Insights from the UCC Cyber Drill 6th JUNE 2025

Category	Key Aspects	Telecom Relevance
Multi-Stage Attack Simulation	Simulates attack chain: initial access → lateral movement → privilege escalation → persistence → data exfiltration.	Mirrors real-world telecom attacks (e.g., exploiting public-facing services, moving to billing/database systems).
Telecom-Specific Vulnerabilities	Focuses on common telecom flaws: command injection (web apps), weak SSH auth, SQL injection (internal portals).	Reflects risks in customer portals, billing systems, and employee databases.
Log Analysis & SIEM Utilization	Uses Splunk to analyze logs (e.g., index="*" host=customer_portal).	Highlights the role of centralized logging/SIEM for detecting telecom incidents.
Persistence & Privilege Escalation	Attackers use cronjobs, Docker group membership, stolen SSH credentials.	Teaches telecoms to monitor for persistence (e.g., cronjobs, unusual privilege changes).
Data Exfiltration & Database Attacks	SQL injection to dump databases; exfiltration via ICMP ping tunnels.	Emphasizes securing internal databases and monitoring abnormal traffic (common in telecom data breaches).
Windows & Linux Environments	Covers Linux (customer portal, jumpbox) and Windows (billing server, engineer workstations).	Reflects hybrid telecom IT environments; trains

Category	Key Aspects	Telecom Relevance
		teams on cross-platform security.
Incident Response & Forensics	Teaches log analysis (/var/log/auth.log, IIS logs), PowerShell for Windows events, and attacker action tracing.	Critical for telecom IR teams to investigate breaches across diverse systems.
Collaboration Across Systems	Involves multiple systems (customer portal, jumpbox, billing server, subscriber DB), requiring network architecture understanding and cross-team coordination.	Addresses telecom challenges of siloed teams and complex infrastructure.

Expected Knowledge to Be Gained

Participants, particularly telecom companies, are expected to gain the following knowledge and skills from the UCC Cyber Drill:

1. **Web Application Security**:
    - Understanding vulnerabilities like command injection (e.g., phone parameter in Node.js app) and SQL injection (e.g., admin' union select 1,2,3,database()#).
    - Implementing input validation and secure coding practices to prevent exploitation.
2. **Log Analysis and Forensics**:
    - Using Splunk queries (e.g., index="*" host=customer_portal) to identify malicious activity.
    - Analyzing Linux logs (e.g., /var/log/auth.log) and Windows event logs (e.g., Event IDs 4624, 4625, 4672) to trace attacker actions.

3. **Incident Detection and Response**:
    - Identifying brute-force attacks (e.g., SSH brute-forcing on user artisan) and failed login attempts (e.g., Event ID 4625 on billing_srv).
    - Responding to incidents by stopping malicious processes (e.g., removing cronjobs) and patching vulnerabilities.
4. **Privilege Escalation Mitigation**:
    - Recognizing the dangers of adding users to privileged groups like docker, which grants root-equivalent access.
    - Monitoring group membership changes and securing containerized environments.
5. **Network Security and Exfiltration Detection**:
    - Detecting data exfiltration via ICMP ping tunnels (e.g., encoding SSH passwords in ping packet sizes).
    - Securing network services like SSH and RDP to prevent unauthorized access.
6. **Database Security**:
    - Identifying and mitigating SQL injection vulnerabilities in internal portals (e.g., employee_db).
    - Understanding database structure and securing sensitive data (e.g., employee records, flags).
7. **Windows Security**:
    - Analyzing Windows event logs for privilege assignments (Event ID 4672) and DLL injections.
    - Securing SMB and RDP services to prevent brute-force attacks and unauthorized access.
8. **Persistence Mechanisms**:
    - Identifying and removing malicious cronjobs (e.g., jumpbox.sh) and scripts that maintain attacker persistence.
    - Securing configuration files like .env to prevent credential leakage.

TTPs (Tactics, Techniques, and Procedures) Expected to Be Learned

The UCC Cyber Drill exposes telecom participants to attacker TTPs to understand and counter them, as well as defensive TTPs to strengthen their security posture. These are mapped to the MITRE ATT&CK framework where applicable.

