



S.V.K.P & Dr K.S. RAJU ARTS & SCIENCE COLLEGE PENUGONDA (A)



CYBER SECURITY

Presented by

K.Vineet Kumar



Confidentiality



Integrity



Availability



Threat
Prevention

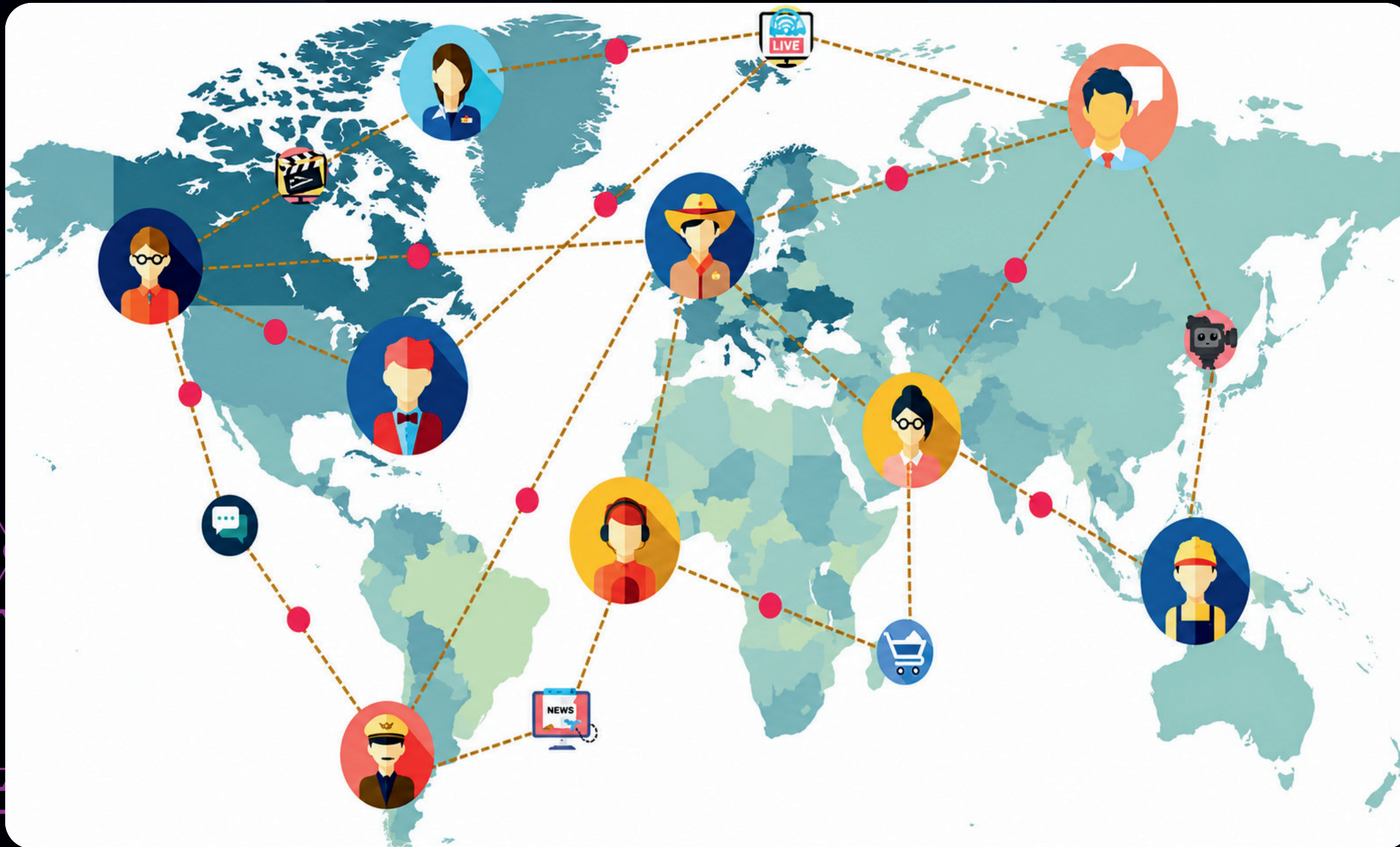


Safe Digital
World





DIGITAL ERA



CYBER CRIMINAL



1. CONTACT INFO
2. CREDIT CARD INFO
3. BANK A/C INFO

THIRD PARTIES
PROFIT



POLITICAL &
SOCIAL ACTIVITES



AUTHENTICATION

VS

AUTHORIZATION



App Development

AUTHENTICATION

Login credentials
Right User Name
Right password



Web Development

AUTHORIZATION

What You are allocated to do
Ex : Our College Website

Set of Things : Student Portal , Staff Portal

No changes in Website for MARKS , ATTENDANCE

No Permissions



INTEGRITY

Original Data

Duplicate Data



CONFIDENTIALITY

Privacy Maintain



AVAILABILITY

- Data Crash
- Servers Crash

- Apps
- Website

Personal Data

Profession Data



BASICS OF CRYPTOGRAPHY



EXAMPLE

H	E	L	L	O
↓	↓	↓	↓	↓
*	#	*	#	*



Symmetric

- **Symmetric cryptography/ symmetric cipher**
- **Model used only one key for the process**
- **Both encryption and decryption of data.**
- **it is also called secret key / private key cryptography**

Asymmetric

- **Asymmetric cryptography used two keys**
- **One key for encryption**
- **Secound key for decryption**



