#MissionBeforeMoney

Class #12 – Chapter 21

Jake Smithula

CISSP® MENTOR PROGRAM – SESSION TWELVE

INTRODUCTION

Agenda –

- Welcome
- Introduction
- Malicious Code
- Malware evasion techniques
- Web-based threats
- Secure coding practices
- Review Quiz

CISSP® MENTOR PROGRAM – SESSION TWELVE

FRSECURE CISSP MENTOR PROGRAM LIVE STREAM

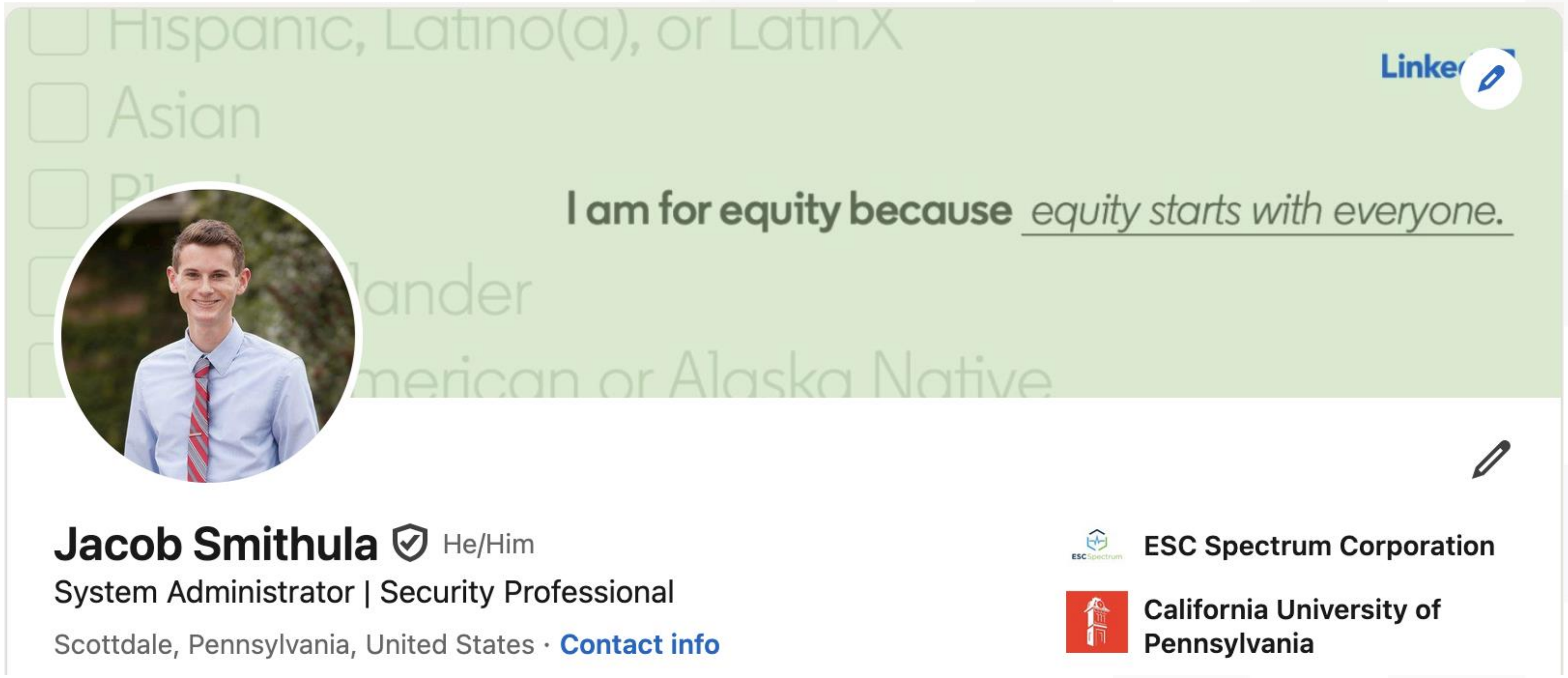
Quick housekeeping reminder.

