

True Nature

Analizator în timp real al securității website-urilor
accesate



Pîrvu Daniel Cătălin

Cuprins

Introducere	3
Tipuri de trafic în rețele de calculatoare.....	4
Domain Name System (DNS)	4
Hypertext Transfer Protocol (HTTP).....	6
Utilitară în analiza traficului în rețele de calculatoare.....	8
Ce este un sniffer?	8
Wireshark / Tshark / Pyshark.....	9
Utilitară în analiza securității website-urilor.....	10
Wapiti.....	10
Skipfish / Arachni	13
True Nature.....	14
Descriere	14
Cum funcționează?	14
Cum se instalează?.....	16
Cum se folosește?	17
Concluzii	17
Tabela de figuri	18
Surse figuri	18
Bibliografie	18

Introducere

Vedem cum zilnic apar noi și noi tehnologii și unelte de lucru pe care le putem folosi pentru a analiza cât de poate de în detaliu datele pe care le transmitem sau pe care le primim.

De asemenea, apar noi și noi unelete cu care putem verifica securitatea website-urilor pe care le accesam. Acestea sunt utile pentru ambele categorii de persoane. În primul rând pentru dezvoltatori pentru că pot verifica ușor că munca lor este sigură împotriva diferitelor tipuri de atacuri. Pentru hacker acestea fac parte din prima fază de dezvoltare al unui atac, cea a căutării intense a vulnerabilităților care mai apoi să poată fi exploataate.

Ambale categorii de unelte sunt foarte eficiente și sunt în continuă îmbunătățire însă le lipsește o proprietate care ar putea eficientiza și spori timpul de lucru. Vorbim de automatizare. Mare parte din aceste tehnologii trebuie agregate într-o structură mai complexă pentru a ajunge la rezultatul dorit.

True Nature este o astfel de structură ce vine în ajutorul analizatorilor traficului de date și aduce în plus elementul de automatizare și de integrare cu altă tehnologie utilă în scanarea vulnerabilităților website-urilor.

Ca și în cazul menționat mai devreme, acesta poate fi folosit în scopuri bune, utilitatea lui fiind analiza automată a securității website-urilor pe care le accesam zi de zi. În scop negativ, un hacker ce nu are o anumită țintă clară poate folosi True Nature și astfel prin simpla navigare pe Internet poate afla informații și posibile ținte.

Tipuri de trafic în rețele de calculatoare

Domain Name System (DNS)

Unul din cele mai des utilizate protocole existente în traficul între rețele de calculatoare este Domain Name System sau simplu spus DNS.

Rolul lui este de a translata numele de domeniu într-o adresa IP fie ea publică sau privată, atunci când vorbim în context local. De ce am avea nevoie de aşa ceva? Răspunsul este pentru că browser-ele pe care le folosim zi de zi folosesc protocolul Internet (IP) și deci nu știu cum să lucreze cu un nume de domeniu.

DNS vine în ajutorul utilizatorilor care pot memora mai ușor un anumit website prin URL-ul acestuia, URL care conține numele de domeniu în locul unei secvențe de numere ca 209.165.200.254 sau mai complicată dacă vorbim de adrese IPv6.

Pentru a înțelege procesul de translatare realizat de DNS trebuie în primul rând să înțelegem componentele implicate:

1. *Recurzorul* poate fi perceput atât ca un software cât și ca un server și este responsabil de primirea cererilor de translatare de la celelalte aplicații precum browser-e. Tot el este responsabil de crearea și trimiterea cererilor adiționale pentru aflarea adresei IP dorite.
2. Serverul *domeniu rădăcina* este primul pas din procesul DNS și reprezintă locul ce ne poate da index-ul unui grup mai mare de calculatoare, de obicei asemuindu-se prin tipul de activitate sau locație geografică, ce ne poate filtra și duce mai rapid către rezultat.
3. *Serverul de nivel înalt* este colecția în care se poate găsi translatarea ultimei părți de domeniu. De exemplu în cazul "example.com", serverul de nivel înalt este "com".
4. *Serverul autoritar sau recursiv*. Serverul autoritar este ultimul hop în aflarea adresei IP și reprezintă colecția de tip dicționar care ne poate întoarce adresa IP atribuită domeniului din cererea noastră inițială. Un server recursiv nu conține adresa IP dorită și astfel poate crea la rândul lui cereri DNS către alte servere.

Cererea de tip DNS numită și DNS Lookup este compusă dintr-o serie de etape ascunse din punctul de vedere al utilizatorului obișnuit.

La accesarea unui domeniu să zicem "example.com" pachetul ajunge la un recursor.

