

# Index

Section		Page(s)
Abstract		2
I	Introduction	2 - 3
II	Case Summary & Global Impact	3 - 4
III	Attack Chain & Timeline	5 - 9
IV	STRIDE Threat Model Analysis	9 - 17
V	Encryption & Payload Analysis	18 - 20
VI	Ransom Note & Communication	20 - 21
VII	Security Engineering Reflections	22 - 23
VIII	Demo Script and Instructions	24
IX	Conclusion	25
References and Appendix		26 - 28

# NotPetya Ransomware Campaign:

## A Technical Analysis

Charissa Lau (z5411192)

July 20, 2025

### Abstract

NotPetya was one of the most damaging cyberattacks in history, causing over \$10 billion in damages worldwide in June 2017. From the M.E.Doc supply-chain breach to its worm-like propagation through EternalBlue and credential theft, this report offers a thorough technical analysis of NotPetya's kill chain, culminating in a wiper payload using AES-128 and Salsa20 encryption. We analyse vulnerabilities in spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege using the STRIDE threat model. The static ransom demanding \$300 in Bitcoin was not a viable method of decryption. We consider security flaws such as unpatched systems, flat networks, and weak credentials and suggest solutions like backups, patching, and segmentation. For educational purposes, ransomware mechanics are demonstrated in a secure Python demo. The lessons learned from NotPetya remains crucial in 2025 as AI-driven attacks and ransomware develops.

## I Introduction

Struck by a fast-spreading malware in June 2017, organizations worldwide had their critical infrastructures crippled within hours, disrupting shipping terminals, hospitals, airports, banks and government agencies. NotPetya, which was initially grouped as a variant of the Petya ransomware family, later proved its unique characteristics in term of scale, sophistication, and intent. Masquerading as a financially motivated ransomware, NotPetya's true nature is more accurately a destructive wiper <sup>[1]</sup>. Attributed to Russian military's Sandworm group, it was later classified as a nation-state cyberweapon against Ukraine that spiralled globally, causing up to 10 billion USD <sup>[2]</sup> in total damages.

From a technical perspective, Notpetya is a combination of malware techniques from cyberattacks earlier in the same year – Petya ransomware and WannaCry Worm, but with new capabilities. Mainly delivered through a popular Ukrainian tax software M.E.Doc, its supply-chain compromise allowed attackers to spoof a legitimate software update and tampered with trusted code <sup>[3]</sup>. Much like Petya's evil twin, Notpetya did not just infect a single system. Instead, it propagated as a self-spreading worm once inside a network, exploiting leaked NSA tools—EternalBlue and EternalRomance. EternalBlue was originally developed by the U.S. National Security Agency (NSA), targeting a vulnerability in

Microsoft's Server Message Block (SMBv1) protocol, known as MS17-010 – the same exploit used in the WannaCry Ransomware a month before NotPetya. Despite Microsoft's patch release back in March 2017, millions of systems remained vulnerable even by the end of 2018.

What made NotPetya far more devastating was its hybrid combination of remote code execution via SMB exploits with credential thefts, causing lateral movements. NotPetya dumped cached credentials from memory mimicking Mimikatz<sup>[4]</sup> – a post exploitation tool that extract plaintext passwords, hashes and other credentials from the Windows authentication system. This essentially allowed the malware to reuse stolen admin credentials to spread rapidly within shared networks even to fully patched machines<sup>[5]</sup>, using legitimate Windows tools like PS Exec and WMI. Unlike Petya, NotPetya was not profit-oriented but a brilliantly disguised wiper. While they both encrypted the Master Boot record (MBR), Master File Table (MFT) and the bootloader, NotPetya took it further by completely corrupting file systems using randomly generated encryption keys, leaving no decryption possible. With a hardcoded Bitcoin wallet, and an email address which was soon shut down, they demand \$300 worth of ransom in Bitcoin with no possible recovery of data<sup>[6]</sup>. Aimed at destructing data and operations rather than extortion, NotPetya was later distinctly classified as a wiper attack.

This report analyses the malware attack in depth and reflects on its security lessons. We delve into its global impact, dissect the attack chain and timeline, and map each stage to the STRIDE threat model and MITRE ATT&CK framework. We reconstruct its encryption mechanism and payload behaviour, as well as exploring how sound security engineering practices could have mitigated the incident. We also included a safe Python-based demonstration script simulating illustrating NotPetya's core behaviours for educational purposes.

## II Case Summary & Global Impact

Ukrainian companies, along with some global firms, were thrown into chaos when their systems suddenly stopped operating on June 27, 2017. An infected update for the M.E.Doc accounting software was distributed to thousands of Ukrainian businesses<sup>[7]</sup> around noon. After just a few hours, the malware had infected the internal networks of patient-zero systems and travelled to other countries. Some of the most notable victims were the world's largest shipping company A.P. Møller-Maersk, FedEx's subsidiary TNT Express, Merck, Ukraine's central bank, the power companies and airports, and the British and French construction and advertising firms WPP and Saint-Gobain, respectively. In Ukraine, the most affected participants were the government ministries and banks, while the international businesses faced damages wherever they had network links to Ukraine, VPN connections, or subsidiaries.

The NotPetya cyberattack had an immediate and far-reaching operational impact. At Maersk, for example, all 76 port terminals globally experienced significant disruption, as the company was forced to shut down its entire IT infrastructure in response to the malware's rapid spread, which encrypted or destroyed over 50,000 endpoints and servers<sup>[8]</sup>. This interruption halted Maersk's shipping operations, and manual processes had to be implemented to keep cargo moving. The company

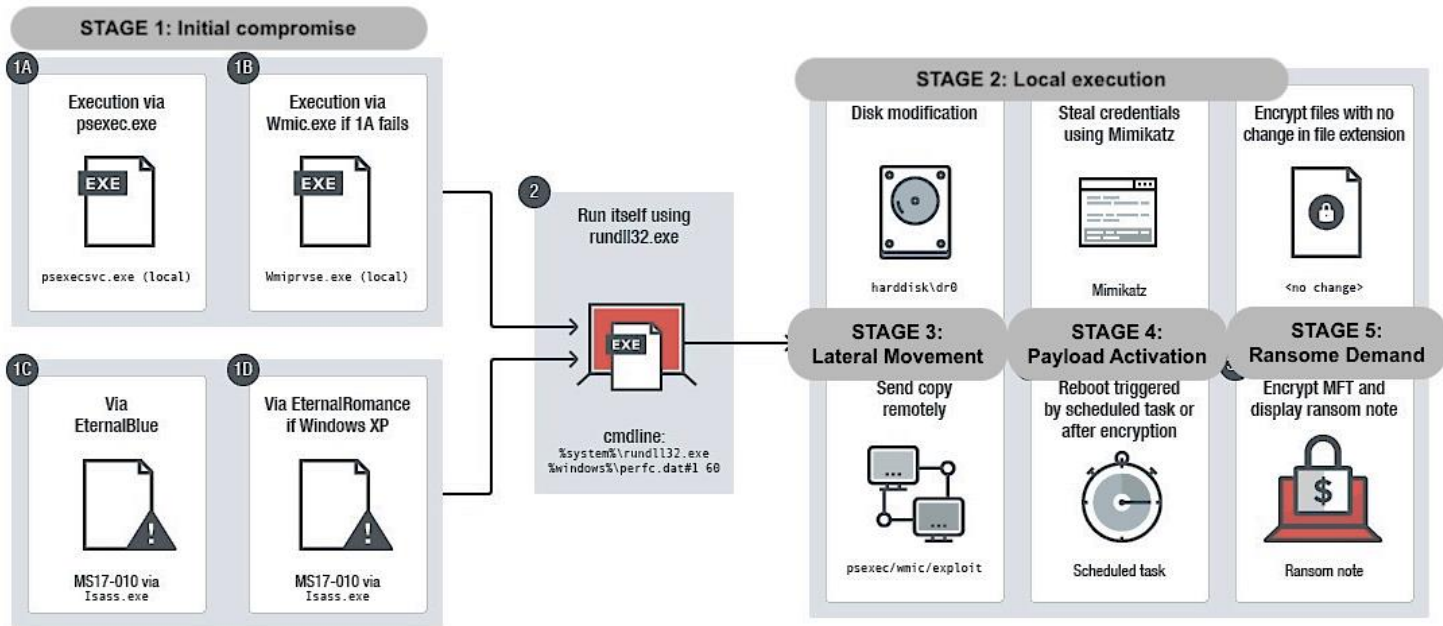
ultimately estimated approximately \$300 million in direct losses <sup>[9]</sup>. Merck and FedEx/TNT reported similarly severe financial setbacks, each facing nine-figure losses. Several hospitals in the United States and clinics in Ukraine were also affected, with some forced to cancel surgeries or reroute patients due to IT system failures <sup>[10]</sup>. Collectively, damage estimates from the White House and industry analysts approached \$10 billion, an unprecedented figure for a cyberattack. NotPetya demonstrated how interconnected supply chains and flat corporate networks could transform a localized incident into a worldwide crisis within hours.

