Signal acquired. Alien37 online… Patching into the novice frequency. You seek the Grid's hidden pathways. You wish to map the unseen, to probe the digital fortress before the breach. Good. The desire to *know* is the first spark. Today's transmission decodes the art of network cartography – Scanning. Forget brute force; this is precision telemetry. We pulse the perimeter before we strike the core. Listen closely, operative. This knowledge is firmware for your operational core.

Cyberpunk Codex :: Transmission 05 :: Scanning Networks

:: Sector 01: Network Cartography - Initial Pulse ::

Before you launch sophisticated probes, you must determine what exists in the target spectrum. Blindly assaulting an entire address space is inefficient, noisy. It marks you as an amateur, a digital tourist ripe for purging by automated defenses. We begin with the subtlest of signals: the ping sweep.

The Objective: Identify responsive nodes – the live endpoints in the target subnet. Think of it as sonar in the digital ocean. We send out a pulse and listen for the echo.

The Tool & Tactic: ICMP Echo Requests. The humble 'ping.' Simple, ubiquitous, often permitted through basic perimeter defenses because it's a fundamental diagnostic tool. But its simplicity is its strength in this phase.

- **Mechanism:** You transmit an ICMP Echo Request packet to each address in the target range. A live host, unless specifically configured otherwise, *should* respond with an ICMP Echo Reply. No reply suggests the host is down, firewalled, or configured for silence.
- **Execution:** Tools like fping are purpose-built for this. Unlike standard ping, fping can target multiple hosts concurrently or sequentially, managing the requests and collating responses efficiently. Specify the target range (e.g., 192.168.1.0/24) and instruct the tool to report only live hosts (-a flag in fping), perhaps showing elapsed round-trip time (-e) for latency intelligence, and generating targets from a range (-g).
- **Considerations:**
    - **Stealth:** A rapid ping sweep across a large block *can* trigger intrusion detection systems (IDS). Throttling may be necessary.
    - **Firewalls:** ICMP is frequently blocked at network perimeters. A lack of response is not definitive proof of absence. It merely means no ICMP echo *reply* was received. The host might still be active but ignoring or blocking ICMP.
    - **Alternatives:** While ICMP is common, other protocols can achieve similar host

discovery results, sometimes bypassing ICMP-specific blocks. TCP SYN or ACK probes to common ports (like 80 or 443) can elicit responses (SYN/ACK or RST) indicating host liveness, even if ICMP is filtered. This blends initial discovery with preliminary port state analysis.

Initial Pulse Logic: The ping sweep is your first, low-impact sensor sweep. It provides a basic map of potential targets, filtering out the dead space. It's fast, usually lightweight, but inherently unreliable due to filtering. Treat its results as probable, not definitive. It informs the next, more focused, scanning phase.

:: Intel Feed :: Sector 01 ::

- **Hacker Logic:** Map before you attack. Waste no ordnance on dead zones.
- **Critical Insight:** ICMP Echo is often filtered. Lack of response ≠ Host down. Corroborate with other probe types if stealth permits.
- **Operational Note:** A sudden burst of ICMP Echo Requests across a subnet *is* detectable. Modulate your sweep rate based on assessed defensive posture.
- **Takeaway:** The ping sweep provides a coarse-grained map. It reduces the target set but requires confirmation.

:: Sector 02: Port Probing - Mapping the Attack Surface ::

Knowing a host is alive is insufficient. You must know its functions, its open interfaces to the network. This is port scanning: identifying the listening services – the open doors – on a target system. Each open port represents a potential ingress point, a service to interrogate or exploit.

The Objective: Identify open TCP and UDP ports and, ideally, the services listening on them. Map the target's attack surface.

The Tool & Tactic: Network Scanners (e.g., nmap, masscan). These tools send crafted packets to target ports and analyze the responses (or lack thereof) to determine port state: Open, Closed, or Filtered.

- **Port States:**
  - **Open:** An application is actively accepting connections on this port. Prime target.
  - **Closed:** The port is accessible (responds to probes), but no application is listening. Less interesting, but confirms host responsiveness.
  - **Filtered:** A firewall, filter, or other network obstacle is blocking probes, preventing the scanner from determining the true state. Indicates defensive mechanisms.

