APPLYING A DISTRIBUTED ELASTICSEARCH CLUSTER TO INCREASE SECURITY IN IOT DEVICES

by

Spencer C. Riner, BS

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Computer Information Systems

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2021

APPLYING A DISTRIBUTED ELASTICSEARCH CLUSTER

TO INCREASE SECURITY IN IOT DEVICES

by

Spencer C. Riner

APPROVED BY

Andrew Yang, PhD, Chair

Kewei Sha, PhD, Committee Member

Wei Wei, PhD, Committee Member

RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

David Garrison, PhD, Interim Associate Dean

Miguel Gonzales, PhD, Dean

Dedication

For my wife, son, and parents.

Acknowledgments

I have been lucky to receive continuous support, advice, and encouragement during the writing of this thesis.

I would like to express my deepest gratitude to my thesis chair, Professor Andrew Yang, for being a supportive teacher and mentor throughout my education at University of Houston-Clear Lake. I’m thankful for your valuable advice and reassurance while writing this paper.

In addition to Dr. Yang, Professor Wei Wei has also been a great support to me during my education. Thank you for offering constant guidance and advice. My gratitude also goes out to Professor Kewei Sha for serving on the committee for my thesis and offering useful feedback throughout its development.

I would also like to acknowledge each of my colleagues at CACI International and the Johnson Space Center who have allowed me to pursue this goal while continuing to advance my career in information technology.

Finally, I would like to thank dearly my wife Marisa who has lent a sympathetic ear during the writing of this thesis and has never wavered in her support of my goals.

ABSTRACT

APPLYING A DISTRIBUTED ELASTICSEARCH CLUSTER TO INCREASE SECURITY IN IOT DEVICES

Spencer C. Riner
University of Houston-Clear Lake, 2021

Thesis Chair: Andrew Yang, PhD

Internet of Things devices have permeated the daily lives of many by offering useful insights into their daily lives through the use of sensors and cameras to collect and display real-world data. Increasingly, these devices are the targets of cyber attacks that exploit various protocols essential to their functionality such as Telnet and UPnP. Elasticsearch is a distributed indexing system that allows for the horizontal scaling of coordinated node devices to increase their performance. Elasticsearch, along with its companion software elements Logstash and Kibana, are studied to identify their role in increasing the security of Internet of Things by testing the CallStranger UPnP exploit.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I:

INTRODUCTION

Problem Overview

The use of Internet of Things devices in homes and offices has grown considerably in the last ten years as a way of gaining increased information about one's environment. The Amazon Ring product line promises to "protect what matters most at home," suggesting an increased level of home security by using their products [1]. Internet-connected baby monitors, coffeemakers, light bulbs, and alarm clocks all offer a higher degree of control and observation over our environments than ever possible. However, due to a competitive marketplace and a lack of stringent regulation, many IoT devices ship with serious security vulnerabilities that are often open to attack. In Grand Prairie, TX, a couple's Ring security system was hacked as the attacker threatened to "terminate" the owners of the house and gained control of their doorbell to make it appear as if they were currently at the couple's home [2]. A compromised network can also lead to the loss of sensitive personal information like passwords and social security numbers, potentially leading to identity theft.

Because it is unlikely that the companies who manufacturer this category of products will spend the resources needed to make their products resistant to cyber attacks, the responsibility falls to the users of these products to be proactive about preventing attacks that may threaten the safety of their personal space and information.

Purpose of the Study

This study intends to explore the use of the open-source indexing platform Elasticsearch [3] for providing increased security services to IoT devices. The services to be evaluated are confidentiality, data integrity, and data availability. Confidentiality is relevant in an Internet of Things scenario because a successful breach in a home network can lead to the loss of sensitive and private data, such as user names, passwords, and credit card details. One of the attacks observed in this study known as DNS rebinding can easily harvest the personal details of a user by redirecting their web traffic to a compromised server that is indistinguishable from the authentic site it is impersonating [4]. Integrity of IoT device data, both of the data transmitted and the origin of that data, should be acknowledged as well since these devices collect real-world data about the owner’s environment. For instance, an Internet-connected smoke detector that is transmitting invalid data could cause a false positive in the detection of a house fire, or worse, a false negative. Finally, the availability of IoT devices determines if their features are accessible to the user. Internet connected cameras, for instance, may be used to monitor activity at a user’s front door or garage and send alerts if motion is detected. If these devices are not available, their utility is effectively removed.