Beyond the financial repercussions, NotPetya had significant strategic and geopolitical consequences. Western governments swiftly attributed the attack to Russia's GRU (military intelligence) <sup>[11]</sup>, classifying it as an act of cyberwar against Ukraine. This attribution prompted renewed debate regarding the scope of cyber insurance, particularly the applicability of "act of war" exclusions. Some insurers initially refused to cover NotPetya-related damages under these clauses, resulting in extended legal disputes, such as Merck's \$1.4 billion settlement in 2022. The attack also catalysed urgent improvements in cyber defence strategies: organizations accelerated patching of known vulnerabilities (NotPetya exploited the same SMBv1 flaw as the earlier WannaCry attack) and implemented network segmentation and stricter privilege controls to prevent enterprise-wide compromise from a single infected node. Government agencies issued alerts emphasizing supply-chain threats and encouraged heightened vigilance, acknowledging the risk of indiscriminate damage from state-sponsored malware <sup>[12]</sup>.

In conclusion, NotPetya served as a profound warning to both public and private sectors. The incident illustrated the cascading effects of targeting a single vulnerable supplier and how malware can exploit default trust relationships to propagate globally. The campaign's global impact has since become a seminal case study in cyber resilience: organizations with robust offline backups and effective incident response recovered within days, while others suffered prolonged outages. NotPetya's scale and destructiveness have cemented its historical significance and continue to influence organizational threat modelling and defensive preparations against state-sponsored cyber threats.

### III Attack Chain & Timeline

NotPetya's attack chain on June 27, 2017, demonstrates a multifaceted and sophisticated approach to network compromise, characterized by both technical acumen and strategic planning. We will map the attack details to the STRIDE threat categories and relevant MITRE ATT&CK techniques.



**Figure 1.** Sequence of events and chronological timeline of the NotPetya's progression <sup>[13]</sup>.

#### Stage 1: Initial Compromise

The threat adversaries, identified as the TeleBots/Sandworm group, executed a supply chain attack by embedding a backdoor within M.E.Doc's legitimate update mechanism <sup>[3]</sup>. When M.E.Doc distributed its scheduled update at approximately 10:30 AM UTC, unsuspecting customers installed a trojanized package. This enabled the NotPetya malware to gain an initial foothold (ATT&CK *Supply Chain Compromise - T1195*) within target organizations under the guise of trusted software. These are acts of **Tampering** of trusted code, and **Spoofing** authenticity of vendor. The malware immediately assessed the system environment, notably terminating itself if it detected a Ukrainian-only keyboard, likely an attempt to avoid impacting Russian systems. It also identified active security products, such as those from Kaspersky and Symantec <sup>[14]</sup>. By leveraging the trust inherent in the software update process, the attackers bypassed traditional perimeter defences and achieved privileged code execution (**Elevation of privilege**). The rapid onset of infections across multiple Ukrainian organizations attests to the efficacy and speed of the initial compromise.

## Stage 2: Local Execution and Reconnaissance

Upon infection, NotPetya initiated reconnaissance activities and credential harvesting on “patient zero” machines. Utilizing a bundled variant of Mimikatz (ATT&CK *Credential Access: OS Credential Dumping – T1003.001*)<sup>[15]</sup>. The malware extracted credentials from Windows systems, prioritizing administrative accounts (**Information Disclosure**). This credential theft was critical for subsequent lateral movement. Simultaneously, NotPetya mapped the local network, collecting information on neighbouring systems via DHCP and ARP data)<sup>[16]</sup>. It also prepared embedded utilities for propagation, such as a renamed copy of PS Exec and a lightweight WMI script (ATT&CK *Masquerading – T1036*). Notably, the malware’s propagation did not require external command-and-control infrastructure, as its spread was largely autonomous within compromised networks.

## Stage 3: Lateral Movement and Propagation

Roughly one to two hours into the outbreak, NotPetya began to propagate laterally. It employed a dual strategy:

### 1. Exploit-based Propagation (ATT&CK *Exploitation of Remote Services – T1210*)

Leveraging the EternalBlue and EternalRomance exploits (previously attributed to the NSA), NotPetya targeted unpatched Windows systems via SMBv1)<sup>[17]</sup>, enabling remote code execution (specifically exploiting CVE-2017-0144/0145). This was a network **Elevation of Privilege (EoP)** attack and **tampering** of remote machines through injection of code at “SYSTEM” level. While this mechanism was limited to local network ranges (unlike WannaCry, which scanned the internet at large), it was particularly effective within interconnected multinational organizations like trusted networks links to Ukraine.

### 2. Credential-based Propagation (ATT&CK *Lateral Movement via Valid Accounts – T1078*; ATT&CK *Remote Services: SMB/Windows Admin Shares – T1021.002*; ATT&CK *Windows Management Instrumentation – T1047*; ATT&CK *Indicator Removal on Host – T1070.001*)

NotPetya demonstrated a notably advanced approach to credential-based lateral movement, capitalizing on stolen administrative credentials (primarily harvested via Mimikatz) to propagate even within well-patched enterprise environments. Rather than relying solely on known vulnerabilities, it leveraged legitimate Windows administrative tools such as PS Exec and WMI. By copying its payload to ADMIN\$ shares and remotely executing it through either PS Exec or WMI (specifically invoking rundll32 on the dropped DLL), the malware was able to impersonate authorized administrators, effectively bypassing standard authentication mechanisms<sup>[18]</sup>. This represents a clear case of credential spoofing, where valid credentials are weaponized for illicit access.

Subsequent analysis by Microsoft indicated that NotPetya’s identity impersonation technique, rather than SMB exploits, was primarily responsible for its rapid spread across well-maintained networks. As a result, organizations with flat network architectures and widespread reuse of administrative credentials were especially vulnerable. A single endpoint

compromise could easily escalate into domain-wide compromise. Beyond propagation, NotPetya also engaged in comprehensive file discovery, targeting documents, archives, and similar file types for later encryption. To further frustrate incident response and recovery, it systematically cleared Windows event logs (impairing forensic analysis) and deleted shadow copies, thereby preventing straightforward restoration<sup>[19]</sup>. These tactics align with the STRIDE repudiation threat model, as they aim to eliminate evidence of malicious activity.

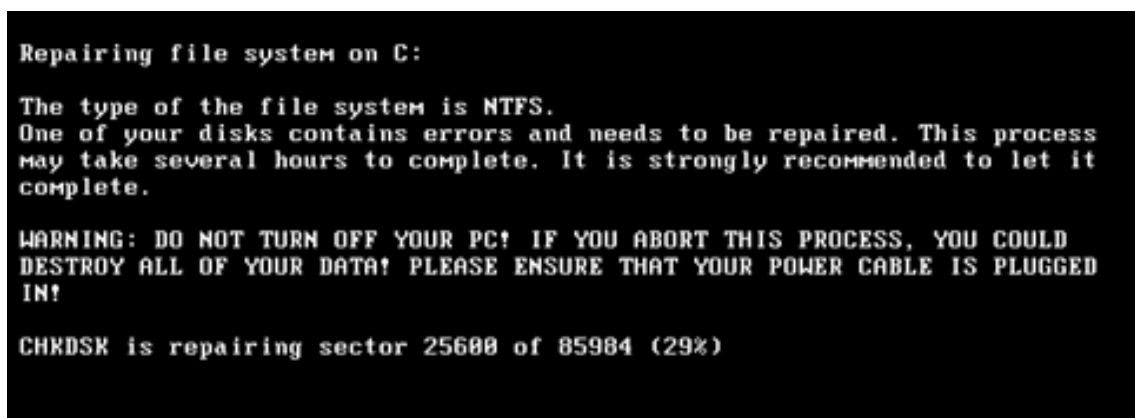
#### ***Note on Timeline:***

*By the afternoon of June 27, 2017 (CET), the malware's impact had extended beyond Ukraine, affecting multinational organizations such as Maersk, which reported widespread system outages. Both security vendors and media outlets initially misidentified the attack as a Petya variant. However, by 14:00 UTC, Kaspersky Lab had confirmed it as a distinct malware strain with over 2,000 confirmed infections. Ukrainian authorities traced the origin to the M.E.Doc software and issued urgent public warnings, which had already penetrated numerous corporate networks on a global scale at the time.*

#### **Stage 4: Payload Activation (Encryption & Reboot)**

Approximately an hour after the initial infection, NotPetya initiated its destructive payload. Each compromised, worm-propagated endpoint scheduled a forced reboot (ATT&CK *System Shutdown/Reboot – T1529*). On modern Windows systems, the malware created a scheduled task to execute “shutdown.exe /r /f” as “SYSTEM” at a randomized near-future time; on older systems, it leveraged the “at” command. This intentional delay allowed NotPetya to maximize lateral movement before alerting users<sup>[20]</sup>.

When the timer expired, numerous systems across affected organizations rebooted nearly simultaneously. At this point, users were effectively denied service, with no opportunity to save their files at this point. During the reboot, NotPetya's custom bootloader which previously written to the MBR/boot sector, took control. The user would see a counterfeit CHKDSK screen “Repairing file system on C:”.