Acesta crează cererea și o trimite către serverul domeniu rădăcina.

La rândul lui răspunde înapoi cu adresa serverului de nivel înalt ce stochează domenii din familia celui dorit de utilizator, în cazul exemplului anterior pentru ".com".

Recurstorul trimite acum cererea către serverul de nivel înalt și primește adresa IP a serverului autoritar care îl poate returna rezultatul final.

Recurstorul face cererea către serverul autoritar care întoarce adresa IP asociată domeniului de la care am plecat.

Ultimii pași ar fi ca recursorul să transmită browser-ului adresa IP pentru a putea continua traficul către adevărată adresa și stocarea rezultatului într-un cache local pentru o perioadă de timp pentru a eficientiza timpul de răspuns.

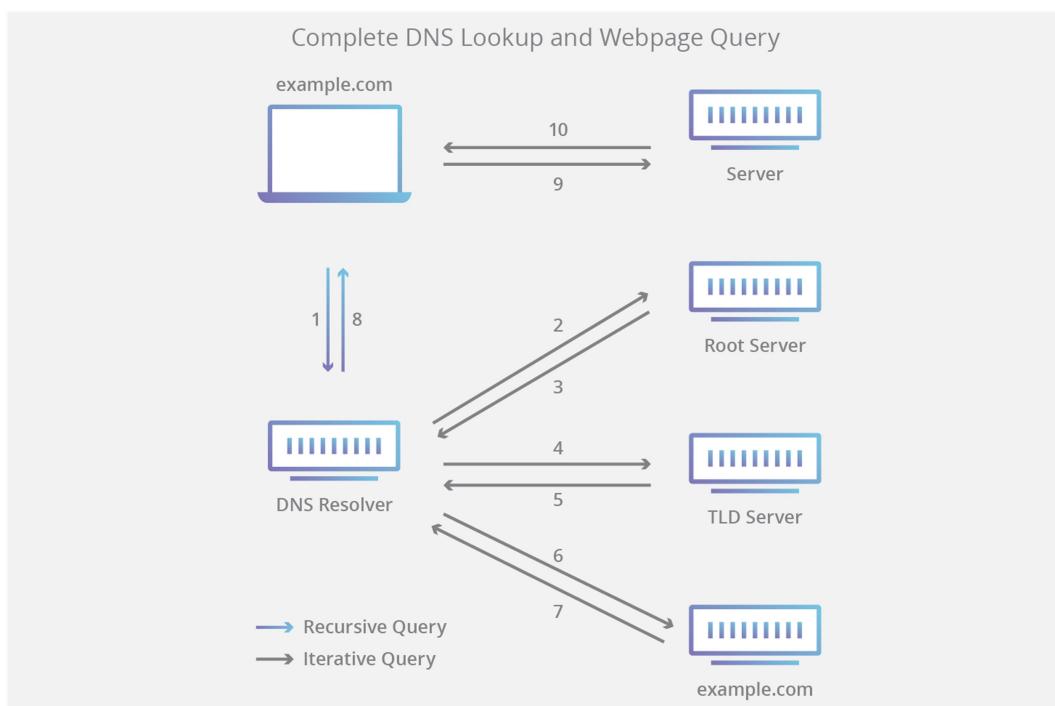


Figure 1 Cerere DNS

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol este un protocol de nivel 7 care permite împachetarea și accesul la resurse ce sunt stocate pe servere World Wide Web (WWW). Reprezintă baza a ceea ce înseamnă schimb de date în trafic de tip Web.

Este un protocol de tip client-server iar comunicarea dintre cele două părți se face prin mesaje. Clientul ce este în general un browser, dar poate fi și un utilitar precum curl, trimite cereri către un server și primește înapoi mesaje numite răspunsuri. Între cele două componenete de regulă se află și altele numite proxy-uri. Acestea au rolul de a opera ca porți de acces sau ca medii de stocare pentru cache.

Clientul sau agentul este cel care întotdeauna inițiază cererile, iar fiecare cerere este trimisă către un server, dar nu obligatoriu același. Pentru a genera o pagină Web agentul face o cerere pentru a primi de la server structura de baza a paginii în format HTML, după ce primește răspunsul și parsează fișierul trimis alături de elementele adiționale precum stilizare (CSS), funcționalitate (JS) sau obținere de fișiere media.

HTTP este ușor de citit și înțeles de către utilizatori, nu depinde de o stare dar aduce elemente la nivel de sesiune. Pentru că nu depinde de o stare, nu există o corelație între două cereri succesive realizate pe același canal. Contextul comun al unei multimi de cereri poate fi însă asigurat de cookies și sesiune.

