COS30015 IT Security

Week 5

Presented by YICUN TIAN - YI

28 August 2024

Acknowledgement of Country

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne's Australian campuses are located in Melbourne's east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne's Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures, and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations.

Malware & Vulnerabilities

Agenda

Malware & Vulnerabilities, Assignment 1

Malware

- Types
- Analysis
- Detection

Vulnerabilities

- CVE
- Vulnerability Management

SWINBURNE UNIVERSITY OF TECHNOLOGY

Terms

Let's set some common language

Across both topics today, and from a cyber security perspective

- Malware: Malicious Software
- Goodware, benign software: Safe, normal program
- Vulnerability: A flaw within an organisation per its controls or system procedures, implementation, software or hardware which could be exploited
- Exploit: Software or code (mostly) that takes advantage of a vulnerability to gain unauthorised access, disclosure, privilege or perform malicious actions on a system (see compromise)
- Implant: Code which is often injected or inserted into a system to maintain command and control (C2), or perform malicious actions after compromise
- Compromise: unauthorised disclosure, modification, substitution or use of sensitive data, unauthorised access or modification to systems, devices or processes
- Payload: Malicious action or function of malware
- Sandbox
  - Usually an application we can run malware in to analyse it

Malware

History

What a time to be a live

History in the making

- 1980s and Onward:
  - Virus concept from 1949 lecture by John von Neumann (“self-reproducing automata”)
  - Modern viruses start with Elk Cloner (1982) on Apple II
  - Harmless, but spread to all attached disks—first major outbreak
- 1990s:
  - Windows OS popularity increases
  - More viruses written for Windows platform
  - Macro viruses in Microsoft Word
- 2002-2007:
  - IM worms on AOL AIM, MSN, Yahoo Messenger
  - Social engineering lures—malicious IMs
  - IM worm spreads through contact lists
- 2005-2009:
  - Adware growth—unwanted ads on screens
  - Exploited legit software, led to lawsuits
  - Tech support scams have origins here
- 2007-2009:
- Malware targets Myspace, then Facebook, Twitter
- 2013:
  - Ransomware "CryptoLocker" targets Windows
  - Forced victims to pay $3M, spawned imitators
- 2013-2017:
  - Ransomware thrives via Trojans, malvertising
  - 2017 sees massive outbreaks impacting businesses
- 2017:
  - Cryptojacking emerges—secretly mining cryptocurrency on others' devices
- 2018-2019:
  - Ransomware resurgence, shifting to businesses
  - GandCrab, Ryuk ransomware spike in attacks
  - 365% increase in business attacks from 2018-2019

*what would we define good guys using malware, is it still malware?

https://www.malwarebytes.com/malware

Malware (cont.)

Often developed and used by cyber actors*

Malware

- Trojans and Backdoors:
  - Trojans appear useful, perform malicious actions when run
  - May steal info, download more malware, provide hacker access
- Ransomware:
  - Locks computer/files, demands payment for access
  - Extortion by malware, difficult to defend against
- Keyloggers:
  - Logs keystrokes, sends data to scammers
  - Captures passwords, bank info, credit card numbers
- Viruses and Worms:
  - Viruses infect files, run with file use
  - Worms spread between computers independently
  - Both can steal, download, delete, or spam
- Adware and Spyware:
  - Adware displays unwanted ads in browsers
  - Spyware secretly observes user activities
- Rootkit:
  - Provides attacker with admin privileges (root access)
  - Stays hidden from users, OS, and software
- Exploits and Zero-Day Exploits:
  - Exploits use system bugs to grant access
  - Zero-day exploits are unpatched vulnerabilities
- Malicious Cryptomining (Cryptojacking):
  - Trojans install, mine cryptocurrency using your resources
  - Attacker gains coins, not you—resource theft
- Webshell
  - Software used on compromised web servers

SWINBURNE UNIVERSITY OF TECHNOLOGY

Malware Delivery Vectors

How does malware even get onto our systems?

