Příloha RD08 – Zajištění bezpečnostních testů

č. sml. Objednatele: ČÚZK-16929/2019-24 č. sml. Zhotovitele: CZBAP-127

1 Úvod

Tento dokument stanovuje pravidla a postupy pro provádění bezpečnostních testů v prostředí Objednatele k zajištění bezpečnostního testování RÚIAN (dále též „bezpečnostní testování“ nebo „bezpečnostní testy“).

RÚIAN je komplexem tří systémů: ISÚI (informační systém územní identifikace), RÚIAN (registr územní identifikace, adres a nemovitostí) a VDP (veřejný dálkový přístup). Všechny tři systémy jsou v následujícím textu označovány zjednodušujícím termínem RÚIAN. Pokud je podstatné, aby byly rozlišeny systémy ISÚI, RÚIAN a VDP jednotlivě, je tato skutečnost v dokumentu výslovně uvedena. Interní testování Zhotovitele v oblasti bezpečnosti prováděné na technologické infrastruktuře Zhotovitele není obsahem tohoto dokumentu.

2 Členění zranitelností podle závažnosti

2.1 CRITICAL

Kritická, vyžaduje zpravidla okamžitý zásah nebo odstavení systému.

2.2 IMPORTANT

Důležitá, může být zdrojem budoucích potíží, je nezbytná náprava dle možností co nejdříve.

2.3 MEDIUM

Střední stupeň závažnosti, zvyšuje pravděpodobnost úspěšného útoku, zpravidla vyžaduje splnění určitých podmínek.

2.4 LOW

Nízký stupeň závažnosti, pouze mírně zvyšuje pravděpodobnost úspěšného útoku, vyžaduje splnění určitých podmínek.

2.5 INFORMATION

Informativní, nejedná se ve skutečnosti o zranitelnost, ale o informaci.

3 Pravidla a způsob provádění bezpečnostních testů

Bezpečnostní testování bude prováděno v testovacím prostředí Objednatele a v předem stanoveném a Objednatelem odsouhlaseném rozsahu.

Výjimky z rozsahu bezpečnostních testů jsou možné pouze po předchozím odsouhlasení Objednatele.

3.1 Kritéria pro stanovení rozsahu bezpečnostního testování

Bezpečnostní testování může být vyvoláno následujícími faktory:

3.1.1 Příprava dodávky RÚIAN

Zhotovitel při navrhování rozsahu bezpečnostního testu posuzuje:

- zda změna RÚIAN zasahuje přímo do bezpečnostních vlastností RÚIAN (změna je s přímým bezpečnostním dopadem, např. zavedení nové webové služby, změna technologie), nebo zda změna má nebo může mít nepřímý bezpečnostní dopad nebo zda může zasáhnout do

bezpečnostních opatření RÚIAN (např. doplněný nebo změněný modul bez přímé vazby na bezpečnostní opatření),

- rozsah změn RÚIAN.

Závazný minimální rozsah bezpečnostních testů, v závislosti na charakteru změny vyjádřeném číslem verze dodávky RÚIAN, je uveden v následující tabulce.

Změna RÚIAN označená jako	Jedná se o verzi změny (dodávky) označenou	Rozsah prováděných bezpečnostních testů Zhotovitelem
Velká	X.Y (např. 3.0, 3.1, ..)	Bude vždy provedena kompletní sada bezpečnostních testů dle kapitoly Bezpečnostní testy webových aplikací a služeb tohoto dokumentu.
Malá	X.Y.Z (např. 3.0.1, 3.1.2, …)	Bude provedena kompletní sada bezpečnostních testů pouze v případě, že bude implementována alespoň jedna změna RÚIAN s možným přímým nebo nepřímým bezpečnostním dopadem.
Patch/hotfix	X.Y.Z.xx (např. 3.0.1.03)	Zhotovitelem budou provedeny bezpečnostní testy vybraných a navržených testovacích scénářů pro příslušnou změnu s možným bezpečnostním dopadem, případně i další bezpečnostní testy navržené Objednatelem nad rámec návrhu Zhotovitele.
Změna bezpečnostního mechanismu		Budou provedeny bezpečnostní testy Zhotovitelem vybraných a navržených bezpečnostních testovacích scénářů pro příslušnou změnu na základě povahy této změny
Nová hrozba		Budou provedeny bezpečnostní testy vybraných a případně nově navržených bezpečnostních testovacích scénářů pro příslušnou hrozbu na základě povahy této hrozby. Návrh dá vždy Zhotovitel, Objednatel ale může navrhnout vlastní bezpečnostní scénář.

Bezpečnostní testy začleňuje Zhotovitel do harmonogramu dané dodávky RÚIAN (viz příloha 15 ZD).

Pokud není v období 12 měsíců plánována / dodána dodávka RÚIAN typu X.Y, začlení Zhotovitel provedení kompletní sady bezpečnostních testů do vhodné dodávky RÚIAN typu X.Y.Z tak, aby odstup od minulého provedení kompletní sady bezpečnostních testů nebyl větší než 12 měsíců, případně lze po dohodě se Objednatelem provést na v té době vhodném testovacím prostředí Objednatele kompletní sadu bezpečnostních testů bez vazby na konkrétní dodávku RÚIAN.

3.1.2 Zjištění výskytu relevantní zranitelnosti v průběhu kybernetického bezpečnostního incidentu

V takovém případě je bezpečnostní testování prováděno v rozsahu nezbytném pro ověření, zda kybernetický bezpečnostní incident nebyl způsoben zranitelností.

Provedení bezpečnostních testů navrhuje Zhotovitel na základě zjištěných informací; součástí návrhu je i vhodné začlenění do harmonogramů aktuálních/plánovaných dodávek RÚIAN.