Transportul este controlat de nivelul inferior ale stivei OSI șiiese din aria de acțiune al protocolului HTTP. Totuși avem nevoie de transmisie sigură între cele două terțe participante la trafic (client și server) și astfel ne bazăm pe protocolul TCP. El este în prima fază folosit pentru a stabili canalul de comunicare între cele două entități.

Corpul unei cereri conține câteva elemente simplu de înțeles: metodă care este în general GET sau POST, locația resursei, versiunea protocolului, posibile alte date în antetul cererii și, dacă vorbim de metodă POST, conține și un corp de date.

Corpul unui răspuns este foarte asemănător cu cel al unei cereri. Elementele cheie sunt codul și mesajul status.

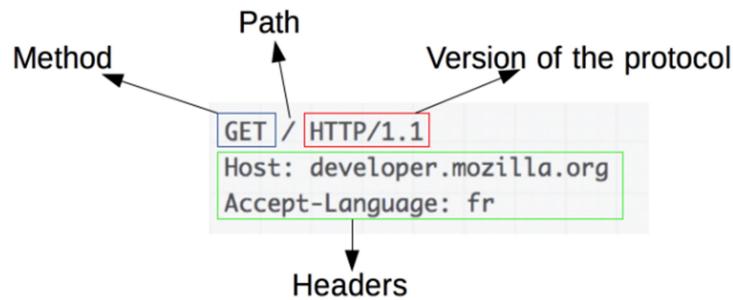


Figure 2 Corp cerere HTTP

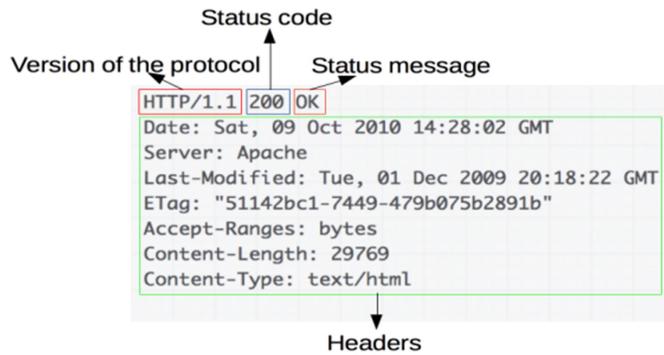


Figure 3 Corp raspuns HTTP

Un procedeu des întâlnit în aplicațiile web este redirectarea URL (URL redirection sau URL forwarding). Acesta permite website-urilor să poate fi accesate prin mai mult de o locație URL. Procesul este unul simplu, la accesarea unui URL care nu este cel final dorit de dezvoltatori, se va face redirect către cea finală.

Procedeul se poate aplica și colecțiilor ample, vorbim aici de domenii. De exemplu accesarea domeniului ‘facebook.org’ trimite de fapt către accesarea ‘facebook.com’.

De ce am avea nevoie de aşa ceva? În primul rând pentru o posibilă reducere a lungimii URL-ului și pentru a rezolva mici probleme de scriere (Accesarea fcbook.com trimite către scrierea corectă facebook.com). Permite deținerea mai multor domenii sub același proprietar și îmbunătățește ușor securitatea prin combaterea atacurilor de tip phishing care se bazează pe acestea mici diferențe insesizabile.

Redirectarea ajută în primul rând la aflarea URL-ului de baza al website-ului. Astfel putem ajunge pe pagină de start al unui website doar prin introducerea domeniului în URL.

Utilitare în analiza traficului în rețele de calculatoare

Ce este un sniffer?

Un sniffer este un utilitar ce permite monitorizarea traficului în timp real, capabil de capturarea pachetelor atât trimise cât și primite de la alte dispozitive.

În general ele sunt folosite de administratorii de rețea pentru a menține un flow stabil de date și pentru a vedea atunci când există nereguli. De regulă primul semn prin care ne putem da seama că dispozitivul nostru este atacat este traficul intens către și dinspre dispozitiv și lățimea de bandă utilizată.

Capabilitățile unui astfel de software nu se extind foarte mult. Un sniffer poate doar captura pachete și poate vedea natura și conținutul lor.

Deși utilizat în principal în scopuri bune, un astfel de software poate fi folosit și de către atacatori. Prin simpla monitorizare acestia pot afla informații prețioase precum emailuri, mesaje sau informații mult mai sensibile precum credențiale sau informații financiare. Locul preferat în care aceștia pot acționa este o zonă acoperită de o rețea WiFi publică și nesecurizată.

