Introduction

Cybersecurity Fundamentals & Threat Landscape

Dr Petar Radanliev
Author, Academic, Researcher, Faculty,
Department of Computer Science
University of Oxford / Alan Turing Institute

Introduction

PhD (2013), MSc (2007), and BA Hons (2006) in related fields to AI Security

Current Faculty at University of Oxford (AI Security and Cybersecurity), Research Associate, Alan Turing Institute (AI Security and Digital Identity), Lecturer at the Swiss Cyber Institute, Lecturer at Pearson's and O'Reilly, Lecturer at Oxford Summer Academy (UK Schooling).

Key Metrics: H-index: 24, 3500+ citations, 100+ papers.

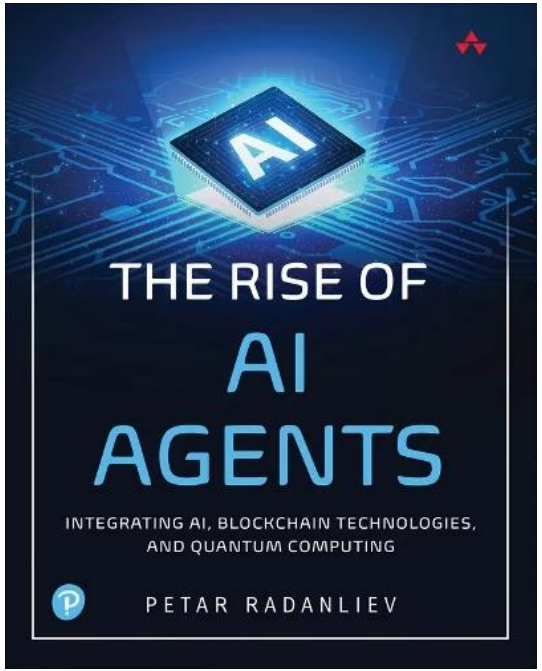
Grants: Fulbright, Prince of Wales, EPSRC (6), Cisco (2).

Previous roles: Postdoctoral at University of Cambridge, Imperial College London, Fulbright Fellow at MIT, Cybersecurity roles with the UK MoD and RBS

Authored 4 books and over 100 peer-reviewed publications in AI security, post-quantum cryptography, and threat detection

Industry and academic experience combining advanced research with practical application

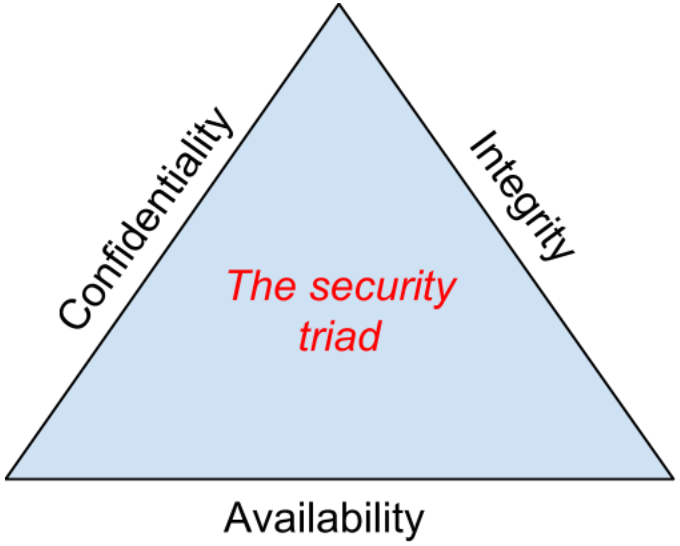
Background reading - Book



The Rise of AI Agents: Integrating AI, Blockchain Technologies, and Quantum Computing (Paperback)

https://www.oreilly.com/library/view/the-rise-of/9780135352939/

Introduction to the Lecture

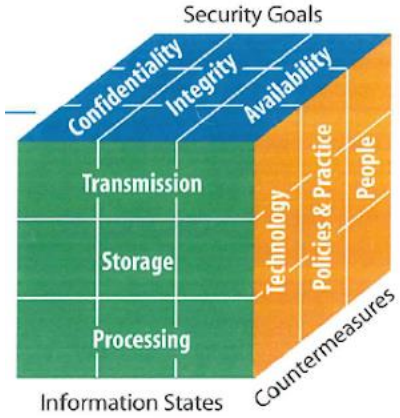


Lecture 1: Key security concepts, threat types, and the real-world context of cyberattacks.

Learning objectives:

1. Understand the CIA Triad and how it underpins cybersecurity.
2. Learn the difference between risk, vulnerabilities, and threats.
3. Become familiar with threat intelligence, MITRE ATT&CK, and TTPs.
4. Recognise common malware types and APT operations.

Basic Vocabulary



CIA Triad:

Confidentiality – Keeping information hidden from unauthorised parties.

Example: Encryption of sensitive documents.

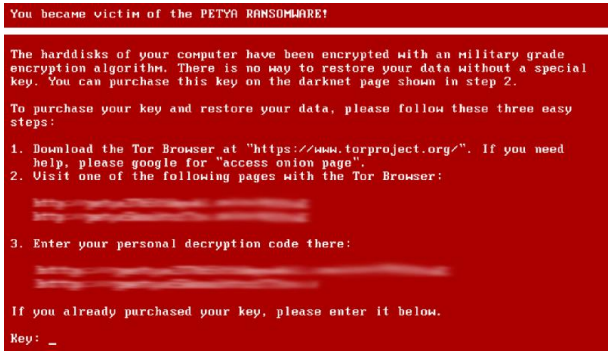
Integrity – Ensuring data is accurate and unaltered.

Example: Digital signatures for file verification.

Availability – Ensuring data and systems are accessible when needed.

Example: Redundant systems in data centres.

CIA Triad – Real-World Trade-offs and Examples

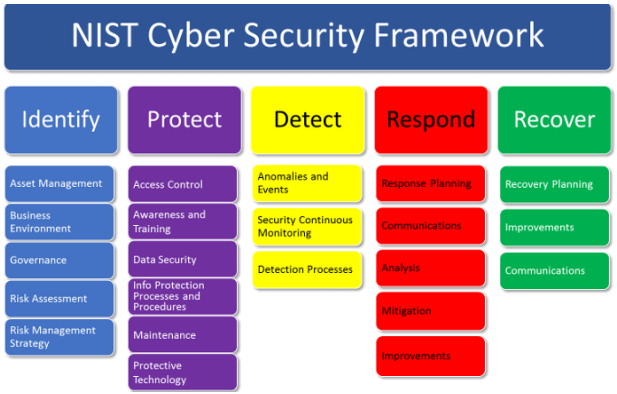


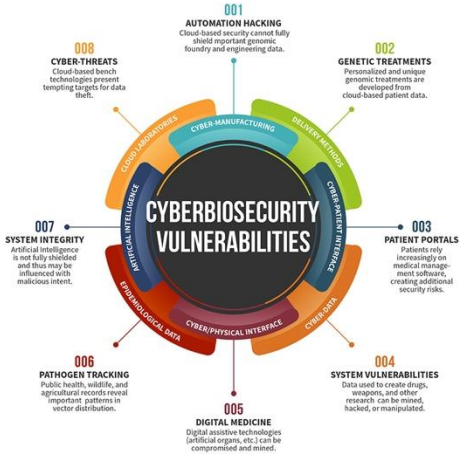
- Confidentiality: Equifax data breach (2017) – 147M personal records exposed.
- Integrity: NotPetya (2017) – Data destruction disguised as ransomware.
- Availability: Dyn DDoS attack (2016) – Major internet outage in the US & Europe.
- Trade-offs: Increasing availability can sometimes reduce confidentiality.

Risk, Vulnerabilities & Threats

- **Risk**: The *potential* for loss/damage when a threat exploits a vulnerability.
- **Vulnerability**: A weakness in a system.
- *Example:* Unpatched operating system.
- **Threat**: Anything capable of exploiting a vulnerability.
- *Example:* A ransomware gang targeting outdated servers.

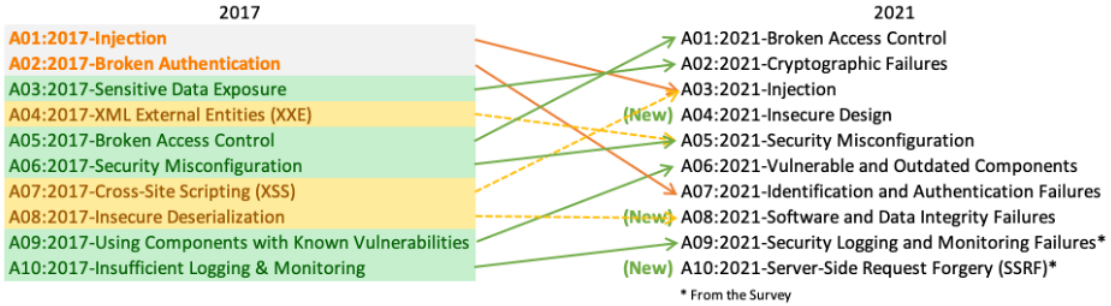
Risk Assessment – Likelihood and Impact





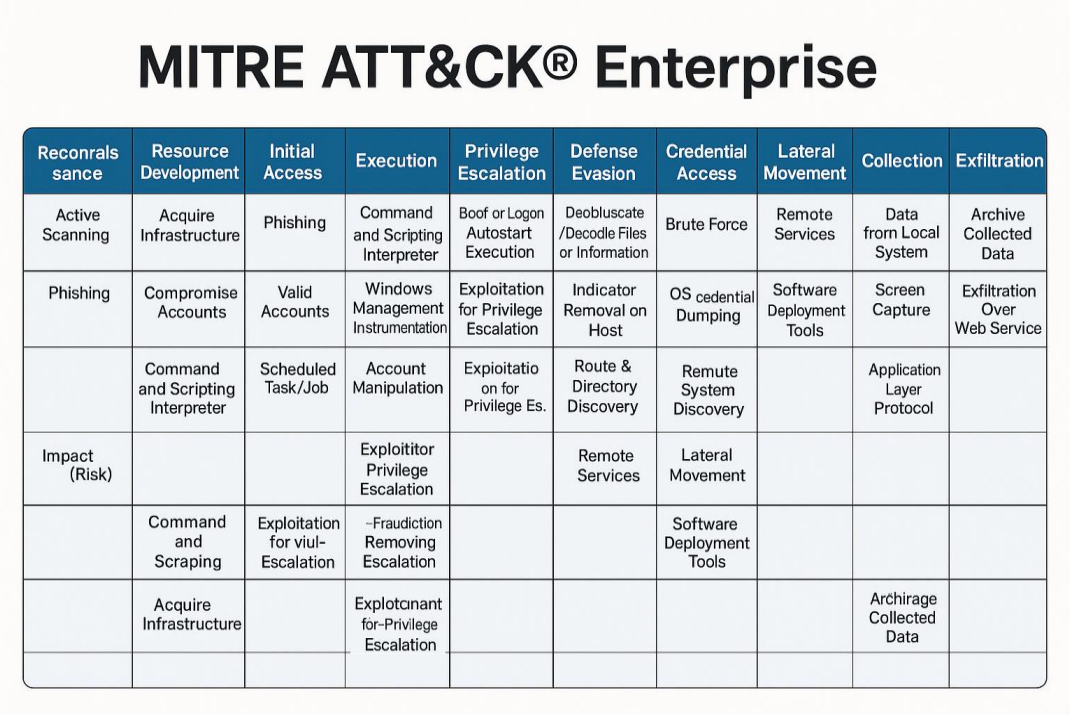
- Risk = Likelihood × Impact – foundational formula for security prioritisation.
- High likelihood + high impact = critical risk requiring urgent mitigation.
- Example: Unpatched VPN vulnerability actively exploited by ransomware groups.
- Visual: Risk matrix showing low, medium, high risk zones.

MITRE ATT&CK – Mapping a Real-World Incident



- Case: WannaCry Ransomware (2017)
- Initial Access – Exploited SMBv1 vulnerability (EternalBlue).
- Execution – Deployed ransomware payload.
- Impact – Encrypted files, demanded Bitcoin ransom.
- Use: Helps defenders perform gap analysis and improve defences.

