Pentest+

Chapter 1 – Pre-Engagement Activities

* Planning and preparation for an engagement, it is critical to have a clear understanding of customer adjectives
  + scope, schedule, expectations
* Target Audience
  + understand the businesses purpose and function to ID reporting levels and responsibility
    - define stakeholders (management, legal, security, IT, pentesters)
      * Executive management participation is required
      * Legal/Contracting – necessary
      * Security/IT – essential
      * Pentester – vital
* Impact Analysis
  + key part of requirements management, formal pro/con analysis
    - reason and understanding
    - captured and organized in documents or project management tools
  + Prepare 🡪 Brainstorm 🡪 Take Action
    - Prepare – organizational budget, technical constraints
    - Brainstorm – what ifs, methodologies
    - Take Action – ID risk factors during IA influence scope
* Scope and Methodology
  + testing requirements, target selection, scheduling and timelines, testing strategy
    - Requirements – kinds of tests (net, web, mobile, wireless, social engineering, physical)
      * use scoping documents or pre-engagement survey – informal process
      * General Questions of Pre-Engagement Interactions
        + also consider company pre-mergers and supply chains

know all the players and define authorized boundaries

* + - Selection – discovery/validation of targets
    - Timelines – determine when an engagement takes place – ensure sufficient time
    - Strategy – depends on organizational rationale
      * black, white, grey box
* Type of Assessment
  + Three Distinct Types – perspective distinction
    - Goals-based/Objectives-based
    - Compliance-based
    - Red Teaming
  + Goals/Objective-based – evaluate security within an organization through cyber attack simulation
    - each goal/objective provides general instruction for a given test scenario
      * requires a lot of forethought
      * helps assess people, process, and technology
  + Compliance-based – audit ability to follow and implement a given set of security standards within an environment
    - * PCI, HIPAA, FISMA
    - more of a vulnerability assessment than a pentest, unless explicity required by policy/compliance
      * GRC Model – Governance, Risk, and Compliance
    - ensure entry points into networks are fully covered and documented
      * PCI DSS requires int/ext testing
    - Different testing requirements based on data type and data handling
      * PCI audit vs Pentest in support of PCI compliance
        + need Qualified Security Assessor to determine if parameters fulfill compliance
  + Red Team-based – tests organization with a real world attacks
    - time to detection, time to response, and resilience against specifically modeled threats using varied tactics
      * mimic APTs, Lockheed Martin Cyber Kill Chain
        + PrivEsc, Lateral Esc, Persistence
    - Lengthy, risky, costly
    - lots of planning and consideration, RoE, escalation path 🡪 chain of command
* Threat Modeling
  + - Id, quantify and address the risks associated with an organizations info system
    - PenTesting Execution Standard
      * ensure mutual understanding of what is important (risk appetite, threat emulation, and goal setting)
    - ID assets, define security profiles, ID and prioritize threats, determine countermeasures
  + ID Assets – person, place, thing
  + Architecture Overview – documents application or system, physical/logical implementation
  + Decompose the Application – breaks down tech for entry points, trust boundaries to develop security profile for vulnerability ID
    - validation, authorization, auditing, logging, cryptography
  + ID the Threats – categorize interior/exterior threats, who/what/how, determine threat actors (kiddies, APTs, hacktivist, insiders)
    - * Network Threats
      * Host Threats
      * Application Threats
    - Document threats and rate them
      * risk = probability (1-10) \* damage potential (1-100)
      * develop and justify test cases to demonstrate risk
      * creates test cases to evaluate countermeasures which can be linked together to formulate attack paths
* Target Selection
  + - done during scope of an engagement
  + Internal and External
    - * all first party equipment or third party services being used
      * onsite vs remote
    - Internal – behind external facing firewall using three temporary methods of access
      * limited access – initial exploitation
      * user-level access – basic permissions
      * privileged-level access – admin, domain controller
* Contractual Agreements
  + - enforced by law, required parties with contract signing authority
      * NDAs, MSAs, SOWs
  + NDAs
    - confidentiality agreement, protects proprietary information and intellectual property
    - important for pentests when outsourced
  + MSAs
    - party agreements that govern all other future transactions and agreements
      * payment terms, product warranties, intellectual property ownership, dispute resolution, allocation of risk, indemnification
    - social responsibility, ethics, access, etc
    - open ended
  + SOWs
    - formal, routine in project management
    - project-specific work to be executed by vendor
    - can be found in MSA
      * purpose, scope or location of work, period of performance, deliverables schedule, applicable industry standards, acceptance criteria, special requirements, payment schedule
    - RoE
      * guidelines and constraints regarding execution of a pentest
      * can be part of SOW or separate deliverable
        + needs management, vendor, third parties
        + scope, location, standards, timelines, comms and escalation paths
      * NIST SP 800-115 App. B
    - Support Resources
      * WSDL, WADL, SOAP Project File, SDK Documentation, SWAGGER Documentation, XSD, Sample Application Requests, Architecture Diagrams

Chapter 2 – Getting to Know Your Targets

* Footprinting and Reconnaissance
* Information Gathering
  + passive information gathering – least resistance and detection
  + active information gathering
* Tools, Methods, and Frameworks
  + - social media, websites, email, public forums, etc. 🡪 digital footprint
  + Data Mining
    - OSINT Framework – provides nodes for paths to collecting info on a specific subject
    - Maltego (Paterva)
      * Maltego CE in Kali Linux
        + displays data to plot node entities discovered through the use of a transform (data querying code)

can be chained together by machines

* + - * + demonstrates entity relationships to draw conlcusions about network configuration
        + can integrate info from Shodan and Censys
    - Specialized Search Engines
      * + Shodan
        + Censys
      * Shodan – parses banners for services and category data by device
        + can search by top countries, services, organizations, Oss, Products
      * Censys – similar to shodan
        + IPv4 hosts, Top Million Websites, Certificates
        + By AS, protocol, tags
  + DNS, Website, and Email Footprinting
    - WHOIS directory service,
      * help ID info about domains
    - DNS forward/reverse lookups are common during info gathering
      * dig tool in cmd line
    - theHarvester
      * passive or active queries to gather target info
      * analyze and vet info
    - Recon-ng
      * similar to theHarvester
      * recon-ng in kali
        + uses Recon, Reporting, Import, Exploitation, and Discovery modules
  + MetaData Analysis
    - fingerprinting organizations with collected archives (FOCA) tool
      * analyze common document types

