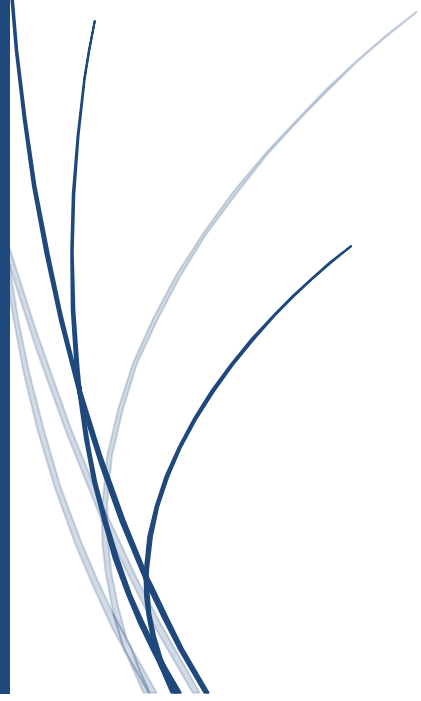
Marzo 2023

Project: FIghting Cybercrime with OSINT (FICO)
WP1: Studio sonde per raccolta dati
Task 1.1 Progettazione dell'ambiente edge
D.1.1.1 Report descrittivo



Prof. Gianluca Reali, Prof. Mauro Femminella
UNIVERSITA' DEGLI STUDI DI PERUGIA

Sommario

Tecnologie di riferimento per la raccolta dei dati in ambito EDGE ***2***

1. Introduzione ***2***

2. Zeek ***3***

2.2 I log di Zeek ***5***

2.3 Cattura dei dati ***6***

2.4 Realizzazione della sonda di cattura del traffico ***7***

3. Analisi dei dati. Il programma RITA. ***8***

4. Integrazione. ***12***

5. Riferimenti Bibliografici ***13***

Tecnologie di riferimento per la raccolta dei dati in ambito EDGE

1. Introduzione

Questo deliverable illustra lo sviluppo di una sonda per la cattura del traffico Internet realizzata in reti locali.
Gli strumenti che si sono stati identificati per la raccolta dei dati e per la successiva analisi sono Zeek [1] e RITA [2] (Real Intelligence Threat Analytics). Questi strumenti sono stati utilizzati in ambito EDGE mediante la realizzazione basata sull'uso di un dispositivo Raspberry Pi. L'analisi dei dati raccolti è stata effettuata in post-processing mediante l'uso di RITA in un sistema Linux Ubuntu 22.04, versione Xubuntu, per leggerezza e versatilità.

Zeek è un software open-source operante in modalità passiva, che viene usato sia per raccogliere traffico IP sia per analizzarlo per la valutazione dei rischi e delle prestazioni.
Il servizio che viene sicuramente utilizzato dalla maggior parte di persone che fanno uso di Zeek è quello che consente di investigare attività sospette o

dannose sulla propria rete. Esso consente di creare dei grandissimi database di log che riportano le attività in corse nella rete, sia benevole sia malevole. Qualunque pacchetto IP scambiato è inserito nei log e poi scritto in un file TSV che può essere elaborato da un altro programma, esterno, o anche dallo stesso Zeek. Dopo una valutazione delle alternative disponibili, la scelta è ricaduta su RITA. Questo programma risulta essere allineato con lo stato dell'arte nell'analisi dei log del traffico di rete.
RITA è sato creato da Black Hills Information Security, nata nel 2016 dopo il decesso della madre del titolare dell'azienda, RITA Strand, dalla quale deriva il nome.
L'idea alla base del funzionamento di RITA è semplice. Esso è un programma che nasce con l'obiettivo di creare una nuova tendenza nell'analisi delle possibili minacce, ossia di fare "Hunt teaming", come è chiamata dal titolare John Strand. Questa consiste nel coinvolgimento di ogni singolo individuo e nel fare squadra per rilevare possibili attacchi informatici o malware. Infatti, data la richiesta facilità di uso di RITA, si ritiene che tutti possano creare una difesa individuale e questo dovrebbe contribuire alla diffusione dell'uso del software per migliorare la difesa globale.