In addition to the ability of Elasticsearch to provide these security services, the study will also address the effectiveness of the Elasticsearch platform in deterring the CallStranger UPnP attack [5] through the use of an open source attack script called

StrangerCall [6]. The next section reviews the protocols that enable various types of IoT cyber attacks as well as the implementation details of each attack.

CHAPTER II:

LITERATURE REVIEW

IoT Communication Protocols

Telnet

Telnet is a TCP protocol developed in 1969 that allows for remote connection from another computer on the network [7]. A Telnet connection allows for the execution of commands on the target, and requires no authentication from the remote client. Compared to more modern methods of remote access that provide not only authentication, but encryption (such as SSH), Telnet is widely considered to be an insecure protocol and its use has largely been deprecated. Nonetheless, numerous IoT devices still use the Telnet protocol because of its ease of use [8].

Universal Plug and Play (UPnP)

Universal Plug and Play, or UPnP, “enables compatible devices to be connected to a computer network and to be automatically recognized by all the systems connected to the network” [9]. Like Telnet, the use of the UPnP protocol in IoT devices is largely driven by convenience to end users at the expense of security.

UPnP devices utilize the Simple Service Discovery Protocol (SSDP), a “multicast discovery and search mechanism that uses a multicast variant of HTTP over UDP” [10], to allow UPnP devices to broadcast their presence to other devices on the network. According to a recent study reviewing large-scale reflection DDoS attacks on IoT devices, SSDP is often vulnerable to attacks that exploit the SSDP protocol by redirecting the final request for services to a target device to perform a denial of service [11].

After discovery occurs, UPnP devices and their clients use the Simple Object Access Protocol (SOAP) to send XML documents to the device's service over HTTP [10]. SOAP, combined with the General Event Notification Architecture (GENA), enable communication to and from UPnP devices on a network.

Cyberattacks on IoT Devices

Telnet Attacks

The use of Telnet has been largely deprecated in favor of more secure communication protocols like SSH that provide security services like authentication and encryption [12]. The Mirai botnet works by gaining remote access to IoT devices using the Telnet TCP port and a set of preloaded authentication credentials, then shuts down all points of remote access to protect the infected device from other malware [13].

UPnP Attacks

Although UPnP provides convenience to consumers by allowing the automatic discovery of networked devices like printers and cameras, it can open up networks to cyber attacks such as CallStranger [5] or the Mirai botnet [13]. CallStranger was officially classified as CVE-2020-12695 in June 2020 by the National Institute of Standards and Technology with a severity rating of High [14].

Although allowing UPnP requests over a WAN interface is considered to be a misconfiguration and not recommended [15], many routers have this functionality configured by default. The CallStranger attack works by identifying UPnP SUBSCRIBE endpoints and sending large amounts of data to them, potentially leading to a Denial of Service [16].

# How does Mirai exploit UPnP

DNS Rebinding Attacks

DNS rebinding begins by the registration of a domain name by the attacker which is assigned to the IP address of a DNS server owned by the attacker that serves the victim malicious Javascript code [4]. The attacker must configure the DNS server to have a very short time to live (TTL) so that when the victim queries the name server again, the name server is able to return a local IP address of an IoT device on the victim’s network. At this time, the attacker is able to execute arbitrary GET and POST HTTP requests on the victim’s IoT devices, even if they are behind a firewall. This attack is contingent on the attacker knowing the local IP addresses of the victim’s IoT devices in advance. This can be done trivially by the attacker with the use of malicious Javascript code embedded in a website.

Elasticsearch for System Security

Elasticsearch Cluster Architecture

Elasticsearch is a distributed system that allows for functionality if some components have failed [17]. Elasticsearch uses multiple computers called nodes with the Elasticsearch software installed to separate data into discrete pieces known as shards, which are then replicated across multiple nodes. As Elasticsearch logically separates its data into indexes, a shard is a self-contained index. Many shards are then grouped into logical indexes [18] that are displayed to the users. This architecture allows for redundant data within the cluster and provides availability of data. As the data is distributed in multiple physical locations, the implementation must be planned to avoid any kind of discrepancies in the collection of indexed data.

