INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2018, VOL. 64, NO. 2, PP. 197–202
Manuscript received December 26, 2017; revised April, 2018. DOI: 10.24425/119370

Context Data Acquisition Using Adaptive Non-repudiation Model

Marcin Alan Tunia

***Abstract*—The subject of this study is the non-repudiation security service for network communication using TCP/IP stack. Generated evidence, as well as decision-making process of registering a given event, are context-aware. Non-repudiation is equipped with context-awareness by using widely utilized network tools. The aim of this paper is to present timing results for selected tools execution and to complete the evidence generation time. In some applications it is crucial to gather evidence data as fast as possible because of the rapidly changing network environment. For such situations, in case of prolonged execution time, an output from a tool might imprecisely describe the contextual situation from the time of the occurrence of an event.**

***Keywords*—non-repudiation, contextual security, adaptive security, non-repudiation model**

I. Introduction

RECENT advances in computer science lead towards multiservice and ubiquitous networking and computing. The Internet of Things paradigm is being implemented in a wide range of solutions, which causes the presence of high diversity of the machines connected to public and private networks. Security solutions have to evolve to follow these changes and implement advanced protection mechanisms to counteract new threats. One of the modern security properties is context awareness, which allows for building of flexible security services, able to counteract new, unknown threats. The field of context-aware security services is being intensively studied but not in all areas uniformly. Authentication is an example of a widely studied subject in this area. However non-repudiation, which may be considered a stronger authentication, is not studied with equal intensity. In author's opinion there is a need to develop flexible non-repudiation, able to provide context-aware evidence, to meet current development in military [1] and civil systems [2] [3].

Non-repudiation is one of the security services, which is being implemented in wide range of systems, including commercial and military ones. Systems processing confidential data constitute an example. Full accountability is usually required in this case, including reliable evidence generation. Literature provides various definitions of non-repudiation [4] [5] [6] [7]. The definition used for the purpose of this paper is as follows [8]:

Marcin Alan Tunia is with Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland (e-mail: m.tunia@tele.pw.edu.pl).

Non-repudiation is a security service which assures unambiguous and objective post-factum ascertainment that given subject performed given action.

The main aim of non-repudiation service is to generate, store, distribute and verify evidence [9]. Content may vary between different pieces of evidence and depend on requirements for system being protected as well as on non-repudiation policy. Non-repudiation evidence analysis may be a part of post-factum analysis process leading to reconstruction of the most probable scenario of events which might happen. Thus a piece of evidence is the most valuable if it contains the most useful pieces of information about event being investigated. Such useful information may constitute context concerning each entity taking part in the event. According to Dey at al. context may be defined as follows [10]:

Context is any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.

Context data usage for the purpose of security services is investigated by contextual security approach. By utilizing context data, security services are able to provide higher level of flexibility and higher level of security in comparison with standard security services. For non-repudiation, inclusion of additional data in evidence, concerning context of entities taking part in event being registered, may deliver additional information in a process of post-factum analysis. In the next sections of this paper there is described implementation of adaptable context-aware non-repudiation security service and results of contextual data acquisition time are investigated.

II. Adaptive non-repudiation model

In comparison with standard non-repudiation services being implemented in nowadays systems, the new property of non-repudiation presented in this paper is the ability to adapt content of piece of evidence and type of those pieces being generated. Figure 1 presents difference between standard and contextual non-repudiation. Different steps are marked blue. Both types of non-repudiation wait for certain event to happen. Standard service generates evidence with previously defined strict structure, stores it in the repository and verifies if needed. Contextual non-repudiation service verifies context of an event

including involved entities' context, performs context-based risk analysis and according to its results generates evidence with content adequate to current situation. Then pieces of evidence are stored and verified if needed.