Not an exhaustive list:

- Phishing Emails
- Spear Phishing
- Malvertising
- Drive-By Downloads
- Watering Hole Attacks
- Infected USB Drives
- Social Engineering
- Ransomware Payloads
- Exploit Kits
- Remote Desktop Protocol (RDP) Attacks
- Brute Force Attacks
- Social Media Attacks
- Instant Messaging and Chat
- Fake Software Updates
- Trojanized Applications
- Drive-by App Downloads
- Email Attachments
- Peer-to-Peer Networks
- Physical Media
- Wi-Fi Networks

Preventative Measures

Being cyber safe

Some thoughts

- Preventive Measures:
  - Use antivirus software with daily updates
  - Keep all software updated
  - Employ strong passwords/passphrases
  - Backup files daily
  - Disable unused Microsoft Office macros
  - Regularly review and uninstall unused software

- Secure Application Installation:
  - Malware distributed via spam emails, malicious websites, and fake applications
  - Use reputable app stores for downloads
  - Avoid third-party download sites
  - Don't click on online ads for downloads, use ad-blockers
  - Avoid peer-to-peer network downloads
  - Be cautious with email/instant message links or attachments
  - Scan applications before installing, especially from email/USB

SWINBURNE UNIVERSITY OF TECHNOLOGY

Malware Analysis

Three main types

Automated, Static, Dynamic

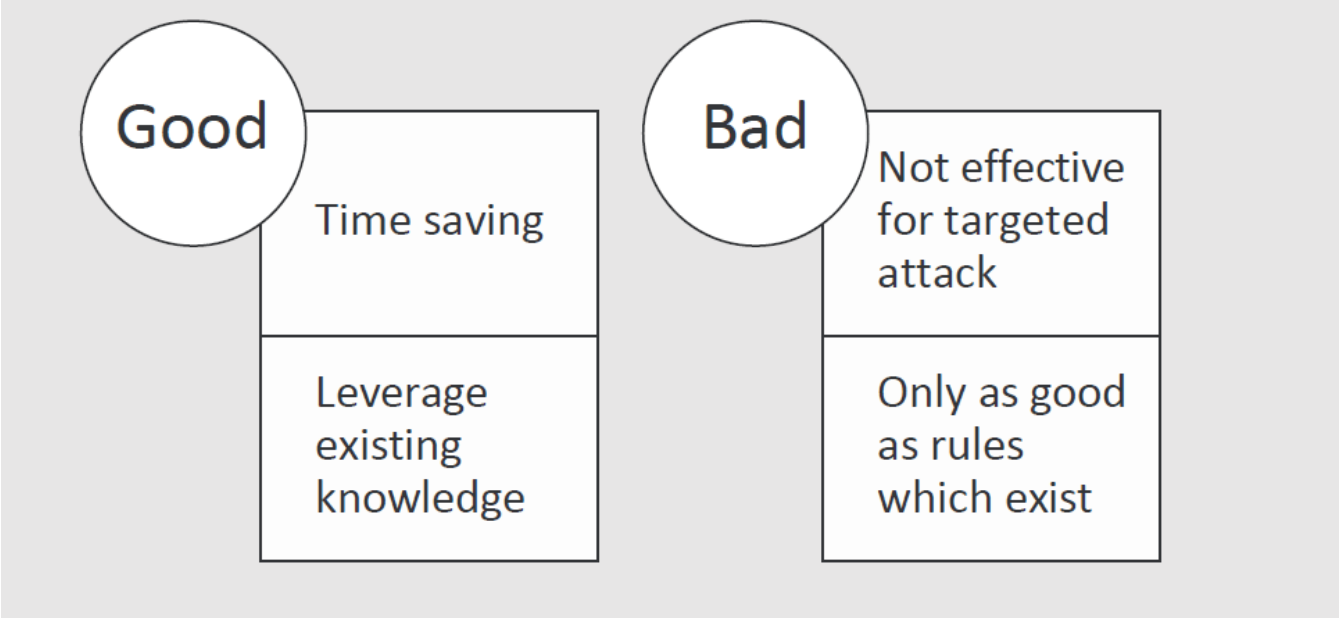
- Automated
  - Use tools to analyse it
- Static
  - Dump the contents and investigate
- Dynamic
  - Run it, and investigate

*what would we define good guys using malware, is it still malware?