SYMMETRIC



HOD

Sender

Practical Exams
Start Today
Morning 8:00 AM

Cipher Text

22548974hg
dyvgdcjnvhg
gjbjvh

Encryption
Key



Decryption
Key

Plain Text

Algorithms



Students

Practical
Exams Start
Today
Morning
8:00 AM



Recipient

ASYMMETRIC



HOD

Sender

Practical Exams
Start Today
Moorning 8:00 AM

1  Public Key
**Encryption
Key**

22548974hg
dyvgdcjnvhg
gjbjvh



**Decryption
Key**
2  Private Key

Practical
Exams Start
Today
Moorning
8:00 AM

Cipher Text

Plain Text

Algorithms



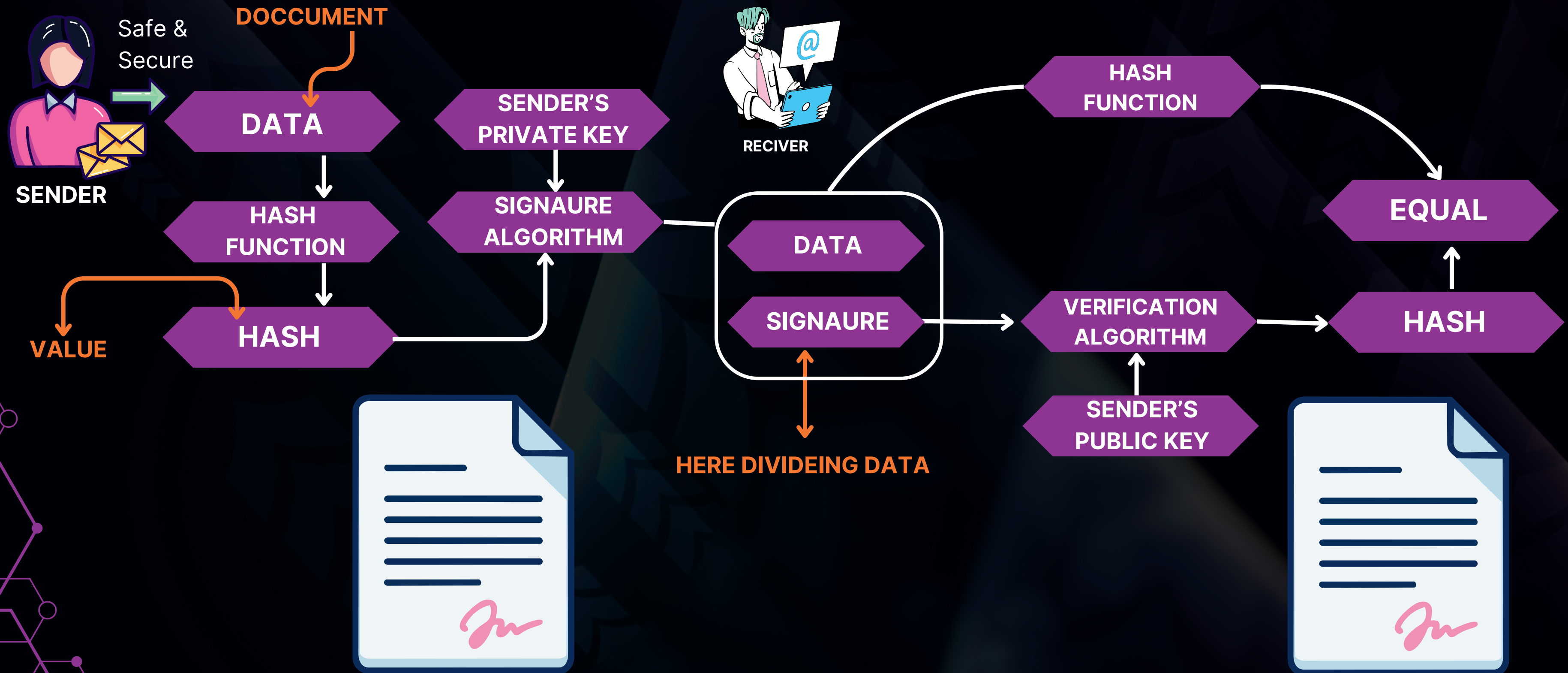
Students



Practical
Exams Start
Today
Moorning
8:00 AM

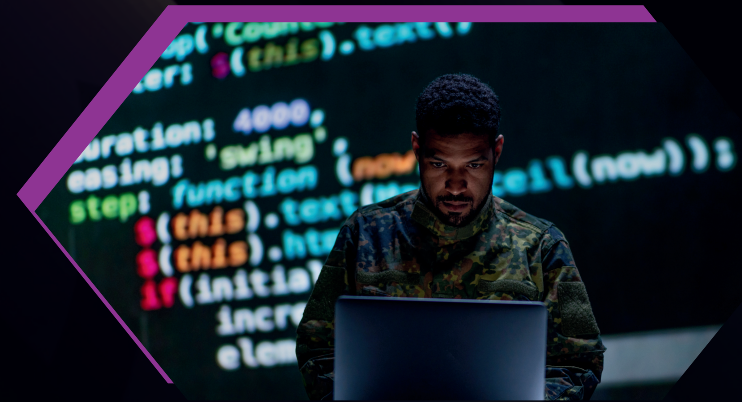
Recipient

DIGITAL SIGNATURE

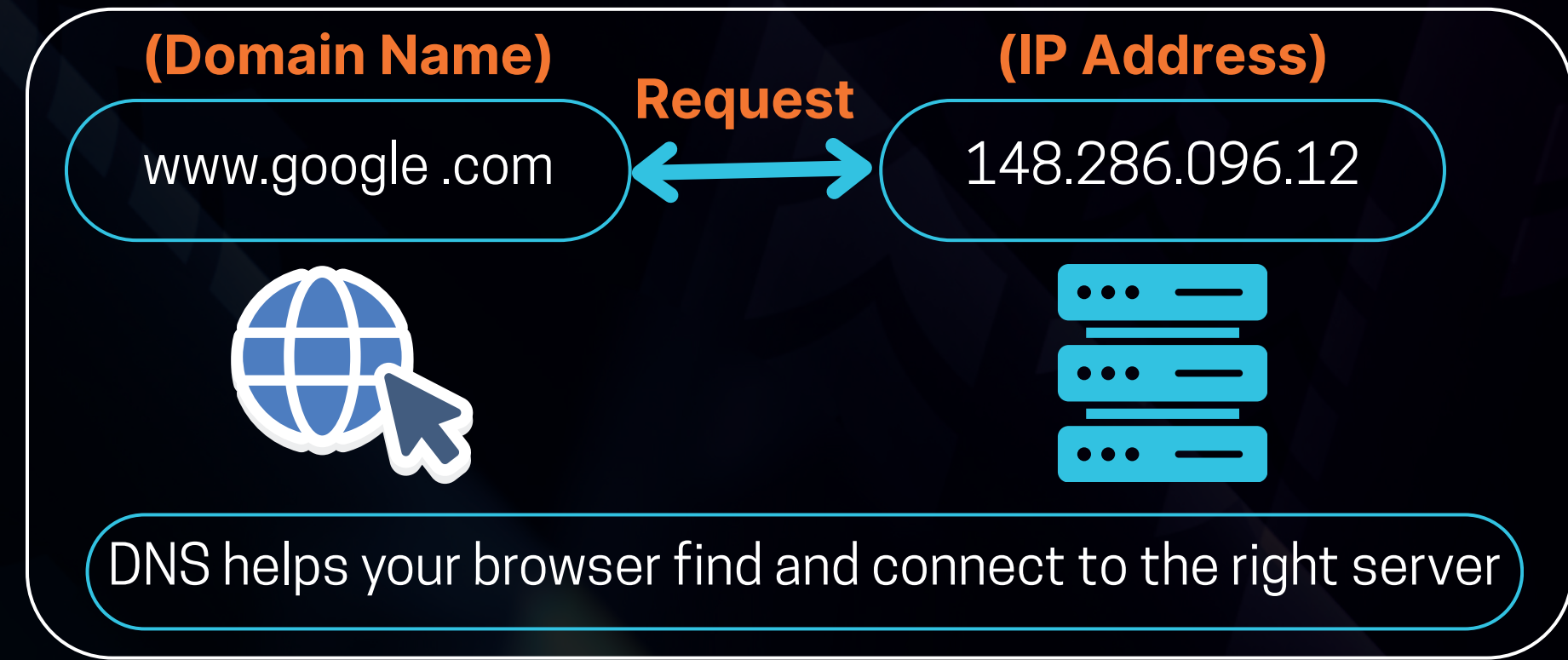


DNS

Domain Name System



- ▶ Like the phonebook of the internet
- ▶ Its translates easy-to-remember domain names
- ▶ Like (www.svkp & dr.k.s raju arts & science college)
- ▶ Into IP Address like (148.286.096.12)
- ▶ Computer used to identify each other on the network



HOW DNS WORK?



1 DATA/FILE DELETION



- Deleted files are not immediately erased.
- They can be recovered until overwritten.
- Tools:** Autopsy, Recover My Files, Ease US Data Recovery.

2 PASSWORD PROTECTION



- Passwords protect systems and hide evidence.
- Attackers use them to block access and hinder investigations.
- Tools:** ophcrack, Rainbow Crack.

ANTI FORENSICS

Techniques used by attackers to **hide, alter or destroy** digital evidence and avoid detection



3 STEGANOGRAPHY



- Hides data inside other files (images, audio, video, etc.).
- Used to hide secret messages, data, or plans.
- Investigators analyze **Stego-Objects** to extract hidden data.

4 TRAIL OBFUSCATION



- Misleads investigators by altering or hiding traces.
