Cloud Computing [1]

- Access to shared pool of configurable computing resources
- Released with minimal management effort or service provider interaction
- Promotes availability

Properties

- Broad network access
  - Capabilities over network
  - Accessed through standard mechanisms
  - Use of platforms (e.g. phones, laptops, and tablets)
- Rapid elasticity
  - Expand/reduce resources specific to service requirement
  - E.g. large number of resources for duration of specific task, released after completion
- Measured service
  - Automated control and optimisation of resources
  - Take advantage of something (appropriate to type of service)
  - E.g. storage, processing, bandwidth, and active user accounts
  - Monitored, controlled, and reported
  - Provides transparency for both provider and consumer
- On-demand self-service
  - Cloud Service Consumer (CSC)
    - Provision computing capabilities
    - E.g. server time and network storage
    - Automatic
    - No human interaction with each service provider
    - On demand – resources not permanent parts
- Resource-pooling
  - Serve multiple CSCs using multi-tenant model
  - Different physical and virtual resources assigned
    - According to consumer demand
  - CSC doesn't know the exact location of provided resources
    - May be able to find country, state, data centre etc.
  - Resources
    - Storage
    - Processing
    - Memory
    - Network bandwidth
    - VMs
  - Sometimes pool resources between different parts of same organization

Network Penetration Testing [2]

Passive Information Gathering Reconnaissance – OSINT

- Collection of **Open-Source Intelligence (OSINT)**
  - Passively collecting data about company
  - Info collected from the internet
  - Alias: **reconnaissance**
- If target is an external web application
  - Execute info-gathering phase
- If target is an intranet or new website not in production enviro yet
  - OSINT is useless (unless client has asked you for this task separately)

Intranet [3]

- Computer network for sharing info, collab tools, OS, and other services within an organization
- Exclusion of access by outsiders

Information Gathering

- Starts with online research as to target's online presence
- Gather info:
  - Company
    - Location & address
    - Email addresses
    - Other companies acquired
    - DNS
    - Business type
    - Structure
    - Blog articles
    - Social network data
    - Cached contents on web
    - Info leaks (e.g. passwords, sensitive info)
  - Employee
    - Names
    - Email addresses
    - Numbers
    - Job position in company
    - Social network data
  - Web application
    - Web-based vulnerabilities on dumpsite
    - Web-page crawling
    - Prog languages used
    - Passive scanning using Burp
- Start on **target's website** – find most info there

Web Search Engines

- Use **different search engines** other than Google
    - E.g. Chinese, Russian search engines (p. 189)
    - Dark web engines
- **Google dorks**
    - Query search engine to reveal sensitive info about the target
    - Exploit-DB website
    - Accomplish
        - Foothold (secure position) on web server
        - Revealing sensitive directories
        - Search for vulnerable files
        - Search for vulnerable servers
        - Reveal detailed error messages
        - Search for target network's vulnerability data
        - Search for miscellaneous devices that belong to target (e.g. IP cameras – cameras with web technologies)
        - Web server information
        - Files with credentials and confidential info
        - Login pages
- **Most popular** Google dorks queries:
    - Domains/subdomains: site [target domain name]
    - Files: filetype [file extension]
    - Strings in URL: inurl [search criteria in the URL]
    - Strings in the title: intitle [search criteria in the title]

Online tools

- Online vulnerabilities for servers and IOT devices: shodan.io
- Dumped leaked info: pastebin.com
- Test and source code leak: github.com
- Online Swiss Army knife tools: dnsstuff.com
- Info about target website:
    - toolbar.netcraft.com/site_report?url=[target domain name]
    - searchdns.netcraft.com

Internet of Things (IOT) Devices [4]

- Connect wirelessly to a network
- Ability to transmit data
- E.g. wearables, smart speakers, smart TVs etc.

