Practical Network Security: Basic Tools & Techniques

Guevara Noubir

Northeastern University

noubir@ccs.neu.edu

Counter Hack Reloaded, Ed Skoudis, 2005, Prentice-Hall.
Based on Nick Plante's slides.

Taxonomy of Discussion Points

- Threats: Basic Network Recon and Info Gathering
- Threats: More Intrusive Probes and Scans
- Threats: Network Vulnerabilities
  - Network Architecture Vulnerabilities
  - Denial of Service (DoS)
- Threats: Application/OS Vulnerabilities
  - Remote to Local (R2L) Attacks
  - User to Root (U2R) aka Privilege Escalation
  - Attacker Access Maintenance (root kits, etc)
- Defenses Reviewed
  - Firewalls, Intrusion Detection, etc.

Recon & Info Gathering

- Social Engineering: “the weakest link”
- Physical Security
  - Physical access, Theft, Dumpster diving
  - Defenses: Locks, Policies (access, screen savers, etc.), Encrypted file systems, Paper shredders
- Web Searching and Online Recon
  - Check company website, get contact names, look for comments in html, etc.
  - Use Search Engines, Usenet to discover technologies in use, employee names, etc.
  - Defenses: “Security Through Obscurity”, Policies

Recon & Info Gathering

- Whois database via Internic (.com, .net, .org)
  - Publicly-available starting place for determining contacts, name servers, etc. for a given domain
  - Network Solutions (edu), nic.mil, nic.gov, Allwhois
  - Query listed registrar for detailed whois entries including contacts, postal address, name servers, emails (and formats of email)
  - Also: Use ARIN to find IP blocks for organizations!
  - Whois tool under UNIX
- Whois info is necessary but should be limited to required minimum

Recon & Info Gathering

- DNS Interrogation
  - Tools: nslookup, dig, host, axfr
  - Using the name server, do a zone transfer (type=any) to list all public hosts in a domain and more (ls -d x.com.)
  - Defenses: Don’t leak unnecessary info
    - Don’t use HINFO, TXT records at all, limit host names
    - Restrict zone transfers! Limit to only some local machines and/or secondary DNS servers that need it (allow-transfer directive in BIND)
    - Configure firewall to block TCP 53 except to these hosts (UDP used for lookups, TCP for zone transfers)
    - Transaction Signatures (TSIG security) for trusted hosts
    - Split DNS to discriminate between internal and external hosts

Intrusive Scans and Probes

- Insecure Modems
  - War Dialers (ToneLoc, THC-Scan), Demon Dialers
  - Rogue Remote Access Applications
  - Defenses: Conduct periodic sweeps/checks, create policies explicitly prohibiting behavior
- Determine if a Networked Host is Alive
  - ICMP (Ping, Echo Request/Reply) Sweeps
  - TCP/UDP Packet Sweeps (“TCP Ping”)
  - Defenses: Configure firewalls, border routers to limit ICMP, UDP traffic to specific systems. Monitor with IDS
  - Problems with these proposed defenses?

Intrusive Scans & Probes

- Rudimentary Network Mapping
  - Use traceroute to determine an access path diagram
    - Different packets may take different routes through different interfaces with different ACLs
    - UDP (UNIX) vs. ICMP Time Exceeded (Windows)
  - Cheops, VisualRoute, NeoTrace provide neat graphic representations for mapping
  - Defenses: (see previous) filter ICMP time exceeded, etc.
- Other Recon Tools
  - Sam Spade-ish recon suites
    - Assemble many of these tools in one place
  - Research/Attack Websites

Intrusive Scans & Probes

- Port Scanning using Nmap
  - TCP Connect, TCP SYN Scans
  - TCP FIN, Xmas Tree, Null Scans (Protocol Violations)
  - TCP ACK, UDP Scanning
  - Some sneakier than others
    - Ex: TCP SYN doesn’t complete handshake so connect isn’t logged by many apps (if open we get SYN-ACK response, if closed we get a RESET or ICMP unreachable)
    - Ex: ACK scan can trick some packet filters. If we get a RESET, packet got through filtering device == “unfiltered”. If no response or ICMP unreachable, port is “filtered”
    - Set source port so it looks more “normal” e.g. TCP port 20
    - Use decoys to confuse, Timing Options, Basic Fragmentation

Intrusive Scans & Probes

- Nmap (continued)
  - Combinations of these scans allow NMAP to also perform Active OS Fingerprinting/Identification
    - Based on a database of OS characteristics
    - Also measures ISN predictability (IP spoof attacks)
  - Defenses: tweak logging and monitoring
    - Firewalls/routers should log things like this (e.g. SYN scans) and IDS should note patterns of behavior
    - Use of stateful firewalls for packet filtering?
    - Scan your own systems before attackers do
- All-Purpose Vulnerability Scanners
  - Automate the process of connecting and checking for current vulnerabilities. Ex: Nessus (!), SAINT, SATAN

