Introduction to Network Penetration Testing

James Shewmaker

jims@bluenotch.com

Outline

- Definitions and Concepts
- Key Tools
- Targets and Scenarios

What is a Penetration Test?

- A penetration test (pentest) is a systematic probing of a system
  - A system could be any combination of applications, hosts, or networks
  - Emphasis on how deep one can get into the system
  - Sometimes confused with Audits or Assessments

Testing Areas

- What areas can we test?
  - Response / Work flow / Policy
  - Physical
  - Logical
    - Network
    - Host
    - Application

Why are We Testing Anyway?

- How do you **KNOW** your network and systems are secure?
  - Your knowledge is only as good as your last test
  - Your last test is only as good as your weakest link
    - Tools
    - Experience
    - Execution

Authoritative Permission

- Permission must be
  - From the proper authority
  - Very specific in detail
- Unfortunate example
  - Oregon vs. Randal Schwartz, 1995
    - Mr. Schwartz used a back-door and cracked passwords in the course of his work without explicit permission

Pentest Targets / Scope

- Scope is WHAT to test
- Highlight points of Interest or Value
  - Hosts
    - Service
    - Application
  - Network
    - Internal devices
    - Perimeter devices
- Where is the low hanging fruit?
- How deep can we go?

Pentest Goals

- Determine target discoverability
- Assess state of Incident Response
  - Technical skill assessment
  - Policy and procedure practice
- Document unknown or orphan resources

Rules of Engagement

- Establish the HOW of the pentest
- Decide when to begin and end
  - Set a Date and Time window
  - Don't start at the first opportunity
- Build the testing team
- Establish rules for emergency 24x7

Key Definitions in Pentesting

- Attack Vector – A path to deliver a payload
- Leverage – using a component to better position or exploit another component
- Privilege Escalation – leveraging a low privilege account to a higher privilege account
- Remote Vulnerability – exploiting from an outside source
- Local Vulnerability – exploiting on the system itself
- Red Team / Tiger Team – offensive team
- Blue Team – defensive team

Leverage In-Depth

- Penetration testing is mostly about discovery using leverage
- We tend to see avenues of attack that can be represented in the OSI network model
- We will skew the OSI model slightly to fit into our Penetration Methodology

7) Application (public or private availability, input)
6) Presentation (encryption, checksum/checkpoint)
5) Session (an exchange, a specific instance of TCP traffic)
4) Transport (packaging)
3) Network (switching/routing- How to get from one host to another)
2) Data Link (local - How to get from one host to another)
1) Physical (the CAT5 cable, the USB thumb drive, etc.)

Example Avenue of Attack

Layer	Attack Type/Example
User	Social Engineering
Application	Input Fuzzing
Presentation	Abstraction Assumptions
Session	Cookie Stealing
Transport	Transport
Network	Man in the Middle
Data Link	ARP spoofing
Physical	Console Access

Leverage In-Depth (continued)

- Things that may be leveraged
  - public access
  - private/authorized access
  - resources
  - environment assumptions
  - trust relationships
  - standards and assumptions

Techniques and Tricks

- Social Engineering
  - Manipulating the "user space"
- Physical attacks
  - Stop short of beating the system administrator
  - If you have physical access and can force a reboot, you usually have complete access to the system, for example:
    - Boot backtrack to clear the Windows Administrator password
    - Boot Linux into single-user mode and change the root password
- Combination attacks
  - Tap the keyboard with a key logging device, tell the administrator "I broke in, go check!" and when he does, you have his password . . .