Figura 1: Il logo di Zeek e di RITA

2. Zeek

Lo sviluppo di Zeek è iniziato negli anni '90 da parte di un ricercatore di nome Vern Paxson, presso la Lawrence Berkeley National Laboratory (LBNL). La prima versione venne rilasciata nel 1995 con la denominazione di "Bro" per ricordare il celebre personaggio di George Orwell.
Successivamente. è stato supportato da numerosi istituti e laboratori di ricerca, come il National Science Foundation (NSF) e National Center for Supercomputing Applications (NCSA). Nel 2018 è stato introdotto il nome "Zeek".

Attualmente Zeek e utilizzato da grandi aziende ed associazioni scientifiche per proteggere le propri infrastrutture informatiche. È utilizzato sia come strumento di raccolta, sia per l'analisi del traffico offline.
Il primo beneficio di Zeek consiste nela possibilità di creare dei log dettagliati di ogni connessione di rete sia, operando in modalita wi-fi sia in modalità Ehternet. Questo include anche le possibile interazionicon L'Application layer, cioe tutte le sessioni HTTP con gli URIs corrispondenti, MIME types, risposte del server, richieste al DNS, certificati SSL e sessioni SMTP.
L'impostazione di default di Zeek prevede che tutte le informazioni siano scritte in file TSV (tab separated values) o JSON.
Inoltre, Zeek consente di scegliere un database esterno per immagazzinare, recuperare ed elaborare i dati.
L'architettura di Zeek ad alto livello è semplice, anche se nasconde al suo interno una certa complessità. Esso è architetturalmente stratificato in due grandi componenti, il suo **"Event Engine"**, che riceve in input i pacchetti ricevibili attraverso la propria interfaccia di rete e li riduce in un flusso corrispondente di "eventi" più complessi e ricchi di informazione. L'Event Engine è composto da molti altri sottocomponenti, che sono l' **Input sources**, il **Packet analysis**, il **Session Analysis**, e il **File Analysis**.
Input sources è la componente che ha il compito di ricevere il traffico dalla scheda di rete. il Packet Analysis elabora i pacchetti occupandosi solo dei strati protocollari più bassi, iniziando dal Data Link Layer. Il Session Analysis fa riferimento all'Application Layer, analizzando i protocolli più popolari, come HTTP e FTP. il File analysis analizza il contenuto dei file trasferiti durante le sessioni.
Il secondo macro-componente è lo **"Script Interpreter"**. Esso si occupa di gestire gli eventi gestiti dall'Event Engine tramite un linguaggio di scripting creato appositamente per Zeek. Tramite lo Script Interpreter è possibile ricevere qualsiasi dato da Internet, creare log ed analizzarli.

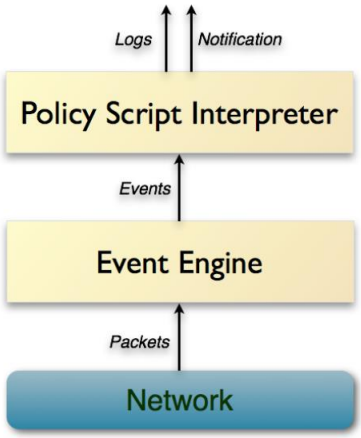


Figura 2: Astrazione dell'architettura di Zeek

2.2 I log di Zeek

Zeek è in grado di generare i seguenti log.

1. conn.log

Il log conn.log e uno dei più importanti e fa riferimenti ai orientati alla connessione, come il Transmission Control Protocol (TCP), invece Zeek tiene traccia di entrambi i protoccoli in questo tipo di log.

2. dns.log

Questa tipologia di log si riferisce allo scambio di traffico con il sistema DNS.

3. http.log

In questi log sono presenti campi relativi agli indirizzi IP e porte utilizzate, ma si possono trovare anche altri campi, come il metodo utilizzato dal protoccolo HTTP e il relativo HOST. Inoltre, sono indicati gli URI utilizzati nella connessione HTTP, lo user agent e i codici di status del protocollo.