Because Elasticsearch is horizontally scalable by adding numerous client machines, there is a hierarchy of node roles for dividing responsibilities within the cluster. In every cluster, there must be a master node. The master node is responsible for index creation and deletion, tracking cluster members, and assigning shards [19]. However, the cluster must also be able to respond intelligently to unexpected loss of a cluster member. In his presentation Elasticsearch Best Practice Architecture, Eric Westburg notes that a cluster must be configured only to elect a master with a quorum [20] to prevent “split brain”. Split brain refers to a situation where two masters are elected without knowing the other exists. In a cluster with three members, this means a new master may only be elected if two master-eligible nodes agree.

Central Log Management

Elasticsearch is well-suited as an aggregator of system logs. The Logstash component of the Elasticsearch software stack can be configured to accept various file types, including plaintext, HTTP events, Syslog messages in RFC3164 format, TCP and UDP events, and IMAP mail messages [21]. As Logstash receives these inputs, it parses the chosen content from these various message types using Filters. An output may then be configured to enable the transmission of the collected data to the Elasticsearch cluster for storage and indexing [22]. As the logs are collected, they can be further filtered, reviewed, and graphed by network administrators using the Kibana web interface. This increased visibility may result in faster identification of network-based threats occurring on a given network.

CHAPTER III:

RESEARCH METHODOLOGY

Introduction

To better understand the potential security services Elasticsearch is able to offer a group of IoT devices, an experiment on a combination of physical and virtualized hardware will be performed using Raspberry Pi devices and the open-source hypervisor VirtualBox. The following sections provide details on the design of the experiment.

Logical Design

Client Devices

The first device discussed is the “attacker” machine. This is the machine that will perform three common IoT attacks on each of the relevant sensor devices. Additionally, a “gateway” machine will be deployed for routing traffic between disparate subnets. In order to simulate the environment of a typical device targeted by the Mirai botnet, the gateway device will forward TCP port 23 to the associated sensor device to simulate Telnet accessibility. A total of six virtualized sensor devices will be deployed using ns-3: three will be connected directly to the gateway machine, and three will be connected to the Elasticsearch cluster.

Network

In order to insulate the environment from any real-world attacks, the experiment is performed using private, non-routable subnets to logically separate the devices.

192.168.1.0/24 contains the attacker virtual machine as well as the gateway. This subnetwork is meant to represent the Internet and a standard routable IP address a home’s router may receive from an Internet provider.

10.10.10.0/24 contains a control device connected directly to the gateway machine. This subnet represents a standard IoT deployment without any intermediate services on a typical local area network in a home or office.

10.10.20.0/24 also contains an identical sensor device, but its communication is routed through an Elasticsearch cluster as a proxy.

See Fig. 1 for an overview of the logical design.

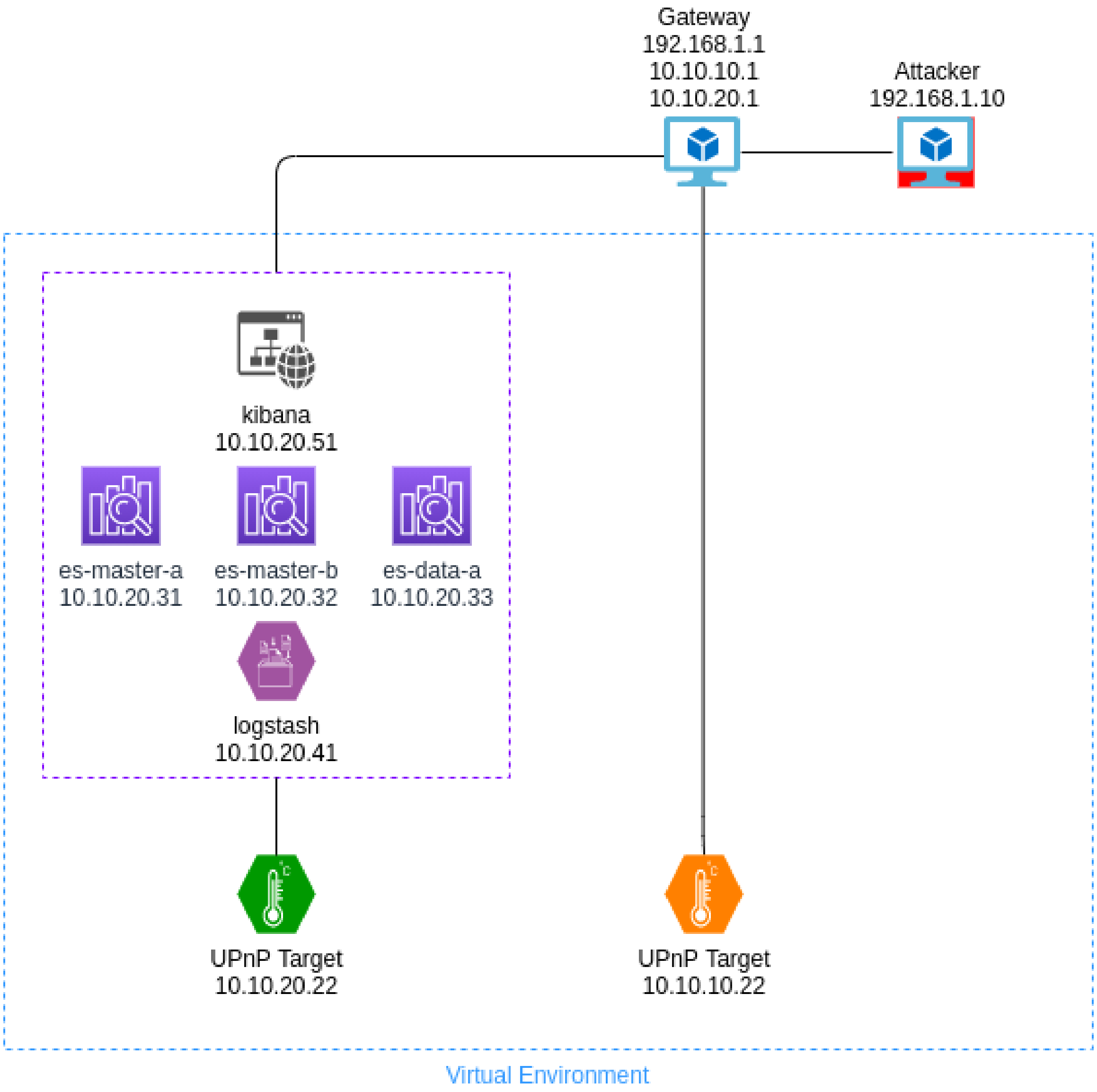

Figure 1: Logical Design of IoT Attack Environment

Elasticsearch Architecture

Elasticsearch is one component in a multi-application architecture known as the ELK stack. This stack contains three primary components: Elasticsearch for indexing; Logstash for log parsing; and Kibana as a web front end. Because Elasticsearch functions as a distributed cluster, 3 of these machines will be dedicated solely to Elasticsearch operations. The cluster configuration of Elasticsearch allows it to be fault-tolerant to unexpected loss of a cluster member. To maintain that property, there must be at least three to resolve any potential split-brain conflicts [20] that lead to inconsistent data between any given node. One virtual machine will be used for Logstash and Kibana each.

Physical Design

Raspberry Pi Devices

Two Raspberry Pi devices are used to perform attacker and routing duties during the experiment. Each Raspberry Pi uses Ubuntu 20.04 LTS as an operating system.

Virtual Machines

The virtual machines used in this experiment are deployed on local test hardware using the open-source hypervisor VirtualBox. Each virtual machine uses Ubuntu 20.04 LTS as an operating system.

Attack Execution

UPnP Attack Using StrangerCall Honeypot

To assess the target devices' vulnerability to CallStranger, the experiment uses an open source honeypot provided by Slovak National CSIRT team (SK-CERT) called StrangerCall [6] that detects vulnerability to the CallStranger attack that targets UPnP devices [5] and simulates a test attack using a Python script. First, the code repository is copied to each participant in the simulation: the control target machine, the Elasticsearch-enabled target machine, and the attacker machine. After compilation and installation of the code, each target machine runs a Python script, hon_upnp.py, that opens a socket on TCP port 1784 and accepts SUBSCRIBE UPnP events.

After successful execution of the listener script, the attacker machine runs a Python script called test_hacker.sh that accepts an IP address and TCP port number as arguments. On successful connection, the test_hacker.sh script outputs log messages to standard out indicating the endpoint was accessed, and the attack simulation has succeeded.