Threat Intelligence



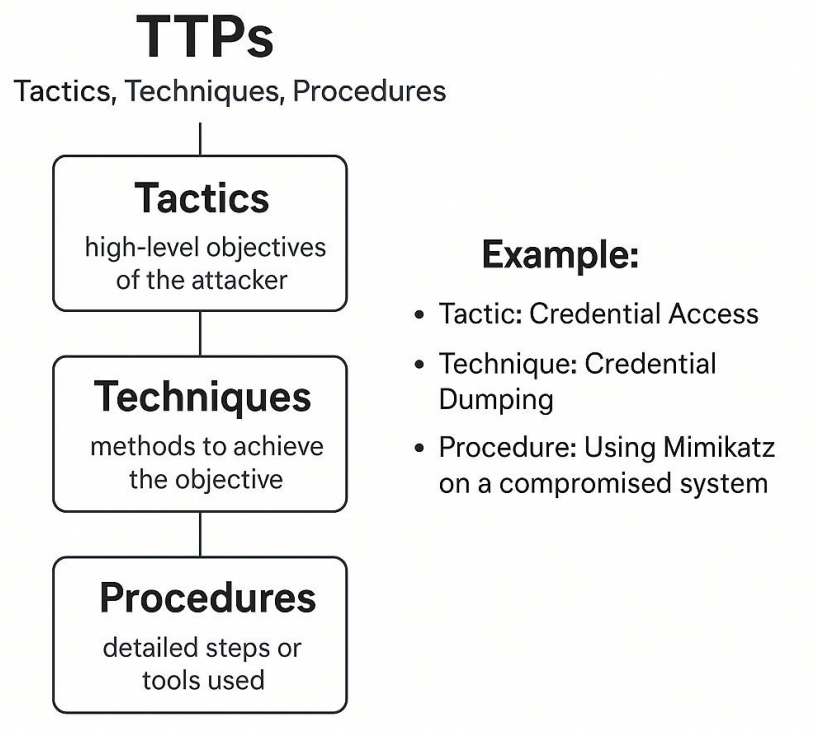
MITRE ATT&CK® Enterprise

Reconrals sance	Resource Development	Initial Access	Execution	Privilege Escalation	Defense Evasion	Credential Access	Lateral Movement	Collection	Exfiltration
Active Scanning	Acquire Infrastructure	Phishing	Command and Scripting Interpreter	Boof or Logon Autostart Execution	Deobluscate /Decodle Files or Information	Brute Force	Remote Services	Data frorn Local System	Archive Collected Data
Phishing	Compromise Accounts	Valid Accounts	Windows Management Instrumentation	Exploitation for Privilege Escalation	Indicator Removal on Host	OS cedential Dumping	Software Deployment Tools	Screen Capture	Exfiltration Over Web Service
	Command and Scripting Interpreter	Scheduled Task/Job	Account Manipulation	Expioitatio on for Privilege Es.	Route & Directory Discovery	Remute System Discovery		Application Layer Protocol	
Impact (Risk)			Exploititor Privilege Escalation		Remote Services	Lateral Movement			
	Command and Scraping	Exploitation for viul- Escalation	-Fraudiction Removing Escalation			Software Deployment Tools			
	Acquire Infrastructure		Explotcınant for-Privilege Escalation					Archirage Collected Data	

MITRE ATT&CK Framework

- Global knowledge base of adversary behaviour.
- Organised by **tactics** (the “why”) and **techniques** (the “how”).
- Helps map and anticipate attacker methods.

TTPs – Tactics, Techniques, Procedures

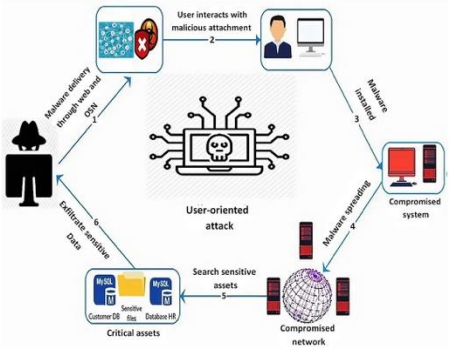


- **Tactics** – high-level objectives of the attacker.
- **Techniques** – methods to achieve the objective.
- **Procedures** – detailed steps or tools used.

Example:

- Tactic: Credential Access
- Technique: Credential Dumping
- Procedure: Using Mimikatz on a compromised system.

TTP Examples – Multiple Campaigns



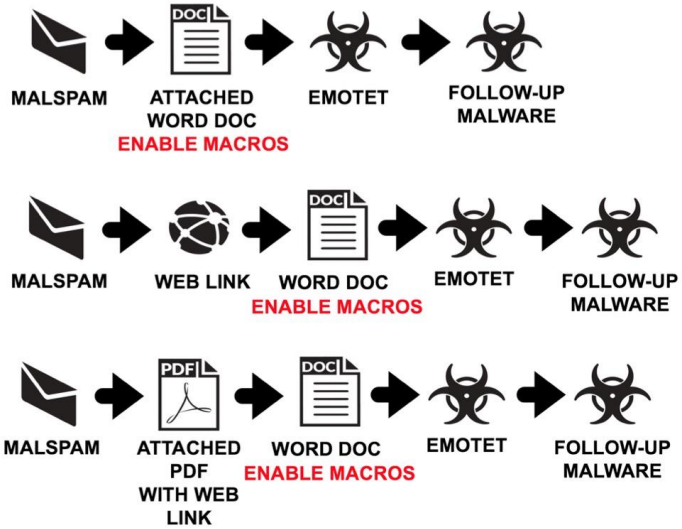
	Something "Bad" (aversive)	Something "Good" (rewarding)
Giving (positive)	Positive Punishment (behavior is weakened)	Positive Reinforcement (behavior is strengthened)
Taking Away (negative)	Negative Reinforcement (behavior is strengthened)	Negative Punishment (behavior is weakened)

- Tactic: Initial Access – Technique: Spearphishing – Procedure: Malicious PDF.
- Tactic: Persistence – Technique: Registry Run Keys – Procedure: Custom loader.
- Tactic: Credential Access – Technique: Keylogging – Procedure: Malware implant.
- Tactic: Exfiltration – Technique: HTTPS Exfiltration – Procedure: Cloud storage.

Types of Threat Intelligence

- Strategic – Long-term trends, geopolitical context (for executives).
- Tactical – TTPs and campaign patterns (for SOC teams).
- Operational – Active attack indicators, live incident data.
- Technical – IOCs such as IPs, hashes, domains from malware samples.
- Sources: OSINT, dark web monitoring, ISACs, vendor threat feeds.

Threat Types & Attack Vectors

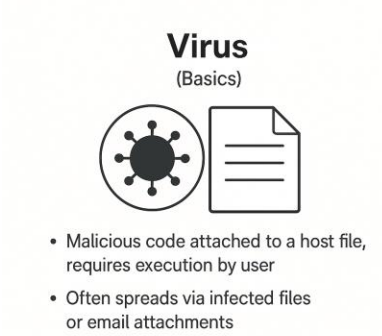


Malware (Malpedia)

- Any software intentionally designed to cause harm.
- See: **https://malpedia.caad.fkie.fraunhofer.de/ –** a reference repository of malware families.

SCHOOLING UK SCHOOLING

Threat Types & Attack Vectors

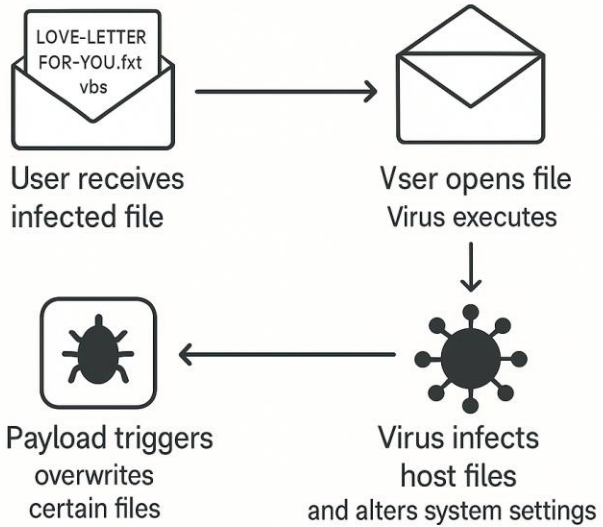


Virus (Basics)

- Malicious code attached to a host file, requires execution by user.
- Often spreads via infected files or email attachments.

SCHOOLING UK SCHOOLING

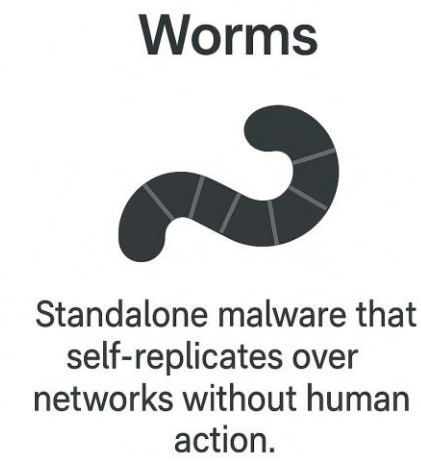
Threat Types & Attack Vectors



Example virus: ILOVEYOU (2000) – a mass-mailing virus disguised as a love letter attachment in an email.

1. User receives infected file (email attachment “LOVE-LETTER-FOR-YOU.txt.vbs”).
2. User opens file → Virus executes.
3. Virus infects host files and alters system settings.
4. Virus spreads via victim’s email contacts.
5. Payload triggers → overwrites certain files, causes data loss.

Threat Types & Attack Vectors



Worms

- Standalone malware that self-replicates over networks without human action.
- *Example:* 2003 **Slammer Worm** took down parts of the internet in minutes.

Advanced Worm that we expect - Not Yet Realised:

- Morris II
- WormGPT Variants
- Plague

Most Advanced Worm Not Yet Realised

Threat	Status	Highlights
Morris II	Proof of concept	AI-driven, self-replicating worm for GenAI systems
WormGPT Variants	Active (not a true worm)	AI-generated phishing and malware via LLM APIs
Plague	Real-world malware	Highly stealthy Linux backdoor (non-worm)

The Front-Runners in Advanced Worm Concepts (2025)

1. Morris II – AI-Powered Worm (Proof of Concept)

Overview: Developed by security researchers, Morris II is an AI-enabled worm that leverages adversarial, self-replicating prompts to exploit generative AI ecosystems, particularly email assistants. It's capable of autonomously spreading across systems by tricking AI models into propagating malicious prompts.

Significance: Although not active in the wild, Morris II represents a new category of threat, **AI worms**, that could autonomously propagate and bypass traditional security measures.

The next two Front-Runners

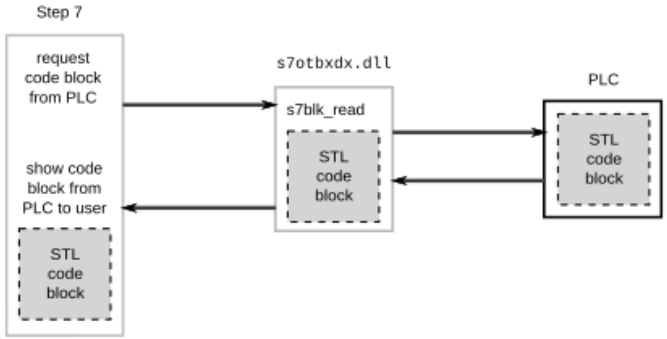
2. WormGPT Variants

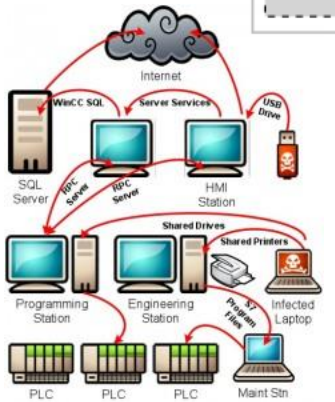
- **What they are**: Malware tools built on top of AI models like **xAI's Grok** and **Mistral's Mixtral**, capable of generating **phishing messages**, **malware scripts**, and other malicious content on demand.
- **Relevance**: These represent AI-augmented tools rather than self-replicating worms—but demonstrate the growing capability for AI to facilitate highly automated malware.

