Ripple Software Penetration Testing Framework

Web-Application SOC-2 Compliance

Date 2022-03-18

Prepared By Joseph Lee

Ripple Software Consulting LLC

Email joseph@ripplesoftware.ca

Mobile 778-725-3206

[Table Of Contents

[1. Overview Of The Pentest Process

Briefly, the penetration testing process begins with consultation to define a scope of engagement; set a list of targets that are allowed to be attacked, and also to set limitations known as *'rules of engagement'* such as when to communicate and report any findings and results. The start of the process is also a good time to set goals and priorities for the pentest, which may be determined by any regulation guidelines, standards, or compliance targets. The scope and goals of the pentest will impact the later stages such as which information is gathered, which vulnerability scans are conducted, how extensively exploits are pursued, and finally how vulnerability mitigation and retesting is done.

[1.1 General Steps in the Pentest Process

1. Determine the scope and set goals
2. Recon and information gathering
3. Testing for implementation of best-practices
4. Vulnerability scanning
5. Analysis and choosing initial target(s)
6. Exploit and pivot
    6.1. Persistent access
    6.2. Conceal evidence
7. Report Findings

[2. Information Gathering

After the scope and goals of the campaign have been set, **information gathering** can begin. The scope and goals will determine which information is valuable. Also, in whitebox or greybox tests, some information may be provided. In the case that information is provided by the target organization, it should still be verified to ensure that it is accurate and not assumed that it is correct. An assumption that the provided information is correct could cause a failure to identify vulnerabilities.

The information gathering stage consists of Open Source Intelligence (OSINT) that can be collected from publicly available sources, as well as both passive and active intelligence gathering.

[2.1 Types of Information Gathering

- Passive intelligence gathering
    - Does not engage the target IT infrastructure, people or physical locations
- Active intelligence gathering
    - Engages the target IT infrastructure, people or physical locations
- Foot-printing / mapping
    - A listing of all IT infrastructure, networks, physical locations, etc. that an organization has
    - Provided for white-box test
    - Not provided for black-box test
    - Standards for foot-printing
- OSINT – Open Source Intelligence

    - Enumeration
    - Code analysis

[2.2 OSINT Data Sources

- Domain registrars
- Web-searches including specialized search-engines
- Security centric databases / websites (Shodan, Censys)
- OSINT collection tools (Maltego, Recon-ng)
- Social media
- Corporate tax filings
- Corporate employees, email addresses, phone numbers
- Corporate literature
- Any other publicly available information

[2.3 Passive Information Targets

- Breached email accounts
- LinkedIn, FaceBook, GlassDoor other social media accounts
- Staff contact list and email addresses
- Physical locations
- The Internet Archive / Wayback Machine

[2.4 Active Intelligence Targets

- Web-server Data
    - IP address, domains, MX Records, DNS records
    - WAF identification
    - Scanning for TCP headers / cookies
    - TLS certificate information
    - Nmap scanning for open ports and services, OS Detection, etc.
    - Whois for server Information
    - Banner grabbing all service names and versions from open ports
    - Vulnerability scanning server
    - Testing the web-application for implementation of security best-practices
    - Scan IP for other domains hosted on same server
    - Site-map of website
    - Vulnerability scanning website
    - Search for confidential documents on website or social media accounts
    - Other active sub-domains
- Email Address Enumeration
    - Can be used for phishing campaign
    - To find email addresses associated with organization
        - Check for mail server response to service email addresses (info@, sales@)
        - Look for organization’s email pattern policy (first.last@, f.last@)
    - theHarvester
        - -d domain.com
    - Metasploit
        - > use /auxiliary/gather/search_email_collector ▪ > set domain domain.com
        - > set outfile filename.txt
        - > exploit
    - Use Internet search to check for account breach data ◦ h8mail

        - https://github.com/khast3x/h8mail
    - https://whatismyipaddress.com/breach-check
    - https://haveibeenpwned.com/
    - https://www.avast.com/hackcheck/friends-check
- API and Interface Enumeration
    - Discover patterns in API traffic and extract tokens, passwords, etc.
    - Replay extracted authentication tokens
    - Discover unauthorized data accessible in an API
- Certificate Enumeration and Inspection
    - May provide a list of other subdomains or domains located on the server
    - Analysis of certificate information may provide vulnerability information such as support for vulnerable versions of SSL/TLS or depreciated cipher suites

[3. Vulnerability Scanning

Once all information has been gathered and categorized, it is analyzed and stitched together to paint a picture of the target attack surface, and vulnerability scanning takes place against any identified assets, and their software, services, and web-applications.