- Techniques:** log tampering, timestamp & file header modification.
- Tool:** Timestomp.

6 ENCRYPTION



- Converts data into secret code using encryption algorithms.
- Makes data unreadable without the right key.
- Protects evidence from unauthorized access.

5 ARTIFACT WIPING



- Deletes or destroys evidence permanently.
- Uses wiping tools, disk cleaning, or physical destruction.
- Eliminates files or entire file systems.



MAIN GOAL

To remain undetected by digital forensic investigators during and after malicious activities.



ANTI FORENSICS THREATENS DIGITAL JUSTICE. (AWARENESS • PREVENTION • INVESTIGATION



Techniques Gaining A Foothold in Cyber Security

Presented By
K.Vineet Kumar
UG CS Lecture



Vulnerabilities

Weaknesses in systems, applications or human behavior that attackers can exploit.

Live Example:

Unpatched software (e.g., Log4Shell vulnerability in Apache Log4j) allows remote code execution.



Unauthorized Access

Gaining access to systems, data or accounts without proper permission.

Live Example:

An attacker uses stolen credentials to log in to a corporate VPN and access internal resources.



Malicious Activities

Actions intended to harm, steal or disrupt systems, data or operations.

Live Example:

Ransomware (e.g., WannaCry) encrypts files and demands payment for decryption.



Techniques Gaining A Foothold in Cyber Security

Social engineering techniques



Manipulating people to reveal confidential info.

Example: Fake IT support calls.

Phishing



Fake emails to steal data or credentials.

Example: "Your account will be suspended" email.

Spear phishing



Targeted phishing to specific individuals.

Example: CEO receives a fake invoice email.

Pretexting



Creating a false scenario to gain trust.

