Mongolian Journal of Engineering and Applied Sciences, 2021, 3(1)

Компьютерын ухаан

ВЕБИЙН АЮУЛГҮЙ БАЙДЛЫН ШИНЖИЛГЭЭНИЙ СИСТЕМИЙН ХӨГЖҮҮЛЭЛТ

Б.Цолмонтамир[1,*], Б.Өсөхбаяр[2],Н.Угтахбаяр[2]

[1]МУИС, Мэдээлэл технологийн газрын сүлжээ холбооны газар,
[2]МУИС, ХШУИС, Электроник, холбооны инженерчлэлийн тэнхим

Received on 2021.04.13; Revised on 2021.06.24; Accepted on 2021.07.01

*Холбоо баригч зохиогч: tsolmontamirb@num.edu.mn.

Хураангуй
Веб системд суурилсан програм хангамж, үйлчилгээний тоо сүүлийн жилүүдэд асар хурдацтай нэмэгдэж байна. Энэхүү хурдтай өсөлтөөс үүдэн вебийн аюулгүй байдал нь бүрэн хангагдаагүй системүүд нэмэгдэх болж, түүний хирээр вебийн эмзэг байдалд тулгуурласан халдлага дайралтууд улам бүр ихсэж байна. Вебийн аюулгүй байдлыг хамгаалахтай холбоотой арга хэрэгсэл ашиглаагүй, аюулгүй байдлыг хангах хангалттай мэдлэггүй хүн программын кодыг бичсэн, баталгаажуулалт байхгүй эсвэл баталгаажуулалтын арга, аргачлал, хэрэглээний алдаа зэрэг олон шалтгаанаас вебийн эмзэг байдал үүсэж байна. Энэ алдаа дутагдлыг багасгаж, веб системийн аюулгүй байдлыг сайжруулах замаар хэрэглэгч болон мэдээллийн аюулгүй байдлыг хангах хэрэгцээ шаардлага зайлшгүй урган гарч ирж байна. Бид энэ судалгааны ажлаар вебийн эмзэг байдлыг тодорхойлох системийн хөгжүүлэлтийг хийж гүйцэтгэсэн. Энэхүү систем нь вебийн эмзэг байдлыг илрүүлэх wapiti, w3af, ZAP, Vega зэрэг бусад програмуудтай харьцуулахад ажиллагааны хувьд ойлгомжтой, хялбар, бүтцийн хувьд энгийн, бусад системүүдтэй интеграц хийх боломжтой, нээлттэй эх кодтой, цаашид нэмж хөгжүүлэх боломжтой, түгээмэл хэрэглэгддэг програмчлалын пайтон хэлийг ашигласан, график интерфейс бүхий цогц систем юм. Судалгааны үр дүнгээс харахад бидний боловсруулсан систем үр дүнгээрээ бусад төстэй системүүдтэй ижил түвшинд ажиллаж, вебийн эмзэг байдлыг тодорхойлох боломжтой байна.

Түлхүүр үг:web application security scanning, web security assessment, enumeration, injection

