → ⑧ 내부 IP 통신(\*\*\*.\*\*\*.1.206:445) → ⑨ C2(222.113.227.73:80) 통신
- ⑩ 정상 파일 복사(dfrgui.exe) → ⑪ 의심 파일 생성(sxshared.dll) → ⑫ 파일 이름 변경(sxshared.dll → pmocibiirfgbhu) → ⑬ C2(222.113.227.73:80) 통신

위 [그림 12]에서 붉은색으로 표시된 악성코드는 아쉽게도 확보 실패했지만 로그 분석에서 정상 dfrgui.exe 와 악성 sxshared.dll 이 DLL-Side Loading 기법을 통해 실행됐음을 확인했으며, dfrgui.exe 가 실행될 때 3 개의 인자값을 사용하는 특징이 있습니다. (아래 [표 10] 참고) 참고로 원래 정상 파일은 C:\Windows 및 하위 폴더에 존재하므로 그 외 경로에 파일이 존재할 경우 PC 를 점검해 볼 필요가 있습니다.

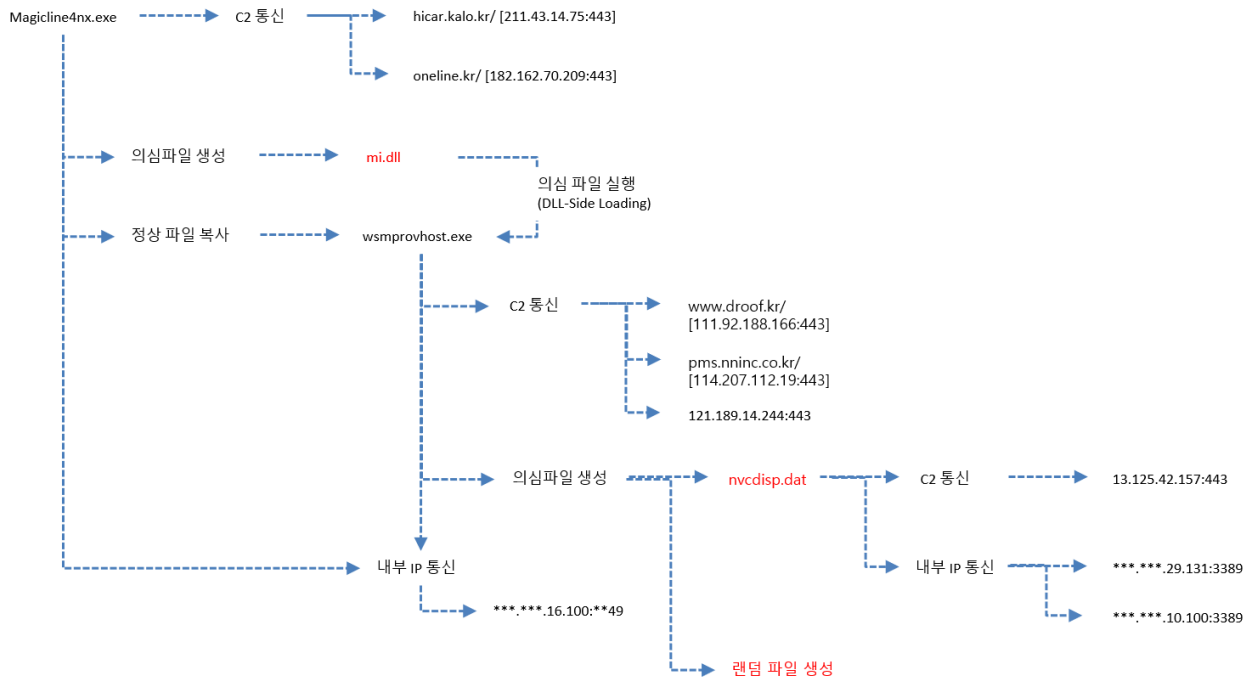


TIME	MESSAGE	PARENT_PROCESS	PARENT_PID	CURRENT_PROCESS	CURRENT_PID	TARGET1
2023-02-23 10:49:45	프로세스 생성	%programfiles%(x86)#dreamsecurity#magicline4nx# <b>magicline4nx.exe</b>	21284	%systemroot%\#syswow64#upnpcont.exe	14312	C:\ProgramData\USOS\shared#dfgrui.exe <b>profile 66FA6C0E6A6E83EC9Z6xEB-iCMG6GlrHuCM4zdee5hAHokHBvBiEx4MjH83lnsQQEKJlwakY18e6ly7N6p7mEG2iRMGpGJHH/yM6zc5e4hA1oknBpBiAx7gJL82MnrEQJ6JFwaoYkMe4I2bNxZ7uECiVcG6GlnHrCMrzYyesRAvok/BnxiKx7EjOs3XnvgQc#programdata#usoshared#dfgrui.exe</b>
2023-02-23 10:49:37	실행 파일 생성	%systemroot%\#syswow64#upnpcont.exe	14312	%systemroot%\#syswow64#cmd.exe	3480	c:\programdata\usoshared#dfgrui.exe
2023-02-23 10:49:04	프로세스 생성	%programfiles%(x86)#dreamsecurity#magicline4nx# <b>magicline4nx.exe</b>	21284	%systemroot%\#syswow64#upnpcont.exe	14312	C:\ProgramData\USOS\shared#dfgrui.exe <b>profile 66FA6C0E6A6E83EC9Z6xEB-iCMG6GlrHuCM4zdee5hAHokHBvBiEx4MjH83lnsQQEKJlwakY18e6ly7N6p7mEG2iRMGpGJHH/yM6zc5e4hA1oknBpBiAx7gJL82MnrEQJ6JFwaoYkMe4I2bNxZ7uECiVcG6GlnHrCMrzYyesRAvok/BnxiKx7EjOs3XnvgQMKJXwacY18e7IznNtp4=c:\programdata#usoshared#wsxshared.dll</b>
2023-02-23 10:48:25	실행 파일 생성	%programfiles%(x86)#dreamsecurity#magicline4nx# <b>magicline4nx.exe</b>	21284	%systemroot%\#syswow64#upnpcont.exe	14312	
2023-02-23 10:48:07	실행 파일 생성	%systemroot%\#syswow64#upnpcont.exe	14312	%systemroot%\#syswow64#cmd.exe	13012	c:\programdata#usoshared#dfgrui.exe

[표 10] 악성코드 실행 흔적

## 2) Case Study 2: 2023년 04월 언론 B사 PC 1

아래 [그림 11]은 언론 B사의 PC 1에서 수집한 로그를 분석하여 악성코드의 주요 행위를 도식화한 것으로 제조 A 기업 PC 3의 악성 행위와 비교했을 때 DLL-Side Loading에서 악용하는 파일 변경, 악성코드를 실행할 때 인지값을 1개 사용했다는 점을 제외하면 주요 행위는 동일합니다. 그리고 wsmprovhost.exe가 숫자, 영어 소문자로 조합된 파일(ex, %APPDATA%\#59ogv3c20h)을 생성하는 행위도 확인했지만 확보 실패로 분석은 불가능했습니다.