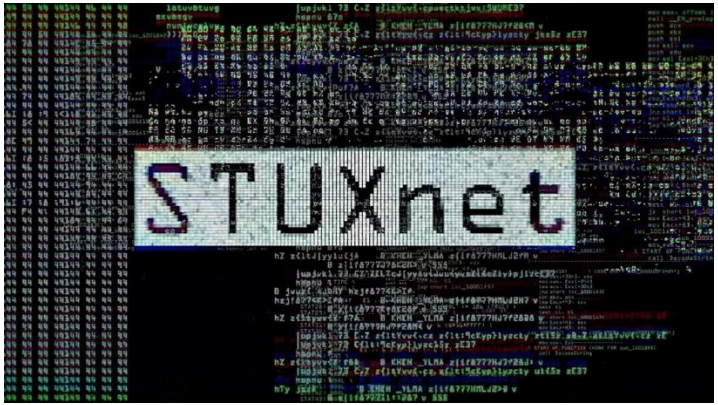
3. Plague - Advanced Stealth Malware (Non-Worm)

- **Description**: A sophisticated Linux **Pluggable Authentication Module (PAM)** backdoor discovered in August 2025. It evades detection, embeds deeply into authentication mechanisms, and persists through system updates—though it is not a worm by classification.

Worms – Notable Historical Example







- Stuxnet (2010) – Targeted Iranian nuclear centrifuges.
- Used multiple zero-days and a programmable logic controller payload.
- Blended physical and cyberattack vectors.
- Demonstrates potential of state-sponsored worms.

Threat Types & Attack Vectors

Ransomware

- Encrypts victim's data, demands payment for decryption.
- **Real-life example:** WannaCry (2017) infected 200,000+ systems, impacting NHS in the UK.
- **Top 3 families currently prevalent** (as per recent threat reports):

1. **LockBit**
2. **BlackCat (ALPHV)**
3. **Clop**

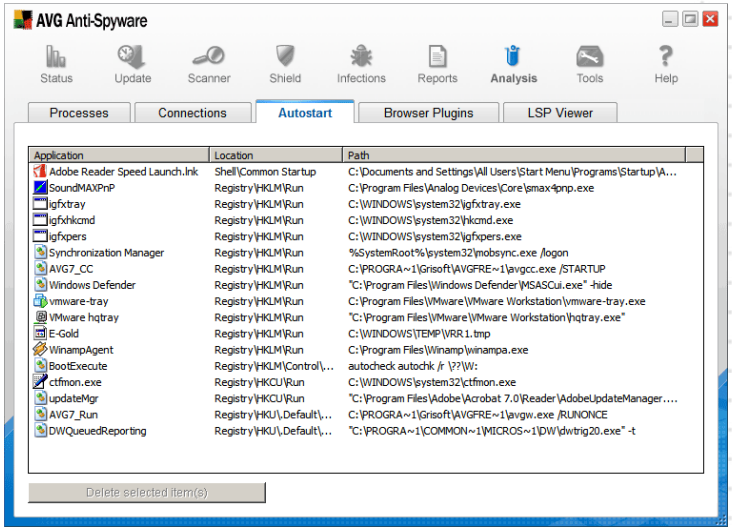
Evolution of Ransomware Tactics

- Single extortion – Payment for decryption only.
- Double extortion – Threat to leak stolen data.
- Triple extortion – Targeting partners/customers of victim organisation.
- Trend: Increased targeting of critical infrastructure and supply chains.

Threat Types & Attack Vectors

Spyware

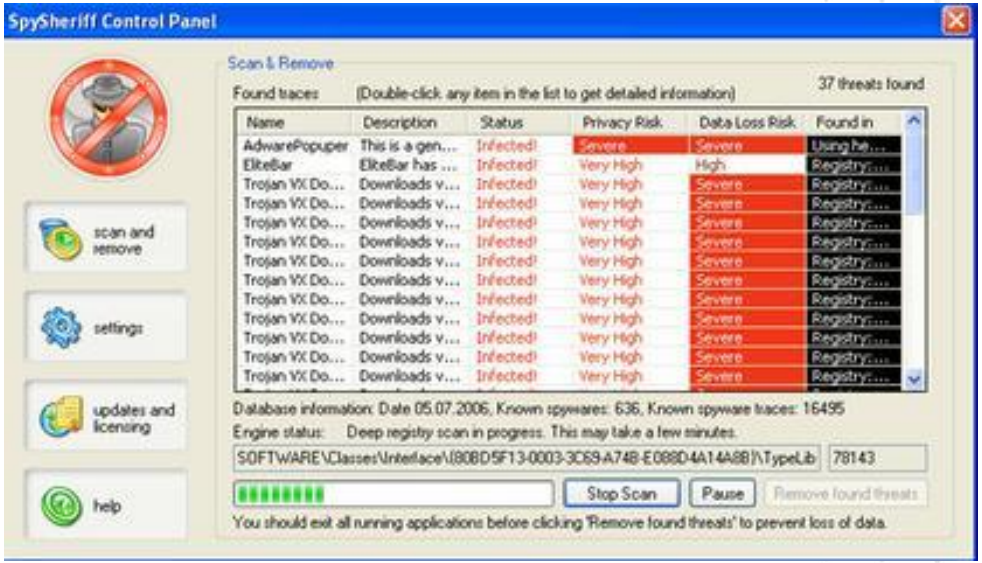
- Stealth software to collect user information without consent.
- *Example:* Keyloggers stealing banking credentials.



Threat Types & Attack Vectors

Adware

- Displays unwanted ads; can slow down system or track user behaviour.
- *Example:* Browser hijackers.



Threat Types & Attack Vectors

APT (Advanced Persistent Threats)

- Highly skilled, well-funded groups with long-term objectives.
- Often state-sponsored.
- Names assigned by security vendors (Mandiant, CrowdStrike, etc.).
- Example: APT28 (“Fancy Bear”) – linked to Russian military intelligence.

APT Campaign Lifecycle

- Reconnaissance – Identifying targets and vulnerabilities.
- Initial Access – Spearphishing, exploiting vulnerabilities.
- Persistence – Backdoors, remote access tools.
- Lateral Movement – Moving within the network.
- Exfiltration – Stealing sensitive data.
- Impact – Disruption, destruction, or strategic gain.

Case Study – NoName057(16)

Profile: Pro-Russian hacktivist group.

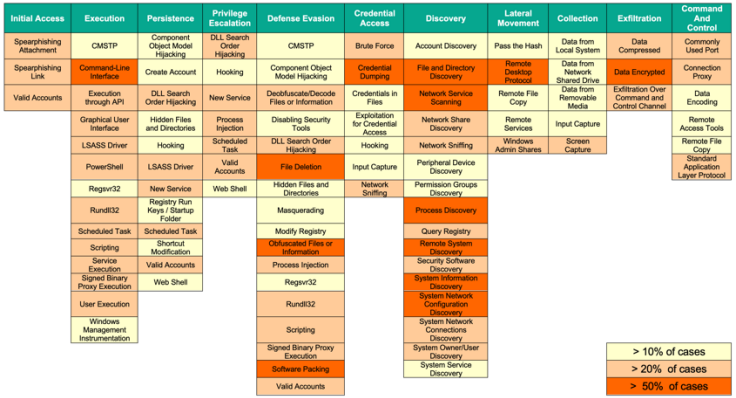
- Known for DDoS attacks on European entities.

Events:

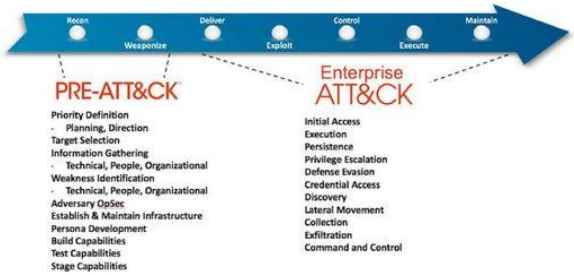
- Attacks during **World Economic Forum (WEF)** in Davos.
- Disruptions during the **Ukraine Peace Conference** in Switzerland.
- **Tactics**: Publishes attack targets openly.
- **Website**: https://witha.name – public DDoS config and victim list.

Threat intelligence techniques to map their activity in MITRE ATT&CK.

NoName057(16) – Mapping & Defences

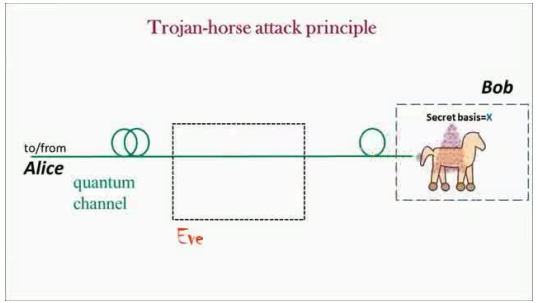


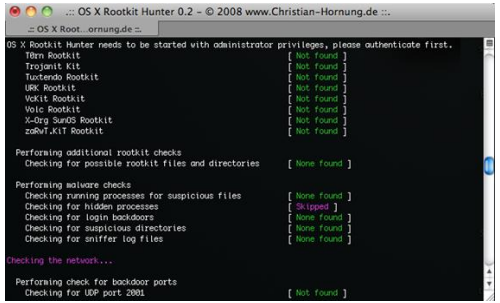
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
Spearphishing Attachment	CMSTP	Component Object Model Hijacking	DLL Search Order Hijacking	CMSTP	Brute Force	Account Discovery	Pass the Hash	Data from Local System	Data Compressed	Commonly Used Port
Spearphishing Link	Command-Line Interface	Create Account	Hooking	Component Object Model Hijacking	Credential Dumping	File and Directory Discovery	Remote Desktop Protocol	Data from Network Shared Drive	Data Encrypted	Connection Proxy
Valid Accounts	Execution through API	DLL Search Order Hijacking	New Service	Deobfuscate/Decode Files or Information	Credentials in Files	Network Service Scanning	Remote File Copy	Data from Removable Media	Exfiltration Over Command and Control Channel	Data Encoding
	Graphical User Interface	Hidden Files and Directories	Process Injection	Disabling Security Tools	Exploitation for Credential Access	Network Share Discovery	Remote Services	Input Capture		Remote Access Tools
	LSASS Driver	Hooking	Scheduled Task	DLL Search Order Hijacking	Hooking	Network Sniffing	Windows Admin Shares	Screen Capture		Remote File Copy
	PowerShell	LSASS Driver	Valid Accounts	File Deletion	Input Capture	Peripheral Device Discovery				Standard Application Layer Protocol
	Regsvr32	New Service	Web Shell	Hidden Files and Directories	Network Sniffing	Permission Groups Discovery				
	Rundll32	Registry Run Keys / Startup Folder		Masquerading		Process Discovery				
	Scheduled Task	Scheduled Task		Modify Registry		Query Registry				
	Scripting	Shortcut Modification		Obfuscated Files or Information		Remote System Discovery				
	Service Execution	Valid Accounts		Process Injection		Security Software Discovery				
	Signed Binary Proxy Execution	Web Shell		Regsvr32		System Information Discovery				
	User Execution			Rundll32		System Network Configuration Discovery				
	Windows Management Instrumentation			Scripting		System Network Connections Discovery				
				Signed Binary Proxy Execution		System Owner/User Discovery			> 10% of cases	
				Software Packing		System Service Discovery			> 20% of cases	
				Valid Accounts					> 50% of cases	

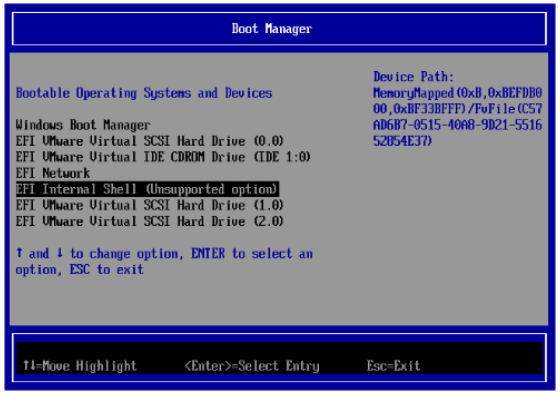


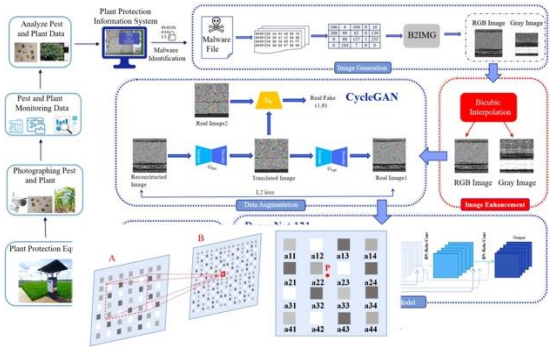
- Mapped to MITRE ATT&CK: T1498 (Network Denial of Service), T1071 (Application Layer Protocol).
- TTPs: Public victim lists, voluntary botnet recruitment.
- Defences: DDoS mitigation services, rate limiting, geo-blocking.
- Use intelligence sharing to pre-empt attacks during high-profile events.