Example: Pretending to be bank support for OTP.

Baiting



Using attractive offers to lure victims.

Example: Free software on a USB drive with malware.

Watering hole attacks



Compromising trusted websites.

Example: Hacking a popular news site to drop malware.

Exploitation of vulnerabilities



Using known flaws in software or systems.

Example: EternalBlue (SMB vulnerability).



More Techniques & Live Examples

Exploiting public-facing applications



Targeting web apps exposed to the internet.

Example: SQL injection on a website.

Drive-by compromise



Malware installed without user interaction.

Example: Malicious ads on infected websites.

Supply chain compromise



Compromising trusted third-party software/hardware.

Example: SolarWinds attack.

Replication through removable media



Spreading malware via USB or external drives.

Example: Infected USB used in a company network.

Valid accounts



Using legitimate accounts for malicious access.

Example: Stolen employee mailbox used for data theft.

Password attacks



Guessing, brute force or credential stuffing.

Example: Using leaked password lists.

External remote services



Misusing exposed services (e.g., RDP, SSH).

Example: Brute forcing RDP to gain access.

Trusted relationships



Abusing trust between entities or users.

Example: Compromised vendor access to internal network.

Wi-Fi networks



Eavesdropping or rogue access points.

Example: Fake Wi-Fi hotspot stealing data.



Post-exploitation and Persistences

Installing backdoors



Maintaining long-term access to the system.
Example: Web shell or reverse backdoor.

Command and control (C2)



Communication with attacker's server.
Example: Malware beaconing to C2 server.



Privilege escalation

Gaining higher access levels.
Example: Exploiting kernel vulnerability to get admin rights.



Lateral movement

Moving across systems in the network.
Example: Using stolen credentials to access other servers.

**BE AWARE
STAY SECURE**

**CYBER SECURITY
IS EVERYONE'S RESPONSIBILITY**



RECONNAISSANCE

Gathering Information | Finding Weaknesses | Planning the Attack

Presented By
K.Vineet Kumar
UG CS Lecture



Reconnaissance

Reconnaissance is the process of gathering information about a target (person, organization, or system) to identify potential vulnerabilities and plan further attacks.



Purpose

- Identify targets
- Discover vulnerabilities
- Gather useful information
- Plan for further attacks



Scope

- People
- Organizations
- Systems
- Networks
- Infrastructure



Types of Reconnaissance

1. Passive Reconnaissance

Gather information without direct interaction with the target.

2. Active Reconnaissance

Interact with the target to gather more detailed information.



Information
=
Advantage



1. Passive Reconnaissance

Uses publicly available information and does not involve direct interaction with the target.

Methods

1. Domain WHOIS Lookups



Example:

whois example.com
→ Reveals domain owner, registrar, creation date, etc.

2. Social Media Analysis



Example:

Searching "Acme Corp" on LinkedIn
→ Finds employees, org structure, tech stack, etc.

3. Public Records



Example:

Google search + government records
→ Finds business registrations, legal filings, etc.



2. Active Reconnaissance

Involves direct interaction with the target to gather more detailed information.

Methods

1. Network Scanning



Example:

Nmap -sn 192.168.1.0/24
→ Discovers live hosts in the network.

2. Port Scanning



Example:

Nmap -p 1-65535 target.com
→ Finds open ports and services.

3. Banner Grabbing



Example:

Nmap -sV target.com
→ Reveals service versions (e.g., Apache 2.4).

4. Ping Sweeps



Example:

Nmap -sn 10.10.10.0/24
→ Identifies active hosts using ICMP.

Live Example (Active Recon)

```
$ nmap -sV -p 22,80,443 target.com
```

Nmap scan report for target.com

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 8.2p1
80/tcp	open	http	Apache 2.4.41
443/tcp	open	https	Apache 2.4.41

We discovered:

- Open ports
- Running services
- Service versions



Tools and Techniques

1. Tools

Nmap



Features:

- Host discovery
- Port scanning
- Service detection
- OS detection

Example:

nmap -sV target.com

Wireshark



Features:

- Packet capture
- Protocol analysis
- Real-time inspection
- Troubleshooting

Example:

wireshark (open .pcap)

Maltego



Features:

- OSINT & link analysis
- Entity relationship mapping
- Data visualization
- Integration with other tools

Example:

maltego (search domain)

2. Techniques

Footprinting



Collecting basic information about the target.

Example Tools:

- Google Dorks
- WHOIS
- Shodan

Scanning



Finding live hosts, open ports, and services.

Example Tools:

- Nmap
- Masscan
- Nessus

Enumeration



Gathering detailed information from services and systems.

Example Tools:

- Nmap
- Wireshark
- Enum4linux

Live Real-World Example

Target: example.com

1. WHOIS Lookup → Find domain owner
2. Google Dorks → Find exposed pages
3. Nmap Scan → Discover open ports
4. Wireshark → Capture and analyze traffic
5. Maltego → Map relationships



Employees
Subdomains
Infrastructure

Information
Leads to
Opportunity

The more you know about your target → The better you can protect → Or the more effectively you can attack



Cyber Security

Be Aware | Be Prepared | Stay Secure



Self-Replicating Malicious Codes



Small Code ... Big Damage ... Be Aware ... Be Safe

What are self-replicating malicious codes?

Self-replicating malicious codes are programs that can make copies of themselves and spread to other files, systems or networks without the user's permission, causing harm.



Types of Self-Replicating Malicious Codes

Worms



- Self-replicate
- Spread through networks
- Do not need a host file

Viruses



- Attach to files/programs
- Need a host
- Replicate when executed

Trojan Horses



- Looks legitimate
- Hides malicious code
- Gives attacker access



How Do Worms Spread?