[그림 11] 언론 B사 PC 1의 악성 행위 도식화

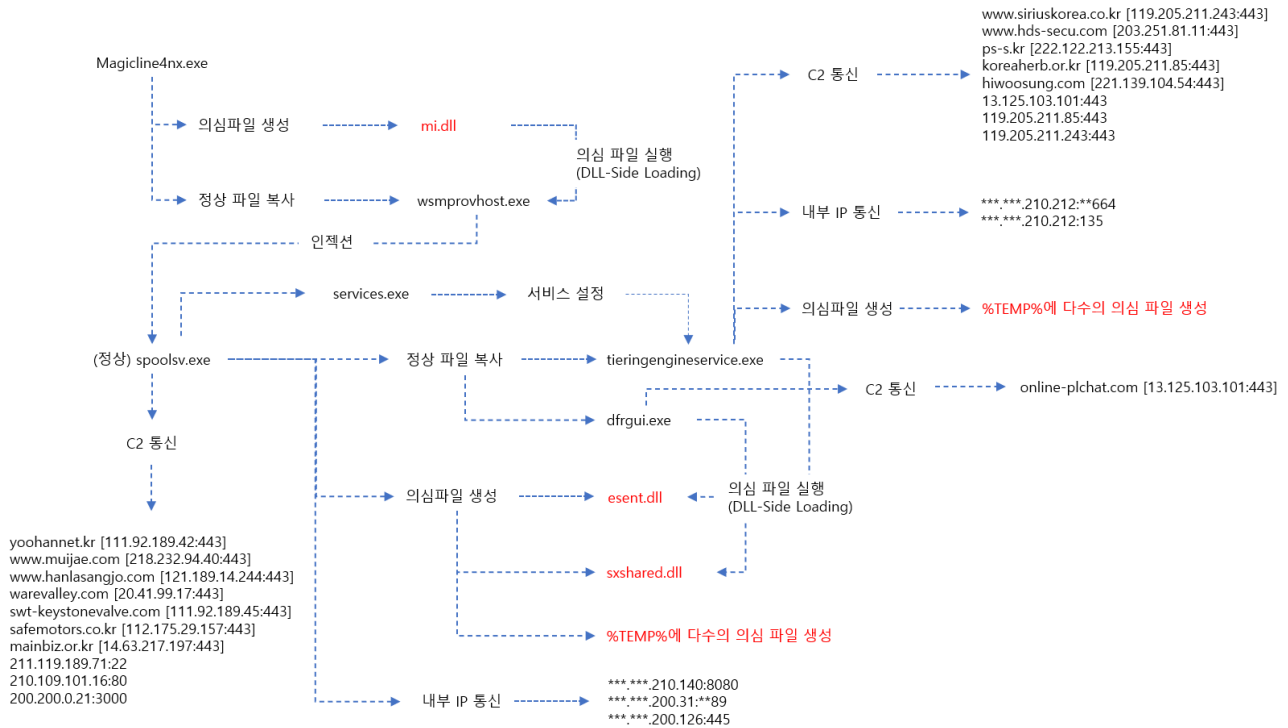
아래 [표 11]은 DLL-Side Loading 기법으로 악성 mi.dll을 실행할 때의 행위 정보로 인자값이 BASE64로 인코딩된 점은 제조 A 기업 PC 3와 동일합니다.

TIME	MESSAGE	CURRENT_PROCESS	CURRENT_PID	TARGET1	TARGET2
2023-04-07 17:05:57	프로세스 생성	%programfiles%(x86)#dreamsecurity#magicline4nx# <b>magicline4nx.exe</b>	5572	C:\ProgramData\PicPick#wsmprovhost.exe <b>Wwoz8j1+RweLIVb7P9D7lu9SWDamwAc0wH2KCtdfJwgzwxzKabVkJqiloOqTyBFB</b>	c:\programdata#picpick#wsmprovhost.exe

[표 11] 악성코드 실행 흔적

### 3) Case Study 3: 2023년 07월 금융 A 기업 PC 1

아래 [그림 12]는 금융 A 기업 PC 1에서 수집한 로그를 분석하여 도식화한 것으로 위 [그림 10] 제조 A 기업 PC 3, [그림 11] 언론 B 사 PC 1에 비해 악성코드 감염으로 인한 C2 통신, 내부 IP 통신, 악성코드 생성 등의 악성 행위가 반복됐습니다. 기존 사례와 비교하면 금융 A 기업의 PC 1에서 악성 행위의 반복은 Lazarus 조직이 해당 PC를 정찰 과정을 넘어서 핵심 해킹 대상으로 삼았다고 판단하고 있습니다. 또한 금융 A 기업 PC 1는 기존 제조 A 기업 PC 3, 언론 B 사 PC 1과 비교할 때 차이점은 정상 spoolsv.exe에 인젝션한 후 주요 악성 행위를 수행한다는 것입니다.



[그림 12] 금융 A 기업 PC 1의 악성 행위 도식화

### 4) Case Study에서 공통점

위 [그림 10] 제조 A 기업 PC 3, [그림 11] 언론 B 사 PC 1, [그림 12] 금융 A 기업 PC 1의 악성 행위는 각 사례마다 다소 차이는 있지만 전체적인 행위는 동일합니다. 또한 위 3가지 사례에서 주목해야 할 공통점은 악성코드 감염 후 내부 IP 통신 시도하는 흔적이 존재한다는 것입니다. Lazarus의 과거 행적, 목적, 의도 등을 종합했을 때 특정 IP로 반복적인 통신 시도는 해당 PC의 중요도 인지와 함께 악성코드 감염으로 제어권을 획득한 후 중요한 자료 탈취 목적으로 판단했습니다.

