"Innovations in Science and Technologies" scientific electronic journal
ISSN: 2181-321
www.innoist.uz
Volume 1
Issue № 3
APRIL, 2024

VEB SERVERLARDA ZAIFLIKLARNI ANIQLOVCHI SAMARALI ALGORITMLAR

Nasrullayev Nurbek Baxtiyorovich
Muhammad al-Xorazmiy nomidagi Toshkent
Axborot texnologiyalari universiteti
PhD dotsent
Barotova Zahro Akmaljon qizi
Muhammad al-Xorazmiy nomidagi Toshkent
Axborot texnologiyalari universiteti
Gmail:zbaratova0402@gmail.com

Annotatsiya: Veb ilovalar tobora rivojlanib bormoqda va hayotning ko'p jabhalarida qo'llanilmoqda. Biroq, SQL in'yektsiyasi va saytlararo skript kabi turli xil xavfli veb zaifliklari mavjud. Bu xakerlarga tijorat yoki siyosiy maqsadlarda yoki shon-shuhrat uchun veb saytlardan foydalanish va hujum qilish imkoniyatini yaratadi. Ushbu zaifliklarni skanerlash va aniqlash uchun ba'zi tadqiqot va tijorat dasturlari ishlab chiqilgan. Ushbu maqolada biz veb xavfsizlik zaifliklarini aniqlash uchun samarali algoritmlarni taqdim etamiz.

Kalit so'zlar: veb zaifliklar, SQL in'yeksiyasi, saytlararo skript, mashinali o'qitish, aniqlash algoritmi

I. Kirish

Veb brauzer mijozning eng asosiy ilovasi bo'lib, u foydalanuvchilar uchun veb sahifalarni ko'rib chiqishning eng samarali usuli hisoblanadi. Hozirgi vaqtda veb brauzer ko'p sonli bozorlarni egallash uchun asosan ko'rish va ma'lumotlarni yuklab olishdan foydalanadi. Shu bilan birga, u hukumat, korxonalar, elektron tijorat platformasi, ijtimoiy tarmoq va boshqa sohalarni ham qamrab oladi[1]. Veb ilovalarning keng qo'llanilishi va rivojlanishi bilan uning funksiyasi va interaktivligi

ham yaxshilanmoqda. Hozirgi vaqtda veb ilovalar xavfsizligi tizimi asosan server tomonidagi xavfsizlik tizimi uchun mo'ljallangan, ammo mijoz tomoni uchun veb ilovalar xavfsizligi siyosati va mexanizmi kamroq. Shu sababli, mijozga asoslangan zaiflikni buzishni aniqlash texnologiyasi kelajakda tarmoq xavfsizligi sohasidagi tadqiqot nuqtasiga aylandi. Tarmoq xavfsizligi tarmoqni uzatish protokoli, serverning qaytish qiymatiga javobi, ma'lumotlar so'rovi va boshqa jihatlarni o'z ichiga oladi. Shunday qilib, har bir zaiflikning ma'lumotlarni uzatish va o'zaro ta'sir qilish jarayonida veb ilova mijoz ilovalari veb xavfsizlik muammolarini keltirib chiqaradi[2]. Kamchilikli jihat dizaynerning apparat, dasturiy ta'minot, protokolni amalga oshirish yoki model xavfsizligi siyosatidagi yetishmasligi. Xakerlar, shuningdek, ruxsatsiz modellarni to'g'ridan-to'g'ri yo'q qilishlari mumkin. Umumiy kamchilik shundaki, dizaynerlar loyihalashdan oldin bu haqida o'ylamaydilar. Veb brauzer mijoz ilovasining zaifligi ham ma'lumotlar yo'qolishining muhim sababidir. Shunga asoslanib, mashinali o'qitish prinsipi bilan birgalikda, veb ilovaning zaifliklarni aniqlash usuli dastur xavfsizligini yaxshilash uchun optimallashtirilgan.

II. Veb zaifliklar

Quyidagi 1.1-jadvalda veb ilova zaifliklari to'rt toifaga bo'lingan - so'rovlarni boshqarish, mijoz tomonidan in'yeksiya, fayl va path in'yeksiyalari va buyruqlar in'yeksiyalari keltirilgan[3].