WHOIS lookup

- Domain name registered in public WHOIS database
- Response to WHOIS request reveals 'juicy' information
    - Names
    - Phone numbers
    - Email addresses
    - Physical addresses
    - Domain expiry dates
    - DNS servers

Domain name system – DNS enumeration

- Reveal info regarding

    - Domain names and IP addresses assigned to target
    - Route between us and destination
- Finds (resolves) domain names to its IP addresses
- Reasons to use:
    - Find out if DNS server allows zone transfer, if so
        - Reveal hostnames and IPs of internet-accessible systems
    - Brute-force – identify domain/subdomains
    - Vulnerable services (e.g. FTP)
    - Find interesting remote admin panels
    - Find misconfigured and/or testing servers (test.domain-name.com)

Zone Transfer [5]

- Copying contents of zone file on primary to secondary DNS server
- Provides fault tolerance
    - Through synchronization of zone file in primary DNS server with zone file in secondary DNS server
- Zone file [6]
    - "Text based file with a format defined in RFC 1035 and 1034 and stored on DNS server."
    - Contains:
        - IP
        - Name data
        - MX and other service records
    - Contain **glue data** which connects them to other DNS servers

Gathering Email Addresses

- **theharverster** script
    - Python tool/script
    - Searches for email addresses and domains
    - Uses popular search engines

Active Information Gathering – services enumeration

- Identify
    - Live hosts
    - Services running on those hosts
- Some skip this test – go straight to vulnerability assessment
    - Execute fancy scanners (e.g. Nessus or Nexpose)
- Four steps:
    1. Get IPs/ranges from client or employer
        a. If internal project, project manager will help
    2. Identify live hosts
    3. List open ports/services on each host
    4. Probe (monitor) each service for more info

Identifying live hosts

- Identify whether host is either:
    - Up and running, or
    - Protected by firewall
- Nmap used

    - Quick ping scan

Identifying open ports/services

- After ping scan
    - Reveal open ports and services
- Use Nmap script to probe each service (p. 194)
    - TCP scan – intranet
    - UDP scan – intranet
    - TCP scan – from the internet (outside boundary)
    - UDP scan – from internet (outside boundary)

Service probing and enumeration

- Take information from above and probe aggressively (pp. 195-198)
- Can be time-consuming

Vulnerability assessment

- Some Nmap scripts will check for vulnerabilities
    - E.g. entering option –script=http*
        - Execute all HTTP scripts, including ones that check for vulnerabilities, e.g. http=vuln-cve2010-2861 (https://nmap.org/nsedoc/scripts/http-vuln-cve2010-2861.html)
- Automatic scanners are used (Nessus or Nexpose)
    - **Offer scanners in the cloud as well**
        - E.g. InsightVM
    - Identify vulnerabilities in the network infrastructure
- Role is taking the results and make sure the flaws exist (that they're **not false positives**)

OpenVas

- Used to practice vulnerability assessment
- Is a vulnerability scanner

Exploitation

- Exploit vulnerability found target machine
- Gets a remote shell

Finding exploits

- Vulnerability will tell you where to find exploit in order to replicate it
- Where to find:
    - Google
    - Exploit-db at exploit-db.com
    - Searchsploit tool
    - Metasploit (search command)
    - Security Focus at sercurityfocus.com
    - GitHub

Listener setup

- Before uploading and executing the payload
    - Set up and execute a **listener** (p.200)
        - "A listener is a component that waits for an incoming connection from an exploited system" [7]

Generating a shell payload using msfvenom

- (p. 201)

Custom Shells

- Can create your own shells (p. 202)

Privilege escalation

- After exploiting vulnerability, most of the time, you will get a limited shell
- Get admin account on victim machine
- Methodologies:
    - Transfer file to victim machine that will make you have root shell
    - Copy-pasting a PowerShell payload (e.g. Empire PowerShell)
    - Use Metasploit/Meterpreter to escalate privileges
    - Manually searching for misconfigured parameters to get an admin/root shell