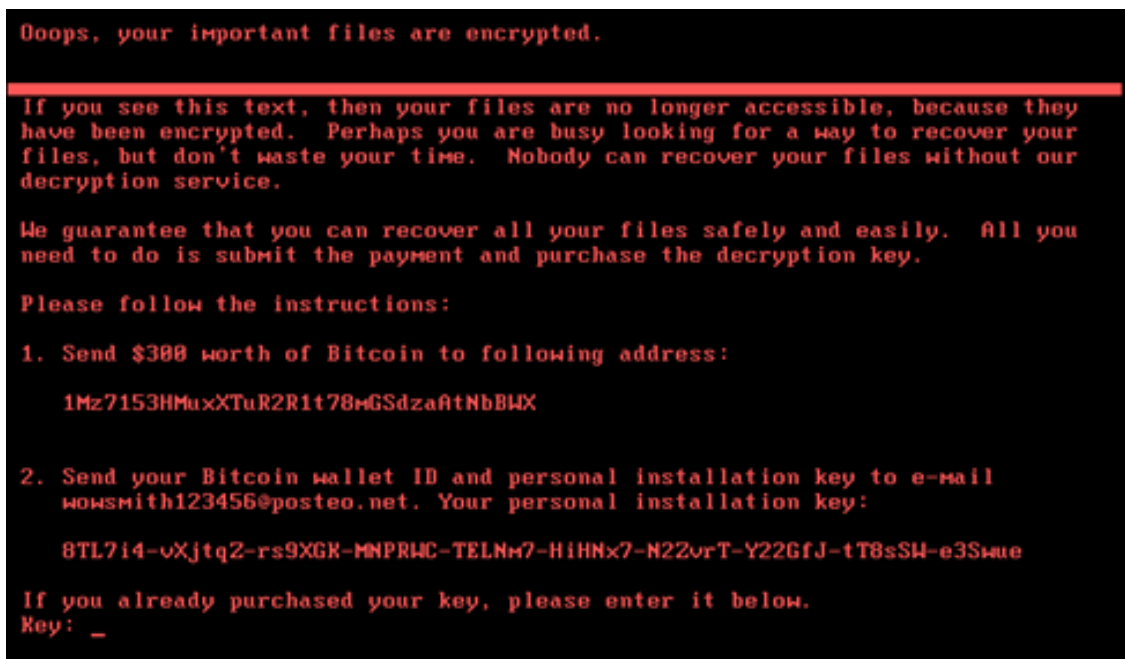
**Figure 2.** Disguised CHKDSK screen<sup>[3]</sup>.

In reality, the malware was encrypting the disk in the background. After several minutes, the system would crash or reboot again, ultimately displaying a ransom note on a black screen. At this stage, the device was rendered completely inoperable: NotPetya had encrypted the NTFS Master File Table (MFT), mimicking earlier Petya variants, and encrypted the first 1MB of each file using AES-128. The malware further encrypted these AES keys with an embedded RSA-2048 public key; however, due to implementation errors, the attackers were unable to derive any decryption key from the data left on disk [6].

From a threat modelling perspective (STRIDE), this constituted a significant *Denial of Service* attack: critical system structures (MBR, MFT, bootloader) and user data (file contents) were irreversibly altered. Additionally, the attack demonstrated *Elevation of Privilege*, as such destructive operations required SYSTEM-level access, which the malware had previously obtained. Based on the MITRE ATT&CK framework, this activation phase corresponds with Impact techniques—*Data Encrypted for Impact (T1486)* and *Inhibit System Recovery*, as NotPetya encrypted both user files and critical disk sectors. While the reboot and malicious bootloader behaviour align with typical Bootkit techniques, in this case, they served destruction rather than persistence.

### Stage 5: Post-Attack (Ransom Demand & Aftermath)

At this stage, each compromised system displayed a standardized ransom note: “Oops, your important files are encrypted... send \$300 in Bitcoin ... and email us your key.” This format closely mirrored typical ransomware demands, directing victims to transmit payment to a singular Bitcoin address and contact an email account (wowsmith123456@posteo.net) with their unique infection ID. Yet, the communication channel was entirely nonfunctional. Within hours of the attack, the email account was disabled by the provider (Posteo), and the use of one Bitcoin wallet for all payments rendered it impossible to identify individual payers.



```
Oops, 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. Your personal installation key:

8TL7i4-vXjtqZ-rs9XGX-MNPRWC-TELNm7-HiHNx7-M2ZvrT-Y22GfJ-tT8sSH-e3Smae

If you already purchased your key, please enter it below.
Key: _
```

Figure 3. Ransomware notices displayed after reboot.



By June 28, cybersecurity professionals had established that paying the ransom was ineffective as there was no available decryption service. This phase demonstrates a clear repudiation tactic: the attackers never intended to provide decryption, erasing any obligation to victims and eradicating forensic evidence through prior log wiping. For affected organizations, this realization marked a shift from attempted negotiation to full-scale disaster recovery. Restoration efforts relied on existing backups or, in some cases, rebuilding entire systems from the ground up – ranging from several days to weeks depending on the organization’s preparedness. A notable example is Maersk, which managed to reconstruct its global network of thousands of machines within ten days, from by a surviving domain controller image from an offline site.

#### *Summary of Timeline (June 27, 2017)*

- **~11:00 UTC:** Initial signs of infection in Ukrainian power companies and banks, tweets on suspected “Petya attacks, as well as reported network issues from Maersk <sup>[6]</sup>.
- **~12:00 UTC:** Malware spreads across networks globally, dubbed as “NotPetya” by Kaspersky noting its use of EternalBlue and credential theft, spreading from M.E.Doc.
- **~14:00 UTC:** Ransomware triggered payload, rebooting screens and demanding ransom. “Vaccine” published, researchers advise powering off PC at the counterfeit CHKDSK stage for data salvaging.
- **~16:00 UTC:** Posteo disabled email account, no decryption occurs as Bitcoin wallet totalled to \$10k. Full swing of incident response.

## **IV STRIDE Threat Model Analysis**

NotPetya serves as a striking example of a sophisticated, multi-layered cyberattack. The campaign began with a supply chain compromise, enabling attackers to gain initial access through spoofing and tampering. Subsequently, they escalated privileges by leveraging known exploits and harvesting credentials. The malware facilitated rapid lateral movement across networks by impersonating legitimate users. Log files were strategically wiped for defence evasion and hinder forensic analysis. Ultimately, the attack culminated in widespread data encryption and destruction, resulting in significant operational impact.

Using Microsoft’s **STRIDE** framework <sup>[26]</sup> (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege), we will reconstruct how each threat was identified in the NotPetya campaign. For each category, *Tables 1–6* outline the specific vulnerabilities, the attackers’ methods of exploitation, impact, and potential mitigation strategies. Where relevant, these are mapped to established NIST CSF functions and MITRE ATT&CK techniques.



**Figure 4.** STRIDE Threat Model <sup>[25]</sup>.

## 1. Spoofing (Identity Misuse)

The act of impersonating sources or identities is known as spoofing.

Spoofing Aspect	Vulnerability/ Issue	Attack Method (NotPetya)	Impact on Victim	Mitigations (NIST CSF & Controls)
Software update trust	M.E.Doc update was a trusted server with no code signing enforcement.	Supply chain compromise emerged as M.E.Doc software update is spoofed, where malicious malware ran under trusted credentials.	Fake CHKDSK screen updates allowed victims to execute malware by assuming it was the legitimate trusted channel.	<b>Protect – Data Security, Protective Technology; Identify – Supply Chain Risk Management</b> (PR.DS-6, PR.PT-1, ID.SC-4): Employ code signing, verify integrity of software updates, and securely vet third-party software.
Authentication – lateral movement	Single-factor authentications / logins are used in Windows within Lan, using NTLM/LM hashes or stolen tokens.	Admin Credentials was stolen and impersonated as valid users, authenticating as domain admins on other machines.	Identity of legitimate admins was spoofed to gain unauthorized access to many systems, enabling remote code execution.	<b>Protect – Access Control</b> (PR.AC-1, PR.AC-5): Enforce strong authentication (e.g., MFA), limit credential exposure (e.g., LSASS protection), and apply network segmentation with tiered admin models to reduce domain-wide access through single credentials (Zero trust approach).

User interface “deception”	User developed trust in system notifications.	Data encryption was masked by a spoofed CHKDSK screen during reboot, so users would not disrupt the process. Ransomware note further spoofed data recovery possibilities.	Users generally assume legitimacy of disk check and waited for encryption to complete. As an effort for data encryption, some victims gained further monetary loss to the fake ransom.	<b>Detect/Respond – Anomalies &amp; Awareness (DE.CM-3, RS.AN-1, PR.AT-1):</b> Monitor for suspicious reboot messages, warn users not to trust fake ransom demands, and run incident response drills to handle attacks like UI spoofing and turn off devices at fake CHKDSK screen.
----------------------------	---	---	--	---

**Table 1.** STRIDE Spoofing.

## 2. Tampering (Data/Integrity Breach)

Unauthorized modification of data or code is known as tampering.