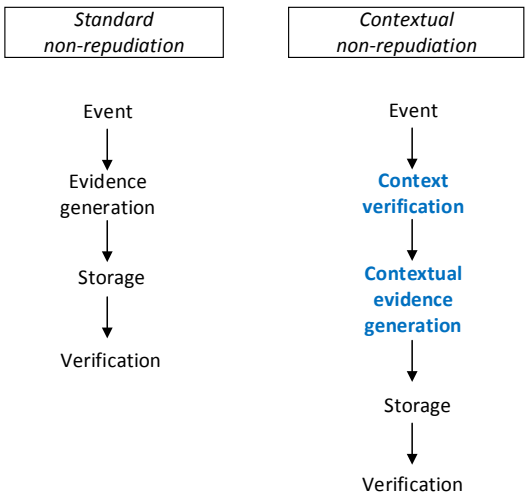

Fig. 1. Standard and contextual non-repudiation comparison

Figure 2 presents flow chart for standard evidence generation process and figure 3 presents contextual evidence generation process. Part of a figure marked in grey rectangle is a contextual part of the process.

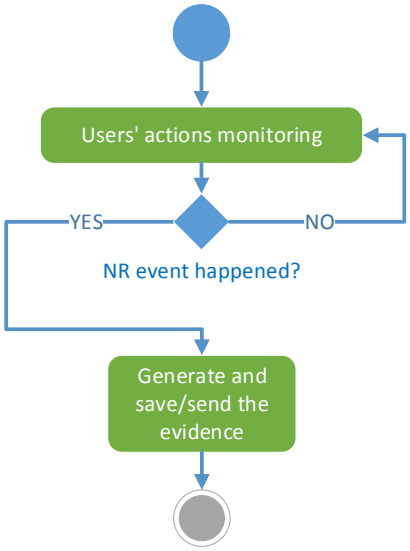

Fig. 2. Standard evidence generation process

The model of non-repudiation used for the purpose of this paper is presented in figure 4. There are two types of main operations: generation of pieces of evidence - G and verification of evidence - W. Evidence generation block has two main inputs: contextual description of an event z and key used to secure a piece of evidence being generated - k_g. A

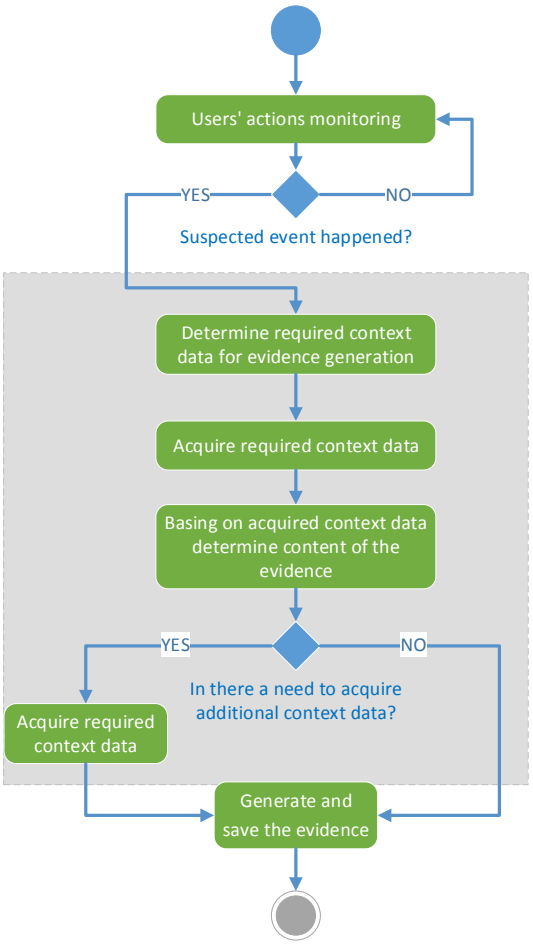

Fig. 3. Contextual evidence generation process

piece of evidence d is generated by block G. This piece can then be verified by block W using security key k_w, which is paired with key k_g. In particular both keys can be the same, for example while using symmetric cryptography based non-repudiation. While using asymmetric cryptography k_g may be considered a private key of a generating entity and k_w may be considered a public key complementary to k_g. The result of verification block operation is binary decision w: 1 - evidence is correct, 0 - evidence is not correct.