- The online/live chat that's provided while live streaming on YouTube is for constructive, respectful, and relevant (about course content) discussion **ONLY**.
- At **NO TIME** is the online chat permitted to be used for disrespectful, offensive, obscene, indecent, or profane remarks or content.
- Please do not comment about controversial subjects, and please **NO DISCUSSION OF POLITICS OR RELIGION**.
- Failure to abide by the rules may result in disabling chat for you.
- **DO NOT share or post copywritten materials. (pdf of book)**

CISSP® MENTOR PROGRAM – LEAD MENTOR INTRO

WHOAMI

Jake Smithula



I am for equity because *equity starts with everyone.*

Jacob Smithula He/Him
System Administrator | Security Professional
Scottdale, Pennsylvania, United States · Contact info

ESC Spectrum Corporation
California University of Pennsylvania

https://www.linkedin.com/in/smithula/

Chapter 21: Malicious Code & Application Attacks

What is malicious code?

- Malicious code is software designed to cause harm, disruption, or unauthorized access
- Includes viruses, worms, trojans, spyware, ransomware, and more
- Can steal data, damage systems, spy on users, or hijack control
- Often hidden in normal-looking files or apps
- A major tool in cybercrime and espionage

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

Common Sources of Malicious Code

- Phishing emails with infected attachments or links
- Malicious websites and drive-by downloads
- Pirated software
- USB drives and other removable media
- Insider threats or misconfigured systems

Chapter 21: Malicious Code & Application Attacks

Script Kiddies and Advanced Threats

- Script Kiddies use premade tools, have little skill
- Advanced Persistent Threats (APTs) are stealthy, well funded and long term
- APTs often involve nation states or organized cybercrime groups
- Both can cause damage, intent and sophistication vary

Chapter 21: Malicious Code & Application Attacks

How Viruses Work

- A virus usually attaches to a host file or program
- Requires user action to activate, opening a file
- Can corrupt data, exfiltrate information or damage systems
- Often goes unnoticed until the damage is done

Chapter 21: Malicious Code & Application Attacks

Mast Boot Record (MBR) Viruses

- Infect the master boot record of a storage device
- Load before the operating system even starts
- Can control or intercept system-level operations
- Often hard to detect and remove
- Common in older attack methods, but still relevant in some systems

Chapter 21: Malicious Code & Application Attacks

File Infector Viruses

- Attach themselves to executable files (.exe, .dll, etc.)
- Activate when the infected file is run
- Can spread to other programs on the same system
- Often corrupts or replaces critical system files
- Some mutate to evade detection

Chapter 21: Malicious Code & Application Attacks

Macro Viruses

- Target documents with embedded scripting (e.g., Word, Excel)
- Use Visual Basic for Applications (VBA) to execute code
- Trigger when a document is opened
- Frequently used in phishing attacks
- Can download more malware or send data externally

Chapter 21: Malicious Code & Application Attacks

Service Injection Viruses

- Inject malicious code into legitimate system processes
- Run with the same privileges as the host service
- Harder to detect because they mimic trusted behavior
- Often used to maintain persistence or hide malware

Chapter 21: Malicious Code & Application Attacks

Multipartite Viruses

- Attack multiple system components: files, boot sector, macros
- Highly versatile and hard to remove completely
- Often re-infect cleaned systems through remaining components
- Combine features of file infectors, MBR viruses, and more

Chapter 21: Malicious Code & Application Attacks

Stealth Viruses

- Hide their presence by intercepting system calls
- Show clean versions of infected files or memory regions
- Avoid detection by antivirus and monitoring tools
- Often used in combination with other virus types

Chapter 21: Malicious Code & Application Attacks

Polymorphic Viruses

- Change their code structure each time they replicate
- Keep the same function, but look different to scanners
- Evade signature-based antivirus solutions
- Use mutation engines to automate transformation