TIME	MESSAGE	CURRENT_PROCESS	CURRENT_PID	TARGET1	TARGET2
------	---------	-----------------	-------------	---------	---------

2023-08-09 11:42:07	네트워크 연결	%systemroot%\system32\spoolsv.exe	4164	0.0.0.0:53788	***.***.211.12:39* *
2023-08-09 11:41:37	네트워크 연결	%systemroot%\system32\spoolsv.exe	4164	0.0.0.0:53785	***.***.211.13:80
2023-08-09 11:41:37	네트워크 연결	%systemroot%\system32\spoolsv.exe	4164	0.0.0.0:53786	***.***.211.13:80
2023-08-09 11:41:35	네트워크 연결	%systemroot%\system32\spoolsv.exe	4164	0.0.0.0:53780	***.***.211.12:39* *
2023-08-09 11:29:45	네트워크 연결	%systemroot%\system32\spoolsv.exe	4164	0.0.0.0:0	<a href="http://www.mujae.com">www.mujae.com</a> / [218.232.94.40: 443]
2023-08-09 09:24:44	프로세스 열기	%systemroot%\system32\spoolsv.exe	4164		%systemroot%\system32\lsass.exe
2023-08-09 09:24:27	네트워크 연결	%systemroot%\system32\spoolsv.exe	4164	0.0.0.0:0	<a href="http://mainbiz.or.kr/">mainbiz.or.kr/</a> [1 4.63.217.197:443]

[표 12] spoolsv.exe(PID: 4164)에 의한 내부 IP 통신 시도 (연한 붉은색으로 표시)

## (2) 악성코드 분석

### 1) DLL-Side Loading

DLL-Side Loading 은 오래전부터 악성코드에서 사용해온 기법으로 정상 EXE 와 악성 DLL 이 한 쌍으로 동일한 경로에 존재하며, 악성 DLL 은 정상 EXE 가 실행하는 정상 DLL 과 똑같은 파일명으로 위장한 것이 특징입니다.

아래 매직라인 취약점을 악용한 악성코드에서 DLL-Side Loading 의 특징을 살펴보면 정상 EXE 와 악성 DLL 은 동일한 폴더에 존재하며, 정상 EXE 가 실행될 때 악성 DLL 을 실행함으로써 악성코드 생성, 정보 유출, C2 통신 등 악성 행위를 수행합니다. 참고로 아래 매직라인 취약점 악용에서 정상 파일은 원래 C:\Windows 및 하위 폴더에 존재하므로 다른 경로에 존재한다면 감염을 의심해 볼 수 있습니다. (아래 [표 13] 참고)

Case	File Path
1	%ALLUSERSPROFILE%\microsoft\crypto\rsa\camerasettingsuihost.exe (정상 EXE)
	%ALLUSERSPROFILE%\microsoft\crypto\rsa\dui70.dll (악성 DLL)
2	%SystemDrive%\Users\%ASD%\kyobo\wsmprovhost.exe (정상 EXE)
	%SystemDrive%\Users\%ASD%\kyobo\mi.dll (악성 DLL)
3	%SystemDrive%\ldplayer\ldplayer64\sgrmlpac.exe (정상 EXE)
	%SystemDrive%\ldplayer\ldplayer64\winhttp.dll (악성 DLL)
4	%ALLUSERSPROFILE%\microsoft\windows\servicesetting\tieringengineservice.exe (정상 EXE)
	%ALLUSERSPROFILE%\microsoft\windows\servicesetting\clusapi.dll 또는 ESENT.dll (악성 DLL)
5	c:\programdata\usoshared\dfogui.exe (정상 EXE)
	c:\programdata\usoshared\sxshared.dll (악성 DLL)

[표 13] 매킨라인 취약점 악용에서 DLL-Side Loading Case

위 [표 13]에서 Case 2, 3 처럼 일부 사례에서 해킹 대상 PC 에 설치된 프로그램 폴더를 임의의 경로로 복사한 후 해당 폴더에 악성코드를 생성하고 실행하는 사례도 있었습니다. 이렇게 하는 이유는 프로그램 폴더에 악성코드를 숨겨 놓음으로써 식별이 어렵도록 하려는 목적으로 판단했습니다.

Lazarus 조직이 DLL-Side Loading 기법으로 악성코드를 실행하기 위해서 악용한 5 개의 정상 EXE 에서 문자열을 살펴보면 아래 [표 14]와 같이 필요한 DLL 파일명이 명시되어 있으며, 정상 EXE 는 DLL 파일명만 동일하면 정상, 악성 여부에 상관없이 실행하기 때문에 악성 DLL 이 똑같은 DLL 파일명으로 위장하는 것입니다. 그리고 악용된 정상 EXE 는 모두 MS 에서 제작했으며, 윈도우 운영체제에서 사용하는 파일입니다.

파일 이름	문자열	파일이름	문자열
camerasettingsuihost.exe	UnInitProcessPriv UnInitThread DUI70.dll CoAddRefServerProcess CoReleaseServerProcess	wsmprovhost.exe	pcwum.dll MI_Application_Initialize mi_clientFT_V1 mi.dll iob_func
sgrmlpac.exe	api-ms-win-core-interlocked-l api-ms-win-core-libraryloader WINHTTP.dll api-ms-win-core-synch-l1-1-0 api-ms-win-core-handle-l1-1-0	tieringengineservice.exe	ESENT.dll OpenCluster GetClusterInformation CloseCluster CLUSAPI.dll
dfogui.exe	SxTracerShouldTrackFailure SxTracerDebuggerBreak SXSHARED.dll CoInitializeEx CoInitializeSecurity	SyncHost.exe	PropVariantToUInt32 PROPSYS.dll WinSync.dll GetStartupInfow UnhandledExceptionFilter

[표 14] DLL-Side Loading 에 악용한 정상 파일의 내부 문자열

위 [표 14]에서 DLL-Side Loading 에 악용한 정상 EXE 의 내부 문자열에는 각 파일이 실행할 때 필요한 DLL 이 명시되어 있습니다. 즉 정상 EXE 는 동일한 이름의 DLL 만 있으면 정상, 악성을 구분하지 않고 실행하기 때문에 악성코드에서 DLL-Side loading 이 가능한 것입니다.

DLL-Side Loading 기법으로 실행된 악성코드는 악성코드 생성, 정보 유출, C2 통신 등의 역할뿐만 아니라 정상 EXE와 정상 DLL 사이에서 다리 역할을 합니다. (악성) winhttp.dll에서 DLL-Side Loading을 구현한 방식은 아래 [표 15]와 같습니다. (악성) winhttp.dll는 함수의 기능이 없는 이름만 가지고 있으며, 아래 [표 15]의 3 단계를 거쳐 (정상) winhttp.dll로부터 실제 각 함수의 주소를 얻어와서 설정합니다. 예를 들어 (정상) SgrmLpac.exe가 Private1()를 호출하면 아래 [표 15]에서 Step 3.의 jmp eax에서 실제 해당 함수의 주소로 분기합니다.

정리하면 정상 EXE의 함수 호출 요청이 있으면 악성 DLL은 해당 요청을 정상 DLL로 전달하는 역할을 한다는 의미입니다.

