# Vulnerability Assessment Tools

Chapter 4

# Episode 4.01

## Web Application Scanners, Part 1

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.
• Web application scanner
- OWASP Zed Attack Proxy (ZAP)
- Burp suite

Objective 3.1 Given a scenario, analyze data as part of security monitoring activities.
• Log review
- Proxy

# Web Application Scanners

- HTTP/HTTPS protocols
- Ports 80 and 443
- OWASP Zed Attack Proxy (ZAP)
  - Proxy inserted between client and server
- Burp suite
  - Also a proxy
  - More manual
  - More granular control

# Episode 4.02

## Web Application Scanners, Part 2

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

- Web application scanner
  - Nikto
  - Arachni

# Web Application Scanners

- Nikto
  - Web server vulnerability scanner
- Arachni
  - Fast, Ruby-based Web application scanner

# Episode 4.03

Scanning

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

• Infrastructure vulnerability scanner

- Nessus

# Vulnerability Scanning Considerations

- Frequency
  - Industry regulations help determine frequency
  - Regulations vary by industry
- Resources
  - People
  - Computation
  - Time

# Vulnerability Scanning Considerations

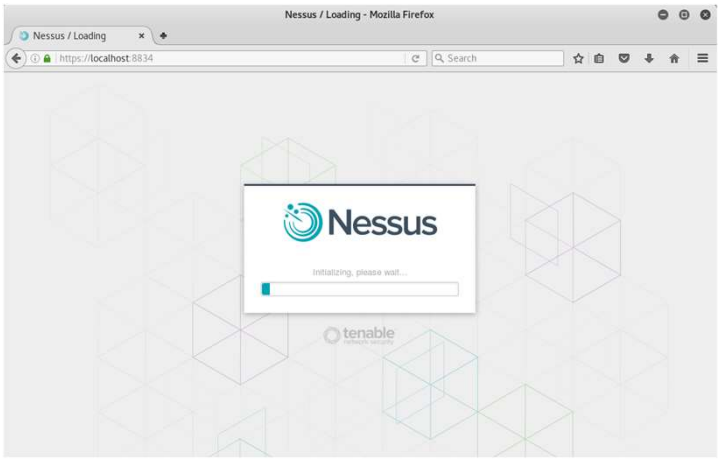
- Constraints
  - Qualified personnel
  - Technical capacity
- Determining frequency
  - Effects on workflow
  - Resources used to execute scans, review results, translate for leadership, and assist in decision making

# Vulnerability Scanning Criteria

- Feeds
- Sensitivity
- Scope
- Privilege
  - Credentialed
  - Non-credentialed
- Server-based
  - Scans executed from central point of contact
- Agent-based
  - Scanning software on all network nodes

# Vulnerability Scanning Criteria

- NASL
- SCAP
- Compliance checks
- Report generation



# SCAP

- Security Content Automation Protocol (SCAP)
  - Created by NIST
  - Standardizes assessment & reporting of vulnerabilities

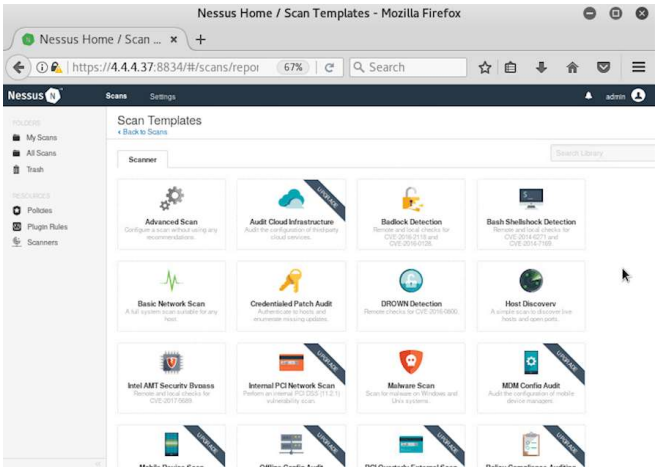
# Episode 4.04

## Configuring and Executing Scans

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

- Infrastructure vulnerability scanner
  - Nessus

# Nessus Demo


Nessus Home / Scan Templates - Mozilla Firefox
Nessus Home / Scan ...
https://4.4.4.37:8834/#/scans/repo
Search
Nessus
Scans
Settings
admin
Scan Templates
« Back to Scans
My Scans
All Scans
Trash
Policies
Plugin Rules
Scanners
Scanner
Advanced Scan
Audit Cloud Infrastructure
Badlock Detection
Bash Shellshock Detection
Basic Network Scan
Credentialed Patch Audit
DROWN Detection
Host Discovery
Intel AMT Security Bypass
Internal PCI Network Scan
Malware Scan
MDM Config Audit