Chapter 21: Malicious Code & Application Attacks

Encrypted Viruses

- Use encryption to hide their payload
- Contain a small decryption routine that runs first
- Payload is decrypted and executed in memory
- Harder to detect via static analysis or signatures

Chapter 21: Malicious Code & Application Attacks

Virus Hoaxes

- False alerts about non-existent viruses or threats
- Spread panic and waste time/resources
- Often include instructions that are themselves harmful
- Can act as social engineering pretexts

Chapter 21: Malicious Code & Application Attacks

Worms

- Self-replicating malware that spreads without user action
- Exploits network vulnerabilities to propagate
- Can consume bandwidth and system resources
- Often used as a delivery mechanism for other malware
- Famous examples: WannaCry, Blaster, Conficker

Chapter 21: Malicious Code & Application Attacks

Logic Bombs

- Malicious code that activates under specific conditions
- Often hidden in legitimate software or scripts
- Triggers: a date, system event, or lack of user activity
- Can delete data, crash systems, or open backdoors
- Common in insider attacks or sabotage

Chapter 21: Malicious Code & Application Attacks

Trojan Horses

- Appear to be useful or harmless software
- Contain hidden malicious functionality
- Do not self-replicate like viruses or worms
- Often used to install backdoors, steal data, or spy
- Common delivery method: fake software, game mods, pirated apps

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

Botnets

- Networks of infected devices (bots) under attacker control
- Controlled remotely via command and control (C2) servers
- Used for spam, DDoS attacks, crypto mining, and more
- Often spread via trojans or worms
- Devices may be compromised without user awareness

Chapter 21: Malicious Code & Application Attacks

Spyware and Adware

- Spyware covertly monitors user activity
- Adware displays unwanted ads, sometimes intrusive or malicious
- Often bundled with free or pirated software
- Can track browsing habits, log keystrokes, or steal credentials
- Slows system performance and compromises privacy

Chapter 21: Malicious Code & Application Attacks

Ransomware

- Encrypts files or systems, demanding payment for access
- Often spread through phishing or exploit kits
- Payment usually demanded in cryptocurrency
- Double extortion: threaten to leak data if unpaid
- A major threat to businesses, governments, and individuals

Chapter 21: Malicious Code & Application Attacks

Malicious Scripts

- Small programs embedded in websites, emails, or applications
- Often written in JavaScript, PowerShell, VBScript, or Bash
- Used for keylogging, redirection, or downloading malware
- Can exploit browser or OS vulnerabilities
- Common in drive-by download attacks

Chapter 21: Malicious Code & Application Attacks

Zero-Day Attacks

- Exploit unknown vulnerabilities with no available patch
- Called "zero-day" because defenders have zero days to prepare
- Highly valuable to attackers and black market buyers
- Often used by nation-states or APT groups
- Detection requires behavior analysis and threat intelligence

Chapter 21: Malicious Code & Application Attacks

Malware Prevention Strategies

- Regular software patching and updates
- Limit user privileges and enforce least privilege
- Educate users about phishing and suspicious behavior
- Use application whitelisting and execution control
- Monitor network traffic and system behavior for anomalies

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

Platforms Vulnerable to Malware

- Windows: most targeted due to popularity and legacy systems
- macOS: less targeted, but increasingly at risk
- Linux: favored for servers, targeted for privilege escalation
- Mobile: Android more exposed due to app side-loading
- IoT devices: weak security and limited update mechanisms

Chapter 21: Malicious Code & Application Attacks

Anti-Malware Software

- Detects, quarantines, and removes known malware
- Signature-based detection is fast but limited
- Heuristic and behavioral analysis spot new threats
- Endpoint Detection and Response (EDR) adds real-time monitoring
- Needs regular updates to remain effective