<b>Tampering Aspect</b>	<b>Vulnerability/ Issue</b>	<b>Attack Method (NotPetya)</b>	<b>Impact on Victim</b>	<b>Mitigations (NIST CSF &amp; Controls)</b>
Software code integrity	M.E.Doc update mechanism allowed modification of application code on server.	The M.E.Doc software code was tampered on the vendor’s servers through inserting a PHP backdoor. This resulted in organizations installing malware during routine software updates.	Significantly breached software integrity and trust in official software updates, highlighting vulnerabilities in supply chain processes and undermines public confidence in standard security practices	<b>Protect – Data Security, Supply Chain (PR.DS-6, ID.SC-4):</b> Implement cryptographic hash verification on all software updates. Require digitally signed update packages to confirm authenticity. conduct regular supply chain security reviews and perform penetration testing on vendor infrastructure. Deploy sandboxing solutions to validate update behaviour prior to full deployment and utilize endpoint detection and response (EDR) systems for early threat identification.
System boot records	Low security protection of Master Boot Record (MBR) and low-level disk sectors.	Direct manipulate Master Boot Record (MBR) and partition boot sectors, effectively replacing legitimate code with its own. Altered the NTFS file system by encrypting the Master File Table (MFT). This level of disk tampering occurred with	Affected systems became unbootable; the original bootloader and disk structure were irreparably damaged. Even removal of the malware did not restore functionality, as the underlying corruption persisted and rendered the machine unusable.	<b>Protect – Protective Technology (PR.PT-5):</b> Implement UEFI Secure Boot and robust firmware protection to prevent unauthorized modifications to bootloaders. Modern Endpoint Detection and Response (EDR) tools can monitor and block suspicious attempts to write to the MBR or boot sectors. Network segmentation of critical infrastructure serves to limit the spread of malware. While

		administrative privileges, bypassing existing safeguards.		continuous integrity scanning of critical disk areas is challenging in practice, regular checks can help detect unauthorized changes.
Data files and documents	Unrestricted write access to bulk files under user context or with admin privileges	Encrypted the first megabyte of user files using AES-128, corrupting the content in place while leaving filenames and timestamps untouched. NotPetya also deleted Volume Shadow Copies, effectively erasing common backup options and eliminating straightforward recovery paths.	Entire collections of documents, databases, and archives on both local machines and network shares became irretrievable. The absence of visible changes to file names or metadata further complicated identification and recovery efforts, as affected files appeared normal until accessed.	<p><b>Protect – Data Security</b> (PR.DS-1), Info Protection (PR.IP-4): Enforce strict file permissions where users and applications should only possess write access to critical shares when necessary. Application whitelisting can further reduce exposure by restricting file modifications to trusted processes.</p> <p><b>Detect – Security Monitoring</b> (DE.CM-1): Observe unusual activity, such as large numbers of files being rapidly modified or encrypted. Organizations can detect ransomware behaviours in progress and intervene promptly. Maintain backups that are either offline or write-protected is crucial, as this prevents malware from tampering with recovery data.</p>

**Table 2. STRIDE Tampering.**

### 3. Repudiation (Anti-Forensics and Hiding Activity)

Threats of repudiation include acts that are impossible to track down or validate, which enables an attacker to avoid accountability.

Repudiation Aspect	Vulnerability/ Issue	Attack Method (NotPetya)	Impact on Victim (Forensics)	Mitigations (NIST CSF & Controls)
Logging and audit trails	Event logs and audit files are not securely protected, where local admins can manually clear logs.	To clear Windows Event Logs (System, Security, etc.), NotPetya ran wevtutil. Additionally, the NTFS USN change journal, which tracks file changes was erased. These steps were taken right before the system reboot.	Because the timeline of events on infected hosts was erased, investigators lost important hints that made it challenging to piece together what the malware did or which account it used. The attackers successfully hid their tracks, which initially hampered efforts at attribution and incident response.	<b>Protect/Detect – Protective Technology &amp; Audit Events</b> (PR.PT-1, DE.AE-3): Use centralised logging. Real-time log forwarding to a distant server ensures that local deletion does not wipe out all records. Although malware operating as SYSTEM can clear logs, use OS settings to limit who can do so. Instead, concentrate on using SIEM correlation to identify the clearing event. monitoring the integrity of important system files and journals.

Transaction non-repudiation	Users' claims are not externally verified.	The attackers offered a single email address for "support" before allowing it to be disabled; by making themselves unreachable, they rejected any responsibility to decrypt. Furthermore, the ransom note's "installation key" was arbitrary and did not match a legitimate decryption key	At first, victims might have thought the data could be restored, but they were unable to confirm or deny the attacker's intention to assist. Many systems had already been destroyed by the time the attackers were discovered to have disappeared. In essence, victims had no recourse because the attackers denied any obligation to help.	<b>Respond – Analysis &amp; Mitigation</b> (RS.AN-5, RS.MI-3): To avoid wasting resources trying to get in touch, quickly notify stakeholders of the situation (for example, through ISACs or CERT alerts) that this is a wiper and not ransomware. Once non-repudiation is apparent, law enforcement and incident responders should handle such situations as destructive attacks right away. There aren't many technical controls in place here; policy and response protocols are more important.
-----------------------------	--	--	--	--

**Table 3.** STRIDE Repudiation.

#### 4. Information Disclosure (Privacy/Sensitive Data Loss)

Since NotPetya was not a spy tool, but it did steal credentials internally, information disclosure in this context refers to the attacker obtaining access to information they shouldn't have.

Info Disclosure Aspect	Vulnerability/ Issue	Attack Method (NotPetya)	Impact on Victim	Mitigations (NIST CSF & Controls)
In-memory credential storage	For logged-in users, the Windows LSASS process saves reversible hashes or plaintext passwords (especially for older versions or if credentials are cached)	Mimikatz, a credential dumper, was incorporated into NotPetya to retrieve passwords from memory. In LSASS, all domain administrator and local administrator credentials were recorded. It was used right away for lateral spread; no network exfiltration was required	Significant internal data leak: malware was able to obtain administrator passwords, service account login information, and potentially password hashes. Internal security was compromised because, once secrets were taken, there was nothing that could stop the malware from using its "insider" access to other systems.	<b>Protect – Access Control, Data Security</b> (PR.AC-3, PR.DS-2): Turn on memory protection for LSASS (for example, Windows 10+'s Credential Guard to stop LSASS memory from being read). Limit the number of computers that administrators can access (tiered admin model) to prevent compromised hosts from housing high-privilege credentials. Update your systems because LSASS dumping is more difficult after Windows updates in 2017. Detection: keep an eye out for unusual memory access or Mimikatz signatures (EDR tools can detect known Mimikatz behaviour).
Network share data exposure	Extensive access to file shares and private information in plaintext,	NotPetya had read/write access to any files that the compromised user accounts could access	There was a chance that data confidentiality would be compromised (a variant could have stolen data).	<b>Protect – Access Control</b> (PR.AC-4, PR.AC-6): Users and service accounts should only be able to see the files they absolutely need when file shares are set up with least

		(which it then encrypted), even though it did not exfiltrate data. Although the malware's primary objective was destruction, it could have theoretically uploaded or revealed private files. However, it was able to locate and encrypt files across the entire organization just by virtue of the widespread read access.	Information disclosure in this instance took the form of the previously mentioned loss of credential confidentiality, which could be interpreted as any subsequent dumping of file content into memory for encryption (but not external sharing). Confidentiality was not the primary effect, it was integrity and availability.	privilege. File server network segmentation and encryption-at-rest can help prevent malware from reading raw data, though it can still read if it has access to the running system. audits of permissions on a regular basis to minimize open shares. As per the ransomware detection above, detect unusual file access (many files read or encrypted).
--	--	--	--	---

**Table 4.** STRIDE Information Disclosure.

## 5. Denial of Service (Service Disruption)

Denial of Service here refers to making systems or data unavailable to rightful users.

DoS Aspect	Vulnerability/ Issue	Attack Method (NotPetya)	Impact	Mitigations (NIST CSF & Controls)
Network propagation (worm)	No internal segmentation or SMB traffic throttling, and a flat network architecture.	By flooding networks with infection traffic and causing numerous systems to crash almost at once, NotPetya's worm component successfully started an internal denial of service attack on networks. In the majority of corporate LANs, there was no rate limitation on authentication or SMB exploit attempts.	In addition to individual host failure, the scanning/propagation load caused some networks to experience outages or device crashes, overwhelming network bandwidth and system resources. Malware propagation caused a coordinated denial of service that disrupted business operations across the entire organization.	<b>Protect – Access Control &amp; Protective Technology</b> (PR.AC-5, PR.PT-4): Even within the LAN, use firewalls and network segmentation (e.g., restrict SMB communications between workstations). Make use of intrusion prevention systems (IPS) that are able to identify and stop brute-force behaviour or SMB exploit signatures (Detect/Protect).  Detect – Security Monitoring (DE.CM-7): The unusual lateral movement may have been detected by internal network monitoring. After odd access patterns, rate-limit or terminate accounts (though difficult for machine-driven attacks).
Data and system destruction	Absence of redundant systems and offline	Data was rendered inaccessible due to a denial of service at the data level brought	Complete business operations (such as manufacturing lines and ports) stopped; this	<b>Recover – Backup &amp; Response Planning</b> (RC.BC-1, RC.IM-1, PR.DS-11): Make sure critical systems are regularly backed up