3.1.3 Informace zjištěné při činnostech prováděných Manažerem nebo Architektem kybernetické bezpečnosti VIS nebo Specialistou kybernetické bezpečnosti

Zdrojem těchto informací může být například sledování informačního servisu NÚKIB nebo security bulletinů; v takovém případě je bezpečnostní testování prováděno, pokud obsahuje komponentu, která může být na zranitelnost náchylná; účelem tohoto bezpečnostního testu je zjištění, zda RÚIAN danou zranitelnost obsahuje.

Provedení bezpečnostní testů navrhuje Zhotovitel na základě zjištěných informací; součástí návrhu je i vhodné začlenění do harmonogramů aktuálních/plánovaných dodávek RÚIAN.

3.2 Pravidelné bezpečnostní testy na produkčním prostředí

Zhotovitel provádí na produkčním prostředí Objednatele pravidelně minimálně 1 x za 12 měsíců sadu základních bezpečnostních testů v rozsahu Přílohy č. 1.

3.2.1 Pravidla provádění bezpečnostních testů

Bezpečnostní testy musí být opakovatelné, aby se výsledky jednotlivých testů daly porovnat a vyhodnotit stav a vývoj bezpečnosti aplikace.

Testy musí být prováděny jak:

- neinvestigativním způsobem, tj. prověření správné implementace bezpečnostních mechanismů v aplikacích, shody s návrhem bezpečnostní architektury a shody vůči relevantním systémovým bezpečnostním politikám,
- tak i investigativním způsobem při realizaci změn webových aplikací, které mohou mít dopad na zajištění důvěrnosti, dostupnosti a integrity aktiv

Bezpečnostní testování provádí Zhotovitel, na základě Zhotovitelem předem zpracovaných testovacích scénářů.

3.3 Způsob provádění bezpečnostních testů

3.3.1 Interní testování Zhotovitele

Na základě schváleného harmonogramu implementace dané změny RÚIAN provede Zhotovitel interní testování v testovacím prostředí Zhotovitele s instalovanou změnou RÚIAN (tj. před schválením instalace změny do produkčního prostředí Objednatele). V případě zjištění bezpečnostní chyby nebo zranitelnosti již v testovacím prostředí Zhotovitele, Zhotovitel zajistí odstranění chyby a opakování bezpečnostních interních testů.

3.3.2 Testování v prostředí Objednatele

Na základě změny RÚIAN Zhotovitel předá garantovi aktiv RÚIAN a manažerovi kybernetické bezpečnosti resortu před zahájením celkového testování informaci o seznamu změn RÚIAN, včetně uvedení, jak danou změnu vyhodnotil, tj. zda tato změna má nebo nemá bezpečnostní dopad a jaký rozsah bezpečnostních testů bude dle tohoto dokumentu minimálně proveden, včetně harmonogramu celkového testování s vyznačením termínů provádění bezpečnostních testů, tj. dodání dokumentu „Plán bezpečnostního testování pro dodávku X“, který musí vždy obsahovat všechny změny RÚIAN, které dodávka zahrnuje, vliv změn na bezpečnost a uvedení, zda změny budou testovány z hlediska bezpečnosti.

Dokument „Plán bezpečnostního testování pro dodávku X“ podléhá schválení ředitele odboru informatiky ČÚZK.

Na základě schváleného dokumentu „Plán bezpečnostního testování pro dodávku X“ provede Zhotovitel testování po instalaci nové změny do provozního prostředí Objednatele, kde se provádějí bezpečnostní testy webových aplikací a webových služeb a penetrační testy uvedené v bodu 4 tohoto

dokumentu.

3.3.3 Pro účely testování poskytne ČÚZK Zhotoviteli:

- testovací účet s přístupem do RÚIAN a jejích webových aplikací a služeb s právy běžného interního a externího uživatele,
- vzdálený přístup do interní sítě Objednatele, nebo fyzický přístup na pracoviště ČÚZK, pokud to bude pro testování potřebné;
- možnost připojení koncového zařízení Zhotovitele (testovacího notebooku nebo serveru) do testovacího prostředí Objednatele.

3.3.4 Zhotovitel je při provádění bezpečnostních testů povinen:

- bezpečnostní testy provádět dle schváleného dokumentu „Plán bezpečnostního testování pro dodávku X“;
- neprovádět nevratné zásahy do systému (v případě úspěšného průniku);
- nepoužívat techniky „sociálního inženýrství“ (telefonáty nebo maily pod předstíranou identitou apod.);
- v případě zjištění závažné skutečnosti v průběhu testování (odstavení některé služby, zjištění závažné slabiny apod.) okamžitě informovat garanta aktiv RÚIAN.

Výsledky bezpečnostních testů zpracuje Zhotovitel do Zprávy o výsledcích bezpečnostních testů a předloží ji i s návrhy opatření a uvedením způsobu, jakým byly zjištěné kritické zranitelnosti vyřešeny Objednateli. V případě, že zpráva obsahuje zjištěné zranitelnosti, Zhotovitel zajistí svolání schůzky Zhotovitele s Objednatelem, kde prezentuje své zjištění a blíže informuje o návrhu/návrzích řešení. Na schůzce Objednatel rozhodne o způsobu odstranění zranitelností nebo jejich eliminaci a dalším postupu. Závěry ze schůzky Zhotovitel zaznamená do zápisu ze schůzky, který podepisuje zástupce Zhotovitele a garant aktiv RÚIAN.

Bez akceptace nemůže být změna instalována do provozního prostředí Objednatele. Nalezené kritické zranitelnosti v oblasti bezpečnosti přitom musí být vždy Zhotovitelem napraveny a opakovaně ověřeny bezpečnostním testem.

3.4 Ochrana dat v průběhu testování

Zhotovitel se v průběhu realizace bezpečnostních testů řídí standardními pravidly pro zajištění důvěrnosti používaných informací, zejména pak:

- tam, kde je to možné, používá anonymizované informace,
- v případech, kdy použití anonymizovaných informací není možné (např. v rámci testování v produkčním prostředí), je povinen zajistit opatření, která znemožní jejich nekontrolovaný únik.