- Through networks (e.g., email, shared folders)
- Via infected USB drives
- Exploiting software vulnerabilities
- Spreading across connected systems



What are the Risks of Malicious Code?

-  **Data theft** – stealing sensitive information
-  **System disruption** – crashing or disabling systems
-  **Financial loss** – fraud, ransomware, downtime
-  **Reputational damage** – loss of trust and brand value

How to Protect Against Malicious Code



Keep software updated

Fixes security vulnerabilities



Use strong passwords

Avoid common and weak passwords



Be cautious with email and downloads

Avoid suspicious links and attachments



Install and update antivirus & anti-malware software

Detects and removes threats



Enable firewalls

Blocks unauthorized access



Implement network security measures

Monitors and protects your network

Live Examples



WannaCry

- Ransomware
- Spreads via networks
- Locks files



ILOVEYOU

- Email worm
- Spreads via email
- Overwrites files



Zeus

- Banking trojan
- Steals credentials
- Financial fraud



Evading Detection and Elevating Privileges in Cyber Security



Understanding Privilege Escalation

Privilege escalation is the process of gaining higher levels of access than intended, allowing an attacker to perform actions they are not authorized to do.



Vertical Privilege Escalation

Gaining higher access on the same system (e.g., from user to administrator).



Horizontal Privilege Escalation

Gaining access to another user's account with the same level of privileges (e.g., user A → user B).



Attack Vectors

- Exploiting software vulnerabilities
- Phishing and social engineering
- Weak or reused credentials
- Malicious files or scripts
- Misconfigured systems



Evasion Techniques

- Obfuscation (code/commands)
- Encoding and encryption
- Living off the land (LOLBins)
- Using legitimate tools
- Anti-analysis techniques



Bypassing Detection Systems

- Signature evasion
- Behavioral manipulation
- Using zero-day exploits
- Polymorphic malware
- Custom payloads



Covering Tracks

- Deleting logs
- Clearing event history
- Modifying timestamps
- Removing artifacts
- Using anti-forensics tools



Abusing Permissions

- Misusing admin accounts
- Exploiting misconfigured ACLs
- Privilege abuse in applications
- Token theft and replay
- Access to sensitive data



Prevention Strategies



Least Privilege

Give only the minimum access required.



Vulnerability Scanning

Find and fix weaknesses before attackers exploit them.



MFA (Multi Factor Authentication)

Use multiple forms of verification.



Network Segmentation

Isolate critical systems and networks.



Access Control

Manage and monitor who can access what.



Regular Auditing

Review logs, permissions and configurations.



Monitoring

Detect and respond to suspicious activity.



User Education

Train users to identify and avoid threats.

Live Examples

- Vertical Escalation:** User gains admin rights via a vulnerability in Windows (e.g., CVE-2021-42278).
- Horizontal Escalation:** Attacker uses stolen credentials to access another user's account.
- Evasion:** Malware uses encrypted payloads to avoid detection (e.g., Emotet).
- Bypassing Detection:** Fileless malware using PowerShell (LOLBins).
- Covering Tracks:** Deleting Windows event logs after access.
- Abusing Permissions:** Exploiting misconfigured S3 bucket or database permissions.



Obfuscation

Hide the Intent • Confuse the Analyst • Stay Ahead



What is Obfuscation?

Obfuscation is the process of making code, data or information harder to understand, analyze or reverse engineer, without changing its original functionality.



Purpose of Obfuscation



Protect Intellectual Property

Keep source code and algorithms safe.



Prevent Malware Analysis

Confuse anti-virus and security tools.



Hinder Reverse Engineering

Make it difficult to understand the logic.



Protect Sensitive Data

Hide important information.



Real-World Scenarios



Malware

Uses obfuscation to avoid detection.



Mobile Apps

Obfuscates code to protect IP.



Ransomware

Hides payload and C2 communication.



Web Applications

Obfuscates JavaScript and API calls.



IoT Devices

Hides firmware logic and credentials.



Types of Obfuscation



Code Obfuscation

Hides the logic of the code.



Data Obfuscation

Protects data from being understood.



Malware Obfuscation

Hides malicious behavior.



Obfuscation vs. Encryption

Obfuscation hides intent, encryption hides content.



Limitations of Obfuscation



Doesn't guarantee security

Can be bypassed with advanced analysis tools and skills.



Can impact performance

May increase code size and reduce execution speed.



LOWER PERFORMANCE



Obfuscation Techniques



String Obfuscation

"Hello" → "x3f!9k"



Polymorphic Obfuscation

Changes code structure



Metamorphic Obfuscation

Alters code logic each time



Dummy Code Insertion

Adds useless code



Control Flow Obfuscation

Jumps & branches are hidden



Renaming

Variables/functions are renamed



Live Examples

String Obfuscation

```
var a = "Hello";  
↓  
var a = '\x48\x65\x6c\x6c\x66';
```

Original vs. obfuscated string

Polymorphic Malware



Same behavior, different code

Control Flow Obfuscation

```
if (x) {  
    goto label1;  
}  
label1:
```

Confusing jumps and branches

Dummy Code Insertion

```
if (true) {  
    int junk = 12345;  
    // dummy code  
}
```

Adds useless instructions





Understanding Privilege Escalation



Here's an example of how a horizontal privilege escalation attack might unfold



1 Initial Access
Attacker gains access with a low-privileged account (e.g., regular user).



2 Exploitation
Finds a vulnerability or weak configuration to escalate privileges.



3 Privilege Escalation
Gains higher privileges (e.g., admin) on the same system or network.

Why it is important to prevent privilege escalation attacks



Common privilege escalation attack techniques and examples

	Credential exploitation	Example: Stolen passwords, reused credentials
	Kernel exploits	Example: Exploiting kernel vulnerabilities (e.g., CVE-2021-xxxx)
	Exploitation of vulnerable software	Example: Outdated software, known vulnerabilities
	Abuse of weak service configurations	Example: Unnecessary services, default settings

Credential Exploitation



- Use of Mimikatz**
Extracts credentials from memory.
- Credential dumping**
Dumps stored credentials.
- Pass-the-hash attacks**
Reuses NTLM hashes.
- Golden ticket attack**
Forged Kerberos tickets for full access.