1 Удиртгал

Аливаа системд халдлага үйлдэхэд тухайн системийн талаарх үнэн зөв мэдээллийг цуглуулах, тухайн мэдээлэл дээр тулгуурлан халдлага хийх боломжийг эрэлхийлэх нь ёс зүйт хакер болон бусад халдлага үйлдэгчдийн хийдэг үндсэн үйлдэл юм. Вебийн эмзэг байдал гэдэг нь халдагч этгээд зөвшөөрөлгүй нэвтрэхээс эхлээд нууц, хаалттай, байгууллагын дотоод хэрэгцээний зэрэг чухал мэдээлэл олж авах улмаар байгууллагад хохирол учруулахуйц аливаа үйлдэл хийхийг хэлэх бөгөөд вебийн аюулгүй байдлыг муу хангаснаар энэхүү боломжийг халдагч этгээдэд олгодог. Сүүлийн жилүүдэд веб програмчлалд суурилсан төрөл бүрийн үйлчилгээ асар хурдацтай нэмэгдэж буй бөгөөд энэхүү өсөлттэй зэрэгцэн түүнд чиглэсэн эмзэг байдлын тоо ч хурдтай өсөж байна. Эмзэг байдлыг ангилан үзэхэд тэдгээрийн ихэнх нь баталгаажуулалт, веб системд тогтмол хийгдэх шаардлагатай арчилгаанаас болдог байна [1]. Вебд суурилсан аливаа системийн хувьд тухайн системд хандах хэрэглэгчийн интерфейстэй холбоотой аливаа эрсдэл түгээмэл тохиолдох боломжтой бөгөөд тэдгээрээс хамгийн түгээмэл тохиолддог нь SQL injection (SQLi), Cross-site scripting (XSS) юм. 2020 онд аюулгүй байдлын программ, судалгааны компани болох Acunetix-ийн боловсруулсан судалгааны үр дүнгээр маш эрсдэлтэй (critical) эмзэг байдлын жагсаалтад SQLi, XSS халдлагуудад өртөмтгий байдал түгээмэл, эрсдэл өндөртэй эмзэг байдлын ангилалд орсон байна [2]. Энэхүү халдлага үүсэх үндсэн нөхцөл нь оролт баталгаажуулахгүй байх, ямар нэг аюулгүй байдлаа хангах аргачлал ашиглаагүй байх, вебийн, веб системийн хамгаалалтын арга техник ашиглаагүй байх зэргээс болох нь түгээмэл байна. Иймээс аливаа байгууллага ашиглаж буй вебийн эмзэг байдлыг тодорхойлж, мэдэх нь чухал асуудал юм. Тухайн вебийн эмзэг байдлыг мэдэхгүй байснаар веб системийн өгөгдлийн сан, дэд бүтцэд эрсдэл үүсэхээс гадна эмзэг байдлын цар хүрээнээс хамааарч тухайн байгууллага, хувь хүнд их хэмжээний хохирол учруулах боломжтой. Зах зээл дээр вебийн эмзэг байдлыг илрүүлэх олон төрлийн программууд бий. Түүний дотроос Wapiti, Wa3f, ZAP, Veg зэрэг программууд өргөн ашиглагддаг бөгөөд тэдгээр нь түгээмэл тохиолддог XSS, SQLi, Directory Traversal гэх мэт олон халдлагуудыг илрүүлдэг давуу талтай боловч нүсэр бүтэцтэй, агуулга томтой, эмзэг байдлыг илрүүлэхдээ аюулгүй байдлын шалгалтыг командын мөрнөөс хийдэг зэрэг дутагдалтай талууд их байдаг.

Энэ нь хэрэглээнд ашиглахад хүндрэл учруулдаг байна. Бид энэхүү судалгаа, туршилтын үр дүнгээр дээрх асуудлуудыг шийдсэн нээлттэй эхийн, график интерфейстэй, аюулгүй байдлын олон эх үүсвэрээс аюулгүй байдлыг шалгах боломжтой, хэрэглэхэд хялбар, хамгийн өндөр эрсдэл бүхий SQLi, XSS эмзэг байдлуудыг төрлөөр нь ангилж илрүүлэх боломжтой системийн хөгжүүлэлтийг хийсэн.

2 Вебийн эмзэг байдал

Аж үйлдвэрийн хурдацтай хувьсгал, дэлхий нийтийн цахим хэрэглээний дэвшил нь бидний өдөр тутмын амьдралд огцом өөрчлөлтийг авчирсаар байна. Ялангуяа, веб програмчлал, веб технологи нь хэрэглэгчдэд хэрэглэхэд хялбар, түргэн шуурхай бөгөөд мэдээллийг цаг алдалгүй авах боломж олгож буй тул түүнд суурилсан үйлчилгээ алхам тутамд шинээр нэмэгдсээр байна. Урьд нь хийдэг байсан файл зөөх, аудио, видео тоглуулах зэрэг олон үйлчилгээ онлайн хэлбэрт шилжиж вебд суурилсан үйлчилгээ болсноор бидний амьдралд илүү ойр болсон. Энэхүү хурдацтай өөрчлөлтийн сөрөг тал нь хувь хүний мэдээллийн нууцлал алдагдах, аюулгүй байдал бүрэн хангагдаагүй вебүүд олшрох зэрэг эрсдэл, аюулыг бий болгож байна. Веб, веб програмчлал бүхий системүүдийн ихэнх нь хэрэглэгчийн оруулж буй өгөгдөлд баталгаажуулалт хийдэггүй, аюулгүй байдлын аливаа шийдлийг бүрэн гүйцэд ашигладаггүйтэй холбоотой эмзэг байдал үүсэж байгаа нь тогтоогдсон. Вебийн аюулгүй байдлыг хангах олон арга байдаг бөгөөд хэрэглэгчээс үүдэлтэй эрсдэлийг бууруулах, веб системээс үүдэлтэй эрсдэлийг бууруулах, веб аппликейшнээс үүдэлтэй эрсдэлийг бууруулах гэсэн үндсэн гурван ангилалд хуваан авч үзэж болно. Эдгээрээс түгээмэл ашиглагддаг арга нь давхар баталгаажуулалт ашиглах, вебийн галт ханыг, WAF (Web Application Firewall), зөв тохируулан ашиглах, өгөгдөл хадгалагдахад нууцлалыг хангах хүчтэй криптографи ашиглах, вебийн аливаа холболтын нууцлалыг хангах зорилгоор холболтын протоколын аюулгүй шийдлийг ашиглах, нууц үгийн стандартыг хэрэгжүүлэх, бодлого тодорхойлох, онц шаардлагатай тохиолдолд өгөгдлийг сэргээх боломжтой байх, нөөц сервер бэлдэн ачааллыг тэнцвэржүүлж, аюулгүй байдлыг ханган ажиллах, вебийн код, веб сервер болон бусад холбогдох дэд системүүдийн шинэчлэл, сайжруулалтыг цаг тухай бүр тогтмол, стандартын дагуу хийж байх юм [3].

