

# **ZAGOTAVLJANJE VARNOSTI V APLIKACIJAH ZDRAVSTVENEGA INFORMACIJSKEGA SISTEMA**

Grega Prešeren, dr. Mina Žele

Astec d.o.o.

e-pošta: grega.preseren@astec.si

URL: <http://www.astec.si>

## *Povzetek*

*V filmu The Net iz leta 1995 so nakazali na problem varovanja informacij v zdravstvenem informacijskem sistemu. Zlonamerni spremembi izvidov v filmu povzročita dve smrti. V tistem času je bilo v realnosti to praktično nemogoče. Danes Ministrstvo za zdravje izvaja projekt eZdravje, ki bo v zdravstvo vpeljal dodatne informacijske tehnologije z namenom izboljšanja zdravstvenih storitev. Zloraba zdravstvenega informacijskega sistema iz filma se zato zdi danes bolj verjetna. Vsebina članka opisuje realne tehnične možnosti zlorab današnjih aplikacij, tudi zdravstvenih, in ukrepe, s katerimi razvijalci in upravljavci aplikacij zmanjšujejo tveganja zlorab.*

## **1. UVOD**

V okviru projekta eZdravje<sup>1</sup> bo v zdravstvu vzpostavljen enoten informacijski sistem, ki bo zagotavljal učinkovit in varen prenos elektronskih dokumentov, gradiv in ostalih zdravstvenih in z zdravstvom povezanih podatkov. Enoten zdravstveni informacijski sistem bo zagotavljal elektronske storitve državljanom, zdravstvenim delavcem, izvajalcem zdravstvene dejavnosti, zavarovalnicam in upravnim organom, ki bodo z uporabo aplikacij dostopali do podatkov in jih izmenjavali. To pomeni, da se zdravstveni podatki ne bodo izmenjevali le v varnem omrežju zNET, temveč se bodo med omenjenimi akterji prenašali tudi preko Interneta, kar prinaša dodatna tveganja za njihovo varnost.

## **2. ZAKONSKE ZAHTEVE**

Zdravstveni podatki so občutljivi osebni podatki, pri obdelavi katerih je potrebno upoštevati zahteve Zakona o varstvu osebnih podatkov (ZVOP) in mnogih drugih zakonov in priporočil opredeljenih v [1]. Poleg preprečevanja nepooblaščenega dostopa do podatkov je pri zdravstvenih podatkih pomembno tudi zagotavljanje celovitosti (pravilnosti) podatkov in dosegljivosti, kar pomeni, da so podatki dostopni vedno in takoj, ko je to potrebno.

## **3. VARNOSTNA POLITIKA**

Ministrstvo za zdravje je pripravilo področne varnostne politike za izvajalce zdravstvene dejavnosti, ki opredeljujejo varnostne zahteve v vseh fazah življenjskega cikla zdravstvenih

---

<sup>1</sup> Opisana vsebina projekta eZdravje temelji na javno dostopnih dokumentih Ministrstva za zdravje, navedenih v literaturi.

aplikacij in določajo organizacijske in tehnične kontrole za zavarovanje zdravstvenih podatkov. Pri razvoju aplikacij je potrebno poskrbeti, da izvajalec vgradi ustrezne varnostne kontrole: revizijske sledi, šifriranje podatkov, mehanizme avtentikacije uporabnikov, mehanizme validacije podatkov idr. Zato se morajo varnostne zahteve opredeliti že v fazi izdelave specifikacij.

#### **4. TEHNIČNE SPECIFIKACIJE**

Tehnične varnostne zahteve, ki so opredeljene v [1], od razvijalcev rešitev pričakujejo uporabo naslednjih varnostnih mehanizmov:

- upoštevanje dobrih praks, kjer ni opredeljen vir dobrih praks;
- uporabo SSL in TLS tehnologij za šifriranje aplikacijskih podatkov pri prenosu skozi omrežje;
- šifriranje podatkov v podatkovnih bazah;
- varnostno presojo funkcionalnih specifikacij aplikacij po zaključni fazi del;
- periodično avtomatično varnostno preverjanje programske opreme s strani izvajalca del (razvijalcev aplikacije);
- revizijo varnostnih ukrepov vsaj enkrat letno s strani certificiranega revizorja;
- varovanje pred nepooblaščenimi dostopi;
- varovanje pred virusi, črvi in drugimi škodljivimi programi.