Other actions by Scattered Spider include



- ✓ Social engineering
- ✓ SIM swapping
- ✓ Cloud account takeover
- ✓ Lateral movement
- ✓ Data exfiltration
- ✓ Ransomware deployment

Tips to prevent privilege escalation techniques and attacks

Use MFA

Implement privileged access management (PAM)

Principle of least privilege (PoLP)

Update and patch systems regularly

Check your settings

Encrypt sensitive data

Monitor user activities

Live Examples

Mimikatz
Dumping credentials from memory (e.g., lsass.exe).

Pass-the-Hash
Reusing NTLM hash to access other systems.

Golden Ticket
Forged ticket for domain administrator access.

CVE-2021-44228 Log4Shell
Vulnerable Software
Exploiting known vulnerabilities (e.g., Log4Shell).

Weak Service Config
Using misconfigured services (e.g., default creds).

GetUserSPNs.py AS-REP / TGS
Kerberoasting
Cracking service account hashes.

Scattered Spider
Using social engineering and cloud access.

Presented By
K.Vineet Kumar
UG CS Lecture





Stealing Information and Exploitation in Cybersecurity



Stealing Information (Data Theft)

Data theft is the unauthorized acquisition of sensitive information from individuals, organizations or systems.



STOLEN DATA = REAL VALUE

Examples of Stolen Information



Personal Identifiable Information (PII)

- Name
- Address
- Aadhaar/SSN
- Date of Birth



Financial Data

- Bank details
- Credit card info
- Payment records



Intellectual Property

- Source code
- Designs
- Patents
- Trade secrets



Login Credentials

- Usernames
- Passwords
- API keys

Common Methods Used to Steal Information



Phishing Attacks

Fake emails, links or websites to trick users.

Example: Fake bank email asks for login details.



Malware

Malicious software that steals data.

Example: Keylogger captures keystrokes.



Man-in-the-Middle (MITM) Attacks

Intercepting communication between two parties.

Example: Public Wi-Fi sniffing.



SQL Injection

Injecting malicious SQL code into an app.

Example: Bypassing login via SQL injection.



Brute Force Attacks

Guessing passwords automatically.

Example: Automated tool tries millions of passwords.

Exploitation



Exploitation means taking advantage of a vulnerability in a system, application or human behavior to gain unauthorized access or perform malicious activities.

FIND → EXPLOIT → GAIN ACCESS

Common Exploitation Activities



Identity Theft

Steal personal identity for fraud.

Example: Fake KYC documents used for loan fraud.



Financial Fraud

Unauthorized transactions, fund transfers.

Example: Phishing leads to bank account compromise.



Corporate Espionage

Steal business data, trade secrets.

Example: Competitor gains source code.



Ransomware Attacks

Encrypt data and demand payment.

Example: WannaCry, LockBit.



Social Engineering

Manipulate people to get information.

Example: Fake IT support calls for remote access.

Impact of Data Theft and Exploitation



Financial Loss

Direct and indirect monetary damage.



Identity Fraud

Misuse of personal information.



Reputational Damage

Loss of trust and brand value.



Data Breaches

Exposure of sensitive data.



Operational Disruption

Service downtime and system failure.



Legal Consequences

Fines and regulatory action.

How to Prevent It



Use MFA

Add an extra layer of security.

Example: Authenticator app or OTP.



Implement Privileged Access Management (PAM)

Control and monitor privileged accounts.

Example: Use PAM tools (e.g., CyberArk).



Principle of Least Privilege (PoLP)

Give only necessary access.

Example: Limit admin rights to specific users.



Update and Patch Systems Regularly

Fix known vulnerabilities.

Example: Apply Windows and application patches.



Check Your Settings

Secure configurations and remove defaults.

Example: Disable unused services and open ports.



Encrypt Semi-Sensitive Data

Protect data at rest and in transit.

Example: Use full-disk encryption and HTTPS.



Monitor User Activities

Detect and respond to suspicious behavior.

Example: SIEM logs, UEBA tools.

Stay Aware • Stay Safe • Stronger Together

Presented By

K.Vineet Kumar
UG CS Lecture





LOAD BALANCING

From Single Server to Multiple Servers

Distribute the Load | Improve Performance | Avoid Downtime

Load Balancing is the process of distributing incoming network traffic across multiple servers to ensure **better performance, high availability** and **avoid any single point of failure**.

1. IPL FINAL MATCH

Live Example



During IPL final, millions of fans try to watch live, so the traffic is distributed across multiple servers to handle the huge load smoothly.

2. D-MART CASH COUNTERS

Real Time Example



In D-Mart, when the supervisor asks customers to go to different counters, **the load is distributed** across multiple counters for **faster service**.

3. INTER RESULTS

Live Example



During Inter results, lakhs of students access the website at the same time. If only a single server is used, it can get overloaded and go down.