Chapter 3 – Network Scanning and Enumeration

* IEEE 802.3 (Eth), 802.15 (Bluetooth,Zigbee), and 802.11 (wireless)
* 802.11 Wireless Standards
  + 802.11, 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac
* Wireless Spectrum Bands
  + 2.4GHz – 14 channels – 20 + 22 MHz separation – only 1-11 used in NA
  + 5.0 GHz – 20 channels – less coverage – faster throughput
    - center frequency – 5 MHz b/w lower, center, upper (2.4) and 10 MHz b/w lower, center, upper (5)
* Wireless Modes and Terminology
  + important to configure wireless in a manner that introduces least amount of risk
  + first determine mode
    - ad-hoc – IBSS
    - infrastructure – use APs to connect as a BSS or ESS
      * BSS emits SSID
      * ESS emits ESSID
        + both connected by Distribution System (DS)
* Wireless Testing Equipment
  + - adapters that support single or dual band, same support external antenna and OS’s
      * chipset and driver considerations
    - kali wireless adapter property commands
      * id, idlist, iwconfig
        + compatible channels, frequencies, encryption, Aps
* Popular Antennas
  + pentesters should be aware of antenna gain (dBi) values
  + omni, patch, yagi
* 802.11 Network Discovery
  + Stumbling 🡪 discover SSIDs, router vendor info, signal strength, MACs, channels, encryption, etc
  + wardriving
* 802.11 Frames
  + Management Frame
  + Control Frame
  + Data Frame
* Wireless Scanning
  + aircrack-ng – RF monitoring and security testing
  + airodump-ng – sniffing to discover/validate targets
    - ID ESSID and BSSID
  + need to first enter airmon-ng cmd
    - adapter may not support monitor mode
    - uses channel hopping or manually camp on a channel
      * pay attention to PWR and BSSID
* Host Discovery
  + nmap + NSE
* Ping Scan
  + nmap ping scan (-sN or -sP flag)
  + greppable, awk
  + list scan, disable certain scans to prevent firewall blocks
* Port Scanning
  + Port Scanning Methods
    - use nmap -sV to determine different between open/closed UDP ports
    - T 0-5 for timing/aggression (default 3)
  + Common Ports and Protocols – 0-65535
    - common system, registered, and dynamic ports
      * system – 0-1023 – root-level privs
      * registered – 1024-49151 – user-level privs
      * dynamic – 49151 + - private
    - disable unused ports
  + TCP Scan
    - default in nmap, reliable
    - more time required, inefficient
      * use -vv
    - use ports as clues to OS detection
  + Half-Open Scan
    - TCP Syn Scan – most popular method
      * fast, effective
      * allows control over timeout response times (need root privs)
      * service detection flag is helpful with half-open
  + UDP Scan
    - common services such as DNS, DHCP, NetBIOS
    - test ports for these services
* Enumeration
  + hosts, machine names, networks, domains, users, groups, network shares, web pages, applications, OS’s, services, tokens, social networking sites
  + use NSE
    - /usr/share/nmap/scripts
      * head -metadata
      * rule – Lua to determine action execution
      * action – script functionality
    - -sC – safe, typical scripts

Chapter 4 – Vulnerability Scanning and Analysis

* Researching Vulnerabilities
  + - MITRE, ATT&CK, CVE, CWE, CAPEC, CERT, DHS
  + CVE
    - CVE Numbering Authority
      * CVE ID, description, references
    - NIST NVD – CVE and analysis scoring
      * Common Vulnerability Scoring System (CVSS)
    - determine vulnerability impact to an environment
  + CWE
    - common software security weaknesses and mitigations for implementing good secure coding practice and software design
    - Three Categories
      * Research Concepts – academic
      * Development Concepts – during development
      * Architectural Concepts – software engineering
    - Basic Identifiers
      * ID, Description, Relationships, Modes, Applicable Platforms, consequences, likelihood of impact/exploit, mitigations, memberships
    - Full Disclosure
  + CAPEC
    - dictionary, attack patterns and methodologies that are in two categories
      * domains of attack
      * mechanisms of attack
    - show relationships between various attack patterns
      * six unique domains
        + social engineering
        + supply chain
        + communications
        + software
        + physical security
        + hardware
      * leverage and emulate known attack patterns
  + ATT&CK
    - attack emulation, modeling TTPs
  + CAPEC & ATT&CK – Common Attacks
* Remote Security Scanning
  + - consider time to scan, fragile systems protocols being used, API quotas, network topology, bandwidth limits
    - host discovery
    - APIs
      * local
      * web
      * program
    - time needed to complete vs time available
    - all critical to understand to prevent DoS
  + Nessus enables authenticated scanning and is best for remote vulnerability scans that require authentication
    - can also deliver exploits
* Credentialed vs Non-credentialed Scanning
  + uncredentialed users get a view of available attack surface when using Non-cred scanning
  + credentialed reduces false positives and determines impact of stolen credentials
* Vulnerability Scan Analysis
  + vuln scanners do not consider environmental factors, mitigating controls, and data protection measures
  + Nessus uses CVSS to prioritize
  + Alternates – OpenVAS, Nexpose, QualyGuard
* Compliance and Configuration Auditing
  + Nessus
  + CIS best practices
  + SCAP-aware tools 🡪 NIST 🡪 OpenSCAP
    - baselines, custom compliance scans
* Non-Traditional Assets
  + SCADA, ICS 🡪 fragile, not security oriented
    - workstations, RTUs, PLCs, Comms Infrastructure, HMI
    - nmap NSE – modbus-discover
* Embedded Systems
  + microcontrollers, microprocessors, IoT
  + RTOS – adheres to task deadlines
    - hard – strict
    - firm and soft – flexibility in timing
  + Often built from recycled code with same vulns
    - DoS, RCE, Overflow, Corruption, InfoDisclosure, PrivEsc
    - default creds, lack of patches
      * IoT Security Companion to CIS CSC
* Web and Database Scanning
  + OWASP
    - Top Ten, Testing Guide, ZAP, Dirbuster, Webgoat
  + Fingerprinting Web and Database Servers
    - determine product/version number
      * TCP/IP Stack Fingerprinting for signature detection
      * nmap/netcat
        + nc.exe in windows
  + Enumerating Information
    - NSE
  + Map the Application Framework
    - robots.txt 🡪 URIs
    - sitemap + crossdomain files
* Brute-Force Directories and Filenames
  + Dirbuster
    - known, common directories/files
* Authentication and Authorization Testing
  + Password Security Testing
    - quality rules, remember functionality, reset/recovery, change process
    - CAPTCHA, MFA, logouts, default logins, lockout/password change notifications, consistent shared-authentication schemes
    - weak security question/answers
  + Default Account Settings
    - GPO settings, UAC
    - Guest Accts
    - use own accounts to authenticate to a host and then escalate privileges as necessary
  + Recovery Passwords
    - * hashing, password storage 🡪 MD5, SHA1,2
        + pay attention to digest size (bits) and length (characters)
      * encryption – symmetric – AES, 3DES
        + asymmetric – DSA, RSA
      * Digital Certificates – x.500
      * PKI
    - hash id script in kali
    - password cracking
      * dictionary, brute force, rainbow
* Session/Management Testing
  + session ID, token
    - should use RNGs
  + OWASP Session Management Cheat Sheet
  + BurpSuite
* Single Sign-On Architectures
  + OAuth, OpenID, SAML, ADFS
    - implementation weaknesses in end user devices
  + OWASP Authentication Cheat Sheet
* Data Validation Testing
  + HTTP Parameter Pollution
    - enter arbitrary values into web parameters in an effort to cause unexpected behavior that could lead to client/server side weaknesses
      * HTML or CMD injection
        + /index.php?id=1234