Network Architecture Attacks

- Sniffing
  - Still lots of unencrypted protocols in common use
  - Defenses: Use encrypted protocol replacements
    - E.g. IPSEC, SSH, HTTPS, SFTP, PGP for mail, etc
  - Sniffers like TcpDump, Ethereal
  - More targeted Sniffers like Dsniff understand specific protocols and can pick out certain types of traffic
    - Passwords in FTP, Telnet sessions, etc
- Sniffing on Switched Networks
  - MAC Flooding results in some switches forwarding packets to all links after its memory is exhausted
  - Spoof ARPs from legitimate hosts to receive their packets, construct a Man-In-The-Middle scenario

Network Architecture Attacks

- Sniffing on Switched Networks (cont'd)
  - Defenses: Static ARP tables where necessary (difficult to manage) e.g., DMZs, between routers & firewalls
- DNS Spoofing
  - Multiple purposes: blackholing and set-up for mitm attacks or site redirects to attacker replica
- Do SSH/HTTPS Prevent these attacks?
  - Not necessarily; built on trust relationships
    - Users must be careful to use only HTTPS sites with valid certificates
    - Must watch out for SSH warning messages if keys don't match previously recorded keys
  - These problems allow for man-in-the-middle scenarios

Network Architecture Attacks

- IP Address Spoofing
  - Simple spoofing: just change the packet's IP address
  - More dangerous: undermining UNIX r-commands (rsh, rhosts), exploiting trust relationships
    - Must be able to predict sequence numbers since attacker never sees SYN-ACK (different LANs)
    - DoS the legitimate host so it can't send RESET
  - Defenses: Make sure sequence numbers are not predictable (vendor patches, etc), avoid using r-commands, don't use IP addresses for "authentication"
  - Also: ingress/egress filtering, deny source-routed packets

R2L, U2R Attacks

- Remote Attacks: Mostly Buffer Overflows in OS, applications
  - Processor and OS-specific
  - Overflow stack, inject shell code to do something nefarious (try wininet.dll under Windows)
    - Also heap, array, integer overflows, etc.
  - R2L = remote to local;
    - Exploit flaw on remote listening application to obtain local user privileges
  - U2R = user to root;
    - Exploit flaw on system (ex: setuid) for privilege escalation
  - Often, backdoors created via Netcat, TFTP, Inetd
- In-depth discussion out of scope for this presentation, unfortunately!

Web-based Attacks

- Web-based flaws important to be wary of too
  - Ex: IIS unicode flaws allow attacker to escape web root directory and run a command as IUSR to upload a copy of netcat and send back a shell… (vendor R2L)
- SQL Injection
  - Inject unexpected mishandled data into web apps, expanded inside the query for surprising results
  - Example: Poorly constructed SQL queries allow attacker to “piggyback” a query modifier in a POST, I.e. listmyinfo.asp?ID=0;delete from users
- Cross-Site Scripting (XSS)
  - Insert scripted data into web apps, which process and return content containing the scripting (send cookies to a malicious third party, etc.)

R2L/U2R and Web App Vulnerabilties

- Defenses: Be aware of standard solutions to these problems, rely on “what has come before”
- Defenses: Patch, patch, patch, patch, and detect too
  - Practice responsible coding for security awareness
    - Beware strcpy!
- Defenses: Practice responsible (“safe”) coding for security awareness
  - Buffer Overflows: (Example) Beware strcpy, etc.
  - Web Applications: (Example) Don’t rely on hidden fields for data security, construct queries carefully escaping quotes, etc
- Where do attackers go from here?
  - Use this information to get to “the next step”
  - Once rooted, installation of root kits, log cleaners, etc.

Password Cracking

- Guessing Passwords via Login Scripting
- Better: Obtain Windows SAM or UNIX /etc/password (/etc/shadow, /etc/secure)
  - Crackers: L0phtCrack (Win), John the Ripper (UNIX)
- Dictionary vs Brute-Force vs Hybrid methods
- Defenses:
  - Strong password policy, password-filtering sw
  - Conduct your own audits
  - Use authentication tools instead if possible
  - Protect encrypted files (shadowing, get rid of MS LM reps, etc.)

