Web Application Security

Dr. Digvijaysinh Rathod
Associate Professor
School of Cyber Security and Digital Forensics
National Forensic Sciences University

digvijay.rathod@nfsu.ac.in

Introduction to Vulnerability Assessment, Life cycle of Vulnerability Assessment

Vulnerability Scanners, Unknown Vulnerability, False Positive

Dr. Digvijaysinh Rathod

HTTP Server virtual host types

- ✓ Introduction
- ✓ The Need for VAPT
- ✓ What is VAPT?
- ✓ Approaches
- ✓ Methodology
- ✓ Services
- ✓ Issues We Identify
- ✓ Tools

SIMPLIFIED ATTACK LIFECYCLE

Initial Reconnaissance: It has two main steps, Selection of target and Research of Target

Penetration: The intruder uses certain methods to compromise the target.

Gaining Foothold: Maintain foothold of the compromised system. It is generally achieved by setting up a backdoor so that the machine can

be accessed later.

SIMPLIFIED ATTACK LIFECYCLE

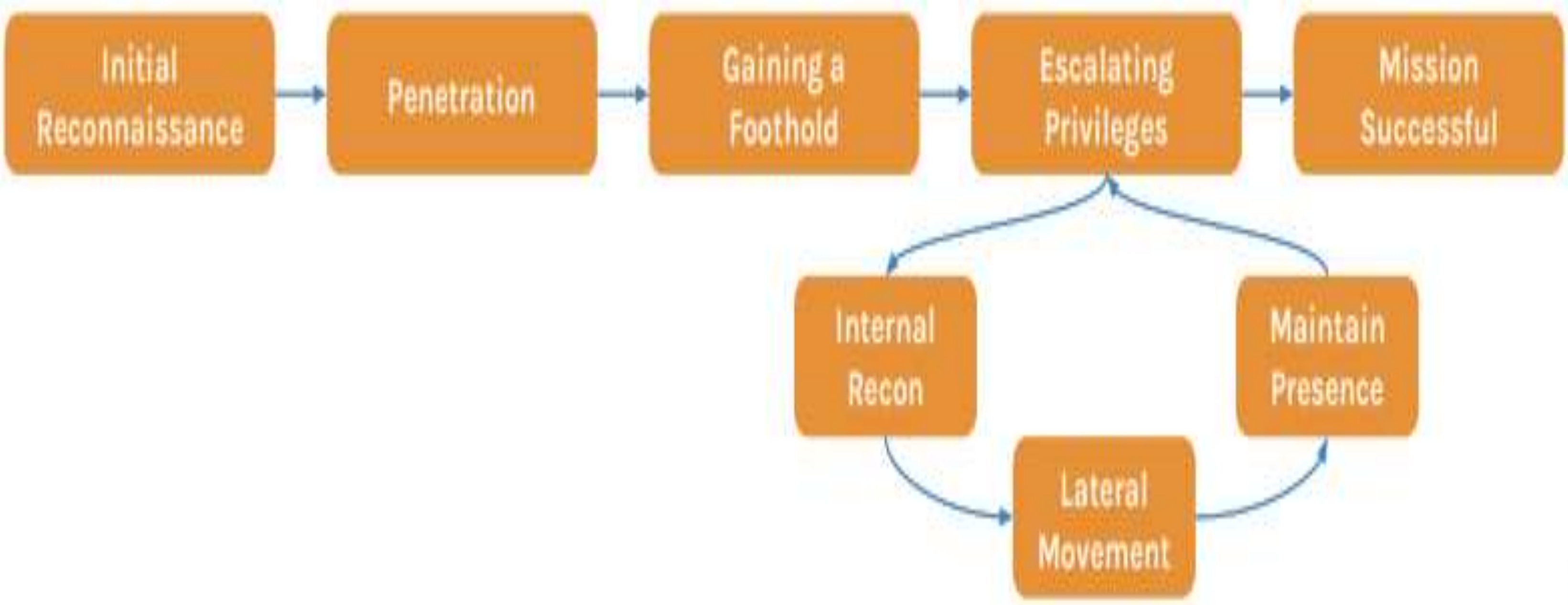
Escalate Privileges: Intruder will obtain a higher level access to the compromised machine by multiple methods with the aim of obtain the administrator login.

Lateral Movement: The intruder generally does not find the desired information on the first machine he has compromised. The attacker now tries to expand the exploitation process to other systems within the same network. If the intruder is caught during lateral movement we get the exact intent of the intruder or exactly what information he might be after.

SIMPLIFIED ATTACK LIFECYCLE

Maintain Presence: The intruder will ensure remote access to the complete environment by installing multiple variants of malware backdoors.

SIMPLIFIED ATTACK LIFECYCLE



Initial Reconnaissance

Penetration

Gaining a Foothold

Escalating Privileges

Mission Successful

Internal Recon

Lateral Movement

Maintain Presence

What is VAPT ?

- ✓ A form of stress testing, which exposes weakness or flaws in a computer system
- ✓ The art of finding an Open Door
- ✓ A valued Assurance Assessment tool
- ✓ VAPT can be used to find flaws in
  - ✓ Specifications, Architecture, Implementation, Software, Hardware, and many more..

What is VAPT ?

- ✓ Vulnerability assessment is the process of identifying, quantifying, and prioritizing the vulnerabilities in a system.
- ✓ Penetration test is an attack on a computer system that looks for security weaknesses, potentially gaining access to the computer's features and data.

APPROACHES

- ✓ Typically VAPT is performed using three approaches
  - ✓ Blackbox
  - ✓ Graybox
  - ✓ Whitebox

BLACKBOX TESTING

- ✓ Also known as “Zero-Knowledge” testing, tester needs to acquire Knowledge and Penetrate
  - ✓ Acquire Knowledge using tools or Social Engineering Techniques
  - ✓ Publicly available information may be given to the penetration tester
- ✓ Benefit:
- ✓ Intended to closely replicate the attack made by an outsider without any information of the system. This kind of testing can give an insight of the robustness of the security when under attack by script kiddies

GRAYBOX TESTING

- ✓ This usually is a combination of blackbox and whitebox testing
- ✓ The aim of this testing is to search for the defects if any due to improper structure or improper usage of applications
- ✓ The tester simulates an inside Employee. The tester is given an account on the internal network and standard access to the network. This test assess internal threats from employees within the company

GRAYBOX TESTING

- ✓ **Benefit:**

Takes the straightforward technique of black-box testing and combines it with the code-targeted systems in white-box testing.