4 Bezpečnostní testy webových aplikací a služeb

V rámci bezpečnostních testů jsou testována rozhraní RÚIAN, k nimž přistupují uživatelé, jak interní z vnitřní sítě resortu, tak externí, kteří mají přístup zajištěn pomocí dálkového přístupu z vnější sítě. Tedy zejména aplikace a služby VDP a ISÚI.

Bezpečnostní testy musí obsahovat vždy otestování:

a) syntaxe všech uživatelských postupů,

b) odolnosti proti známým typům útoků (XSS, CSRF, Session Steal, ClickJacking apod.),

c) zákazu používání tzv. skrytých polí pro důvěrná (citlivá) data,

d) zákazu používání přídavných identifikací uživatelských „session“ a obdobných autentizačních prostředků zakomponovaných v URL,

e) zákazu uvádění názvů souborů a adresářových cest v chybových hlášeních,

f) možností uživatelova odhlášení a automatického odhlášení po definované době jeho nečinnosti,

g) omezení pro používání Cookies na Cookies s časově omezenou platností, které jsou posílány zpět pouze stejnému serveru,

h) Java applety a případné jiné komponenty musí být podepsány důvěryhodnou certifikační autoritou,

i) komunikace aplikace s datovými zdroji v interní síti musí být autentizovaná,

j) možnost napadení DoS útokem,

případně další zranitelnosti definované tímto dokumentem (specifické testy).

4.1 Penetrační testy

Při penetračním testu Zhotovitel minimálně simuluje útok neoprávněné osoby vůči dvěma cílům, a to na:

a) VDP z vnější sítě

b) ISÚI z vnější sítě

Penetrační testy se provádějí, z hlediska efektivity a správnosti, s částečnou znalostí testovaného cíle, tzv. „gray box“.

Oba cíle budou prověřovány ve dvou úrovních a to:

- identifikace a prověření známých zranitelností na úrovni standardních webových služeb serveru;
- a identifikace a prověření známých zranitelností na úrovni architektury vlastní webové aplikace.

Penetrační testy musí vždy ověřit, zda webová aplikace, resp. webová služba, neobsahuje žádnou ze všech známých zranitelností uvedených v Příloze č. 1, bodu 1 tohoto dokumentu, spadajících pod OWASP TOP 10 – 2017.

(https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf)

Za tímto účelem jsou realizovány odpovídající testovací scénáře uvedené v Příloze č. 1 tohoto dokumentu.

Objednatel připouští realizaci bezpečnostních testů níže uvedenými způsoby, přičemž jejich použití k ověření oblastí testování v Příloze č. 1 ponechává na Zhotoviteli.

4.1.1 Automatizované testy a automatizované testy s manuálním podílem

Pro automatizované testy a automatizované testy s manuálním podílem bude použit některý ze SW nástrojů. Druh aktuálně použitého SW nástroje uvede Zhotovitel v dokumentu PBT.

4.1.2 Manuální testování

Manuální testování provede Zhotovitel v těch případech, kdy není možné využít automatizované testy nebo by použití automatizovaných testů nebylo dostatečně efektivní.

4.2 Specifické testy

Metodika OWASP obsahuje standardizované testy, tj. nezahrnuje všechny testovací scénáře zranitelností, které se mohou při vývoji informačního systému vyskytnout. Vzhledem k tomu budou dále pro zajištění bezpečného fungování RÚIAN prováděny též i další specifické testy.

Specifické testy budou vycházet a zohledňovat možná specifika kódu, zjištění ze sledování informačního servisu NÚKIB apod., zranitelnosti zjištěné při provozu RÚIAN, které se vyskytly jako bezpečnostní události nebo incidenty u nichž je nutné zajistit přijetí bezpečnostní opatření k zajištění jejich neopakovatelnosti nebo eliminaci a které vznikly v době před odpovídající aktualizací metodiky OWASP.

Seznam testovacích scénářů pro specifické testy je uveden v Příloze č. 1 tohoto dokumentu.

5 Předmět bezpečnostních testů RÚIAN

Předmětem bezpečnostních testů RÚIAN je:

- ISÚI – https://isui.cuzk.cz/isui/
- RÚIAN
- VDP - https://vdp.cuzk.cz/

Zhotovitel je vždy povinen zahrnout do testování další nové externí a interní části RÚIAN a dle toho aktualizovat tento dokument.

6 Testovací scénáře

Testovací scénáře musí zahrnovat následující údaje:

- název testovacího scénáře,
- ID testovacího scénáře,
- Tester – jméno
- verze systému,
- počet provedení scénáře,
- účel testu – popis, co je testem ověřováno,
- výchozí stav systému a vstupní podmínky,
- kroky testu – popis testovacích kroků a dat používaných pro testování,
- očekávané výsledky – kritéria úspěšnosti testu přiřazené ke každému z testovacích kroků.

7 Zpráva o výsledcích bezpečnostních testů

O provedení bezpečnostních testů pořizuje Zhotovitel Zprávu o výsledku bezpečnostních testů. Zpráva musí vždy obsahovat minimálně:

- Datum a čas provedení bezpečnostního testu
- Na jakém prostředí bylo testováno
- Změny aplikace, které jsou dodávány
- Seznam změn, které podléhají/nepodléhají bezpečnostním testům
- ID testovacího scénáře
- Jméno testera, který testování prováděl
- Manažerský souhrn s důležitými závěry bez technických detailů
- Technickou zprávu shrnující zjištění s technickými detaily a protokoly z testování
- Soupis zjištění

- Je-li součástí zprávy report generovaný nějakým SW nástrojem, je nutné specifikovat název a verzi nástroje, případně verzi pluginů. Zjištění musí být v celé zprávě jednotně klasifikována, přestože jsou použity různé SW nástroje, které mohou mít vlastní klasifikace

Sumarizaci zjištěných zranitelností/závad/slabin, včetně jejich závažnosti podle CVSS. Zprávu o výsledku bezpečnostních testů předá Zhotovitel garantovi aktiv RÚIAN nejpozději do 5-ti pracovních dnů od ukončení bezpečnostních testů.