File Transfers

- Example:
    - Have limited shell into victim's machine
    - OS is Linux
    - Upload Dirty COW to remote server to execute
    - Steps (p. 203)
- Transfer file using PowerShell on Windows

Using PowerShell

- Script on (p. 204)

Using VBScript

- For older versions of Windows with no PowerShell
- Script on (p. 204)

Administrator or root

- Time consuming to type all these commands
- Upload scripted file to victim's machine (from File Transfers)
- Windows privilege escalation exploits often written in Python
    - Convert to executable
- (pp. 205-206)

Cloud Penetration Testing [8]

- Cloud-based services preferred choice for businesses because:
    - Low initial cost
    - Scalability
    - Speed
- Some services have their own policies with penetration tests
- Technical and legal challenges to perform these tests:
    - Don't own the cloud infrastructure/platform/software as an entity but as a service

Why clouds are vulnerable

- Insecure APIs
    - Lead to a large-scale data leak (E.g. Venmo, Airtel etc.)

    - HTTP methods like PUT, POST, DELETE in APIs
        - Allow uploading malware on server or delete data
    - Improper access control
    - Lack of input sanitisation
- Server misconfigurations
    - Capital One data leak
        - Compromise data of roughly 100 million Americans and 6 million Canadians
        - Common
            - Improper permissions
            - Not encrypting data
            - Differentiation between private & public data
- Weak credentials
    - Brute force attacks
    - Automated tools to make guesses
    - Complete account takeover
    - Reused passwords
    - Easily rememberable passwords
    - Verified during cloud penetration testing
- Outdated software
    - Don't use a streamlined update procedure
    - Disable auto updates themselves
    - Cloud services outdated (identify using automated scanners)
- Insecure coding practices
    - Build cloud infrastructure build for as cheap as possible
    - Bugs like SQLi, XSS, CSRF
    - Most common labelled as OWASP top 10
        - Root cause for compromission of cloud-based services

Challenges in cloud penetration testing

- Lack of transparency
    - For uncommon cloud servicers, datacentres managed by third parties
    - User unaware of where data is stored, and hardware/software config being used
    - Exposes to security risks
        - E.g. provider may be hoarding sensitive data without user knowing
    - Popular CSPs like AWS, Azure GCP, etc. known to conduct in-house security audits
    - Lack of transparency means
        - Resources cannot be audited by security auditor of your choice
        - May be unable to respond if underlying resources are hacked
- Resource sharing
    - Share resources across multiple accounts
        - Challenging during cloud penetration testing
    - Sometimes don't take steps for segmentation of all users
    - If need to be compliant with PCI DSS (ensure all companies access, process store or transmit credit card information [11])
        - All other accounts sharing resource and CSP should be PCI DSS compliant too

    - Multiple ways to implement cloud infrastructure
    - Complexity hinders process of cloud penetration testing
- Policy restrictions
    - Each CSP has its own policy with cloud penetration testing
    - Defines endpoints and types of tests
    - Some require submitting an advanced notice
    - Limit scope of conducting them
    - Popular CSPs
        - AWS
            - Can be performed without prior notice
            - Attacks not permitted:
                - DOS and DDoS
                - DNS zone walking
                - Port, Protocol, or Request flooding attacks
            - If performing a network stress test – separate policy
        - Azure
            - Allowed on eight Microsoft products
            - Attacks not permitted:
                - Penetration tests on other azure customers or data other than yours
                - DOS and DDoS or tests that create huge traffic
                - Intensive network fuzzing attacks on Azure VMs
                - Phishing or social engineering attacks against Microsoft's employees
                - Violating Acceptable Use Policy
        - GCP
            - Follow Acceptable Use Policy and ToS
            - No need to inform Google before conducting tests
            - Attacks not permitted
                - Piracy or any illegal activity
                - Phishing
                - Spamming
                - Distributing trojans, ransomware, etc during the tests
                - Violating rights of other GCP users or conducting penetration tests on them
                - Violating or trying to circumvent ToS
                - Interfere with equipment supporting GCP
    - Other factors
        - One machine can host multiple VMs
            - Adds to the scale of cloud pen testing
        - Scope of tests vary from user software to service provider software
            - Add complexity
        - When encryption added
            - Company may not be willing to share encryption keys
            - Not good for auditors (investigates financial records)

