

# MASS: RAZJASNIMO ZMEDO VARNOSTI V APLIKACIJAH

Stanka Šalamun

ACROS d.o.o.

e-pošta: security@acrossecurity.com

URL: <http://www.acrossecurity.com>

## *Povzetek*

*Številni odkriti vdori v najbolj varnostno kritične inteligentne poslovne sisteme so znamenje, da aplikacije, ki jih sestavljajo, niso grajene na način, da bi vzdržale tudi najbolj osnovne napade. Trenutni obstoječi varnostno-informacijski standardi za boj proti vedno bolj motiviranim napadalcem na programsko opremo so nekonkretni ali prezapleteni za preprosto vsakdanjo rabo, zato smo izdelali predlog formalnega modela za opis aktivnosti sistematične gradnje varne programske opreme, ki je predvsem preprost, konkreten, uporaben za obdelavo velikega števila aplikacij in ki omogoča primerjavo - »MASS« (Model for Application Security Strategy). Model smo v praksi preizkusili na razpisih za informacijske sisteme v javni upravi.*

## **1. UVOD**

Če je včasih veljalo, da je aplikacija varna, če ima vgrajeno dovolj veliko število varnostnih mehanizmov, je danes potrebno, da mora zdržati napade zlonamernih napadalcev – da je torej čim manj »varnostno luknjasta«.

Izvajalci in naročniki se pri gradnji varnih sistemov za poslovno inteligenco oklepajo izgovorov, zaradi katerih se velikokrat zavestno odločajo za manj varne izvedbe svojih aplikacij. Izvajalci se izgovarjajo, da popolne varnosti ni, zato so tudi manj resni pri odkrivanju in odpravljanju ranljivosti, tudi tistih izjemno nevarnih in onih, ki jih je sorazmerno lahko odkriti. Prav tako velja splošno mnenje, da graditelji vedo največ o svojem sistemu, pri čemer pozabljajo, da je dovolj, da dobro trenirani napadalci najdejo eno samo napako, pa je varnost podatkov kompromitirana za vedno. Tako jim je težko odkriti tudi osnovne varnostne napake, čeprav jih v teoriji dobro poznajo, kaj šele, da bi redno sledili razvoju novih vrst napadov in spremljanju na novo odkritih vrst ranljivosti. Še največ škode pa naredimo s tem, da krepimo močne varnostne člene in ne najšibkejših. Z vgradnjo najsodobnejše varnostne tehnologije samo neoptimalno zapravljamo sredstva, saj bodo napadalci še vedno prihajali v sistem po šibkih členih. Zato je potreba po modelu strategije gradnje varnih sistemov poslovne inteligence, ki je bil ključni cilj raziskovalnega projekta, predstavljenega v prispevku, vedno večja.

V raziskovalnem projektu smo razvijali preprost model – strategijo za gradnjo varnih

sistemov poslovne inteligence, ki ne bi temeljila na slepi vgradnji varnostnih mehanizmov ter definiranju dostopov uporabnikov, temveč na varnostnih aktivnostih, sistematično vgrajenih v življenjski cikel gradnje programske opreme. Model upošteva dejstvo, da imajo različne aplikacije iz različnih razlogov različne varnostne potrebe, kar pomembno vpliva na izvajanje projektnih aktivnosti. Na podlagi dejanskih varnostnih potreb sistema poslovne inteligence je v model vgrajeno znanje o potrebnih aktivnostih za zagotovitev potrebnega varnostnega nivoja v obliki preprostega seznama aktivnosti, ki v določenih delih razvojnega cikla prinašajo največje učinke.

Na podlagi izvedenega modela za strategijo varnih sistemov poslovne inteligence smo izvedli testno analizo na reprezentančnem vzorcu projektov, za katere smo lahko pridobili vse potrebne informacije. Izkazalo se je, da nam je največ verodostojnih in primerljivih podatkov dostopnih pri javnih naročilih IT projektov, ki smo jih uporabili za izvedbo analize. Rezultati so v zgoščenih obliki predstavljeni v prispevku.