4. files.log

Questi log contengono informazioni relative alle attività che prevedono la scrittura su disco.

5. ftp.log

Zeek riassume tutte le attivita che utilizzano l'FTP File Transfer Protocol, cattura le informazioni essenziali e le incapsula in questo log per capire come un client e il server interegiscono utilizzando il protocollo FTP.

6. pe.log

In questi log “pe” sta per "portable executable". La loro importanza sta nel fatto che qualora Zeek crei un log di questo tipo, esso deve essere il primo ad essere analizzato, perché da riferimento ad attività dannose per il dispositivo.

7. traceroute.log

Traceroute.log è un metodo di diagnostica del traffico che serve per individuare i router intermedi tra il proprio indirizzo IP e un destinatario.

8. dhcp.log

Questo tipo di log serve per controllare il corretto funzionamento dei server DHCP.

9. weird.log e notice.log

Weird.log è un log che tiene traccia di tutte le volte in cui l’analisi effettuata dall’Event Engine non è andata a buon fine, magari per interruzione del traffico o magari per errore dei protocolli utilizzati. Notice.log racchiude tutti eventi rilevati da Zeek degni di una inspezione celere.

2.3 Cattura dei dati

Per la generazione dei file di log, la soluzione migliore consiste nell’uso della shell **ZeekControl**. Per accedevi occorre esguire il programma zeekctl, che tipicamente si trova nella directory *bin*, come segue:

```
$ /opt/zeek/bin/zeekctl
```

Quando ci si trova per la prima volta nella shell di zeek, occorre aggiornare la configurazione tramite il comando "install".
La shell consente l’uso di numerosi comandi. Tra i piu utili è presente il comando di “help”. Il comando "status" permette di controllare lo stato dei nodi attivi, e quindi verificare che Zeek stia raccogliendo dati. I comandi "cleanup","quit", e "restart" servono s gestire il riavvio e l’uscita dalla shell. Per avviare la cattura dei dati e scriverli nei logs occorre usare il comando "start", seguito da “stop” per interrompere la cattura.

2.4 Realizzazione della sonda di cattura del traffico

Per la creazione della sonda di cattura del traffico di rete è stato usato un dispositivo Raspberry Pi. In questo modo è stato realizzato un dispositivo a basso costo, portatile e affidabile. La comodità nell'uso di un Raspberry consiste nel poterlo collegare a qualsiasi rete in Internet senza nessun intermediario, come avverrebbe qualora il software sia installato in una macchina virtuale.
Il primo passo per la realizzazione è l'installazione del sistema operativo. In questo caso è stato scelto "RaspBerry Pi OS" cosi da mantenere il sistema leggero e permettere a Zeek di catturare i dati con tutta la massima capacita computazionale disponibile.
Successivamente è stato installato Zeek nel dispositivo ed è iniziata la fase di cattura e di generazione dei log. Per la verifica della funzionalità si e scelto di catturare i dati in intervalli di 6, 12 e 24 ore.

```
[ZeekControl] > install
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
[ZeekControl] > start
starting zeek ...
[ZeekControl] >
```