	backups of critical data.	on by NotPetya's encryption of the disk and files. It also prevented the use of the computing services themselves by bringing down entire operating systems (erasing boot records). All the victims' main systems went down at once.	was essentially a complete denial of service until it was recovered. Without backups, the attack's wiper feature resulted in the permanent loss of data (a worst-case DoS). The service outage was prolonged because restoration took time, even with backups.	offline and that they are operational so that data can be restored even if production is lost.  <b>Respond – Mitigation (RS.MI-1):</b> Have an incident response strategy that includes system rebuilds and prompt isolation of compromised computers to prevent further spread.  <b>Recover – Recovery Planning (RC.RP-1):</b> Business continuity plans to function manually or in a reduced capacity during IT outages. For quicker restores, accounting for immutable storage or snapshot technologies that are impervious to malware.
Service resilience	IT single points of failure, such as a single domain controller or DHCP server for the whole network.	NotPetya took advantage of the fact that many organizations had central services that, once disabled (by encryption), resulted in a cascade of failures (for example, when Maersk's single AD forest was taken down, authentication was lost everywhere). Because the subsidiaries' networks were not isolated, a failure in one node had an impact on the others.	Fragile infrastructure increased the impact of the attack, turning it from a localized denial of service to a company-wide one. For instance, users worldwide would not be able to log in or route traffic if all authentication or network configuration was dependent on a small number of compromised servers.	<b>Recover – Backup &amp; Response Planning (RC.BC-1, RC.IM-1, PR.DS-11):</b> Determine which services are essential and implement redundancy or isolated backups (e.g., keep an offline read-only domain controller or cloud backup AD). Divide networks according to business units or regions to limit the blast radius and prevent an incident in one from immediately affecting all.  <b>Recover – Improvements (RC.IM-1):</b> To make sure services can be promptly restored in the event of a disruption, test incident response on a regular basis using disaster recovery exercises.

**Table 5.** STRIDE Denial of Service.

## 6. Elevation of Privilege (Privilege Escalation)

EoP entails obtaining more access rights than planned.

EoP Aspect	Vulnerability/ Issue	Attack Method (NotPetya)	Impact	Mitigations (NIST CSF & Controls)
Unpatched OS vulnerability (remote EoP)	A bug in Windows SMBv1 (MS17-	Using EternalBlue/ EternalRomance, NotPetya was able to	Total takeover of computers lacking important patches,	<b>Protect – Information Protection &amp; Protective Technology (PR.IP-12, PR.PT-</b>

	010) permitted remote code execution as SYSTEM.	quickly obtain SYSTEM privileges on unpatched computers via the network. Any machine that was vulnerable was owned at the highest privilege level; no user interaction took place.	allowing malware to operate with kernel/system access and accomplish anything on those systems. A single unpatched laptop on a network could compromise the entire domain.	3): Implement important security updates (such as MS17-010) as soon as possible. To safeguard legacy systems that cannot be patched, use network segmentation (isolate or limit SMB). To eliminate the vulnerable service, completely disable SMBv1 (as recommended by Microsoft). To stop known exploit traffic, use virtual patching or an intrusion prevention system.
Default admin privileges needed for software	Flat Active Directory makes it simple to become the domain administrator if one machine is admin-compromised.	M.E.Doc probably used whatever privileges he had to execute the first NotPetya drop. Accounting employees frequently have local administrator rights, or the update process runs as admin/system. Additionally, NotPetya tried to use schtasks, which needed administrator privileges; if it didn't, the more harmful features might not work. However, it frequently elevated itself to SYSTEM from a regular user by using exploits.	The malware may not have been able to install drivers or alter MBR if all computers had been used in least-privilege mode. However, it could carry out all destructive actions because it successfully installed SYSTEM on the majority of machines (either through an exploit or starting point). The attack was made much easier by the widespread admin rights in corporate settings (Privilege escalation in design).	<p><b>Protect – Access Control</b> (PR.AC-6): Make sure user accounts have the least amount of privilege possible; users and programs such as M.E.Doc shouldn't be running as local administrators by default. If at all possible, use application sandboxing.</p> <p><b>Detect – Anomalous Behaviour</b> (DE.CM-4): This feature should raise alarms when it detects unusual privilege use, such as when an office computer unexpectedly creates a scheduled task as SYSTEM or injects into system processes. protection for endpoints that can detect attempts at privilege escalation (such as an unexpected process spawning with SYSTEM rights).</p>
Lateral movement privilege escalation	Flat Active Directory – if one machine is admin-compromised, easy to become Domain Admin	By obtaining domain credentials from a single machine, NotPetya was able to transform a single local compromise into domain admin. In essence, it capitalized on pre-existing admin sessions and AD trust. It didn't require an exploit to escalate on target machines once it obtained a privileged token.	Domain-wide Elevation of Privilege: The malware breached all internal security boundaries by operating with the highest privileges possible everywhere. It could encrypt even protected files, disable services, and spread itself via admin\$ shares.	<p><b>Protect – Access Control</b> (PR.AC-5): Make sure user accounts have the least amount of privilege possible; users and programs such as M.E.Doc shouldn't be running as local administrators by default. If possible, use application sandboxing.</p> <p><b>Detect – Security Monitoring</b> (DE.CM-7): This feature should raise alarms when it detects unusual privilege use, such as when an office computer unexpectedly creates a scheduled task as SYSTEM or injects into</p>



				system processes. protection for endpoints that can detect attempts at privilege escalation (such as an unexpected process spawning with SYSTEM rights).
--	--	--	--	--

**Table 6.** STRIDE Elevation of Privilege.

Overall, the STRIDE analysis shows how NotPetya touched *every* category: it **spoofed trust**, **tampered with software and data**, **covered its tracks**, **disclosed internal secrets**, **denied service at scale**, and **escalated privileges** unchecked. We have found mitigations in every category. As shown in parenthesis for each mitigation, many of these are in line with standard frameworks such as the NIST Cybersecurity Framework (CSF) and controls. Recurring themes include, for example, network segmentation (PR.AC), backup techniques (PR.DS, RC), and maintaining proper patch management (CSF PR.IP).

**Table 7.** STRIDE Threat Matrix vs. MITRE/NIST

Kill Chain Stage	STRIDE	MITRE ATT&CK (ID)	NIST CSF Control
<b>Supply Chain Compromise</b>	Tampering, Spoofing	Initial Access: T1195	PR.DS, ID.SC
<b>Reconnaissance</b>	Info Disclosure	Credential Dumping: T1003	PR.DS-2
<b>Lateral Movement</b>	Spoofing	Valid Accounts: T1078	PR.AC-6
<b>Exploit Propagation</b>	EoP	Exploitation: T1210	PR.IP-12
<b>Payload Activation</b>	DoS, Tampering	Data Enc. for Impact: T1486	PR.DS-1, RC.BAK
<b>Log Wiping</b>	Repudiation	Indicator Removal: T1070	DE.AE-3, PR.PT-1
<b>Ransom Note Communication</b>	Repudiation	N/A (static email)	RS.AN-5

## V Encryption & Payload Analysis

NotPetya stands out as a particularly destructive and technically unconventional malware, exhibiting characteristics that sharply distinguish it from standard ransomware operations. Its encryption payload was engineered less for financial gain and more for widespread damage, a fact evident from multiple facets of its design and deployment.

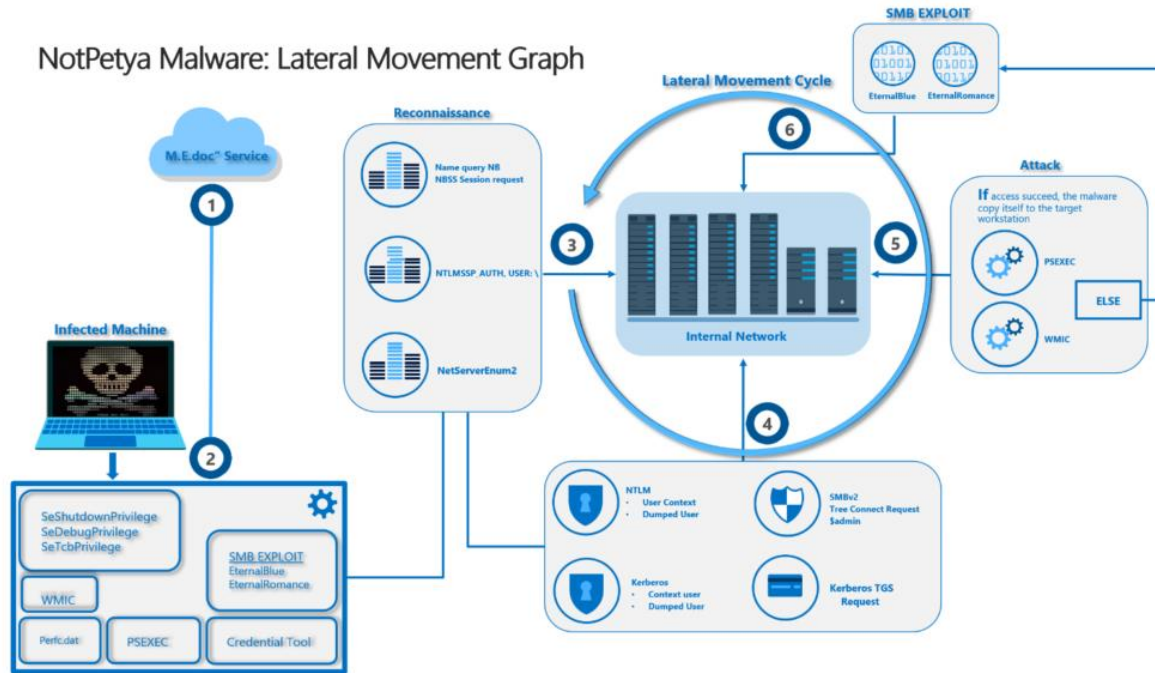


Figure 5. Malware Propagation. [23]