 **Single Server**

- ✗ Overloaded
- ✗ Slow Response
- ✗ May Go Down

 **Multiple Servers**

- ✓ Handles Huge Traffic
- ✓ Fast Response
- ✓ High Availability

Single Server (Without Load Balancing)



- ! Server Overloaded
- ! Slow Performance
- ! Downtime / Service Failure

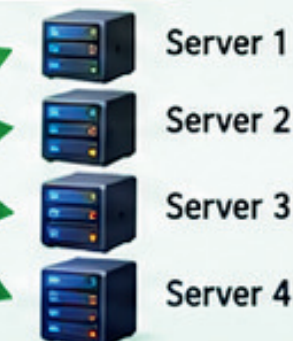
One Server = One Point of Failure

Load Balancing

Multiple Servers (With Load Balancing)



Load Balancer



- ✓ Balanced Load
- ✓ Faster Response
- ✓ High Availability
- ✓ No Single Point of Failure



Presented By

K.Vineet Kumar

UG CS Lecture



DEFENSE AND ANALYSIS TECHNIQUES

MEMORY FORENSICS



Memory Holds The Clues...

Why Memory Forensics Matters in Cybersecurity

Why Memory Forensics Matters in Cybersecurity

- 1. Malware detection**
Finds active malware, injected code, persistence.
Example: Detecting Cobalt Strike beacons in memory.
- 2. Incident response**
Helps understand what happened, who did it, and how.
Example: Identifying attackers' tools and execution flow.
- 3. Data recovery**
Recovers lost or deleted data from memory.
Example: Retrieving deleted files or passwords from memory dumps.
- 4. Digital investigations**
Provides evidence for legal and compliance purposes.
Example: Finding chat logs, browser history, or encrypted keys.
- 5. Uncovering Hidden Malware**
Detects rootkits, bootkits and stealthy processes not visible on disk.
Example: Identifying kernel rootkits in memory.
- 6. Live Analysis**
Allows analysis of running systems without shutting them down.
Example: Capturing memory from a live system to catch active processes.



Challenges in Memory Forensics

- Data Volatility**
Memory is lost when the system is powered off.
Example: Power loss or system crash can destroy evidence.
- Memory Acquisition**
Collecting a complete and accurate memory dump is complex.
Example: Partial dump, compression, or incomplete capture.
- Tool Selection**
Choosing the right tool for the right scenario.
Example: Volatility vs. Rekall vs. Magnet RAM Capture.
- Data Integrity**
Ensuring data is not altered during the process.
Example: Hash verification and write-blocking techniques.

Tools and Techniques

- Memory Dump Tools**
 - Dumplt
 - WinPMEM
 - FTK Imager
 - Magnet RAM Capture

Example: Using WinPMEM to capture RAM from a live Windows system.
- Memory Analysis Tools**
 - Volatility
 - Rekall
 - Magnet AXIOM
 - LiME (Linux)

Example: Using Volatility to find running processes and injected malware.
- Reverse Engineering**
 - IDA Pro
 - Ghidra
 - x64dbg
 - OlllyDbg

Example: Analyzing a memory-resident malware's code to find persistence methods.
- Chain of Custody**
 - Documentation
 - Hashing (MD5/SHA256)
 - Evidence handling
 - Legal compliance

Example: Maintaining logs and hashes for court admissibility.

Real World Examples

- ✓ Ransomware (e.g., Ryuk) – find encryption keys.
- ✓ Credential theft – extract passwords from memory.
- ✓ APT attacks – identify C2 communication.
- ✓ Insider threats – detect suspicious processes.
- ✓ Live response – stop the malware before it spreads.

Memory Tells Stories... We Just Need the Right Tools to Read Them!



Presented by
K.Vineet Kumar





S.V.K.P & Dr K.S. RAJU ARTS & SCIENCE COLLEGE PENUGONDA (A)



HONEYPOTS

Lure • Detect • Defend

Same
Attacks...
Different
Outcome!



How Honeypots Work

1



Attract Hackers

The honeypot looks like a real system or valuable target, luring attackers with fake data, services or vulnerabilities.

2



Watch Closely

Their activities are monitored and recorded in detail.

3



Learn and Analyze

Attack patterns, tools and techniques are studied to understand the threat.

4



Protect Real Systems

The insights help improve defenses and stop real attacks on production systems.



Benefits of Using Honeypots



Better Threat Awareness

Understand attacker behavior and goals.



Early Warning

Detect attacks before real systems are targeted.



Stronger Network Security

Helps improve firewalls, IDS/IPS and policies.



Attack Diversion

Keeps attackers busy while real assets stay safe.



Types of Honeypots



1. Research Honeypots

- Used for research and learning
- No real production data
- Example: Kippo (SSH), Honeyd



2. Production Honeypots

- Deployed in real networks
- Collects real attack data
- Example: Cowrie, Dionaea



Limitations of Honeypots



False Alarms

- Legitimate users may trigger alerts
- Can lead to confusion



Security Risk

- If compromised, attacker may access other systems



Need for Skilled Experts

- Requires analysis and experienced staff
- Needs proper configuration



Real-World Examples



Kippo (SSH Honeypot)

- Captures brute force and login attempts
- Logs attacker commands



Cowrie (SSH/Telnet Honeypot)

- Simulates vulnerable services
- Records full session data



Dionaea (Malware Honeypot)

- Attracts malware and botnets
- Collects samples & behavior



Honeyd (Low-Interaction Honeypot)

- Emulates multiple services
- Easy to deploy and scale

Real Threats
Leave Clues...
Honeypots
Catch Them!

Presented by

K.Vineet Kumar



S.V.K.P. & Dr K.S. RAJU ARTS & SCIENCE COLLEGE PENUGONDA (A)

MALICIOUS CODE NAMING

Identify • Track • Stop • Stay Secure

Different Names
Same Threat...
Be Aware!