increment by one and monitor error statuses

* + Analyzing Error Codes and Stack Traces
    - 1xx – info responses
    - 2xx – success
    - 3xx – redirection
    - 4xx – client errors
    - 5xx – server errors
  + Execution and Injection Flaws
    - lack of proper input validation
    - injection flaws for Oss, LDAP, XML, NoSQL, SQL
      * use Sqlmap
* Vulnerability Mapping
  + map vulnerabilities to exploits
    - prioritize testing activities 🡪 vulnerability mapping 🡪 usefulness metrics

Chapter 5 – Mobile Device and Application Testing

* Mobile Device Architecture
  + system on a chip
    - CPU, GPU, RAM, ROM, Modem (SIM)
  + iOS and Android
* iPhone Operating System
  + Cocoa Touch – GUI
  + Media Services
  + Core Services
  + Core OS
    - (high to low level services)
    - iOS Security Guide, OWASP Mobile Security Testing Guide
  + Hardware Security
  + Secure Boot
  + Code Signatures
  + Sandbox
  + Encryption and Data Protection
  + General Exploit Mitigations
    - uses AES in iDevice for GID and UID
    - sandboxing is used with applications
* Android Operating System
  + applications run their own processes within ART VMs for isolation
  + primary components of Android OS
    - Activities
    - Fragments
    - Intents
    - Broadcast Receivers
    - Content Providers
    - Services
  + hardware Abstraction Layer interfaces with Builtin Components
    - C, C++ for HAL + ART
* Mobile Pentesting Fundamentals
  + - most development occurs at application layer, vulnerabilities stem from poor development practices
    - OWASP Mobile Security Project – Mobile Top 10 and Mobile App Security Checklist
  + Static Analysis
    - security code review, map functionality, ID structure
      * Android Package Kit – JAR format and android manifest XML
      * iOS applications are stored in iOS App Store Package format (IPA) – ZIP compressed
    - Disassembly and Decompiling
      * Info Disclosure Weaknesses 🡪 Custom Encryption Protocols and Configurations 🡪 File and Permissions Analysis
  + Dynamic and Runtime Analysis
    - DAST – blackbox testing
      * Bruteforce PIN/Pattern, Application Functions
      * Binary Attacks, Buffer Caching
      * Client-Side Injection, Sensitive Info Storage
      * Evaulate Shared Application
  + Network Analysis
    - web traffic inspection
      * weak/deprecated encryption
      * app transport security – forces HTTPS
  + Server-Side Testing
    - Nessus, BurpSuite, Nmap
      * default creds
      * session timeouts
      * cmd/SQL injection, input flaws
      * WSDL document exposure
* iOS Application Security Testing
  + Setting up testing environment
    - laptop, USB cable, phone, login info
  + Jailbreaking an iOS Device
    - Untethered – no computer needed
    - Tethered – computer needed to boot
    - Semi-Tethered – jailbreak again upon boot
    - Semi-Untethered – same as above but use jailbreak app on iDevice
  + Cydia Impactor
    - GUI tool to install IPA on device to transfer jailbreak over to device
  + Cydia Package Manager
    - app store for jailbroken devices
    - recommended
      * Class Dump
      * Wget
      * IPA Installer Console
      * OpenSSH
      * Filza
  + Connecting to iOS Device
    - ssh, ssh over USB (iProxy)
  + iOS Functional Testing and Application Mapping
    - OWASP iGoat and DVIA
  + Static Analysis
    - need IPA file
      * Xcode Developer Keys
    - clutch decryption tool
    - Mobile Security Framework
    - MobSF 🡪 GitHub
      * loopback address login
    - Property Lists used by every application 🡪 stores configuration data about applications
      * info disclosure attacks
    - Binary Analysis
      * way to evaluate bugs in compiled software
  + Dynamic Analysis and Reverse Engineering
    - idb program
* Android Application Security Testing
  + Setting Up Android Testing Environment
    - laptop, cable, Android Studio, Rooted Android device
    - DIVA Android App
  + Rooting an Android Device
    - exploit software vulnerability to gain elevated privileges on the device to unlock direct access to Android OS
      * Wondershare
      * Kingo
      * Genymotion
  + Connecting to the Android Device
    - Android Debug Bridge
      * SSH
    - Comes in Android Studio – adb cmd
      * shell 🡪 adb shell
  + Navigating the Android File system
    - internal memory + external storage
      * /
      * /system
      * /data
      * /cache
      * /storage
  + Android Functional Testing and Application Mapping
    - DIVA
      * diva-android 🡪 Github
  + Static Analysis
    - APK Studio 🡪 reverse-engineering framework for disassembly and rebuilding Android Apps
      * signing feature
      * need JDK, Apktool, Uber-apk-signer, adb, zipalign
  + Dynamic Analysis and Reverse Engineering
    - Drozer framework
* Software Assurance Testing
  + - ID vulns in software due to flaws in programming logic
      * Protection and the Control of Information Sharing in Multics
    - Access Control Lists
    - Hierarchical Control of Access Specifications
    - Identification and Authentication of Users
    - Memory Protection
  + Cyber Security Engineering
    - Risk Drives Assurance Decisions
    - Align risk across all interconnected technology and organizational elements
    - software dependencies should be validated and proven trustworthy
    - plan for cyber attacks
    - implement defense in depth security practices
    - assurances should adapt to change
    - organizations should measure assurance effectiveness
  + NIST SARD 🡪 maps to MITRE CWE
  + Understanding Programming Logic
    - Assembly written in ARM, 8086, etc. are divided into three main sections
      * .data 🡪 declares initialization data/constants
      * .bss 🡪 uninitialized variables
      * .text 🡪 keeps code and instructs kernel
    - combined into assembly statements
      * instruction and parameters
  + Common Programming Languages
    - C, C++, Objective-C, Swift
      * need to be compiled into exe for a given processor type
    - Python, Ruby
      * interpreted
    - Bytecode (platform independent) vs. machine-code (processor-specific)
    - uses objects, classes, and inheritance, can be polymorphic and use abstraction and encapsulation
  + Basic Scripting
    - variables, constants, substitutions
    - objects, methods, operators
    - flow control, looping
    - threading, I/O functions