Step-by-Step Cloud Penetration Testing

1. Understand cloud service provider's policies
2. Create testing plan
    - Map out endpoints for testing
        - E.g. user interface, APIs, subnetworks etc
    - Which endpoints to exclude based on policy, user permissions etc.
    - How well app server and VMs can take the load of the tests
    - Laws that need to be followed while testing
    - Which tools to use and types of tests on which endpoints
        - Automated or Manual
    - Get approval from client & inform when to begin
3. Execute plan
    - Tools
        - AWS Inspector
            - Customized security solution for AWS
            - Basic min.
            - Preliminary
        - S3Scanner
            - Open-source
            - Scan S3 buckets for misconfigs and dump their data
        - MicroBurst
            - Collection of PowerShell scripts
            - Scan Azure services for security issues
        - Azucar
            - Uses PowerShell
            - For Azure
        - Cloudsploit
            - Open-source
            - Scan multiple types of CSPs (e.g. Azure, AWS, GCP, OCI, etc.)
4. Detect and fix vulnerabilities
    - Some automated tools may generate false positives
        - Verify each is exploitable before reporting it
        - Repeat for each network layer
    - Report generation
        - Present vulnerabilities in an understandable matter
        - So that they take the vulnerabilities seriously or not
        - Make it well organized and categorized
        - Based on type and level of threat
    - Get in touch with developers to patch them
        - Changes from minor to major
    - If no vulnerability detected, change plan, and perform more elaborate tests

Penetration Testing Methodologies

- Services fall into
    - Infrastructure (IaaS)
    - Platform (PaaS)
    - Software as a service (SaaS)

        - End user doesn't own environment
- Policies
    - Security testing for User-Operated Services
        - Excluded tactics related to disruption of business continuity
        - E.g. DoS
    - Vendor Operated Services
        - Cloud offerings owned and managed by third-party vendor
        - Restricted to implementation and config of cloud environment
            - Not underlying infrastructure
        - E.g. services like Cloudfront and API Gateway config may be pen tested
            - Hosting infrastructure isn't allowed
    - EC2 often pen tested, specifics allowed:
        - APIs (e.g. HTTP/HTTPS)
        - Web and mobile applications hosted by organization
        - Application server and associated stack (e.g. prog langs like Python, React)
        - VMs and OS
    - What can't be tested:
        - Legal and technological constraints
            - Services and applications that belong to AWS
            - Physical hardware, underlying infrastructure, or facility that belong to AWS
            - EC2 environments that belong to other orgs
            - Security appliances managed by other vendors without permission
            - Amazon's small or micro Relational Database Service (RDS)
- Differences from traditional pen testing
    - Refer to ownership of systems
    - Violation of AWS acceptable use policies with traditional 'ethical hacking'
    - Only focus of user-owned assets
    - E.g.
        - Compromising AWS IAM keys
        - Testing S3 bucket logs
    - Specific to AWS cloud
- Planning pen test within cloud
    1. Define scope, including AWS environment and target systems
    2. Run your own preliminary
    3. Determine type of pen test you'd like conduced (e.g. black box, white box, grey box)
    4. Outline expectations for both internal stakeholders and company
    5. Timeline for assessment, receive formal report, potential remediation, and follow-up testing
    6. Develop protocol and rules of pen test
        a. Reveals if client may already have been breached or is under an ongoing (live) attack
    7. Obtain written approval to conduct test by client
        a. Fill out pen test request form
        b. Tell AWS the date testing will take place
        c. Tell AWS IP range the scan or testing will come from
        d. Tell AWS IP range being tested (scope)