# Episode 4.07

## Vulnerability Scanning

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

• Infrastructure vulnerability scanner

- Nessus
- OpenVAS
- Qualys

****NOTE: Episodes 4.05 and 4.06 were initially included in the series but subsequently removed after reviewing the objectives.*

# Vulnerability Scanning

- Qualys
- Nexpose
- Nessus
- OpenVAS
- Microsoft Baseline Security Analyzer
- Nikto

# Episode 4.08

Reverse Engineering

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

- Software assessment tools and techniques
  - Static analysis
  - Dynamic analysis
  - Reverse engineering

# Hardware

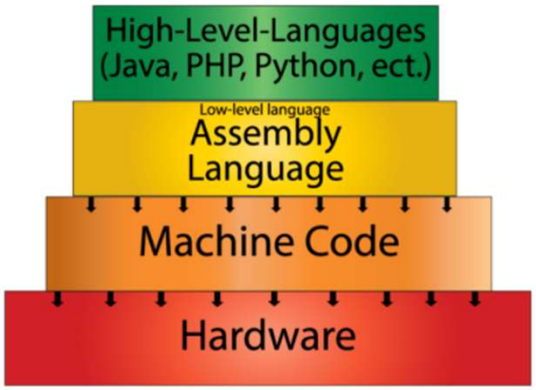
- Lower cost of manufacturing processes making counterfeiting a growing concern
- High-end devices have been counterfeited
- Trusted foundries have been created
  - Assure the integrity of products
  - Supply chain integrity
  - Analysis of the hardware components

# Software

- Detecting suspicious files
  - Name alone won't reveal if it's bad
  - Hashing function
    - Lists of known bad files on the Internet
    - Comparing hashes can uncover these bad files

# Software Creation

- Programming language
  - Represent abstract concepts
    - Selection, sequence, iteration



# Reverse Engineering Software

- Static analysis
  - Read through binary to figure out program structure
- Dynamic analysis
  - Connect binary to debugger to simulate execution
  - Probe program to find out what it's trying to do, resources its calling, output
- Static + dynamic analysis

# Disassembly

- The strings utility displays printable characters
  - Strings demo
- Visual representation of code
  - Binary ninja demo

# Episode 4.09

Enumeration

# Enumeration

- Active scanners
  - Nmap
    - Very popular tool available for most operating systems
    - Network mapper
  - hping
    - Supercharged ping utility
    - Craft custom packets
    - Analyze TCP, UDP, ICMP traffic
- Passive scanners
  - Responder
    - Remote access attack tool that poisons name services

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.
• Enumeration
- Nmap
- hping
- Active vs. passive
- Responder

# Episode 4.10

## Wireless Assessment Tools

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

- Wireless assessment tools
  - Aircrack-ng
  - Reaver
  - oclHashcat

# Wireless Assessment Tools

- Aircrack-ng
  - Open-source wireless security suite
  - Useful to audit WLAN security
- Reaver
  - Utility that exploits WPS weaknesses

# Wireless Assessment Tools

- oclHashcat
  - Password cracker that uses GPU and CPU power

# Episode 4.11

Cloud Assessment Tools

Objective 1.4 Given a scenario, analyze the output from common vulnerability assessment tools.

• Cloud infrastructure assessment tools

- ScoutSuite
- Prowler
- Pacu

# Cloud Infrastructure Assessment Tools

- ScoutSuite
  - Open-source
  - Uses Python
  - Supports AWS, MS Azure, Google Cloud, Alibaba, Oracle Cloud Infrastructure (OCI)
  - Auditing tool for managed services
- Prowler
  - Open-source
  - Similar to ScoutSuite, but mainly for AWS
  - Based on Center for Internet Security (CIS) best practices
- Pacu
  - AWS exploitation framework