Există două tipuri de sniffing în funcție de natura dispozitivelor pe care le folosim. Dacă vorbim despre dispozitive hub care prin natura lor transmit trafic la toate dispozitivele conectate la el, atunci ne putem limita la sniffing de tip pasiv. Dacă avansăm și folosim switch capabil să transmită pachetele doar destinatarului căruia i-a fost trimis pachetul atunci vorbim de sniffing activ. Al doilea tip implică și că software-ul are nevoie să injecteze trafic adițional pe rețea, lucru ce îl face mai ușor detectabil.

Pentru a ne proteja de astfel de atacuri trebuie să ne ferim de rețele publice WiFi pentru că dacă noi avem acces atunci și un atacator poate la fel de bine. Trebuie să evităm să accesăm websiteuri și aplicații nesecurizate. Măsura generală de securitate este să folosim conținut criptat, fie că vorbim de protocolul HTTPS sau mai în detaliu de VPN.

Wireshark / Tshark / Pyshark

Wireshark este cel mai cunoscut utilitar disponibil în mod gratuit și pentru orice sistem de operare pentru realizarea de astfel de monitorizări ale traficului în rețele de calculatoare.

Ce poate să facă? Este folosit în principal pentru că afișează într-un format cități informația binară pe care o captează. De asemenea, deține o interfață grafică care aduce o serie vastă de filtre și unelte cu care se pot genera statistici.

Este capabil să interpreze o serie de peste două mii de protocoale mai vechi sau mai noi și aparținând tuturor nivelelor stivei OSI dar în general vor fi de tip TCP, UDP sau ICMP.

Cu toate că dispune de o monitorizare vastă, el nu trimitе alerte dacă un anumit tipar este întâlnit și de aceea nu trebuie confundat cu un sistem de detectare a intruziunilor (Intrusion Detection System - IDS).

Varianta de terminal a programului Wireshark se numește Tshark și dispune de aceeași funcționalitate însă trebuie compuse comenzi complexe. Tshark este foarte util atunci când vrem să integrăm monitorizarea pachetelor într-o structură mai complexă.

```
$ tshark -h
TShark (Wireshark) 3.4.2 (v3.4.2-0-ga889cf1b1bf9)
Dump and analyze network traffic.
See https://www.wireshark.org for more information.

Usage: tshark [options] ...

Capture interface:
  -i <interface>, --interface <interface>
    name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>, --snapshot-length <snaplen>
    packet snapshot length (def: appropriate maximum)
  -p, --no-promiscuous-mode
    don't capture in promiscuous mode
  -I, --monitor-mode
    capture in monitor mode, if available
  -B <buffer size>, --buffer-size <buffer size>
    size of kernel buffer (def: 2MB)
  -y <link type>, --linktype <link type>
    link layer type (def: first appropriate)
  --time-stamp-type <type> timestamp method for interface
  -D, --list-interfaces
    print list of interfaces and exit
  -L, --list-data-link-types
    print list of link-layer types of iface and exit
  --list-time-stamp-types
    print list of timestamp types for iface and exit

Capture stop conditions:
  -c <packet count>      stop after n packets (def: infinite)
  -a <autostop cond.> ...
    duration:NUM - stop after NUM seconds
    filesize:NUM - stop this file after NUM KB
    files:NUM - stop after NUM files
    packets:NUM - stop after NUM packets
```

Figure 4 Secvență din rezultatul comenzi "tshark -h"

Dacă structura mai complexă alegem să o construim folosind limbajul de programare Python atunci un utilitar care se folosește de Tshark este Pyshark.

Toate cele trei utilitare dispun atât de analiză în timp real cât și de încărcarea unui segment de trafic captat anterior.

Utilitare în analiza securității website-urilor

Wapiti

Wapiti este un analizator al securității website-urilor sau mai amplu spus al aplicațiilor web. Este gratuit și deoarece are la bază limbajul Python se poate folosi pe mai multe sisteme de operare.

Wapiti lucrează într-un blackbox și este independent de codul sursă al aplicației web. El primește un URL de start, după care începe să parcurgă websiteurile aplicației în încercarea de a găsi secvențe de cod sau formulare unde poate injecta date. Odată ce găsește toate aceste zone de acțiune Wapiti începe să injecteze date pentru a testa și afla dacă există vulnerabilități.

Este capabil să detecteze vulnerabilitati din multe zone de interes:

- Descoperirea unor fișiere
- Vulnerabilități ale bazelor de date
- Cross Site Scripting XSS
- Deteția unor metode de evaluare precum eval() sau system()
- Folosirea unor fișiere nesigure (Se bazează pentru asta de baza de date al utilitarului Nikto)
- Prezența unor fișiere de backup ce ar putea arăta codul sursă
- Fișiere .htaccess slabă
- Metode HTTP ce pot fi mascate prin PUT
- Etc.