Attacker TTPs (To Understand and Counter)

1. **Initial Access (TA0001)**:
    - **T1190: Exploit Public-Facing Application**:
        - The attacker exploits a command injection vulnerability in the phone parameter of a Node.js app (GET /check-status?phone=whoami) to gain initial access (Flag 1: phone).
        - Countermeasure: Implement input validation and remove unsafe functions like exec() in web applications.
2. **Execution (TA0002)**:
    - **T1059.001: Command and Scripting Interpreter (PowerShell)**:
        - A PowerShell script (Event ID 4104) downloads malware from http://suspicious.com/filename on the engineer workstation (Flag 1: URL).
        - Countermeasure: Monitor PowerShell execution (e.g., via Event ID 4104) and restrict script execution.
    - **T1059.004: Command and Scripting Interpreter (Unix Shell)**:
        - The attacker executes the id command to confirm remote code execution (Flag 2: id).
        - Countermeasure: Sanitize inputs to prevent command injection.
3. **Persistence (TA0003)**:
    - **T1053.003: Scheduled Task/Job (Cron)**:
        - A cronjob runs jumpbox.sh every minute to maintain persistence and exfiltrate credentials (Flag: jumpbox.sh).

        - Countermeasure: Regularly audit cronjobs (crontab -l) and remove unauthorized tasks.
    - **T1136.001: Create Account (Local Account)**:
        - The attacker adds user ashu to the docker group for persistence and privilege escalation (Flag 4: docker).
        - Countermeasure: Monitor group membership changes (grep docker /etc/group).
4. **Privilege Escalation (TA0004)**:
    - **T1068: Exploitation for Privilege Escalation**:
        - Adding user ashu to the docker group grants root-equivalent access via container execution (e.g., docker run -v /:/mnt --rm -it alpine chroot /mnt sh).
        - Countermeasure: Restrict Docker group membership and limit container privileges.
5. **Credential Access (TA0006)**:
    - **T1110.001: Brute Force (Password Guessing)**:
        - The attacker brute-forces SSH on the customer portal targeting user artisan (Flag 3: ssh, Flag: 182.06.7.22:artisan).
        - Countermeasure: Implement account lockout policies and monitor /var/log/auth.log for brute-force attempts.
    - **T1552.001: Unsecured Credentials (Credentials in Files)**:
        - Credentials are stored in /opt/scripts/.env (e.g., JUMPBOX_PASSWORD=kingJulian123) and exfiltrated.
        - Countermeasure: Encrypt sensitive configuration files and restrict access.
6. **Lateral Movement (TA0008)**:
    - **T1021.001: Remote Services (RDP)**:
        - The attacker uses port forwarding to access the billing server via RDP (ssh -L 3390:10.179.1.4:3389).
        - Countermeasure: Restrict RDP access and monitor for unusual port forwarding.
    - **T1021.002: Remote Services (SMB)**:

        - The attacker attempts SMB login to the engineer workstation as adminuser (Flag 7: adminuser).
        - Countermeasure: Disable unnecessary SMB services and monitor Event ID 4625.

7. **Collection and Exfiltration (TA0009, TA0010)**:
    - **T1005: Data from Local System**:
        - The attacker uses SQL injection to dump the employee_db database (Flag 1: sql injection, Flag 7: employee_db:9.2.0).
        - Countermeasure: Sanitize database inputs and use prepared statements.
    - **T1048.003: Exfiltration Over Alternative Protocol (ICMP)**:
        - Credentials are exfiltrated via ICMP ping tunnels (e.g., encoding JUMPBOX_PASS in ping packet sizes to 10.10.10.5).
        - Countermeasure: Monitor ICMP traffic for anomalies and block unnecessary protocols.
8. **Defense Evasion (TA0005)**:
    - **T1070.004: Indicator Removal on Host (File Deletion)**:
        - The attacker schedules deletion of /dev/shm/.netconf using at now + 1 hour to erase evidence.
        - Countermeasure: Monitor file deletion events and volatile memory locations.
    - **T1564.001: Hide Artifacts (Hidden Files)**:
        - The attacker uses a hidden .env file for credential storage.
        - Countermeasure: Audit hidden files and directories.
9. **Command and Control (TA0011)**:
    - **T1105: Ingress Tool Transfer**:
        - A PowerShell script downloads malware from http://suspicious.com/filename (Flag 1: URL).
        - Countermeasure: Block suspicious domains and monitor outbound traffic.