Eden od ciljev raziskovalnega projekta je bil prva primerjava razvojnih IT projektov na podlagi modelnih metrik. Nenazadnje pa z raziskavo in prispevkom želimo spodbuditi naročnike varnostno kritičnih sistemov poslovne inteligence, da v proces naročanja od samega začetka vključijo optimalne aktivnosti izvajalcev za večjo varnosti programske opreme.

Ker v drugih virih ([1], [2], [3], [4], [5], [6]) bolj podrobno opisujemo modele in standarde, ki so vplivali na zasnovo MASS ([7],[8],[9]), se bomo v tem prispevku usmerili na zapis posameznih sestavnih elementov modela ter samo v obliki povzetka predstavili izsledke raziskave.

## **2. OPIS MODELA »MASS«**

### **2.1 Tri ključna vprašanja**

Model strategije gradnje varnih sistemov je namenjen hitremu in učinkovitemu nadzoru varne gradnje velikega števila aplikacij, zato mora biti proces ocenjevanja preprost. Uporabnik modela si v splošnem zastavi tri ključna vprašanja:

- Kakšen nivo varnosti zares potrebujemo?
- Smo načrtovali ključne aktivnosti za doseganje določenega varnostnega nivoje?
- Smo pred uporabo izvedli vse zahtevane ključne varnostne aktivnosti željenega varnostnega nivoja?

Odgovore na vprašanja poiščemo s pomočjo dveh vprašalnikov. V vprašalniku »Ciljni aplikacijski nivo varnosti (CANV)« določimo potrebo po nivoju informacijske varnosti v aplikaciji. Ocena CANV je izražena kot vrednost med 0 in 6, pri čemer 0 predstavlja manjše varnostne potrebe in 6 najvišje.

V nadaljnjih korakih izpolnjujemo vprašalnik »Ocenjeni aplikacijski nivo varnosti (OANV)«, ki govori o načrtovanih in izvedenih varnostnih aktivnostih pri gradnji sistema poslovne inteligence, glede na fazo razvojnega življenjskega cikla. Oceno OANV primerjamo z oceno CANV ter tako ugotovimo odstopanja načrtovanih ali dejanskih varnostnih aktivnosti od željenega varnostnega nivoja.

### 3. DOLOČITEV CILJNEGA APLIKACIJSKEGA NIVOJA VARNOSTI (»CANV«)

Kljub veliki razpoložljivosti različnih varnostnih mehanizmov ne bi imelo pravega smisla, da bi bila prav vsaka programska oprema zgrajena na popolnoma varen način in da je vsa programska oprema enako varna. Za marsikatero aplikacijo niti ne pričakujemo, da je grajena popolno, je pa nujno, da ocenimo, kak nivo varnosti dejansko potrebuje. Ključni kriteriji za ocenjevanje varnostnih potreb aplikacije ne bodo število in sofisticiranost varnostnih mehanizmov ali uporabniki, temveč predvsem vrsta podatkov, dostop, upravljanje z zasebnostjo in poslovnimi skrivnostmi, vrste uporabljenih tehnologij, število potencialnih uporabnikov, povezljivost, zakonske zahteve, ali aplikacija lahko ogroža premoženje, zdravje ali življenje ljudi in podobno.

V grobem lahko aplikacije popredalčakamo v naslednje nivoje od 0 do 6: "0. Nima varnostnih zahtev", "1. Osnovna varnost", "2. Javni dostop", "3. Podatki", "4. Denar", "5. Skrivnost", "6. Življenje". Če lahko za katerikoli kriterij iz vprašalnika CANV na določenem nivoju odgovorimo pritrdilno, velja, da sodi aplikacija po zahtevah za varnost vsaj na ta določeni nivo. Ocena CANV pa predstavlja najvišji varnostni nivo zahtev, na katerega smo postavili aplikacijo v času ocenjevanja.

V nadaljevanju si pobližje pogledjmo kriterije za posamezne ciljne aplikacijske nivoje varnosti.