Nalezené kritické zranitelnosti (CVSS>7) oznamuje Zhotovitel garantovi aktiv RÚIAN neodkladně po nalezení kritické zranitelnosti.

8 Přechodná ustanovení

Pro provádění testů se vychází ze standardu OWASP TOP10:2017 a metodologie testování podle OWASP v 4.0 (OWASP Testing Guide v4). Zhotovitel se zavazuje, že v případě uvolnění nové verze OWASP Top10 nebo OWASP Testing Guide bude tento dokument do měsíce od vydání nové verze OWASP aktualizovat v souladu s novou verzí a upravit i odpovídající testovací scénáře a používat odpovídající postupy.

Příloha č. 1 - Seznamy testovaných zranitelností, testovacích scénářů a specifických zranitelností

Seznam zranitelností podle OWASP Top 10 - 2017

Zranitelnost	Popis
A01: Injection	Zranitelnost typu injektáže (SQL, LDAP, XPath, NoSQL dotazů; příkazů operačního systému, XML parsování, SMTP hlaviček, programových argumentů, atd.) je velmi běžnou chybou webových aplikací, které nastává, pokud jsou přes neošetřený vstup uživatelem poskytnutá nedůvěryhodná data poslána do překladače jako část příkazu nebo dotazu. Např. u „SQL injection“ jde o vykonání vlastního, pozměněného SQL dotazu za účelem neoprávněného přístupu k informacím, jejich změně nebo i ovládnutí daného zařízení. Zranitelnosti typu injektáže lze snadno zjistit při revizi kódu, ale těžší je zjišťovat jejich přítomnost pomocí testů vzhledem k velké variabilitě manipulace parametrů http dotazů.
A02: Broken Authentication	Vývojáři často vytváří autentizační mechanismy a řízení relací, ale jejich správné vytvoření není jednoduché. Jako výsledek těchto snah bývají často zranitelnosti v oblastech odhlášení, správy hesel, dlouhé časové limity pro relace, aktualizace účtů atd. Útočníci mohou kompromitovat hesla, klíče nebo autentizační identifikátory k předstírání jiných uživatelských identit. Nalezení těchto zranitelností může být občas těžké, protože každá takováto implementace bývá jedinečná.
A03: Sensitive Data Exposure	Nejběžnější chybou je nešifrování citlivých dat. Pokud se používá šifrování, jde o generování slabých klíčů, použití slabých šifrovacích algoritmů nebo slabé hashovací techniky pro hesla. Zranitelnosti v prohlížeči jsou velmi časté a snadno odhalitelné, ale těžko zneužitelné ve velkém měřítku.
A04: XML External Entities (XXE)	Ve výchozím nastavení mnoho starších procesorů XML umožňuje specifikaci externí entity, URI, která je dereferencována a vyhodnocena během zpracování XML. Nástroje SAST mohou tento problém zjistit kontrolou závislostí a konfigurace. Nástroje DAST vyžadují další ruční kroky k odhalení a zneužití tohoto problému. Jde o novou zranitelnost, zatím nebyla testována.
A05: Broken Access Control	Aplikace často používají skutečný název nebo klíč objektu při generování webových stránek. Aplikace ne vždy ověřuje, zda je uživatel oprávněn přistupovat k cílovému objektu. Útočník tak může neoprávněně manipulovat s těmito odkazy a přistupovat k jiným objektům (bez autorizace). Testeři můžou snadno manipulovat hodnoty parametrů k detekci takovýchto zranitelností. Analýza kódu rychle ukáže, zda povolení je řádně ověřeno.
A06: Security Misconfiguration	Bezpečnostně chybná konfigurace může nastat na jakékoliv úrovni infomačního systému ať už to je webový server, aplikační server, databáze, framework, atd. Vývojáři a systéméví administrátoři musí úzce spolupracovat, aby zajistili, že konfigurace všech částí infomačního systému je v pořádku. Automatizované scannery jsou vhodné pro detekci chybějících patchů, použití defaultních účtů, nepotřebných služeb, apod.
A07: Cross Site	XSS je nejrozšířenější zranitelnost webových aplikací. XSS zranitelnost nastává, pokud aplikace zahrne uživatelem poskytnutá data do webové

Zranitelnost	Popis
Scripting (XSS)	stránky a pošle ji do prohlížeče, aniž by tato data řádně validoval nebo byly escapovány. To umožní ve webovém prohlížeči oběti spustit útočníkův skript, který může např. neoprávněně převzít uživatelskou relaci, změnit obsah stránek, instalovat škodlivé programy apod. Detekce většiny XSS zranitelností je poměrně snadná jak testováním, tak i revizí kódu nebo konfigurací webového serveru.
A08: Insecure Deserialization	Tato zranitelnost je zahrnuta do Top 10 na základě průzkumu v oboru, nikoli na kvantifikovatelných údajích o výskytu. Některé nástroje mohou objevit chyby v deserializaci, ale pro potvrzení problému je často potřeba pomoc člověka. Očekává se, že údaje o prevalenci v případě nedostatků způsobených deserializací se zvýší, protože nástroj je stále vyvíjen, aby pomohl identifikovat a řešit problém. Dopad deserializačních zranitelností nemůže být podceňován. Tyto zranitelnosti mohou vést k útokům typu vzdálené spuštění kódu, což je jeden z nejzávažnějších možných útoků.
A09: Using Components with Known Vulnerabilities	Prakticky každá aplikace má problémy s použitím komponent (knihovny, frameworky a další softwarové moduly) obsahujících známé zranitelnosti, protože většina vývojářů se nesoustředí na zajištění aktualizací komponenty/knihoven. V mnoha případech vývojáři ani neznají, jaké všechny komponenty se používají, natož jejich verze. Závislosti komponent situaci ještě zhoršují. Detekce se provádí zpravidla lokálně v rámci zdrojového kódu, ale částečně ji lze provést i pomocí penetračního testu.
A10: Insufficient Logging & Monitoring	Tato zranitelnost je zahrnuta do Top 10 na základě průzkumu v oboru, nikoli na kvantifikovatelných údajích o výskytu. Jedna z možných strategií pro zjištění, zda je správně nastaven monitoring a logování, je prověřit protokoly po penetračním testování. Činnosti testerů by měly být dostatečně zaznamenány, aby bylo možné zjistit, jaké škody by mohly být způsobeny. Nejúspěšnější útoky začínají zkoumáním zranitelnosti. Povolení pokračování takových zkoumání může zvýšit pravděpodobnost úspěšného útoku téměř na 100%.

Seznam specifických zranitelností

Aktuálně bez specifických zranitelností testovaných specifickými testy.

Testovací scénáře OWASP (dle OWASP Testing Guide v4.0)

4.2 Information Gathering (Sběr informací)	
OTG-IG-001 - 4.2.1 **Conduct search engine discovery/reconnaissance for information leakage**	Provést sběr informací o cíli s využitím vyhledávače Google. Provést sběr informací o cíli s využitím robots.txt.
OTG-IG-002 - 4.2.2 **Fingerprint Web Server**	Najít verzi a typ běžícího webového serveru, aby se zjistili známá zranitelná místa a příslušné zneužití, které je třeba použít při testování

OTG-IG-003 - 4.2.3 **Review Webserver Metafiles for Information Leakage**	Analyzovat robots.txt použitím Google Webmaster Tools.
OTG-IG-004 - 4.2.4 **Enumerate Applications on Webserver**	Identifikovat aplikace, které existují v daném rozsahu. Black box pentest
OTG-IG-005 - 4.2.5 **Review Webpage Comments and Metadata for Information Leakage**	Zjistit jaká webová aplikace běží na webovém serveru.
OTG-IG-006 - 4.2.6 **Identify application entry points**	Analyzovat, jak jsou vytvářeny požadavky a typické odpovědi z aplikace
OTG-IG-007 - 4.2.7 **Map execution paths through application**	Mapování cílové aplikace a pochopení hlavních pracovních postupů.
OTG-IG-008 - 4.2.8 **Fingerprint Web Application Framework**	Definovat typ použitého webového rámce tak, aby se upřesnila metodika testování zabezpečení.
OTG-IG-009 - 4.2.9 **Fingerprint Web Application**	Identifikace webové aplikace a verze, aby se zjistili známá zranitelná místa a příslušné zneužití, které je třeba použít při testování.
OTG-IG-010 - 4.2.10 **Map Application Architecture**	Analyzovat architekturu aplikace a mapovat vzájemné vazby mezi aplikací a dalšími programy.
4.3 Configuration and Deployment Management Testing (Testování managementu konfigurace a nasazení)	
OTG-CONFIG-001 - 4.3.1 **Test Network/Infrastructure Configuration**	Otestovat konfiguraci infrastruktury, která podporuje aplikaci, identifikovat slabá místa v zabezpečení RUIAN.
OTG-CONFIG-002 - 4.3.2 **Test Application Platform Configuration**	Přezkoumání a testování konfigurace. Testování přítomnosti defaultních nastavení, jako např. Directory traversal vulnerability, Use of sendmail.jsp atd.
OTG-CONFIG-003 - 4.3.3 **Test File Extensions Handling for Sensitive Information**	Určení způsobu, jakým webové servery zpracovávají požadavky odpovídající souborům s různými rozšířeními, mohou pomoci pochopit chování webového serveru v závislosti na druhu souborů, ke kterým je přístup.
OTG-CONFIG-004 - 4.3.4 **Review Old, Backup and Unreferenced Files for Sensitive Information**	Prověřit a vyhledat nereferenční nebo zapomenuté soubory, které lze použít k získání důležitých informací o infrastruktuře nebo pověřeních.
OTG-CONFIG-005 - 4.3.5 **Enumerate Infrastructure and Application Admin Interfaces**	Rozhraní správce mohou být nastaveny v aplikaci nebo na aplikačním serveru, což umožňuje určitým uživatelům provádět privilegované činnosti na webu. Provést testy s cílem zjistit, zda a jak může tato privilegovaná funkce získat přístup neoprávněnému nebo standardnímu uživateli.
OTG-CONFIG-006 - 4.3.6 **Test HTTP Methods**	Zjistit povolené http metody a možnosti jejich zneužití včetně Cross Site Tracing (XST).