Figura 3: Fase di avvio della cattura dei dati

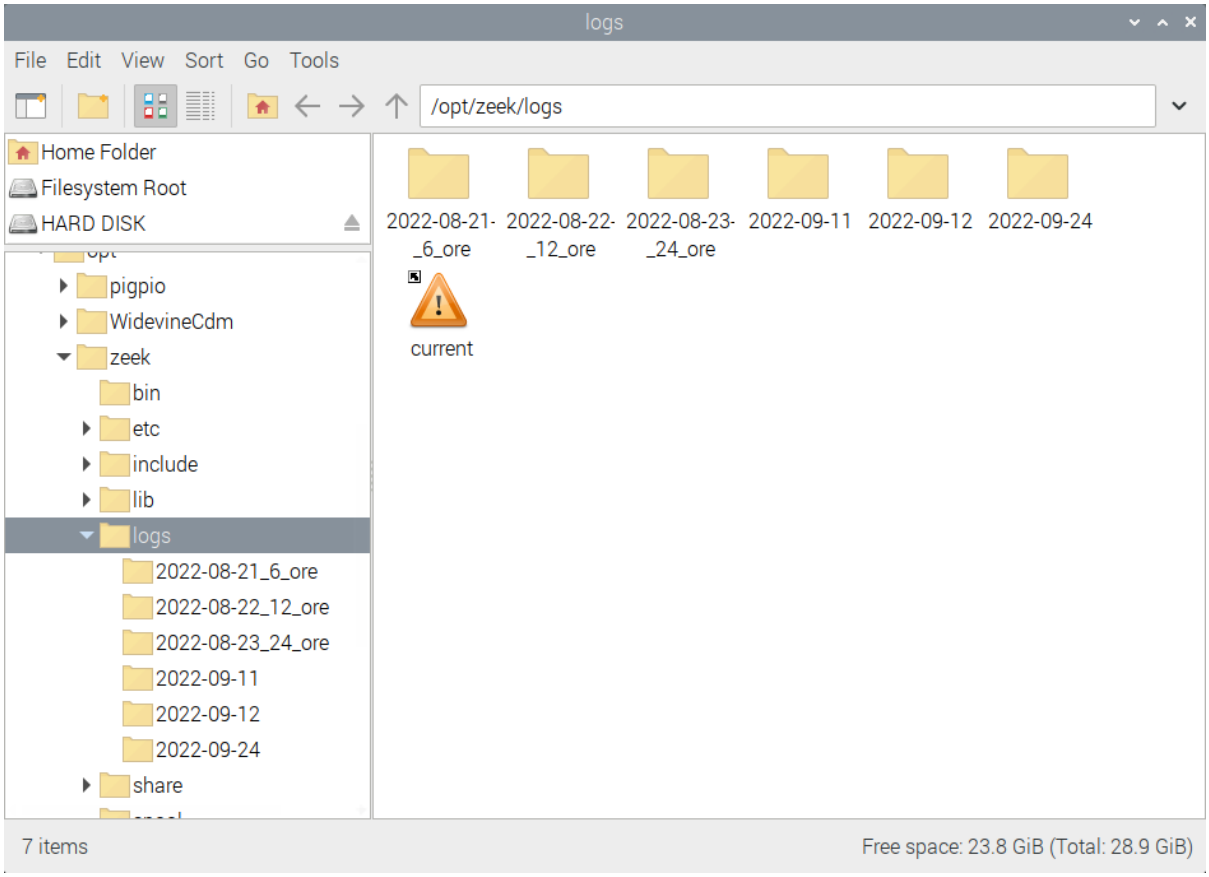


Figura 4: Directory di log generate da Zeek durante 6, 12 e 24 ore.

3. Analisi dei dati. Il programma RITA.

RITA è nata con l'intento di creare un programma in grado di analizzare in modo molto approfondito il traffico IP. Il creatore del programma, John Strand, ha rimarcato la dicotomia fra gli attacchi malware e le relative contromisure, poiché si tenta spesso di creare dei sistemi *statici*, che possono neutralizzare qualsiasi tipo di attacco automaticamente, quando esistono degli studi sulla necessità di rilevare automaticamente e *dinamicamente* i malware, per individuare le loro vulnerabilità. Inoltre, è rimarcata la necessità di non focalizzare l'attenzione ai flussi individuali, come quelli relativi alle connessioni TCP, ma che occorra analizzare flussi di comunicazione a livello più ampio. Proprio per questo è stato implementato RITA.

RITA riceve in input i log di Zeek. Esistono due metodi per creare i logs di Zeek, ossia catturarli direttamente tramite la sua shell, come illustrato in precedenza, o

utilizzare un file PCAP e trasformalo in un log di Zeek tramite il comando dedicato di Zeek "zeek -r".
RITA e include algoritmi di machine learning ed è specializzato per individuare le seguenti tipologie di attacchi informatici:

- **Beacon**
- **DNS Tunneling**
- **Blacklist Checking**

Un beacon consiste in una attivita sospetta che invia dei pacchetti da uno specifico indirizzo IP. RITA, per individuare la presenza di un beacon utilizza diversi metodi di rilevamento. Nella fase iniziale dello sviluppo si riteneva che per elaborare un data-set grande, un algoritmo di machine learning come il K-means clustering fosse la scelta giusta. Successivamente si è passati all'algoritmo MADMOM - median average distribution of the mean - che permette di restituire i beacon come output, tramite alcuni accorgimenti. RITA, attraverso questo algoritmo, accomuna tutti gli indirizzi IP che mettono insieme al meglio le proprietà dei beacon. La prima caratteristica di un perfetto beacon è l'intervallo di invio dei pacchetti. Questo intervallo può essere elevato o ridotto, ma l'aspettp importante è trovare una consistenza nell'intervallo di tempo fra gli invii.
La seconda caratteristica dei beacon è la dimensione dei pacchetti inviati, che tende ad essere poco variabile. La terza e ultima caratteristica per un beacon è il tempo di connessione. Se anche questo risulta pressoché costante, questo è sicuramente una buona indicazione. Infine, un'ultima strategia che RITA utilizza è quella di cercare la consistenza nell'inconsistenza. Ad esempio, se un flusso ha un intervallo medio di invio dei pacchetti di 10 secondi, e risulta che molti pacchetti hanno un tempo variabile di inter-partenza in un intervallo fra gli 8 e i 12 secondi, è possibile ipotizzare che la fonte stia cercando di nascondere il suo comportamento da beacon per aggirare i sistemi di sicurezza. In sintesi, RITA verifica il valore medio del jitter del periodo di invio dei pacchetti.

