Human Attack Vectors in SOC

Not every organization has the expertise to operate a SOC on its own and relies on a Managed Security Services Provider (MSSP), a company that delivers outsourced security services, most commonly SOC, to its clients. Working at MSSP is typically high-pressure, but it is also a good option to QuickStart your career.

Why are Humans targeted?

Humans are targeted because of the access they can provide to websites, mailboxes, or databases. Some threat actors hunt for specific access, while others are not so selective and just breach as many accounts as they can and decide how to use them later

Attack targeting humans share a common trait: they rely on manipulating the victim into helping the attacker, whether knowingly or not. This tactic is known as social engineering, and it works by exploiting human psychology rather than technical flaws. For a tactic to be succeed it is designed to be:

- **Trustworthy:** The attacker must appear legitimate, so the victim trusts them
- **Emotional:** The attack must trigger feelings like urgency, fear, or curiosity

Humans can be targeted using

- Phishing attack
- Malware Downloads
- Deep fakes
- Impersonation
- Other Attacks like keylogging, Physical Attack etc.

How To Defend Humans?

Defending against threat involves two key tasks: Mitigation and Detection. Mitigation aims to prevent or reduce the chance and impact of attacks, for example by training employee's orc deploying a good anti-phishing solution.

As a SOC Analyst, your task is to detect and investigate attacks, because after sometime you will be tired of manual mitigation & will want automated way and there comes the SOC suite of automated solutions. Some solutions are:

- Anti-phishing solution
- Antivirus / EDR solution
- "Trust but verify" principle
- Security awareness training

System Attack Vectors & Vulnerabilities

System as Attack Vectors:

In SOC, we don't just protect users or data we protect SYSTEMS, because every system can be abused as an Attack vector.

1. Gain Initial access
2. Move laterally
3. Maintain persistence
4. Exfiltrate data
5. Launch further attacks

Human Oriented Attacks

Users are those who started the attack mostly: (Before external factors, Users are responsible at first)

By inserting a malicious USB found on a street, downloading malware from pirated resources, or simply reusing a weak password everywhere. 81% of breaches involve stolen or breached password.

Example:

- Rubber Ducky
- Keyloggers binded with legitimate software
- Clicking on suspicious links.
- Opening any email or sent attachments
- Storing passwords at unsecure portals
- Trusting soon
- Using outdated / pirated software's

Supply chain attacks

Your pc is home to hundreds of apps, including web browsers, messengers, development, and entertainment software. Every app depends on thousands of libraries.

If threats actors manage to breach on of the apps or libraries and push an update to all its users, all of them will be compromised. This technique is called a supply chain attack. The most famous examples are the SolarWinds and 3CX breaches which affected thousands of companies.

It is hard to protect from supply chain attack since you can't always control all the software present On your laptops, servers and web apps.

Vulnerabilities

Every piece of software have security flaws. In 2024, over 40,000 software vulnerabilities were published and more than 300 were actively exploited in major attacks

Vulnerability is a weakness in a system, software, hardware, or process that can be exploited by an attacker to gain access, escalate privileges, execute code. Steal data, Disrupt services, & more...

Software Vulnerabilities

Every piece of software has flaws, but some take years to discover. For example, Shellshock, a major Linux vulnerability, existed since 1992 but wasn't found until 2014. In the worst-case scenario, attacker discovers vulnerability before anyone else. This is known as a zero-day, and only your SOC skills can determine whether it gets detected in time.

Once a vulnerability is made public, it is assigned a common Vulnerabilities and Exposure (CVE) number. From that moment, it's a race: attackers develop exploits while defenders rush to update their systems.

An answer to a CVE is always a patch- an update supplied by the software vendor. Even for zero days. You'll have to wait for a patch, vigilantly monitor for exploitation traces, and try to secure stressful period before the patch is released. For example, by:

- Restricting access to the system to only trusted Ips
- Applying temporary measures provided by the vendor
- Blocking known attack pattern on Ips or WAF

Example of Software Vulnerabilities

- SQL Injection
- Command Injection
- Buffer Overflow
- Cross-site Scripting (XSS)
- Remote Code Execution (RCE)

SOC Detection:

- WAF alerts
- Web server logs
- Unusual Post requests
- EDR alerts on servers

Operating System Vulnerabilities

Example

- Windows Vulnerabilities
- Linux Vulnerabilities
- Mac OS kernels or services Vulnerabilities etc.
- For example: Privilege escalation bugs, SMB vulnerabilities (EternalBlue), Kernel exploit etc.

SOC Detection:

- Sudden privilege change
- Exploit behavior in EDR
- Kernel driver loading alert etc.

Network Vulnerabilities

Example

- Open ports
- Weak Protocols (Telnet, FTP)
- Unpatched routers / Firewalls
- Exposed admin interfaces

SOC Detection

- IDS/IPS alerts
- Firewall logs
- Exercise authentication failures

Authentication and Authorization Vulnerabilities

Example

- Weak passwords
- No MFA
- Broken access control
- Privilege escalation paths

SOC Detection

- IAM alerts
- Access logs
- Role change events

Configured Dependent Vulnerabilities

Example

- Debug modes enabled
- Default Credentials
- Excessive permissions

SOC Detection:

- Admin panel exposed
- Default credentials unchanged
- System takeover

Zero-day vulnerabilities

A vulnerability which patch does not exist yet as it unknown to the vendor yet and are used by malicious actors like hackers.