OTG-CONFIG-007 - 4.3.7 **Test HTTP Strict Transport Security**	Ověřit, zda web používá hlavičku HTTP, aby bylo zajištěno, že všechna data budou šifrována z webového prohlížeče na server.
OTG-CONFIG-008 - 4.3.8 **Test RIA cross domain policy**	Rich Internet Application (RIA) používá politiku Adobe crossdomain.xml pro řízení cross domain přístupů. Testovat konfiguraci soubory zásad popisujících omezení přístupu proti CSRF útokům.
OTG-CONFIG-009 - 4.3.9 **Test File Permission**	Testovat konfiguraci oprávnění souboru pro ochranu před zneužitím eskalace privilegií, injekci DLL nebo neoprávněným přístupem k souborům
4.4 Identity Management Testing (Testování managementu identit)	
OTG-IDENT-001 - 4.4.1 **Test Role Definitions**	Otestovat a pokusit se zachytit záhlaví paketů a jejich prohlížení. Využije se WebScarab nebo jiný libovolný webový proxy.
OTG-IDENT-002 - 4.4.2 **Test User Registration Process**	Ověřit, zda jsou požadavky na totožnost pro registraci uživatelů sladěny s požadavky definovaných politik a zabezpečení. Ověřit proces registrace, zda je validní.
OTG-IDENT-003 - 4.4.3 **Test Account Provisioning Process**	Prověřit existenci defaultních nebo snadno uhodnutelných uživatelských účtů. Ověřte, které účty mohou poskytovat další účty a jaký typ.
OTG- IDENT -004 - 4.4.4 **Testing for Account Enumeration and Guessable User Account**	Ověřit zda je možné získat uživatelská jména interakcí s autentizačním mechanismem aplikace. Provést útok hrubou silou na přihlašovací údaje.
OTG- IDENT - 005 - 4.4.5 **Testing for Weak or unenforced username policy**	Prověřit zda lze obejít autentizační mechanismus.
4.5 Autentification Testing (Testování Autentifikace)	
OTG-AUTH-001 - 4.5.1 **Testing for Credentials Transported over an Encrypted Channel**	Testovat, že uživatelská autentifikační data jsou přenášena přes šifrovaný kanál, aby se zabránilo zachycení útočníkem.
OTG- AUTH -002 - 4.5.2 **Testing for default credentials**	Provést test na přítomnost defaultních nebo známých uživatelských jmen a hesel pro zařízení v síti, která by vedla k úspěšné autentizaci
OTG- AUTH -003 - 4.5.3 **Testing for Weak lock out mechanism**	Prověřit aplikaci na možnou zranitelnost mechanismu blokování účtů odolnost vůči brute-force útokům. Vyhodnoťte odolnost mechanismu odblokování před neoprávněným odblokováním účtu
OTG-AUTH-004 - 4.5.4 **Testing for Bypassing Authentication Schema**	Zjistit zda lze obejít autentifikační opatření tím, že manipulujete s žádostmi a podváděním aplikace, že si uživatel již ověřil. Toho lze dosáhnout buď úpravou daného parametru adresy URL, manipulací s formulářem nebo paděláním relací.

OTG- AUTH -005 - 4.5.5 **Testing for Vulnerable Remember Password**	Hledejte hesla uložená v souboru cookie. Zkontrolujte soubory cookie uložené v aplikaci. Ověřte, zda pověření nejsou uložena v čistém textu, ale jsou šifrovaná. Prověřte mechanismus hashování: je-li to běžný, dobře známý algoritmus, zkontrolujte jeho sílu
OTG- AUTH -005 - 4.5.6 **Testing for Browser cache weakness**	Testovat zranitelnost prohlížeče na dříve zadané citlivé informace.
OTG- AUTH -005 - 4.5.7 **Testing for Weak password policy**	Testovat odolnost aplikace před brute-force útokům uhádnutí hesla pomocí dostupných slovníků hesel vyhodnocením požadavků na délku, složitost, opětovné použití a expiraci hesel.
OTG- AUTH -008 - 4.5.8 **Testing for Weak security question/answer**	Testovat na přítomnost lehce uhodnutelných otázek pro obnovu hesla.
OTG- AUTH -009 - 4.5.9 **Testing for weak password change or reset functionalities**	Určete odolnost aplikace proti možnosti změny účtu, která umožňuje někomu změnit heslo účtu. Určete odolnost funkce resetování hesel proti uhádnutí nebo obejití
OTG- AUTH-010 - 4.5.10 **Testing for Weaker authentication in alternative channel**	Provedení testů k identifikaci alternativních kanálů a, v závislosti na rozsahu testování, identifikovat zranitelnosti autentifikace.
4.6 Authorization Testing (Prověření autorizace)	
OTG-AUTHZ-001 - 4.6.1 **Testing Directory traversal/file include**	Testovat odolnost aplikace vůči Path Traversal útoku.
OTG- AUTHZ -002 - 4.6.2 **Testing for bypassing authorization schema**	Prověřit zda lze obejít autorizační mechanismus (např. přístup k funkcím/datům náležícím jiné uživatelské roli).
OTG- AUTHZ -003 - 4.6.3 **Testing for Privilege Escalation**	Prověřit aplikaci na zranitelnost typu eskalace privilegií.
OTG- AUTHZ -004 - 4.6.4 Testing for Insecure Direct Object References	Prověřit aplikaci na zranitelnost výskytu nesprávných odkazů na přímý objekt, když aplikace poskytuje přímý přístup k objektům založeným na uživatelském vstupu. V důsledku této zranitelnosti mohou útočníci obejít autorizaci a přístup k prostředkům přímo v systému, například databázové záznamy nebo soubory.
4.7 Session Management Testing (Správa relace)	
OWASP-SESS-001 - 4.7.1 **Testing for Session Management Schema**	Zkontrolovat cookie a jiné identifikátory relace zda jsou vytvořené bezpečným a nepředvídatelným způsobem.
OWASP- SESS -002 - 4.7.2 **Testing for Cookies attributes**	Prověřit správné nastavení cookie atributů.
OWASP- SESS -003 - 4.7.3 **Testing for Session Fixation**	Prověřit aplikaci na možnou zranitelnost session fixation (po úspěšné autentizaci se nezmění identifikátor relace).