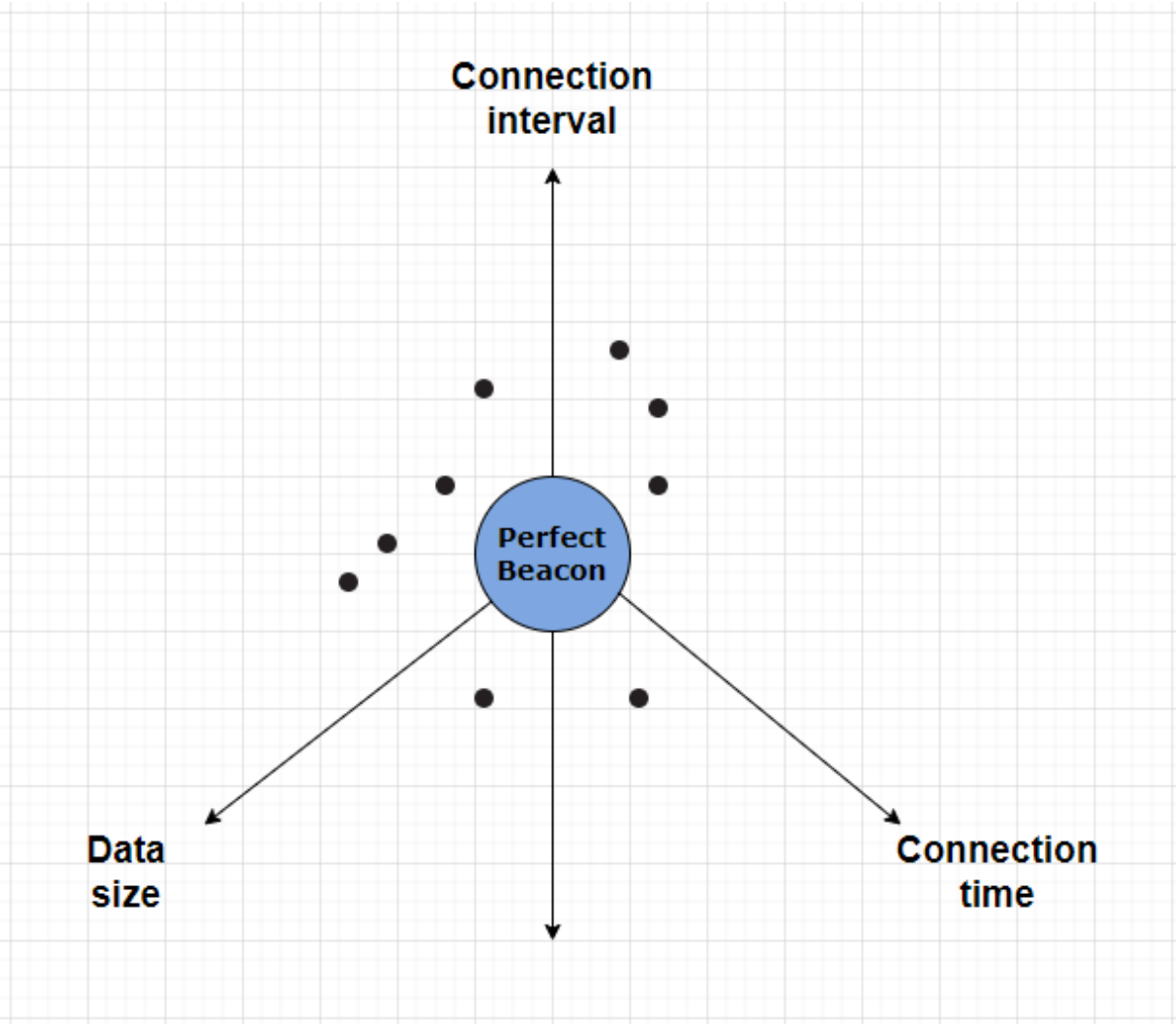
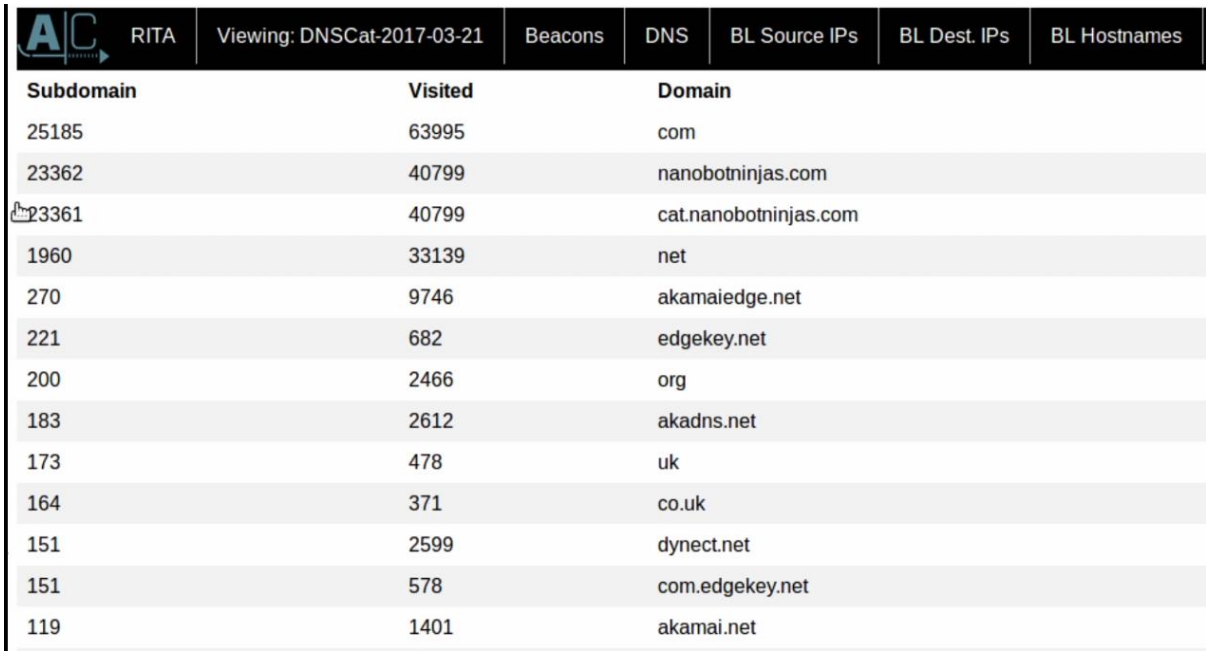


Figura 5: Caratteristiche dei beacon.