#### 3.1 NIVO 1 - »OSNOVNA« – Aplikacija z osnovnimi varnostnimi zahtevami

- **Kompleksna:** Aplikacija ima široko področje napada ("attack surface"), ki je dostopno potencialnim napadalcem. Je kompleksna, vsebuje veliko število programskih modulov, programskih vmesnikov, veliko število vrstic kode in kompleksno arhitekturo. Podpira večje število različnih protokolov. V tako kompleksni aplikaciji pričakujemo večje število možnih napak.
- **Intranet povezljiva:** Aplikacija se povezuje z drugimi internimi aplikacijami ali omogoča povezavo z drugimi internimi bazami podatkov.
- **Aplikacija na računalniku z dostopom do Interneta:** Aplikacija je namenjena za uporabo na računalniku, ki ima posreden ali neposreden dostop do Interneta, osebnih podatkov ali poslovnih skrivnosti.

### 3.2 NIVO 2 - »JAVNI DOSTOP« - Aplikacija z višjimi varnostnimi zahtevami

- **Omogoča anonimnost:** Aplikacija omogoča anonimno uporabo ali prijavo v sistem kot "gost".
- **Veliko uporabnikov:** Aplikacija ima veliko število potencialnih uporabnikov.
- **Dostopna z Interneta:** Aplikacija je v delu ali celoti dostopna internetnim uporabnikom.

### 3.3 NIVO 3 - »PODATKI« - Aplikacija z visokimi varnostnimi zahtevami

- **Dostop do osebnih podatkov:** Aplikacija dostopa do ali upravlja z osebnimi podatki, kot so definirani v ZVOP.
- **Dostop do poslovnih skrivnosti:** Aplikacija dostopa do ali upravlja s poslovnimi skrivnostmi, ki so jasno označeni kot take.
- **Visoka razpoložljivost:** Aplikacija mora biti dostopna ves čas

### 3.4 NIVO 4 – »DENAR« - Varnostno kritična aplikacija

- **Finančni podatki in transakcije:** Aplikacija izvaja ali dostopa do finančnih transakcij. Aplikacija dostopa do ali upravlja s finančnimi podatki ali podatki o lastnini.
- **Posebne metode varovanja:** Aplikacija dostopa do ali upravlja s podatki, za katere zakonodaja predpisuje posebne metode varovanja, kot so ZVDAGA, ZEPEP, ZDavP-2, ZZPPZ (razen kar predpisuje ZVOP-1). Aplikacija je podvržena usklajenosti s strokovnimi standardi, kot so HL7, HIPAA, DICOM ipd.
- **IntErnet povezljiva:** Aplikacija se po javnih računalniških omrežjih povezuje z drugimi aplikacijami ali omogoča povezavo z drugimi bazami podatkov.

### 3.5 NIVO 5 – »SKRIVNOST« - Visoko varnostno kritična aplikacija

- **Občutljivi osebni podatki:** Aplikacija dostopa do ali upravlja z občutljivimi osebnimi podatki, kot so definirani v ZVOP-1.
- **Zaupni podatki:** Aplikacija dostopa do ali upravlja s podatki, ki so jasno označeni kot stroga poslovna skrivnost ali s podatki, ki so po ZTP označenih z oznako "ZAUPNO".

### 3.6 NIVO 6 – »ŽIVLJENJE« - Izjemno varnostno kritična aplikacija

- **Ogroža zdravje in življenja:** Aplikacija dostopa do ali upravlja s podatki, katerih nedostopnost ali spreminjanje bi lahko ogrozilo zdravje ali življenje ljudi. V to kategorijo spadajo podatki, kot so: medicinske kartoteke, zdravniški recepti, izvidi. Sem spadajo tudi aplikacije, ki upravljajo z medicinskimi napravami, prevoznimi sredstvi, delovnimi stroji, prometno signalizacijo, energetskimi objekti, vodnimi zajetji ipd.
- **Tajni podatki:** Aplikacija dostopa do ali upravlja s podatki, ki so po ZTP označenih z oznako "TAJNO" ali "STROGO TAJNO". Aplikacija dostopa do ali upravlja s podatki, ki so vojaška skrivnost.