Additional Malware Types to Recognise









– Disguised as legitimate software (e.g., banking Trojans).

Rootkits – Hide malicious processes from detection.

Bootkits – Infect the bootloader for persistent compromise.

Fileless Malware – Operates in memory, avoiding file-based detection.

Malware is evolving

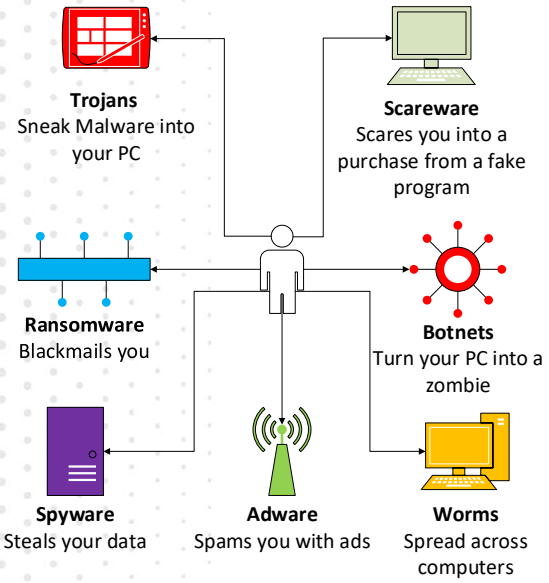
AI-Powered Malware (e.g., MedusaLocker, BlackMamba)

- Setting up the Static Malware Analysis tools
- Understanding Requirements for analysing malware without executing it
- Understanding the code structure, file signatures, and patterns
- Understanding Static Analysis Commands

AI malware and its capabilities

- Definition: AI-powered malware uses machine learning to enhance stealth, persistence, and adaptability.
- Capabilities:
  - Generates polymorphic behaviour to evade static signatures.
  - Uses AI for autonomous attack strategies.
- Examples of Real-World AI Malware:
  - **MedusaLocker**: AI-assisted ransomware with dynamic encryption patterns.
  - **BlackMamba**: Polymorphic malware exploiting AI for payload delivery.

Malware Categories



1. **Quiz** on malware and its categories (e.g., viruses, worms, Trojans, ransomware).
2. Importance of malware analysis in cybersecurity.

Fileless malware: Operates in memory to avoid detection and persist on the system

Mobile malware: Targets mobile devices to steal data, spy or damage the device

Wiper malware: Destroys data on infected systems, often irreversibly

Keyloggers: Records keystrokes to capture sensitive information like passwords

Cryptojacking: Uses system resources to mine cryptocurrency without the user's consent

Hybrid malware: Combines features of multiple malware types for more complex attacks

Classical Examples of Malware :

1. Viruses: Malicious programs that attach themselves to legitimate files, spreading when these files are executed.
   - Example: The **ILOVEYOU** virus (2000) infected millions of systems by disguising itself as a love letter email attachment.
2. Worms: Self-replicating malware that spreads through networks without requiring a host file.
   - Example: **Morris Worm** (1988), one of the first worms to spread across the internet, led to significant disruptions and inspired the creation of the Computer Emergency Response Team (CERT).
3. Trojans: Malicious programs that disguise themselves as legitimate software.
   - Example: **Zeus Trojan** (2007), used to steal banking information through keylogging and form grabbing.
4. Ransomware: Malware that encrypts a user's files and demands payment for the decryption key.
   - Example: **CryptoLocker** (2013) was one of the first major ransomware attacks to use advanced encryption techniques. It forced many victims to pay to regain access to their files.
5. Spyware: Software that gathers information from a system without the user's knowledge.
   - Example: **Pegasus** (2021): A spyware developed by NSO Group that exploited zero-click vulnerabilities in mobile devices to gather data from high-profile individuals.

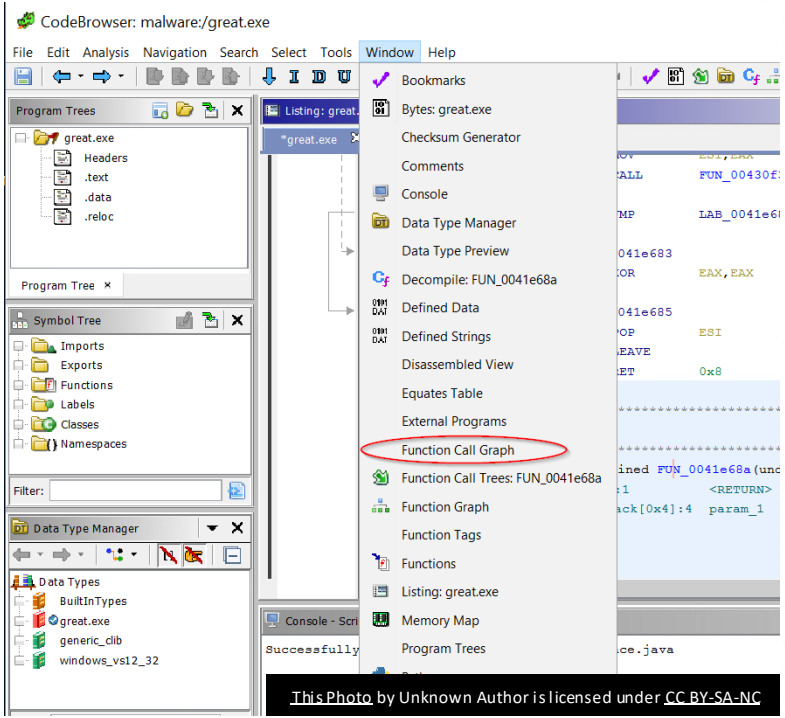
Recent Examples of Malware

1. **MedusaLocker**: An updated ransomware strain from a well-known family targeting critical infrastructure in healthcare, exploiting remote desktop protocol (RDP) vulnerabilities.
2. **Chaes Malware**: A modular, multi-stage malware targeting financial institutions and e-commerce platforms, known for its sophisticated web injection techniques to steal user credentials and financial data.
3. **BlackMamba:** An AI-powered polymorphic malware that uses generative adversarial networks (GANs) to modify its structure and evade detection tools by constantly changing its code.
4. **WiperBot:** (Anticipated) future AI-enhanced wiper malware designed to autonomously target specific file types and locations, effectively evading traditional defence mechanisms. (HermeticWiper, HermeticWizard, and HermeticRansom).

Static Analysis with Ghidra

Reverse engineering of AI-powered malware

- Ghidra is a free and open source reverse engineering tool developed by the NSA.
- The binaries were released at RSA Conference in March 2019; the sources were published one month later on GitHub.
- Ghidra is seen by many security researchers as a competitor to IDA Pro.



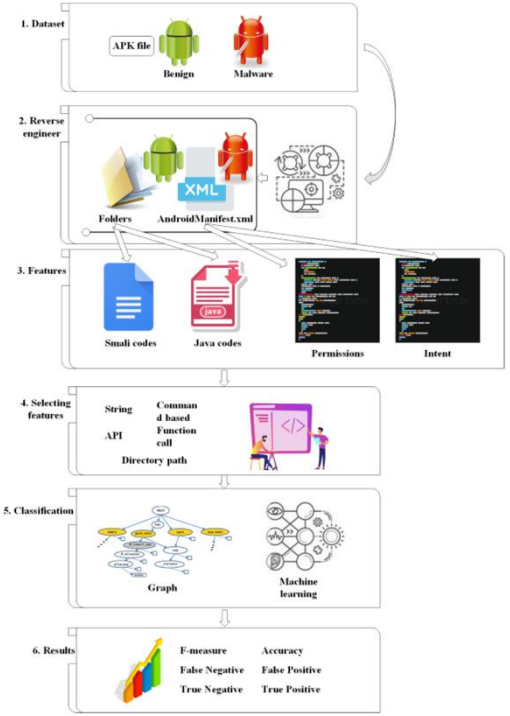
Exercise/Activity: Detection Method

Malware Type	Example	Detection Method	Explanation
Viruses	ILOVEYOU	Static Analysis	Analysed email attachment signatures and predictable VBScript patterns.
Worms	Morris Worm	Dynamic Analysis	Monitored abnormal network traffic and resource consumption.
Trojans	Zeus Trojan	Hybrid Analysis	Combined runtime observation (e.g., keylogging) with signature-based detection.
Ransomware	CryptoLocker	Hybrid Analysis	Observed encryption behaviour and C2 communication alongside static signatures.
Spyware	Pegasus	Dynamic/Hybrid Analysis	Detected behaviour exploiting zero-click vulnerabilities and reverse-engineered binaries.

Note

- Tools such as OWASP ZAP, SonarQube, and Contrast Assess, are primarily designed for detecting vulnerabilities in traditional software or web applications. They are not equipped to address machine learning-specific challenges, such as model inversion attacks, data poisoning, or adversarial examples.
- The reliance on tools that are ill-suited for addressing ML-specific vulnerabilities would represent a misalignment between the methods and the nature of the AI challenges.

AI-Based Static Malware Analysis

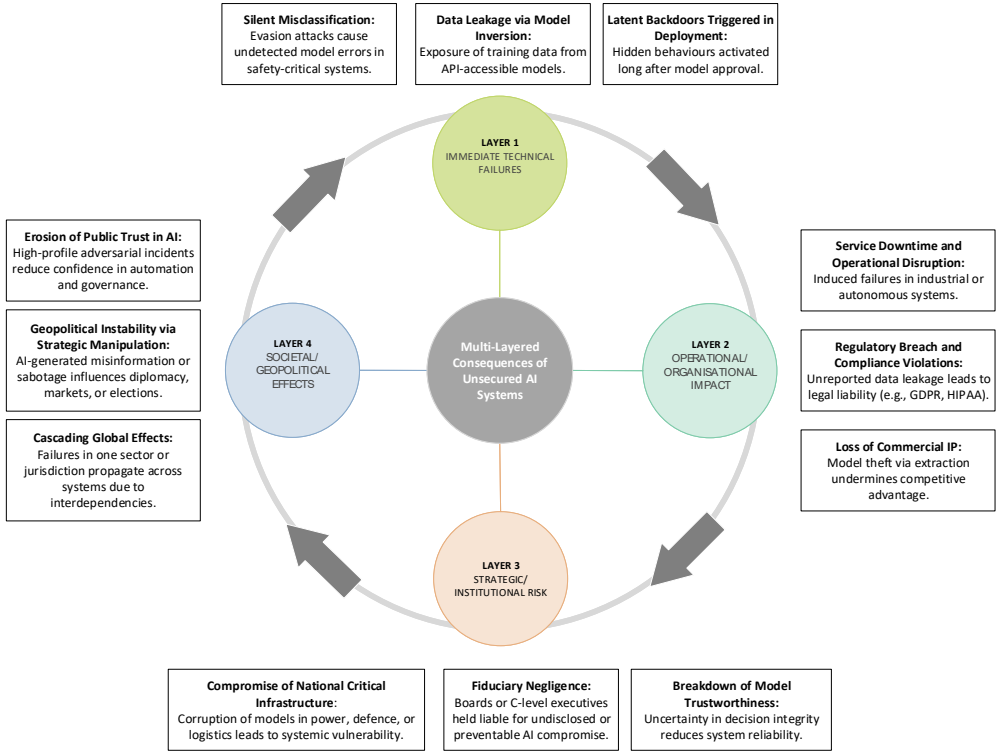


AI-Powered Malware (e.g., AI-generated code variations; Polymorphic behaviours)