Techniques and Tricks (continued)

- Literally any bit of information may be valuable as Recon
  - You may find systems that are "expendable" to the owner but may contain something you can use
  - You may find read-only access to logs that you can watch (valuable when you try to create errors on purpose)
- Look for trust relationships
  - Get access to one, and you have some sort of access to the other
- Look for indirect paths
  - Perhaps the SQL server is not public, but is used by the web server
  - Maybe the SQL server is also used for internal database
- Constantly adding to your collection of intelligence

Techniques Summary

- Explore everything possible, from the largest hosts or applications down to every avenue of input.
  - Look for more holes
  - Look for more applications
  - Look for more clues
- Penetration is focused on "deep," but do not forget "wide"
- Use Leverage wherever you can find it
- Take careful notes
- Think about assumptions and context, you may be able to manipulate the environment to open up new doors
- If you run into a dead end, start from what you do know and explore
- Lower hanging fruit is usually picked first, so start there
- **DO NOT EXCEED THE TEST BOUNDARIES**

Process of Discoverability

- Each test should be repeatable
- Different kinds of progression
  - Going from what you know directly to what you do not know (serial)
  - Educated guessing using similar examples seen so far (parallel)
  - Guessing based on gut feeling or even random idea (brute force and usually pointless)

Typical Penetration Phases

- Perform Reconnaissance
  - Initial mapping and information gathering, focus on observation
- Port / Vulnerability Scan
  - Probe applications for potential leverage
- Manipulate and Exploit
  - Manipulate vulnerabilities and flaws for benefit
- Bootstrap the Penetration
  - Start the process over again from this new vantage point: Recon, Probe, and Exploit new avenues or objects

Recon Tools

- Off site information gathering
  - Google / whois / Maltego / DNS
- Network mapping
  - nmap / nessus
- Host fingerprinting
  - queso / p0f
- Service probing
  - netcat / webscarab
- Custom scripts to harvest info

Network Monitoring Tools

- Discovery new of assets to leverage
- Identification of testing breakage
  - wireshark
  - dsniff
  - snort

Perimeter Tools

- Any packet tool
  - hping (packet crafting)
  - firewalk (firewall testing Mapping out a firewall)
- Check other perimeter devices
  - Perhaps an IDS that make assumptions about fragmentation?
- Don't forget to think outside the building
  - Modem pool
  - Wireless

Application Tools

- HTTP proxies
  - Use webscarab to modify HTTP traffic
- Fuzzers
  - Recon but with consequences
    - Try to break the application to learn more about it
    - All attempts are likely logged
  - Designed to test input parameters
    - Type of input (alphanumeric, numbers, or object)
    - Size of input (underflow or overflow)
- netcat
  - Manually probe responses from application
- Custom programs or scripts
  - Context and Environment leverage
    - Possible race condition
    - Weak path or file system permissions

Exploit Tools

- Test for vulnerability
- Leverage vulnerability
- Favorites
  - Hydra (password guessing)
  - Metasploit (modular exploits)
  - Custom (ie: www.milw0rm.com)

Potential Targets

- Network
  - Perimeter devices
  - Internal nodes
- Hosts
  - Public facing
  - Private leveraging
- Applications
  - Escalated Access
  - Valuable Data

Penetration Testing Styles

- Styles
  - Black Box (Scenario A)
    - Begin with a clean slate and no insider knowledge
    - Simulates random target approach
  - Crystal Box (Scenario B)
    - Some previous knowledge
    - Specific targets
- Approaches
  - Internal (usually not Black Box)
  - External (completely outside the firewall)

Black Box Testing

- Starting with nothing
  - Reconnaissance
    - What shows up with a network sweep?
    - Anything interesting?
      - Hosts
      - Applications
    - Stop and think about how one might find this target in the first place (Google maybe?)
  - Going deeper and see what else you can find

Crystal box Testing

- Tends to involve specific targets
- Easier to define scope than Black Box
- Must be even more careful to NOT make bad assumptions

Aftermath - Now What?