1.1-jadval

Veb zaiflik toifalarida zaifliklarning sinflanishi

Zaiflik toifasi	Umumiy tushuncha	Zaiflik sinflari
So'rovlarni boshqarish	Ma'lumotlar bazalari kabi ma'lumotlarni saqlaydigan tuzilmalar bilan bog'liq zaifliklar va bu yerda zararli kod so'rovlarni	SQL in'yeksiyalari XPath in'yeksiyalari LDAP in'yeksiyalari NoSQL in'yeksiyalari

	manipulyatsiya qiladi va ularni o'zgartiradi	
Mijoz tomonidan in'yeksiyalar	Zararli maqsadlar bilan bog'liq zaifliklar mijoz tomonidan kiritilgan kod, masalan javaScript, va server tomonidan qayta ishlanadi	Cross-site scriping Email in'yeksiyalari Spamming Header in'yeksiyalari
Fayl va path in'yeksiyalari	Tegishli ravishda nisbiy yo'llar yoki fayllarni boshqaradigan zaifliklar sinfi: boshqa joyga yoki kirishga yo'naltirish mahalliy tizim va veb-ilova fayllari	Fayllarni masofaviy kiritish Mahalliy faylni kiritish Katalog/yo'lni o'tkazish
Buyruq in'yeksiyalari	Fayl tizimi buyruqlari va PHP ko'rsatmalarini kiritish orqali foydalaniladigan zaifliklar	OS kommanda in'yeksiyalari PHP kod in'yeksiyalari

SQL in'yeksiyasi veb zaiflik bo'lib, foydalanuvchilar kiritgan ma'lumotlarning qat'iy nazorati yo'qligi sababli yuzaga keladi, buning natijasida hakerlar buyruqlarni bajara olishmaydi. SQL in'yeksiyasi hujumi SQL so'rovi bayonotini soxtalashtirish uchun so'rovni kiritish orqali ushbu zaiflikdan foydalanadi, shu bilan ma'lumotlar bazasini o'g'irlanishi va yo'q qilinishi kabilar kelib chiqadi[4].

XSS hujumi – veb ilovalarda keng tarqalgan hujum bo'lib, veb sahifalarda zararli kodni (mijoz tomonidagi saytga) joylashtiradi. XSS g'oyasiga ko'ra, tajovuzkorlar brauzerga skriptlarni joylashtirishi mumkin bo'ladi va bu orqali foydalanuvchining sessiyasi qabul qilinadi, veb saytlar buziladi, ichki tarmoqlarning port tekshiruvini qo'lga kiritadi[3].

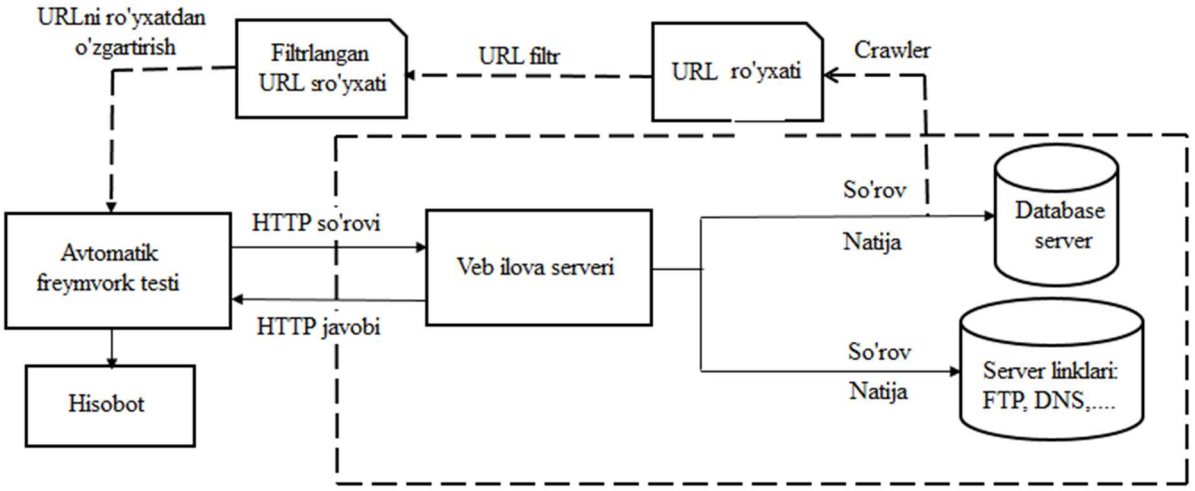
Buferning to'lib ketishi (BoF) - kirish ma'lumotlari stekning vaqtinchalik saqlash hajmidan oshib ketganda yuzaga keladigan xato. Bu, odatda, foydalanuvchi dastur serveriga katta hajmdagi ma'lumotlarni yuborganida va keyin bu katta hajmdagi ma'lumotlarga toksik in'yeksiya hujumi sodir bo'lganda sodir bo'ladi[4].