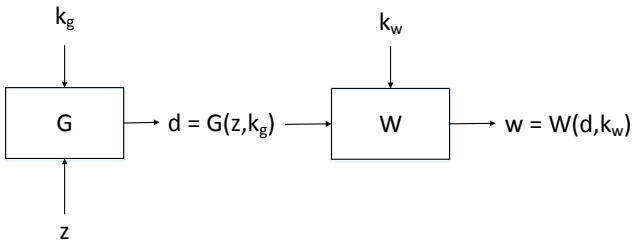

Fig. 4. Non-repudiation simplified model

III. Assumptions for model implementation

The following assumptions were made for the purpose of the implementation being described:

- Non-repudiation security service is based on trusted third party, which generates reliable evidence,
- Non-repudiation security service keeps records on certain network events, which are identified using packets analysis,
- Non-repudiation security service is implemented on a server which is properly protected against unauthorized access (physical and through a network),
- Software running on a server is constantly updated in order to minimize risk which results from bugs lowering security level,
- Cryptographic keys used for securing pieces of evidence are kept in secure manner and are protected against disclosure to unauthorized entities.

IV. Implementation

The architecture of the service is composed of a trusted server for an evidence generation, a port mirrored traffic of protected link, an intrusion detection system acting as a communication context analyzer, a set of tools for information gathering acting as context gathering tools for an evidence content, an evidence-generation engine using XML digital signatures [11] with strong cryptographic algorithms. The mirrored traffic from the protected link is forwarded to a trusted server on which further processing is performed. Packets received by the trusted server are processed by an intrusion detection system engine, which searches anomalies according to the defined signatures. If an anomaly is detected, adequate tools are launched to gather contextual input for an evidence. After the collection of all tools output, the evidence is generated. The figure 5 presents architecture of the implementation. The abbreviation are as follows:

- *MON-NET* - Monitored network for which pieces of evidence are generated,
- *MON-NET* - Public network which is connected to monitored network. This connection is monitored by non-repudiation enabled network equipment,
- *I-IE-NRS* - Interface between port-mirroring enabled network equipment and intrusion detection system (IDS),
- *I-INC* - Interface between IDS and evidence generation module (EGM) used to pass incidents alert from IDS to EGM,
- *I-EG-DB* - Interface between EGM and database used to transfer pieces of evidence to the repository (evidence database),
- *I-CON-PUB* - Interface between EGM and public network. This interface is used by context gathering tools for the purpose of acquisition context about site located in public network,
- *I-CON-MON* - Interface between EGM and monitored network. This interface is used by context gathering tools for the purpose of acquisition context about site located in monitored network.

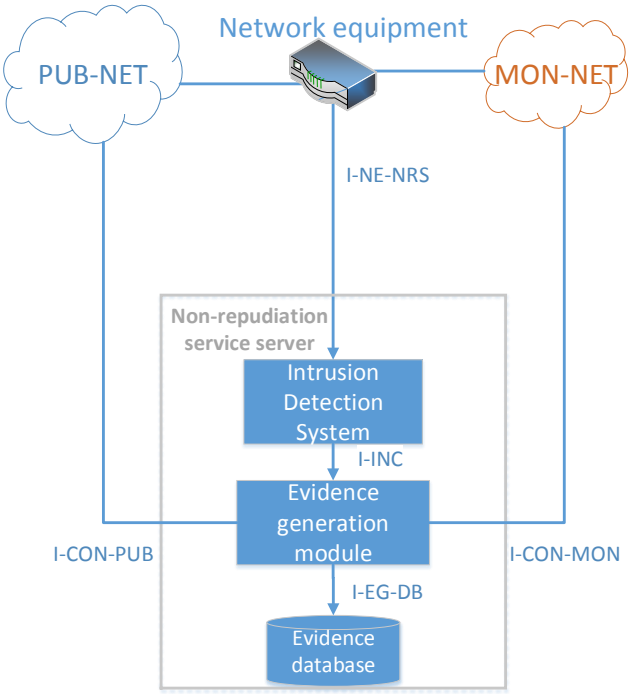

Fig. 5. Implementation architecture

The figure 6 shows example of XML-based piece of evidence, which is being signed by generating site. The elements are as follows:

- Numerical identifier of non-repudiation policy under which the piece of evidence is being generated,
- Alfanumerical identifier of non-repudiation type (e.g. non-repudiation of origin, non-repudiation of submission, non-repudiation of receiving, non-repudiation of transfer),
- The description of an alert, which was generated by an IDS and passed to EGM,
- Context description of a sender of the packet which caused IDS alert. The context is in the form of results from context gathering tools,
- Context description of a recipient of the packet which caused IDS alert. The context is in the form of results from context gathering tools,
- Date and time of evidence generation.