Poate acționa folosind ambele metode principale HTTP (GET și POST). Se poate folosi de formulare complexe și poate chiar injecta date sub formă de fișiere încărcate.

Rezultatul poate fi expus în mai multe forme, cel implicit fiind cel de pagină HTML.

Deși încă din start se poate seta un nivel de afișare al informațiilor cât mai explicit și un timp maxim în care scannerul își poate face treaba, putem și noi să oprim sau mai bine spus să punem pe pauză scanarea pentru a vedea ce a reușit să obțină până la momentul respectiv.

Putem apoi alege să sară peste anumite etape (de exemplu să nu mai continue să caute alte pagini și alte posibile zone de acțiune sau să treacă peste un anumit tipar de atacuri) sau chiar să oprim acolo întregul proces.

Oprirea manuală se poate încheia cu sau fără generarea raportului, pe când comportamentul implicit atunci când scanarea își termină procesul normal este de a genera raport în locația specificată în comandă sau dacă nu se precizează în locația de baza a utilizatorului de sistem ce inițiază scanarea.

Figure 5 Seacentă din rezultatul comenzi "wapiti -h"

În următoarea poză putem vedea cum Wapiti este pornit și cum afișează în terminal fiecare informație utilă utilizatorului pentru că acesta să vadă parcursul în timp.

Figure 6 Conținutul scanării Wapiti în terminal

Raportul implicit de tip HTML arată de forma următoare.

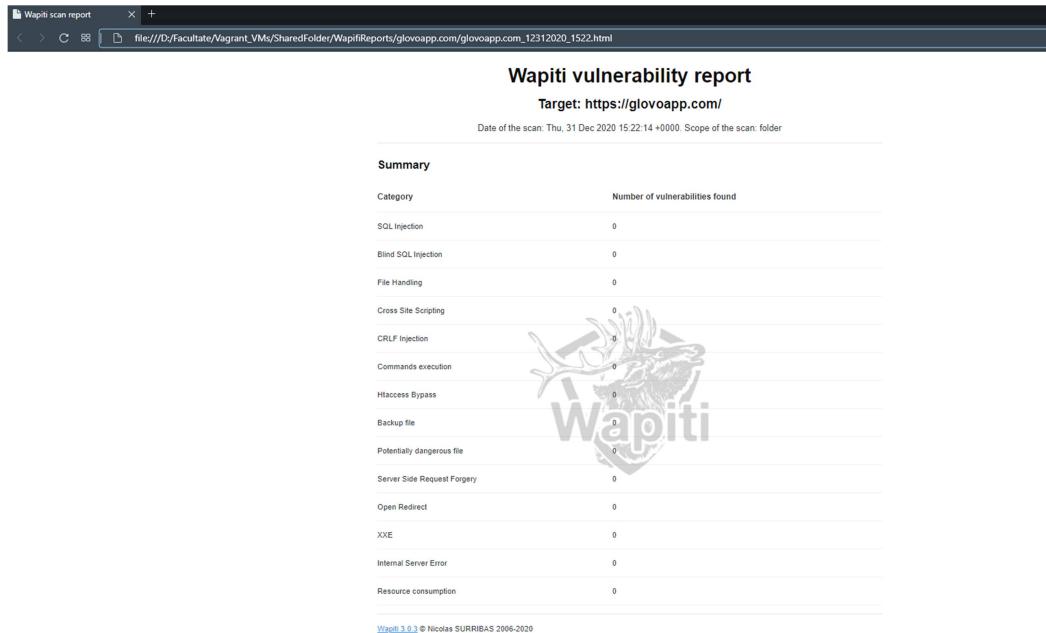


Figure 7 Raport HTML Wapiti

Raportul arată sub formă tabelară numărul de vulnerabilități găsite (în cazul website-ului testat zero) pentru fiecare categorie de atacuri. Dacă ar fi găsit vulnerabilități raportul ar fi fost mai extins și ar fi arătat și metodele de preventie împotriva atacurilor.

Skipfish / Arachni

Printre alte utilitare de scanare a securității website-urilor se numără Skipfish și Arachi.

Acestea au o desfășurare a acțiunilor similară dacă nu chiar identică cu Wapiti. Este vorba doar de performanță și de preferință formatului rezultatului atunci când alegem ce utilitar să folosim.

```
vagrant@kali:~$ skipfish -h
Skipfish web application scanner - version 2.10b
Usage: skipfish [ options ... ] -W wordlist -o output_dir start_url [ start_url2 ... ]

Authentication and access options:

-A user:pass      - use specified HTTP authentication credentials
-F host=IP        - pretend that 'host' resolves to 'IP'
-C name=val       - append a custom cookie to all requests
-H name=val       - append a custom HTTP header to all requests
-b (i|f|p)        - use headers consistent with MSIE / Firefox / iPhone
-N                - do not accept any new cookies
--auth-form url   - form authentication URL
--auth-user user   - form authentication user
--auth-pass pass   - form authentication password
--auth-verify-url - URL for in-session detection

Crawl scope options:

-d max_depth      - maximum crawl tree depth (16)
-c max_child       - maximum children to index per node (512)
-x max_desc        - maximum descendants to index per branch (8192)
-r r_limit         - max total number of requests to send (100000000)
-p crawl%          - node and link crawl probability (100%)
-q hex             - repeat probabilistic scan with given seed
-I string          - only follow URLs matching 'string'
-X string          - exclude URLs matching 'string'
-K string          - do not fuzz parameters named 'string'
-D domain          - crawl cross-site links to another domain
-B domain          - trust, but do not crawl, another domain
-Z                - do not descend into 5xx locations
-O                - do not submit any forms
-P                - do not parse HTML, etc, to find new links

Reporting options:

-o dir            - write output to specified directory (required)
-M                - log warnings about mixed content / non-SSL passwords
-E                - log all HTTP/1.0 / HTTP/1.1 caching intent mismatches
-U                - log all external URLs and e-mails seen
-Q                - completely suppress duplicate nodes in reports
-u                - be quiet, disable realtime progress stats
-v                - enable runtime logging (to stderr)

Dictionary management options:

-W wordlist        - use a specified read-write wordlist (required)
-S wordlist        - load a supplemental read-only wordlist
-L                - do not auto-learn new keywords for the site
-Y                - do not fuzz extensions in directory brute-force
-R age             - purge words hit more than 'age' scans ago
-T name=val        - add new form auto-fill rule
-G max_guess       - maximum number of keyword guesses to keep (256)
-z sigfile         - load signatures from this file
```

Figure 8 Secvență din rezultatul comenzi "skipfish -h"

True Nature

Descriere

True Nature este un utilitar menit să combine toate tehnologiile descrise până acum într-un format simplu de utilizat, automatizat și transparent.

În urma executării programului, la o locație aleasă de noi la început, vom găsi rapoartele de securitate realizate pentru website-urile pe care navigăm.

Cum funcționează?

În primul rând programul detectează tipul de sistem de operare. Acest lucru este util în verificarea corectitudinii următorei decizii ce trebuie realizată de utilizator.

Tot în partea de început se preiau informațiile din fișierul de configurare. Aceste configurații includ calea în sistemul de fișiere unde se vor salva rapoartele și două liste, una care conține secvențe de domenii pe care le dorim exclude din traficul captat și una care conține host-uri explicite care trebuie exclude.

Se folosește Tshark pentru a afla și afișa lista interfețelor de rețea pe care le putem folosi. Urmează ca utilizatorul prin introducere manuală de la tastatură să aleagă interfața întâi. În cazul sistemelor Linux trebuie specificat numele explicit al interfeței, pe când în cazul sistemului Windows trebuie specificat numărul interfeței.

```
danielP@DESKTOP-H4Q0N24 MINGW64 /d/Facultate/Vagrant_VMs/SharedFolder/TrueNature (main)
$ python TrueNature.py

The current OS is Windows

The current config is :
domains_excluded : ['facebook', 'google', 'microsoft', 'youtube']
hosts_excluded : ['scontent.ftce2-1.fna.fbcdn.net', 'get.geo.opera.com', 'weather.opera-api.com', 'suggestions.opera-api.com', 'af.opera.com']
reports_path : D:\Facultate\Vagrant_VMs\SharedFolder\WapifiReports

The list of your network interfaces is:
1. \Device\NPF_{F5546770-9D07-456A-8387-266EE6CAF3DC} (vEthernet (WSL))
2. \Device\NPF_{388F4C44-D232-4AA-9503-53364B65411A} (Local Area Connection* 8)
3. \Device\NPF_{BACF153B-7304-47F6-B1A2-DFAAB4A6E95F} (Local Area Connection* 1)
4. \Device\NPF_{9E2341C5-B93B-4303-A56C-83CD1F0F2B20} (VirtualBox Host-Only Network)
5. \Device\NPF_{E68D0688-D61A-443A-813C-DFD858DEAF24} (Ethernet)
6. \Device\NPF_{D8E32608-E50C-4C1A-996C-FD901DEDC063} (Ethernet (BluestacksNetwo))
7. \Device\NPF_{C1779BF9-2BC3-43BF-A95A-C0ED13350B71} (Local Area Connection* 7)
8. \Device\NPF_{08839581-308C-4EA3-B502-001FB074C764} (Ethernet (Default Switch))
9. \Device\NPF_{E53EB23C-401F-4B91-8E5F-9C260AC0C043} (Local Area Connection* 9)
10. \Device\NPF_{7AC1FF98-5C78-42D7-B268-06EE7EFA5713} (Local Area Connection* 10)
11. \Device\NPF_{2896ED28-A5CA-4FFC-B7A3-B65234AD9413} (Wi-Fi)
12. \Device\NPF_Loopback (Adapter for loopback traffic capture)
13. \Device\NPF_{3C3D40F8-6504-4738-8BB0-EC952DC7E351} (Ethernet 2)

Please type the number of the interface you want to sniff the traffic from! : |
```