Tehnične specifikacije v [1] ne predvidevajo vseh uveljavljenih varnostnih mehanizmov, ki se uporabljajo za odkrivanje in odpravo varnostnih ranljivosti v na primer bančnih sistemih. Pomanjkljivost je pri izvajanju varnostnega preverjanja aplikacij iz dveh razlogov:

- varnostno preverjanje naj bi izvajal sam razvijalec aplikacije in ne nek tretji izvajalec, ki ni vpleten v nobeno drugo fazo;
- varnostno preverjanje naj bi se izvajalo z uporabo avtomatske varnostne programske opreme, ki ne more odkriti določenih pomanjkljivosti.

#### **5. POTENCIALNE RANLJIVOSTI**

zVEM predvideva vsaj dva portala, to sta Spletni portal za državljane in Spletni portal za zdravstvene delavce [2]. Ker je trend tudi v drugih panogah, npr. v bančnem sektorju, da večina aplikacij uporablja spletne tehnologije, verjamemo, da bodo tudi zdravstvene aplikacije tega tipa. Vsi javno dostopni portali bodo zagotovo na udaru zunanjih napadalcev. Statistika napadov pa kaže, da se večina zlorab aplikacij zgodi znotraj podjetij oz. ustanov. V zVEM so to izvajalci zdravstvenih storitev, zavarovalnice, upravni organi in ostali. Razvijalci aplikacij morajo zato poleg varnostne politike upoštevati tudi dobro prakso in priporočila pri načrtovanju spletnih aplikacij. Spletne aplikacije omogočajo določene specifične napade.

#### **6. STATISTIKA RANLJIVOSTI APLIKACIJ V SLOVENSKEM PROSTORU**

Na lestvici najpogostejših ranljivosti spletnih aplikacij »OWASP Top Ten for 2010« je na vrhu vrivanje, npr. vrivanje SQL, ki pa je vedno manj pogosto zaradi ustrezne zaščite spletnih aplikacij. Praksa v slovenskem prostoru kaže, da so najbolj pogoste ranljivosti spletnih aplikacij nepravilna avtorizacija dostopov do podatkov, neustrezna avtentikacija uporabnika, XSS (*cross-site scripting*), ne optimalno upravljanje uporabniške seje ter neustrezna konfiguracija in posodobljenost HTTP oz. aplikacijskih strežnikov. Vse te ranljivosti so uvrščene tik pod vrh omenjene lestvice. Ker pomanjkljivosti in posledično ranljivosti spletnih

aplikacij vedno obstajajo, jih je običajno mogoče izrabiti, kljub temu da je varnostna politika ostalega IS ustrezno implementirana. Posledice izrabe teh ranljivosti so lahko neavtoriziran dostop do podatkov, izvajanje akcij v imenu drugega uporabnika, poneverjanje podatkov itd. V eZdravju to pomeni dostop do zdravniških kartotek, spreminjanje diagnoz, poneverjanje napotitve k zdravniku specialistu idr., kar ima lahko za posledico ogrožanje zdravja in življenja državljanov.

## 6.1 Nepravilna avtorizacija

Ranljivost nepravilne avtorizacije omogoča legitimnemu uporabniku aplikacije, da z uporabo določenih trikov dostopa do podatkov, za katere nima avtorizacije. V praksi to pomeni, da se uporabnik prijavi v aplikacijo s svojim veljavnim uporabniškim imenom in geslom, in z uporabo trikov dostopa do podatkov drugega uporabnika aplikacije. V zdravstvu to pomeni razkrivanje občutljivih osebnih podatkov o pacientih. Nekateri triki, ki to omogočajo so npr. spremembe podatkov, ki jih spletni brskalnik pošlje spletnemu strežniku v HTTP zahtevah. Ti podatki so lahko GET oz. POST parametri, vrednosti piškotkov (*cookies*) ipd. Spremembo teh podatkov omogočajo prestrežniška orodja (*interception proxy*), ki jih lahko sami namestimo na delovno postajo.