WHITEBOX TESTING

- ✓ Also known as “Internal Testing”, testers are given full information about the target system they are supposed to attack. Information

  includes:

  - ✓ Technology overviews
  - ✓ Data flow and network diagrams
  - ✓ Code snippets

WHITEBOX TESTING

- ✓ Benefits:
  - ✓ Reveals more vulnerabilities and may be faster
  - ✓ Compared to replicate an attack from a criminal hacker that knows the company infrastructure very well.
  - ✓ Checks robustness against internal threats

METHODOLOGY

- ✓ It is the process intended to reveal flaws in the security mechanisms, protect data and maintain functionality as intended.

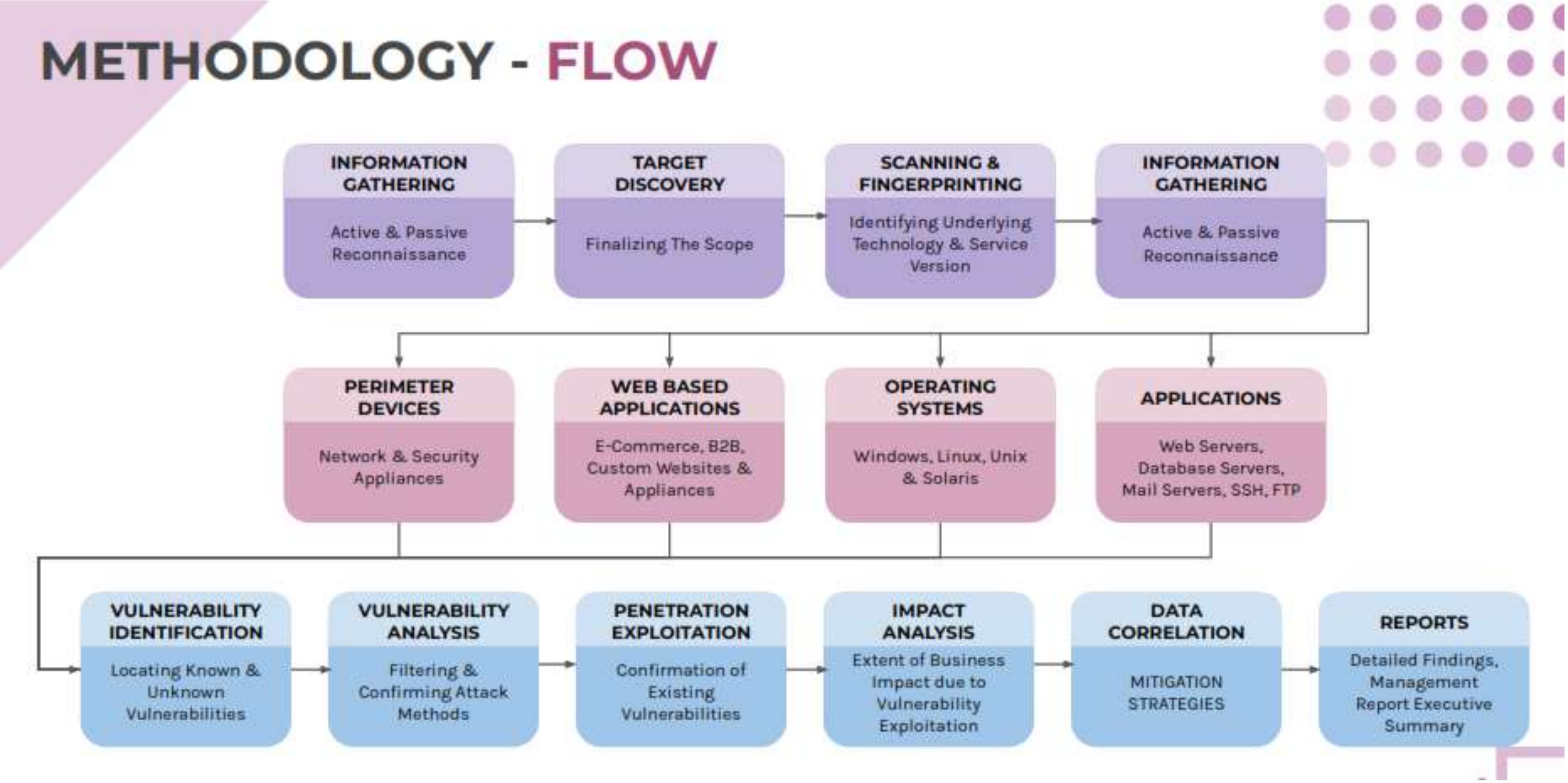
METHODOLOGY

- ✓ Vulnerability Assessment and Penetration Testing
- ✓ methodology is derived from the following best practices:
- ✓ OWASP, OSSTMM and WASC security guidelines
- ✓ Business logic vulnerability verification
- ✓ False positive elimination
- ✓ Utilization of automated commercial, proprietary and other industry leading tools
- ✓ Manual testing for identification and verification of critical and exploitable vulnerabilities
- ✓ Reassessment to ensure all gaps are fixed

METHODOLOGY

- ✓ Vulnerability Assessment and Penetration Testing
- ✓ methodology is derived from the following best practices:
- ✓ OWASP, OSSTMM and WASC security guidelines
- ✓ Business logic vulnerability verification
- ✓ False positive elimination
- ✓ Utilization of automated commercial, proprietary and other industry leading tools
- ✓ Manual testing for identification and verification of critical and exploitable vulnerabilities
- ✓ Reassessment to ensure all gaps are fixed