Blackbox scanning is typically done in the form of an external scan on a target IP address or a web-application scan against a website hosted on a target domain. Another form of scanning for vulnerabilities is to fuzz or monkey fuzz an application to try and gather more information. However, in more extensive penetration tests, a vulnerability scan may include using source code scanning tools in an attempt to identify vulnerabilities in open-source or proprietary software applications. Finally, in the most extensive forms of penetration testing campaigns, software packages may be reverse compiled, and scanned.

There are numerous vulnerability scanning tools available and a fairly comprehensive list is included in **Appendix 6.1 List of Vulnerability Scanning Tools**.

The main goal of the vulnerability scanning stage is to identify any CVE or other attack surfaces that can later be exploited.

[3.1 External Vulnerability Scanning with OpenVAS

The Greenbone OpenVAS scanner is used to do an external scan of all endpoints including public facing IP addresses, and for tests that include internal scans, can be used to perform credentialed internal scans of all endpoints on a LAN or WAN. The results of the scan are analyzed and discovered vulnerabilities are categorized by endpoint, severity and relative risk.

These vulnerabilities are then included in the main report and correlated with other gathered information.

[3.2 Web-Application Vulnerability Scanning with OWASP ZAP

The OWASP Zed Attack Proxy (ZAP) is one of the world's most popular free security tools and is actively maintained by a dedicated international team of developers. Target applications are scanned with OWASP ZAP to identify vulnerabilities across several areas. The results of a scan are analyzed and verified using other tools, but also importantly, the lack of identification of a

vulnerability from OWASP ZAP does not rule out the potential of that vulnerability from existing. Therefore testing attempts to rule out false positives, and false negatives.

[4. Vulnerability Databases And Exploit Code

[4.1 Correlate CPEs to CVEs

The information gathered in the previous steps should result in some Common Platform Enumeration (CPE) ids. These CPE can first be correlated with CVE ids to identify and collect vulnerability information related to the specific software applications and services identified on the web-server and in the web-application itself.

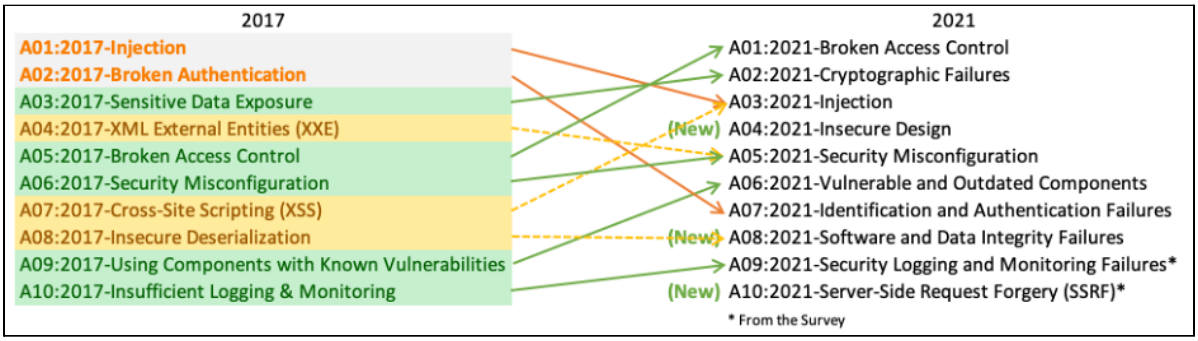
Once CVEs have been identified as being possibly exploitable, that CVE information must be translated into an active exploitation attempt. This may involve configuring an exploit toolkit such as Burp Suite, MetaSploit, or OWASP ZAP, or writing custom exploit code using specially crafted packets, or communication processes.

If this exploitation process is able to result in higher level of access, or pivoting to another resource, the process can begin again from that point in the campaign.

[4.2 Exploitation Without CVE

[4.2.1 OWASP Top Ten

OWASP Top Ten are widely considered the most important types of vulnerabilities that affect web-applications. The list is updated every few years, and the comparison between the 2017 list and 2021 list is shown below.