Per quanto riguarda il DNS tunneling, questa è una metodologia spesso usata per aggirare le misure di sicurezza di un sistema o un apparato digitale, che consiste nel mimetizzare la trasmissione di pacchetti IP tra due interlocutori tramite server di DNS noti come i DNS di Google (8.8.8.8, o 8.8.8.4). Infatti, in molte organizzazioni, al traffico che è indirizzato ai server DNS di Google è autorizzato il transito per evitare di bloccare qualche sistema importante dell'infrastruttura informatica. Tuttavia, può capitare che dietro a questi scambi si nascondano trasmissioni malevole. Ad esempio, nella figura seguente è mostrato un esempio di questo fenomeno evidenziato da RITA. In particolare, si può notare che un interlocutore "nanobotninja.com" ha effettuato circa 23 000 Domain request, sintomo di un comportamento anomalo. Questo è dovuto al fatto che si puo utilizzare il DNS come comando nascosto per far uscire dalla rete

informazioni delicate, che in questo modo potrebbero passare inosservate. Per perpetrare questo tipo attacco, le Domain request devono essere sempre diverse, poiché altrimenti la richiesta si fermerebbe al sistema DNS locale che risponderebbe con il contenuto memorizzato nella cache.



RITA | Viewing: DNSCat-2017-03-21 | Beacons | DNS | BL Source IPs | BL Dest. IPs | BL Hostnames

Subdomain	Visited	Domain
25185	63995	com
23362	40799	nanobotninjas.com
23361	40799	cat.nanobotninjas.com
1960	33139	net
270	9746	akamaiedge.net
221	682	edgekey.net
200	2466	org
183	2612	akadns.net
173	478	uk
164	371	co.uk
151	2599	dynect.net
151	578	com.edgekey.net
119	1401	akamai.net

Figura 6: Html-report di RITA riguardante i Domain request.

Il terzo obiettivo di RITA è la verifiche che gli indirizzi IP catturati da Zeek non siano presenti in una Black-list. Per questo motivo, mentre analizza i file logs di Zeek RITA cerca di inoltrare molte richieste a Black-list famose per controllare se sta analizzando dati che provengono da siti malevoli.
La configurazione del programma è stata effettuata a partire dalla configurazione base RITA. Se una rete utilizza lo standard RFC1918 riguardante gli indirizzi IP che sono nel range tra 10.0.0.0/8,172.16.0.0/12 e 192.168.0.0/16, non occorre cambiare nulla. Al contrario, occorre modificare il file etc/rita/config.yaml alla voce *Filtering: InternalSubnets*, includendo la propria rete.
Un'altra voce del file "config.yaml" che può essere cambiata è *Filtering: AlwaysInclude*. La funzionalità che si può aggiungere cambiando questa impostazione consiste nell'aggiunta di un DNS server interno, cosi da verificare la sorgente delle richieste DNS.

4. Integrazione

Per creare uno strumento che possa catturare e analizzare il traffico IP facendo uso di Zeek e RITA è stato usato un Raspberry Pi che, come si e già visto, ci ha permesso di catturare i dati tramite Zeek e salvarli nel proprio file system. La problematica che è emersa consiste nel fatto che RITA risulta compatibile con pochi sistemi operativi, e nessuno di questi poteva essere installato su un Raspberry. In particolare, RITA richiedeva non solo una lista che ricomprende solo tre sistemi operativi (Ubuntu 16.04, Ubuntu 20.04 e CentOS), ma anche siano compatibili con l'architettura AMD64. Per risolvere questa problematica è stato deciso di far catturare comunque i dati dal Raspberry tramite Zeek, per poi analizzarli tramite macchina virtuale, collegando il dispositivo mediante ssh.
In questo modo Zeek può catturare i dati direttamente dalla rete, poi RITA può analizzarli offline.
L'ultima fase è stato l'analisi dei dati tramite RITA, e di comprenderne a fondo l' utilizzo. La cattura dei dati di Zeek e stata effettuata in tre test, da 6, 12 e 24 ore.
L'html report di RITA e composto da doverse tab , tra cui "Beacons", "Beacons FQDN", "Beacons proxy", "Strobes", DNS, "BL source IPS", "BL Dest. IPs", "Bl Hostnames", "Long Connections", e "User agents". Sono tutte riguardanti i tre punti focali di RITA. Infatti, le prime tre servono per analizzare i "Beacons" rilevati, invece "DNS", "User Agents" e "Long Connection" servono a rilevare un attacco di "DNS tunneling". Infine "Blsource Ips", "BL Dest Ips" e "BL Hostnames" sono liste di "Black Listing" ricavate da altri server.
Un importante dato riscontrato in tutti i test, sono appunto le tab che riguardano le "Black list", che sono state sempre vuote, confermando appunto che nessun dato e stato scambiato dalla connessione di rete locale verso indirizzi IP già in altre "Black list".
A titolo di esempio si riporta l'analisi che ha generato la quantità maggiore di risultati, effettuata su 24 ore, ossia l'analisi dei beacon in 24 ore.