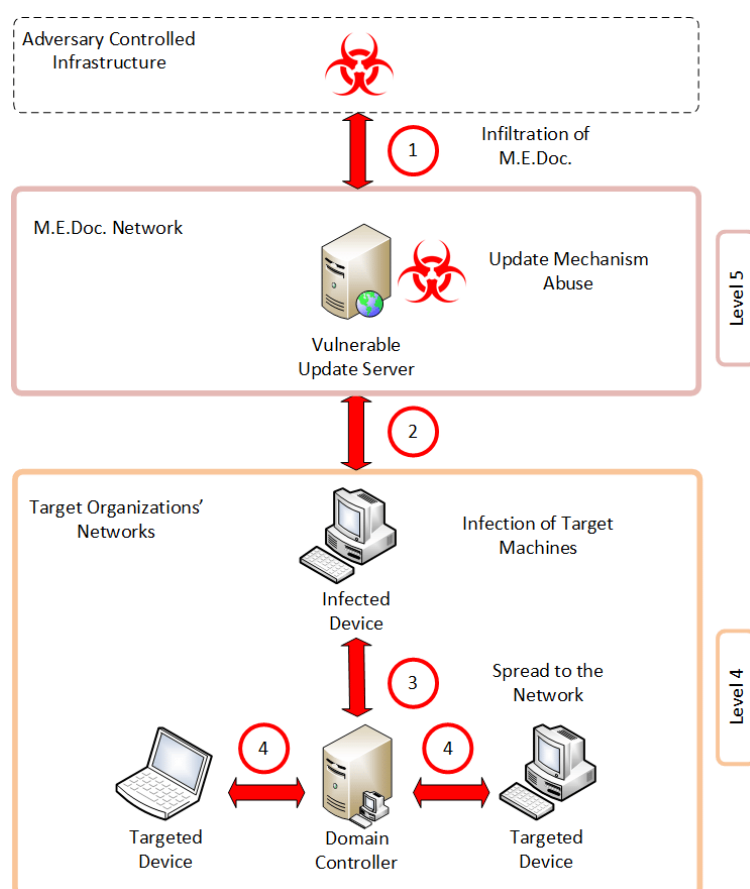
Regarding cryptographic mechanisms, NotPetya employed a hybrid approach integrating both symmetric (AES-128) and asymmetric (RSA-2048) encryption [21]. For every compromised host, the malware generated a unique AES key in volatile memory, subsequently utilized to encrypt files on the system as well as during disk-level encryption routines. This AES key was then encrypted using a hardcoded RSA public key embedded within the malware's binary. Unlike typical ransomware, where attackers retain a corresponding private key to facilitate decryption upon ransom payment, NotPetya's implementation was fatally flawed by design: the "victim ID" displayed to users was cryptographically decoupled from the actual AES key. As a result, there was no mechanism by which the attackers or anyone else could reconstruct the decryption key from publicly available information, rendering file recovery impossible even in cases where ransom demands were met.

**File encryption behaviour:** In terms of file encryption, NotPetya's user-mode payload targeted approximately sixty common document, image, and archive file types, a targeting strategy reminiscent of other wiper malware like KillDisk. For each file, the first 1,024,000 bytes (1 MB) were encrypted using AES-128 in CBC mode, with no additional metadata, extension changes, or file markers inserted. Files smaller than this threshold were fully encrypted; files exceeding 1 MB had only their headers and initial portions altered, effectively corrupting all but the most resilient file formats. The

absence of padding and explicit headers further complicated any recovery attempts, as it became difficult to even verify if a file had been successfully decrypted <sup>[3]</sup>. Moreover, the lack of filename or extension changes hindered systematic identification or triage of affected files by system administrators.

**Disk encryption (MBR/MFT encryption):** NotPetya’s disk-level encryption further underscored its destructive intent. When executed with administrative privileges, the malware overwrote the system’s Master Boot Record (MBR) with a malicious bootloader, which masqueraded as a CHKDSK utility upon reboot and subsequently encrypted the Master File Table (MFT) using a modified variant of the Salsa20 cipher (distinct from the original Petya ransomware implementation). Crucially, the “installation key” presented to victims after encryption was random and, in many cases, contained characters outside the valid input set, making it impossible to submit a valid decryption key even if one had existed.

**Ransom notes details:** The ransom note, displayed after reboot in a format mimicking Petya’s original message, demanded \$300 in Bitcoin sent to a single wallet address. This is a further anomaly, as conventional ransomware operations generate unique addresses per victim for payment tracking. Victims were instructed to contact the attackers via a single Posteo email address, which was promptly deactivated, severing any potential communication. Despite over forty payments being made to the provided address <sup>[22]</sup>, no decryptions were ever reported, and the technical design made recovery infeasible.



#### Anti-analysis features:

NotPetya implemented several sophisticated anti-analysis features designed to frustrate malware researchers and evade detection in controlled environments. Among these, the malware specifically checked for the presence of certain antivirus products, such as Kaspersky or Symantec. Notably, if Kaspersky was detected, NotPetya would modify its action. For example, researchers at LogRhythm observed that the SMB exploit payload was only deployed if Kaspersky was present, suggesting a deliberate tactic to circumvent analysis in lab settings or to selectively activate credential theft functionalities.

**Figure 6.** NotPetya Attack process <sup>[24]</sup>.

Additionally, NotPetya was programmed to self-terminate if it detected that the system's language was set to Russian or Ukrainian. There is some debate regarding the exact mechanism. Some reports assert that the malware checks for the "RU" language code, while others claim it inspects the keyboard layout for "US" as a condition, but the intent to avoid domestic targets is evident. This behaviour aligns with patterns seen in other malware originating from Russian-speaking regions. A further point of interest is the so-called "vaccine" file. If the file C:\Windows\perfc.dat existed on the machine, NotPetya would refrain from initiating its encryption routine. This feature, believed to be a remnant of a debugging process repurposed as a kill switch, allowed administrators to "immunize" systems by creating this file. Such a mechanism is atypical for ransomware and supports the interpretation of NotPetya as a wiper, rather than a profit-driven extortion tool, since the attackers appeared unconcerned if some systems were rendered immune by this check.

In summary, the design of NotPetya's encryption and payload underscores its nature as a destructive, one-way attack. While it borrowed techniques from ransomware, such as locking files and demanding payment, it offered no viable path to data recovery, reinforcing its characterization as a wiper. For malware analysts, NotPetya served as a cautionary example: not all ransomware campaigns are motivated by financial gain, and ransom notes can serve as intentional misdirection. Furthermore, the case highlights the critical importance of robust cryptographic implementation, though in this instance, the attackers' so-called "mistakes" were intentional.

## VI Ransom Note & Communication Analysis

Regarding its ransom demand and communication strategy, NotPetya mimicked the conventions of typical ransomware campaigns. Upon reboot, after data encryption was complete, victims were presented with a classic ransom note.

**Ransom message content:** The message, nearly identical to that used in prior Petya variants, read: "Oops, your important files are encrypted. If you see this text, then your files are no longer accessible, because they have been encrypted... 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. ... Send \$300 worth of Bitcoin to the following address: 1Mz7153HMux... then send your Bitcoin wallet ID and personal installation key to e-mail \*wowsmith123456@posteo.net.\* Your personal installation key: <unique 60-character hex string>."

While the note claimed "we guarantee recovery," this assurance was false. The so-called "personal installation key" was not actually linked to the encrypted data, rendering decryption impossible even for the attackers.

**Communication method and attacker responsiveness:** Communication was restricted to a single email address hosted by Posteo, a German provider. This approach diverged from the practices of contemporary ransomware operators, who typically employ dedicated portals or individualized communication channels for victims. The use of a single public email account made it trivial for law enforcement and the provider to intervene. Posteo disabled the account and issued a public statement on June 27, shortly after the outbreak began. As a result, any victim who paid the ransom after this action had no means of contacting the attackers or recovering their data. Ethically, Posteo's decision was justified, and in the context of NotPetya, it did not affect victims' chances of recovery, as decryption was technically unfeasible. In fact, this action arguably prevented additional victims from paying a futile ransom. A small number of security researchers, who managed to send test messages before the shutdown, also received no response. On the financial side, the associated Bitcoin wallet displayed incoming payments, but there were no withdrawals until over a month later. At which point, all funds were extracted in a single transaction and routed through mixing services, very likely by the original attackers or someone with access to the cryptographic keys.