(악성) winhttp.dll의 함수 리스트		코드	데이터
Name	Address		
DllCanUnloadNow	0000000180001BC0	Step 1. 데이터를 0xB848로 변경	; (BF) (악성) <winhttp.DllCanUnloadNow> ; 00007FFAEAB81BC0 <   B8 05000000   mov eax,5 ; 00007FFAEAB81BC5   C3   ret
DllGetClassObject	0000000180001BD0	mov [rsp+278h+Src], 0B848h ; mov [rsp+278h+var_254], 0E0FFh ;	
Private1	0000000180001BE0	mov edi, [r12+rax*4]	; (AF) (악성) <winhttp.DllCanUnloadNow> ; 00007FFAEAB81BC0 <   48:B8 000000C3CCCCCCCC   mov rax,CCCCCCCC3000000 ; 00007FFAEAB81BCA   CC   int3 ; 00007FFAEAB81BCB   CC   int3
SvchostPushServiceGlobals	0000000180001BF0	add rdi, rsi	
WinHttpRequestHeaders	0000000180001C00	mov rcx, rdi ; void * call sub_180001880 ;	
WinHttpRequestHeadersEx	0000000180001C10		
WinHttpAutoProxySvcMain	0000000180001C20		
WinHttpCheckPlatform	0000000180001C30	Step 2.	; (BF) (악성) <winhttp.DllCanUnloadNow> ; 00007FFAEAB81BC0 <   48:B8 000000C3CCCCCCCC   mov rax,CCCCCCCC3000000 ; 00007FFAEAB81BCA   CC   int3 ; 00007FFAEAB81BCB   CC   int3
WinHttpCloseHandle	0000000180001C40	정상 winhttp.dll의 DllCanUnloadNow 주소로 설정	
WinHttpConnect	0000000180001C50		
WinHttpConnectionDeletePolicyEntries	0000000180001C60		
WinHttpConnectionDeleteProxyInfo	0000000180001C70	mov r8d, 8 ; Size lea rdx, [rsp+278h+var_250] ; lea rcx, [rdi+2] ; void * call sub_180001880 ;	; (AF) (악성) <winhttp.DllCanUnloadNow> ; 00007FFAEAB81BC0 <   48:B8 509C0020F87F0000   mov rax,<winhttp.DllCanUnloadNow> ; 00007FFAEAB81BCA   CC   int3 ; 00007FFAEAB81BCB   CC   int3
WinHttpConnectionFreeNameList	0000000180001C80		
WinHttpConnectionFreeProxyInfo	0000000180001C90		
WinHttpConnectionFreeProxyList	0000000180001CA0		
WinHttpConnectionGetNameList	0000000180001CB0		
WinHttpConnectionGetProxyInfo	0000000180001CC0		
WinHttpConnectionGetProxyList	0000000180001CD0		
WinHttpConnectionSetPolicyEntries	0000000180001CE0		
WinHttpConnectionSetProxyInfo	0000000180001CF0	Step 3. 데이터를 0xE0FF로 변경	; (BF) (악성) <winhttp.DllCanUnloadNow> ; 00007FFAEAB81BC0 <   48:B8 309C0020F87F0000   mov rax,<winhttp.DllCanUnloadNow> ; 00007FFAEAB81BCA   CC   int3 ; 00007FFAEAB81BCB   CC   int3

[표 15] 악성 winhttp.dll에서 DLL-Side Loading 동작 방식

## 2) 악성코드와 인자값

Lazarus 조직의 악성코드 분석이 까다로운 이유는 파일 보호를 위해서 Themida나 VMProtect와 같은 상용 프로그램 사용하는 것도 있지만 악성코드 실행할 때 인자값을 사용하기 때문입니다. 바꿔 말하면 악성코드에서 사용하는 인자값을 모르면 악성코드 분석은 불가능하다는 의미입니다.

악성코드에서 사용하는 인자값은 악성코드 내부에 존재하는 암호화된 데이터 또는 암호화된 파일을 읽어서 복호화하는데 사용합니다.

### ① 2023년 02월 제조 A기업의 PC3

제조 A기업의 PC3에서 수집한 로그 분석에서는 DLL-Side Loading으로 악성 DLL인 sxshared.dll 실행에 색깔로 표시한 3개의 인자값을 사용했습니다.

- 악성코드 실행 방식

- dfrgui.exe profile 66FA6C0E6A6E83EC

9Z6xEB+icMG6GlrHuCM4zdee5hAHokHBvBiEx4MjH83InsQKEKJlwakYI8e6ly7N6p7mEG2iRMGpGJH  
H/yM6zcSe4hA1oknBpBiAx7gjL82MnrEQJ6JFwaoYkMe4I2bNxZ7uECiiVcG6GlnHrCMrzYyesRAvok/Br  
xiKx7EjOs3XnvgQMKJXwacYI8e7lznNtp4=

그 당시 제조 A 기업의 PC 3 에서 악성 DLL 인 sxshared.dll 는 확보 실패로 분석은 불가능했지만 올해 06 월 미상의 PC 에서 sxshared.dll 는 확보했으나 해당 악성 DLL 실행에 필요한 인자값은 확보 실패하여 분석은 불가능했습니다.

② 2023 년 04 월 언론 B 사의 PC 1

언론 B 사의 PC 1 은 악성 DLL mi.dll 실행에 색깔로 표시한 인자값 1 개를 사용했으며, 제조 A 기업의 PC 3 처럼 악성 DLL 에 맞는 인자값을 알지 못하면 분석은 불가능합니다.

- 악성코드 실행 방식

- wsmprovhost.exe **Wwoz8j1+RweLiVb7P9D7lu9SWDamwAc0wH2KCdtfJwgzxwzKabVkWqiloOqTyBFB**

3) 파일리스(Fileless) 악성코드

IT 기업에서 발견된 악성 winhttp.dll 은 파일리스(Fileless) 악성코드를 메모리에 생성과 생성된 파일리스 악성코드가 실행되도록 파일 재구성하는 역할만하며, 실제 악성 행위(자료 유출, 악성코드 생성, C2 통신 등)는 메모리에 생성된 파일리스 악성코드가 담당합니다. (아래 [표 15] 참고)

악성 winhttp.dll 의 파일리스 악성코드 호출	파일리스 악성코드의 시작 주소(Entry Point)
<pre> ; CODE XREF: sub_1800028D0+2D3↑j ; sub_1800028D0+2DD↑j ... mov     rax, [rdi] mov     ecx, [rax+28h] test    ecx, ecx jz      short loc_180002C22 cmp     dword ptr [rdi+20h], 0 lea     rax, [rcx+rbp] jz      short loc_180002C16 xor     r8d, r8d mov     rcx, rbp lea     edx, [r8+1] call    rax                     ; 메모리에 복호화된 PE의 Entry Point                 </pre>	<pre> 00001800AD948 48:895C24 08 mov qword ptr ss:[rsp+8],rbx 00001800AD94D 48:897424 10 mov qword ptr ss:[rsp+10],rsi 00001800AD952 57          push rdi 00001800AD953 48:83EC 20 sub rsp,20 00001800AD957 49:8BF8     mov rdi,r8 00001800AD95A 8BDA     mov ebx,edx 00001800AD95C 48:8BF1     mov rsi,rcx 00001800AD95F 83FA 01    cmp edx,1 00001800AD962 75 05     jne 1800AD969 00001800AD964 E8 93030000 call 1800ADCFC                 </pre>

[표 15] 악성 winhttp.dll 과 파일리스(Fileless) 악성코드의 관계

악성 winhttp.dll 이 메모리에 생성한 파일리스 악성코드의 C2 는 평문으로 존재하며, 아래와 같습니다.