Figure 9 Secvența de alegere a interfeței de lucru

După ce obținem aceste informații programul poate începe execuția efectivă.

Se vor realiza captări ale traficului de tip DNS în intervale de timp de câte 5 secunde. Captarea pachetelor se realizează cu ajutorul lui Pyshark care primește atât interfață cât și un filtru cu care putem exclude anumite domenii descoperite, filtru ce este creat automat pe baza celor două liste din fișierul de configurație.

După fiecare interval, se parcurg domeniile aflate și se încearcă accesarea domeniilor folosind protocolul HTTP. Rolul este de a afla prin mecanismul de redirectare adresa URL adevărată pe care o vom folosi la analiza de securitate.

Înainte de pornirea scanării fiecărui domeniu se verifică să nu generăm rapoarte pentru domenii deja testate.

Dacă domeniul este nou, se pornește intr-un nou proces scanarea folosind Wapiti. Deoarece se execută scanări în paralel putem eficientiza timpul general al programului.

```
The traffic filter constructed is
(dns.response_to) and (not dns.resp.name contains facebook) and (not dns.resp.name contains google) and (not dns.resp.name eq get.geo.opera.com) and (not dns.resp.name eq weather.opera-api.com) and (not dns.resp.name eq suggestions.opera-ap

Currently extracted sites :
- min-api.cryptoCompare.com
- min-api.cryptoCompare.com
- min-api.cryptoCompare.com
- min-api.cryptoCompare.com
- media-exp1.licdn.com
- www.linkedin.com
- static-exp1.licdn.com
- static-exp1.licdn.com
- media-exp1.licdn.com
- www.linkedin.com
- www.linkedin.com
- media-exp1.licdn.com

New possible data from which to extract new exceptions:

1
- Dns: min-api.cryptoCompare.com
- Url: https://min-api.cryptoCompare.com/
- Path: D:\Facultate\Tehnici Moderne Pt Securizarea Informatiilor\TrueNature\WapifiReports\min-api.cryptoCompare.com

Wapiti command will start in another process for https://min-api.cryptoCompare.com/

2
- Dns: www.linkedin.com
- Url: https://www.linkedin.com/
- Path: D:\Facultate\Tehnici Moderne Pt Securizarea Informatiilor\TrueNature\WapifiReports\www.linkedin.com

Wapiti command will start in another process for https://www.linkedin.com/

The program has been stopped.
Please check the reports directory.

The new possible exceptions to be added in the host area are:
- min-api.cryptoCompare.com
- www.linkedin.com
```

Figure 10 Mesajele afișate în terminal în timpul și după terminarea programului

Cu toate acestea, pentru siguranță, programul aşteaptă că fiecare domeniu aflat în intervalul de timp să fie analizat, scurt spus aşteaptă că toate procesele paralele să se termine.

Acest lucru poate crește timpul general de execuție pentru că natura utilitarului Wapiti este de a căuta noi și noi adrese și înte plecând de la adresa URL pe care o dăm că parametru și astfel se poate ajunge la o structură foarte mare derivată.

Aceste argumente pro și contra sunt date tocmai de utilizarea scanerului de vulnerabilități în modul implicit(în care adună posibile zone de testare și din pagini derivate). Se poate limita Wapiti doar la scanarea website-ului dat ca parametru însă asta ar nu ar genera la fel de multă încredere în scanarea securității aplicației web per total.

Cum se instalează?

Programul este scris folosind limbajul de programare Python și de aceea este necesară instalarea lui pe sistemul de operare. Recomandarea este de a instala ambele versiuni (atât Python 2 cât și Python 3).