2.1 Cross-Site Scripting

Энэхүү XSS эмзэг байдал нь аливаа вебийн хэрэглэгчийн интерфейст буй оролтын талбарт аюулгүй байдлын шалгалтуудыг хийгээгүйгээс үүдэн хакеруудад тухайн код сервер дээр ажиллан системд халдах, системийн мэдээллийг цуглуулах, системд хандаж буй хэрэглэгчид халдах, системийн найдвартай, тогтвортой байдлыг алдагдуулах зэрэг үйлдэл хийх боломжийг олгодог байна. Хэдийгээр энэ нь тухайн веб болон вебийг ашиглаж буй хэрэглэгчдэд хор хохирол их учруулж болох боловч түүнийг илрүүлэх, засах боломж харьцангуй хялбар байдаг. Уг эмзэг байдал бүхий вебээр дамжуулан халдагч этгээд тухайн вебд хандаж буй хэрэглэгчийг халдлагын өртөгч болгох боломжтой учир хэрэглэгчдэд ихээхэн хохирол учруулна. Энэ халдлага нь Reflected болон Persistent гэсэн үндсэн хоёр төрөлд хуваагдана [4]. Persistent XSS-д өртсөнөөр тухайн вебийн өгөгдлийн санд халдагч этгээдийн оруулсан хор хөнөөлтэй код хадгалагдах ба ямар нэгэн хэрэглэгч тухайн вебд хандах үед халдлагын код ажиллаж хэрэглэгчид халдварлаж, кодын дагуу төрөл бүрийн үйлдлүүдийг гүйцэтгэнэ. Уг халдлага амжилттай болох тохиолдол нь веб сервер аюулгүй байдлын шийдэлгүй, кодын аюулгүй байдлыг шалгадаггүй, зөвшөөрөгдөөгүй үйлдлийг хязгаарлаагүйтэй холбоотой бөгөөд хэрвээ тухайн веб нь дээрх шалгалтыг хийдэггүй, баталгаажуулалт ашигладаггүй бол халдлагад өртөх магадлал их болох юм [5]. Иймд Reflected XSS халдлага нь амжилттай хийгдсэн тохиолдолд тухайн халдлагын хамрах хүрээ ихсэж, хор хохирол нэмэгдэх эрсдэлүүдийг бий болгоно.

2.2 SQL Injection

SQLi нь өгөгдлийн санд хэрэглэгч эсвэл бусад системээс ирж буй SQL query-г шалгах хамгаалалтын ямар нэгэн аргачлал ашиглаагүй тохиолдолд өгөгдлийн сангийн мэдээллийг олж авах, мэдээлэл хуулах зэрэг хөнөөлт үйлдлүүд хийх боломжийг олгох бөгөөд MySQL, Microsoft SQL Server, Oracle гэх мэт түгээмэл ашиглагддаг бүх төрлийн өгөгдлийн сан уг халдлагад өртөх боломжтой [5]. Уг халдлага нь хөгжүүлэгчийн алдаа эсвэл вебийн системийн ажиллагааны алдаа зэргээс болж их тохиолддог. SQLi ашиглан халдагч нь SQL командыг өгөгдлийн сангийн сервер рүү илгээж, өгөгдөлд зөвшөөрөлгүй хандах эсвэл өгөгдлийн сангийн сервер ажиллаж байгаа системийг нь бүхэлд нь эзэмших оролдлого хийх бөгөөд тэрхүү боломжийг олж авснаар өөрийн хүссэн үйлдэл хийх боломжтой болдог. Simple SQLi нь энгийн, түгээмэл ашигладаг халдлагын төрөл юм. Үүний нэг жишээ нь: Хэрвээ веб нь хэрэглэгчийн ID дугаар дээр тулгуурлан хэрэглэгчийн мэдээллийг харуулдаг гэж үзвэл. Ямар нэг дугаар бүхий SQL query үүсгэн илгээх үед сонгосон дугаар бүхий хэрэглэгчийн мэдээлэл өгөгдлийн санд байхгүй тохиолдолд хоосон утга буцаах эсвэл алдааны мэдээлэл гарч ирнэ. Энэ тохиолдолд халдагч этгээд оролтын хэсэгт “111 OR 1=1” гэсэн утга бүхий query илгээсэн гэж үзвэл энэхүү логик үйлдлийн хариу нь үргэлж үнэн байх юм. Ингэснээр серверт уг query “SELECT * FROM users WHERE userId = 111 OR 1=1;” гэж дамжуулагдах бөгөөд үргэлж үнэн нөхцөл биелж өгөгдлийн сангаас харгалзах утгыг буцаах юм [6]. Энэхүү энгийн query-г дамжуулах боломжтой вебд хэрэглэгчийн мэдээлэл, хэрэглэгчийн хувийн мэдээлэл, нууц зэрэглэлийн мэдээлэл гэх зэ-