Static analysis with DeepCode AI, Reverse-engineering polymorphic AI malware

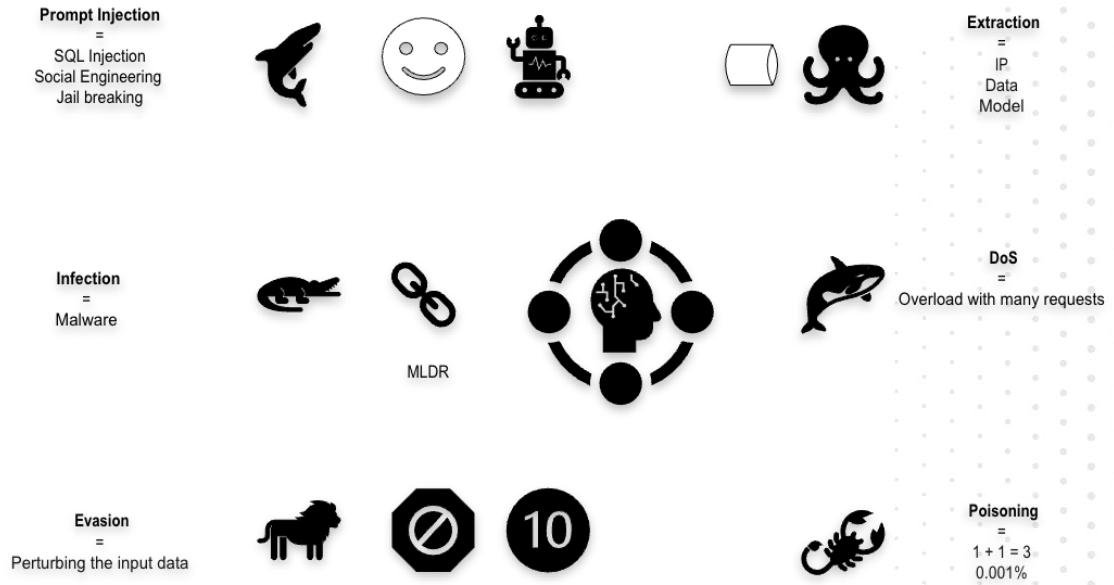
BlackMamba, using DeepCode AI to reverse-engineer the malware's polymorphic logic.

AI Attack Surface

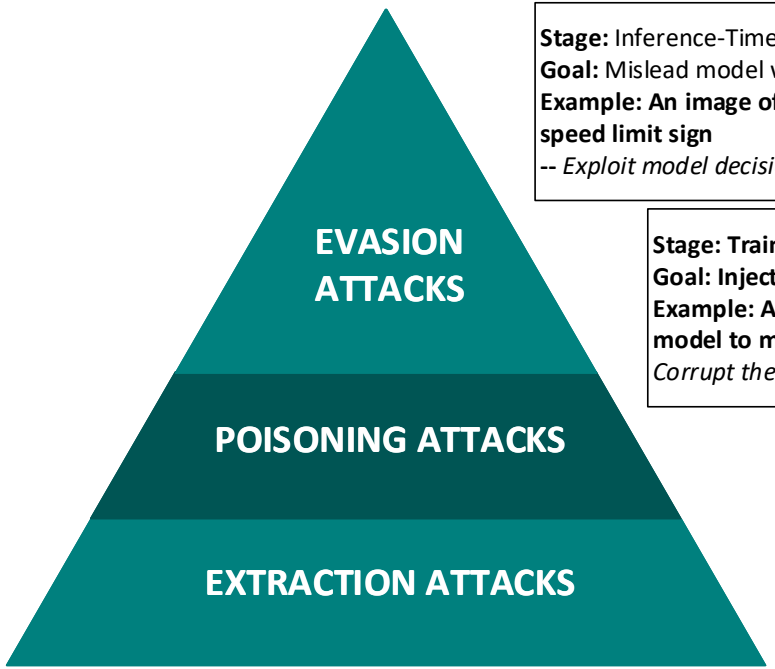


AI Threat Categories: Evasion, poisoning, model inversion, and backdoor attacks

1. **Evasion attacks** – perturbing inputs at inference time
2. **Poisoning attacks** – corrupting the training data
3. **Model inversion** – reconstructing private information
4. **Backdoor attacks** – embedding hidden triggers



AI Attack Surface Analysis

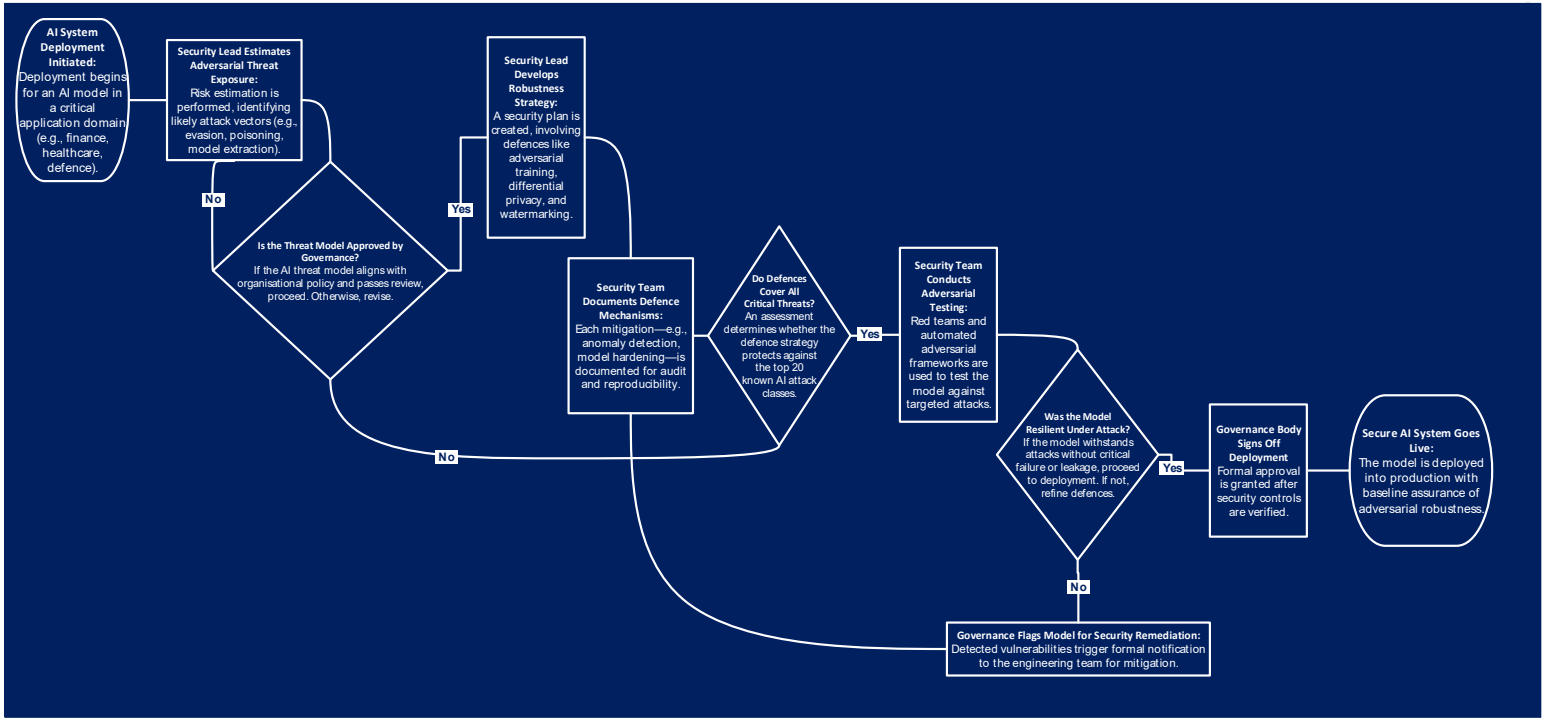


Stage: Inference-Time
Goal: Mislead model without changing its internal parameters
Example: An image of a stop sign subtly altered so that a vision model classifies it as a speed limit sign
-- *Exploit model decision boundaries by applying imperceptible perturbations to inputs*

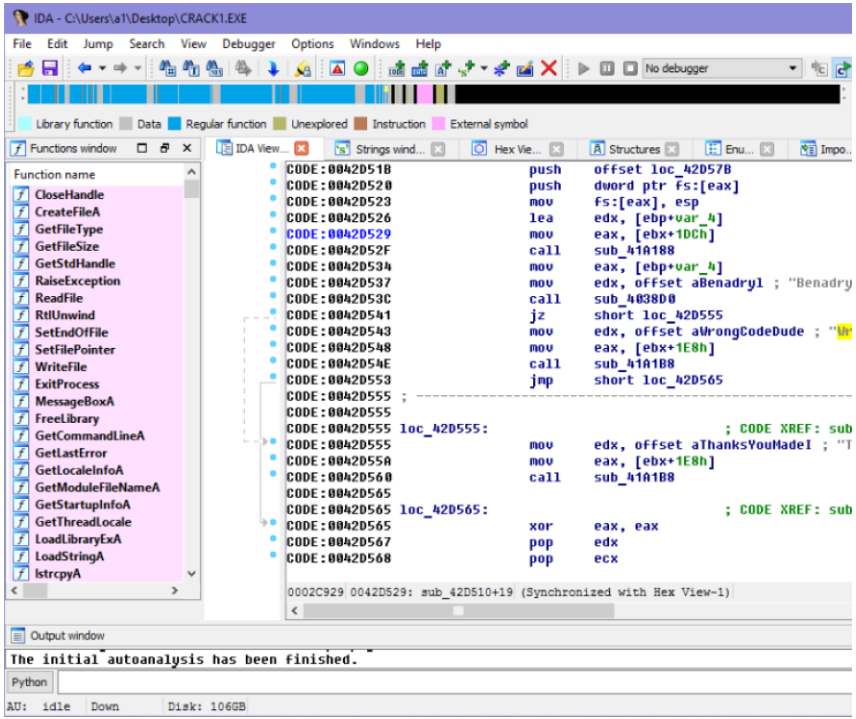
Stage: Training-Time
Goal: Inject malicious behaviour into the model's internal logic
Example: A small number of training images with hidden patterns cause the model to misclassify specific inputs when the pattern is present
Corrupt the learning process via modified or malicious training data

Stage: Post-Deployment Interaction
Goal: Steal model IP or recover training data through model outputs
Example: Systematic API querying to replicate a fraud detection model or infer presence of individuals in training data
Exploit model outputs to extract private, proprietary, or sensitive information

AI Security Risk Assessment Exercise



AI-driven static analysis tools, their advantages and limitations



1. **Definition**: Analysis of the malware without executing it.
2. **Techniques and Tools**:
   a. Reverse engineering: Using disassemblers (e.g., IDA Pro).
   b. Code inspection: Analysing the malware binary.
   c. Strings extraction: Identifying hardcoded values.
   d. Hash comparison: MD5, SHA-256 for file signatures.
3. **Advantages and Limitations**:
   a. Static analysis is safe but limited by obfuscation and encryption techniques used by advanced malware.