Research was done on the following tools: nslookup [12], nmap [13], whois [14], ping, online geolocalization service ipinfo.io [15] and offline geolocalization service geoiplookup [16]. Table I contains short description of each tool, which was evaluated according to the execution time.

V. Research on evidence generation

On the basis of prepared implementation the research on evidence generation time was conducted. The time of full evidence generation was measured as well as the time of each contextual tool output generation. For each incident alerted by IDS a piece of evidence was generated with all context data available (corresponding maximum risk situation) and another

```
<Evidence>
  <MeritoricalPart>
    <PolicyId>
        Non-repudiation_policy_id
    </PolicyId>
    <NonRepudiationType>
        Non-repudiation_type_id
    </NonRepudiationType>
    <SitesInvolved>
        <Site1>
            Packet_sender_IP_address
        </Site1>
        <Site2>
            Packet_recipient_IP_address
        </Site2>
    </SitesInvolved>
    <EventContext>
        Incident_description_generated_from_IDS
    </EventContext>
    <OriginatingSiteContext>
        Packet_sender_context_data
    </OriginatingSiteContext>
    <DestinationSiteContext>
        Packet_recipient_context_data
    </DestinationSiteContext>
    <GenerationTime>
        Evidence_generation_time
    </GenerationTime>
  </MeritoricalPart>
</Evidence>
```

Fig. 6. Example of XML-based piece of evidence

TABLE I
CONTEXT DELIVERY TOOLS USED IN THE IMPLEMENTATION

Tool name	Description
Nslookup	Tool used to query DNS servers.
ipinfo.io	Tool used to estimate physical location of the host. It requires IP address or URL and queries online localization databases.
geoiplookup	Tool used to estimate physical location of the host like ipinfo.io but it queries local, previously downloaded localization databases.
nmap	Tool used for network exploration and security audits. This network scanner delivers various data (e.g. open ports, running services, used operating system) on devices reachable through a network.
whois	Tool used to query WHOIS databases for information on users and owners of Internet resources.
ping	Tool used to run network diagnostics with ICMP protocol.

one with basic data (corresponding minimum risk situation). The properties of the experiment are as follows:

- Research was conducted in university production environment,
- 780 pieces of evidence were generated and stored,
- Pieces of evidence were being collected during 45 days,
- Platform used for the experiment was Ubuntu 16.04.2 LTS, 16GB RAM, AMD FX(tm)-4170 Quad-Core Processor, 4,2Ghz.

Figure 7 presents obtained results of generation time measurement for a piece of evidence with maximum and minimum risk level. Evidence generation time for minimum risk is approximately constant and the values are from 0,07s to 15,21s with mean 0,55s and median 0,13s. Such low results are a consequence of registering only basic information without deep context inquiry. Pieces of evidence for minimum risk contain only the following context parameters: IPv4 addresses for sender and recipient, local time of evidence generating site, the description of an alert generated by an IDS.

Evidence generation time for maximum risk is diversified and has values from 5,09s to 294,37s with mean 52,31s and median 43,34s. Such spread of values is caused among others by application of tools which produce results in diversified time depending on network conditions as well as on the machines of packets' senders and recipients. This drawback is recompensed with additional context data valuable for post-factum analysis of evidence.