METHODOLOGY Flow



METHODOLOGY - FLOW

INFORMATION GATHERING
Active & Passive Reconnaissance

TARGET DISCOVERY
Finalizing The Scope

SCANNING & FINGERPRINTING
Identifying Underlying Technology & Service Version

INFORMATION GATHERING
Active & Passive Reconnaissance

PERIMETER DEVICES
Network & Security Appliances

WEB BASED APPLICATIONS
E-Commerce, B2B, Custom Websites & Appliances

OPERATING SYSTEMS
Windows, Linux, Unix & Solaris

APPLICATIONS
Web Servers, Database Servers, Mail Servers, SSH, FTP

VULNERABILITY IDENTIFICATION
Locating Known & Unknown Vulnerabilities

VULNERABILITY ANALYSIS
Filtering & Confirming Attack Methods

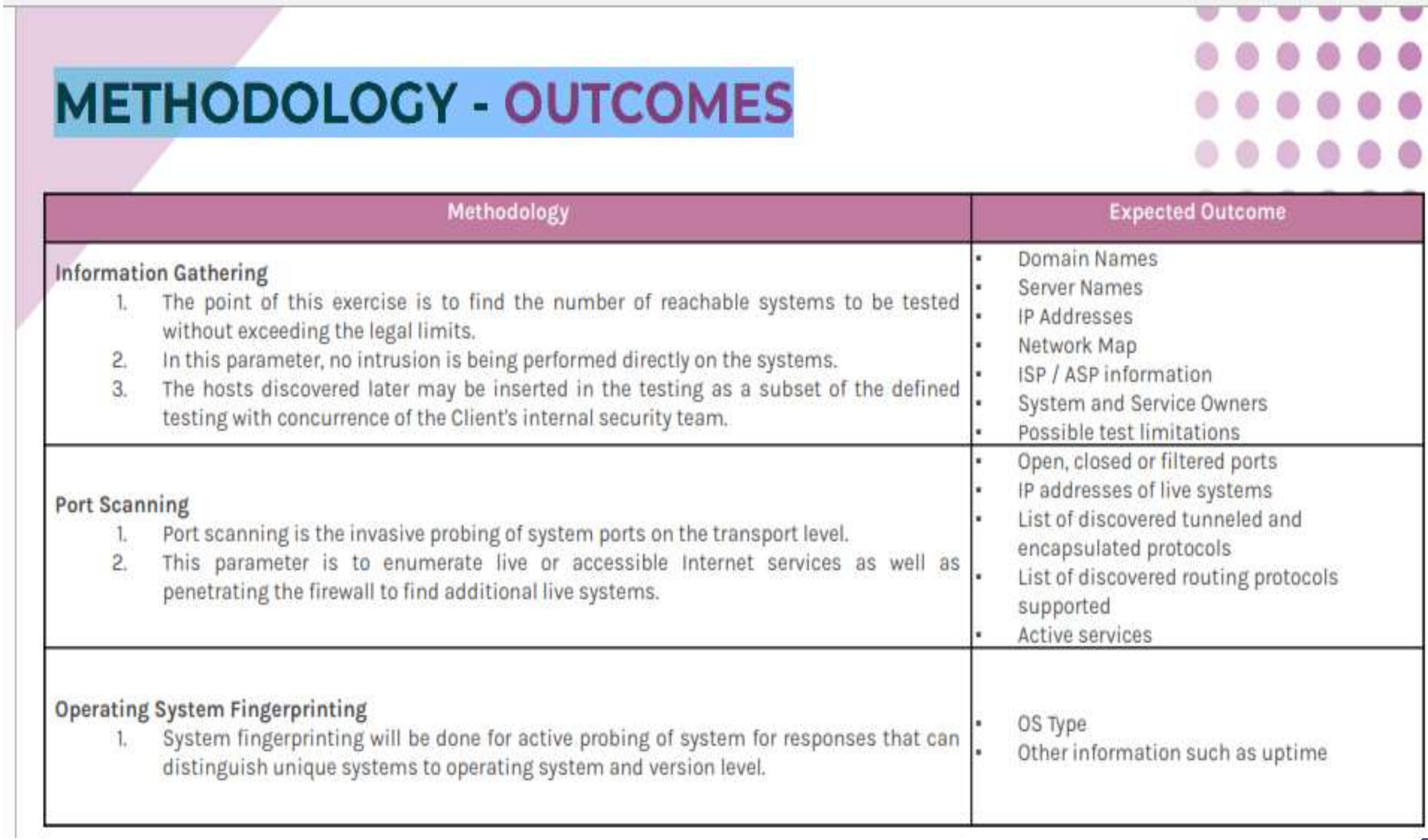
PENETRATION EXPLOITATION
Confirmation of Existing Vulnerabilities