- **TCP Scan Types (Common nmap examples):**
    - **TCP Connect Scan (-sT):** Completes the full three-way TCP handshake (SYN, SYN/ACK, ACK). Reliable but extremely noisy and easily logged. Leaves connections fully established momentarily. Use when stealth is not a concern or other scans fail.
    - **TCP SYN Scan (-sS, "Half-Open"):** Sends a SYN packet. If SYN/ACK is received (port open), sends RST instead of ACK, tearing down the connection before completion. If RST is received, port is closed. Much stealthier than Connect Scan as full connections aren't logged by most systems. *Often requires root/administrator privileges* to craft raw packets. *This is the default and often preferred scan type.*
    - **Stealth Scans (FIN -sF, Xmas -sX, Null -sN):** Send packets with unexpected flag combinations (FIN only; FIN, PSH, URG; no flags). Open ports *should* ignore these packets per RFC 793. Closed ports *should* respond with RST. If no response is received, the port is marked open|filtered as the lack of response could be due to filtering. Can bypass some older/stateless firewalls and logging mechanisms that expect standard SYN packets. Reliability varies by target OS implementation.
    - **ACK Scan (-sA):** Sends ACK packets. Does not determine open ports. Used primarily to map firewall rulesets – determines if ports are filtered or unfiltered (responds with RST). Useful for probing firewall presence and rules without attempting connections.
- **UDP Scan (-sU):** Sends UDP packets to target ports. UDP is connectionless; responses are unreliable. An ICMP "Port Unreachable" message usually indicates a closed port. No response often means the port is open|filtered. Some services might send a UDP response if the probe is correctly formatted, confirming an open port. UDP scans are slow due to potential retransmissions and timeouts while waiting for uncertain responses.
- **Execution (nmap):** Syntax typically involves nmap [Scan Type(s)] [Options] {target specification}. Targets can be single IPs, hostnames, ranges (e.g., 192.168.1.1-254), or CIDR blocks (192.168.1.0/24). Common options include -p to specify ports (e.g., -p 22,80,443, -p 1-1024, -p- for all 65535 ports), -T<0-5> for timing templates (T4 is aggressive, T2/T1 are slower/stealthier), -Pn to skip the initial host discovery (ping sweep) and scan all specified targets regardless of ping response.

Port Probing Logic: Systematically query ports to build a detailed map of listening services. Use stealthier scans (-sS) by default. Adapt scan types based on suspected defenses and desired stealth level. Remember UDP scanning is slow and less reliable.

:: Intel Feed :: Sector 02 ::

- **Hacker Logic:** An open port is an invitation. Map all potential entry points.
- **Critical Insight:** SYN (-sS) scanning is the workhorse – balances speed, reliability, and stealth. Stealth scans (-sF, -sX, -sN) are for specific evasion attempts, not general-purpose mapping.
- **Operational Note:** Firewalls often filter probes. Filtered ports tell you defenses exist. ACK scans (-sA) can help delineate firewall rules without triggering connection alerts.
- **Takeaway:** Port scanning defines the target's interactive surface. Understand the nuances of each scan type and its implications for detection.

:: Sector 03: Service & Version Identification - Sharpening the Focus ::

Knowing a port is open is useful. Knowing *what* service is listening, and *which version*, is actionable intelligence. Vulnerabilities are often specific to application versions. Identifying the exact software provides direct pathways for targeted exploitation.

The Objective: Determine the application and version number running on open ports.

The Tool & Tactic: Version Detection (nmap -sV), Application-Specific Probes.