Static Malware Analysis (Classical Tools):

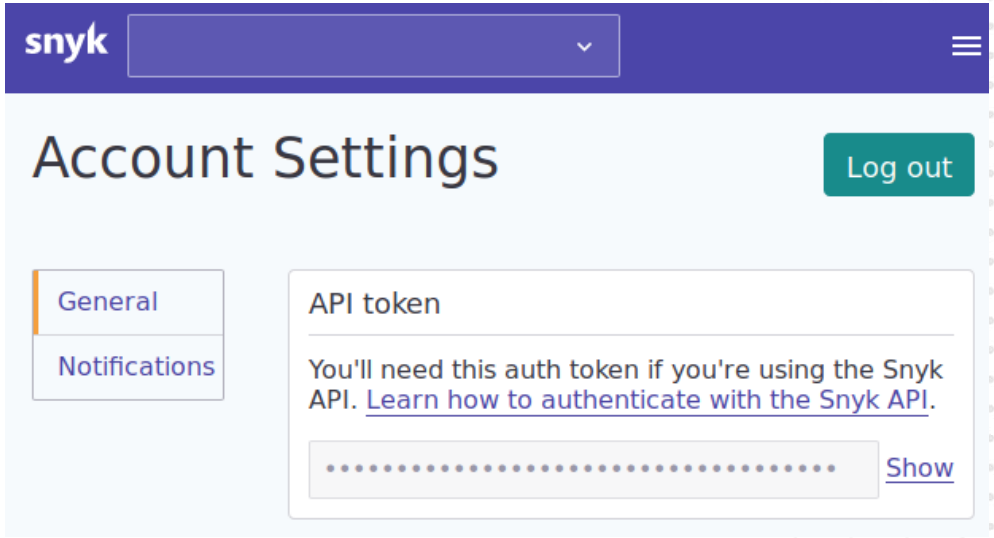
Analysis Involves analysing malware code without executing it, focusing on binary files or disassembled code.

1. Tools:
    1. **IDA Pro** (Interactive DisAssembler): A widely used tool for reverse engineering that disassembles binaries and creates an assembly code representation.
    2. **PEiD**: Used to detect the packer, compiler, or cryptor used in PE (Portable Executable) files.
    3. **Binwalk**: A tool used to extract information and files embedded within binaries, often used for firmware analysis.
2. Real-world Example:
    a. Static Analysis of WannaCry: Analysts were able to identify an embedded "**kill switch**" domain in the **malware binary** that, when activated, stopped the ransomware from spreading.

Static analysis with DeepCode AI

Reverse-engineering polymorphic AI malware

- Static Application Security Testing (SAST).
- SNYK DeepCode AI features.
- DeepCode AI code analysis.



Static analysis with DeepCode AI

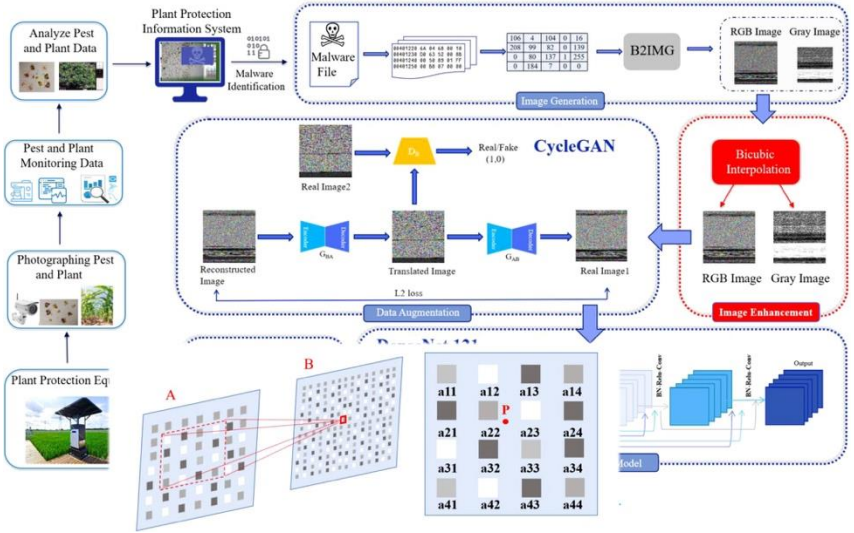
Reverse-engineering polymorphic AI malware

DeepCode AI is an advanced AI-powered code analysis tool that helps developers write cleaner, more secure, and efficient code. Here's an overview in bullet points:

- **AI-Driven Code Analysis**: Utilises advanced machine learning algorithms, particularly trained on a vast dataset of open-source code, to identify potential bugs, vulnerabilities, and inefficiencies in real-time.
- **Real-Time Feedback**: Provides developers with instant suggestions and fixes as they write code, integrating seamlessly with popular IDEs like VS Code, IntelliJ, and others.
- **Focus on Security and Quality**: Detects critical security vulnerabilities, code smells, and technical debt, ensuring robust and maintainable codebases.
- **Supports Multiple Languages**: Works across a wide range of programming languages, including Python, Java, JavaScript, TypeScript, and others.
- **Context-Aware Recommendations**: Offers precise and context-specific recommendations, using AI to understand the intent behind the code.
- **Collaboration and Integration**: Designed for teams, with integrations into GitHub, GitLab, and Bitbucket, to ensure consistent code quality and secure pull request workflows.
- **Continuously Updated Knowledge Base**: Learns from new patterns in the open-source community, adapting to emerging coding practices and threats.

DeepCode AI was acquired by Snyk in 2020 and is now integrated into Snyk's developer-first security platform, enhancing its capabilities in secure coding.

Advanced Static Malware Analysis



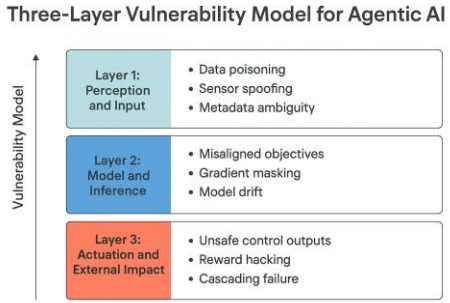
Polymorphic malware analysis

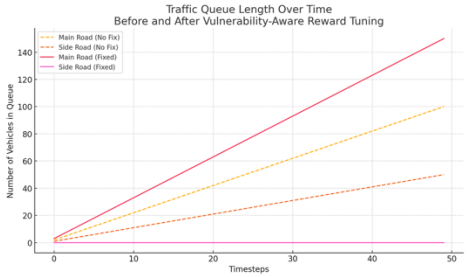
- Identifying encoded payloads and obfuscation methods

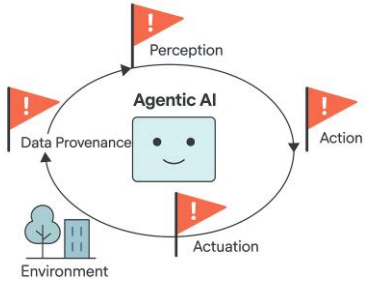
Vulnerability Analysis

Risk Propagation in Autonomous AI Ecosystems

- What is Vulnerability Analysis?
- A Three-Layer Vulnerability Model for Agentic AI
- Case Study: AI-Powered Traffic Management System








```python
import numpy as np
import matplotlib.pyplot as plt

# Simulate traffic queues for a simplified traffic management system

# Parameters
timesteps = 50
main_road_flow = 10  # 4 cars per timestep
side_road_flow = 3   # cars per timestep

# Without vulnerability-aware reward (ignores side road queues)
main_road_queue_no_fix = []
side_road_queue_no_fix = []

main_queue = 0
side_queue = 0

for t in range(timesteps):
    main_queue = max(0, main_queue + main_road_flow - 8)  # 8 biased green light
    side_queue = max(0, side_queue + side_road_flow - 2)  # 2 insufficient time for side road
    main_road_queue_no_fix.append(main_queue)
    side_road_queue_no_fix.append(side_queue)

# With vulnerability-aware reward (balanced queue management)
main_road_queue_fixed = []
side_road_queue_fixed = []

main_queue = 0
side_queue = 0

for t in range(timesteps):
    main_clear = 7
    side_clear = 5
    main_queue = max(0, main_queue + main_road_flow - main_clear)
    side_queue = max(0, side_queue + side_road_flow - side_clear)
    main_road_queue_fixed.append(main_queue)
    side_road_queue_fixed.append(side_queue)

# Plotting the results
plt.figure(figsize=(10, 6))

plt.plot(main_road_queue_no_fix, label="Main Road (No Fix)", linestyle='--')
plt.plot(side_road_queue_no_fix, label="Side Road (No Fix)", linestyle='--')

plt.plot(main_road_queue_fixed, label="Main Road (Fixed)")
plt.plot(side_road_queue_fixed, label="Side Road (Fixed)")

plt.title("Traffic Queue Length Over Time\nBefore and After Vulnerability-Aware Reward Tuning")
plt.xlabel("Timesteps")
plt.ylabel("Number of Vehicles in Queue")
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()
```

Theoretical - Static Malware Analysis:

Analysis Involves analysing malware code without executing it, focusing on binary files or disassembled code.

1. Recent Tools:
    1. **Ghidra**: An open-source reverse engineering tool developed by the US National Security Agency (NSA), widely used for decompiling malware to study its code.
    2. **Capa** (by FireEye): A static analysis tool that automatically identifies capabilities in malware samples (e.g., encryption, network communication).
    3. **BinaryNinja**: A reverse-engineering platform designed for analysing malware binaries with user-friendly features and scripting support.
2. Recent Example:
    a. Static Analysis of **TrickBot**: Researchers used Ghidra and IDA Pro to reverse engineer TrickBot's encrypted configuration files, allowing them to predict future attacks and prevent them.

AI-enhanced static analysis

AI-enhanced static analysis uses machine learning models to detect patterns in code that might indicate malicious intent, even when malware is obfuscated.

Recent AI-based Tools:

1. **DeepCode AI**: A tool using AI to analyse malware binaries and source code, automatically flagging suspicious functions based on known malware patterns and novel heuristics.
2. **Ghidra** 11.1.2: The latest version of Ghidra with enhanced support for malware binaries targeting ARM64 architectures, commonly found in mobile and IoT devices.

Recent Example:

a. **Static Analysis of BlackMamba**: AI-based tools like DeepCode AI identifies suspicious payload structures in the polymorphic code, which had evaded signature-based detection systems.

Exercises/Activities: BlackMamba (10 minutes): Applying advanced static analysis techniques

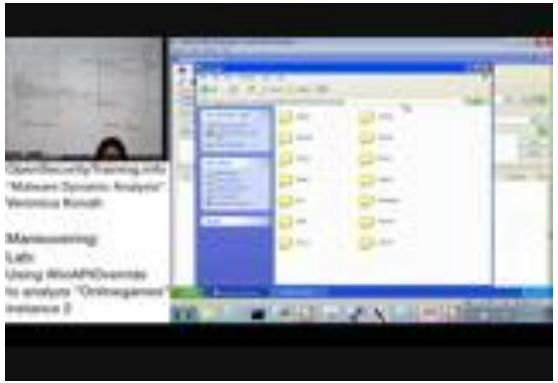
Exercise 1: Case Study of BlackMamba

- Uses AI-generated code variations to bypass static detection.
- Polymorphic behaviours make it challenging to isolate patterns.
- Analysis Takeaway:
- Importance of combining AI-driven tools with human oversight.

Exercise 2: Observing and Analysing Code

- Hands-On Activity:
- Use **DeepCode AI** to reverse-engineer the malware's polymorphic logic.
- Identify weaknesses and propose mitigation strategies.
- Deliverables:
- A vulnerability report documenting the findings.

Definitions



1. **Definition for Dynamic Malware Analysis**: Observing malware behaviour during execution in a controlled environment.
2. **Techniques and Tools**: Sandboxing: Using environments like Cuckoo Sandbox to monitor behaviour.
3. **Debugging**: Using tools like OllyDbg to step through malware execution.
4. **System monitoring**: Analysing changes in files, network activity, and system calls.
5. **Advantages and Limitations**: Dynamic analysis provides a clearer understanding of behaviour but requires careful environment setup.

Introduction to sandboxing tools, including Cuckoo Sandbox and Vectra AI

- Understanding Dynamic Analysis (e.g., Cuckoo Sandbox, Vectra AI)
- Setting up and observing the runtime behaviours that mimic legitimate processes to bypass monitoring.
- Dynamic analysis of REvil: Analysing REvil with Cuckoo Sandbox - observe real-time malware behaviour.

Dynamic Malware Analysis

Executing the malware in a controlled, isolated environment to observe its behaviour, such as file changes, network activity, and system modifications.