OWASP- SESS -004 - 4.7.4 **Testing for Exposed Session Variables**	Zjistit zda jsou identifikátory relace dostatečně chráněné.
OWASP- SESS -005 - 4.7.5 **Testing for CSRF**	Testovat odolnost aplikace vůči CSRF útoku.
OWASP- SESS -006 - 4.7.6 **Testing for logout functionality**	Testovat možnost prvků uživatelského rozhraní, která umožňují uživateli ručně se odhlásit se. Ověřit nastavení ukončení relace po určitém čase bez aktivity (časový limit relace). Ověřit správné zneplatnění stavu relace na straně serveru.
OWASP- SESS -007 - 4.7.7 **Test Session Timeout**	Otestovat že aplikace automaticky odhlásí uživatele, když byl uživatel po určitou dobu nečinný
OWASP- SESS -008 - 4.7.8 **Testing for Session puzzling**	Testovat zabezpečení aplikace na přítomnost a používání stejné proměnné relace pro více než jeden účel.
4.8 Input Validation Testing (Testování validace dat)	
OTG-INPVAL-001 - 4.8.1 **Testing for Reflected Cross Site Scripting**	Prověřit existenci nepersistentních XSS (Cross Site Scripting) zranitelností.
OTG- INPVAL -002 - 4.8.2 **Testing for Stored Cross Site Scripting**	Prověřit existenci persistentních XSS (Cross Site Scripting) zranitelností.
OTG- INPVAL -003 - 4.8.3 **Testing for DOM based Cross Site Scripting**	Prověřit existenci DOM (document object model) XSS zranitelností.
OTG- INPVAL -004 - 4.8.4 **Testing for HTTP Parametr polution**	Prověřit existenci XSF (Cross Site Flashing) zranitelností.
OTG- INPVAL-005 - 4.8.5 **SQL Injection**	Prověřit existenci SQL Injection zranitelností.
OTG-DV-006 - 4.8.6 **LDAP Injection**	Prověřit existenci LDAP Injection zranitelností.
OTG-DV-007 - 4.8.7 **ORM Injection**	Prověřit existenci ORM Injection (Object Relational Mapping) zranitelností.
OTG-DV-008 - 4.8.8 **XML Injection**	Prověřit existenci XML Injection zranitelností.
OTG-DV-009 - 4.8.9 **SSL Injection**	Prověřit existenci SSI Injection (Server-Side Includes) zranitelností.
OTG-DV-010 - 4.8.10 **XPath Injection**	Prověřit existenci XPath Injection (XML Path Language) zranitelností.
OTG-DV-011 - 4.8.11 **IMAP/SMTP Injection**	Prověřit existenci IMAP/SMTP zranitelností.
OTG-DV-012 - 4.8.12 **Testing for Code Injection**	Prověřit existenci Code Injection zranitelností.
OTG-DV-012 - 4.8.12.1 **Testing for Local File Inclusion**	Prověřit existenci zranitelností (LFI) v podobě volání nějakého lokálního souboru skriptem.
OTG-DV-012 - 4.8.12.2 **Testing for Remote File Inclusion**	Prověřit existenci zranitelností (RFI) v podobě volání nějaké webové aplikace externím skriptem.

OTG-DV-013 - 4.8.13 Testing for Command Injection	Prověřit existenci zranitelností umožňující spuštění příkazů operačního systému.
OTG-DV-014 - 4.8.14 **Testing for Buffer overflow**	Prověřit existenci zranitelností umožňující přetečení zásobníku.
OTG-DV-015 - 4.8.15 **Incubated vulnerability**	Provést test vícenásobný zneužití zranitelností, kdy je např. nahrán škodlivý obsah aplikace uživatelům, kteří následně tento kód spustí.
OTG-DV-016 - 4.8.16 **Testing for HTTP Splitting/Smuggling**	Prověřit existenci zranitelností v http hlavičce.
OTG-DV-016 - 4.8.17 **Testing for HTTP Incoming requests**	Prověřit existenci zranitelností v http vstupním požadavku.
4.9 Testing for Error Handling (Testování zranitelností na dostupnost služeb)	
OTG-ERR-001 - 4.9.1 **Analysis of Error Codes**	Prověřit existenci Denial of Service (DoS) zranitelností na SQL zástupné znaky.
OTG-ERR-002 - 4.9.2 **Analysis of Stack Traces**	Zjistit zda lze pomocí špatně zadaných hesel uzamknout platný uživatelský účet.
OTG-DS-003 - 4.9.3 **Testing for DoS Buffer Overflows**	Zjistit zda lze pomocí přetečení zásobníku způsobit DoS.
OTG-DS-004 - 4.9.4 **User Specified Object Allocation**	Zkontrolovat, zda je možné vyčerpat zdroje serveru tím, že se alokuje velmi vysoký počet objektů.
OTG-DS-005 - 4.9.5 **User Input as a Loop Counter**	Zkontrolovat, zda je možné vnutit aplikaci smyčku prostřednictvím kódu segmentu, který potřebuje významnou část výpočetních zdrojů, aby se snížila celkovou výkonnost např. tím, že uživatel může přímo nebo nepřímo přiřadit hodnotu, která bude používána jako čítač ve smyčce.
OTG-DS-006 - 4.9.6 **Writing User Provided Data to Disk**	Zkontrolovat, zda je možné vyčerpat zdroje serveru tím, že se zaplní disk logy.
OTG-DS-007 - 4.9.7 **Failure to Release Resources**	Zkontrolovat, zda aplikace řádně uvolní zdroje (soubory a / nebo paměť) poté, co byly použité.
OTG-DS-008 - 4.9.8 **Storing too Much Data in Session**	Zkontrolovat, zda je možné přidělit velké množství dat do uživatelské relace, aby server vyčerpal své paměťové zdroje.
4.10 Testing for weak Cryptography (Testování slabé kryptografie)	
OTG-CRYPST-001 - 4.10.1 **Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection**	Testovat nedostatečnou sílu SSL/TLS