рэг агуулагдаж байсан гэж тооцвол уг нөхцөлийн үр дүнд өгөгдлийн сангаас хэрэглэгчдийн мэдээллийг буцааж тухайн мэдээллээр системд нэвтрэх, өгөгдөл алдагдах боломж бүрдэнэ.

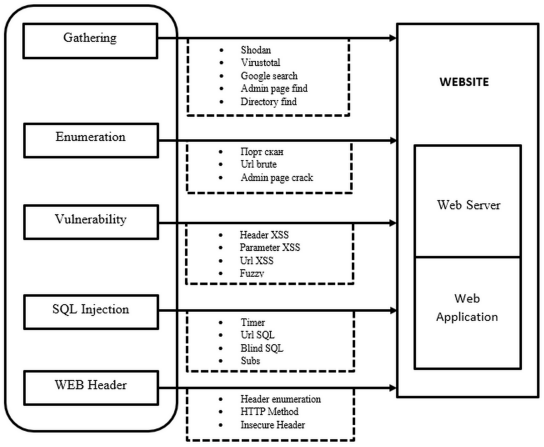
2.3 HTTP Header

Хэрэглэгч, веб хоорондын хүсэлт, хариу болон бусад нэмэлт мэдээллийг дамжуулах үед HTTP header ашигладаг. HTTP талбар нь General, Request, Response, Entity гэсэн дөрвөн талбараас бүрдэнэ. General header нь хүсэлт болон хариултын аль алинд нь хэрэглэгдэх хэдий ч өгөгдлийн санд өөрчлөлт оруулахгүй. Үүнд request url, request method, status code, remote address, referrer policy зэрэг агуулагдана. Request header хэсэгт хэрэглэгчийн илгээж буй хүсэлтийн мэдээллийг агуулдаг. Request header дотор cookie, user agent, cache-control, accept-language, accept-encoding, method, path гэсэн талбарууд орно. Response header гэх энэхүү талбарт серверээс хэрэглэгчийн илгээсэн хүсэлтэд хариулж буй мэдээллийг агуулдаг. Response header дотор server, date, server-timing, cache-control, content type, content-security-policy, secure header зэрэг орно [7].

3 Вебийн аюулгүй байдлын шинжилгээний програмын дизайн, хөгжүүлэлт

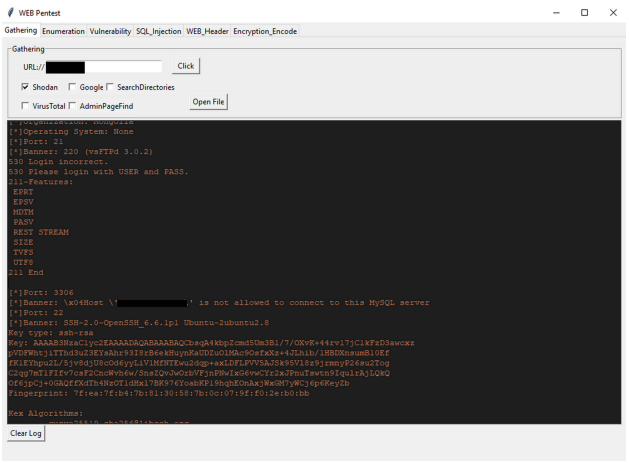
Бидний ажлын үр дүн өнөөдрийн байдлаар зах зээлд ашиглагдаж буй вебийн аюулгүй байдлын шинжилгээ хийх программын дутагдалтай талыг нөхөх юм. Уг программаар вебийн эмзэг байдлыг тодорхойлохоос гадна, вебийн, веб серверийн талаар мэдээлэл цуглуулах боломжтой. Үүнээс гадна нээлттэй эхийн программ хэлбэрээр хөгжүүлж буй тул бидний бүтээсэн программыг ашиглаж буй аюулгүй байдлын мэргэжилтэн, судлаачид цааш хөгжүүлэн, өөрсдийн хэрэгцээ шаардлагад нийцүүлэн ашиглах боломжтой. Бид судалгааны үр дүнд үндсэн 5 модулиас бүрдэхээр тооцсон бөгөөд а. Gathering (Мэдээлэл цуглуулах хэсэг), б. Enumeration (Системийн дэлгэрэнгүй мэдээллийг цуглуулах, илрүүлэх хэсэг) в. Эмзэг байдлыг тодорхойлох, илрүүлэх хэсэг г. SQLi илрүүлэх хэсэг д. Web header-ийн мэдээлэл цуглуулах хэсэг гэсэн модулиуд байна. Зураг 1-т программын ажиллагааны ерөнхий архитектурыг харуулсан бөгөөд модуль бүр ашиглагдах сан, өгөгдлийн бүтэцтэй байхаар тооцоолсон бөгөөд энэ нь бусад судлаач, инженерүүдэд ашиглах боломж олгох, системийн үр дүнгээр гарсан мэдээллийг дараагийн судалгаанд ашиглахтай холбоотой боловсруулсан шийдэл юм.