Chapter 6 – Social Engineering

* Motivation Techniques
  + elicitation
  + authority, Likeness, Scarcity, fear, social proof
* Social Engineering Attacks
  + - baiting, shoulder surfing, pretexting
    - waterholing, phishing
  + Phishing
    - email-based
      * spear phishing
      * whaling
        + opportunistic vs targeted
        + Social Engineering Toolkit + BeEF
    - phone-based
      * SMS phishing
      * voice phishing
* Countermeasures
  + - security training
    - fine-tuning tech controls
    - active defense (IDS/IPS), CCTV
    - sensitive asset destruction
    - sensitive info handling SOPs
    - annual testing

Chapter 7 – Network-based Attacks

* Name Resolution Exploits
  + - DNS, LLMNR, NetBIOS
      * DNS – application layer
      * LLMNR + NetBIOS – Microsoft OS
        + local name resolution
        + WINS
    - ADDS – Windows DNS server
    - Linux uses daemons – BIND, named
      * both int/ext
    - Forward Lookup – provide IP for FQDN
    - Reverse Lookup – provide FQDN for IP
    - cached requests can leave DNS clients susceptible to cache poisoning and spoofing attacks
* DNS Spoofing and Cache Poisoning
  + nmap nse 🡪 dns-cache-snoop-nse
    - conducted against internal DNS server
    - see where a target is browsing on the internet to allow
      * waterholing
      * DNS Spoofing
      * DNS Cache Poisoning
    - DNS Spoofing uses MiTM to exploit a target with a malicious redirect or steal sensitive info
      * Ettercap Tool + BeEF
        + steps on pages 165-169
* Attacking LLMNR and NetBIOS
  + NetBIOS – helped facilitate communications over network between Microsoft devices
    - L4+L5 function – TCP/IP (137/138 UDP; 139 TCP)
    - protocol management, messaging and data transfer, host name resolution
    - ports typically found on LANs as NetBIOS heavily relies on trust
  + LLMNR – mimics IPv4 + IPv6 DNS for LANs
    - a malicious user can spoof an authoritative source by responding to LLMNR (5355 UDP) and NetBT (137 UDP)
      * target replies with authentication, exposing username and NTLMv2 hash
    - Responder tool 🡪 LLMNR, NBT-NS, MDNS poisoner
      * usr/share/responder in kali
      * use grep to review actively poisoned ports via PID
    - hosts follow sequence for host name resolution
      * DNS 🡪 LLMNR 🡪 NetBIOS NBNs
        + LLMNR/NBNS query is the focus of this attack
      * check Poisoners-Session-log for poisoned answers and responses
        + compromised info saved in txt, usable by JtR
    - Can also use Windows Proxy Auto-Discovery Protocol (WPAD) attack through Responder to force HTTP whenever a WPAD request is detected
      * MS16-077 to mitigate
        + point to DNS server manually
        + disable PADS in IE

disabled through AD

* + Windows uses NT or NTLM for remote authentication during a Pass the Hash attack
    - Stored in SAM
    - NTLMv2 = NTLM + Challenge/Response Algorithm
      * cannot be replayed
* Stress Testing Applications and Protocols
  + Denial of Service Attacks
    - CAPEC Mapping
      * Flooding – CAPEC-125, sustained client engagement – CAPEC-227
      * Excessive Allocation – CAPEC-130
      * Resource Leak Exposure – CAPEC-131
    - most common method is Flooding
      * TCP Flood – CAPEC-482
      * UDP Flood – CAPEC-486
      * ICMP Flood – CAPEC-487
      * HTTP Flood – CAPEC-488
      * SSL Flood – CAPEC-489
      * XML Flood – CAPEC-528
      * Amplification – CAPEC-490
    - DDoS
  + Executing DDoS Attacks
    - Imperva
      * Volume-Based 🡪 Bits Per Second 🡪 Bandwidth
      * Protocol 🡪 Packets Per Second 🡪 Server Resources
      * Application-Layer 🡪 Requests Per Second 🡪 Service Crash
    - Hping
      * DoS via TCP, UDP, ICMP, Raw-IP
* Network Packet Manipulation
* Analyzing and Inspecting Packets
  + - wireshark, tcpdump
      * run nmap port scan, capture using tcpdump and analyze with wireshark
        + check fourth layer, type and code
        + IANA protocol assignments
  + Forge and Decode Packets
    - may need to spoof IP address of authorized host
    - use Scapy for packet manipulation
      * Scapy 🡪 CAPEC-93 (log injection-tampering-forging)
        + mislead audit or cover tracks
        + expose LFI, XSS, CSRF, RCE opportunities
    - Syslog messages sent via UDP are not validated and canbe easily spoofed over the network
      * use Scapy to send Syslog-like messages to logging server to invalidate logging capability
* Layer-2 Attacks
  + Attacking the Spanning Tree Protocol
    - * STP (802.1D), RSTP (802.1 W+D), MST (802.1S+Q)
      * Wireshark
        + inspect BPDUs – STP ID = 0, RSTP/MST=3
      * biggest threat is DoS as STP does not authenticate root bridge
        + malicious BPDUs to elect fake new root bridge
  + VLAN Hopping (ATT&CK – PRE-T1092)
    - switch-spoofing – emulate trunking switch via 802.1Q
      * default config/improper config 🡪 dynamic desirable
    - double tagging
      * native VLANs allow fake prepending VLAN tag
        + bypass L3 access controls
* Bypassing Network Access Controls
  + 802.1x – NAC
    - hardware or software based
      * Network Security Groups (Cloud)
    - DHCP-based NAC uses proxy to validate DHCP requests
  + Three Methods to Bypass
    - Violate Trust Relationships – IP, MAC spoofing device that is in an NSG that doesn’t use 802.1x
    - Exploiting Implementation Wekanesses – weak authentication, using a static IP to avoid DHCP
    - Taking Advantage of Configuration Weaknesses – NAC on IPv4 but IPv6
  + MITRE ATT&CK – T1200
  + 802.1x and NAC: Best Practices for Effective Network Access Controls
* Attacking Common Protocols
  + Exploiting SNMPv1
    - v1+v2 – cleartext passwords
    - most SNMPs ship with default public or private community strings
      * allow for modification of SNMP variables
    - NSE 🡪 snmp-info.nse, snmp-netstat.nse
      * Metasploit 🡪 auxiliary/scanner/snmp/snmp\_enum
* Poorly Configured File Sharing
  + - Network Share Discovery – T1135 – smb
    - Data from Network Shared Drives – T1039
    - Windows Admin Shares – T1077
    - Remote File Copy – T1105
  + Exploiting SMB Shares
    - GPOs control file/printer sharing in AD
      * disable network discovery
      * publish shares to prevent browsing
      * prevent sharing direct from user profiles
      * restrict sharing only to advanced/privileged
    - nmap NSE 🡪 smb-enum-shares.nse
    - sysinterals and Impacket 🡪 execute cmds against OS
      * psexec
    - enum4linux.pl
  + Attacking FTP service
    - anonymous user functionality
      * ftp-anon.nse
      * ftp-syst.nse
      * exploitDB
    - over 1,000 FTP exploits
  + Exploiting Samba and NFS
    - Samba 🡪 Common Internet File System protocol
    - Cross OS file share
    - samba uses SMB + NetBIOS to let Linux/Unix talk to Windows
      * 50 exploits
      * metaplsoits
    - NSF 🡪 nsf-showmount.nse
      * rpcinfo is important to specify mount port
      * File systems exported with root access
        + mount critical directory to local file system
        + determine if root, if so, can modify files and move tools on remote system as root
    - Recommended BestPractices
      * nosuid, root squash, restricted root access
  + Abusing SMTP
    - servers with open mail relays allow anonymous users to connect over internet to send mail as impersonated email address
      * allow internal spear phishing
      * Metasploit smtp\_relay
        + VRFY/EXPN cmd
      * smtp-enum\_users.nse

Chapter 8 – Wireless and RF Attacks

* Wireless Encryption Standards
  + - need correct antenna, good distance
    - use EMF shielding, SSID hiding, MAC filtering
  + WEP, WPA, WPA2, WPA3
* Cracking WEP
  + 802.11a+b
  + secret key reliance
    - RC4 stream cipher
    - CRC-32 checksum
    - 24bit IV – quickly reused
  + replaying ARP packet to generate new IVs
    - kali
      * airmon-ng
      * airodump-ng
      * aireplay-ng
      * aircrack-ng
  + can also do fragmentation attack
    - aireplay, packetforge-ng
    - exploit PRGA in RCA
* WPA
  + - Enterprise requires RADIUS
    - WPA uses PSK + TKIP – WEP + RC4
    - WPA2 uses PSK + AES
    - PSK 🡪 PMK (Hash of PSK via PBKDF2) 🡪 PTK
    - GTK used to encrypt broadcast traffic on network
    - PMK = PBKDF2 (pswd, ssid, ssid length, 4096, 256)
      * never shared over network
  + Cracking WPA-PSK and WPA2-PSK
    - both susceptible to brute force
    - use info collected from four-way handshake
      * use PBKDF2 algorithm to generate PMK values from wordlist
      * aircrack utilities
        + airodump – collect handshake
        + aircrack + wordlist
      * kali genpmk cmd to generate rainbow table
        + then use cowpatty cmd with four-way and genpmk hash file
      * hashcat
        + cap2hccapx
        + dictionary attack
  + Cracking WPS
    - PIN susceptible to BF attacks
    - Kali “wash” tool
    - ID all WPS-enabled networks
      * reaver cmd to BF WPS PIN
        + use PIN and receiver to retrieve password
      * try and bypass lockouts using reaver time delay and macchanger
* Wireless Attacks and Exploitation
  + - deauthentication to capture handshake for offline password attacks
    - Wireless Jamming
      * WiFi\_Jammer module in Websploit Framework
        + opt-get install websploit
        + use wifi/wifi\_jammer @ wsf>

run

* + Man-in-the-Middle Attacks
    - Evil Twin Access Points
      * Kali 🡪 HostAP (apt-get install hostapd)
        + impersonation against PAP, CHAP, MS-CHAPv2 via fake EAP message
        + hostapd-wipe tool
      * airbase-ng utility
        + airbase-ng -a <bssid> - -essid <wireless name> -c <channel> <interface>
        + issue deauth packets via aireplay -ng

aireplay-ng -deauth0 -a <target AP MAC> <interface> -ignore-negative-one

* + - Session Hijacking
      * wireshark 🡪 session cookie extraction
    - SSL Stripping and Downgrading
      * use proxy server so that client connecting to SSL-enabled website, certificate will be stripped away
        + no SSL cert in browser
        + HTTPS downgraded to HTTP

data will be plaintext

* + - * websites enforce HSTS to mitigate this
* Attacking Bluetooth
  + - * vulnerable if legacy or faulty
      * short PINs, public pairing
  + Device Discovery
    - Blueprinting
      * BlueZ 🡪 basic recon
        + hciconfig cmd
        + hcitool + “bluelog” scanner
  + Data Exfiltration and Compromise
    - Bluesnarfing
      * firmware exploits to steal info from a wireless device
      * bluesnarfer tool
    - Bluebugging
      * attacker remains listed in phone and issue AT cmds to control device
      * mitigated via updates and PINs
        + PIN can be derived from authentication packets
  + Eliciting Unwanted Messages
    - Bluejacking
      * electronic business card sent to victim
        + mitigate by minimizing discovery mode time
    - Bluesmacking
      * DoS via echo requests over L2CAP
        + 12 ping cmd in kali
  + Primary Layers of Bluetooth
    - SDP – service discovery
    - LMP – Link Management Protocol
    - L2CAP – Logical Link Control and Adaptation
    - RFCOMM – Radio Frequency Communication uses L2CAP
    - TCS – Telephony Control Protocol

Chapter 9 – Web and Database Attacks

* Server-Side Attacks
  + - common in DMZ
  + Injection Attacks
    - * SQL Injection
      * Command Injection
      * XXE Injection
    - inspect client-side code (JS) or server-side code (PHP)
    - server side code can provide an additional later of protection by using similar validation rules to ensure data is properly validated
      * improper input validation can lead to XXE, command injection, SQLi
  + SQL Injection
    - common syntax among databases
      * INSERT
      * SELECT
      * UPDATE
      * DELETE
    - mysql -u <user> -p -h <ip addr>
      * + obtain shell privileges through UDFs
        + show privileges cmd
        + use <databases> cmd
      * login first
      * show databases
      * use mysql
    - sqlmap -d “mysql://root:password@ip:port/test” - -os-shell
    - http params are subject to SQL injection attacks with integer and strings
      * in a GET or POST request
      * uses URL encoding, which prevents harmful ASCII input
    - Different SQLi Attacks
      * Blind-Injection – T/F questions evaluated by server response
        + Boolean – linear-based
        + Time – binary
      * Error-based Injection – DB errors are used to derive a valid statement that could be used to extract additional content
      * Union Query Injection – builds on top of SELECT() statement used in query to extend results beyond what was intended
      * Stacked Queries Injection – works by terminating the original query and executing another query
    - To evaluate if a parameter is injectable, try a series of injection criteria to elicit an error from DB
      * error-based
      * develop valid SQL injection syntax
    - Command Injection
      * leverage Metasploit with msfvenom and generate a meterpreter session
      * use “curl” and filter out returned command input
    - XML eXternal Entity Injection
      * entities can be declared internal or external to the application
        + uses Document Type Definition (DTD) to define structure and attributes
      * Injection occurs when DTD pointer is modified or appended during HTTP POST request to point to another location other than the original entity declaration and is processed without validation
        + PoC scripts are available
        + PHP Expect module increases this vulnerability and allow bash shell command exe