Faylni kiritish (FI) xavfli zaiflik bo‘lib, xakerlarga veb serverdagi maxfiy fayllarga ruxsatsiz kirish imkonini beradi, zararli fayllarni include() funksiyasiga qo‘ng‘iroq qilish orqali ishga tushiradi. Ushbu ekspluatatsiya texnikasining asosini boshqa dasturlash tillaridagi include(), require(), ... kabi fayllarni qo‘shish funksiyalari va shunga o‘xshash funksiyalar tashkil etadi. Masofaviy faylni kiritish (RFI) zaifligi xakerlarga masofaviy faylni maqsadli xostga kiritish va bajarish imkonini beradi. Xakerlar RFIdan foydalanuvchining mashinasida ham, serverida ham zararli kodni ishlatishi mumkin. Ushbu turdagi hujumning ta’siri seans tokenlari yoki foydalanuvchi ma’lumotlarini vaqtincha o‘g‘irlashdan tortib, butun server tizimini yo‘q qilish uchun veb qobiqlar yoki zararli kodlarni yuklashgacha o‘zgarib turadi[5].

III. Taklif etilayotgan algoritmlar

Ushbu taqdiqot ishida veb saytlardan ma’lumotlarni yig‘ish uchun veb crawler algoritmlari taklif qilinadi. Kirish ma’lumotlari sifatida manba URL manzillari ro‘yxati olinadi. Ular skanerdan o‘tkaziladi va pastki URL manzillarini topadi, keyin ularni ko‘rib chiqish uchun manzillar ro‘yxatiga qo‘shadi. Jarayon butun havola ro‘yxatga olinmaguncha va saqlanmaguncha takrorlanadi. Brauzer jarayoni veb sayt va uning mazmuni, jumladan veb sayt URL manzili, sarlavha, meta teg, sahifa mazmuni va havolalar haqida ma’lumot to‘playdi. Qaytarilgan natijalar qidiruv tizimi tomonidan belgilanadi va qayta tartibga solinadi. Yig‘ilgan havolalar xavfsizlik zaifliklarini aniqlash jarayoniga kiritiladi. Umuman olganda, bunda veb-serverdan testlash va qayta aloqa usulidan foydalanamiz. Veb sayt sifatida kiritish bilan brauzer jarayoni orqali URL manzillari haqida ma’lumot yig‘iladi. Tegishli URL filtrlari har bir zaiflik bilan yaratilgan. Ushbu filtrlar kirish ma’lumotlarining URL manzillarini tasniflash va qisqartirish uchun foydalanuvchi ilova bilan o‘zaro aloqada bo‘lgan maydonlar, shakllar, parametrlar va o‘zgaruvchilar ma''lumotlaridan foydalanadi. Avtomatik freymvork sinov algoritmi veb ilovadan HTTP so‘rovi va HTTP javobiga asoslangan yondashuvdan foydalanadi. Kiruvchi

so‘rovlar so‘rov vazifalarini bajaradi va veb server, SQL server yoki FTP server kabi veb sayt komponentlaridan fikr mulohaza oladi. Interaktiv jarayonlar saqlanadi va qayta ishlanadi. Ushbu freymvork natijalarni avtomatik ravishda sinab ko‘rish, qabul qilish va solishtirish, zaiflik mavjudligini aniqlash uchun qiymatlarni qaytarish uchun yaratilgan. Natijalar qayd qilinadi va tegishli ma’lumotlar bilan yakuniy hisobot qilinadi (1-rasm).



1-rasm. Zaifliklarni aniqlash algoritmi umumiy diagrammasi

Ma’lumotlar bazasini optimallashtirish uchun mashinali o‘qitish texnikasini qo‘lladik. Ma’lumotlar bazasi zaifliklarni aniqlash uchun parametr sifatida ishlatiladigan ma’lumotlar massivida saqlanadigan maxsus belgilar, foydali yuk segmentlari, zaifliklardan foydalanish satrlari, ... (payload deb ataladi) dan iborat. Dasturiy ta’minot vositasi vaqt samaradorligiga erishish uchun ma’lumotlar bazasi quyidagicha optimmalashtirildi:

1-qadam	Ma’lumotlar bazasidagi (ma’lumotlar massivi) a_i massivdagi $payload_i$ni n ta *payload* elementi bilan chaqirish.