Automated Analysis

Use of software tools and algorithms to <u>automatically</u> examine and analyse malicious software

Initial information source



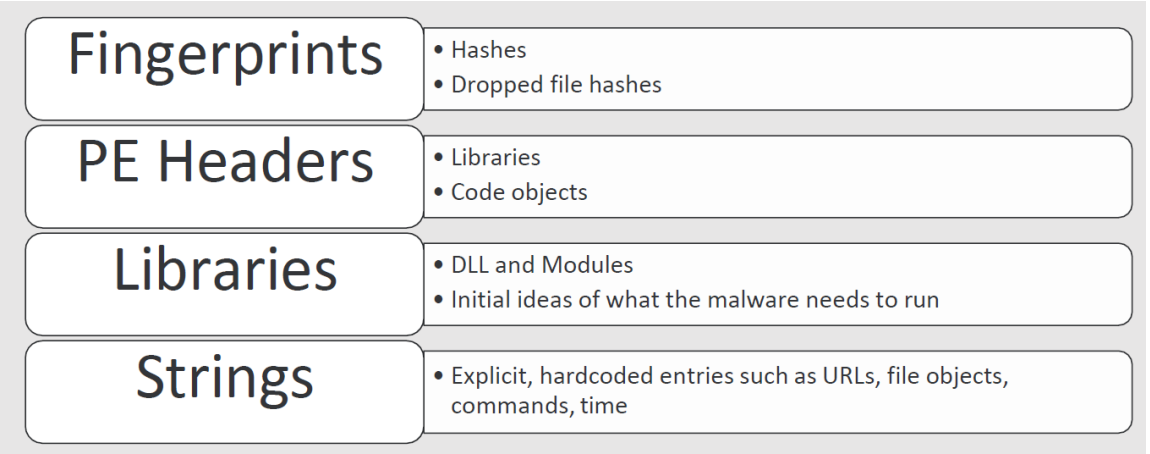
Static Analysis

A technique used to analyse and understand the behaviour and functionality of malicious software, without actually executing it.

Two steps

- Malware sample is first disassembled or decompiled into its component parts, allowing analysts to examine its code structure and functions.
- Analysts then examine the code for known patterns of malicious behavior, such as calls to system APIs, use of encryption or obfuscation techniques, and the presence of known exploit payloads.