## 4. DOLOČITEV OCENJENEGA APLIKACIJSKEGA NIVOJA VARNOSTI (OANV)

Ocenjen aplikacijski nivo varnosti ugotavljamo skozi določevanje skupin aktivnosti, ki jih delimo med ključne aktivnosti (»1. pogodbene zaveze«, »2. Varnostne zahteve«, »3. Varnostna arhitektura«, »4. Varnostno preverjanje«) in druge aktivnosti (»5. Ljudje«, »6. Varno kodiranje«, »7. Preverjanje varnostnih funkcij«, »8. Ključne ranljivosti«, »9. Metrike«, »10. Varnostni standardi in prakse«). Za vsako skupino aktivnosti odgovarjamo na 3 vprašanja o načrtovanih ali izvedenih aktivnostih in za odgovore prejmemo točke, ki so glede na dejanski vpliv aktivnosti na varnost obtežena (1 točka za manjši vpliv, 2 točki za srednji vpliv, 3 točke za največji vpliv).

Idealno bi bilo, če bi pri prav vsaki skupini aktivnosti (tako ključne kot druge) zbrali vsaj tako OANV oceno, da bi bila vsaj enaka, če ne večja aplikacijskemu CANV nivoju. Po zasnovi modela pričakujemo, da aplikacija za vsako skupino ključnih aktivnosti zbere vsaj toliko točk, kot je njena CANV ocena, da so aktivnosti primerno obtežene z željenim ciljnim nivojem aplikacijske varnosti.

### 4.1. Pogodbene zaveze

- **Zahteva po sodelovanju varnostnega strokovnjaka:** V pogodbi je določeno, da na projektu pri izvajalcu ves čas sodeluje izkušen varnostni strokovnjak, saj ljudje, ki razvijajo, praviloma nimajo razvite napadalske miselnosti.
- **Določitev varnostnih mejnikov in metrik:** V naročniški pogodbi so jasno zapisani varnostni mejniki in varnostne metrike. Če so varnostni mejniki združeni s projektnimi mejniki, morajo biti posebej označeni. Na ta način se določi bodoči nivo varnosti aplikacij in načine merjenja uspeha. Primeri mejnikov: varnostni test sprejemljivosti, "security push". Primeri metrik: števec ranljivosti, sDD, sDAR, krivulja odkrivanja ranljivosti ipd.
- **Odgovornost za ranljivosti:** V naročniški pogodbi je jasno zapisano, da je odpravljanje vseh ranljivosti vsaj v obdobju 1 leta vključeno v pogodbeno ceno. V

pogodbi je tudi določeno, da ima odpravljanje ranljivosti najvišjo prioriteto. Na ta način so vse pogodbene stranke pred podpisom bolj stimulirane k oceni stroškov napak.

#### 4.2. Varnostne zahteve

- **Varnostne funkcionalnosti določene:** Zahtevane varnostne funkcionalnosti so dobro definirane, zadostne in lahko uporabljive. Primeri varnostnih funkcionalnosti so vgrajevanje pametnih kartic, uporaba digitalnih potrdil, biometrija, šifriranje, upravljanje dostopov do podatkov, izdelava varnostnih kopij, anonimizacija ipd.
- **Določene revizijske funkcije:** V funkcionalnih zahtevah so izrecno zapisane obvezne funkcije beleženja dnevniških zapiskov, sledenje dogodkov, zapis dostopov in spreminjanja podatkov. Še posebej natančno je določena obdelava visoko privilegiranega dostopa ter izjeme.
- **Določeno ciljno varnostno stanje:** Za vsako podatkovno sredstvo je določen najnižji zahtevan nivo varnosti na podlagi ocene zaupnosti, celovitosti in dostopnosti podatkov ("CIA - Confidentiality, Integrity, Accesibility").