## 6.2 Neustrezna avtentikacija

Primeri neustrezne avtentikacije sta npr. t.i. BASIC in DIGEST HTTP avtentikaciji, ki sta definirani skupaj s protokolom HTTP. Ključna pomanjkljivost obeh je, da se uporabniško ime in geslo vedno pošiljata v HTTP glavi in možnost napada s surovo silo (*brute force*). Pomanjkljivosti, ki se pogosto pojavijo pri drugih avtentikacijskih shemah so:

- da se uporabniški račun ne zaklene po določenem številu nepravilnih prijav, kar omogoča napad s surovo silo;
- da se ob pravilno vpisanem uporabniškim imenom in napačnim geslom izpiše sporočilo oblike "napačno geslo", kar omogoča ugibanje uporabniških imen s surovo silo;
- da se pri uporabi X.509 certifikata za avtentikacijo na nivoju SSL in TLS ne preverja konkretni uporabniški certifikat (CN), pač pa le certifikatna agencija (CA), ki je izdala certifikat (*Issuer*);
- da ni povezave (uparitve) med X.509 certifikatom in uporabniškim imenom in geslom, kar omogoča prijavo v nek uporabniški račun z uporabo certifikata tretje osebe, ki mora biti izdan s strani iste certifikatne agencije.

## 6.3 XSS

XSS (*cross-site scripting*) omogoča, da se škodljiva koda izvede v brskalniku uporabnika aplikacije, tipično je to javascript koda. Javascript koda omogoča, da napadalec izvede naslednje akcije:

- ukrade identifikator uporabniške seje, t.i. piškotek, kar mu omogoča vpadanje v uporabniško sejo brez posedovanja uporabniškega imena in gesla – napadalec lahko uporablja aplikacijo v imenu tistega, ki mu je ukradel piškotek;
- ugotovi na katere spletne naslove (URI) zahaja uporabnik aplikacije, kar napadalcu omogoča profiliranje uporabnika in pripravo smiselnih nadaljnjih scenarijev napadov;
- preusmeri uporabnika iz aplikacije na nek drug spletni naslov, kar omogoča ribarjenje (*phishing*);
- odpre nov zavihek v brskalniku z namenom ribarjenja;
- in še mnogo drugih škodljivih akcij, tudi v povezavi z orodjem Metasploit.

Že nekaj časa sta znani dve obliki XSS napadov, to sta t.i. *persistent* in *reflected* XSS. Oba je na nek način mogoče zaznati na strežniški strani, tudi v log datotekah, kjer bi se videla vrinjena škodljiva koda. V zadnjem času pa se je pojavila še tretja oblika XSS, to je t.i. DOM XSS oz. tudi DOM-based XSS. Gre za posebno obliko reflected XSS, ki pa je ni mogoče zaznati na strani strežnika. To pomeni, da vzdrževalec aplikacije tega napada ne bo nikoli zaznal. Možnost tega napada je mogoče odkriti le z zunanjim varnostnim pregledom aplikacije oz. varnostnim pregledom izvirne kode aplikacije.

#### **6.4 Neoptimalno upravljanje uporabniške seje**

Neoptimalno upravljanje uporabniške seje se nanaša na upravljanje z identifikatorjem uporabniške seje, ki je običajno piškotek. Le tega je potrebno spremeniti vedno, ko se uporabnik na novo prijavi v aplikacijo in ga razveljaviti, ko se uporabnik odjavi iz aplikacije. Nekaterne aplikacije odjave sploh ne omogočajo, kar pomeni, da veljavna uporabniška seje ostane aktivna dokler ne poteče določen čas. Prav tako je priporočljivo, da se uporabniška seja prekine po določenem času neaktivnosti uporabnika. Če ta priporočila niso upoštevana, obstaja bistveno večja verjetnost vpadanja v uporabniško sejo, kar pomeni možnost uporabe aplikacije v imenu nekega veljavnega uporabnika brez poznavanja njegovega uporabniškega imena in gesla.

#### **6.5 Neustrezna konfiguracija in posodobljenost strežnikov**

Če strežniška programska oprema ni posodobljena, obstajajo neodpravljene splošno znane ranljivosti. Običajno na svetovnem spletu obstajajo opisani postopki in orodja, ki znajo izrabit te ranljivosti z namenom pridobitve administrativnega dostopa do strežnika oz. onemogočanja storitev (*denial of service* – DOS). Najpogostejši primer neustrezne konfiguracije strežniške programske opreme je neustrezna konfiguracija SSL in TLS protokolov. Priporočljivo je, da sta omogočena le SSL 3.0, TLS 1.0 oz. višje verzije omenjenih protokolov. Pri tem mora biti onemogočena uporaba šibkih šifirnih algoritmov (npr. DES), šibkih t.i. *hash* algoritmov (npr. MD5) in neustrezno kratkih šifirnih ključev (pod 128-bit). Onemogočena mora biti tudi uporaba t.i. anonimnega SSL. Če konfiguracija ni ustrezna je pri prisluškovanju prometa (npr. z uporabo ARP zastrupljanja) potencialno možno dešifrirati promet.