IMPACT ANALYSIS
Extent of Business Impact due to Vulnerability Exploitation

DATA CORRELATION
MITIGATION STRATEGIES

REPORTS
Detailed Findings, Management Report Executive Summary

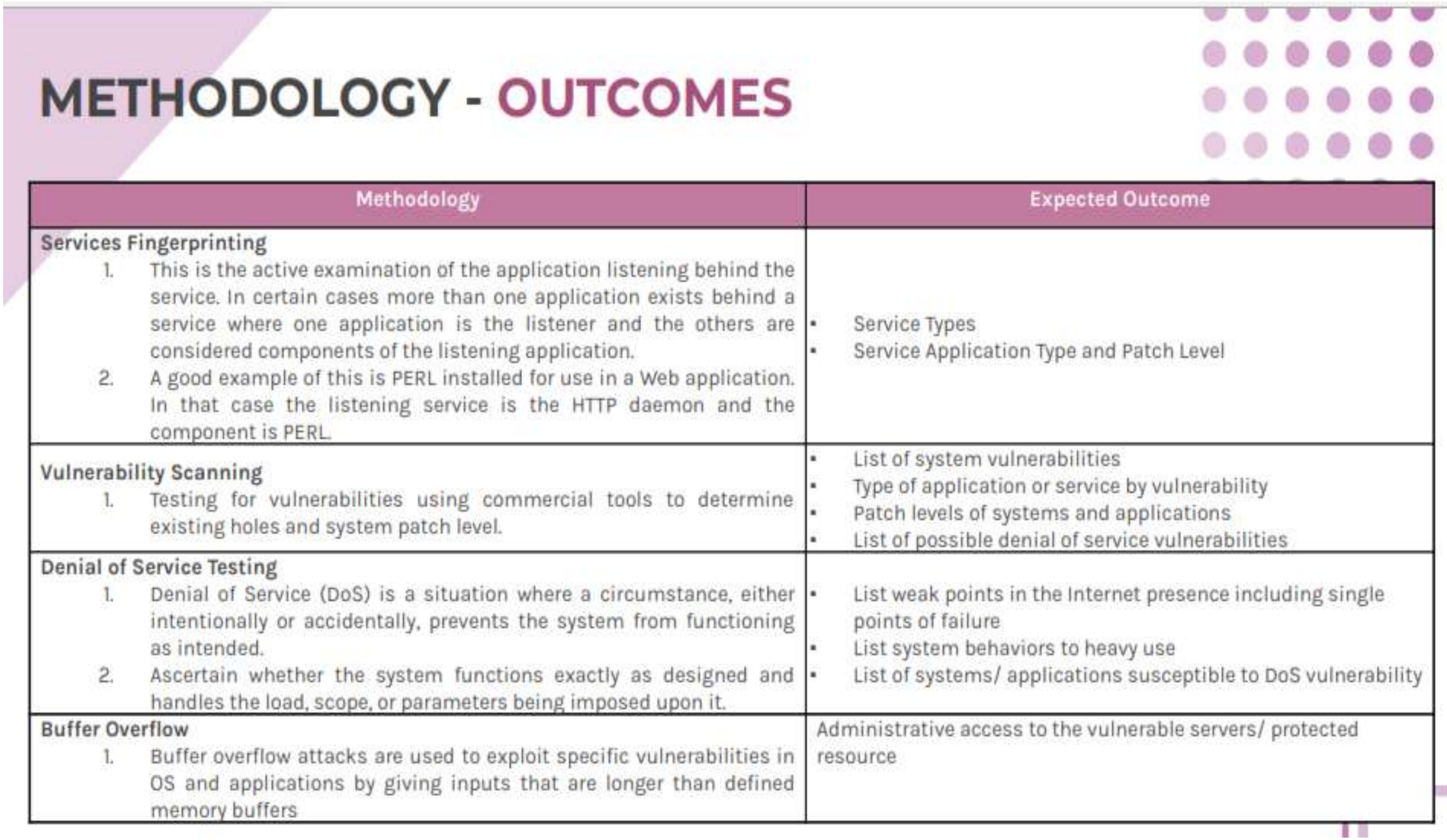
METHODOLOGY - OUTCOMES



METHODOLOGY - OUTCOMES

Methodology	Expected Outcome
Information Gathering 1. The point of this exercise is to find the number of reachable systems to be tested without exceeding the legal limits. 2. In this parameter, no intrusion is being performed directly on the systems. 3. The hosts discovered later may be inserted in the testing as a subset of the defined testing with concurrence of the Client's internal security team.	▪ Domain Names ▪ Server Names ▪ IP Addresses ▪ Network Map ▪ ISP / ASP information ▪ System and Service Owners ▪ Possible test limitations
Port Scanning 1. Port scanning is the invasive probing of system ports on the transport level. 2. This parameter is to enumerate live or accessible Internet services as well as penetrating the firewall to find additional live systems.	▪ Open, closed or filtered ports ▪ IP addresses of live systems ▪ List of discovered tunneled and encapsulated protocols ▪ List of discovered routing protocols supported ▪ Active services
Operating System Fingerprinting 1. System fingerprinting will be done for active probing of system for responses that can distinguish unique systems to operating system and version level.	▪ OS Type ▪ Other information such as uptime