Модулиуд нь өөр өөрсдийн үйлдлийг гүйцэтгэхдээ хоорондоо харилцан ажиллах болон бие даасан байдлаар үүрэг гүйцэтгэх боломжтой байхаар загварчилсан. Жишээ нь бид энэхүү судалгааны үр дүнд модулиуд хоорондоо хамааралтай буюу эхний модулийн гаралтын үр дүнгээс хамааран дараагийн



Зураг 1: Ажиллагааны загварчлал.

модулийн хийгдэх үйлдэл тодорхойлогдох замаар гүйцэтгэсэн.



Зураг 2: Програмын интерфэйс.

Зураг 2-т мэдээлэл цуглуулах модулийн ажиллагааны талбар буюу ерөнхий интерфейс болон мэдээлэл цуглуулах үр дүнгийн хэсгийг харууллаа. Системийн модуль бүрийн ажиллах талбар өөр өөр байх боловч үр дүнг мэдээлэх ерөнхий гаралтын интерфейстэй байна. Зураг 2-т харагдаж байгаа нь системийн эхний модуль болох Gathering хэсэг юм. Gathering хэсэг дээр URL: гэсэн оролтын хэсэгт мэдээлэл цуглуулах гэж байгаа сайтын URL-г оруулж, доор нь байгаа Shodan, Google, SearchDirectories, Virustotal, AdminPageFind гэсэн сонголтуудаас сонгоод “Click” товчлуурыг дарснаар систем ажиллаж, үр дүнгийн мэдээллийг гаралтын хэсэгт харуулна.

3.1 Мэдээлэл цуглуулах модуль

Мэдээлэл цуглуулах (gathering) модуль нь тухайн системийн талаарх ерөнхий мэдээллийг хайх хэсэг юм. Ерөнхий мэдээлэл гэдэгт ямар ip хаягтай, ямар хост дээр байршсан, ямар сервер дээр ажиллаж байгаа гэх мэт олон төрлийн мэдээллүүд багтана. Бидний боловсруулсан систем ту-

хайн вебээс shodan, virustotal, google search, admin page find, directory find зэрэг хайлтын функуудийг ашиглаж мэдээлэл олж авах боломжтой. Жишээ нь, shodan нь интернетэд холбогдсон төхөөрөмжүүдийн талаар мэдээлэл цуглуулдаг хайлтын систем юм.Shodan,Virustotal нь пайтон хэлний сантай байдаг. Тэрхүү пайтон хэлний сантай холбогдож API түлхүүр ашиглан хэрэгтэй мэдээллээ цуглуулна. Энэ систем мэдээллийн ихэнх хэсгийг төхөөрөмж дээр ажилладаг программ хангамжийн мета өгөгдөл болох баннераас цуглуулах боломжтой байдаг [8]. Энэ нь серверийн программ хангамжийн талаарх мэдээлэл, ямар үйлчилгээг дэмждэг эсвэл хэрэглэгч сервертэй харьцахаасаа өмнөх мэдэхийг хүсч байгаа зэргийг харуулна. Virustotal нь зааж өгсөн веб системийг 70 гаруй антивирус программаар шалгахаас гадна тухайн URL буюу домайн нь хар жагсаалтад багтсан эсэхийг шалгахаас авахуулаад бусад нэмэлт мэдээлэл авах боломжтой байдаг [9]. Мөн бид системээ search directory, admin page find гэх мэт веб системийн чухал бүрэлдэхүүн хэсгүүдийн мэдээллийг хайх сонголттой байхаар зохион байгуулсан.

3.2 Системийн дэлгэрэнгүй мэдээлэл илрүүлэх модуль

Энэ хэсэгт цуглуулсан мэдээлэл дээрээ үндэслэн admin page crack, url brute extension функцүүдийг хэрэгжүүлэх мөн порт скан үйлдлийг гүйцэтгэнэ. Ингэснээр тухайн веб сайт нь ямар үйлчилгээ үзүүлдэг, ямар порт ашигладаг, ямар файлууд агуулсан гэх мэт нарийвчилсан мэдээллүүдийг цуглуулна. Энэ систем порт скан үйлдлийг scapy модуль дээр суурилж гүйцэтгэдэг бөгөөд үүнийг системд ашигласнаар бусад төрлийн скан хийдэг функцуудыг бодвол илүү хурдан, галт ханаар хамгаалагдсан эсэхийг тодорхойлох боломж олгодог давуу талтай. Жишээ нь системийн gathering хэсгээс админ хуудас хайх сонголтыг идэвхжүүлснээр админ хуудсыг илрүүлж, тухайн хуудасны нэвтрэх нэр болон нууц үгийг тодорхойлох боломжтой болно.

