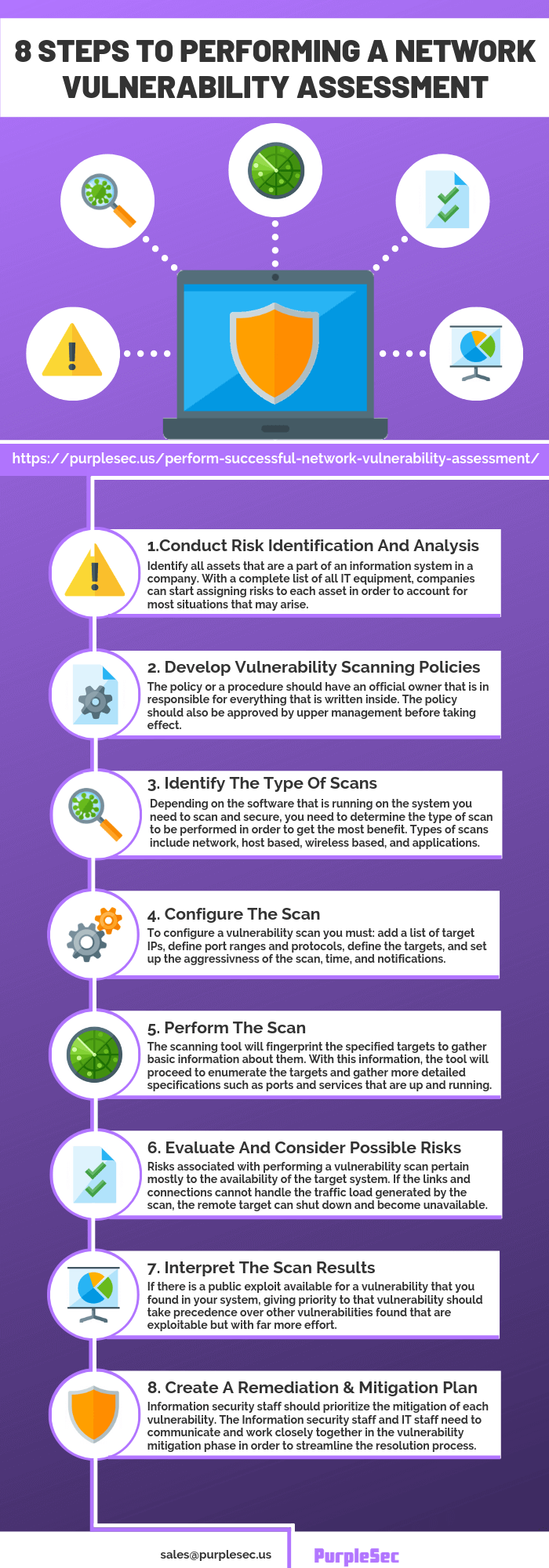
**Vulnerability Assessment and Methodologies:**

* **Vulnerability assessment:**

A **vulnerability** **assessment** is a systematic review of security weaknesses in an information system. It evaluates if the system is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation, if and whenever needed.

**Vulnerability** **assessment** is the process of **defining**, **identifying**, **classifying**, and **prioritizing** vulnerabilities in systems, applications, and networks. It provides an organization with the needed visibility into the risks that exist concerning external threats designed to take advantage of vulnerabilities

* **Methodology:**



**Penetration testing:**

* Penetration testing (also called pen testing) is the practice of testing a computer system, network or Web application to find vulnerabilities (identifying weakness & issues, patch, upgraded new version, misconfiguration etc.) that an attacker could exploit.

**Methodology:**



* **Reconnaissance:**

This phase has all the steps to gather evidence and information on the targets you want to attack.

Gather information regarding the target: IP address, domain, sub domain, network architecture, employees with in organization such as email, telephone etc. What tool include for attacking, how to attack. What info we are getting, useful or not.

* **Scanning:**

Take the information you gathered in recon and actively apply tools and techniques to gather more in depth information on targets.