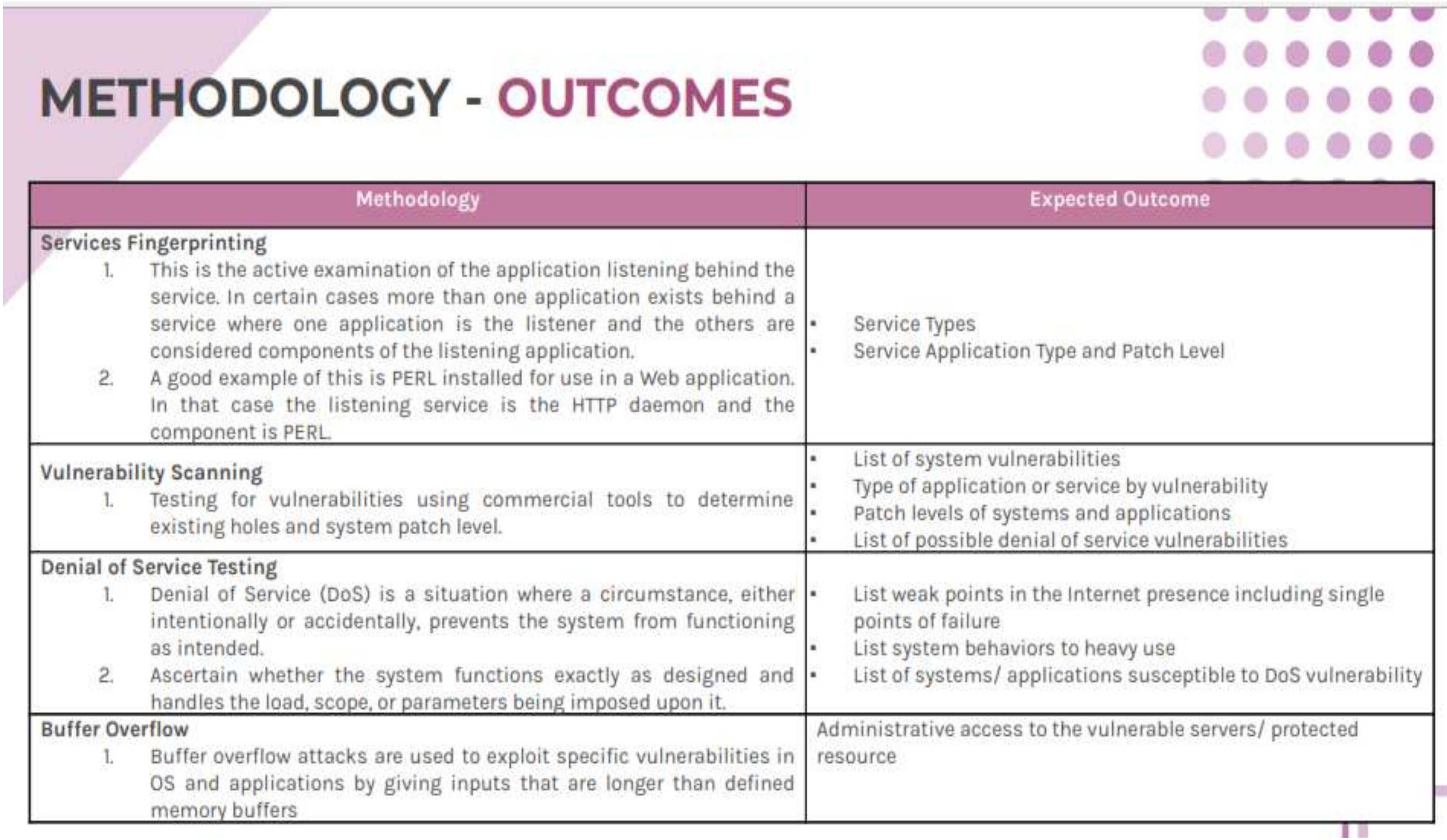
METHODOLOGY - OUTCOMES



METHODOLOGY - OUTCOMES

Methodology	Expected Outcome
Services Fingerprinting 1. This is the active examination of the application listening behind the service. In certain cases more than one application exists behind a service where one application is the listener and the others are considered components of the listening application. 2. A good example of this is PERL installed for use in a Web application. In that case the listening service is the HTTP daemon and the component is PERL.	▪ Service Types ▪ Service Application Type and Patch Level
Vulnerability Scanning 1. Testing for vulnerabilities using commercial tools to determine existing holes and system patch level.	▪ List of system vulnerabilities ▪ Type of application or service by vulnerability ▪ Patch levels of systems and applications ▪ List of possible denial of service vulnerabilities
Denial of Service Testing 1. Denial of Service (DoS) is a situation where a circumstance, either intentionally or accidentally, prevents the system from functioning as intended. 2. Ascertain whether the system functions exactly as designed and handles the load, scope, or parameters being imposed upon it.	▪ List weak points in the Internet presence including single points of failure ▪ List system behaviors to heavy use ▪ List of systems/ applications susceptible to DoS vulnerability
Buffer Overflow 1. Buffer overflow attacks are used to exploit specific vulnerabilities in OS and applications by giving inputs that are longer than defined memory buffers	Administrative access to the vulnerable servers/ protected resource

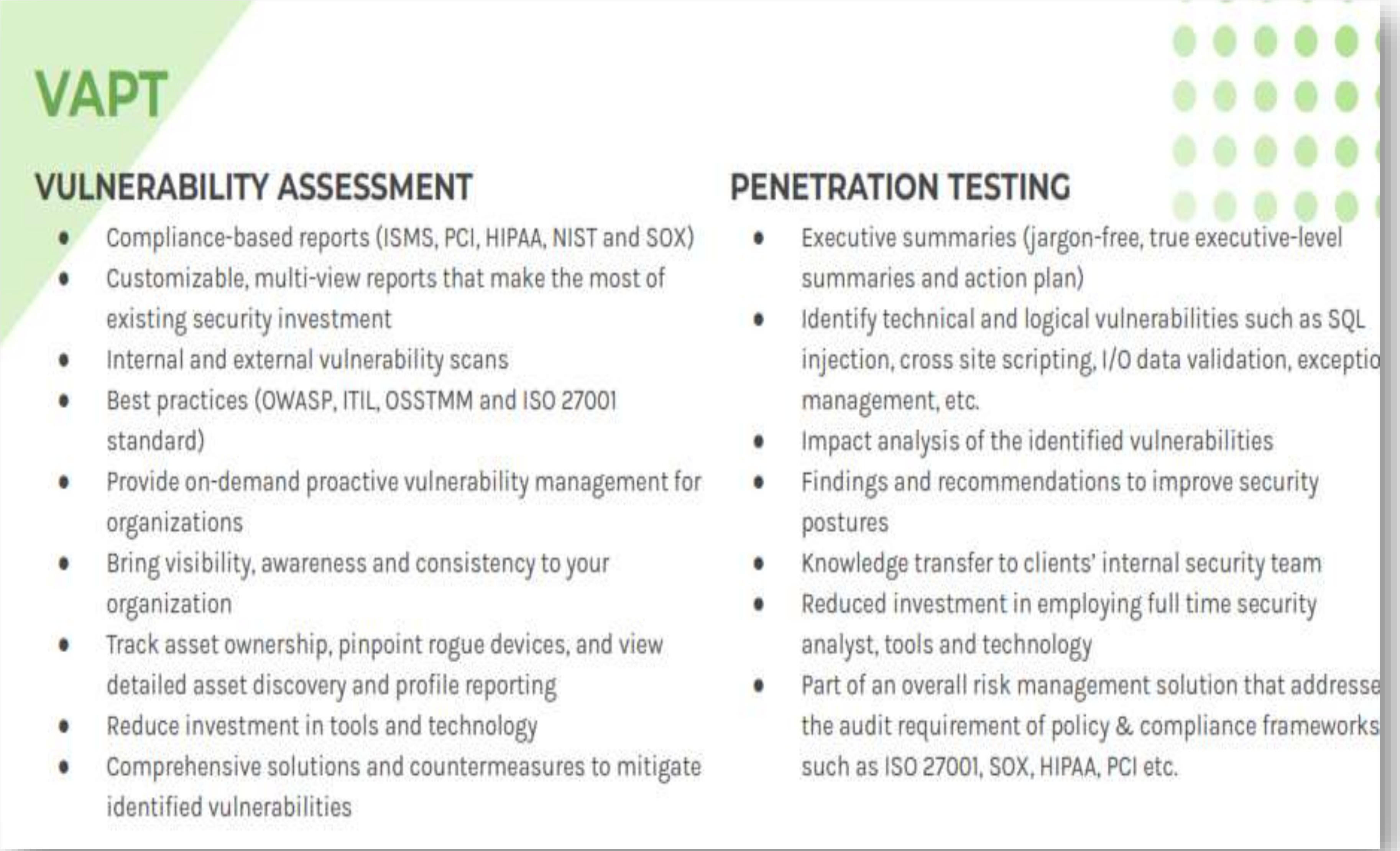
METHODOLOGY - OUTCOMES



METHODOLOGY - OUTCOMES

Methodology	Expected Outcome
Services Fingerprinting 1. This is the active examination of the application listening behind the service. In certain cases more than one application exists behind a service where one application is the listener and the others are considered components of the listening application. 2. A good example of this is PERL installed for use in a Web application. In that case the listening service is the HTTP daemon and the component is PERL.	▪ Service Types ▪ Service Application Type and Patch Level
Vulnerability Scanning 1. Testing for vulnerabilities using commercial tools to determine existing holes and system patch level.	▪ List of system vulnerabilities ▪ Type of application or service by vulnerability ▪ Patch levels of systems and applications ▪ List of possible denial of service vulnerabilities
Denial of Service Testing 1. Denial of Service (DoS) is a situation where a circumstance, either intentionally or accidentally, prevents the system from functioning as intended. 2. Ascertain whether the system functions exactly as designed and handles the load, scope, or parameters being imposed upon it.	▪ List weak points in the Internet presence including single points of failure ▪ List system behaviors to heavy use ▪ List of systems/ applications susceptible to DoS vulnerability
Buffer Overflow 1. Buffer overflow attacks are used to exploit specific vulnerabilities in OS and applications by giving inputs that are longer than defined memory buffers	Administrative access to the vulnerable servers/ protected resource