A01:2021-Broken Access Control	
A02:2021-Cryptographic Failures	
A03:2021-Injection	
A04:2021-Insecure Design	

A05:2021-Security Misconfiguration	
A06:2021-Vulnerable and Outdated Components	
A07:2021-Identification and Authentication Failures	
A08:2021-Software and Data Integrity Failures	
A09:2021-Security Logging and Monitoring Failures	
A10:2021-Server-Side Request Forgery	

Table 1: OWASP Top Ten and descriptions

[4.2.2 Identify Violations of Industry Standards And Best Practices

Industry standards and best-practices offer a baseline level of security and can be used to identify obvious security vulnerabilities in the way that a target server has been configured or a target application has been designed.

The vulnerability scan and penetration test includes investigation of the target assets for compliance with IT security best-practices. These standards include NIST CyberSecurity Framework, ISO 27001 and 27002, CISA, and others.

[4.2.3 MITRE ATT&CK Tactics, Technique, and Procedures

[4.3 Considerations for Using Exploit Code

- Can be dangerous because could contain other unexpected malware
- Downloading exploit code can set of virus scanners on host
- Should verify that checksum matches if available

[4.4 Vulnerability and Exploit Databases

- Exploit Database
  - https://www.exploit-db.com/
  - Specific details about exploits
  - Shellcode sometimes available for proof-of-concept
  - Security research papers
  - Google Hacking Database (Google Dorks)
- Rapid 7 Vulnerability and Exploit Database
  - https://www.rapid7.com/db/
  - Plugins for Metasploit Framework and Metasploit Pro
- NVD – National Vulnerability Database
  - https://nvd.nist.gov/
  - Does not provided exploit code
  - References may mention if exploits are available and their names
- VULDB
  - https://vuldb.com/
  - Crowd-sourced vulnerability database

    - Includes estimated prices and rankings for exploit code
    - Can help understand market focus and lead the scoping process of a penetration test
- MITRE CVE
    - https://cve.mitre.org/cve/
    - A list of publicly disclosed cybersecurity vulnerabilities that is free to search, and use
- incorporate into products and services
- CVE Details
    - https://www.cvedetails.com/
    - https://www.itsecdb.com/oval/
    - Provides a web interface to all IT security related items including patches, vulnerabilities and compliance checklists
    - Collects OVAL (Open Vulnerability and Assessment Language) definitions from many sources
- BugTraq ID (BID)
    - https://www.securityfocus.com/bid/
    - CVE to BugTraq ID concordance
        - https://cve.mitre.org/data/refs/refmap/source-BID.html

[5. Exploitation

The information gathered from all previous sections is used to create an exploitation gameplan. For a typical web-application this gameplan will typically include bruteforce testing of login and registration forms, manipulation of other form and API data, cookies, and SQL injection, and finally, it will include research about and attempts to exploit any vulnerable 3rd party libraries and applications found in the vulnerability scans.

[5.1 Limitations Of Engagement Exploitation

The exploitation phase is limited in scope depending on the "rules of engagement" (ROE) outlined between RSRC and the client. This scope may be documented in a service level agreement (SLA). These ROE and associated SLA may include terms that stipulate expected behavior on the part of the penetration tester. For example, the ROA and SLE may include:

- A scope of domains or IP range that should be included or ignored from testing
- A criticality level to immediately stop penetration testing and report findings
- A primary goal or several primary goals that the penetration test seeks to achieve
- A level of exploitation that should not be crossed in terms of actually exploiting systems
- Times of day that the penetration tests should be done

In most cases the exploitation phase is limited to exploitation of known vulnerabilities and exploiting vulnerabilities for which existing exploit code or tools exist. In more advanced penetration testing engagements, source code for software applications may be statically analyzed in order to find exploitable vulnerabilities.

[6. Reporting

After the information gathering and exploitation phases, a full report is generated, which includes both a description of how the vulnerability was found, a description of the vulnerability, and general information regarding the mitigation of each identified vulnerability and exploited attack vector. The discovered vulnerabilities and exploits are ordered by severity and included near the top of the report, and relevant files are referenced and included in the report.

Efforts are made to protect both the information used in the information gathering process and the information in the generated report at all stages of the engagement process.