Tools include IDA Pro, OllyDbg, and Ghidra



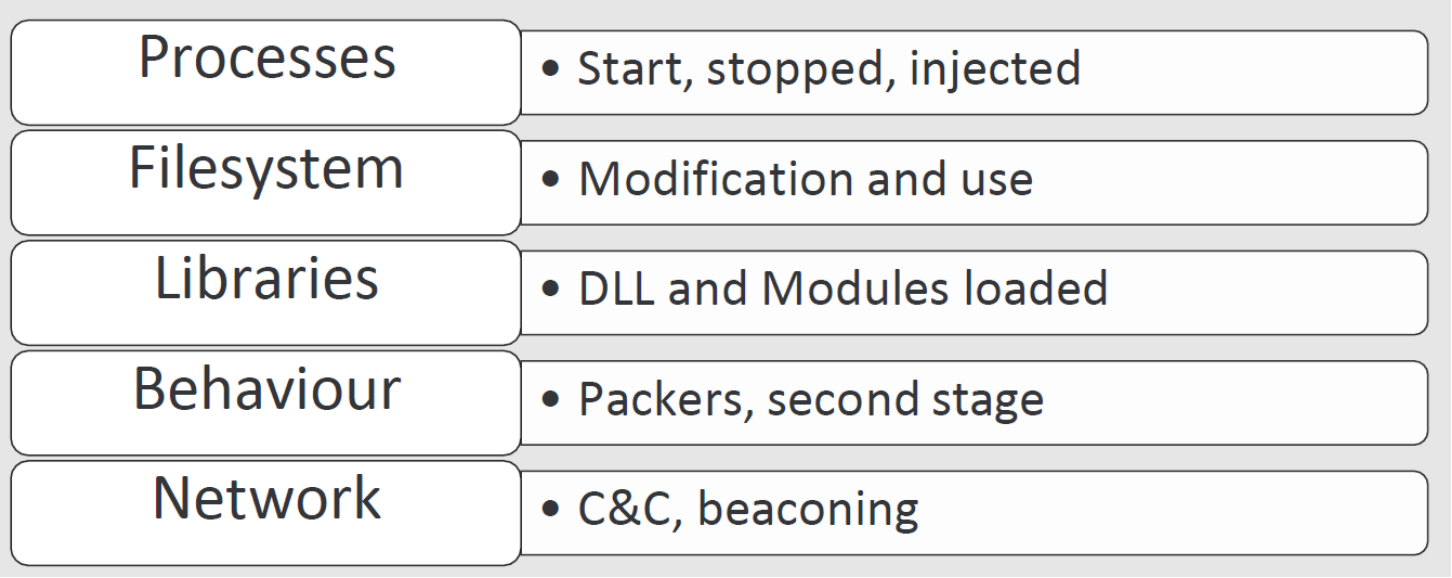
Dynamic Analysis

Analyse and understand the behaviour and functionality of malicious software by executing it in a controlled environment.

Running the malware sample in a sandboxed environment, a virtual machine, container or specialised tools

Used to identify previously unknown malware variants

Cuckoo, Cape v2, Joes Sandbox



SWINBURNE UNIVERSITY OF TECHNOLOGY

Vulnerabilities

Understanding Vulnerabilities

The good and the bad

Vulnerabilities

- Vulnerabilities are **weaknesses** in software or systems that can be **exploited** by attackers
- They can **lead to** unauthorised access, data breaches, or system crashes
- Vulnerabilities can be **caused by** coding errors, design flaws, or configuration issues

- **CVEs** (Common Vulnerabilities and Exposures) are standardised identifiers for known vulnerabilities
- **Vulnerability databases** like CVE Details and NVD list and provide information about CVEs

- **Vulnerability scanning tools** help identify weaknesses in systems and software
- **Patches** are updates released by software vendors to fix vulnerabilities
- Organisations should **regularly update software and systems** to protect against known vulnerabilities
- **Zero-day vulnerabilities** are exploited by attackers before vendors release patches
- Threat actors **actively** exploit unpatched vulnerabilities, making timely updates crucial

CVEs

What, types and how does something become a known vulnerability

The flow

- CVEs are standardised identifiers for publicly known cybersecurity vulnerabilities
- Assigned to vulnerabilities by the MITRE Corporation to aid in tracking and information sharing
- Importance of CVEs
  - CVEs provide a common language for discussing vulnerabilities across the industry
  - They help security professionals and vendors understand the nature and severity of vulnerabilities
- Types of CVEs
  - Buffer Overflow: Occurs when a program writes more data into a buffer than it can hold, potentially allowing attackers to execute malicious code
  - SQL Injection: Attackers inject malicious SQL queries into input fields to manipulate databases
  - Cross-Site Scripting (XSS): Malicious scripts are injected into web pages viewed by others, compromising user data
  - Denial of Service (DoS): Attackers flood a system to overload it, causing it to crash or become unresponsive
  - Privilege Escalation: Attackers exploit vulnerabilities to gain unauthorised access to higher levels of system privileges
  - Remote Code Execution (RCE): Allows attackers to execute code remotely, taking control of systems
  - Authentication Bypass: Exploits that let attackers circumvent authentication measures
  - Information Disclosure: Vulnerabilities that expose sensitive data
  - Man-in-the-Middle (MitM): Attackers intercept and manipulate communications between parties
  - Zero-Day: Exploited vulnerabilities before they are publicly known, leaving no time for mitigation
- Lifecycle of a CVE
  - Discovery: Researchers or attackers identify a vulnerability
  - Report: Vulnerability details are reported to the affected vendor or project
  - Mitigation: Vendor develops and releases patches or updates to fix the vulnerability
  - CVE Assignment: MITRE assigns a CVE identifier
  - Public Disclosure: Vulnerability details are published, helping users take action