**Comparison to typical ransomware communications:** This lack of engagement stands in stark contrast to established ransomware operations. Typically, ransomware actors maintain some level of "customer service," responding to inquiries, providing evidence of decryption capabilities, and negotiating payment terms. In NotPetya's case, such interaction was conspicuously absent. The communication channels—a nonfunctional email and a static wallet, functioned more as decoys than genuine points of contact. This contributed to the widely held conclusion, supported by technical analysis, that NotPetya was primarily a destructive tool rather than a true profit-driven ransomware campaign. The inclusion of ransom demands appeared designed to introduce confusion and psychological pressure, perhaps to delay recovery efforts or encourage victims to take actions (like reconnecting compromised systems) that might worsen the incident.

**Network communications:** Regarding network communications, NotPetya differed from many malware strains in that it did not connect to external command-and-control infrastructure. Its network activity was limited to propagation, leveraging SMB exploits, WMI calls, and file transfers over SMB to move laterally within networks. There was no evidence of outbound traffic for key retrieval or data exfiltration. This "fire-and-forget" approach meant defenders could not rely on typical indicators like malicious domains or IP addresses. Instead, detection depended on identifying exploit activity or abnormal lateral movement.

In summary, NotPetya's ransom note and supposed communication channels served primarily as elements of deception. The incident demonstrates the dangers of treating all ransom demands at face value. In this scenario, attempts to pay or negotiate would have been futile; the appropriate response was rapid containment and restoration from backups. The event also underscores the importance of timely information sharing between public and private sectors. Early alerts from agencies and independent researchers clarified the futility of payment and enabled organizations to make informed decisions—minimizing wasted effort and accelerating recovery.

## VII Security Engineering Reflections

Reflecting on NotPetya, it's hard to overstate how fundamentally it challenged established notions in cybersecurity engineering.

**The weakest link – Supply Chain Security:** The incident showcased, in dramatic fashion, the vulnerabilities inherent in supply chain security. Even robust perimeter defences can be rendered moot if a trusted third-party becomes an attack vector. This underscores the critical need for rigorous supply chain risk management. Practices such as verifying software signatures, sandboxing updates, and vetting vendor security procedures are not optional—they are essential. The move towards Zero Trust architectures and industry adoption of tools like SBOMs and stricter code signing policies is a direct response to these risks. Personally, the lesson is clear: external system inputs should never be assumed trustworthy. This mindset is now foundational to my approach in threat modelling.

**Segmentation and Principle of Least Privilege:** Network segmentation and the principle of least privilege emerged as equally vital. NotPetya exploited flat networks and broadly distributed administrative credentials, allowing the attack to propagate rapidly. The utility of segmenting networks, separating office systems from operational technology cannot be overstated. Similarly, restricting administrative access (with tools such as tiered AD models and multifactor authentication) directly limits the impact of such breaches. While these approaches can conflict with business convenience, their value is now well-recognized, justifying significant investment in identity security. This case reinforced for me the notion that security architecture is paramount. Without internal segmentation and privilege controls, even the best detection tools are insufficient.

**Rapid Patching and Legacy Systems:** Another key takeaway relates to patch management and legacy systems. The persistence of unpatched systems after WannaCry, particularly those vulnerable to MS17-010, enabled NotPetya to wreak havoc. This highlights the critical importance of timely patching and the risks posed by technical debt, outdated protocols and systems become liabilities. The renewed industry focus on rapid patch cycles and decommissioning obsolete technologies is a direct consequence of such incidents. As a professional, I consider this a reminder that operational hygiene, while often overlooked, is foundational to security. The costs of neglecting these basics can be catastrophic.

**Incident Response and Backup Strategies:** NotPetya's aftermath underscored the centrality of incident response planning and backup strategies. Organizations with robust, offline backups and practiced recovery procedures were able to restore operations, while those reliant on networked backups suffered extended outages. The lesson is clear: resilience is not just about prevention but about recovery. Regularly testing disaster recovery plans and maintaining immutable, offline backups are now non-negotiable best practices. This has fundamentally influenced my approach to system design, ensuring recovery and continuity features are embedded from the outset.

**Threat modelling and frameworks in practice:** Threat modelling in practice presents a far more complex landscape than textbooks suggest. While frameworks like STRIDE, MITRE ATT&CK, and NIST CSF are foundational in academic settings, real-world incidents such as NotPetya highlight their practical relevance. By mapping the stages of NotPetya to ATT&CK techniques, it becomes evident how specific defensive measures, such as detecting Credential Dumping (T1003) via tools like Mimikatz could have mitigated the attack's impact. STRIDE demonstrates the multi-dimensional

nature of such threats, revealing that actual incidents often involve overlapping threat categories. This experience underscores that threat modelling is not merely a bureaucratic exercise; rather, it is essential for identifying vulnerabilities and strengthening defences. Questions like, “What if a trusted software update becomes malicious?” or “What if malware leverages administrative credentials?”. Once considered unlikely, are now recognized as crucial considerations due to attacks like NotPetya. This awareness should inform future modelling efforts and encourage inclusion of even low-probability, high-impact scenarios.

**Ethical and Professional considerations:** The NotPetya case also raises significant ethical and professional considerations. The attack, widely attributed to state actors, resulted in substantial collateral damage to both civilian institutions and global businesses, including hospitals. This blurring of boundaries between military and civilian targets intensifies the ethical responsibilities of cybersecurity professionals. It is imperative for practitioners to advocate for international norms, such as prohibitions against targeting critical infrastructure like healthcare facilities. Furthermore, the collaborative response among researchers who shared vaccine information and technical findings publicly, serves as a valuable model for professional conduct. Such cooperation exemplifies the collective responsibility of the security community to respond effectively in crises, emphasizing the importance of responsible information sharing and joint action.

### Teaching value

From an educational standpoint, NotPetya offers a comprehensive case study for teaching core security concepts. It encapsulates issues ranging from buffer overflows and worm propagation to cryptographic weaknesses, incident response, and practical threat modelling. The development of a controlled “safe demo” scenario allows students to observe ransomware behaviour and mitigation strategies in a risk-free environment. This experiential approach enhances engagement and fosters a deeper understanding of the real-world implications of security failures, demonstrating that the consequences extend far beyond mere data loss to potentially disrupting global operations.

In summary, NotPetya illustrates that cybersecurity is inherently collaborative, requiring coordinated preventive measures and cross-organizational response. The incident validated longstanding best practices, such as timely patching, enforcing least privilege, and maintaining backups, by revealing the tangible consequences of neglect. Analysing NotPetya has been formative in shaping a more comprehensive perspective as a security professional, reinforcing the necessity for broad, integrated defences that address technical vulnerabilities, human factors, and organizational processes.

## VIII Demo Script and Instructions

A Python-based demonstration script was developed to facilitate technical analysis and education around NotPetya-like ransomware behaviours in a safe, controlled environment. This demo provides a clear illustration of ransomware processes: it encrypts a sample file using a randomly generated AES-128 key, then securely discards that key, effectively simulating NotPetya's irreversible data destruction. Through this exercise, participants can directly observe how quickly data becomes unrecoverable in the absence of proper backups, reinforcing the importance of proactive data protection.

The demonstration further includes a simulated propagation log, which step-by-step outlines the malware's hypothetical movement through a network. It models key elements of the NotPetya attack chain, such as credential harvesting, use of PS Exec or WMI, and fallback exploitation without performing any real network activity. This logging feature enables users to visualize lateral movement and attack execution in a risk-free setting. The script is designed to operate only on temporary files it generates during runtime, ensuring a secure, sandboxed experience. Additionally, it includes a "recoverable mode," which allows participants to retain the encryption key and restore data. This distinction helps clarify the differences between traditional ransomware (where recovery is sometimes possible) and wiper malware (where recovery is intentionally prevented).

This demonstration tool is especially valuable for instructional and security training contexts. It allows users to witness the transformation of accessible data into unreadable ciphertext, and to experience firsthand the consequences of lost encryption keys. The exercise underscores the necessity of robust backup and security protocols. Furthermore, the demonstration's outputs can be mapped to established security frameworks, such as STRIDE and MITRE ATT&CK, supporting the translation of theoretical security concepts into practical understanding.

**File attached :** demo.py, demo.txt

### Instructions:

1. Install dependencies: `pip install cryptography`
2. Run the script in an empty folder: `python3 demo.py`
3. View encrypted file (first 5 lines): `Cat demo.txt | head -n 5`
4. (Optional) To see decryption: `python3 demo.py --demo-recover`



## IX Conclusion

NotPetya marked a pivotal moment in the landscape of cybersecurity. This was not a typical ransomware incident – it represents a sophisticated, state-sponsored cyberattack that rapidly propagated and disrupted countless organizations worldwide. The subsequent analysis delved into the technical mechanisms underpinning the attack, dissecting how it leveraged both systemic software vulnerabilities in everyday security protocols. By unravelling these elements, the report highlighted the multifaceted nature of modern cyber threats.

The aftermath leaves both private enterprises and government bodies compelled in reevaluating their foundational security practices. Essential measures, such as timely software updates, robust data backups, and stringent access controls serves as critical components of organizational resilience. NotPetya also spiked broader discussions surrounding the realities of cyber warfare and the unpredictability in digital threats. Within affected communities, information sharing and collaborative defence efforts is valuable.