#### 4.3. Varnostna arhitektura

- **Varnostni profil:** Zahtevan je popis ključnih podatkovnih naborov (podatkovnih zbirk, datotek, registrov, sistemskih nastavitev ipd.), aplikacijskih vstopnih in izstopnih točk, komunikacijskih vmesnikov, uporabnikov z dostopi, uporabljenih varnostnih mehanizmov, željenih vgrajenih mehanizmov, integriranih drugih aplikacij ali modulov. Jasno so zapisane varnostne predpostavke.
- **Arhitekturni model groženj:** Arhitekturni model groženj ("threat model") je opisan, na več nivojih je jasno določen potek podatkov in meje zaupanja, potencialni napadi in grožnje so temeljito dokumentirani in analizirani, pripravljena so drevesa napadov in mitigacije groženj.
- **Določitev in redukcija področja napada ("attack surface"):** Zahteva se priprava določitve področja napada ("attack surface") - sistematična analiza vseh potencialnih točk napada ter pregled vseh aplikacijskih in integracijskih vmesnikov. Pričakuje se omejitev področja napada na čim manj vmesniških mest ter izdelava seznama kategoriziranih potencialnih groženj in napadov (recimo po modelu "STRIDE - Spoofing, Tampering, Repudiation, Information disclosure, DOS, Elevation of privileges").

#### 4.4. Varnostno preverjanje

- **Avtomatsko varnostno preverjanje:** Določeno je, da se izvaja avtomatsko testiranje po metodi "black box" ali da se uporabljajo orodja za avtomatsko

iskanje ranljivosti (statična ali dinamična, penetracijski preizkusi). Po vsakem avtomatskem preverjanju je nujno obvezno ročno analiziranje rezultatov.

- **Ročno varnostno preverjanje:** Avtomatska orodja najdejo predvsem znane in preproste oblike ranljivosti, ne pa tudi logičnih napak, bolj zapletenih izvedb znanih ranljivosti ter novih vrst ranljivosti. Zato je priporočljivo izvajati ročno varnostno "black box" preverjanje in ročno varnostno preverjanje kode.
- **Neodvisne poglobljene simulacije napadov:** Izvajajo se simulacije napadov usposobljenih napadalcev z namenom doseganja določenih varnostnih ciljev. Izvajajo se neodvisni preizkusi znanih vrst napadov, sistematično preverjanje znanih ranljivosti, preizkusi napadov po drevesu napadov, aplikacijsko značilni napadi, okoljsko specifični napadi ipd. Način izvajanja preizkusov je predvsem negativno varnostno testiranje.

#### 4.5. Ljudje

- **Projektni člani varnostno usposobljeni:** Vsi člani razvojne ekipe so uspešno opravili splošna usposabljanja za varno delo z računalnikom in izobraževanje redno obnavljajo.
- **Varnostno usposobljene skupine:** Posamezne skupine sodelujočih dokazujejo potrebna specialistična znanja s področja informacijske varnosti. Razvijalci in arhitekti se izobražujejo na področju varnega kodiranja, arhitekti poznajo model groženj, preizkuševalci se izobražujejo na področju iskanja znanih ranljivosti ipd.
- **Sodelovanje neodvisnega zunanjega strokovnjaka za aplikacijsko varnost:** Pri vseh fazah projekta sodeluje strokovnjak za aplikacijsko varnost, ki ni član razvojne ekipe. Njegova vloga je skrb za stalno opozarjanje na varnostne grožnje v aplikaciji.

#### 4.6. Varno kodiranje

- **Uporaba razvijalskih avtomatskih orodij:** V času razvoja je zahtevana uporaba razvijalskih orodij za nadzor posameznih modulov ali enot, s katerimi se odkriva tok napačnih podatkov. Gre za avtomatsko higiensko preizkušanje v času razvoja z uporabo znanih razvijalskih orodij - stresnega testiranja ("fuzzing testing"), preizkušanja posameznih modulov ("unit testing"), uporaba orodij za nadzor pomnilnika ipd.
- **Varno kodiranje:** Razvijalci obvladujejo osnove varnega kodiranja (izobraževanje, izkušnje) in se znajo v večini primerov izogniti TOP lestvicam ranljivosti (OWASP, WASC, SANS ipd.).