Zero Day

A special type of Vulnerability

Characteristics

- The term "zero-day" refers to the fact that developers **have zero days to address and fix the vulnerability before it's exploited**
- A zero-day vulnerability is a **security flaw** in software or systems that is exploited by attackers before the vendor becomes aware of it
- These vulnerabilities are **highly valuable to attackers**, as they can **target users who are unaware of the issue**
- Zero-day exploits can **lead to** unauthorised access, data breaches, and other malicious activities
- Attackers can **sell** zero-day exploits on the **black market** or use them for targeted attacks
- The discovery of zero-days may occur through independent research or by malicious actors
- Mitigation involves using **intrusion detection systems, regularly updating software, and employing strong security practices**
- **Software vendors release patches as soon as** the vulnerability is identified to minimise the window of opportunity for attacks

SWINBURNE UNIVERSITY OF TECHNOLOGY

Vulnerability

Weaknesses or flaws in a computer system, network, application, or device

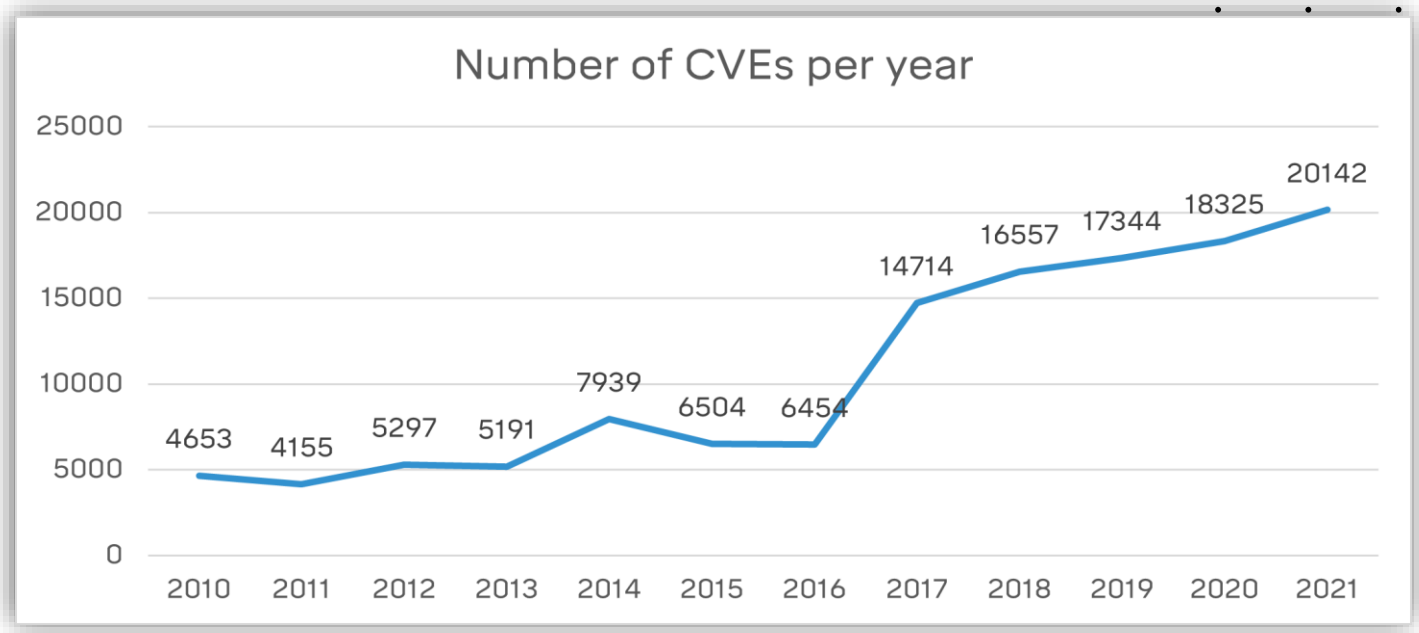
Exploited by attackers to gain unauthorised access, steal sensitive information

Threat

- The hypothetical event wherein an attacker uses the vulnerability.
- malware, phishing attacks, social engineering