Denial of Service

- Remotely stopping service
  - land (src/dst ip same), jolt2 (ip fragment badly structured offset), teardrop (overlapping fragments), etc.
  - Mostly older exploits, prey on flaws in TCP stack
  - Defenses: patch everything, keep up to date
- Remotely exhausting resources
  - Synflood: send lots of SYNs
  - Smurf: directed broadcast attack
  - Defenses:
    - adequate bandwidth, redundant paths, failover strategies.
    - Increase size of connection queue if necessary.
    - Traffic shaping can help
    - Ingress/Egress filtering at firewall, border routers
    - SYN cookies eliminate connection queue

Denial of Service

- The new(er) threat: DDoS
  - Takes advantage of distributed nature of the ‘Net
  - Zombies live on numerous hosts, remotely controlled
    - Examples: TFN2k, Trin00, Stacheldraht
  - Newer threats feature encrypted client-server communication (sometimes stealthy via ICMP, etc.), decoy capabilities, built-in updaters, and a variety of attack types
    - Harder and harder to trace sources
  - Defenses: Consider all previous advice. Also, do your part to keep zombies off systems
    - Detect and remove
  - Best defense is rapid detection; work with your ISP to help eliminate flood with upstream filters

Denial of Service

- DoS (all forms) sometimes used as diversions to hide “real” attacks
  - Flooding behavior can help to conceal something much more devious
  - Be alert!

All-Purpose Defenses 1

- Stay up to date with OS service patches and security-list mailings [most important!]
- Follow principle of least privilege with user accounts
- Harden your systems
  - Close all unused ports, don't run services you don't need
  - Do you really need a C compiler on your webserver?
- Find your vulnerabilities before attackers do and check regularly
  - Probing Tools, Vulnerability Scanners, etc.
- Centrally log all relevant information and monitor as appropriate
  - Network monitoring packages, Intrusion Detection including file integrity checks for system executables

All-Purpose Defenses 2

- Use of Encryption where possible for communication
  - Non-snakeoil certificates for production systems
- Good Solid Policies, Recovery Plans
  - Scripted post-mortems important so no on-the-spot-decisions
- Of course... Regular Backups of crucial data!
  - Be able to recover critical systems with little notice, think about data mirroring and redundancy

Defenses: Firewalls 1

- Stateful Packet Filters
  - Remember earlier packets
  - Allow new packets originating from outside in only if they are associated with earlier packets
- Proxy-Based Firewalls
  - Operates at the application level, so it “knows when a session is present”
  - “Safer” but operate differently; lower performance and you may need features of packet filter

Defenses: Firewalls 2

- Audit your Firewall with Firewalk
  - Determine which packets are allowed through a firewall or router
  - Utilizes TTL field of IP header, given two IP addresses
  - Response from “one hop beyond” indicates port is open
  - Use this information to harden your firewall, configure it for a minimal set of rules!
  - Is it worth filtering ICMP time exceeded messages? Would cripple attacker’s use of Firewalk but may present administrative problems

Defenses: Intrusion Detection

- Deploy an IDS to “watch” for suspicious traffic on your network
  - Equivalent of a network watchguard, “heads up”
  - Must keep it up to date
  - NIDS vs. HIDS
- Problems: Information Correlation
  - How to correlate to provide “scenario views”?
  - Must carefully tune to find relevant information, limit false positives and wasted time

Defenses: Intrusion Detection 2

- Problems: IDS Evasion
  - Attackers mess with the appearance of traffic so it doesn't match a signature
    - Fragmentation
      - Some can't handle it at all, others can quickly become exhausted with a flood of fragments -- fail open or closed?
      - Tiny Fragment Attack (IDS looks for port number to make filtering decisions, first packet is so small it doesn't have it)
      - Fragment Overlap Attack (second fragment overlaps and writes over "okay" port number with "sneaky" one)
      - FragRouter Tool
    - Minor modifications to popular attacks (ex: overflow strings)
      - Whisker CGI scanner tool provides: URL encoding (unicode), directory insertion, fake parameter, session splicing, many more at application level (ex: HTTP)

Not Covered Here But Should Be!!!

- Session Hijacking Mechanisms
- Netcat usage, other common tools
  - Ngrep, LSOF, Log Analyzers, Monitoring Tools
- Much more in the way of R2L, U2R methods and defenses
  - Buffer Overflows, Privilege Escalation
- Wireless Security
- Backdoors/Rootkits/Trojans
  - Vulnerability Maintenance, log cleaners

Tools

- IPFW, IPTables, IPF, etc.
- Snort, ACID, Tripwire
- Ethereal, Tcpdump, Snoop
- Nmap, Dsniff
- FragRouter
- Nessus, Whisker
- Netcat, Nagios
- Ettercap, Hunt
- John The Ripper, L0phtCrack (LC4/5)
- ArpWatch
- Firewalk

Web Links

- www.securityfocus.com (inc. BugTraq)
- cve.mitre.org
- icat.nist.gov
- www.cert.org
- www.packetstormsecurity.org
- www.packetfactory.net
- www.phrack.org
- www.honeynet.org