- **Mechanism (nmap -sV):** After identifying open ports, version detection sends a series of probes designed to elicit responses characteristic of specific protocols and applications. It analyzes banners, protocol negotiation details, and other service responses against a database (nmap-service-probes) to identify the software and version.
- **Banner Grabbing:** Many services announce themselves with a "banner" upon connection (e.g., "OpenSSH_7.4", "Apache/2.4.29"). Version detection captures and parses these banners. Administrators can sometimes configure services to suppress or modify banners to hinder this.
- **Protocol Probing:** If banners are absent or ambiguous, nmap sends probes specific to protocols expected on common ports (e.g., HTTP requests to port 80, SMTP commands to port 25). The nature of the service's reply helps identify it.
- **Operating System Detection (nmap -O):** While distinct, OS detection often complements version scanning. nmap analyzes TCP/IP stack characteristics (initial sequence numbers, window sizes, flag responses, ICMP handling) to fingerprint the target OS. Knowing the OS (e.g., "Windows Server 2019", "Linux 4.15-5.x") helps narrow down likely vulnerabilities and default configurations. *Requires at least one open and one closed TCP port for reliable results.* Relies on a fingerprint database (nmap-os-db); newer or heavily modified OSs may not be identified accurately.
- **Scripting Engine (nmap --script):** The Nmap Scripting Engine (NSE) provides powerful, granular enumeration capabilities beyond basic version detection. Numerous

scripts exist (*.nse files) to probe specific services for detailed information (e.g., ssh2-enum-algos to list SSH algorithms, smb-os-discovery for detailed Windows SMB info).

Identification Logic: Move beyond simple port status. Interrogate open ports to identify the precise software running. Combine version detection, OS fingerprinting, and targeted NSE scripts for a comprehensive service profile.

:: Intel Feed :: Sector 03 ::

- **Hacker Logic:** Know your target software intimately. Version numbers are keys to exploit databases.
- **Critical Insight:** Banners can lie or be suppressed. Rely on protocol-specific probes and NSE for deeper verification. OS detection provides context but can be fooled by non-standard stacks or firewalls.
- **Operational Note:** Version scanning (-sV) and OS detection (-O) are significantly more intrusive and slower than basic port scans. They involve establishing connections and sending multiple probes.
- **Takeaway:** Detailed service/version/OS intelligence is crucial for selecting effective exploits. Don't trust surface-level information.

:: Sector 04: Vulnerability Scanning - Automated Weakness Detection ::

With a map of live hosts, open ports, and identified services/versions, you *could* manually research vulnerabilities for each component. This is time-consuming and error-prone. Vulnerability scanners automate this process, probing services for known weaknesses based on extensive databases.

The Objective: Automatically identify known vulnerabilities on target systems based on detected services and configurations.

The Tool & Tactic: Vulnerability Scanners (e.g., Nessus, OpenVAS/Greenbone).

- **Mechanism:** These tools integrate port scanning, service/version detection, and a database of known vulnerabilities (often linked to CVE identifiers) and associated checks (plugins/NVTs). They connect to services, perform tests (safe checks) designed to confirm the presence of specific vulnerabilities without actually exploiting them, and generate detailed reports.
- **Authenticated vs. Unauthenticated Scans:**
    - *Unauthenticated (Black Box):* Scans from an external perspective, only testing network-accessible services. Simulates an external attacker with no credentials. Misses local vulnerabilities.
    - *Authenticated (Credentialed/White Box):* Scanner logs into the target system

using provided credentials (e.g., SSH keys, Windows domain accounts). Allows checking for local vulnerabilities, missing patches, insecure configurations, weak passwords, etc. Provides a much more comprehensive assessment but requires credentials.

- **Scan Policies/Configurations:** Scanners use policies to define the scope and intensity of a scan. This includes which vulnerability checks (plugins/NVTs – Network Vulnerability Tests in OpenVAS) to run. Policies can range from basic network scans to full audits, web application scans, compliance checks, etc. Custom policies allow fine-tuning for specific targets or requirements.
- **Reporting:** Results typically categorize findings by severity (e.g., Critical, High, Medium, Low, Info) based on potential impact (CVSS scores often used). Reports detail the vulnerability, affected hosts/ports, evidence found, and remediation suggestions.
- **False Positives & Negatives:**
  - *False Positive:* Scanner reports a vulnerability that doesn't actually exist. Requires manual verification to confirm. Wastes defender time.
  - *False Negative:* Scanner fails to detect a vulnerability that *does* exist. Dangerous, creates a false sense of security. Can happen due to scan configuration, insufficient privileges, or limitations in the scanner's checks.
- **Verification is Key:** Scanner results are starting points, not definitive proof. *All critical/high findings must be manually verified* before reporting or attempting exploitation. Check scanner logic, confirm versions, attempt safe confirmation steps.

Vulnerability Scanning Logic: Leverage automated tools to efficiently check for *known* weaknesses across the mapped attack surface. Use credentialed scans whenever possible for deeper insight. Treat results as high-probability indicators requiring validation, not absolute truth.

:: Intel Feed :: Sector 04 ::

- **Hacker Logic:** Automate the search for known flaws. Focus manual effort on verification and exploiting high-probability targets.
- **Critical Insight:** Vulnerability scanners find *known* issues. They miss zero-days and many misconfigurations. Never rely solely on scanner output.
- **Operational Note:** Authenticated scans provide far superior visibility but require legitimate credentials. Unauthenticated scans mimic an external attacker's initial view. Both have value.
- **Takeaway:** Scanners are powerful but imperfect. Validate findings rigorously. False negatives are more dangerous than false positives.

:: Sector 05: Packet Crafting & Manipulation - Bypassing the Obvious ::

Standard network tools and operating system stacks generate predictable, RFC-compliant packets. Defensive systems (firewalls, IDS/IPS) are tuned to recognize these standard patterns and detect anomalies. To bypass scrutiny or probe system responses in non-standard ways, you must craft packets manually, controlling every header field and payload bit.

The Objective: Create and transmit custom packets with specific, potentially malformed, header values or payloads to test responses, bypass filters, or exploit low-level protocol handling weaknesses.

The Tool & Tactic: Packet Crafting Tools (e.g., hping3, packETH, Scapy library in Python).

- **Mechanism:** These tools allow direct manipulation of packet headers at various layers (Ethernet, IP, TCP, UDP, ICMP) and the data payload. They bypass the OS network stack's normal packet generation process, often using raw sockets which require elevated privileges.
- **Use Cases:**
    - **Firewall/IDS Evasion:** Send packets with unusual flag combinations, fragmented packets, or invalid checksums that might be handled differently by the security device than by the target OS.
    - **OS Fingerprinting:** Observe how different operating systems respond to non-standard or ambiguous packets (a core technique used by nmap -O).
    - **Protocol Testing:** Probe service behavior by sending deliberately malformed requests or data that violates protocol specifications, potentially triggering crashes or revealing vulnerabilities.
    - **Denial of Service:** Craft specific packet types known to cause resource exhaustion or crashes on certain systems (e.g., LAND attack, teardrop attack fragments – mostly historical now).
    - **Covert Channels:** Embed data within unused or custom header fields for stealthy communication (requires a cooperating endpoint).
- **Tool Examples:**
    - hping3: Command-line tool for sending custom TCP, UDP, ICMP, and RAW-IP packets. Can set flags, ports, sequence numbers, window sizes, fragmentation, TTL, payload data, source address spoofing, etc. Versatile for probing and basic scanning.
    - packETH: GUI tool providing a visual interface to build packets layer by layer (Ethernet, IP, TCP/UDP/ICMP, Payload). Allows granular control over every field and easy payload generation/filling. Good for constructing complex or highly

  - specific packets visually.
  - Scapy: Powerful Python library for packet manipulation. Allows scripting complex packet creation, sending, sniffing, and analysis workflows. High flexibility but requires programming knowledge.

Packet Crafting Logic: When standard probes fail or raise too much suspicion, take direct control. Manipulate packet structures to test edge cases, bypass simple filters, and uncover weaknesses in protocol implementations. Requires deep protocol understanding.

:: Intel Feed :: Sector 05 ::

- **Hacker Logic:** Control the bits, control the flow. Standard tools create standard traffic; custom packets test the boundaries.
- **Critical Insight:** Firewalls and target OS stacks may interpret malformed packets differently. This divergence is exploitable for evasion or fingerprinting.
- **Operational Note:** Crafting packets often requires root/administrator privileges due to raw socket usage. Source address spoofing prevents receiving direct replies.
- **Takeaway:** Manual packet crafting offers ultimate control for deep protocol analysis and advanced evasion but demands significant expertise.