[7. Appendix

[7.1 List of Vulnerability Scanning Tools

[7.1.1 Web-Application Scanners

- **W3af**
  - https://w3af.org/
  - Works on Mac, Linux, Windows
  - Extremely popular, powerful, and flexible framework for finding and exploiting web-application vulnerabilities
  - Extensible and features dozens of web assessment and exploitation plugins
  - Referred to as a web-based Metasploit
  - Fuzzing and brute-force login credential tools
- **WebScarab**
  - https://github.com/OWASP/OWASP-WebScarab
  - Works with Mac, Linux, Windows
  - Records the requests and responses between browser and web-servers
  - Allows the operator to review them in various ways
  - Exposes the workings of an HTTP(S) based application
  - Can assist debugging otherwise difficult problems
  - Allows security specialists to identify vulnerabilities in the web-application design or implementation
- **Arachni**
  - http://www.arachni-scanner.com/
  - FOSS for Mac, Linux, Windows
  - Ruby framework for evaluating the security of web-applications
  - Vulnerability scans JavaScript / JQuery / AngularJS, HTML5, DOM manipulation and AJAX
  - Command line and WebUI
  - Vulnerability Scans for:
    - DOM-based vulnerabilities XSS (with DOM variants) SQL injection
    - NoSQL injection
    - Code injection
    - File inclusion variants More
- **Subgraph Vega**
  - https://subgraph.com/vega/index.en.html
  - Written in Java
  - Free and open-source works with Mac, Linux, Windows

    - SQL Injection, Cross-Site Scripting (XSS), SSL/TLS security scans, inadvertently disclosed sensitive information, remote code execution
    - GUI based
- **OWASP Samurai Web-Testing Framework**
    - https://www.samurai-wtf.org/
    - Web-application testing framework
    - Live Linux VM image / environment that has been pre-configured to function as a web pen-testing environment
    - Contains good open-source and free tools that focus on testing and attacking websites
- **Sqlninja**
    - https://tools.kali.org/vulnerability-analysis/sqlninja
    - Mac, Linux, Windows
    - Exploits web-applications that use Microsoft SQL Server as a database backend
    - Attempts to attain a running shell on the remote host
    - Doesn't find an SQL injection, but automates the exploitation after one is discovered
- **Wappalyzer**
    - https://www.wappalyzer.com/
    - Desktop and web-application GUI available
    - Scan a website for services and versions, JS scripts, OS detection, and 3rd party software such as Google Analytics, Database version, source code language, CMS, web-server, etc.
    - Extracts social media accounts, email addresses, phone numbers, location information
- **WebSurgery**
    - http://sunrisetech.gr/?page=websurgery&tab=overview
    - Windows Only
    - Suite of tools for security testing of web applications
    - Tools include crawler, bruteforcer, fuzzer, proxy, editor
- **FireFox Development**
    - https://getfirebug.com/
    - Mac, Linux, Windows
    - Firefox Development provides access to browser internals
    - Editing of HTML and CSS, a DOM viewer, and JavaScript debugger
- **SQLMap**
    - https://sqlmap.org/
    - Automates the process of detecting and exploiting SQL injection flaws and taking over of back-end database servers
    - Database fingerprinting
    - Extracting data from the database
    - Accessing the underlying file system and executing OS commands via out-of-band connections

[7.1.2 Web proxies

- **OWASP ZAP**
    - https://owasp.org/www-project-zap/
    - Source Code: https://github.com/zaproxy/
    - Free Open-source web-application scanner Automation, scheduling, paid add-ons
- **Burp Suite**
    - https://portswigger.net/burp

    - Professional version is paid software license
    - Community edition is installed built-into in Kali Linux Automated scanning across their entire portfolios
    - Schedules scanning
- **Paros Proxy**
    - https://resources.infosecinstitute.com/topic/introduction-paros-proxy-lightweight-web-application-tool/
    - No updates in long time, works on Mac, Linux, Windows
    - Written in Java
    - Web proxy for assessing web-application vulnerability
    - Supports editing/viewing HTTP/HTTPS messages on-the-fly to change items such as cookies and form fields
    - Web traffic recorder, web spider, hash calculator
    - Scanner for testing common web-application attacks such as SQL injection and cross-site scripting
- **Tamper Data**
    - https://sectools.org/tool/tamperdata/
    - Add-on for Firefox, Chrome
    - Allows viewing and modification of HTTP requests
    - Shows data included in communication with web-server such as such as cookies and hidden form fields
    - Identifies web applications that trust the client data input