Scanning involves taking the information discovered during reconnaissance and using it to examine the network and identify specific vulnerability.

* **Gaining Access:**

Gaining access refers to the point where the attacker obtains access to the operating system or applications on the computer or network. The attacker can gain access at the operating system level, application level, or network level.

The hacking attack can be delivered to the target system via a local area network (LAN), either wired or wireless; local access to a PC; the Internet; or offline.

* **Maintaining Access:**

Once a hacker has gained access to a target system, they want to keep that access for future exploitation and attacks.

Sometimes, hackers harden the system from other hackers or security personnel by securing their exclusive access with backdoors, rootkits, key loggers, and Trojans.

Once the hacker owns the system, they can use it as a base to launch additional attacks. In this case, the owned system is sometimes referred to as a zombie system.

* **Covering Tracks:**

Once hackers have been able to gain and maintain access, they cover their tracks to avoid detection by security personnel, to continue to use the owned system, to remove evidence of hacking, or to avoid legal action.

By executing the script, a variety of critical files are replaced with Trojan versions, hiding the attacker in seconds.

**Threat Modeling:**

Threat modeling is the practice of identifying and prioritizing potential threats and security mitigations to protect something of value, such as confidential data or intellectual property. When you are threat modeling, you bring the security architect, the operations/infrastructure team and lead developers together

Components for threat modeling **methodologies**:

* **STRIDE threat modeling**

STRIDE is a threat model, created by Microsoft engineers, which is meant to guide the discovery of threats in a system. It is used along with a model of the target system. This makes it most effective for evaluating individual systems.

* **Threat intelligence**

This area includes information about types of threats, affected systems, detection mechanisms, tools and processes used to exploit vulnerabilities, and motivations of attackers.

* **Asset identification**

Teams need a real-time inventory of components and data in use, where those assets are located and what security measures are in use. This inventory helps security teams track assets with known vulnerabilities.

* **Mitigation capabilities**

Mitigation capabilities generally refer to technology to protect, detect and respond to a certain type of threat, but can also refer to an organization’s security expertise and abilities, and their processes. Assessing your existing capabilities will help you determine whether you need to add additional resources to mitigate a threat.

* **Risk assessment**

Risk assessments correlate threat intelligence with asset inventories. These tools are necessary for teams to understand the current status of their systems and to develop a plan for addressing vulnerabilities.

* **Threat mapping**

Threat mapping is a process that follows the potential path of threats through your systems. It is used to model how attackers might move from resource to resource and helps teams anticipate where defenses can be more effectively layered or applied.

**OWASP Top 10 vulnerabilities**

1. **Injection:**

Web applications pass parameters when they access external systems or the local operating system.

If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the web application.

E.g.

* HTML Injection
* XSS
* SQL Injection

1. **Broke Authentication and Session Management:**

Account credentials and session tokens are not properly protected.

Attackers that can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities.

1. **Cross Site Scripting**

The web application can be used as a mechanism to transport an attack to an end user's browser.

A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user.

1. **Insecure direct object reference**

Insecure Direct Object References occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files.

1. **Security Misconfiguration**

Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box.

1. **Sensitive Data Exposure**

Sensitive Data Exposure occurs when an application does not adequately protect sensitive information.

The data can vary and anything from passwords, session tokens, credit card data to private health data and more can be exposed

1. **Missing Function Level Access Control**

Restrictions on what authenticated users are allowed to do are not properly enforced.

Attackers can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorized functions

1. **Cross site Request Forgery**

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.

1. **Using Components with known vulnerabilities**

Using libraries, frameworks, code in your application having vulnerabilities.   
There can be vulnerabilities in previous versions etc.

1. **Invalidated redirects and forwards**

Invalidated redirects and forwards are possible when a web application accepts untrusted input that could cause the web application to redirect the request to a URL contained within untrusted input. By modifying untrusted URL input to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials.