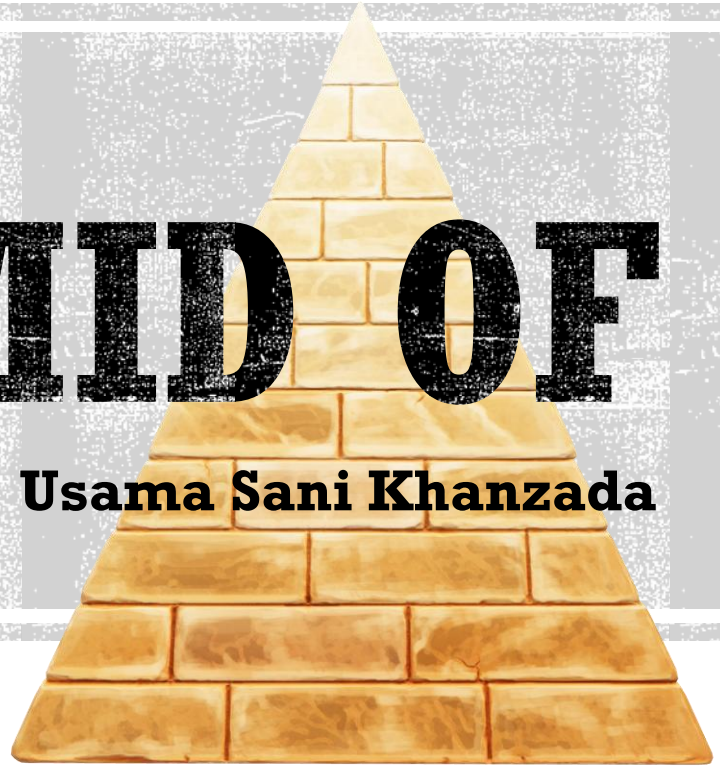
PYRAMID OF PAIN

Usama Sani Khanzada

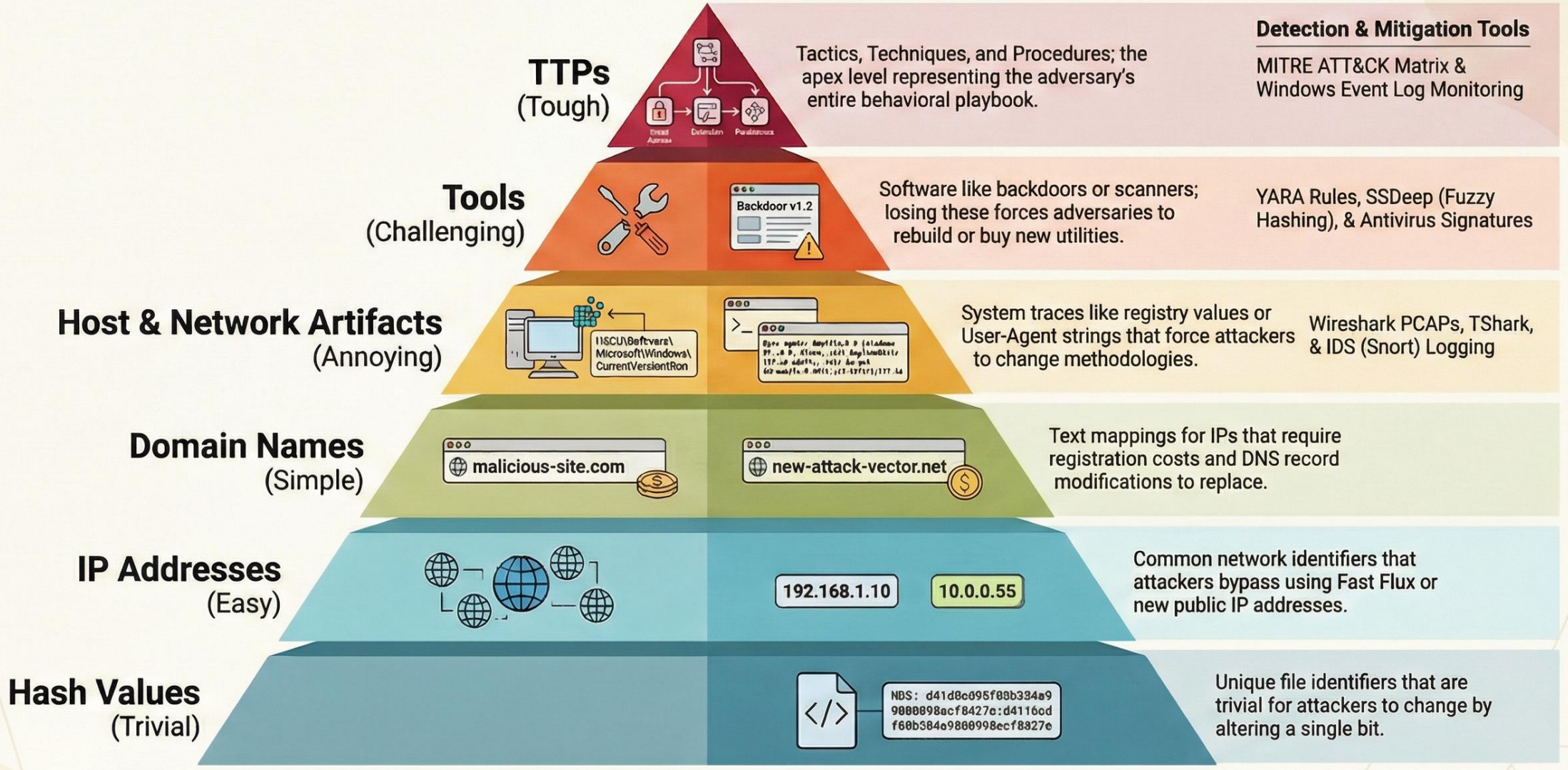


Source: Tryhackme

Visit: https://tryhackme.com

The Pyramid of Pain: Disrupting the Adversary

This conceptual model ranks security indicators by the difficulty for adversaries to change them after defenders detect and mitigate, improving Cyber Threat Intelligence (CTI).



INTRODUCTION: TURNING DETECTION INTO CONSEQUENCE

The Concept

What is the model?

The pyramid of pain is a conceptual model used in cyber threat intelligence (CTI) and Incident Response. It categorizes Indicators of Compromise (IOCs) not by detection difficulty, but by the pain inflicted on the adversary when those indicators are denied

The Goal