- **Načela varnega kodiranja:** Razvojne ekipe se dosledno držijo uporabe stabilnih, varnostno preizkušenih knjižnic ter pišejo kodo po lastnih načelih varnega kodiranja ("secure coding guidelines").

#### 4.7. Preverjanje varnostnih funkcij

- **Načrtovano testiranje varnostnih funkcionalnosti:** Načrt preizkušanja funkcionalnosti vsebuje obvezno preverjenje delovanja varnostnih funkcij, pa tudi preverjanje delovanja revizijskih funkcionalnosti.
- **Regresijsko testiranje varnostnih funkcionalnosti:** Po vsaki večji spremembi izdelka se zahteva uspešna izvedba regresijskega testiranja varnostnih in revizijskih funkcij, ročna ali avtomatizirana.
- **Izkušeni etični hekerji:** Varnostne funkcionalnosti in ranljivosti iščejo izkušeni varnostni strokovnjaki z napadalsko mentaliteto, ki so se v preteklosti že potrdili v iskanju ranljivosti. Preizkušanje varnosti ni naloga za začetnike ali razvijalce, ki ne razumejo napadalske mentalitete.

#### 4.8. Ključne ranljivosti

- **Iskanje TOP ranljivosti:** Sistematično iskanje in odpravljanje TOP ranljivosti (recimo OWASP, WASC, SANS/CWE TOP ipd.) ter znane ranljivosti in upoštevanje groženj v aplikacijah in modulih, ki jih integriramo. Zahtevano je dosledno verifikiranje popravkov.
- **Iskanje kompleksnih ranljivosti:** Iskanje in odpravljanje ranljivosti ter verifikiranje popravkov logičnih, kompleksnih in sofisticiranih ranljivosti. Spremljanje novih vrst napadov in napak.
- **Vrednotenje ranljivosti:** Zahtevano je ločevanje ranljivosti od drugih funkcionalnih napak in vrednotenje napak s stališča varnostnih posledic, klasificiranje ranljivosti po varnostnih parametrih ter vodenje evidence ranljivosti po splošnih (CVSS, CWE ipd.) ali lastnih klasifikacijah.

#### 4.9. Metrike

- **Preverjanje usklajenosti varnostnih zahtev glede na zakonodajo:** v sprejemnih testiranjih je zahteva po preverjanju usklajenosti z zakonodajo, ki zagotavlja varnost in zasebnost (ZEPEP, ZVOP ipd.). V sprejemnih testiranjih je zahteva po preverjanju usklajenosti s splošnimi varnostnimi standardi (kot recimo OWASP ASVS ipd.).
- **Uporaba procesnih metrik:** Za posamezne aktivnosti, povezane z varnostnimi zahtevami, je zahtevano evidentiranje porabljenega časa in sredstev. Uporabljajo



se prilagojene metrike, kot so "sDAR" (security Defect Arrival Rate), "sDD" (security Defect Density), krivulja odkrivanja ranljivosti ipd.

- Uporaba lastnih varnostnih aplikacijskih metric: Varnostne metrike in njihove željene ciljne vrednosti so določene hkrati z osnovnimi varnostnimi zahtevami.

#### 4.10. Varnostni standardi in prakse

- **Splošni varnostni standardi:** Zahteva se upoštevanje splošnih informacijsko-varnostnih standardov (ISO 2700x, COBIT, PCI DSS, SANS CAG ipd.). Čeprav splošni varnostni standardi ne dajejo natančnih napotkov za gradnjo varnih aplikacij, pomagajo dvigovati varnostno zavest sodelujočih.
- **Zrelostni aplikacijsko-varnostni modeli:** Zahteva se upoštevanje zrelostnih aplikacijskih varnostnih standardov, kot so OWASP Open SAMM, BSIMM ipd.
- **Aplikacijski varnostni standardi in prakse:** Zahteva se upoštevanje specializiranih aplikacijsko-varnostnih standardov, modelov in dobrih praks, kot so OWASP ASVS, ISO 15408, Microsoft SDL ipd, pa tudi lastnih ali branžnih pravilnikov s področja varne gradnje programske opreme (zdravstveni aplikacijski varnostno-informacijski standardi ipd.).