cp /bin/ksh /tmp; chmod 4777 /tmp/ksh

korn shell binary

* + Attacking Authentication and Session Management
    - Three-Types – Credential Brute Forcing
      * authentication bypass
      * session predication
  + Brute-Force Login Pages
    - HTTP POST requests – CeWL tool & Hydra
    - BurpSuite + Proxy
  + Authentication Bypass
    - * forced browsing
      * SQLi
      * Parameter Modification
      * Session ID prediction
    - Forced Browsing
      * web application only enforces access control on the login page and nowhere else on the site, authentication can be bypassed by accessing a page on website without authenticating
    - Parameter Modification
      * if authentication is based on a fixed parameter
        + authenticated=no/yes
      * change parameter and attempt restricted page access
      * can also unhide hidden form fields in browser
        + ex. isAdmin=0, change to 1
    - Predictable Session Tokens
      * since session-based authentication is stateful, both client + server record cookies
      * response header with Set-Cookie will have options
        + Set-Cookie options

HTTPOnly 🡪 no IS access

Path 🡪 defines valid URLs

Domain 🡪 define valid domains

Expires 🡪 expiration time

Secure 🡪 prevent unencrypted transmission

* + - * + if lacking sufficient randomness and complexity, can be brute-forced

collect cookie samples

analyze to determine randomness

create valid malicious cookie

all in Burp Sequencer

* + - * + find valid sessions by looking for cookie numbers that are unaccounted for
  + Inclusion Attacks
    - * load arbitrary file content
        + LFI + RFI
    - Remote and Local File Inclusions
      * take advantage of dynamic file inclusion capabilities in web applications
      * LFI 🡪 files outside of web root and renders contents of local OS files on browser
        + leads to RCE
        + Test LFI with PHP wrappers

Expect wrapper

input& stream

* + - * RFI allows files/pages to be displayed inside vulnerable web page
        + allow code to run on server or client
      * Both mitigated with input validation
      * can also perform malicious file uploads
        + web shells and backdoors
  + Exploiting Security Misconfigurations
    - * default passwords, unpatched flaws/bugs, unprotected files/directions, plugins
    - Weak Access Controls
      * ACLs and Policies
      * if lacking, can access restricted folders
        + directory indexing
      * remove indexing option in Apache
    - Exposing Sensitive Data
      * poor encryption, error messages, internal function references
        + Insecure Direct Object Reference (IDOR)
        + allows schema inference of DBs
    - Directory and Path Traversals
      * parameters visible in URL that can be changed, variables in code that ID absolute path
        + ../../../../
      * mitigated by base-naming sensitive variables
* Client-Side Attacks
  + HTML Injection
    - inject arbitrary code without proper sanitization
      * stored – server-side
      * reflected – nonpersistent
  + XSS
    - Reflected – single HTTP Response
    - Stored – injected in log
    - DOM-based – passed down to the browser from application during runtime
  + CSRF
    - causes user to perform an action against a trusted website where user is already authenticated with a valid session
  + Clickjacking
    - client-side security issue
      * load legit context through malicious website

Chapter 10 – Attacking Local Host Vulnerabilities

* OS Vulnerabilities
  + common – DoS, CE, GainInfo, GainPrivs
    - DoS – kernel exploits, appexploits, memory corruption
* Postexploitation
  + - * gain situational awareness
      * collect useful info
      * exfiltrate data
    - Discovery
    - Collection
    - Exfiltration
  + Gain Situational Awareness
    - Account Discovery – T1087
    - File and Directory Discovery – T1083
    - Network Share Discovery – T1135
    - Password Policy Discovery – T1201
    - Permissions Groups Discovery – T1609
    - Process Discovery – T1057
    - Remote System Discovery – T1018
    - System Network Connections Discovery – T1049
  + Collecting Information
    - Clipboard Data – T1115
    - Input Capture – T1056
    - Screen Capture – T1113
    - Data From Information Repositories – T1213
    - Data from Local System – T1005
  + Exfiltration
    - Exfil over C2 Channel – T1041
    - Exfil over Alternative Protocol – T1048
    - Data Encrypted – T1022
  + Privilege Escalation
    - Exploit for Privilege Escalation – T1068
* Linux Privilege Escalation
  + Linux Kernel-level exploits
    - search for exploits with uname -a for sys info
    - Metasploit 🡪 meterpreter 🡪 local\_exploit\_suggester
    - race conditions 🡪 way memory sub systems handle copy-on-wire breakage of private read-only memory mappings
* Finding SUID/GUID executables
  + setuid + setgid bit permissions applied to files on OS via chmod 4777, chmod o+s, chmod g+s
  + T1166 – shell escape 🡪 special bit should not be applied to applications with known vulns
* Locate all setuid exe files
  + find / -perm -u=s -type f -exec ls -al {} \; 2>/dev/null
    - or
  + find / -user root -perm -4000 -orint 2>/dev/null
* Exploiting Sudo Configurations
  + /etc/sudoers file
    - use visudo
  + /etc/groups fiel
  + read and write privileges
    - sudo caching 🡪 T1206
      * echo ‘Defaults !tty\_tickets’ >> /etc/sudoers
        + sudo spans across all TTY sessions
      * Then execute sudo /bin/basj
        + sudo cat /etc/shadow
* Windows Privilege Escalation
  + Windows kernel-level exploits – 20-30% of CVEs
    - use Nessus or lester
    - meterpreter runs entirely within memory and leaves little trace
      * windows/smb/psexec
        + need local admin group perms
      * msfvenom payload, copy over to smbclient and execute
    - can use windows/gather/enum-patches
    - can also use wmic utility
    - accesschk.exe
      * -cv – verbose permissions
      * -uv – command verbosity and error suppression
      * -an – search for rights for assigned user account and looks for objects that have no access
      * -cw – looks for permissions given Windows service and returns objects that have write access
      * -uc – looks for permissions for given windows service and suppression errors
  + Credential Dumping – T1003
    - Group Policy Preferences
      * allowed domain admins to create domain policies, stored in SYSVOL
        + contains logon scripts, GPdata, other domain-wide data

this can be decrypted with leaked cPassword value

* + - * Get-DecryptedCpassword ps function in Powersploit tool kit
      * Metasploit post/windows/gather/credentials/gpp
    - Security Accounts Manager
      * retrieve contents of SAM file using
        + pwdumpx.exe
        + Mimikatz
        + gsecdump
        + secretsdump.py
      * contents available in local registry
        + regedit

HKEY\_LOCAL\_MACHINE(HKLM reg hive)

need SYSTEM-level privileges to extract entries (both SAM + SYSTEM)