Chapter 21: Malicious Code & Application Attacks

File and System Integrity Monitoring

- Detects unauthorized changes to critical files and configurations
- Often used to monitor OS, registry, app binaries, and logs
- Can alert on malware that alters system settings
- Supports compliance and forensic analysis

Chapter 21: Malicious Code & Application Attacks

Advanced Threat Protection (ATP)

- Combines multiple tools to detect sophisticated threats
- Includes sandboxing, behavior analytics, threat intel, and AI
- Often cloud-based with centralized management
- Focuses on early detection and rapid response
- Essential for defending against APTs and zero-days

Chapter 21: Malicious Code & Application Attacks

Buffer Overflow Attacks

- Occur when a program writes more data than a buffer can hold
- Excess data spills into adjacent memory areas
- Can crash programs or allow code execution
- Common in low-level languages like C and C++
- Defense: bounds checking, memory-safe coding, ASLR, DEP

Chapter 21: Malicious Code & Application Attacks

Time of Check to Time of Use (TOCTOU)

- Race condition between validation and execution
- System checks a file or resource, then accesses it later
- Attacker replaces or modifies resource in the meantime
- Can lead to privilege escalation or unauthorized access
- Fix: atomic operations and locks to prevent switching

Chapter 21: Malicious Code & Application Attacks

Backdoors

- Hidden access methods bypassing normal authentication
- Left intentionally by developers or inserted maliciously
- Used for maintenance, or abuse
- Can persist through reboots and updates
- Must be monitored and removed in production environments

Chapter 21: Malicious Code & Application Attacks

Privilege Escalation and Rootkits

- Privilege escalation: gaining higher access than authorized
- Vertical (user to admin) or horizontal (user to another user)
- Rootkits hide malware by modifying the OS
- Kernel-mode rootkits operate with full system privileges
- Difficult to detect and remove

Chapter 21: Malicious Code & Application Attacks

Injection Vulnerabilities Overview

- Occur when untrusted input is interpreted as code
- Allow attackers to manipulate databases, files, or OS commands
- Most common in web applications
- Input validation and sanitization are key defenses
- Types include SQL, code, and command injection

Chapter 21: Malicious Code & Application Attacks

SQL Injection Attacks

- Modify SQL queries through crafted input
- Can reveal, alter, or delete database content
- Bypass authentication or execute admin commands
- Affects poorly coded login forms and search fields
- Use parameterized queries or stored procedures to defend

Chapter 21: Malicious Code & Application Attacks

Code Injection Attacks

- Inject malicious code into a running application
- Code is executed by the application or interpreter
- Often targets server-side scripting environments
- Can lead to remote code execution (RCE)
- Prevented by strict input validation and sandboxing

Chapter 21: Malicious Code & Application Attacks

Command Injection Attacks

- Target system-level commands executed by an application
- Attacker appends OS commands to user input
- Can execute arbitrary shell commands on the server
- Frequently exploited in input forms and URL parameters
- Prevention: avoid command calls or sanitize input rigorously

Chapter 21: Malicious Code & Application Attacks

Exploiting Authorization Vulnerabilities

- Occurs when users gain access to unauthorized resources
- Often due to improper access controls in the application
- Horizontal: access another user’s data
- Vertical: perform admin functions without proper rights
- Must enforce access checks at every layer

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

Insecure Direct Object References (IDOR)

- Occurs when apps expose internal object IDs (e.g., user ID, file name)
- Attackers modify input to access unauthorized resources
- Common in URLs, APIs, and file systems
- Fix: enforce access checks and use indirect references

Chapter 21: Malicious Code & Application Attacks

Directory Traversal Attacks

- Exploit improper input validation in file paths
- Use ../ sequences to access files outside allowed directories
- Can read system files (e.g., /etc/passwd, boot.ini)
- Common in file download/upload and viewing apps
- Fix: sanitize path input and use file access APIs securely