Risks

- The likelihood and impact of harm or damage caused by a threat.
- Risks management : risk assessment, risk mitigation, risk avoidance.
- financial losses, reputational damage, loss of confidential information



https://www.cvedetails.com/browse-by-date.php

SWINBURNE UNIVERSITY OF TECHNOLOGY

Vulnerability Types

Vulnerabilities can exist in any type of software

Impact

- gain unauthorized access
- steal data
- cause other harm

Types

- Web application vulnerabilities: OWASP Top Ten.
- System vulnerabilities: buffer overflows, privilege escalation, insufficient input validation.
- Application vulnerabilities : insecure data storage, weak authentication mechanisms, insecure communications.

CVSS Score

A framework to assign a severity score to software vulnerabilities based on the potential impact and likelihood of exploitation

To help prioritize vulnerability remediation efforts, with higher-scored vulnerabilities typically given higher priority for mitigation

CVSS v2.0 Ratings	
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

CVSS v3.0 Ratings	
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

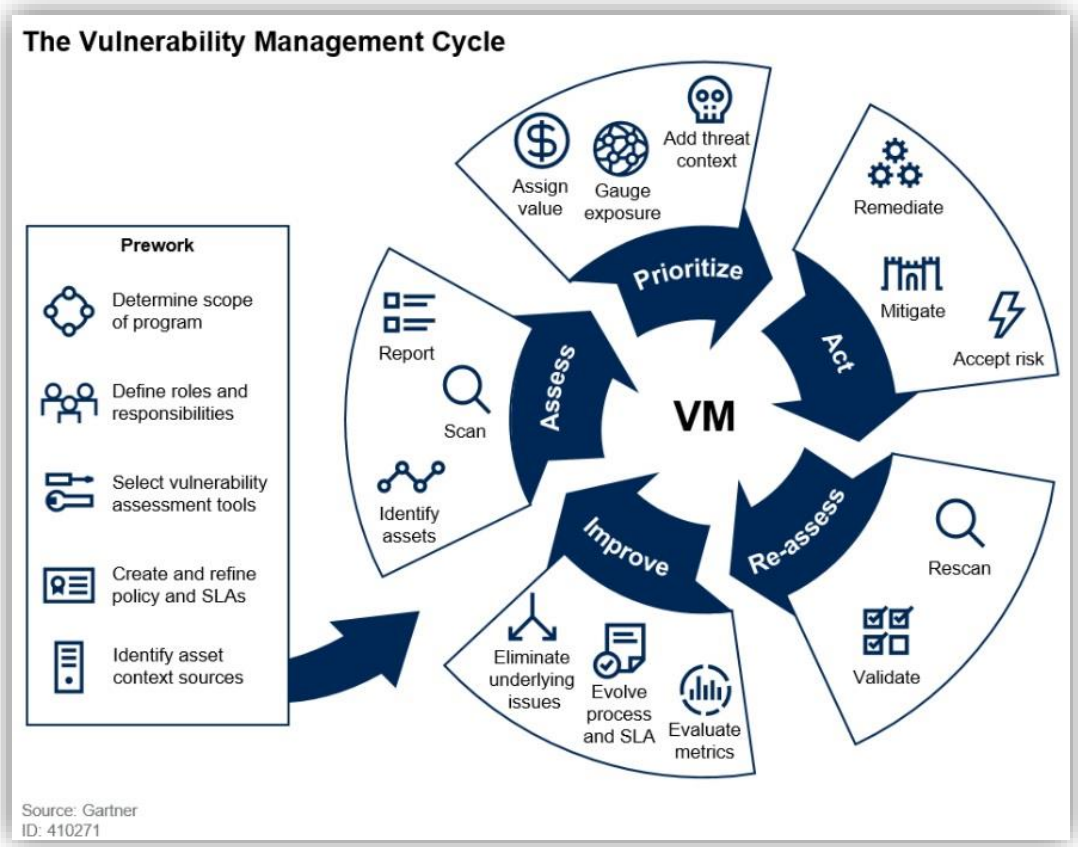
CVE-2022-45608	An issue was discovered in ThingsBoard 3.4.1, allows low privileged attackers (CUSTOMER_USER) to gain escalated privileges (vertically) and become an Administrator (TENANT_ADMIN) or (SYS_ADMIN) on the web application. It is important to note that in order to accomplish this, the attacker must know the corresponding API's parameter (authority : value). **Published:** 三月 01, 2023; 11:15:09 上午 -0500	*V3.1:* 8.8 HIGH *V2.0:*(not available)
CVE-2023-26281	IBM HTTP Server 8.5 used by IBM WebSphere Application Server could allow a remote user to cause a denial of service using a specially crafted URL. IBM X-Force ID: 248296. **Published:** 三月 01, 2023; 3:15:14 上午 -0500	*V3.1:* 7.5 HIGH *V2.0:*(not available)
CVE-2023-26039	ZoneMinder is a free, open source Closed-circuit television software application for Linux which supports IP, USB and Analog cameras. Versions prior to 1.36.33 and 1.37.33 contain an OS Command Injection via daemonControl() in (/web/api/app/Controller/HostController.php). Any authenticated user can construct an api command to execute any shell command as the web user. This issue is patched in versions 1.36.33 and 1.37.33. **Published:** 二月 24, 2023; 9:15:13 下午 -0500	*V3.1:* 8.8 HIGH *V2.0:*(not available)
CVE-2023-26038	ZoneMinder is a free, open source Closed-circuit television software application for Linux which supports IP, USB and Analog cameras. Versions prior to 1.36.33 and 1.37.33 contain a Local File Inclusion (Untrusted Search Path) vulnerability via web/ajax/modal.php, where an arbitrary php file path can be passed in the request and loaded. This issue is patched in versions 1.36.33 and 1.37.33. **Published:** 二月 24, 2023; 9:15:13 下午 -0500	*V3.1:* 6.5 MEDIUM *V2.0:*(not available)
CVE-2023-26036	ZoneMinder is a free, open source Closed-circuit television software application for Linux which supports IP, USB and Analog cameras. Versions prior to 1.36.33 and 1.37.33 contain a Local File Inclusion (Untrusted Search Path) vulnerability via /web/index.php. By controlling $view, any local file ending in .php can be executed. This is supposed to be mitigated by calling detaintPath, however dentaintPath does not properly sandbox the path. This can be exploited by constructing paths like "..././", which get replaced by "../". This issue is patched in versions 1.36.33 and 1.37.33.	*V3.1:* 9.8 CRITICAL *V2.0:*(not available)