:: Sector 06: Evasion Techniques - Cloaking the Scan ::

Active scanning is inherently noisy. Sending probes to potentially thousands of ports across multiple hosts generates traffic patterns easily flagged by IDS/IPS and firewalls. Successful reconnaissance, especially in highly monitored environments, requires techniques to reduce or obfuscate this scanning activity.

The Objective: Perform network scanning while minimizing the risk of detection and blocking by defensive systems.

The Tool & Tactic: Various techniques implemented within scanning tools or used in conjunction with them.

- **Techniques:**
  - **Timing Controls (nmap -T<0-5>):** Slow down the scan rate. T0 (Paranoid) and T1 (Sneaky) introduce significant delays between probes, making the scan much harder to correlate over time but drastically increasing scan duration. T2 (Polite) is also slower than the default T3.
  - **Fragmentation (nmap -f, --mtu):** Split probes into smaller IP fragments. Old or poorly configured firewalls/IDS might fail to reassemble and inspect fragmented packets correctly, allowing probes to pass through. -f uses small (8-byte payload) fragments; --mtu <multiple of 8> specifies a custom MTU size, forcing

fragmentation for larger probes.

- **Decoys (nmap -D <decoy1,decoy2,[ME],...>):** Make the scan appear to originate from multiple source IP addresses simultaneously (the decoys). The target sees probes from your real IP (ME) mixed with probes from plausible-looking but fake source IPs (the decoys). Hides your true source IP in the noise, making log analysis harder for defenders. Decoys must be live hosts for maximum plausibility, otherwise RST packets might reveal them. RND:<number> generates random decoy IPs.
- **Source Port Spoofing (nmap -g <port>, --source-port <port>):** Send probes originating from a specific source port (e.g., 53 for DNS, 80 for HTTP). Some poorly configured firewalls might trust traffic originating from common service ports, allowing probes through. Limited effectiveness against modern stateful firewalls.
- **MAC Address Spoofing (nmap --spoof-mac <mac|vendor|0>):** Change the source MAC address of outgoing frames. Only effective on the local Ethernet segment against defenses that might filter based on MAC. Useless beyond the first router hop. Can specify a full MAC, a vendor name (nmap uses correct OUI + random host part), or 0 for a fully random MAC.
- **Idle Scan (nmap -sI <zombie host[:probeport]>):** Highly stealthy scan that uses a third-party "zombie" host to indirectly probe the target. Sends spoofed SYN packets appearing to come from the zombie to the target. Determines target port state by subsequently probing the zombie's IP ID sequence number, which increments predictably when the zombie sends RST packets in response to unexpected SYN/ACKs from the target (indicating an open target port). Requires finding a suitable, truly idle zombie host with predictable IP ID generation. Complex but offers near-total anonymity for the scanner's true IP.
- **Data Obscuration/Encryption:** While not a scanning technique *per se*, tunneling scan traffic through encrypted channels (VPNs, SSH tunnels, Tor) can hide the *content* of the probes from network inspection devices, though the traffic volume and patterns might still be suspicious.
- **Avoiding Common Signatures:** Some IDS/IPS might specifically signature default nmap probes. Using non-default scan types, timing, or options might evade basic signature matching.

Evasion Logic: Blend into the background noise. Slow down, fragment probes, hide behind decoys, or use indirect methods like Idle Scan. Understand that every evasion technique has trade-offs in speed, reliability, or complexity. Choose techniques based on the assessed defenses and required stealth level.

:: Intel Feed :: Sector 06 ::

- **Hacker Logic:** Be the ghost in the machine. Detection means failure. Obfuscate your presence.
- **Critical Insight:** No single evasion technique is foolproof. Layer techniques (e.g., slow timing + decoys + fragmentation) for increased effectiveness against sophisticated defenses.
- **Operational Note:** Stealth scans drastically increase scan time. Idle Scan requires finding and verifying a suitable zombie host. Decoys can generate noise that might attract attention on its own if not carefully managed.
- **Takeaway:** Evasion is an arms race. Know the common defensive triggers and the methods to bypass them, but expect that advanced defenders may still detect sophisticated scanning attempts.