</> Types of Malicious Code Naming

- 1 Functionality-Based Naming**
Based on what the malware does (e.g., steals data, deletes files, encrypts).
Examples:
 - Trojan-Downloader
 - Ransomware
 - Spyware
- 2 Origin-Based Naming**
Based on where it comes from (e.g., country, organization, region).
Examples:
 - China-Backdoor
 - Russia-APT
 - US-Spyware
- 3 Family-Based Naming**
Based on the malware family or variant it belongs to.
Examples:
 - Conficker Family
 - Zeus Family
 - Emotet Family
- 4 Attack Method-Based Naming**
Based on how it spreads or attacks (e.g., phishing, brute force, exploit).
Examples:
 - Phishing.Malware
 - BruteForce.Win32
 - Exploit.Kit
- 5 Author-Defined Naming**
Based on the creator's chosen name or internal label.
Examples:
 - AgentTesla
 - DarkComet
 - Lokibot

💡 Why Naming Is Important

- Helps Identify Threats**
Quickly recognize the type and behavior of malware.
Example: "Ransomware" → look for encryption activity.
- Makes It Easier to Track**
Helps in monitoring, correlation and threat intelligence.
Example: Track "Emotet" across multiple systems.
- Supports Teamwork**
Enables clear communication among security teams.
Example: Use common names in incident reports and alerts.
- Raises Awareness**
Helps users and analysts understand and respond faster.
Example: "Phishing" name warns users about suspicious emails.

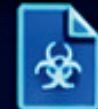
**The Right Name
Can Make
A Big Difference!**

📄 Common Real-World Examples


WannaCry
Ransomware
(encrypts files)


Conficker
Worm
(spreads via USB)


Zeus
Trojan
(steals banking data)


AgentTesla
Spyware
(steals credentials)


Emotet
Trojan
(spreads via email)

Presented by
K.Vineet Kumar

 **Detect Faster** |  **Respond Smarter** |  **Stay Safer**

Know the Names → Understand the Threat → Build a Safer Digital World



AUTOMATED MALACIOUS CODE ANALYSIS SYSTEM

Detect • Analyze • Protect



Main Parts of Automated Malware Analysis

1 Sandboxing
Run suspicious files in an isolated environment to observe behavior.

2 Static Analysis
Analyze the code without executing it (e.g., signatures, strings, structure).

3 Dynamic Analysis
Observe behavior during execution (e.g., system calls, network activity).

4 Automated Tool
Tools used for analysis and monitoring:
• Debuggers • Process Monitor



Real-World Examples



Ransomware
(e.g., WannaCry)

Sandbox detects file activity and network connections.



Banking Trojan
(e.g., Zeus)

Dynamic analysis monitors keylogging and data exfiltration.



Fileless Malware
(e.g., PowerShell)

Static analysis checks scripts and behavior patterns.



Cryptojacking
(e.g., CoinMiner)

Process monitor detects unusual CPU usage and network activity.

Benefits

- Faster Detection**
Identifies threats quickly.
- Better Security**
Prevents data breaches.
- Stronger Threat Hunting**
Finds hidden and advanced threats.
- Saves Time and Effort**
Automates the analysis process.
- Finds Unknown Threats**
Detects zero-day and new malware.

Challenges

- Changing Malware**
Uses obfuscation and polymorphism.
- Too Complex**
Advanced techniques and multiple layers.
- False Alarms**
May flag safe files as malicious.

Automate Analysis → Stop Threats → Safer Tomorrow

Presented by **K.Vineet Kumar**



INTRUSION DETECTION SYSTEM

Detect • Analyze • Protect



Key Points



1 Watching the Network

Monitors traffic, logs and system activities in real-time.



2 Spotting Problems

Detects suspicious behavior, patterns and potential attacks.



3 Sending Alerts

Notifies administrators about possible threats.



Types of IDS



Network-based IDS (NIDS)

- Monitors network traffic.
- Detects attacks across the network.
- Example: Snort, Suricata



Host-based IDS (HIDS)

- Monitors individual devices.
- Checks system logs & files.
- Example: OSSEC, Wazuh



Detection Methods Used by IDS



Signature-based Detection

- Matches known attack patterns.
- Low false positives (for known threats).
- Example: Snort, Suricata



Anomaly-based Detection

- Detects unusual behavior.
- Good for unknown attacks.
- Example: Zeek, AI/ML-based IDS



Hybrid Detection

- Combines signature + anomaly.
- More accurate and reliable.
- Example: Snort (with anomaly modules)



Real-World Examples

DDoS Attack



- IDS detects unusual traffic volume.
- Blocks or alerts the attack.

Example: Suricata / NIDS

Brute Force Login



- Detects multiple failed login attempts.
- Alerts admin / blocks IP.

Example: OSSEC / HIDS

Malware Communication



- Identifies suspicious outbound connections.
- Stops C2 communication.

Example: Snort / NIDS

Unauthorized Access



- Detects abnormal user behavior or file changes.
- Triggers immediate alert.

Example: Wazuh / HIDS



Limitations



False Positives

Legitimate activity may be flagged as malicious.



False Negatives

Some real threats may be missed.



Evolving Threats

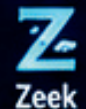
New and sophisticated attacks can bypass detection.



Performance Impact

Can consume system resources and affect performance.

Common Tools & Technologies



Smarter Detection Stronger Security

Keep Networks Safe | Stop Threats Early



Presented by
K.Vineet Kumar

Thank you



*Your Support
and Attention
Mean a Lot!*



— Presented by —
K.Vineet Kumar