- Sometimes you have to fix something testing may have broke
- The test is pointless without careful and precise documentation
- Documentation is pointless if it is not available
- Use the test results to plan corrective action
  - Specific patching and configuration
  - Plan for future patches and tests

Scenario A - Introduction

- You are a consultant
- Given a company name, find out everything you can
- Highlight points of interest along the way
  - Potential risks
  - Working controls
- No network sniffing
- No breaking of applications
- Avoid disturbing production data

Scenario A – Stage 0

GET FINAL,
AUTHORITIVE,
WRITTEN, AND
SIGNED PERMISSION!

Scenario A – Stage 1

- Exhaust off site recon opportunities
  - Anyone for dumpster diving?
  - Start with free utility services
    - Google
      - using the company name to find indexed sites
      - using the "site:mydomain.com" feature to narrow down the results
      - using the "filetype:xls" to search for data
        - » Could be private
        - » Could be public data but valuable as recon

Scenario A – Stage 1 Continued

- uptime.netcraft.com
  - Netcraft may have the host type, web server version, and uptime
- samspade.org
  - Can check for various things
    - » DNS records
- isc.sans.org
  - Can check an IP for reported info
  - Sometimes every little bit of info helps

– Note and evaluate the information so far

- Where can go from what we know to the next level?

Scenario A – Stage 2

- DNS will be valuable
  - Does whois tell us that the DNS is hosted elsewhere?
  - Can we do a name transfer or figure out how to enumerate records?
- Email addresses of administrators in whois
- Maybe their website or email is on a shared hosting box, how can we tell? Is there any info we can leverage?
- Remember to keep recon as limited as possible to postpone detection

Scenario A – Stage 3

- Skip Network Monitoring, outside the boundaries for this test
- Check out the perimeter
  - Start a traceroute to the public IP addresses you have so far
  - Note any host or service that shows up, ones that definitely do not exist, and ones that are unknown
  - If you find new services or hosts, conduct initial recon on these new items before proceeding to Stage 4

Scenario A – Stage 4

- Check the reachable hosts and devices with specific tools
  - Craft packets with hping to elicit responses from known and unknown devices
  - Manipulate reachable applications, try to break them manually or with a fuzzer; try to generate error messages
  - Look for new clues that may reveal another private host or other resource you cannot see directly

Scenario A – Overall Results

- Documentation
  - Process of discovery
  - Tree diagram representing where and how deep the discovery went
  - List of publicly reachable devices and applications
  - List of test conditions and generated errors
  - List of known exploitable conditions

Pentesting Hands-on Results

- Initial reconnaissance
  - Service Map of host using nmap
- Network monitoring
  - Switched network, no benefit
- Perimeter tests
  - Host is target, perimeter testing is no different than host testing
- Application tests
  - Testing reveals vulnerabilities
- Exploit
  - Leverage vulnerability with recon
- Bootstrap the test to the next level

Scenario B – Introduction

- Defcon 2005 Capture the Flag prequalification
- Emphasis on penetration and vulnerability discovery
  - Kickoff: an email with an http link to begin the contest, and a username and password for later use
  - There are eight (8) flags representing valuable data
  - That is all you have to go on!

Scenario B – Stage 0

- HTTP link reveals several things
  - Hostname dujour.kenshoto.com
  - Running webserver on port 80
  - Some content
- Where does recon stop and preliminary testing start?
  - Sometimes you don't know what you're testing until you break it!

Scenario B – Stage 0 (continued)

- Recon
  - Host info
    - nmap/netcraft/whois/arin
  - Host services
    - Map them with nmap/nessus/amap
    - We see that there is an HTTP service, we can try exploring
      - Default directories/files
      - View source for clues in comments or other links

Scenario B – Stage 0 Results

- Nmap says

  22/tcp open ssh OpenSSH 3.8.1p1 (protocol 2.0)

  80/tcp open http Apache httpd 2.0.53 ((FreeBSD))

  6969/tcp open acmsoda?