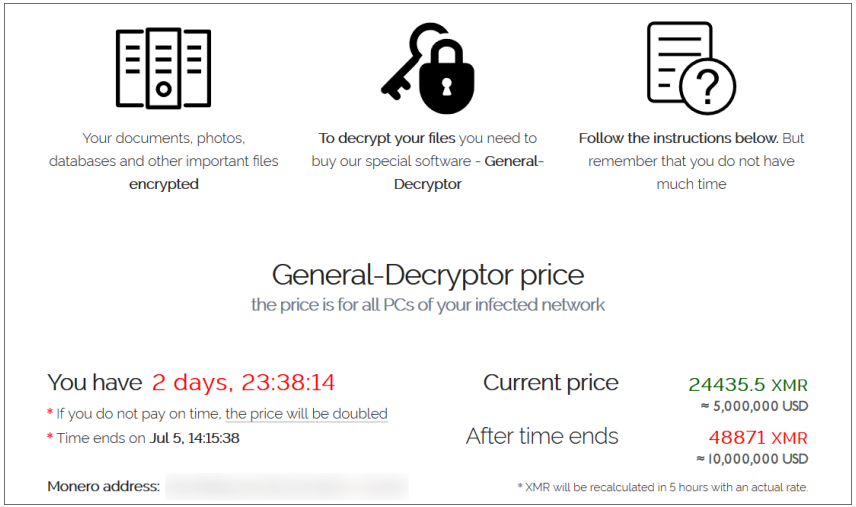
Tools:

1. **Cuckoo Sandbox:** A leading tool that allows the automated analysis of suspicious files in a virtualised environment, logging all activities such as network communication, system changes, and file manipulation.
2. **Process Monitor:** A system monitoring tool that tracks real-time file system, registry, and process/thread activity, useful for detecting malware actions like file creation and registry modifications.
3. **Wireshark:** A packet analyser that monitors network traffic and helps identify malicious communication from malware.

Real-world Example:

a. **Dynamic Analysis of Stuxnet:** Through dynamic analysis, researchers discovered that Stuxnet altered PLCs to control industrial machinery, particularly centrifuges in nuclear plants, which would not have been visible through static analysis alone.

Dynamic analysis of REvil: Using Cuckoo Sandbox to observe real-time malware behaviour.



- Execute **REvil ransomware** in a sandbox environment.
- Record API calls, registry changes, and network traffic patterns.
- Generate a behavioural profile of the malware, highlighting suspicious activities.

Advanced Dynamic Analysis

- Learn how AI malware dynamically adapts to evade detection (e.g., sandbox evasion).
- Behavioural Analysis: Analyzing advanced malware strains in cloud environments.
- Behavioural Analysis with Falcon X AI

Dynamic Malware Analysis

Running the malware in a controlled environment (sandbox) to observe behaviour like network communication and system changes.

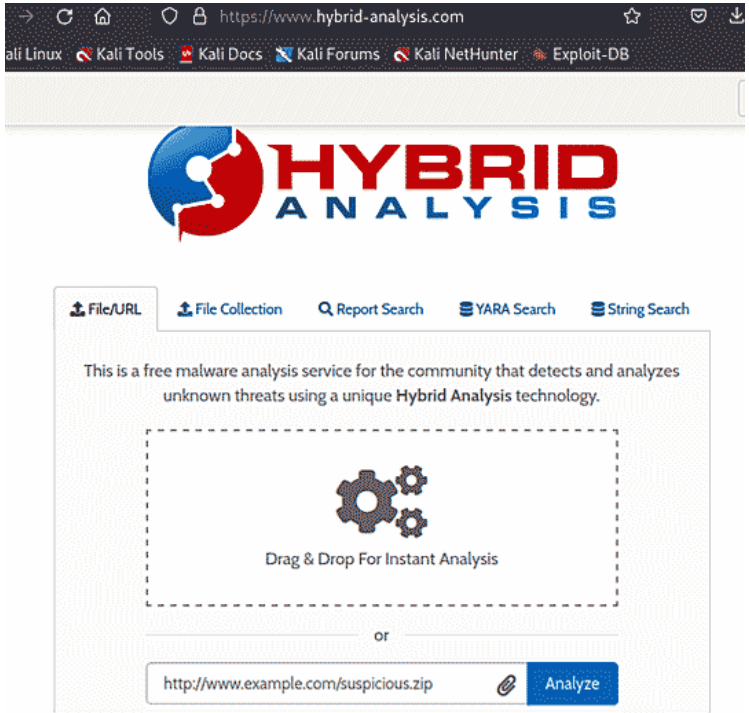
Recent Tools:

1. **Cape Sandbox**: An extension of the Cuckoo Sandbox that provides enhanced capabilities for behavioural analysis of malware, particularly focusing on ransomware and info stealers.
2. **Falcon Sandbox (CrowdStrike)**: A cloud-based malware sandbox that allows for safe execution of malware and detailed reporting of observed behaviours.
3. **APKiD**: A tool used for dynamic analysis of Android malware, detecting obfuscation techniques used by mobile malware.

Recent Example:

a. Dynamic Analysis of **REvil** Ransomware: By executing REvil in an isolated environment, analysts tracked the ransomware's network communications and observed the files it encrypted, enabling quicker response to new REvil strains after the 2021 Kaseya attack.

Real-World Case Studies



- Setting up and observing the runtime behaviours that mimic legitimate processes to bypass monitoring.
- Dynamic analysis of a WiperBot: Identifying patterns in AI-enhanced wiper malware.
- Using **Hybrid Analysis 2.0** and **Cape Sandbox 2024**, and Bayesian optimisation to improve threat detection
- Simulate and perform adversarial attacks on AI models

Dynamic Malware Analysis of AI-driven attacks

Running the malware in a controlled environment (sandbox) to observe behaviour like network communication and system changes.

Recent Tools (2024):

1. **Hybrid Analysis 2.0:** The latest version of the sandbox environment that now integrates AI to detect previously unknown malware behaviour and patterns.
2. **Cape Sandbox 2024:** Updated with advanced support for cloud environments, enabling analysis of malware that specifically targets cloud infrastructure.

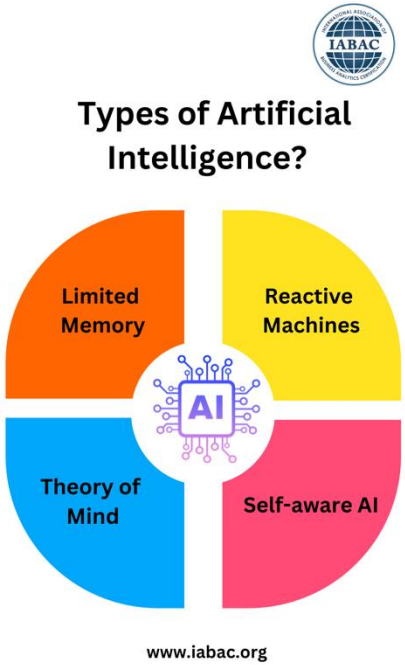
Recent Example:

1. **Dynamic Analysis of MedusaLocker:** Through Cape Sandbox, researchers discovered that the ransomware had adapted to exploit vulnerabilities in healthcare systems' virtual private networks (VPNs) to propagate within hospitals, showing increased sophistication in its behaviour.

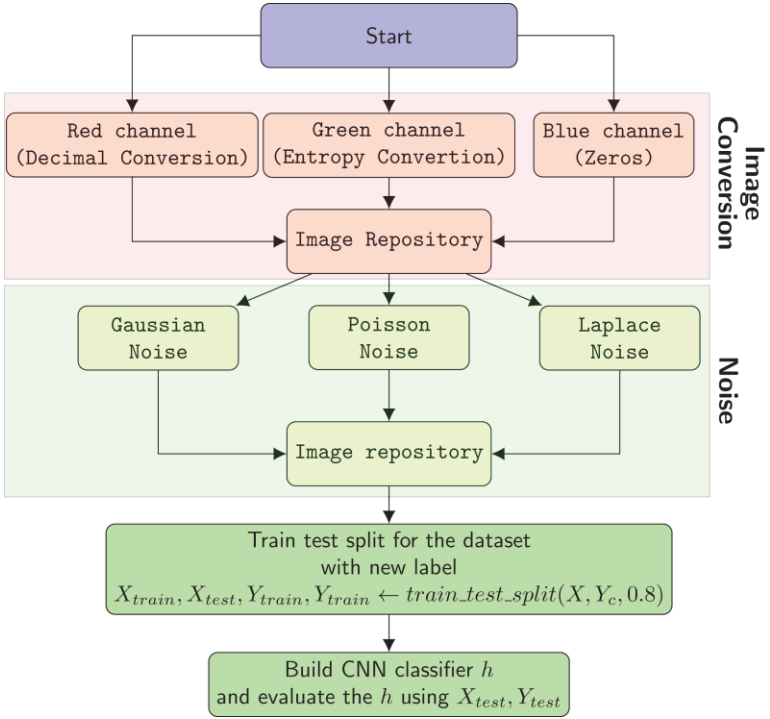
Identifying patterns in AI-enhanced wiper malware.

AI-based dynamic analysis focuses on observing malware behaviour in a sandbox, using AI to detect subtle and complex patterns of malicious behaviour.

- **Vectra AI:** Uses AI and machine learning to detect real-time malicious behaviour by analysing network traffic, endpoint activity, and file system changes. Vectra focuses on identifying deviations in normal system behaviour indicative of malware activity.
- **Falcon X AI (CrowdStrike):** Employs AI models to enhance malware analysis by observing the behaviour of files in a controlled environment and correlating it with known attack patterns and anomalies.



Defending from AI-Based Malware Threats



- Understanding the Key Components for Test Framework Development (e.g., Library of Target AI Models , Training Datasets, Automation Tools)
- Deploying countermeasures: Using AI-based detection tools to mitigate ransomware attacks.
- Evaluate AI models under adversarial scenarios
- Develop AI-enhanced threat detection systems
- Implement and deploy AI-driven countermeasures

Advanced techniques for defending against AI-enhanced threats.

Key Components for Test Framework Development (e.g., Library of Target AI Models , Training Datasets, Automation Tools)

- Define Objectives and Scope
  - Purpose
  - Model Scope
  - Security Metrics
- Select or Build AI Models
  - Choose representative models - GPT, ResNet, BERT
  - Pre-train models
  - Include variations

Curate Training and Test Datasets

- Normal Datasets
- Malicious Inputs

Develop Framework Components

- Environment Setup
- Automation Tools
- Threat Models

Implement Logging and Analysis

Test the Framework

Document and Iteratively Improve

Exercise/Activity: Deploying countermeasures

Using AI-based detection tools to mitigate ransomware attacks. AI Threat Detection using Bayesian Optimization

Objective: Detect adversarial inputs, anomalies, or security vulnerabilities in black-box AI systems by treating the detection process as a black-box optimization problem.

Key Steps: Define Input Space: Perturbations, synthetic inputs, or latent features.

Objective Function: Measure adversarial success, anomaly likelihood, or misclassification probability.

- Bayesian Optimization Loop:
  - Surrogate Model: Gaussian Process (GP) or alternatives.
  - Acquisition Function: Guides exploration vs. exploitation (e.g., EI, UCB).
  - Evaluate & Update: Query the AI, log results, and refine the model.

Workflow:

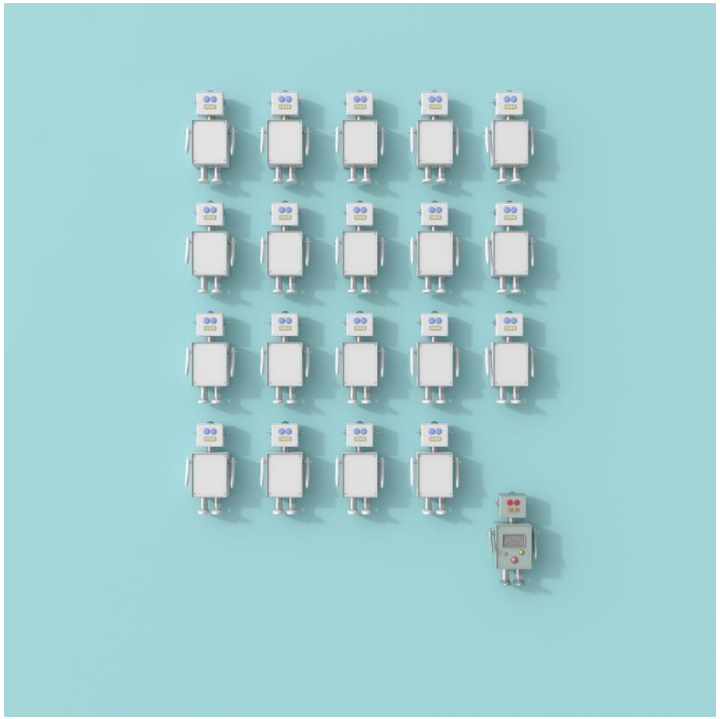
- Step 1: Sample initial inputs.
- Step 2: Query the AI model to evaluate the objective function.
- Step 3: Use BO to find next promising inputs.
- Step 4: Iterate until convergence or evaluation limit.