Shift from simply blocking bad data to dismantling the adversary's capabilities. Force them to expend time, money and development effort.

"The amount of pain you cause an adversary depends on the types of indicators you are able to make use of"

--David Bianco

Perspective Shift

Traditional Defense: "Can we detect this?" → Pyramid Strategy: "How much does this hurt the attacker"

1. HASH VALUE
PAIN LEVEL: TRIVIAL (ZERO PAIN FOR ADVERSARY)

Definition

A hash is a numeric fingerprint of fixed length uniquely identifying data.

- MD5 (128 bit): Fast, but prone to collisions. Not cryptographically secure
- SHA-1 (160 bit): Deprecated by NIST
- SHA-256 (256 bit): The modern standard

Defensive Tool: VirusTotal, MetaDefender (OPSWAT)

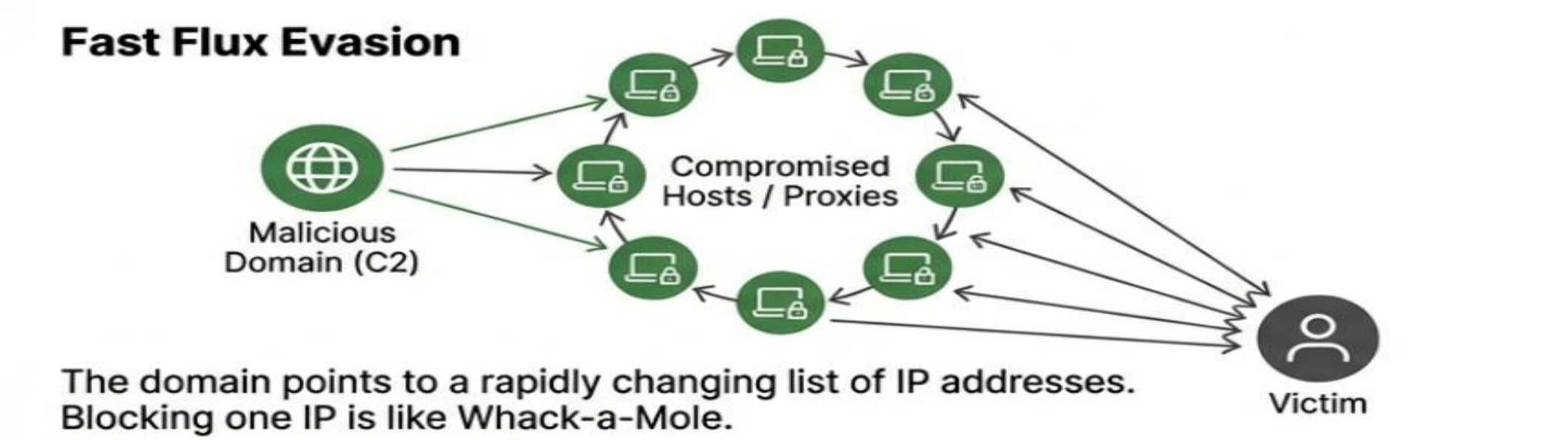
Why is it Trivial?