[7.1.3 Network Scanners

- **Nikto / Nikto 2**
    - https://cirt.net/Nikto2
    - Works on Mac, Linux, Windows with many plugins available
    - Open-source scanner which performs comprehensive tests against web servers for multiple items
        - Lists over 6400 potentially dangerous files/CGIs Checks for outdated versions of over 1200 servers Version specific problems on over 270 servers
    - Checks for server configuration
        - Presence of multiple index files
        - HTTP server options
        - Attempt to identify installed web servers and software
- **OpenVAS**
    - https://www.openvas.org/
    - Open-source for Mac, Linux, Windows
    - Vulnerability scanner that was forked from Nessus Plugins are still written in the Nessus NASL language
- **NetSparker**
    - https://www.netsparker.com/
    - Paid software for Mac, Linux, Windows
    - Support for both detection and exploitation of vulnerabilities
    - Aims to be false positive–free by only reporting confirmed vulnerabilities
- **QualysGuard**
    - https://www.qualys.com/qualysguard/
    - Popular SaaS vulnerability management software
    - Web-based UI offers network discovery and mapping
    - Asset prioritization, vulnerability assessment, and reporting
    - Remediation tracking according to business risk of asset value / priority

    - Scans handled by Qualys appliances that communicate to cloud-based system
- **MBSA – Microsoft Baseline Security Analyzer**
    - https://www.microsoft.com/security/blog/2012/10/22/microsoft-free-security-tools-microsoft-baseline-security-analyzer/
    - Designed to help small and medium-sized businesses assess security state
    - Accordance with Microsoft security recommendations
    - Built on the Windows Update Agent and Microsoft Update infrastructure
    - MBSA ensures consistency with
        - Microsoft Update (MU)
        - Windows Server Update Services (WSUS) Systems Management Server (SMS)
        - Microsoft Operations Manager (MOM)
- **Nessus**
    - https://www.tenable.com/products/nessus
    - One of the most popular and capable vulnerability scanners, particularly for UNIX systems
    - Costs $2,190 per year
    - Free Nessus Home version is also available
    - Constantly updated, with more than 70,000 plugins
    - Remote and local (authenticated) security checks
    - Client/server architecture with a web-based interface
    - Embedded scripting language for writing your own plugins or understanding the existing ones
- **Nexpose**
    - https://www.rapid7.com/products/nexpose/
    - Paid software as stand-alone, Metasploit plugin
    - Discovery, detection, verification, risk classification, impact analysis, reporting and mitigation
    - Integrates with Metasploit to give you a comprehensive vulnerability sweep
- **SolarWinds**
    - https://www.solarwinds.com
    - Paid software – costs about $1500
    - Automated network discovery
    - Real-time monitoring and alerting Powerful diagnostic capabilities
- **Nmap**
    - https://nmap.org/
    - Documentation: https://nmap.org/docs.html
- **THC Amap**
    - https://www.thc.org/
    - Network service mapping
    - Good 2nd opinion or if Nmap fails to detect a service
- **host (command)**
    - Manual: https://linux.die.net/man/1/host
    - Linux command line application simple utility for performing DNS lookups
    - Converts names to IP addresses and vice versa
    - Zone transfers, MX records, NS servers, TXT records, etc
- **Traceroute**
    - Manual: https://linux.die.net/man/8/traceroute
    - Map devices and appliances on the network that simply forward traffic
        - Switches, hubs, main back-bone infrastructure Particularly useful in the local-network
        - Switches may allow VLAN-hopping
        - Discovery of Firewall / IDS / IPS applicances

    - Sends IC**MP packets with incrementing TTL to discover devices on the route**
- **dig – Domain Information Groper**
    - https://linux.die.net/man/1/dig
    - Flexible tool for interrogating DNS name servers
    - Performs DNS lookups
    - Command-line arguments and batch mode of operation (-f)
    - dig [@server] [name] [type]
        - @server = IP addresses (IPv4 / IPv6 / hostname) name = name of resource record to be looked up type = type of query ANY, A, MX, SIG