- hxxps://www.happinesscc.com/mobile/include/func.asp (211.107.159.74)
- hxxp://www.hankooktop.com/ko/company/info.asp (210.109.101.16)
- hxxp://yoohannet.kr/min/tmp/process/proc.php (111.92.189.42)

① 감염 PC 식별자

파일리스 악성코드의 실행으로 C2 와 최초 통신할 때 전송하는 데이터는 아래와 같습니다.

- ex) **MT4ifnVkJAzUEDVpFWIEZNVOM1cKAlFqYndlMDCwT3IyaG1wbzB6UDM2amU4UzhT**

위 데이터가 생성되는 과정은 아래 [표 16]과 같으며, 생성된 데이터는 감염 PC 의 식별자로 사용합니다.

단계	설명
1. SIGNBTLG + GetTickCount()	<b>SIGNBTLG6e755acffxY22Qi9</b> (문자열 조합)
2. GetTickCount()	bwe070Or2hmpo0zP36je8S8S (24 바이트 길이의 암호키 생성)
3. XOR 암호화	SIGNBTLG6e755acffxY22Qi9 (XOR) bwe070Or2hmpo0zP36je8S8S → (암호화된 데이터) <b>1&gt;"~ud.5..ZEZQ.6UN3W</b>
4. BASE64 인코딩	(암호화된 데이터) 1>"~ud.5..ZEZQ.6UN3W → <b>MT4ifnVkAzUEDVpFWIEZNVOM1cKAIFq</b>
5. BASE64 인코딩	bwe070Or2hmpo0zP36je8S8S → <b>YndIMDcwT3IyaG1wbzB6UDM2amU4UzhT</b>
6. 데이터 조합	<b>MT4ifnVkAzUEDVpFWIEZNVOM1cKAIFqYndIMDcwT3IyaG1wbzB6UDM2amU4UzhT</b>

[표 16] 감염 식별자 생성 과정

### ② C2 통신

PC에서 생성한 감염 식별자 전송 이후 통신이 정상적으로 진행되면 파일리스 악성코드는 InternetReadFile()로 C2 명령이 포함된 웹 페이지를 읽어온 후 아래 [그림 13]의 코드로 웹 페이지에 존재하는 암호화된 C2 명령 추출하여 복호화 후 실행합니다. 하지만 실제 동적 분석에서는 정상적인 통신이 발생하지 않아 상세한 C2 명령은 확인하지 못했습니다.

```

20  memset(Str1, 0, sizeof(Str1));
21  memset(v17, 0, sizeof(v17));
22  memset(Destination, 0, 0x104ui64);
23  memset(v14, 0, sizeof(v14));
24  strcpy(
25  v20,
26  "<!DOCTYPE html><html><head></head><body marginwidth=\"0\" marginheight=\"0\" style=\"background-color:transparent\"><script>");
27  strcpy(v18, "</script></body></html>");
28  v5 = 0;
29  v15 = 12;
30  if ( !a1 || a2 < 0xA5 )
31  return 0i64;
32  v12 = sub_1800A2570(a1, v20);
33  if ( !v12 )
34  return 0i64;
35  if ( (_DWORD)v12 != (_DWORD)a1 )
36  return 0i64;
37  v13 = sub_1800A2570(a1, v18);
    
```

[그림 13] C2 명령 추출 및 복호화

만약 감염 조건을 완벽하게 갖춘 PC에서 파일리스 악성코드가 실행되어 C2와 정상적인 통신이 가능했다면 악성코드 생성, 자료 유출 등의 피해가 발생할 수 있으며, 아래 [표 17]은 설명을 뒷받침하는 근거입니다.

TIME	MESSAGE	CURRENT_PROCESS	CURRENT_PID	TARGET1	TARGET2
2023-08-06 10:11:44	실행 파일 생성	%systemroot%\system32\spoolsv.exe	4448	c:\programdata\microsoft\windows\servicesetting\wtieringengineservice.exe	
2023-08-06 10:11:44	실행 파일 생성	%systemroot%\system32\spoolsv.exe	4448	c:\programdata\microsoft\windows\servicesetting\Wesent.dll (악성 DLL)	
2023-08-06 10:04:58	네트워크 연결	%systemroot%\system32\spoolsv.exe	4448	0.0.0.0:50144	112.175.29.157:443
2023-08-06 10:04:58	네트워크 연결	%systemroot%\system32\spoolsv.exe	4448	0.0.0.0	safemotors.co.kr/ [112.175.29.157:443]

[표 17] 금융 A 기업 PC 1의 C2 통신과 악성코드 생성

### ③ 악성코드 생성

정상적인 C2 통신이 가능한 PC에서 확인할 수 있었던 증상은 [표 17]에서 설명한 것처럼 악성코드 생성 행위이며, 일부 기업의 협조로 로그를 수집할 시점에 생성된 악성코드는 대부분 삭제되어 확보 실패했습니다.



■ pick.dat(또는 cermgr.exe)

Lazarus 조직은 오픈 소스 파일의 소스 코드에 악성 기능을 삽입한 후 유포하는 사례가 있으며, NotePad++가 대표적입니다. NotePad++는 텍스트 편집 기능 이외에도 다양한 기능을 지원하기 때문에 널리 사용되고 있는 인기 텍스트 편집기로 해당 파일의 소스는 깃허브(<https://github.com/notepad-plus-plus/notepad-plus-plus>)에 공개되어 있어 누구나 소스를 다운로드하여 자신의 의도에 맞게 수정할 수 있습니다.