## 5. POVZETEK ANALIZE CANV IN OANV JAVNIH NAROČIL

Nivo ciljne aplikativne varnosti je potrebno določiti v času določanja poslovnih zahtev. Prav tako je potrebno v času naročanja izvedbe podrobno razumeti, načrtovati in naročiti ključne aktivnosti, ki so potrebne, da aplikacije in celotni sistem zgradimo na dovolj varen način. Zato smo v testne namene uporabili javne dostopne podatke javnih naročil IT projektov, saj smatramo, da so v času naročanja zelo primerni za uporabo metodologije. Ob določitvi končne cene izvedbe bi v idealnem primeru že zahteve morale vsebovati dovolj jasno definirane varnostne predpostavke, aktivnosti in varnostne cilje, saj je to edini način, da jih bodo izvajalci lahko vključili v ponudbe.

Referenčni seznam za izgradnjo seznama testnih projektov javnih naročil so predstavljale javne objave v obdobju 1.1.2009 – 30.9.2009 na spletnem portalu [www.enarocanje.si](http://www.enarocanje.si) in referenčne povezave na nadaljnjo dokumentacijo. V prvotno raziskavo smo vključili javne IT projekte, ki so v obdobju analize javno objavili naročilo, popravek naročila, objavo izbora ali preklic naročila. Smatramo, da smo na ta način v raziskavi zaobjeli širok referenčni vzorec kompleksne programske opreme, ki mora biti zgrajena primerno varno. Zaradi izjemne pomembnosti bodočega IT projekta za slovensko družbo smo na seznam vključili tudi dokumentacijo povabila k strokovnemu dialogu za projekt zVEM (december 2009).

Analiza ocen CANV je pokazala, da v povprečju javna naročila presegajo ciljni nivo 4. Ugotavljamo, da je ocena CANV visoka zaradi skoraj neizogibnega upravljanja osebnih podatkov državljanov v javni upravi. Bodoče aplikacije v javni upravi so tudi zelo

povezljive (na Internetu in Intranetu), kompleksne in so instalirane na računalnikih, ki so povezani v Internet.

Analiza ocen OANV je pokazala na aktivnosti, ki jih naročniki pri javnih naročilih naročajo najpogosteje in ki jih sploh ne naročajo. Med ključnimi aktivnostmi jih največ pozornosti posveča določanju varnostnim zahtevam (zaradi zahtev po izvedbi varnostnih in revizijskih funkcionalnosti), med drugimi aktivnostmi pa uporaba varnostnih standardov in praks.

*a) po skupinah aktivnosti*

<b>OANV aktivnosti</b>	<b>37</b>	<b>13+1</b>
<b>KLJUČNE AKTIVNOSTI</b>		
1. Pogodbene zaveze	0,5 <sup>1</sup>	1,2
2. Varnostne zahteve	1,9	2,2
3. Varnostna arhitektura	0,1	0,2
4. Varnostno preverjanje	0,0	0,1
<b>DRUGE AKTIVNOSTI</b>		
5. Ljudje	0,1	0,0
6. Varno kodiranje	0,0	0,0
7. Preverjanje varnostnih funkcij	0,0	0,1
8. Ključne ranljivosti	0,0	0,0
9. Metrike	0,2	0,4
10. Varnostni standardi in prakse	0,3	0,5