### **7. PREVENTIVNI UKREPI**

Preprečevanje pojava navedenih in preostalih ranljivosti omogoča le dosledno upoštevanje vzpostavljenih varnostnih politik, ustrezno načrtovanje razvoja aplikacij, uporaba standardnih protokolov in močnih šifirnih algoritmov, uporaba uveljavljenih programerskih orodij in okvirjev (*web application frameworks*), "varno" programiranje (kodiranje), kjer se morajo programerji zavedati aktualnih ranljivosti, ter upoštevanje priporočil dobre prakse s strani različnih združenj, kot so OWASP, CERT itd. Kljub upoštevanju vseh priporočil se iz različnih vzrokov med razvojem aplikacije vedno pojavijo ranljivosti. Pred vpeljavo aplikacije v produkcijsko okolje je zato potrebno opraviti tudi varnostni pregled aplikacije, s katerim zunanji izvajalec (to je nekdo, ki ni sodeloval pri načrtovanju in razvoju aplikacije) odkriva ranljivosti in scenarije zlorabe ranljivosti, oz. opraviti tudi varnostni pregled izvirne kode. Vse odkrite ranljivosti je potrebno odpraviti in potrditi njihovo odpravo z verifikacijskim varnostnim pregledom. Na ta način lahko zagotovimo, da je aplikacija varna za uporabo, možnosti zlorabe pa so minimalne.

Druga vrsta preprečevanja zlorab aplikacij pa je izobraževanje končnih uporabnikov aplikacij. Večina zlorab aplikacij ne temelji na tehničnih pomanjkljivostih in ranljivostih aplikacij, temveč na nepazljivi uporabi. Najbolj znana taka primera sta t.i. socialni inženiring (*social engineering*) in ribarjenje (*phishing*). Uporabnike je potrebno informirati o takšnih vrstah napadov, tako da prepoznajo napad, še preden se zgodi, oz. da prepoznajo posledice, ko se napad že zgodi.

## **8. ZAKLJUČEK**

Ministrstvo za zdravje je z varnostnimi politikami pripravilo ustrezen okvir za izvedbo varnega in učinkovitega informacijskega sistema eZdravje. Izvajalci zdravstvene dejavnosti pa morajo varnostno politiko upoštevati in vpeljati. Pri razvoju aplikacij je to še posebej pomembno, saj bodo aplikacije vstopna točka za dostop do občutljivih osebnih podatkov za širok krog različnih uporabnikov. Prav tako je potrebno ustrezno informirati uporabnike o pravilni uporabi aplikacij.

## **LITERATURA**

1. [www.mz.gov.si/fileadmin/mz.gov.si/pageuploads/eZdravje/Strokovni\\_dialog/Ogrodje\\_zVEM\\_-\\_Specifikacija\\_zahtev\\_osnutek\\_3.0\\_za\\_strokovni\\_dialog.pdf](http://www.mz.gov.si/fileadmin/mz.gov.si/pageuploads/eZdravje/Strokovni_dialog/Ogrodje_zVEM_-_Specifikacija_zahtev_osnutek_3.0_za_strokovni_dialog.pdf)  
Vzpostavitev ogrodja zVEM, Tehnične specifikacije.
2. [www.mz.gov.si/fileadmin/mz.gov.si/pageuploads/eZdravje/Strokovni\\_dialog/eZdravje\\_\\_Ogrodje\\_zVEM\\_Strokovne\\_podlage\\_04052010.pdf](http://www.mz.gov.si/fileadmin/mz.gov.si/pageuploads/eZdravje/Strokovni_dialog/eZdravje__Ogrodje_zVEM_Strokovne_podlage_04052010.pdf)  
Vzpostavitev ogrodja nacionalnega zdravstveno informacijskega sistema zVEM, Strokovne podlage.