METHODOLOGY - OUTCOMES



VAPT

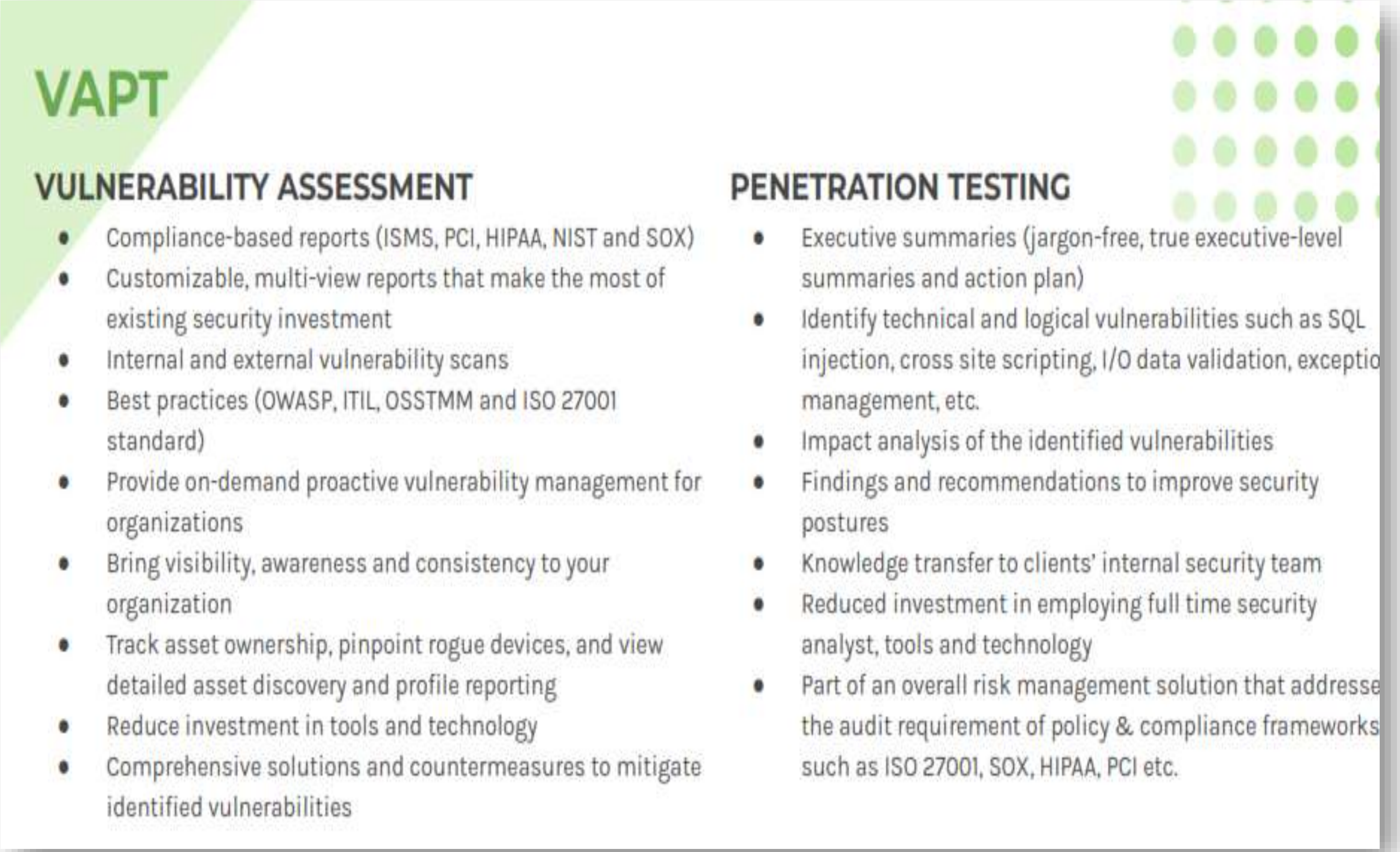
VULNERABILITY ASSESSMENT

- Compliance-based reports (ISMS, PCI, HIPAA, NIST and SOX)
- Customizable, multi-view reports that make the most of existing security investment
- Internal and external vulnerability scans
- Best practices (OWASP, ITIL, OSSTMM and ISO 27001 standard)
- Provide on-demand proactive vulnerability management for organizations
- Bring visibility, awareness and consistency to your organization
- Track asset ownership, pinpoint rogue devices, and view detailed asset discovery and profile reporting
- Reduce investment in tools and technology
- Comprehensive solutions and countermeasures to mitigate identified vulnerabilities

PENETRATION TESTING

- Executive summaries (jargon-free, true executive-level summaries and action plan)
- Identify technical and logical vulnerabilities such as SQL injection, cross site scripting, I/O data validation, exceptio management, etc.
- Impact analysis of the identified vulnerabilities
- Findings and recommendations to improve security postures
- Knowledge transfer to clients' internal security team
- Reduced investment in employing full time security analyst, tools and technology
- Part of an overall risk management solution that addresse the audit requirement of policy & compliance frameworks such as ISO 27001, SOX, HIPAA, PCI etc.