De asemenea este necesară instalarea administratorului de module Python numit *pip*. Pentru sistemul Windows vine preinstalat cu Python însă pentru sisteme bazate pe Linux trebuie instalat separat.

Înainte de a trece la instalarea modulelor avem nevoie de Wireshark. Aceasta se poate instala în cazul Windows ca un program executabil obisnuit. Pentru Linux se poate instala ca un pachet nou al sistemului de operare.

Având utilitarul pip instalat, putem instala modulele necesare care sunt Pyshark, PyYAML.

Instalarea analizatorului Wapiti se realizează diferit. Pentru Windows, trebuie descărcată arhiva de pe pagina oficială și apoi utilitarul se poate instala, folosind într-un terminal cu drepturi de administrator, cu ajutorul comenzii *python setup.py install*. Pentru sisteme Linux se realizează mai simplu cu ajutorul administratorului de pachete de pe respectiva distribuție. De asemenea pe Kali Linux este preinstalat.

Cum se folosește?

După ce avem toate cele necesare instalate putem folosi True Nature ca un program obișnuit scris în Python. Pentru a-l porni folosim comanda `python TrueNature.py` sau `python3 TrueNature.py` după caz.

Dacă nu avem setată o locație unde să se salveze rapoartele atunci vom primi un mesaj iar programul se oprește. În fișierul de configurare `Config.yml` trebuie setată calea absolută către un director existent. Trebuie să avem grija ca la finalul șirului de caractere să marcăm că este vorba despre un director prin adăugarea unui simbol slash (exemplu `D:\TrueNature\WapifiReports\`).

Dacă avem configurația realizată, programul ne va întreba de pe ce interfață să captureze trafic. După ce alegem, programul începe să lucreze fără a mai fi nevoie de interacțiunea noastră.

Programul se oprește doar manual prin secvența de taste `Ctrl + C`. După oprire se afișează lista de posibile înregistrări pe care manual le putem adăuga în fișierul de configurare pentru a reduce numărul de pachete capturate și pentru a face lucrurile mai ușoare pentru celelalte utilitare implicate.

Concluzii

Plecând de la faptul că True Nature își dorește a fi un program ce realizează analiză de securitate și folosește utilitare doar în acest sens nu trebuie confundat cu un sistem de detectie. El reprezintă o automatizare și o simplificare a unui proces recurrent și realizat în mod normal manual implicând multe utilitare, fiecare cu scop diferit.

Natura elementelor folosite îl face, de asemenea, să poată fi utilizat și în scopuri negative atunci când un atacator nu dorește să se grăbească în alegerea unei anumite ținte. Aici însă procesul consider că ar putea fi realizat mult mai eficient fără implicarea programului True Nature.

De menționat este și faptul că toate tehnologiile și utilitarele folosite sunt în continuă îmbunătățire. Astfel instalarea noilor versiuni ale elementelor folosite implică automat îmbunătățirea programului True Nature. De asemenea, pentru că programul este modularizat putem alege să schimbăm anumite structuri, de exemplu să folosim alt analizator de securitate web sau să realizăm capturarea pachetelor bazându-ne pe un context mai general.

Tabela de figuri

Figure 1 Cerere DNS.....	5
Figure 2 Corp cerere HTTP	7
Figure 3 Corp raspuns HTTP.....	7
Figure 4 Secvență din rezultatul comenzi "tshark -h".....	9
Figure 5 Secvență din rezultatul comenzi "wapiti -h"	11
Figure 6 Conținutul scanării Wapiti în terminal	12
Figure 7 Raport HTML Wapiti	12
Figure 8 Secvență din rezultatul comenzi "skipfish -h"	13
Figure 9 Secvență de alegere a interfeței de lucru	14
Figure 10 Mesajele afișate în terminal în timpul și după terminarea programului.....	15

Surse figuri

Figure 1 - <https://www.cloudflare.com/learning/dns/what-is-dns/>

Figure 2 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

Figure 3 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>

Bibliografie

1. <https://www.cloudflare.com/learning/dns/what-is-dns/>
2. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
3. https://en.wikipedia.org/wiki/URL_redirection
4. <https://www.avast.com/c-sniffer>
5. <https://www.avg.com/en/signal/what-is-sniffer>
6. <https://www.csoonline.com/article/3305805/what-is-wireshark-what-this-essential-troubleshooting-tool-does-and-how-to-use-it.html>
7. <https://wapiti.sourceforge.io>
8. <https://tools.kali.org/web-applications/skipfish>
9. <https://www.wireshark.org/docs/>
10. <https://www.wireshark.org/docs/man-pages/tshark.html>
11. <https://kiminewt.github.io/pyshark/>