3.3 Эмзэг байдал илрүүлэх модуль

Энэ хэсэгт системийн эмзэг байдлыг шалгах XSS тест хийнэ. Бид энэ хэсэгт пайтон хэлний HTML болон XML файлуудаас мэдээлэл татах Beautiful Soup санг ашигласан [10]. Энэ санг ашигласнаар HTML талбартай хялбар ажиллаж оролтын хэсгүүдийг ялгах боломжтой болсон. Энэ санг ашигласнаар HTML талбартай хялбар ажиллаж оролтын хэсгүүдийг ялгах боломжтой болсон. XSS тестийг ганцхан системийн оролт (input) хэсэгт шалгахгүйгээр бусад аргаар шалгах боломжтой Мөн header XSS-ээр дамжуулан хэрэглэгч, веб хоорондын харилцааны HTTP header хэсэгт XSS хортой код илгээх боломжтой. Мөн түүнчлэн parameter XSS-ээр дамжуулан системийн дамжуулж байгаа параметрт XSS хортой код илгээж шалгана [11]. Түүнээс гадна вебийн эмзэг байдлын түгээмэл цоорхойнуудын нэг болох директор гэтлэх (Directory Traversal) халдлагад өртөмтгий байдлыг тодорхойлох боломжтой. Энэ функцийг ашиглахад системээс тест хийх payload-ийг шууд ачаалахаас гадна payload-ийг сольж болно.

3.4 SQLi илрүүлэх модуль

Энэ хэсэгт SQL тарилга (SQLi) халдлагад өртөмтгий эсэхийг тодорхойлно. Энэ халдлагад өртөх боломжтой эсэхийг тодорхойлохдоо SQL query-г тусгай тэмдэгт ашиглан илгээснээр системийн дэлгэцэд алдааны мессэж гарч ирэх байдлаар үр дүнг нь мэдээлнэ. Дэлгэцэд алдааны мессэж гарч ирснээр халдагч веб системд ажиллаж байгаа өгөгдлийн сан, серверийн талаарх мэдээллүүдийг авснаар дараагийн алхмуудыг хэрэгжүүлэх үндэс болно. Blind SQLi буюу сохор гэж нэрлэдэг халдлага нь хортой код оруулах болон тусгай тэмдэгт агуулсан query оруулах үед веб системээс ямар нэгэн алдааны мессэж илэрдэггүй буюу хариу үйлдэл үзүүлэхгүй байх үед хэрэгжүүлдэг халдлага юм. Алдааны мессэж гарч ирэхгүй байснаар тухайн систем ямар төрлийн өгөгдлийн сан ашиглаж байгаа болон түүн дээр тулгуурлан халдлагыг хэрэгжүүлэхэд тодорхойгүй байдлыг бий болгоно [11]. Ингэснээр тухайн веб системд SQLi төрлийн цоорхой байгаа үгүйг мэдэж чадахгүй байдалд хүргэнэ. Мөн энэ төрлийн халдлагыг хийхэд Boolean based, Time based гэсэн 2 төрлийн аргыг ашиглаж халдлагыг гүйцэтгэх боломжтой. Boolean based арга нь Boolean тэгшитгэл дээр суурилагдаж зохиогдсон төрөл юм [12]. Boolean тэгшитгэл буюу үнэн эсвэл худал гэсэн SQL query серверлүү явуулж ямар хариу гарч байгаагаас шалтгаалж дараагийн нөхцөл байдлыг хийхэд хүргэнэ. Time based арга нь мөн адил дээрх Boolean аргатай ижил бөгөөд Boolean тэгшитгэл үнэн эсвэл худал гэсэн хариуны хугацаанаас хамааарч яг аль нь вэ гэдгийг тодорхойлж болох арга юм. Бид эдгээр аргуудад тулгуурлан SQLi эмзэг байдлыг тодорхойлох функцийг ашигласан.

3.5 WEB header тодорхойлох модуль

Энэ модуль нь веб хэрэглэгч хоорондын харилцаанд чухал үүрэг гүйцэтгэдэг http header-ийн мэдээллийн талаар судална. Эндээс бид HTTP header-ийн ямар төрлийг хэрэглэж байгаа болон хамгаалалтын ямар арга, хэрэгслийг header хэсэгтээ ашиглаж байгаа болохыг тодорхойлно. Мөн сервер хэрэглэгч хоорондын харилцаанд хамгаалалттай күүки ашиглаж байгаа эсэх болон ямар сервер ашиглаж байна гэдгийг тодорхойлж болно.