METHODOLOGY - OUTCOMES



VAPT

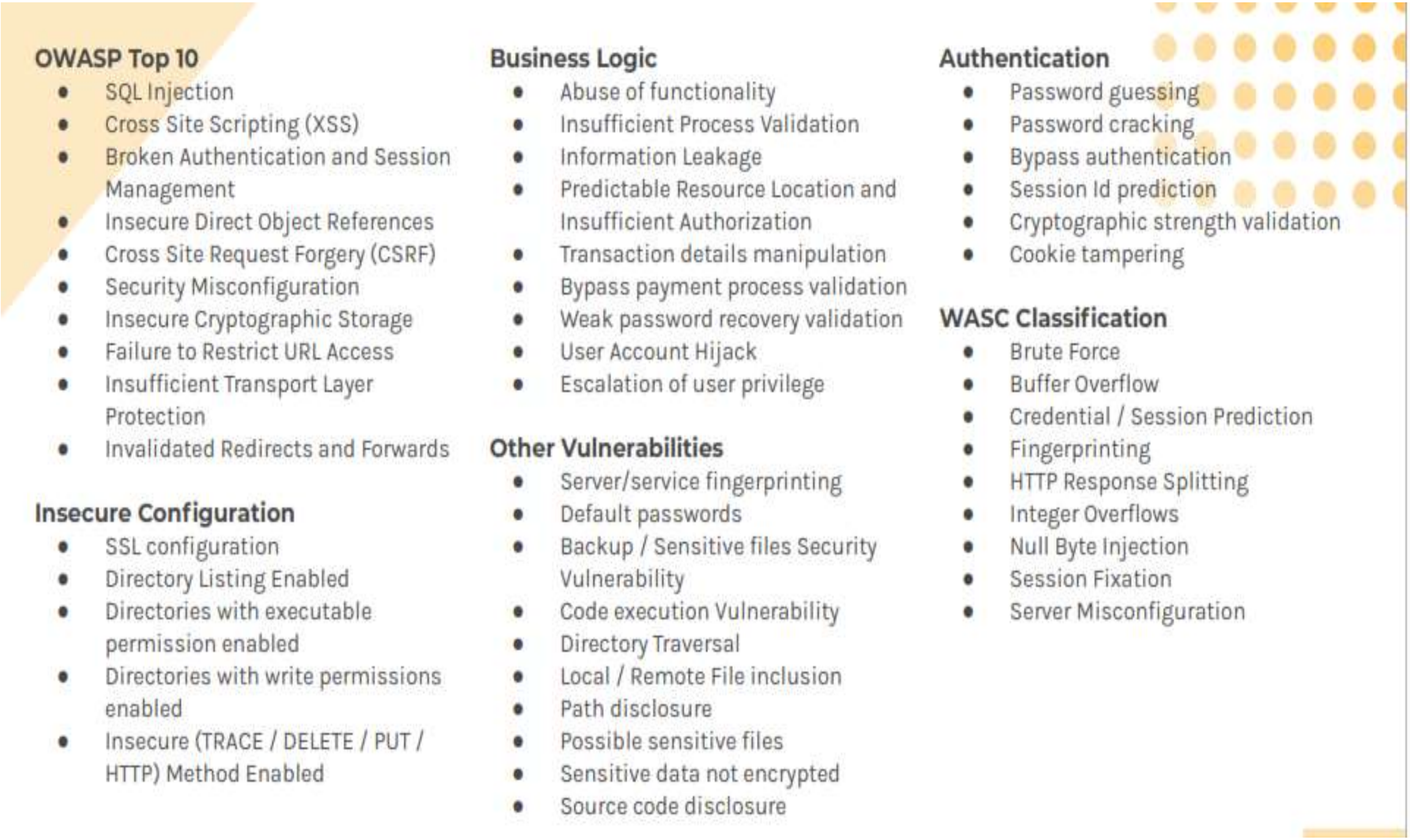
VULNERABILITY ASSESSMENT

- Compliance-based reports (ISMS, PCI, HIPAA, NIST and SOX)
- Customizable, multi-view reports that make the most of existing security investment
- Internal and external vulnerability scans
- Best practices (OWASP, ITIL, OSSTMM and ISO 27001 standard)
- Provide on-demand proactive vulnerability management for organizations
- Bring visibility, awareness and consistency to your organization
- Track asset ownership, pinpoint rogue devices, and view detailed asset discovery and profile reporting
- Reduce investment in tools and technology
- Comprehensive solutions and countermeasures to mitigate identified vulnerabilities

PENETRATION TESTING

- Executive summaries (jargon-free, true executive-level summaries and action plan)
- Identify technical and logical vulnerabilities such as SQL injection, cross site scripting, I/O data validation, exceptio management, etc.
- Impact analysis of the identified vulnerabilities
- Findings and recommendations to improve security postures
- Knowledge transfer to clients' internal security team
- Reduced investment in employing full time security analyst, tools and technology
- Part of an overall risk management solution that addresse the audit requirement of policy & compliance frameworks such as ISO 27001, SOX, HIPAA, PCI etc.

COMMON ISSUES



OWASP Top 10

- SQL Injection
- Cross Site Scripting (XSS)
- Broken Authentication and Session Management
- Insecure Direct Object References
- Cross Site Request Forgery (CSRF)
- Security Misconfiguration
- Insecure Cryptographic Storage
- Failure to Restrict URL Access
- Insufficient Transport Layer Protection
- Invalidated Redirects and Forwards

Insecure Configuration

- SSL configuration
- Directory Listing Enabled
- Directories with executable permission enabled
- Directories with write permissions enabled
- Insecure (TRACE / DELETE / PUT / HTTP) Method Enabled

Business Logic

- Abuse of functionality
- Insufficient Process Validation
- Information Leakage
- Predictable Resource Location and Insufficient Authorization
- Transaction details manipulation
- Bypass payment process validation
- Weak password recovery validation
- User Account Hijack
- Escalation of user privilege

Other Vulnerabilities

- Server/service fingerprinting
- Default passwords
- Backup / Sensitive files Security Vulnerability
- Code execution Vulnerability
- Directory Traversal
- Local / Remote File inclusion
- Path disclosure
- Possible sensitive files
- Sensitive data not encrypted
- Source code disclosure