* + - * + copy required objects to files on the system

download to local via meterpreter

* + - * impacket\_sceretsdump in kali
      * can also use Mimikatz
        + can be used in 32+64 bit
        + meterpreter 🡪 mimikatz\_command
    - Local Security Authority
      * manages systems security policy
      * stored in HKLM\Security\Policy\Secrets
      * use Mimikatz to extract LSA secrets (host/domain acct login creds, acct info)
      * LSASS credentials stored in NTLM,LM,cleartext
      * SSD is a DLL for security accessibility to apps
        + SSPI acts as an SSP interface to facilitate credential storing
      * following SSPs allow access to subsystem
        + Msv – logons
        + Wdigest – HTTP + SASL use
        + TSPkg – web service security
        + Kerberos
        + CredSSP – SSO for RDSs
      * extract with Mimikatz via sekurlsa::logonPasswords option
    - Service Principal Names
      * Kerberos
        + tickets have ticket lifetime and renewable lifetime
      * requires SPNs be associated with one service account logon acct
        + determines which service account hash to use in order to encrypt service ticket

host-based SPN: randomized by default

arbitrary SPN: sometimes limited to a domain user account

* + - * compromise TGT, can request multiple TGS tickets from DC or arbitrary SPN
        + if arbitrary SPN is tied to domain user, NTLM hash can be compromised
      * Kerberoasting
        + done via Empire framework and Invoke-Kerberoast ps script or Mimikatz with Kerberos :: list /export

convert .kirbi files to JtR format

* + - Unattended Installation
      * answer files on network shares or local admin workstations that could aid in exploitation
        + update answer files to create local account or service on system and repackage installation file for remote compromise
    - DLL Hijacking Attacks – T1038
      * if DLL doesn’t exit, or insecurely implemented, can elevate privileges by forcing applications to load and execute a malicious DLL
      * standard process
        + program installation directory
        + windows system directory (Sys32)
        + windows directory (sys)
        + current working directory
        + directories in sys PATH environment variable
        + directories in user PATH environment variable
      * use ProcMon to search for DLL search order hijacking
        + missing DLL files
        + if confirmed missing from all paths, use meterpreter reverse\_tcp shell payload in highest priority directory you have write permissions in
* Exploitable Services
  + Buffer Overflows
    - variables stored in memory ordered statically (stack) or dynamically (heap) in RAM
      * push and pop via LIFO model
      * to free a block from the stack, adjust pointer to next memory address
    - if data needed is known ahead of time use stacks, otherwise use heap
    - heap is application specific, slower than stack, but holds more data
      * regulated by virtual memory allocated at runtime
      * buffer overflows in heal can cause issues 🡪 not protected by CPU
    - important to know what is put into buffer size and execute bounds checking on all memory access
    - stack based are similar
      * user supplied input not being validated is often a culprit
        + stock canary 🡪 detects overflow before malicious code is executed

segmentation fault is thrown when restricted memory area is almost subject to overwrite

* + - * + Data Execution Prevention controls (NX) controls execution on stack

leaves malicious payloads useless

* + Unquoted Service Paths
    - * when a new service is created on local OS (via sc.exe), unqie key is created in the registry
        + HKLM\SYSTEM\Current Control Set\services
      * users can search with wmic cmd to find unquoted service paths
        + C:\Program.exe
        + C:\Program Files.exe
        + C:\Program Files(x86)\Shared.exe
        + C:\Program Files(x86)\Shared Services\Shared.exe
        + C:\Program Files(x86)\Shared Services\shared commands
        + C:\Program Files(x86)\Shared Services\shared commands\wincerbile.exe
    - Result of CreateProcess function
      * use icacls cmd
        + msfvenom meterpreter\_reverse\_tcp\_payload
        + configure multihandler

migrate to another process after reboot

* + - when admins intermingle processes and applications with lower-privileges user accounts, could cause a situation as above
      * mitigate with surrounding PATH variable in quotes when possible
      * be aware of search order, cleanup registry keys
      * T1058
* Lateral Movement
  + pivoting via OpenSSH, RDP, FTP, VNC
    - SSH Hijacking
    - third Party Software
    - Remote Services
    - Remote File Copy
    - RDP
    - Windows Remote Management
  + Lateral Movement in Linux
    - Attacking VNC Servers
      * sessions lacking required authentication
        + VNC password also stored in Home/.vnc/passwd in DES

use vncpasswd.py

* + - SSH Hijacking
      * + username/password harvesting
      * remote forwarding 🡪 ssh -f -N -T -R port:localhost:port url
      * local forwarding 🡪 ssh -L port:url:port URL
      * dynamic application-level port forwarding
      * public log authentication 🡪 SSO using SSH agents
      * X-server forwarding
        + chmod -R 700 ~/.ssh
        + chmod -R 600 ~/.ssh/config
        + chmod -R 600 ~/.ssh/id\_rsa
        + chmod -R 600 ~/.ssh/known\_hosts
      * Metasploit auxiliary/scanner/ssh/ssh\_login 🡪 ssh\_to\_metepreter
      * Metasploit 🡪 multi/gather/ssh\_creds

gather ssh creds

* + - * + to verify pub/priv key pair match

ssh-keygen -y -f <priv key>

then read pub key to verify

read known\_hosts files

* + - * Metasploit 🡪 scanner/ssh/ssh\_login\_pubkey
        + creds cmd prints valid creds from Credential table verified by Metasploit
      * SSH agent forwarding
        + enabled in .ssh/config or etc/ssh/sshd\_config

temp file pointer is created in /tmp that points to users SSH agent

compromising agent allows password circumnavigation

not require to establish SSH remote connection for hosts configurated to use SSH agents

agent . <PID>

ps -ef | grep -i <PID>

ID user account and agent

* + - Exploiting Legacy Services
      * Telnet, RSH/Rlogin/REXEC (R-services)

rhosts file

group name with (+) is trusted, w/ (-) is untrusted

settings controlled by user account

mount NFS share to modify users rhost file and login as any user desired

* + - Lateral Movement in Windows
      * + PsExec, WinRM, RDP, WMI, etc
      * RPC/DCOM – T1175
        + windows API utilizes the COM component to interact between software object

can be accessed through higher TCP port tanges

* + - * + remote communication through application macros
        + PtH attacks against NTLM – T1075

bypass need by PsExec SysInterals command

Metasploit psexec module

* + - * + can also use Mimikatz
      * Windows Remote Access
        + RDP – need to be in RDP group in a domain

prone to MiTM due to hard-coded RSA private key in mstlsapi.dll

retrieve with Nessus Plugin ID:18405

* + - * + WinRM – enables RCE

use Invoke-Command or Enter-PsSession

* Maintaining Persistence
  + - backdoors, persistence
  + Windows Logon Scripts – T1037
  + Windows Scheduled Taks – T1053
  + Local Job Scheduling – T1168
  + Launch Daemon – T1160
  + Malicious Software Update – CAPEC-186
  + Create Account – T1136
* Covering Your Tracks
  + Clearing Command History
    - unset HISTFILE – temp hist not written to disk
    - export HISTFILE=0
    - history -c – clear temp history
    - set =o history – prevent cmds from recording
  + Timestomping
    - * modify file timestamps
      * meterpreter timestomp feature
    - in Windows
      * Powershell
        + LastWriteTime
        + LastAccessTime
        + CreationTime
      * powershell -command Get-Process \* -IncludeUserName
      * to download a remote exe
        + first define newobj = new-object System.Net.WebClient
        + newobj.DownloadFile(“/url/”, “\path\”)
      * all accessed by Get-Item
  + File Deletion
    - /var/log/\* or Event Viewer – mitigate with Syslog