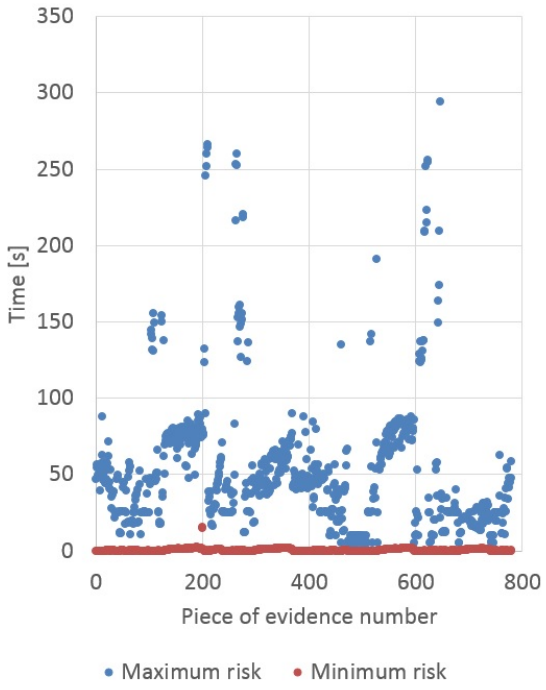

Fig. 7. Evidence generation time for different risk levels

Figures 8 and 9 present results of execution time measurement for maximum risk evidence with a breakdown by each context delivery tool used for sender and receiver of a packet causing IDS alert. Table II presents description of abbreviations used in the figures. Implementation involves parallel invocation of context delivery tools to shorten overall evidence generation time. Thus evidence generation time is shorter than the sum of times of each context delivery tool. Mean value

and median differ significantly for some of tools (e.g. *ping* and *nmap*). It is caused by the following characteristics of research environment:

- 10% of packets generating IDS alerts came from monitored network and were containing target address in public network,
- 90% of packets generating IDS alerts came from public network and were containing target address in monitored network,
- Some machines in monitored network did not respond to ICMP requests, and *ping* tool waited 10 seconds to reach answer timeout,
- Due to law restrictions *nmap* scans were not performed for addresses from outside of monitored network. Thus the *nmap* execution time for that cases was nearly 0 seconds.

TABLE II
EXPLANATION OF ABBREVIATIONS USED IN THE FIGURES

Abbreviation	**Explanation**
Evidence_gen_time	Total evidence generation time
Ping_dst	Time of *ping* tool operation for destination IP address
Nmap_src	Time of *nmap* tool operation for source IP address
Ping_src	Time of *ping* tool operation for source IP address
Nmap_dst	Time of *nmap* tool operation for destination IP address
Nslookup_dst	Time of *nslookup* tool operation for destination IP address using 8.8.8.8 DNS server
Nslookup_dst_local	Time of *nslookup* tool operation for destination IP address using local DNS server
Online_geoIP_dst	Time of *ipinfo.io* tool operation for destination IP address
Offline_geoIP_dst	Time of *geoiplookup* tool operation for destination IP address
Whois_dst	Time of *whois* tool operation for destination IP address
Online_geoIP_src	Time of *ipinfo.io* tool operation for source IP address
Offline_geoIP_src	Time of *geoiplookup* tool operation for source IP address
Nslookup_src	Time of *nslookup* tool operation for source IP address using 8.8.8.8 DNS server
Nslookup_src_local	Time of *nslookup* tool operation for source IP address using local DNS server
Whois_src	Time of *whois* tool operation for source IP address

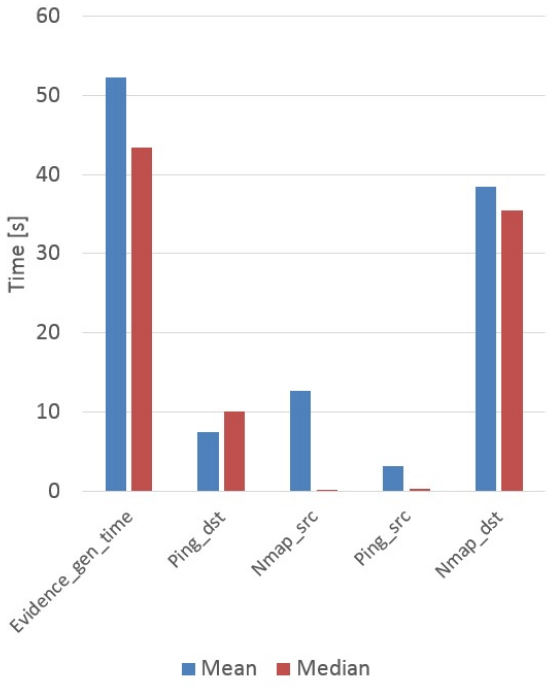

Fig. 8. Evidence generation time and evidence components generation time - part 1

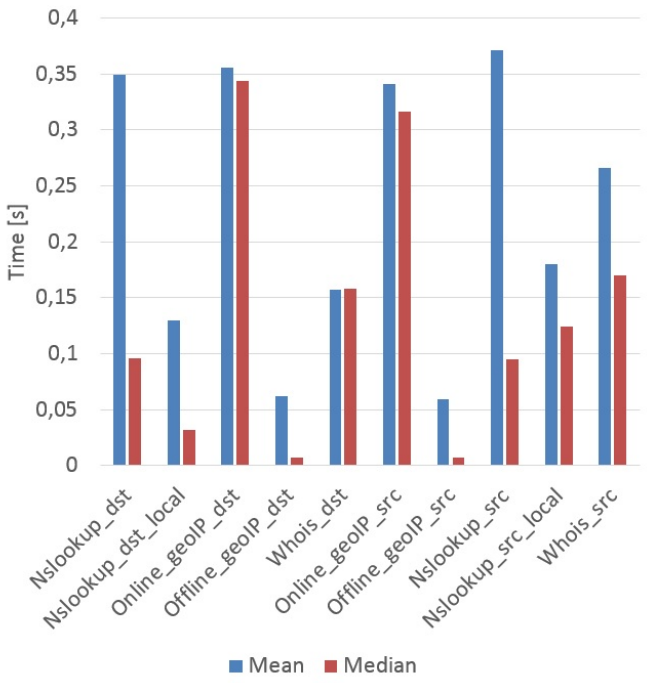

Fig. 9. Evidence components generation time - part 2

VI. CONCLUSION

In this paper the model of context-aware network non-repudiation security service and its practical implementation is presented. The implementation includes intrusion detection system as real-time context analysing tool and other network tools used for context acquisition about sites of the

communication. On the basis of prepared implementation the research on evidence generation time is described. Each tool was evaluated according to measurements made in real life environment.

Contextual non-repudiation may have applications in public sector, where law regulations require reliable and elastic evidence generation in personal data or other sensitive data access control modules. Another field of applications may constitute military and government systems sector, where the strict regulations for classified information access monitoring have to be fulfilled.

REFERENCES

[1] Kott, Alexander, et al. Visualizing the tactical ground battlefield in the year 2050: Workshop report. No. ARL-SR-0327. ARMY RESEARCH LAB ADELPHI MD COMPUTATIONAL AND INFORMATION SCIENCES DIRECTORATE, 2015.

[2] European Comission. "Cybersecurity Strategy of the European Union: An Open, Safe and Secure Cyberspace.", 2013.

[3] European Commission. Decision C(2017)2468 of 24 April 2017. "Horizon 2020, Work Programme 2016 - 2017, 14. Secure societies - Protecting freedom and security of Europe and its citizens", 2017.

[4] Dashti, Mohammad Torabi. Keeping fairness alive. Diss. PhD thesis, Vrije Universiteit, Amsterdam, 2008.

[5] Wu, Wei, et al. "How to achieve non-repudiation of origin with privacy protection in cloud computing." Journal of Computer and System Sciences 79.8 (2013): 1200-1213.

[6] Zhou, Jianying, and Dieter Gollman. "A fair non-repudiation protocol." Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on. IEEE, 1996.

[7] Zhou, Jianying, and Dieter Gollmann. "An efficient non-repudiation protocol." Computer Security Foundations Workshop, 1997. Proceedings., 10Th. IEEE, 1997.

[8] Tunia, Marcin Alan. 2016. Rozpoznawanie kontek-stu na potrzeby usugi niezaprzeczalnoci, Przegld Telekomunikacyjny-Wiadomoci Telekomunika-cyjne, nr 8-9/2016

[9] ISO/IEC,. 2009. Information Technology Security Techniques Non-Repudiation Part 1: General. 13888-1:2009.

[10] Dey, Anind, et al. 2001. "A Conceptual Framework And A Toolkit For Sup-porting The Rapid Prototyping Of Context-Aware Applications". Human-Computer Interaction 16 (2), 97-166.

[11] W3C Recommendation 10 June 2008. "XML Signature Syntax and Processing (Second Edition)", 2008.

[12] Linux Manpages Online nslookup, https://man.cx/nslookup, Accessed: 2017-04-28.

[13] Nmap: the Network Mapper Free Security Scanner, https://nmap.org, Accessed:

[14] Linuxcommand.org whois manual page, http://linuxcommand.org/man_pages/whois1.html, Accessed: 2017-04-28.

[15] IP Address Details ipinfo.io, http://ipinfo.io, Accessed: 2017-04-28.

[16] Linux man page geoiplookup, https://linux.die.net/man/1/geoiplookup, Accessed: 2017-04-28.