Lazarus 조직처럼 악성코드 제작에 오픈 소스 파일을 악용할 경우 악성 기능이 어디에 위치해 있는지 찾으려면 정상 파일과 비교하거나 파일을 처음부터 분석해야 하는 어려움이 발생할 수 있습니다. Lazarus 조직은 이를 의도하고 pick.dat 를 제작 및 유포한 것으로 판단했습니다. 아래 [그림 14]는 pick.dat 가 특정 경로의 파일을 읽어 들이는 악성 기능으로 pick.dat 뿐만 아니라 Lazarus 조직이 제작한 악성코드의 상당수가 악성 기능을 수행하기 위해서 인자값을 사용하거나 별도의 파일에 존재하는 악성 기능을 불러오는 방식을 사용합니다. 바꾸어 말하면 인자값을 모르거나 아래 [그림 14]와 같이 특정 파일을 확보 실패하면 분석은 불가능하다는 의미입니다.

```

push    eax
push    eax                ; String2
push    0AFCAB3C4h        ; int
call    sub_47C560         ; API 주소 구하기
                                ; https://github.com/tildedennis/malware/blob/master/neutrino_bot_5.1/api_hashes
add     esp, 0Ch
mov     CreateFileW_0, eax

                                ; CODE XREF: sub_47C6A0+577↑j
push    0
push    80h
push    3
push    0                  ; C:\\Temp\\{222A245B-E637-4AE9-A93F-A59CA119A75E}
push    1
push    80000000h
push    esi
call    eax ; CreateFileW_0 ; CreateFileW(): 파일 오픈, C:\\Temp\\{222A245B-E637-4AE9-A93F-A59CA119A75E}

```

[그림 14] pick.dat 의 파일 읽기

위 [그림 14]에서 {222A245B-E637-4AE9-A93F-A59CA119A75E}는 확보 실패하여 추가 분석은 불가능했지만 방산 B, IT F, 바이오 A 기업 등에서 pick.dat 의 감염과 악성 행위 흔적을 확인했습니다. (아래 [표 18] 참고)

TIME	MESSAGE	CURRENT_PROCESS	CURRENT_PID	TARGET1	TARGET2	설명
2023-05-29 15:43:27	네트워크 연결	c:\softforum\certst orage\cermgr.exe	206572	0.0.0.0:594 74	210.116.92.175:44 3	C2 통신
2023-05-29 15:43:27	네트워크 연결	c:\softforum\certst orage\cermgr.exe	206572	0.0.0.0:0	mot.korea.ac.kr/[2 10.116.92.175:443]	C2 통신
2023-05-29 15:47:36	네트워크 연결	c:\softforum\certst orage\cermgr.exe	206572	0.0.0.0:595 51	***.***.98.108:***3 5	내부 IP 통신

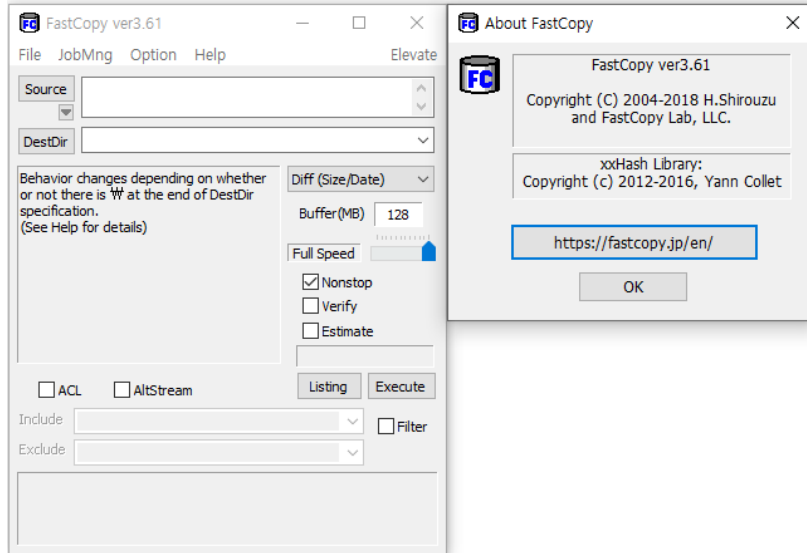
[표 18] IT F 기업에서 pick.dat(cermgr.exe)의 악성 행위 흔적



■ FastCopy.exe

공개된 파일을 악용했을 때 장점은 새로 개발하는 것보다 시간, 노력 등 리소스 소모를 줄일 수 있고, 무엇보다 해킹 주체를 특정하기 어렵도록 만들거나 숨길 수 있습니다.

FastCopy는 자료 복사 또는 백업할 때 사용할 수 있는 프로그램으로 <https://fastcopy.jp/>에서 다운로드할 수 있으며, 현재 최신 버전은 ver 5.3.0입니다. 이번 해킹에서 Lazarus 조직이 악용한 FastCopy의 버전은 Ver 3.61로 2023년 02월 바이오 A 기업, 2023년 07월 금융 A 기업 등 일부 기업에서 악용 흔적을 확인했습니다.



[그림 15] FastCopy 실행 화면

예시로 아래 [표 18]은 금융 A 기업의 PC 1에서 1238419.tmp(FastCopy.exe)로 생성 및 실행하여 ls.dat 파일을 ls.key로 복사하는 행위 흔적입니다. 하지만 ls.dat는 확보 실패로 해당 파일의 용도는 확인하지 못했습니다.

TIME	MESSAGE	CURRENT_PROCESS	CURRENT_PID	TARGET1
2023-07-19 00:15:30	DLL 로드	%systemroot%\system32\spoolsv.exe	1044	%systemroot%\temp\1238419.tmp
2023-07-19 00:15:30	프로세스의 비정상적인 실행	%systemroot%\system32\spoolsv.exe	1044	c:\windows\temp\1238419.tmp /modfile C:\ProgramData\ls.dat C:\ProgramData\ls.key
2023-07-19 00:00:35	실행 파일 생성	%systemroot%\system32\spoolsv.exe	1044	%systemroot%\temp\1238419.tmp

[표 19] 금융 A 기업의 PC 1에서 FastCopy 악용 행위 흔적

■ Acrobat.tmp

구글 크롬, MS 엣지, 파이어 폭스 등 다수의 브라우저에 저장된 정보(비밀번호, 북마크, 쿠키, 히스토리, 신용카드 정보, 다운로드 링크 등)를 추출하여 CSV로 저장하는 악성코드로서 깃허브(<https://github.com/idiotc4t/Reflective-HackBrowserData>)에 공개된 GO 언어 기반의 Reflective-HackBrowserData 소스로 제작됐습니다. (아래 [그림 16] 참고)

이름	TargetPath	Url	TotalBytes	StartTime	EndTime
	C:\Users\...Downloads\W	<-v ht ww	78786632	2023-04-1	2023-04-1
chrome_bookmark.csv	C:\Users\...Downloads\W	in_ ht ww	65401376	2023-04-1	2023-04-1
chrome_credit.csv	C:\Users\...Downloads\W	er- ht ww	95927184	2023-04-1	2023-04-1
chrome_download.csv	C:\Users\...Downloads\W	tuj ht ww	6840480	2023-04-1	2023-04-1
chrome_history.csv	C:\Users\...Downloads\W	ins ht ww	1762624	2023-04-2	2023-04-2
chrome_password.csv	C:\Users\...Downloads\W	5.2 ht ev	1.169E+09	2023-04-2	1601-01-0
microsoft_edge_credit.csv	C:\Users\...Downloads\W	tuj ht ev	1385632	2023-04-2	2023-04-2
	C:\Users\...Downloads\W	4r ht ni	9327984	2023-05-0	2023-05-0
	C:\Users\...Downloads\W	4r ht ub	16479392	2023-05-0	2023-05-0

[그림 16] Acrobat.tmp 실행으로 정보 추출 예시

브라우저의 자동 완성 기능은 각 홈페이지마다 사용자의 계정 정보를 저장한 후 재 접속 시 자동으로 입력해 주기 때문에 사용자에게 편리성을 제공하지만 보안 측면에서 좋은 기능은 아닙니다. Acrobat.tmp 와 같은 악성코드에 의해서 정보 추출이 가능하고 외부로 유출되어 악용 가능성이 있으므로 브라우저의 자동 완성 기능은 가급적 사용하지 않는 것이 좋으며 사용자 자신만의 비밀번호 규칙을 만들어서 사용하는 것이 좋습니다.

FastCopy.exe 처럼 공개된 자원을 악용하면 해킹 조직을 특정하는 것은 어려울 수 있습니다. 악성코드의 연관 관계가 파악되지 않은 상태에서 Acrobat.tmp 만으로 해킹 조직을 특정하는 것은 어려울 수 있지만 Lazarus 조직의 악성코드를 분석해본 경험이 있는 분석가라면 에서 자주 사용하는 패턴임을 의심해 볼 수 있습니다. 하지만 이 역시 의심이므로 확신으로 바꾸려면 해당 파일과 관련된 다양한 정보를 수집하고 분석하는 것이 필요하며, 많은 시간이 필요합니다. 이것이 Lazarus 조직이 공개된 자원을 사용하는 이유 중 하나입니다.

■ java8.exe

아웃룩에 저장된 이메일 계정 정보를 탈취할 목적의 악성코드로 방산 C 기업의 PC 에서 수집한 악성코드로 아웃룩 관련 레지스트리 키에 저장된 IMAP, POP3, HTTP, SMTP 등의 비밀번호를 추출하는 기능이 존재합니다.

java8.exe 이 Lazarus 조직의 악성코드라고 판단한 근거는 아래의 행위가 방산 C 기업의 PC 에서 확인됐기 때문입니다. usosh.cpl 는 Lazarus 조직의 악성코드로 악성 DLL 이므로 rundll32.exe 와 함께 실행됐으며, 해당 악성코드가 실행되면서 아웃룩에 저장된 이메일 계정 정보를 탈취할 목적의 악성코드인 java8.exe 를 생성한 것으로 판단했습니다.

Report Date	Process	Module	Behavior	Data
2023-04-07 16:33:12	rundll32.exe	usosh.cpl	Creates executable file	java8.exe

[표 20] java8.exe 생성 행위

(3) Lazarus 조직으로 판단하는 근거

매직라인 취약점을 악용한 워터링 홀의 주체가 Lazarus 조직이라고 판단한 근거는 아래와 같습니다.

- 악성코드 재구성 코드 동일
- \*.SYS 파일 생성 과정 동일

### 1) 악성코드 재구성 코드 동일

아래 [표 21]은 Lazarus 조직의 최근 3 년의 해킹 사고(2020 년 09 월 의료 A, 2023 년 02 월 미상, 2023 년 07 월 IT)에서 해당 조직으로 확인한 악성코드 사이에 유사성을 비교한 것으로 파일리스 악성코드를 복호화 후 메모리에서 실행 가능하도록 재구성하는 과정의 코드가 동일함을 알 수 있습니다.

2020 년 09 월 의료 A PCAudit.sys	2023 년 02 월 미상 fjproc.sys	2023 년 07 월 IT winhttp.dll
<pre> mov     edx, [rax+0B0h] add     rdx, rbx mov     eax, [rdx] test    eax, eax jz      short loc_180001B9E nop     dword ptr [rax+rax+00h]  ; CODE X ; DATA X  mov     r10d, eax mov     eax, [rdx+4] xor     r8d, r8d sub     rax, 8 add     r10, rbx lea     r9, [rdx+8] test    rax, 0FFFFFFFFFFFFFFFEh jbe     short loc_180001B92 nop     dword ptr [rax+00h]  ; CODE X  movzx   eax, word ptr [r9] mov     ecx, eax shr     eax, 0Ch and     ecx, 0FFFh cmp     eax, 3 jz      short loc_180001B72 cmp     eax, 0Ah jnz     short loc_180001B79 movsxd  rax, ecx add     [rax+r10], r11 jmp     short loc_180001B79  -----  movsxd  rax, ecx add     [rax+r10], r11d  ; CODE X ; sub_1f  mov     ecx, [rdx+4] inc     r8d sub     rcx, 8 mov     eax, r8d shr     rcx, 1 add     r9, 2 cmp     rax, rcx jb      short loc_180001B50  ; CODE X  mov     eax, [rdx+4] add     rdx, rax mov     eax, [rdx] test    eax, eax jnz     short loc_180001B53 </pre>	<pre> mov     edx, [rax+0B0h] add     rdx, rdi mov     eax, [rdx] test    eax, eax jz      short loc_1800395F8 xor     ebx, ebx  ; CODE X  mov     r10d, eax lea     r9, [rdx+8] mov     eax, [rdx+4] add     r10, rdi sub     rax, 8 mov     r8d, ebx test    rax, 0FFFFFFFFFFFFFFFEh jbe     short loc_1800395EC nop  ; CODE X  movzx   eax, word ptr [r9] mov     ecx, eax shr     eax, 0Ch and     ecx, 0FFFh cmp     eax, 3 jz      short loc_1800395CF cmp     eax, 0Ah jnz     short loc_1800395D3 add     [rcx+r10], r11 jmp     short loc_1800395D3  -----  add     [rcx+r10], r11d  ; CODE X ; sub_1f  mov     ecx, [rdx+4] inc     r8d sub     rcx, 8 mov     eax, r8d shr     rcx, 1 add     r9, 2 cmp     rax, rcx jb      short loc_1800395B0  ; CODE X  mov     eax, [rdx+4] add     rdx, rax mov     eax, [rdx] test    eax, eax jnz     short loc_180039593 </pre>	<pre> mov     edx, [rax+0B0h] add     rdx, rdi mov     eax, [rdx] test    eax, eax jz      short loc_1800026F8 xor     ebx, ebx  ; CODE X  mov     r10d, eax lea     r9, [rdx+8] mov     eax, [rdx+4] add     r10, rdi sub     rax, 8 mov     r8d, ebx test    rax, 0FFFFFFFFFFFFFFFEh jbe     short loc_1800026EC nop  ; CODE X  movzx   eax, word ptr [r9] mov     ecx, eax shr     eax, 0Ch and     ecx, 0FFFh cmp     eax, 3 jz      short loc_1800026CF cmp     eax, 0Ah jnz     short loc_1800026D3 add     [rcx+r10], r11 jmp     short loc_1800026D3  -----  add     [rcx+r10], r11d  ; CODE X ; sub_1f  mov     ecx, [rdx+4] inc     r8d sub     rcx, 8 mov     eax, r8d shr     rcx, 1 add     r9, 2 cmp     rax, rcx jb      short loc_1800026B0  ; CODE X  mov     eax, [rdx+4] add     rdx, rax mov     eax, [rdx] test    eax, eax jnz     short loc_180002693 </pre>