4 Туршилтын үр дүн

Бидний хөгжүүлсэн веб тест хийдэг систем XSS, SQL Injection цоорхойг шалгахаас гадна HTTP header мэдээллийг давхар шалгана. Ингэснээр HTTP header-ийн XSS protection, x-frame-options

гэх мэт вебийн аюулгүй байдалд нөлөөлдөг асуудлуудыг тодорхойлно. Судалгааны ажлын явцад Wa3f, Wapiti, ZAP, Vega программуудыг туршихад олон тооны файлтай механик аргаар харьцаж, олон үйлдэл хийх сул талуудтай байсан. Программуудын тус бүрийн сул талыг дурьдвал Wa3f программ нь пайтон хэл дээр бичигдсэн sql, xss халдлагыг илрүүлэх боломжтой ч мэдээлэл цуглуулах модуль байхгүй, windows үйлдлийн систем дээр суулгахад хүндрэлтэй, ZAP программ нь windows үйлдлийн систем дээр сайн ажиллаж байгаа ч нүсэр бүтэцтэй учир хэрэглэгчийн компьютерыг удаашруулах болон гацаах сул талтай байна, Wapiti программ нь пайтон хэл дээр бичигдсэн ч график интерфейсгүй, мэдээлэл цуглуулах модуль байхгүй, Vega программ нь бусдыгаа бодвол ажиллагааны хувьд удаан байсан. Эдгээр сул талуудыг багасгах зорилгоор өөрсдийн хөгжүүлж байгаа системдээ машин сургалтын арга хэрэглэж бүтцийн хувьд жижиг болгон механик үйлдлийг нь багасгах юм. Бид системээ хөгжүүлэх явцдаа эхлээд туршилтын зорилгоор ашиглагддаг цоорхой веб системүүд болох WebGoat, DWVA, Bwapp программууд дээр шалгах замаар программын сайжруулалтыг хийсэн. Эдгээр цоорхой веб программууд дээр амжилттай туршиж зорилтод цоорхойтой хэсгүүдийг илрүүлсний дараа түүнийгээ баталгаажуулж Монголын зарим веб сайтууд дээр туршилтыг хийж гүйцэтгэсэн. Туршилт хийх веб сайтуудыг олоход Dork систем ашигласан. Хүснэгт 1-н эхний баганад тест хийсэн сайтуудыг аюулгүй байдал, хууль эрх зүйн асуудал үүсгэхгүй байхаар Site гэж нэрлэн, дугаарласан ба дараагийн баганад эмзэг байдлын тест хийж байгаа арга болон төрлүүдийг сүүлийн баганад тест хийсэн үр дүнг харуулсан.

Хүснэгт 1: Туршилтын үр дүн харуулсан байдал

Тест хийсэн сайт	Эмзэг байдлын тест хийсэн арга		Үр дүн
Site 1	XSS	Header	1
		Parameter	1
		URL_Scan	1
		Directory Traversal	0
	SQL Injection	BlindSQL_Time	0
		BLindSQL_Boolean	0
		URLSql	0
	WEB Header	Insecure header	1
Site 2	XSS	Header	0
		Parameter	0
		URL_Scan	0
		Directory Traversal	0
	SQL Injection	BlindSQL_Time	0
		BLindSQL_Boolean	1
		URLSql	1
	WEB Header	Insecure header	0
Site 3	XSS	Header	0
		Parameter	1
		URL_Scan	1
		Directory Traversal	1
	SQL Injection	BlindSQL_Time	0
		BLindSQL_Boolean	1
		URLSql	1
	WEB Header	Insecure header	1

Програм тодорхой үр дүнг гаргахдаа дараах байдлаар ажиллана. Хэрэв вебд эмзэг байдал байгаа нь батлагдвал "True", эмзэг байдлын нэр, ашигласан payload болон эмзэг байдлын талаарх товч тодорхойлолт дэлгэцэнд хэвлэгдэнэ. Хэрвээ эмзэг байдал байхгүй нь батлагдвал "False"болон "No Vulnerability"гэсэн үг дэлгэцэнд гарч ирнэ.

Шалгалтын үр дүнг илэрхийлэхдээ хэрэв веб систем тухайн эмзэг байдлыг агуулсан байвал 1 үгүй бол 0 гэсэн тэмдэглэгээг хэрэглэсэн.