Chapter 21: Malicious Code & Application Attacks

File Inclusion Attacks

- Exploit functions that include or load files in an application
- Local File Inclusion (LFI): Load files already on the server
- Remote File Inclusion (RFI): Load external files from the web
- Often used to run arbitrary code or steal data
- Fix: limit file paths, sanitize inputs, and disable remote includes

Chapter 21: Malicious Code & Application Attacks

Cross-Site Scripting (XSS) Overview

- Inject malicious scripts into trusted websites
- Targets users, not servers
- Exploits poor input/output handling
- Can steal cookies, log keystrokes, or redirect users
- Three main types: Reflected, Stored, and DOM-based

Chapter 21: Malicious Code & Application Attacks

Reflected XSS

- Script is included in a URL or form input
- Executed immediately when the server reflects it back
- Requires user interaction (e.g., clicking a malicious link)
- Often used in phishing or redirect attacks
- Fix: encode output and validate input

Chapter 21: Malicious Code & Application Attacks

Stored (Persistent) XSS

- Malicious script is saved on the server (e.g., database)
- Executed whenever another user loads the affected page
- High impact, affects all visitors to a vulnerable page
- Common in forums, comment sections, and profiles
- Fix: sanitize inputs and encode outputs across all views

Chapter 21: Malicious Code & Application Attacks

Cross-Site Request Forgery (CSRF)

- Tricks user into submitting unwanted actions from a trusted site
- Exploits browser’s automatic cookie/session handling
- Common targets: changing passwords, sending money, or deleting data
- Requires a logged-in victim and a vulnerable application
- Fix: use CSRF tokens and same-site cookie attributes

Chapter 21: Malicious Code & Application Attacks

Server-Side Request Forgery (SSRF)

- Exploits a server’s ability to make HTTP requests
- Attacker tricks server into accessing internal or external resources
- Can lead to internal network scanning, metadata theft, or RCE
- Cloud services are especially vulnerable
- Defenses: whitelist URLs, validate user input, and restrict outbound traffic

Chapter 21: Malicious Code & Application Attacks

Session Hijacking

- Attacker steals or predicts a valid user session ID
- Grants unauthorized access without login credentials
- Methods: packet sniffing, XSS, insecure cookies
- Can impersonate users and access sensitive data
- Prevention: use HTTPS, secure cookies, regenerate IDs after login

Chapter 21: Malicious Code & Application Attacks

Input Validation

- First line of defense against injection attacks and data corruption
- Validate data type, length, format, and content
- Whitelist acceptable input instead of blacklisting bad input
- Enforce both client-side and server-side validation
- Apply validation to all user input, even “hidden” fields

Chapter 21: Malicious Code & Application Attacks

Web Application Firewalls (WAFs)

- Filter and monitor HTTP traffic between users and web apps
- Block common attack patterns: XSS, SQLi, CSRF, etc
- Can be signature-based, behavioral, or AI-driven
- Useful for protecting legacy apps or filling security gaps
- Not a replacement for secure coding

Chapter 21: Malicious Code & Application Attacks

Database Security

- Limit database permissions using least privilege
- Use encryption for sensitive fields and full-disk storage
- Separate database and application servers
- Monitor query logs for anomalies or injection attempts
- Use secure authentication and patch regularly

Chapter 21: Malicious Code & Application Attacks

Parameterized Queries and Stored Procedures

- Prevent SQL injection by separating code from data
- Use placeholders for user input (? or @param) in queries
- Stored procedures encapsulate logic and reduce input handling errors
- Ensure procedures are securely written and do not concatenate strings

Chapter 21: Malicious Code & Application Attacks

Obfuscation and Camouflage

- Techniques to make code harder to understand or reverse-engineer
- Used by attackers (malware) and defenders (protection)
- Examples: code minification, packing, string encoding
- May hide malicious intent or protect intellectual property
- Legitimate use in DRM or proprietary algorithms