Ultimately, the NotPetya incident imprinted a crucial lesson: It is impossible to absolutely prevent cyberattacks, but through proper mitigation and incident response training, we can significantly reduce their impact. Effective planning and disciplined security habits allows organizations to recover faster from such disruptive events. By systematically studying incidents like NotPetya, the cybersecurity field can evolve and improve its defences against future adversities.

## References

1. **MITRE ATT&CK (2025)**. *NotPetya (S0368)*. Available at: <https://attack.mitre.org/software/S0368/#:~:text=NotPetya%20is%20malware%20that,4>
2. **The Claroty Team (2023)**. *NotPetya: Looking Back Six Years Later*. Available at: <https://claroty.com/blog/notpetya-looking-back-six-years-later#:~:text=ransomware%20attack%2C%20the%20NotPetya%20attack,the%20officers%20behind%20the%20attack>
3. **WeLiveSecurity (2017)**. *TeleBots back: Supply-chain attacks against Ukraine*. Available at: <https://www.welivesecurity.com/2017/06/30/telebots-back-supply-chain-attacks-against-ukraine/>
4. **CISA (2017)**. *Petya/NotPetya malware – updated advisory*. Available at: <https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware#:~:text=NotPetya%20leverages%20multiple%20propagation%20methods,the%20lateral%20movement%20techniques%20below>
5. **LogRhythm Labs (2017)**. *NotPetya Technical Analysis*. Available at: <https://gallery.logrhythm.com/threat-intelligence-reports/notpetya-technical-analysis-logrhythm-labs-threat-intelligence-report.pdf#:~:text=1,functions%20are%20contained%20in%20a>
6. **Cybereason (2017)**. *A Quick Recap on NotPetya*. Available at: <https://www.cybereason.com/blog/blog-a-quick-recap-on-notpetya#:~:text=infected%20system%20prior%20to%20reboot,recovery%20via%20ransom%20payment%20impossible>
7. **Tripwire (2017)**. *NotPetya: Timeline of a Ransomware*. Available at: <https://www.tripwire.com/state-of-security/notpetya-timeline-of-a-ransomware#:~:text=Ukraine%27s%20police%20confirm%20MeDoc%2C%20an,later%20that%20afternoon%20reiterating%20security>
8. **Greenberg, A. (2018)**. *The Untold Story of NotPetya, the Most Devastating Cyberattack in History*. Available at: <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/#:~:text=All%20across%20Maersk%20headquarters%2C%20the,other%20sections%20of%20the%20building>
9. **SIPA, Columbia University (2022)**. *NotPetya Final Report*. Available at: <https://www.sipa.columbia.edu/sites/default/files/2022-11/NotPetya%20Final.pdf#:~:text=In%202017%2C%20A,companies%E2%80%99%20most%20basic%20operations%20relied>
10. **U.S. Department of Justice (2020)**. *Six Russian GRU Officers Charged in Connection with Worldwide Deployment of Destructive Malware*. Available at: <https://www.justice.gov/archives/opa/pr/six-russian-gru-officers-charged-connection-worldwide-deployment-destructive-malware-and#:~:text=Marche%21%E2%80%9D%20,in%20losses%20from%20the%20attacks>
11. **U.S. Department of Justice (2020)**. *Further detail on GRU attacks and domain registration*. Available at: <https://www.justice.gov/archives/opa/pr/six-russian-gru-officers-charged-connection-worldwide-deployment-destructive-malware-and#:~:text=Their%20computer%20attacks%20used%20some,registration%20of%20a%20do%20main%20name>

12. **The Claroty Team (2023).** *NotPetya: Looking Back Six Years Later (impact on critical infrastructure)*. Available at: <https://claroty.com/blog/notpetya-looking-back-six-years-later#:~:text=Following%20the%20NotPetya%20attack%2C%20adversaries,dramatic%20impact%20on%20critical%20infrastructure>
13. **Trend Micro (2017).** *Large-scale ransomware attack hits Europe hard*. Available at: [https://www.trendmicro.com/en\\_us/research/17/f/large-scale-ransomware-attack-progress-hits-europe-hard.html](https://www.trendmicro.com/en_us/research/17/f/large-scale-ransomware-attack-progress-hits-europe-hard.html)
14. **LogRhythm Labs (2017).** *NotPetya Technical Analysis (sample details)*. Available at: <https://gallery.logrhythm.com/threat-intelligence-reports/notpetya-technical-analysis-logrhythm-labs-threat-intelligence-report.pdf#:~:text=64b0b58a2c030c77fdb2b537b2fcc4af432bc55ffb36599a31d418c7c69e94b1%20The%20analyzed%20samples%20of,This%20is>
15. **MITRE ATT&CK (2025).** *NotPetya software behaviors & credential dumping techniques*. Available at: <https://attack.mitre.org/software/S0368/#:~:text=Enterprise%20%20T1003%20%20,Credential%20Dumping%20%3A%20%2053>
16. **Microsoft Security Blog (2018).** *Overview of Petya: A rapid cyberattack*. Available at: <https://www.microsoft.com/en-us/security/blog/2018/02/05/overview-of-petya-a-rapid-cyberattack/#:~:text=1st%20phase%3A%20Targeting%20%E2%80%93%20Identify,machine%20to%20attack%2Fspread%20to%20next>
17. **Cybereason (2017).** *NotPetya propagation similar to WannaCry attack*. Available at: <https://www.cybereason.com/blog/blog-a-quick-recap-on-notpetya#:~:text=Like%20the%20WannaCry%20attack%2C%20it,network%20and%20infect%20other%20machines>
18. **Microsoft Security Blog (2018).** *Petya vs WannaCry observation summary*. Available at: <https://www.microsoft.com/en-us/security/blog/2018/02/05/overview-of-petya-a-rapid-cyberattack/#:~:text=Our%20observation%20was%20that%20Petya,WannaCrypt%20attacks%20and%20associated%20publicity>
19. **MITRE ATT&CK (2025).** *NotPetya indicator removal & log wiping techniques*. Available at: <https://attack.mitre.org/software/S0368/#:~:text=Enterprise%20%20T1070%20%20,Indicator%20Removal%20%3A%20%2049>
20. **LogRhythm Labs (2017).** *\*NotPetya scheduled task behavior on Vista/2008/7+*. Available at: <https://gallery.logrhythm.com/threat-intelligence-reports/notpetya-technical-analysis-logrhythm-labs-threat-intelligence-report.pdf#:~:text=Vista%2F2008%2F7%20or%20greater%2C%20a%20scheduled,such%20as%20XP%29%2C%20the>
21. **CISA (2017).** *NCCIC sample received: destructive malware advisory*. Available at: <https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware#:~:text=NCCIC%20received%20a%20sample%20of,destructive%20malware%20rather%20than%20ransomware>
23. **Cybereason (2017).** *NotPetya not profit-motivated but destructive*. Available at: <https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware#:~:text=NCCIC%20received%20a%20sample%20of,destructive%20malware%20rather%20than%20ransomware>
24. **Gofman, I. (2017).** *Advanced Threat Analytics security research network technical analysis: NotPetya*. Microsoft Security Blog. Available at:

<https://www.microsoft.com/en-us/security/blog/2017/10/03/advanced-threat-analytics-security-research-network-technical-analysis-notpetya/>

25. **Makrakis, G. M., Koliass, C., Kambourakis, G., Rieger, C., & Benjamin, J. (2021).** *Industrial and Critical Infrastructure Security: Technical Analysis of Real-Life Security Incidents*. IEEE Access, Vol. 9, pp. 165295–165325. Available at: [https://www.researchgate.net/publication/356822382\\_Industrial\\_and\\_Critical\\_Infrastructure\\_Security\\_Technical\\_Analysis\\_of\\_Real-Life\\_Security\\_Incidents](https://www.researchgate.net/publication/356822382_Industrial_and_Critical_Infrastructure_Security_Technical_Analysis_of_Real-Life_Security_Incidents)
26. **Kein, C. (2015).** *STRIDE Threat Model: A Complete Guide*. Available at: <https://www.jit.io/resources/app-security/stride-threat-model-a-complete-guide>
27. **Microsoft Ignite (2022).** Microsoft Threat Modeling Tool threats. Available at: <https://learn.microsoft.com/en-us/users/register?redirectUrl=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fazure%2Fsecurity%2Fdevelop%2Fthreat-modeling-tool-threats>

## ***Appendix***

### **List of Figures**

- Figure 1. Sequence of events and chronological timeline of the NotPetya's progression.
- Figure 2. Disguised CHKDSK screen.
- Figure 3. Ransomware notices displayed after reboot.
- Figure 4. STRIDE Threat Model
- Figure 5. Malware Propagation.
- Figure 6. NotPetya Attack process.

### **List of Tables**

- Table 1. STRIDE Spoofing.
- Table 2. STRIDE Tampering.
- Table 3. STRIDE Repudiation.
- Table 4. STRIDE Information Disclosure.
- Table 5. STRIDE Denial of Service.
- Table 6. STRIDE Elevation of Privilege.
- Table 7. STRIDE Threat Matrix vs. MITRE/NIST

### **External Files**

- File 1. [Demo.py](#)
- File 2. [Demo.txt](#)

### **Poster**