Туршилтын үр дүнгээс харахад Site 1 нь XSS халдлагад өртөх боломжтой, хамгаалалтын Header агуулаагүй байна. Site 2 нь харин SQLi халдлагад илүү өртөмтгий ч хамгаалалтын Header ашигласан байгаа нь XSS халдлагаас сэргийлж боломжтой. Харин Site 3 нь XSS болон SQLi аль алинд нь өртөмтгий мөн дээрээс нь хамгаалалтын Header агуулаагүй байгаа нь халдлагад нөгөө 2 сайтаасаа илүү өртөмтгий байдал тодорхойлогдож байна. Эндээс дүгнэж үзэхэд туршилт явуулсан веб сайтуудын аюулгүй байдлын хамгаалалтууд бүрэн системтэй биш байгаа нь ажиглагдаж байна. Учир нь зарим сайтыг XSS халдлагаас хамгаалсан мөртлөө SQLi халдлагаас хамгаалаагүй, эсвэл SQLi халдлагаас хамгаалсан боловч XSS халдлагыг хамгаалаагүй байгаа нь туршилтын үр дүнгээр илэрч байна.

5 Дүгнэлт

Уг судалгаа, туршилтын ажлын үр дүнд веб системийн мэдээлэл цуглуулах, эмзэг байдлыг тодорхойлох тэр дундаа хамгийн түгээмэл тохиолддог бөгөөд ихээхэн хохирол учруулдаг SQLi, XSS халдлагын төрлүүдийг илрүүлэх боломжтой программын дизайн боловсруулж, модулиудыг хөгжүүлж үр дүнг туршилтаар баталгаажуулсан. Энэхүү системийг хэрэглээнд нэвтрүүлснээр хэрэглэгчид хандалт хийж буй веб нь хөнөөлтэй код агуулсан эсэхийг мэдэх боломжтой бол веб системийн админ, аюулгүй байдлын мэргэжилтнүүд өөрсдийн системийн аюулгүй байдлыг шалгах, ингэснээр аюулгүй байдлыг хамгаалахтай холбоотой авч хэрэгжүүлэх цаашдын арга хэмжээг тодорхойлох боломжтой болох зэрэг олон давуу тал бий болно. Үүнээс гадна судлаачид уг программыг ашиглан төрөл бүрийн судалгаа хийх боломж бүрдэнэ гэж үзэж байна. Бид цаашид энэхүү программыг илүү сайжруулах зорилгоор вебийн эмзэг байдал бүрээр CVSS оноо тооцоолох, гаралтын үр дүнг машин сургалт болон бусад судалгаанд ашиглах боломжтой хэлбэрээр боловсруулдаг модуль хөгжүүлэх зэрэг ажлуудыг хийхээр төлөвлөж байна.

Зохиогчийн оролцоо

Б.Цолмонтамир програмын дизайныг гаргаж, кодыг бичсэн. Б.Өсөхбаяр санаа боловсруулж програмын ерөнхий архитектурыг гаргасан. Н.Угтахбаяр програмын модулиудад шинэчлэн сайжруулах алгоритмын аргачлалыг тусгасан. Б. Өсөхбаяр, Б.Цолмонтамир, Н.Угтахбаяр нар өгүүлэллийг засаж сайжруулж, утга агуулгын алдааг хянаж, өгүүллийн сүүлийн хувилбарыг бичсэн.

Ашиг сонирхлын зөрчилгүйн баталгаа

Зохиогчид ашиг сонирхлын зөрчилгүй гэдгээ баталж байна.

Ашигласан ном

[1] Jovanovic N, Kruegel C, Kirda E. Pixy: A static analysis tool for detecting web application vulnerabilities. In: 2006 IEEE Symposium on Security and Privacy (S&P'06). IEEE; 2006. p. 6–pp.

[2] Acunetix Web Application Vulnerability Report 2020;. https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2020/.

[3] Lin JC, Chen JM, Liu CH. An automatic mechanism for sanitizing malicious injection. In: 2008 The 9th International Conference for Young Computer Scientists. IEEE; 2008. p. 1470–1475.

[4] Wang R, Xu G, Zeng X, Li X, Feng Z. TT-XSS: A novel taint tracking based dynamic detection framework for DOM Cross-Site Scripting. Journal of Parallel and Distributed Computing. 2018;118:100–106.

[5] Scott D, Sharp R. Abstracting application-level web security. In: Proceedings of the 11th international conference on World Wide Web; 2002. p. 396–407.

[6] Mishra S. SQL injection detection using machine learning. 2019.

[7] HTTP headers;. https://www.geeksforgeeks.org/http-headers/.

[8] What is Shodan?;. https://help.shodan.io/the-basics/what-is-shodan.

[9] How it works;. https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works.

[10] Beautiful Soup Documentation;. https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[11] Suhina V, Groš S, Kalafatić Z. Detecting vulnerabilities in Web applications by clustering Web pages. Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. 2008.

[12] Choudhary S, Singh N. Safety Measures and Auto Detection against SQL Injection Attacks. International Journal of Engineering and Advanced Technology (IJEAT). 2019;9(2):2827–2833.