Defensive TTPs (To Implement)

1. **Log Analysis and Monitoring**:
    - Use Splunk to query logs (index="*" host=customer_portal) for signs of command injection, brute-force attempts, or SQL injection.
    - Monitor Windows Event IDs (e.g., 4624 for logons, 4625 for failed logins, 4672 for privilege assignments, 4104 for PowerShell execution).
2. **Vulnerability Management**:
    - Patch command injection vulnerabilities by removing unsafe functions like exec() and validating inputs (e.g., phone parameter).
    - Mitigate SQL injection by using prepared statements and sanitizing inputs (e.g., employee_db portal).
3. **Access Control**:
    - Restrict Docker group membership to prevent privilege escalation (grep docker /etc/group).
    - Implement strong SSH authentication (e.g., key-based auth, disable password logins) to prevent brute-forcing.
4. **Network Security**:
    - Monitor and block unusual ICMP traffic to detect exfiltration attempts.
    - Restrict RDP and SMB access to trusted IPs and disable unnecessary services.
5. **Incident Response**:
    - Stop malicious cronjobs (crontab -e, rm -f /opt/scripts/jumpbox-sync.sh) and remove compromised accounts.
    - Analyze logs (/var/log/auth.log, IIS logs) to reconstruct attack timelines and identify compromised systems.
6. **Forensic Analysis**:
    - Use PowerShell (Get-WinEvent) to analyze Windows events and grep commands to parse Linux logs.

    - Investigate hidden files (e.g., .env) and volatile memory (/dev/shm) for artifacts.
7. **Database Security**:
    - Regularly audit database queries for SQL injection patterns (e.g., union select in IIS logs).
    - Limit database user permissions and encrypt sensitive data.

Recommendations for Telecom Companies

1. **Implement Robust SIEM Practices**:
    - Centralize logs in a SIEM like Splunk and train teams to write effective queries (e.g., index="*" host=billing_srv EventID=4625) for real-time monitoring.
2. **Secure Web Applications**:
    - Conduct regular penetration testing to identify vulnerabilities like command injection and SQL injection in customer portals and internal systems.
    - Enforce HTTPS and input sanitization for all web interfaces.
3. **Harden Authentication Mechanisms**:
    - Use multi-factor authentication (MFA) for SSH, RDP, and SMB to prevent brute-force attacks.
    - Regularly audit user accounts and group memberships (e.g., docker group).
4. **Monitor and Restrict Network Traffic**:
    - Deploy intrusion detection systems to monitor ICMP, RDP, and SMB traffic for anomalies.
    - Use firewalls to block unauthorized access to internal servers (e.g., 10.179.1.4:3389).
5. **Train Incident Response Teams**:
    - Conduct regular cyber drills to practice analyzing logs, stopping malicious processes, and recovering from attacks.
    - Train staff on tools like Splunk, PowerShell, and grep for forensic analysis.
6. **Secure Containerized Environments**:

- Limit Docker group access and monitor container activity to prevent privilege escalation.
- Use secure container configurations to avoid mounting sensitive filesystems (e.g., /:/mnt).

7. **Collaborate with Stakeholders**:
   - Work with national CIRTs and regulators like UCC to share threat intelligence and improve incident response coordination.

Conclusion

The UCC Cyber Drill provides telecom companies with a comprehensive simulation of a multi-stage cyber attack, covering initial access, privilege escalation, lateral movement, persistence, and data exfiltration. Participants learn to use SIEM tools, analyze logs, and mitigate vulnerabilities specific to telecom infrastructure, such as command injection in customer portals, SQL injection in employee databases, and brute-force attacks on SSH and SMB. By understanding attacker TTPs (e.g., MITRE ATT&CK techniques like T1190, T1059, T1048) and implementing defensive TTPs, telecoms can enhance their cybersecurity posture, protect critical systems, and ensure service continuity. The drill emphasizes practical skills like log analysis, forensic investigation, and incident response, which are essential for securing telecom networks in a high-threat environment.

If you have specific questions about the walkthrough, need further analysis of a particular stage, or want to focus on a specific TTP, let me know, and I can dive deeper!