2-qadam	Har bir a_i ga k ning ustuvor vazni beriladi. k qanchalik katta bo‘lsa, hujum turlarida yuk/belgi shunchalik keng tarqalgan va afzalroq foydalaniladi; massiv k vaznining kamayish tartibida joylashtirilgan. a_i elementlarini baholash ko‘plab nuqsonli veb saytlar bilan tajriba jarayoni asosida amalga oshirildi, ulardan biz mashinaga eng ko‘p ishlatiladigan a_i ni baholashga yordam berish uchun o‘qituvchili mashinali o‘qitish usullaridan foydalanamiz; foydalanilmagan a_i massivdan (yoki ro‘yxat) o‘chiriladi.
3-qadam	3-bosqichda input uchun a_i elementini ketma-ket qidirish algoritmi bo‘yicha chiqarib oling, katta k bo‘lgan a_i ustunlik qiladi. Bu yerda qo‘llaniladigan ketma-ket qidiruv algoritmi haqiqatan ham samarali, chunki a_i k vazniga ega va k ga muvofiq kamayish tartibida tartiblangan.
4-qadam	Ma’lumotlar bazasini yangilang. Mashinali o‘qitishni qo‘llash skanerlash natijalari asosida amalga oshiriladi va bunda har doim ma’lumotlar bazasi yaxshilanib va optimallashtirilib boriladi. Algoritm qaysi foydali yuklardan ko‘proq foydalanilishini, zaifliklarni aniqlash ehtimoli yuqori ekanligini aniqlaydi, zaifliklarni aniqlash algoritmida qaysi foydali yuk ishlatilganligini tekshirib, ma’lumotlar bazasini yangilaydi. Algoritmning input qismidagi payloadlar vazni kam bo‘lishi va boshqa payloadlar ham qo‘shilib borilishi mumkin.

XSSni aniqlash algoritmi

XSS xususiyatlaridan kelib chiqib, ularni aniqlashning yangi algoritmi taklif qilinadi. Algoritm XSS zaifliklarini aniqlash algoritmining operatsion tafsilotlarini ko‘rsatadi.

1-algoritm. *XSS zaifliklarini aniqlash*

1	Input: veb-saytdan URL manzillar, XSS yuki
2	Output: XSS zaifliklarini o‘z ichiga olgan URL manzillar to‘plami
3	Brauzer jarayoni va ma’lumotlar filtri orqali veb saytdan URL manzillarni oling.
4	XSS xato sinov funksiyasidan foydali yukni tekshiradigan XSS ro‘yxatini ishga tushiring, foydali yukni $payload_1$ dan $payload_n$ gacha raqamlang.

5	$URL_{NEW} = URL_{OLD} + payload_i$ bilan so‘rovlarni serverga qayta yuboring.
6	Server so‘rovlarga javob berishini tekshiring. Ha bo‘lsa, bu XSS zaifligi boshlang‘ich URL manzilida mavjudligini anglatadi, natijani yozib oling va tekshirish jarayonini yakunlang. Agar yo‘q bo‘lsa, i = i + 1 ni oshirishda davom eting; 5-qadamga qayting va ro‘yxatdagi barcha yuklarni bajaring.

SQL in’yeksiyasini aniqlash algoritmi

SQLi zaifliklarni aniqlash algoritmi xuddi shunday tarzda ma’lumotlar bazasidan foydalanish uchun maxsus kodlarga "in’yeksiya" usullaridan foydalanadigan SI zaifligini va zaifliklarni aniqlash uchun ishlatiladi. 2-algoritm SQLi zaifliklarini aniqlash algoritmining operatsion tafsilotlarini ko‘rsatadi.