https://nvd.nist.gov/

Vulnerabilities Management

The process of identifying, prioritizing, and addressing vulnerabilities in software, hardware, and other IT assets

- Identification: Identifying potential vulnerabilities in the system, such as security assessments, network scans, or vulnerability reports.
- Assessment: Evaluating the severity and impact of each vulnerability, such as risk analysis, threat modelling, or vulnerability testing.
- Prioritization: based on their severity and potential impact, such as through a risk rating system
- Remediation: Developing and implementing a plan to mitigate or eliminate vulnerabilities, such as software updates, security patches, or configuration changes.
- Monitoring: Continuously monitoring the system for new vulnerabilities and security threats and updating the vulnerability management plan as needed.



Detecting Vulnerabilities

Identifying security weaknesses in software, hardware, or other IT systems that could be exploited by attackers

- Automated Scanners: typically use predefined attack patterns to test
- Penetration Testing
- Code Review: involves examining the website's code, such as code injection, buffer overflows, or insecure API calls.
- Log Analysis: analysing the website's server, such as repeated failed login attempts or unusual traffic patterns.

Tools

- Nessus
- Nmap + Metasploit
- Qualys

Patching (Essential 8)

A set of mitigation strategies developed by the Australian Cyber Security Centre to help organizations protect against cyber threats.

Maturity Model levels (0-3)

Based on the ACSC's experience in producing cyber threat intelligence, responding to cyber security incidents, conducting penetration testing and assisting organisations to implement the Essential Eight.

https://www.cyber.gov.au/acsc/view-all-content/essential-eight

SWINBURNE UNIVERSITY OF TECHNOLOGY

Thank You