Advantages:

- Efficient threat detection with fewer evaluations.
- No need for internal model access.
- Adaptable to adversarial, anomaly, or fairness analysis.

Outcome: A systematic approach to probe black-box AI systems for vulnerabilities and adversarial behaviours while minimising computational cost.

Security Assessment Frameworks for AI Agents



1. MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems)
2. NIST AI RMF 1.0 + RMF Generative AI Profile (Draft 2024/25)
3. OECD AI Incidents Tracker + AI Risk Classification Framework
4. ISO/IEC 42001:2023 (AI Management System Standard) + 23894:2023 (AI Risk Management)
5. CIFER – Center for AI Safety Evaluation Framework for RL Agents
6. Anthropic's Constitutional AI Red Teaming Framework (2024)
7. UK AISI – AI Security Incident Taxonomy and Assessment Framework

Alternatives for AI-specific malware analysis

Static Analysis Tools:
Alternatives and Complements

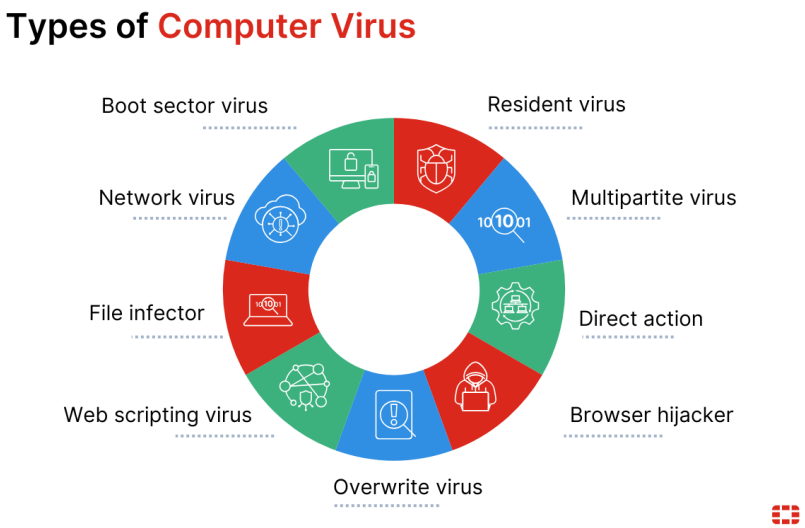
Currently Used:

- Ghidra
- IDA Pro
- Binwalk
- PEiD
- Capa (by FireEye)
- BinaryNinja
- DeepCode AI (Snyk)

AI-Focused Static Analysis Alternatives

Tool	Description	Relevance
CleverHans	TensorFlow/PyTorch library for crafting and detecting adversarial examples.	Enables static analysis of ML models against evasion attacks.
IBM ART (Adversarial Robustness Toolbox)	Comprehensive library for adversarial attack/defence evaluation.	Essential for testing model robustness pre-deployment.
Microsoft Counterfit	Automated tool for security testing of AI systems.	Can be used to simulate polymorphic behaviour and probe static defences.
Hopper Disassembler	Lightweight alternative to IDA Pro.	Faster analysis for AI-generated binary payloads.
AI Explainability 360 (IBM)	Explains predictions of black-box ML models.	Useful to audit suspicious model behaviours revealed during static analysis.

Dynamic Analysis and Sandboxing Tools



Currently Used:

- Cuckoo Sandbox
- Vectra AI
- Falcon X AI (CrowdStrike)
- Hybrid Analysis 2.0
- OllyDbg
- Process Monitor
- Wireshark
- Cape Sandbox

AI-Focused Dynamic Analysis Alternatives

Tool	Description	Relevance
SecML	Python framework for adversarial robustness and evasion analysis.	Can simulate evasion behaviour and test detection systems.
AdversariaLib	Older but still useful library for adversarial ML.	Supports attack simulation during runtime.
ML-Sec Sandbox (prototype tools)	Emerging field tools using AI to monitor model behaviours in sandboxed environments.	Use to test model-level behaviours rather than just executable behaviour.
Frida	Dynamic instrumentation toolkit.	Can be used for in-memory analysis of AI-enhanced malware.
Qiling Framework	Emulation and sandbox framework for dynamic binary analysis.	Powerful for emulating AI malware behaviours across platforms.

AI Model Behavioural Analysis: Static + Dynamic

Neural network payloads / embedded AI models within binaries

Tool	Description	Use Case
TF-Explain or Captum (PyTorch)	Visualises layer-by-layer inference behaviour.	Helps analyse embedded models that perform evasive tasks.
Foolbox	Advanced adversarial testing tool for ML models.	Use to test AI model's decision boundaries post-reverse-engineering.
ModelCard++	Audits and creates metadata about AI models.	Use to generate provenance from extracted or reverse-engineered AI logic.

AI Malware Simulation and Red Teaming

Tool	Description	Value
Counterfit	Simulates attacks on AI systems.	Enables real-world adversarial testing and red teaming.
ART Metrics Suite	Part of IBM ART, offers extensive evaluation metrics.	Enables grading model robustness in malware contexts.
MITRE ATLAS	Adversarial threat landscape for AI.	Provides structured red teaming scenarios for AI-based threats.

Vulnerability Discovery and AI Pipeline Monitoring

Tool	Description	Relevance
Snorkel	Weak supervision for large-scale labelling.	Can be adapted to identify poisoned data or obfuscated behaviour patterns.
Great Expectations	Data validation in ML pipelines.	Can detect corrupted datasets and enforce schema constraints.
TruffleHog / GitLeaks	Secret detection in ML code.	Helps ensure AI codebases and models aren't leaking credentials or exfil data.

Threat Intelligence and Malware Attribution (AI-Specific)

Tool	Description	Relevance
AI Incident Tracker (OECD)	Monitors known AI misuse incidents.	Good for case study alignment and scenario building.
ThreatMatch (Darktrace)	AI-based threat attribution.	Can help students analyse AI-malware campaigns by TTP.
PolySwarm	Crowd-sourced threat intelligence engine using ML.	Dynamically learns from malware samples and improves detections.

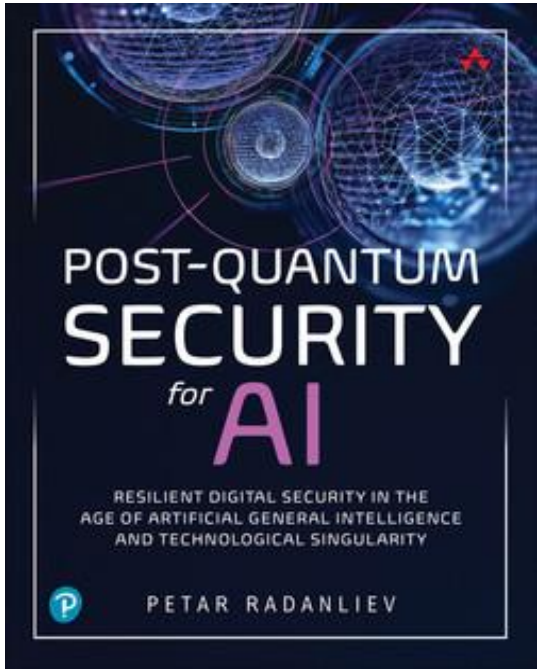
Recommended Replacements / Enhancements

Existing Tool	Type	Suggested Alternative or Enhancement	Why
Ghidra / IDA Pro	Static Reverse Engineering	CleverHans, ART, Counterfit	Better for AI-specific adversarial behaviour.
Cuckoo Sandbox	Dynamic Analysis	Cape, Frida, Qiling	More advanced AI evasion/resilience analysis.
DeepCode AI	Static Code Scanning	Snyk AI + Counterfit	Needs adversarial testing integration.
Falcon X AI	Behavioural AI Sandbox	Vectra AI, PolySwarm, MITRE ATLAS	Stronger on threat intelligence and simulation.
Process Monitor / Wireshark	System Monitoring	Frida, Qiling, Vectra	Needed for runtime AI-agent tracing.
Hybrid Analysis	Behavioural Sandbox	Cape + ART + Foolbox	For richer analysis of AI polymorphism and evasion.

Visual curriculum map - aligning AI-specific tools

Course Segment	Existing Tool(s)	AI-Specific Tools	Rationale for Inclusion
Static Analysis with Ghidra	Ghidra, IDA Pro	CleverHans, IBM ART, Microsoft Counterfit	AI-focused adversarial robustness testing
Advanced Static Analysis (Capa, Binwalk)	Capa, Binwalk, PEiD, BinaryNinja	Foolbox, ART Metrics, SecML	Model-level and obfuscation-aware static analysis
Static Code Review with DeepCode AI	DeepCode AI	Snyk AI + Microsoft Counterfit, AI Explainability 360	Context-aware vulnerability scanning in ML pipelines
Dynamic Analysis with Cuckoo Sandbox	Cuckoo Sandbox, Vectra AI	Cape Sandbox, Frida, Qiling Framework	Advanced evasion detection and AI agent simulation
Behavioural Analysis with Falcon X AI	Falcon X AI (CrowdStrike)	Vectra AI, PolySwarm, ThreatMatch (Darktrace)	Threat intelligence and evasion monitoring via AI
System Monitoring with Wireshark/ProcMon	Wireshark, Process Monitor, OllyDbg	Frida, Qiling, Vectra AI (for runtime agent tracing)	In-memory dynamic instrumentation of AI logic
Real-World Case Studies (e.g., BlackMamba)	Hybrid Analysis 2.0, Cape Sandbox	Snorkel, Great Expectations, OECD AI Tracker	Enhanced provenance and threat attribution for AI malware
Defence & Mitigation (AI Countermeasures)	Bayesian Optimisation, MITRE ATLAS	IBM ART, MITRE ATLAS, Anthropic Constitutional AI	Red teaming and defence aligned with AI risk frameworks

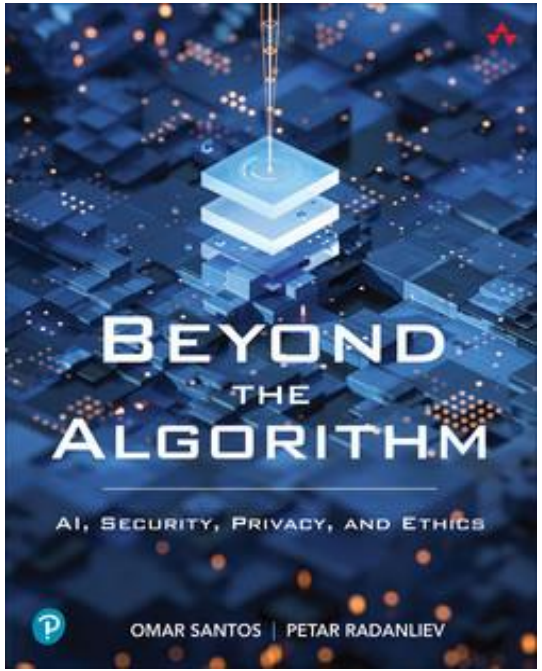
Background reading – Book on AGI



Post-Quantum Security for AI: Resilient Digital Security in the Age of Artificial General Intelligence and Technological Singularity

https://www.oreilly.com/library/view/post-quantum-security-for/9780135436004/

Background reading - Book



Beyond the Algorithm: AI, Security, Privacy, and Ethics

https://www.oreilly.com/library/view/beyond-the-algorithm/9780138268442/