***2-algoritm**. SQLi zaifliklarini aniqlash*

1	Input: veb sayt URL manzillari, maxsus belgilar ro‘yxati
2	Output: SQLi zaifliklarini o‘z ichiga olgan URL manzillar to‘plami
3	Crawler jarayoni va filtr kiritish URL manzili orqali veb saytdan URL manzillarni oling.
4	SQLi xatolarni qayta ishlash funksiyasidan SQLini sinovdan o‘tkazuvchi maxsus belgilar va skriptlar ro‘yxatini ishga tushiring; belgilarni i = 1 dan n gacha raqamlang.
5	$URL_{NEW} = URL_{OLD} + Character_i$ bilan so‘rovlarni serverga yuboring.
6	Agar server so‘rovlarga javob bersa, bu boshlang‘ich URL manzilida SQLi zaifligi mavjudligini anglatadi, natijani yozib oling va jarayonni yakunlang. Agar yo‘q bo‘lsa, i = i + 1 ni oshirishda davom eting; 5-qadamga qayting va ro‘yxatdagi barcha belgilarni ko‘rib chiqing

Buferning to‘lib ketishini aniqlash

Bufer to‘lib ketishi uchun qayta aloqa darajasini baholash uchun tizimning sezgirligiga qaraganda ko‘proq kirish ma’lumotlarini taqdim etish orqali zaifliklarni aniqlash uchun qora quti testidan foydalanamiz.

Ma’lumotlar to‘lib ketgan ba’zi hollarda, kiritilgan zararli kodlar ishga tushirilgan yoki yo‘qligini tekshirish kerak. Ushbu ishda qilingan yaxshilanishlardan biri xato holatidagi va normal holatdagi server javoblarini tasdiqlash va solishtirish

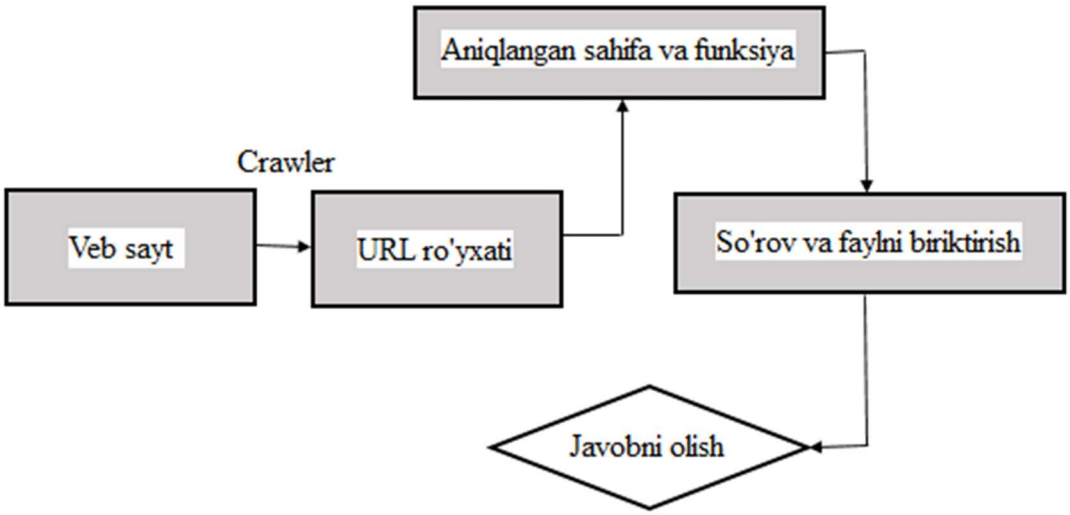
uchun tuzatuvchidan foydalanishdir. 3-algoritm BoF zaifliklarini aniqlash algoritmining operatsion tafsilotlarini ko'rsatadi.

***3-algoritm**. BoF zaifliklarini aniqlash*

1	Input: veb sayt URL manzillari
2	Output: BoF zaifliklarini o'z ichiga olgan URL manzillari to'plami
3	Veb saytning tuzilishi testlash jarayoni uchun URL manzillar to'plamini crawlerdan olish orqali amalga oshiriladi.
4	Veb saytga joylashtirilgan parametrlarni, o'zgaruvchilarni, tarkibni yoki oqimlarni bajarishni filtrlash orqali bufer to'lib ketishiga imkon beruvchi joylarni aniqlang va filtrlang. Bu jarayon algoritmni tezlashtiradigan test uchun URL manzillar hajmini kamaytiradi.
5	Qora quti testidan foydalanib, serverning javob berish qobiliyatini sinab ko'rish uchun o'z navbatida yetarli hajmdagi yoki nostandart formatdagi ma'lumotlar paketlarini yuboring. Ba'zi hollarda zaiflik aniqlanadi, ammo zararli kod bajarilmaydi. Tuzatish vositasi bu muammoni hal qilish uchun ishlatiladi, bu bufer to'lib ketishini ishga tushirganda ijro etuvchi oqim va server holatini tekshiradi.
6	Qora quti sinovidan olingan test natijalari va tuzatuvchi to'planadi va xabar qilinadi.