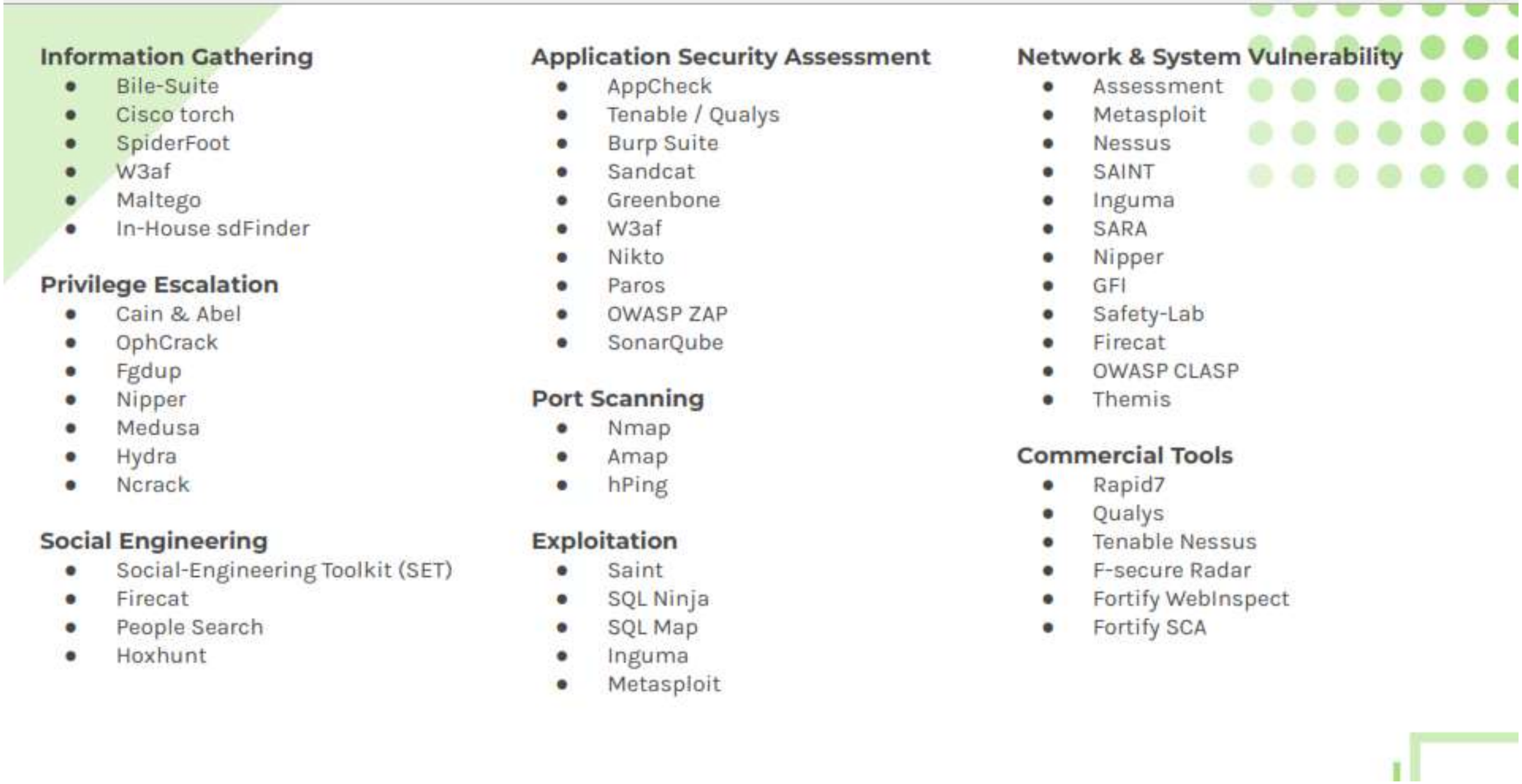
Authentication

- Password guessing
- Password cracking
- Bypass authentication
- Session Id prediction
- Cryptographic strength validation
- Cookie tampering

WASC Classification

- Brute Force
- Buffer Overflow
- Credential / Session Prediction
- Fingerprinting
- HTTP Response Splitting
- Integer Overflows
- Null Byte Injection
- Session Fixation
- Server Misconfiguration

METHODOLOGY - OUTCOMES



Information Gathering

- Bile-Suite
- Cisco torch
- SpiderFoot
- W3af
- Maltego
- In-House sdFinder

Privilege Escalation

- Cain & Abel
- OphCrack
- Fgdup
- Nipper
- Medusa
- Hydra
- Ncrack

Social Engineering

- Social-Engineering Toolkit (SET)
- Firecat
- People Search
- Hoxhunt

Application Security Assessment

- AppCheck
- Tenable / Qualys
- Burp Suite
- Sandcat
- Greenbone
- W3af
- Nikto
- Paros
- OWASP ZAP
- SonarQube

Port Scanning

- Nmap
- Amap
- hPing

Exploitation

- Saint
- SQL Ninja
- SQL Map
- Inguma
- Metasploit

Network & System Vulnerability

- Assessment
- Metasploit
- Nessus
- SAINT
- Inguma
- SARA
- Nipper
- GFI
- Safety-Lab
- Firecat
- OWASP CLASP
- Themis

Commercial Tools

- Rapid7
- Qualys
- Tenable Nessus
- F-secure Radar
- Fortify WebInspect
- Fortify SCA

False Positive and False Nagetive

- ✓ Burpsuite and Metasploit are commonly used tools for conducting PT.
- ✓ False positive and false negative are two important aspects of VAPT process.
- ✓ A false positive is when vulnerability actually does not exist, but it gets reported.
- ✓ A false negative is when vulnerability actually exists but it is not reported.

False Positive and False Nagetive

- ✓ Burpsuite and Metasploit are commonly used tools for conducting PT.
- ✓ False positive and false negative are two important aspects of VAPT process.
- ✓ A false positive is when vulnerability actually does not exist, but it gets reported.
- ✓ A false negative is when vulnerability actually exists but it is not reported.

Known vulnerabilities

- ✓ Known vulnerabilities are those that have been publicly reported and have a unique identifier, such as a CVE (Common Vulnerabilities and Exposures) number.
- ✓ These vulnerabilities are usually easier to detect and patch, as there are databases and sources that provide information and solutions for them.
- ✓ To scan for known vulnerabilities, you can use tools that compare your system or network configuration with the latest vulnerability data, such as Nmap, Nessus, or OpenVAS.

Known vulnerabilities

- ✓ These tools can perform different types of scans, such as port scans, service scans, or vulnerability scans, depending on your needs and goals.

Unknown vulnerabilities

- ✓ Unknown vulnerabilities are those that have not been disclosed or discovered yet, and therefore have no identifier or reference.
- ✓ These vulnerabilities are harder to detect and exploit, as they require more analysis and creativity. To scan for unknown vulnerabilities, you can use tools that test your system or network for potential flaws or weaknesses, such as Metasploit, Burp Suite, or ZAP.

Unknown vulnerabilities

- ✓ These tools can perform different types of tests, such as fuzzing, injection, or brute force, depending on the target and the attack vector.

Mobile Phone Security

Dr. Digvijaysinh Rathod
Associate Professor
School of Cyber Security and Digital Forensics
National Forensic Sciences University

digvijay.rathod@nfsu.ac.in