  19150/tcp open unknown

Scenario B – Stage 1

- http://dujour.kenshoto.com/cgi-bin/stage1
- Learn more about the application by viewing the source, exploring, and breaking the web application
- HTML form has a hidden form field named "message"
- http://dujour.kenshoto.com/cgi-bin/stage1?message=aa
  - File Not Found error!
  - We also learn that the application doesn't discern between POST and GET variables

Scenario B – Stage 1 Results

- Manipulate the CGI link from Stage 0
  - http://dujour.kenshoto.com/cgi-bin/stage1?message=/etc/passwd
  - Download /etc/passwd and /etc/passwd.shadow
  - Combine and crack with john and a decent dictionary
- Meanwhile more recon by leveraging Stage 1
  - Other key system files (/etc/hosts.allow)
  - Other application files (/usr/local/etc/apache/httpd.conf)
  - Log files (/var/log/messages)
  - User files (/root/.history)
  - Temporary files (/tmp/*)
- Discover username/password
  - breakme/apple1 and root/fred

Scenario B – Stage 2

- ssh breakme@dujour.kenshoto.com
  - Returns binary data and closes connection
  - Appears to be base64 encoded
  - No other new leads
- Can continue this route or look for lower hanging fruit
  - Let's map the services on the host
  - Don't forget to watch log files with the CGI form
    - Note typical behaviour
    - Note error messages due to your probes

Scenario B – Stage 2 Results

- Reverse Engineering the Binary reveals clues
  - Useful tools include gdb, strings, and metasploit
- Binary appears to communicate on port 6969
- Vulnerable to overflow

Scenario B – Stage 3

- Play with the binary with netcat (using username/password from Stage 0)

  $ nc -vv dujour.kenshoto.com 6969 Warning:
  inverse host lookup failed for
  206.131.226.59: Unknown host
  dujour.kenshoto.com [206.131.226.59] 6969
  open AUTH:team13:tUqXasJuxM

  OK
- Back to trying to break it
  - Try bad username/passwords
  - Try invalid input after authentication
  - Step through the binary with a debugger
    - Helpful tools are gdm and ktrace
  - Find an overflow to leverage

Scenario B – Stage 3 Results

- Eventually overflow the binary to open up a remote shell
- Attempt to leverage this new shell
  - Create another account
  - Enable ssh access
  - Hide evidence of penetration?
  - Create misleading evidence?

Scenario B – Stage 4

- We have access to the machine, so now what?
  - Privilege Escalation
  - Data discovery
- Possible avenues
  - Perform forensics on the box for clues to valuable data access
  - Perform monitoring on the box to reveal admin or other user access
  - Brute force additional access

Scenario B – Overall Results

- Repeated the process, trying all existing accounts and system binaries
- This scenario had several types of overflows, including heap, stack, and format string overflows in local applications
- Report includes the extent of how deep we reached into the system

Conclusions

- Where do we go from here?
  - Increase penetration testing depth
  - Increase penetration testing scope
- How can I reliably test production networks and systems?
  - <sarcasm>Be sure to let the rest of us know when you find the perfect way </sarcasm>

Summary

- Methodology Concepts
  - Precise and Systematic Testing
  - Leverage anything an attacker might have or obtain
- Tools
  - Like anything else, use what your comfortable with and what would be appropriate for the environment
- Targets and Scenarios
  - Get Permission and Prepare
- Documentation
  - Precisely document what you find: good, bad, or seemingly insignificant
- Aftermath
  - Put the knowledge gained from testing to good use

For More Information

- Oregon vs. Randal Schwartz http://www.lightlink.com/spacenka/fors/
- The Hacker's Choice http://www.thc.org
- Metasploit http://www.metasploit.org
- Nmap http://www.insecure.org
- IP Address Allocation
  - http://www.ietf.org/rfc/rfc1918.txt
  - http://www.iana.org/assignments/ipv4-address-space
- Snort http://www.snort.org
- Ftester http://ftester.sourceforge.net
- WHAX http://www.iwhax.net
- Backtrack http://www.remoteexploit.org