[표 21] 악성코드의 유사성 비교

## 2) \*.SYS 파일 생성 과정 동일

2020년 09월 PCAudit.sys를 생성하는 dllhost.exe (MD5: b0b8bef5c366657d59f002bb8b5b40b2)의 동적 분석에서 랜덤 파일명.XML을 생성한 후 %SYSTEM%W\*.SYS로 복사하는 행위를 확인했습니다. 2023년 02월 제조 A 기업 PC 3의 로그에서 동일한 행위만 발췌한 것으로 아래 [그림 17], [표 22]의 행위 정보를 비교했을 때 \*.SYS 악성코드 생성 방식이 동일한 것으로 판단했습니다.

Process Name	PID	Operation	Path	Result	Detail
cmd.exe	5960	CreateFile	C:\Temp\dnqmr.xml	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open
cmd.exe	5960	QueryBasicInformationFile	C:\Temp\dnqmr.xml	SUCCESS	CreationTime: 2023-05-14 오전 12:05:59, LastAccessTime: 2023-05-14
cmd.exe	5960	CloseFile	C:\Temp\dnqmr.xml	SUCCESS	
cmd.exe	5960	CreateFile	C:\Temp\dnqmr.xml	SUCCESS	Desired Access: Read Attributes, Delete, Synchronize, Disposition
cmd.exe	5960	QueryAttributeTagFile	C:\Temp\dnqmr.xml	SUCCESS	Attributes: A, ReparseTag: 0x0
cmd.exe	5960	QueryBasicInformationFile	C:\Temp\dnqmr.xml	SUCCESS	CreationTime: 2023-05-14 오전 12:05:59, LastAccessTime: 2023-05-14
cmd.exe	5960	CreateFile	C:\Windows\System32	SUCCESS	Desired Access: Write Data/Add File, Synchronize, Disposition: Op
cmd.exe	5960	SetRenameInformationFile	C:\Temp\dnqmr.xml	SUCCESS	ReplaceIfExists: True, FileName: C:\Windows\System32\PCAudit.sys
cmd.exe	5960	CloseFile	C:\Windows\System32	SUCCESS	
cmd.exe	5960	CloseFile	C:\Windows\System32\PCAudit.sys	SUCCESS	

[그림 17] 2020년 09월 의료 A에서 발견된 dllhost.exe의 행위 정보

TIME	MESSAGE	PARENT_PROCESS	PARENT_PID	CURRENT_PROCESS	CURRENT_PID	TARGET1
2023-02-23 10:27:46	실행 파일 생성	%systemroot%\syswow64\rundll32.exe	25372	%systemroot%\system32\rundll32.exe	8476	%systemroot%\system32\djproc.sys
2023-02-23 10:27:46	프로세스 생성	%systemroot%\syswow64\upnpcont.exe	14312	%systemroot%\syswow64\cmd.exe	11696	\??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1
2023-02-23 10:27:45		%systemroot%\syswow64\rundll32.exe	25372	%systemroot%\system32\rundll32.exe	8476	
2023-02-23 10:27:41	프로세스 생성	%systemroot%\syswow64\upnpcont.exe	14312	%systemroot%\syswow64\cmd.exe	16448	\??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1
2023-02-23 10:27:05	실행 파일 생성	%programfiles%\ (x86)\dreamsecurity\magicline4nx\magicline4nx.exe	21284	%systemroot%\syswow64\upnpcont.exe	14312	c:\programdata\default.xml

[표 22] 2023년 02월 제조 A 기업 PC 3의 행위 정보

## 6. 에필로그

이번 이슈를 추적하고 분석하는 과정에서 Lazarus 조직이 악성코드 유포에 매직라인 취약점을 악용한다는 것은 다수의 로그 분석을 통해서 확인했지만 매직라인의 취약점 상세 분석과 분석에 필요한 Exploit 코드를 확보하지 못한 것은 아쉬움이자 앞으로 해결해야 할 숙제입니다. 어쩌면 앞으로도 불가능할 수도 있는 숙제를 해결하기 위해서 Lazarus 조직의 매직라인 취약점 악용 사례에 대해서 계속 추적할 것이며, 확보할 수 있다면 상세 분석 진행 및 결과는 다양한 방법으로 공개할 계획입니다.

지금 해결하지 못한 숙제는 안랩이 민간 회사로서 분석과 추적의 범위와 한계, 해킹 대상 기업의 내부 사정 그리고 가장 중요한 Lazarus 조직의 빠른 흔적 삭제, 제한된 범위의 해킹 등 여러가지가 복합적으로 작용한

것입니다. 이는 국가의 개입과 Lazarus 조직의 매직라인 취약점을 악용한 해킹은 우리나라의 사이버 안보에 대한 고도화된 사이버 위협으로 해석할 수 있습니다.

고도화된 사이버 안보 위협에 대응하기 위해서는 해킹 대상 기업, IT 보안기업, 국가기관 등이 상호 신뢰를 바탕으로 정보 공유 및 협업이 필요합니다. 정보 공유 및 협업을 통해서 사이버 안보 위협으로 인한 피해를 최소화할 수 있으며, 대응, 복구에 들어가는 사회적인 자원(인력, 비용 등) 소모를 줄일 수 있습니다. 그리고 사이버 안보 위협이 물리적인 안보 위협으로 확대되는 것을 최소화할 수 있습니다.

## More security, More freedom

(주)안랩

경기도 성남시 분당구 판교역로 220 (우) 13493

대표전화 : 031-722-8000 | 구매문의 : 1588-3096 | 팩스 : 031-722-8901

[www.ahnlab.com](http://www.ahnlab.com)

이 보고서는 저작권법에 의해 보호받는 저작물로서 영리목적의 무단전재와 무단복제를 금합니다.

이 보고서의 내용의 전부 또는 일부 인용, 가공 시 안랩에서 발간된 보고서임을 밝혀 주시기 바랍니다.

이 보고서에 수록된 내용 또는 배포에 관한 모든 문의는 안랩(031-722-8000)으로 부탁드립니다.

해당 보고서는 <https://atip.ahnlab.com> 을 통해 이용할 수 있습니다.

© AhnLab, Inc. All rights reserved.