Sezuvchanlik indeksi (SI) va Fayl kiritish (FI)ni aniqlash

FI va SI zaifliklari tajovuzkorga ularni serverga zararli fayllar yoki qobiqlarni bajaradigan zaifliklarni kiritish imkonini beradi. 2-rasmda skanerlar modeli va usuli batafsil tasvirlangan.



2-rasm. Veb serverda FI va SI zaifliklarini aniqlash bosqichlari

4-algoritm FI va SI zaifliklarini aniqlash algoritmining operatsion tafsilotlarini ko‘rsatadi.

4-algoritm *FI va SI zaifliklarini aniqlash.*

1	Input: veb sayt URL manzillar
2	Output: FI yoki SI zaifliklarini o‘z ichiga olgan URL manzillar to‘plami
3	Tekshirish jarayoni veb sayt tuzilishini oladi.
4	Login.php, cart.php va hokazo kabi zaifliklarni o‘z ichiga olishi mumkin bo‘lgan manzillarda filtrlashni amalga oshiring. Ushbu filtrlash bosqichi tekshirilishi kerak bo‘lgan manzillar sonini kamaytiradi.
5	Yaratilgan testlash ma’lumotlari to‘plami bilan, o‘z navbatida, tizim javobini sinab ko‘rish uchun ekspluatatsiya kodi va qobiq kodini o‘z ichiga olgan so‘rovlarni yuboring. Sinov funksiyasi manba kodidagi include(), require() kabi funksiyalarni yoki yuklangan qobiq kodlari uchun serverning javob berish qobiliyatini sinash uchun mo‘ljallangan.
6	Javoblar natijalarini yozib oling va hisobotni eksport qiling.

IV. Xulosa

Ushbu maqolada SQL in’yeksiyasi, XSS, BoF, FI, SI kabi keng tarqalgan veb xavfsizlik zaifliklarini ko‘rib chiqdik. Shunday qilib, biz veb ilovaning zaif tomonlarini aniqlash samaradorligini oshirish uchun algoritm va takomillashtirishni taklif qilamiz.

Shunga asoslanib, ushbu maqola mashinali o‘qitishga asoslangan veb ilovalar uchun zaiflikni aniqlash algoritmlarini taklif qiladi. U ariza maydonlari yoki URL parametrlari yordamida dastur zaifliklarining xususiyatlarini tasniflaydi va xususiyatlarni aniqlash natijalariga ko‘ra zaifliklarni topadi va aniqlaydi. Shu bilan birga, qo‘lda aniqlashning zerikarli va qoldirilishiga yo‘l qo‘ymaslik uchun mashinali o‘qitish prinsipi bilan birlashtirib, aniqlash bosqichlarini soddalashtiradi. Ammo veb ilovalarda hali ham ba’zi xavfsizlik zaifliklari mavjud bo‘lib, ularni

to'liq aniqlash qiyin. Shuning uchun zaifliklarni aniqlash usullarini qo'llash haqiqiy tarmoq xavfsizligi tamoyillari bilan birlashtirilishi kerak.

Foydalanilgan adabiyotlar:

1. Ali, N.S. (2018) 'Investigation framework of web applications vulnerabilities, attacks and protection techniques in structured query language injection attacks', *International Journal of Wireless and Mobile Computing*, Vol. 14, No. 2, pp.103–122, DOI: 10.1504/IJWMC.2018. 091137

2. Tan B , Elnaggar R , Fung J M , et al. (2020) Towards Hardware-Based IP Vulnerability Detection and Post-Deployment Patching in Systems-on-Chip[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99):1-1.

3. Antunes, N. and Vieira, M. (2011) 'Defending against web application vulnerabilities', *Computer*, Vol. 45, No. 2, pp.66–72, DOI: 10.1109/mc.2011.259

4. Yi M , Xu X , Xu L . (2019) An Intelligent Communication Warning Vulnerability Detection Algorithm Based on IoT Technology[J]. IEEE Access, 7(99):164803-164814.

5. Hoang Viet Long and Tong Anh Tuan. (2020) An efficient algorithm and tool for detecting dangerous website vulnerabilities, Int. J. Web and Grid Services, Vol. 16, No.1.