Chapter 11 – Physical Penetration Testing

* Layers 🡪 Monitoring, Detecting, Preventing
  + CAPEC-514
  + Bypassing Physical Security – 390
    - Physical Locks – 391
  + Lock Bumping – 392
  + Bypassing Electronic Locks and Access Controls – 395
  + Physical Theft – 507
  + Obfuscation – 607
* Keeping the Honest People Honest
  + Environmental Threats
    - * external fences, perimeter barrier
      * access control points
    - piggybacking, tailgating – no consent from victim
  + Physical and Environmental Protection
    - NIST 800-53 PE; FIPS 199
      * PE chart p349-350
      * PE 1-20
    - same requirements are technical assessment required
      * RFID cloning
        + easy, legacy
* Physical Locks and Security
  + - entrance locks, deadlocks 🡪 delaying device
    - security lighting, early warning detection, fencing
  + Mechanical Locks
    - electronic locks
      * fail safe/fail secure
    - Warded Lock
      * obstructions around keyhole to prevent from opening with wrong key
    - Tumbler Locks
      * pin – driver + key pins with shear line
      * wafer – protruding wafers on sides of cylinder
      * lever
    - Combination Locks
      * single dial, multiple dial, padlock shims
      * speed dial
      * bruteforce methods
    - Cipher Locks
      * common for sensitive access controls
      * cipher lock codes are often shared
        + PSK, maintain ACLs
        + Bruteforce, strong magnets
* Basic Tools and Opening Techniques
  + - basic lock kit – tension wrench, lock picks, bump key
  + Single-Pin Picking
    - pin tumbler lock
  + Jiggling and Raking
    - Jiggling – stick lockpick into keyway and jiggle pick in an upward and downward motion
      * manual pick gun
    - Raking – dragging lockpick back and forth in keyway, rapidly
      * harshest, not forensically sound
  + Lock Bumping and Bump Keys
    - Lock Bumping – Bruteforce method for opening pin tumbler lock similar to jiggling
      * arbitrarily manifest shear line with a rubber hammer
        + countered with trap pin or shallow drilling
  + Alarms and Early Wrning Systems
    - wireless alarms operate between 400-900 MHz
    - Passive Infrared Sensors – counter with Styrofoam
    - ultrasonic sensor – used in small areas
      * microwave sensors
      * magnetic switches
      * egress sensors
* Physical Device Security
  + Cold Boot Attack
    - recover disk encryption keys from RAM when power is cycled
      * when drives are cold/frozen
    - reboot and insert USB with auto key finding software (AESKeyFinder/RSAKeyFinder)
  + BIOS Attacks
    - can be accessed with specific keys on bootup
    - reprioritize boot order
    - counter with BIOS password of UEFI
  + USB Keylogger
    - positioned inline or within keyboard itself
    - keelog

Chapter 12 – Reporting and Communication

* Writing the Pentest Report
  + - formal, impersonal, avoid accusations
    - crop photos, annotate
    - provide tangible evidence
    - scan data, screenshots, notes
    - minimize false positives
* Drafting the Report
  + - based on assessment type + PTES
      * Executive-level reporting 🡪 BLUF
      * technical reporting
    - PCI Suggested
      * Execute Summary
      * Statement of Scope
      * Statement of Methodology
      * Statement of Limitations
      * Testing Narrative
      * Segmentation Testing Results
      * Findings
      * Tools Used
      * Cleaning Up the Environment
    - GSA 🡪 FedRAMP for CSP
  + Executive Summary
    - BLUF
      * Timeline
      * Methodology
      * Observations
      * Issues
      * High Level/Significant Findings
    - High Level/Significant Findings
      * shared Local admin creds
      * weak password complexity
      * plaintext passwords
      * SQL injections
    - Remediation
  + Scope
    - found in SOW
  + Limitations
    - hours, bandwidth, legacy systems, overhead
  + Methodology – SOW
    - * testing narrative document
      * document issues encountered
      * address every objective and method agreed upon
    - PCI PTG + FedRAMP PTGV2.0
  + Network Info Gathering
    - OSINT
    - Enumerate + Inventory – Live Network Endpoints
      * network service availability
    - Fingerprinting OS + network
    - Perform Vuln IDs
  + Web Application/API Testing
    - Perform OSINT on target web app
    - ID target application architecture
    - ID account roles and authorization bounds
    - Map all content and functionality
    - ID all user controlled input entry points
    - perform web app server config checks
  + Mobile Platform and Application Testing
    - OSINT on target web app
    - map all content and functionality
  + Social Engineering Testing
  + Simulated Internal Attack Testing
    - OSINT personnel responsible for target system management
    - perform scoping exercise with customer to determine potential attack vector
    - perform vuln ID
  + False Positives
    - verify findings with credentialed checked or cross-referencing
  + Findings and Remediation
    - include actionable items only
      * when it occurred
      * who was involved
      * what happened
      * where did it occur
      * why did it happen
    - use Finding ID category
      * vulnerability
      * impact
      * risk rating
      * remediation
      * references
  + Web Application/API Exploitaiton
  + Mobile Application Exploitaion
  + Network Exploitation
  + Social Engineering Exploitation
  + Simulated Internal Attack Exploitation
  + Post-exploitation
  + Conclusion
    - appendix
    - table of contents
    - table of figures
  + Post-engagement cleanup
    - ensure environment is back to normal
    - clear away exploits, tools, shells, etc
* Report Handling
  + - secure handling and disposition of pentest report
      * should be an agreed upon method in RoE, storage time too
      * ensure confidentiality and integrity
    - Destroy all secondaries versions of report
* Post-Report Delivery Activities
  + Customer Debriefing
    - handling with delicate sensitivity
    - group discussion lessons learned
  + Follow-up Actions
    - retesting, be wary of scope creep
* Communication is Key
  + RoE, Communication and Escalation paths
  + Situational awareness
    - de-escalation and de-confliction
  + communication triggers
    - critical findings
    - stage completion
    - risky stage initiation
    - indicators of prior compromise
    - goal reprioritization, readjustment of priorities

END