Remediation Testing [12]

- What is it
  - Retesting vulnerabilities during pen test
  - Ensure issues arose during pen test are properly identified, fixed and no longer a threat
- Goals
  - Ensure solution have been put in place to resolve identified issues have been implemented properly
  - Vulnerabilities secured
  - Ensure issues are no longer there that appeared in pen test

Web Application Vulnerabilities

File Inclusion

- Include file in URL (using path)
  - File local to server – **Local File Inclusion**
  - Or, the path can point to a remote file – **Remote File Inclusion**

Local File Inclusion

- Allow directory traversal characters (e.g. ../) injected
- Example:
  - http://domain_name/index.php?file=../../../../etc/passwd

Remote File Inclusion

- Remote file outside boundaries of web server

Cross-Site Scripting

- XSS
- Exploited when attacker can successfully execute any type of script on victim's browser
- Why the flaws exist
  - Didn't validate request
  - Incorrectly encoded response of application
- JavaScript most common
- Types
  - Stored
  - Reflected
  - DOM Injection

Reflected XSS

- Page displays user something that can be manipulated dynamically through:
  - URL
  - Or, body of page

Bibliography

1. Stallings, W. and Brown, L., *Computer Security Principles and* Practice (2018), 4th Edition, Pearson, Hudson Street, New York, pp. 446-470 [Accessed 15 Oct. 2021]
2. Khawaja, G., *Practical Web Penetration Testing (2018)*, Packt Publishing, Birmingham, [online] pp. 187-206 [Accessed 16 Oct. 2021]

3. *Intranet* (2021), [Online] Available at: https://en.wikipedia.org/wiki/Intranet [Accessed 16 Oct. 2021]
4. Posey, B., *IoT devices (internet of things devices)* (2021), [Online] Available at: https://internetofthingsagenda.techtarget.com/definition/IoT-device [Accessed 16 Oct. 2021]
5. Gregg. M., *Zone Transfer* (2006), [Online] Available at: https://www.sciencedirect.com/topics/computer-science/zone-transfer#:~:text=Zone%20transfer%20is%20the%20process,in%20a%20secondary%20DNS%20server. [Accessed 16 Oct. 2021]
6. Steve, *DNS Zones and Zone Files Explained* (2020), [Online] Available at: http://www.steves-internet-guide.com/dns-zones-explained/#:~:text=A%20zone%20file%20is%20a,to%20the%20other%20DNS%20servers. [Accessed 16 Oct. 2021]
7. *Listener*, [Online] Available at: https://docs.rapid7.com/metasploit/listeners/ [Accessed 16 Oct. 2021]
8. Varghese, J., *A Complete Guide on Cloud Penetration Testing* (2021), [Online] Available at: https://sussex.primo.exlibrisgroup.com/discovery/fulldisplay?docid=cdi_askewsholts_vlebooks_9781788628723&context=PC&vid=44SUS_INST:44SUS_VU1&lang=en&search_scope=MyInst_and_CI_no_BLDS&adaptor=Primo%20Central&tab=MyInst_and_CI_no_BLDS&query=any,contains,Gus%20Khawaja&offset=0 [Accessed 17 Oct. 2021]
9. *Penetration Testing Methodologies*, [Online] Available at: https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies [Accessed 17 Oct. 2021]
10. Caudill, B., *Penetration Testing in the AWS Cloud: What you Need to Know*, [Online] Available at: https://rhinosecuritylabs.com/penetration-testing/penetration-testing-aws-cloud-need-know/ [Accessed 17 Oct. 2021]
11. *What is PCI?*, [Online] Available at: https://www.pcicomplianceguide.org/faq/#1 [Accessed 17 Oct. 2021]
12. Borrego, A., *Penetration Testing Remediation FAQ's* (2020), Available at: https://www.emagined.com/blog/penetration-testing-remediation-faqs [Accessed 18 Oct. 2021]