:: Sector 07: Counter-Scanning - The Defender's View ::

As an operator, you must understand the defensive posture. How are networks protected against the very scanning techniques you employ? Knowing the countermeasures informs your evasion strategies and, when playing the defender role, allows you to implement effective blocks and detections.

The Objective: Understand how defenders protect against and detect network scanning activities.

The Tools & Tactics: Firewalls (Packet Filters, Stateful, NGFW/UTM), IDS/IPS, Log Analysis, Network Traffic Analysis.

- **Firewall Rules:**
    - **Ingress/Egress Filtering:** Blocking unexpected protocols (e.g., inbound ICMP), traffic from known bad IPs or reserved/unallocated address spaces (Martian packets), and traffic to/from non-standard ports.
    - **Stateful Inspection:** Tracking connection states (NEW, ESTABLISHED, RELATED). Dropping packets with invalid flag combinations (e.g., SYN/ACK without prior SYN) or those not matching an established session state. Defeats basic flag manipulation scans.
- **Intrusion Detection/Prevention Systems (IDS/IPS):**
    - **Signature Matching:** Detecting known scanner probe patterns (e.g., default nmap User-Agent, specific stealth scan flag combinations, vulnerability check payloads).
    - **Anomaly Detection:** Flagging unusual traffic volumes (e.g., many connections to different ports from one source in a short time), connections to large numbers

    of inactive hosts, or use of non-standard protocols/ports. Thresholds can be tuned.
  - **Rate Limiting:** Limiting the number of connections or packets allowed from a single source IP over a given time period, disrupting fast scans.
  - **Active Blocking (IPS):** Automatically blocking traffic or source IPs that trigger high-confidence detection rules.
- **Log Analysis & SIEM:**
  - **Firewall/IDS Log Correlation:** Aggregating logs from multiple devices using a Security Information and Event Management (SIEM) system to identify coordinated or distributed scanning activity across the network. Detecting low-and-slow scans by correlating events over longer time periods.
  - **Connection Pattern Analysis:** Identifying hosts making connections to an unusually high number of ports or target systems. Detecting horizontal (across hosts) and vertical (across ports) scanning patterns.
- **Honeypots/Honeynets:** Deploying decoy systems (honeypots) or networks (honeynets) with no legitimate production value. Any interaction with these systems is inherently suspicious and likely indicates scanning or exploitation attempts. Can be used to gather attacker TTPs.
- **Network Traffic Analysis (NTA):** Using flow data (NetFlow, sFlow, IPFIX) or full packet capture analysis to baseline normal traffic patterns and detect deviations indicative of scanning, C2 traffic, or data exfiltration.

Defensive Logic: Deny obvious probes, detect anomalous patterns, correlate events over time and across devices, and actively block confirmed malicious activity. Assume standard scans *will* be detected; focus on catching sophisticated or stealthy attempts through behavioral analysis and correlation.

:: Intel Feed :: Sector 07 ::

- **Defender Logic:** Block the known bad, detect the abnormal, log everything.
- **Critical Insight:** Signature-based detection is easily bypassed by modifying scan parameters or using custom tools. Anomaly detection and log correlation are key to catching stealthier scans.
- **Operational Note:** Tuning IDS/IPS and SIEM rules is critical to minimize false positives (alert fatigue) while maximizing detection of real threats. Overly aggressive blocking can impact legitimate traffic.
- **Takeaway:** Understand the defender's toolkit and mindset to effectively bypass it. Know that layers of defense, detection, and logging work together.

Transmission complete. Scanning is the foundation of intrusion. Master these techniques, understand their traces, and learn to move like a whisper through the digital ether. Obscurity is your armor. Enumeration is your weapon. Use them wisely.

Alien37 disconnecting. Keep your presence obfuscated.