SOC Detection

- Abnormal processes
- Suspicious network traffic
- Lateral movement

Life cycle of zero-day vulnerability

1. A new vulnerability is discovered.
2. A method to exploit vulnerability discovered.
3. Cyber Criminal leverage the vulnerability to cause damage.
4. Vulnerability discovered by the software vendors.
5. Patch releases by the software vendors.

Life cycle of zero-day Attack

1. A software program with an inherent security weakness or flaw is released. This defect is unknown to developers at the time of launch
2. An attacker discovered the vulnerability in the software program
3. The attacker takes advantage of the vulnerability and releases a malware into the software through social engineering techniques or phishing.
4. The attacker uses this opportunity to steal data from the infected system or may install spyware or a backdoor to enable future attacks
5. Once the attack is launched the developers detected and created a patch to mitigate it.

Common Misconfiguration for SOC

Cloud Misconfigurations

- Public S3 buckets
- Open Azure Blob Storage
- Over-permissive IAM roles
- Exposed access key

SOC Example

S3 bucket public → Sensitive data indexed by search engines → Data breach

Misconfiguration isn't a bug in the software but a mistake in how the system was set up, often by the IT team. These errors happen frequently usually to make things simpler.

Network Misconfigurations

- Firewalls allow all traffic
- Internal service exposed externally
- No network segmentation (All devices on same network)

SOC Example:

Internal DB port exposed→External Scan finds it→Database accessed

Identity Misconfigurations

- MFA Disabled for admins
- Excessive group memberships
- Shared admin accounts

SOC Example:

Compromised User → User already have admin privilege → No escalation needed

Endpoint misconfiguration

- Antivirus disabled
- Local admin access
- USB allowed everywhere

SOC Example

Malware executed → No EDR → Persistence Established

Most breaches didn't happen because of zero-days, they happen because of misconfigurations and unpatched vulnerabilities.

Major System Categories Used as Attack Vectors

What are endpoints?

- Laptops
- Desktop
- VDI (Virtual Desktop Infrastructure)
- Workstation

How attackers use them:

- Phishing email → malicious attachment
- Drive by downloads
- USB-based malware
- Exploiting unpatched software

Why are Endpoints dangerous?

Users have access

Users click things

Often weakest Security point

What is Endpoint Security?

The policies, processes and technology controls used to protect the confidentiality, Integrity and availability of an endpoint system.

Endpoints Security

- Anti malware
- Endpoint Security Control
- Data security
- Application Security
- IAM
- Acceptable use policies
- Configuration management
- System monitoring

Servers (Application/ Database/ File)

Types:

- Web server
- App server
- DB server
- File server

How attackers abuse servers:

- Exploiting
- Log4j
- SQL Injection
- RCE Vulnerabilities
- Weak admin credentials
- Exposed Services

Why servers are high value:

- Run Critical apps
- Store Sensitive Data
- Often Trusted Internally

Network Devices (Routers, Firewall, VPNs)

Devices:

- Routers
- Firewalls
- VPN Gateways
- Load balancers

How attacker abuse them:

- Default Credentials
- VPN vulnerabilities
- Misconfigured Firewall
- Firmware exploit

Initial Access: Compromised VPN Accounts → **Credential Access:** Utilizing DCsync and SECRETS Dump to harvest credentials → **Defense Evasion:** File Staged in directories, C:\programdata and c:\systemtest → **Discovery:** use of netapp.exe a variant of netscan → **Privilege Escalation:** Exploiting CVE 2020-1472 aka the Zerologon Privilege escalation vulnerability to create admin accounts → **Latteral Movement:** the use of WMIE.exe → **Execution persistence and C&C:** The abuse of anydesk.exe

Cloud Systems (Azure, AWS, GCP)

Common cloud attack vector:

- Public S3 buckets
- Exposed access keys
- Over-permissive IAM Roles
- Metadata service abuse

- Human errors
- Unmanaged Attack Surface
- Insufficient Access Management
- Misconfigured Cloud Storage
- Shared Technology weakness
- Insider threats

About Me

I'm an aspiring **SOC Analyst** and **Web Application Penetration Tester**, actively building my expertise in both defensive and offensive security. My journey involves hands-on learning with SIEM platforms (Splunk, Wazuh), EDR solutions (CrowdStrike, SentinelOne), and pentesting tools like Burp Suite, Nmap, and Nessus.

I'm passionate about transforming security alerts into actionable insights and uncovering vulnerabilities through structured methodologies like the OWASP Top 10. My goal is to contribute meaningfully to organizations by bridging the gap between threat detection and secure application development.

Let's Connect:

I regularly share insights, projects, and learning experiences on my professional platforms. Follow my journey and feel free to connect:

- **LinkedIn:** Usama Sani Khanzada | LinkedIn
- **GitHub:** Usama Sani

Your feedback, collaboration ideas, and questions are always welcome. Let's learn and grow together in the ever-evolving field of cybersecurity!