malware.exe  Hash A: D1A008E3 → command: echo "append" >> malware.exe

malware.exe  Hash B: 9D52B46F → (modified) changing a single bit of data results in completely different hash, bypassing signature-based detection instantly

3. IP ADDSRESSES
PAIN LEVEL: EASY(ADVERSARY RECOVERS INSTANTLY)

Logical address identifying devices. Blocking Ips is the most common defense, but attacker rarely use their own infrastructure. They utilize proxies, VPN's or Tor nodes



KEY TAKEAWAY BOX

Defensive Value: Good for short-term containment. Poor for long-term eradication.

4. DOMAIN
PAIN LEVEL: SIMPLE (REQUIRES ADMINISTRATIVE EFFORT)

Mapping Ips to strings (e.g. evilcorp.com) harder to change than IP because it requires registration and DNS notification. However, loose provider standards and API's make this automated and cheap

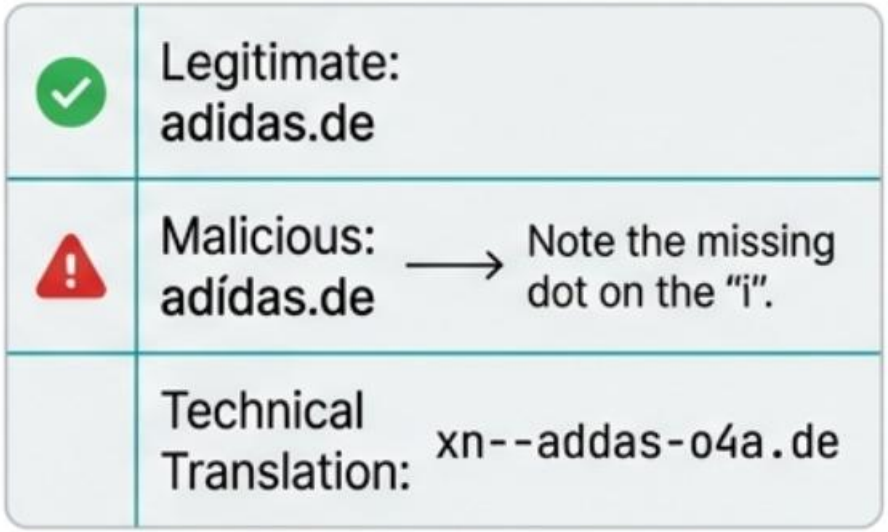
URL Shorteners

Attacker hide behind bit.ly or tinyurl.com

Detection Tip: Append '+' to the URL (eg. 'tinyurl.com/xyz+') to preview the destination.

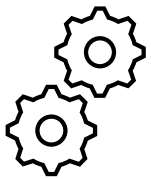
Adversary Tactic: Punycode

Using Unicode characters to mimic legitimate domains.



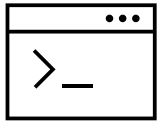
5 HOST ARTIFACTS
PAIN LEVEL: ANNOYING (REQUIRES MISCONFIGURATION)

Observable left specifically on the victim's system. Detection here forces the adversary to re-code or re-configure their installation methods.



Registry key
Persistence Mechanisms (e.g. run key services).