Score	Source	Destination	Connections	Avg. Bytes	Intvl. Range	Size Range	Intvl. Mode	Size Mode	Intvl. Mode Count	Size Mode Count	Intvl. Skew	Size Skew
0.835	192.168.1.103	24.105.29.76	285	4834.000	860	1967	301	1294	145	216	0.000	0.000
0.834	192.168.1.228	91.189.91.157	43	76.000	16	0	2048	0	31	43	0.000	0.000
0.829	192.168.1.103	13.107.42.12	29	3954.000	7146	2858	3602	1963	16	15	0.000	0.000
0.790	192.168.1.103	20.54.36.229	60	455.000	1612	182	1680	181	28	48	0.000	0.000
0.668	192.168.1.103	108.139.246.176	157	3594.000	5100	946	300	1698	119	53	0.000	-0.951
0.664	87.10.184.109	192.168.1.103	47	44.000	6750	0	463	44	2	47	-0.025	0.000
0.662	87.10.34.190	192.168.1.103	24	44.000	2878	0	3896	44	2	24	0.033	0.000
0.657	192.168.1.103	52.137.110.235	74	3765.000	656	122	901	1880	57	19	0.000	-0.038
0.648	65.108.45.46	192.168.1.103	101	44.000	13563	0	26	44	2	101	0.124	0.000
0.647	183.136.225.42	192.168.1.103	22	44.000	1479	0	3403	44	1	22	0.123	0.000
0.634	69.67.150.36	192.168.1.103	25	44.000	15909	0	581	44	1	25	-0.201	0.000
0.629	87.10.211.124	192.168.1.103	27	44.000	7097	0	252	44	1	27	0.228	0.000
0.627	109.205.213.6	192.168.1.103	22	44.000	5666	0	1400	44	1	22	0.243	0.000
0.623	192.168.1.103	152.199.20.80	26	2935.000	57592	70	1	1014	2	3	0.000	0.000
0.620	87.10.160.84	192.168.1.103	24	44.000	2961	0	1776	44	1	24	-0.283	0.000
0.615	45.93.16.71	192.168.1.103	25	44.000	7849	0	641	44	1	25	0.313	0.000
0.609	192.168.1.103	137.221.105.136	38	4957.000	5925	725	902	1282	4	29	0.333	0.000
0.608	46.101.43.19	192.168.1.103	33	44.000	12647	0	6	44	1	33	0.359	0.000

Figura 7: Analisi dei beacon su 24 ore.

Si può notare che RITA ha rilevato la presenza di numerosi Beacon, con valori di score che vanno da 0.835 a 0.351. Uno score di 0.835 indica che ha rilevato un beacon quasi perfetto. Tuttavia, approfondendo la ricerca su ognuno degli indirizzi IP relativi a valori di alto, questi risultavano appartenenti ad organizzazioni note e non costituiscono un pericolo.

5. Riferimenti Bibliografici

[1] Zeek: An Open Source Network Security Monitoring Tool, https://zeek.org/

[2] Active Countermeasurs. Real intelligence threat analytics (r-i-t-a), 2022. https://github.com/activecm/rita