*b) po posameznih aktivnostih*

<b>OANV aktivnosti</b>	<b>37</b>
<b>NAJVEČKRAT NAROČANE AKTIVNOSTI</b>	
Varnostne funkcionalnosti določene	28
Določene revizijske funkcije	22
Uporaba splošnih varnostnih standardov	8
Preverjanje usklajenosti varnostnih zahtev z zakonodajo	7
Odgovornost za ranljivosti	6
<b>DRUGE AKTIVNOSTI</b>	
Varnostni profil	3
Zahteva po sodelovanju varnostnega strokovnjaka	2
Projektni člani varnostno usposobljeni, varnostno usposobljene skupine, načrtovano testiranje varnostnih funkcionalnosti, avtomatsko varnostno preverjanje	1

**Tabela 1: Najpogostejši odgovori OANV a) po skupinah aktivnosti; b) po posameznih aktivnostih**

Nadaljnja analiza rezultatov je pokazala, da naročniki v svoje zahteve ne zapisujejo nekaterih ključnih aktivnosti, ki bi bistveno povečala končno varnostno stanje sistema poslovne inteligence. Tako v celotnem testnem vzorcu nismo zaznali načrtovanja aktivnosti, kot so določitev varnostnih mejnikov in metrik, določeno ciljno varnostno stanje, arhitekturni model groženj, določitev in redukcija področja napada ("attack surface") in ročno varnostno preverjanje. Drugih aktivnosti, ki jih naročniki ne zapisujejo

<sup>1</sup> Povprečno število zbranih točk za področje (od 6 možnih)

v zahtevnike, je veliko, naj izpostavimo pomanjkanje dobrih varnostno-razvijalskih praks, načel varnega kodiranja, iskanje TOP ranljivosti in zahtev po sodelovanju izkušenih etičnih hakerjev.

Pričakujemo, da neodvisne poglobljene simulacije napadov naročniki izvedejo neodvisno od javnega naročila, saj jih zaradi zahtevane neodvisnosti ne more izvesti izbrani izvajalec.

## **6. ZAKLJUČEK**

Model strategije gradnje varnih sistemov poslovne inteligence predstavlja edinstveno orodje vsake razvojne ekipe ali naročnika za preprosto varnostno kategoriziranje večjega števila svojih aplikacij v sistemih poslovne inteligence in omogoča določevanje potrebnih optimalnih varnostnih aktivnosti. Skozi analizo uporabljenega projektnega testnega vzorca se je pokazal za zelo primerno orodje za obvladovanje varnostnih zahtev v zelo kompleksnih sistemih.

Ocenjujemo, da gre za edinstven model strategije za gradnjo varnih sistemov poslovne inteligence, prav tako smo z rezultati analize dobili prvi sistematični pogled na naročanje varnostnih zahtev v projektih javne uprave. Upamo, da bodo projektni rezultati naročnikom olajšali delo in vzpodbudili k še bolj doslednemu določanju pravih varnostnih zahtev in potrebnih ključnih varnostnih aktivnosti.

## **7. ZAHVALA**

Ministrstvu za visoko šolstvo, znanost in tehnologijo RS ter podjetju ACROS d.o.o. se zahvaljujem za financiranje raziskovalnega projekta.

## **8. LITERATURA**

1. PCI-DSS, <https://www.pcisecuritystandards.org/>
2. SANS Consensus Audit Guidelines, <http://www.sans.org/cag>
3. NIST's 800-53, <http://csrc.nist.gov/publications/nistpubs/800-53-Rev2/sp800-53-rev2-final.pdf>
4. OWASP SAMM in OWASP ASVS, <http://www.owasp.org>
5. BSIMM, <http://www.bsi-mm.com/>
6. ISO 15408 (Common Criteria), <http://www.commoncriteriaportal.org/>

7. ŠALAMUN, Stanka: 3 kilograme aplikacijske varnosti, prosim!, *17. mednarodna konferenca o revidiranju in kontroli informacijskih sistemov*, zbornik str. 99 – 116
8. ŠALAMUN, Stanka: Model strategije gradnje varnih sistemov poslovne inteligence, *DSI 2010*
9. ŠALAMUN, Stanka: 3 kilograme aplikacijske varnosti, prosim! (različica javna uprava), *INPRO 2010*