OTG-CRYPST-002 - 4.10.2 **Testing for Padding Oracle**	Testovat na chyby „Padding Oracle“ neboli funkce aplikace, která dešifruje zašifrované údaje poskytované klientem, např. stavy interní relace uložené v klientovi a úniku stavu platnosti funkce po dešifrování. Existence této zranitelnosti umožňuje útočníkovi dešifrovat šifrované data a šifrovat libovolná data bez znalosti klíčů použitého pro tyto kryptografické operace.
OTG-CRYPST-003 - 4.10.3 **Testing for Sensitive information sent via unencrypted channels**	Testovat na chyby zabezpečení přenosového kanálu, v kterém mohou být přenášeny informace v čistém textu. Zkontrolovat, zda jsou tyto informace přenášeny přes protokol HTTP namísto protokolu HTTPS nebo zda jsou používány slabé Cypher algoritmy.
OTG-CRYPST-004 - 4.10.4 **Testing for Weak Encryption**	Testovat na přítomnost slabých kryptokódů.
4.11 Business logic testing (Prověření logiky aplikace)	
OTG-BUSLOGIC-001 - 4.11.1 **Testing for Business Logic data validation**	Testovat na chyby v logice aplikace umožňující uživateli provést operaci s daty jiným způsobem než bylo navrženo.
OTG-BUSLOGIC-002 - 4.11.2 **Test Ability to forge requests**	Testovat zranitelnosti vůči využití proxy k odeslání žádostí HTTP POST / GET do aplikace Zkontrolujte projektovou dokumentaci a použijte průzkumné testování, které hledá odhadnutelnou, předvídatelnou nebo skrytou funkcionalitu polí.
OTG-BUSLOGIC-003 - 4.11.3 **Test integrity checks**	Testovat na chyby v zajištění integrity aplikace. Odolnost vůči nepovolenému odeslání hodnot skrytých polí serveru pomocí serveru proxy
OTG-BUSLOGIC-004 - 4.11.4 **Test for Process Timing**	Testovat na časové odezvy při nesprávném zadání autentifikačních údajů.
OTG-BUSLOGIC-005 - 4.11.5 **Test number of times a function can be used limits**	Zkontrolujte projektovou dokumentaci a použijte testování, které hledá funkce nebo funkce v aplikaci nebo systému, které by neměly být prováděny více než jednou nebo pouze určitým počtem opakování během pracovního postupu v aplikaci.
OTG-BUSLOGIC-006 - 4.11.6 **Testing for the Circumvention of Work Flows**	Testovat na chyby v logice aplikace umožňující uživateli provést operaci s daty jiným způsobem než bylo navrženo.
OTG-BUSLOGIC-007 - 4.11.7 **Test defenses against application misuse**	Testovat na přítomnost obranných mechanismů v aplikační vrstvě, které chrání aplikaci proti nesprávnému použití nebo neplatnému použití platné funkce, které se snaží kompromitovat webovou aplikaci, identifikovat slabé stránky a zneužívat zranitelnosti.

OTG-BUSLOGIC-008 - 4.11.8 **Test Upload of Unexpected File Types**	Testovat mechanismus ověřování správného typu souborů. Aplikace může očekávat, že budou na zpracovávány pouze určité typy souborů, jako jsou soubory .CSV, .txt. Aplikace musí ověřovat nahraný soubor buď podle přípony (pro ověření souboru s nízkou jistotou) nebo podle obsahu (ověření souboru s vysokou jistotou). To může vést k neočekávaným výsledkům systému nebo databáze v rámci aplikace / systému nebo k tomu, že útočníkům poskytnou další metody pro využití aplikace / systému.
OTG-BUSLOGIC-009 - 4.11.9 **Test Upload of Malicious Files**	Testovat na zranitelnost vůči škodlivým kódům.
4.12 Client Side Testing (Testování klienta)	
OTG-CLIENT-001 - 4.12.1 **Testing for DOM-based Cross site scripting**	Prověřit existenci DOM (document object model) XSS zranitelností.
OTG-CLIENT-002 - 4.10.2 **Testing for JavaScript Execution**	Otestovat provádění JAVA skriptů a ověřit, zda nelze získat osobní data uživatele nebo upravit obsah web stránky, kterou uživatel může vidět. Chyba zabezpečení typu JavaScript Injection je podtyp CrossScriptingu (XSS), který zahrnuje možnost vkládat libovolný kód JavaScript, který aplikace provádí uvnitř prohlížeče oběti.
OTG-CLIENT-003 - 4.12.3 **Testing for HTML Injection**	Prověřit odolnost vůči zranitelnosti typu HTML injection.
OTG-CLIENT-004 - 4.12.4 **Testing for Client Side URL Redirect**	Zkontrolovat odolnost aplikace, když aplikace přijímá nedůvěryhodný vstup, který obsahuje hodnotu URL, aniž by jej dezinfikoval. Odolnost vůči přesměrování webové aplikace na jinou stránku.
OTG-CLIENT-005 - 4.12.5 **Testing for CSS Injection**	Prověřit odolnost vůči zranitelnosti typu CSS Injection.
OTG-CLIENT-006 - 4.12.6 **Testing for Client Side Resource Manipulation**	Otestovat odolnost vůči zranitelnosti typu Client Side Resource Manipulation.
OTG-CLIENT-007 - 4.12.7 **Test Cross Origin Resource Sharing**	Prověřit používání CORS a otestovat, že není změněn Javascriptem. Otestovat protokoly na úrovni aplikace, že se používají k ochraně citlivých dat.
OTG-CLIENT-008 - 4.12.8 **Testing for Cross site flashing**	Prověřit existenci XSF (Cross Site Flashing) zranitelností.
OTG-CLIENT-009 - 4.12.9 **Testing for Clickjacking**	Otestovat odolnost vůči útokům typu Clickjacking

OTG-CLIENT-010 - 4.12.10 **Testing WebSockets**	Prověřit zda je webová služba přístupná přes HTTP a zda server ověřuje hlavičku Origin v počátečním handshake HTTP WebSocket. Pokud server neověřuje záhlaví původu v počátečním handshake serveru WebSocket, server WebSocket může přijímat připojení z libovolného původu.
OTG-CLIENT-011 - 4.10.12.11 **Test Web Messaging**	Je třeba provést ruční testování a kód JavaScript analyzovat hledáním implementace služby Web Messaging. Zejména je třeba prověřit, jak webové stránky omezují zprávy z nedůvěryhodné domény a jak se s nimi zachází i pro důvěryhodné domény
OTG-CLIENT-012 - 4.10.12 **Test Local Storage**	Prověřit existenci lokálního uložiště (Web Storage nebo Offline Storage), což je mechanismus pro ukládání dat jako párů klíč / hodnota svázaných s doménou a vynucených stejnou zásadou původu (SOP). Existují dva objekty, localStorage, který je trvalý a má uchovat data i po restartování prohlížeče / systému a sessionStorage, který je dočasný a bude existovat pouze dokud nebude okno nebo karta uzavřena.