Drop Files
Malicious Executables (e.g. regidle.exe) or scripts in temp folder

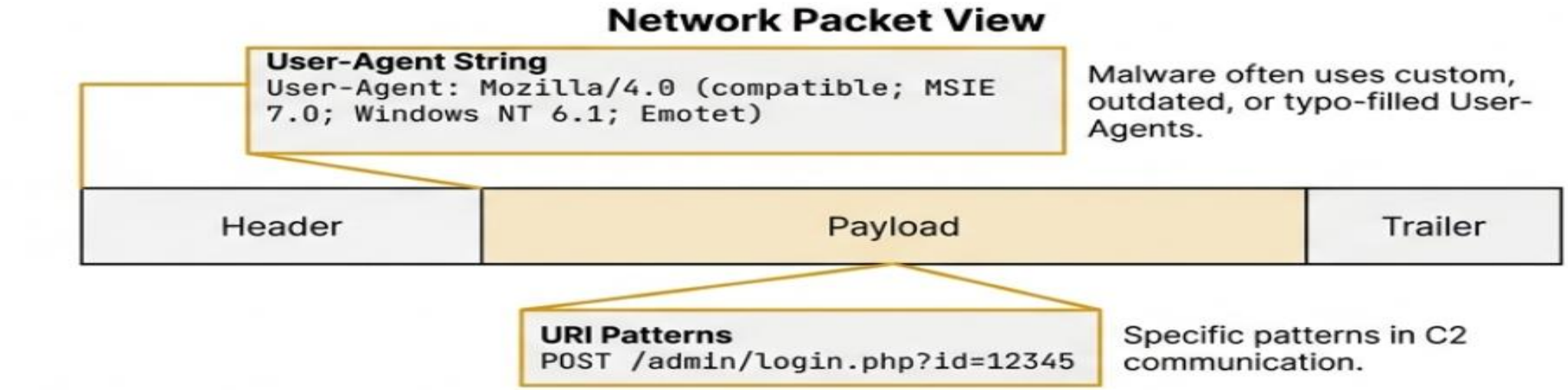


Process execution
Suspicious patterns, such as Microsoft word Spawning PowerShell

Pain factor
The attacker cannot just flip a bit. They must test a new configuration to ensure stability, slowing down their campaign

6. NETWORK ATRIFACTS
PAIN LEVEL: ANNOYING (DISRUPTS COMMUNICATION CHANNELS)

Observable generated by interaction between the attacker and victim ‘on the wire’. These often remain constant even if Ips change



Tools list:

- Wireshark/TShark: Analyze PCAP files
- Snort/Suricata: IDS rules targeting byte sequences

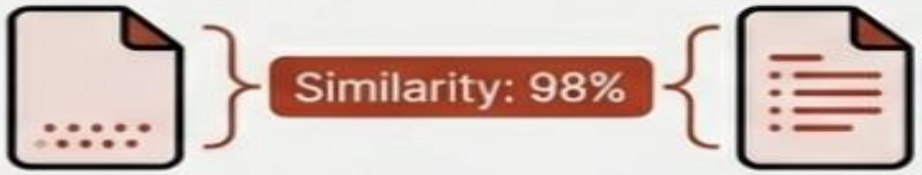
7 TOOLS
PAIN LEVEL: CHALLENGING (REQUIRES NEW DEVELOPMENT)

Detecting the software utilities themselves, not just their output example: Spearphishing generators, Backdoor, C2 Beacons (Cobalt Strike), Password Crackers.

The Pain: The adversary must crap the tool and build or learn a new one. This costs money and significant time.

Defensive Weapons

Fuzzy Hashing (SSDeep)



Detects similarity between files, catching variants of the same tool even if the standard hash changes.

YARA Rules

Pattern matching aimed at the code structure (the "DNA") of the malware family.

8. TTPs
PAIN LEVEL: TOUGH (EXISTENTIAL CRISIS FOR THE CAMPAIGN)

Tactics: The Goal (e.g. Exfiltration)

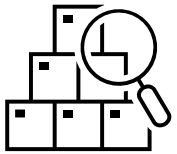
Techniques: How it is achieved (e.g. scheduled task)

Procedure: Specific Implementation Steps

If you detect the behavior, the adversary cannot simply code a new tool. They must re learn how to hack.

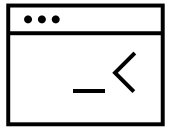
Most adversaries will give up and move to a softer target.

PRACTICAL APPLICATION PYRAMID IN ACTION



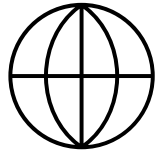
Threat Hunting

Start with the hypothesis based on TTPs (e.g. "PowerShell being used for lateral movement").
Map Findings back to the pyramid to understand impact



Detection Engineering

Audit your SIEM rules.
If 90% target IPs/Hashes, your defense is brittle.
Focus efforts on writing YARA rules (Tools) and behavioral analytics (TTPs)



Adversary Research

Analyze APT groups (e.g. Chimera). JetBrains Mono
Identify their behaviors.
Build defense that break their specific workflow, not just their temporary infrastructure.

CONCLUSION: THE STRATEGIC ADVANTAGE

"Don't just block indicators; dismantle the adversary's capacity to operate.“

Types vs. Cost/Pain

Indicator Types	Defender Cost	Attacker Pain
Hash, IP, Domain	Low (Automated)	Trivial (Annoyance)
Artifacts, Tools, TTPs	High (Requires Expertise)	Devastating (Operational Stop)

Evaluate your security posture. Move your defenses up the pyramid.

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!