Chapter 21: Malicious Code & Application Attacks

Code Signing

- Uses digital signatures to verify code authenticity and integrity
- Confirms the code comes from a trusted source
- Prevents tampering during transit or installation
- Common in operating systems, browser extensions, drivers
- Relies on PKI and certificates

Chapter 21: Malicious Code & Application Attacks

Code Reuse and Software Diversity

- Reusing code reduces development time but increases shared risk
- Vulnerabilities in shared libraries affect all dependent apps
- Software monoculture makes wide-scale attacks easier
- Diversity (e.g., varying OSes, compilers) can reduce systemic risk
- Use vetted, maintained libraries and monitor dependencies

Chapter 21: Malicious Code & Application Attacks

Code Repositories and Integrity Measurement

- Version control systems (e.g., Git) manage source code securely
- Track changes, authorship, and history of each file
- Supports code reviews, auditing, and rollbacks
- Integrity measurement detects unauthorized changes
- Secure repositories against unauthorized access and tampering

Chapter 21: Malicious Code & Application Attacks

Secure Coding Practices

- Use established guidelines (e.g., OWASP, SEI CERT)
- Validate input, handle output securely, manage errors gracefully
- Avoid unsafe functions and libraries
- Perform regular code reviews and static analysis
- Document and follow secure development lifecycle (SDLC)

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

Secure Code Comments and Error Handling

- Comments help developers but may reveal sensitive logic
- Avoid writing secrets or TODOs like “fix later” in comments
- Improper error handling may expose internal system details
- Show generic error messages to users; log detailed info securely

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

Hard-Coded Credentials

- Embedding usernames, passwords, or API keys in code is a critical flaw
- Often forgotten and committed to version control
- Can lead to full system compromise if leaked
- Use secure vaults or environment variables instead
- Regularly rotate and audit credentials

Chapter 21: Malicious Code & Application Attacks

Summary

- Malware takes many forms: viruses, worms, trojans, ransomware, spyware, and more.
- Attackers vary in skill and motive—from script kiddies to nation-state APTs.
- Application vulnerabilities like injection, XSS, CSRF, and buffer overflows are common attack vectors.
- Prevention requires layered defenses: patching, least privilege, code reviews, and behavioral monitoring.
- Secure coding practices, validation, and access control are essential for modern software security.

Chapter 21: Malicious Code & Application Attacks

What is a characteristic of a worm that distinguishes it from a virus?

A. It requires user action to spread
B. It hides in macro scripts
C. Is self-replicates without user intervention
D. It only affect Linux systems

Chapter 21: Malicious Code & Application Attacks

What kind of attack tricks a server into making HTTP requests to internal or external resources on behalf of the attacker?

A. Cross-site scripting
B. Server-side request forgery
C. SQL Injection
D. Directory traversal

Chapter 21: Malicious Code & Application Attacks

What control is most effective against SQL injection attacks?

A. Parameterized Queries
B. Disabling autocomplete
C. Code Signing
D. Using TLS

Chapter 21: Malicious Code & Application Attacks

What is the primary security risk of hard-coded credentials in source code?

A. They degrade performance
B. They make debugging difficult
C. They can be easily discovered and exploited
D. They interfere with version control

Chapter 21: Malicious Code & Application Attacks

What type of attack is MOST likely to result from inadequate server-side access control?

A. Denial of service
B. Insecure Direct Object Reference (IDOR)
C. CSRF
D. Code Injection

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21: Malicious Code & Application Attacks

What best describes the purpose of a web application firewall (WAF)?

A. Encrypts web traffic
B. Prevents OS-level exploits
C. Prevents phishing emails
D. Filters and blocks malicious HTTP traffic

CISSP® MENTOR PROGRAM – SESSION TWELVE

Chapter 21

The end!

- You made it to the end of session twelve!

- Domains:
- 3 – Security architecture and engineering
- 7 – Security Operations
- 8 – Software development security

I'll see you all on Wednesday!