Attack Monitoring

Zeek

In addition to the StrangerCall honeypot, the open-source monitoring tool Zeek [23] runs on the gateway machine to monitor UPnP traffic that occurs on the network. Zeek is a packet collection tool that passively listens to network traffic on a network interface. The gateway machine is suited to this application, as it has network interfaces on all applicable subnetworks. In addition to the standard Zeek functionality, an extension that detects CallStranger exploitation attempts called CallStranger-Detector [24] is installed on the gateway.

The CallStranger-Detector attempts to identify exploitation attempts by monitoring for UPnP SUBSCRIBE events and UPnP NOTIFY events. In the log output, four notice types are possible:

- CallStranger_Data_Exfiltration_Attempt
- CallStranger_Data_Exfiltration_Success
- CallStranger_UpnP_Request_Callback_To_External_Host
- CallStranger_UpnP_To_External_Host

The presence of any of these notices will indicate a potential CallStranger exploitation and further support the output of the test_hacker.py script.

Measurement of Results

UPnP Attack Using StrangerCall HoneyPot

To gauge the effectiveness of the Elasticsearch cluster in preventing the CallStranger attack, a combination of the output of the test_hacker.sh script and the Zeek network monitoring logs will be used to establish a baseline. The attack will first be executed on the target with IP address 10.10.10.22 that is not connected to the Elasticsearch cluster. For the baseline to be established, the simulated attack on that host must provide positive feedback and indicate a successful connection attempt. Additionally, the Zeek log output should provide at least one of the previously identified notice types. To establish the Elasticsearch cluster as a feasible solution to the CallStranger attack, the attack executed on the target with IP address 10.10.20.22 using the Elasticsearch cluster as a proxy must exhibit no positive output when performed.

CHAPTER IV:

EXPERIMENTAL RESULTS

Baseline Results

StrangerCall Honeypot Script Output

When running hon_upnp on the baseline target host at 10.10.10.22, the following output is observed when initiating the script:

```
hon_upnp INFO     Started UPnP listener on 0.0.0.0:1784
```

This message indicates the successful initiation of the listener script. When running the test_hacker.sh script on the attacker machine at 192.168.1.10, the following results are displayed:

```
*   Trying 10.10.10.22:1784...
* TCP_NODELAY set
* Connected to 10.10.10.22 (10.10.10.22) port 1784 (#0)
> SUBSCRIBE / HTTP/1.1
> Host: 10.10.10.22:1784
> User-Agent: curl/7.68.0
> Accept: */*
> NT: upnp:event
> TIMEOUT: Second-180
> CALLBACK: <http://10.0.0.1>
>
```

This series of message indicated a successful connection from the attacker machine and serves as the baseline condition for a successful attack.

Zeek Monitoring Output

The Zeek monitoring tool is initialized using the zeekctl command line utility. When this is executed, the gateway machine passively listens to all network traffic that occurs on its Ethernet network interface, eth0. When testing using the test_hacker.sh script, Zeek does not observe any of the four specified log messages in notice.log that indicate a CallStranger exploitation attack. However, this is a false negative due to the tool ignoring UPnP activity that occurs on a non-routable subnetwork.

In lieu of this, we observe all traffic that occurs on port 1784 of the baseline target machine, including several SUBSCRIBE requests. The truncated output is shown below:

```
192.168.1.10    37240   10.10.10.22     1784    tcp     http
18.438253       151     0       SF      T       T       0       ShADafF
192.168.1.10    37242   10.10.10.22     1784    tcp     http
18.388555       151     0       SF      T       T       0       ShADafF
192.168.1.10    37244   10.10.10.22     1784    tcp     http    8.540421
151     0       SF      T       T       0       ShADafF
192.168.1.10    37240   10.10.10.22     1784    1       SUBSCRIBE
10.10.20.1      /       -       -       curl/7.68.0
192.168.1.10    37242   10.10.10.22     1784    1       SUBSCRIBE
10.10.20.1      /       -       -       curl/7.68.0
192.168.1.10    37244   10.10.10.22     1784    1       SUBSCRIBE
10.10.20.1      /       -       -       curl/7.68.0
```

These messages indicate a successful connection on port 1784 from the attacker machine to the target machine and provide another metric for establishing a baseline of what a successful attack looks like when monitoring with Zeek.

Mitigation Using Port Forwarding

The strategy employed in this experiment for using Elasticsearch for CallStranger vulnerability mitigation is known as TCP port forwarding. Using this approach, the router specifies an alternate destination for incoming TCP requests on a given host; in this case, TCP port 1784. In most home networks, the entire local area network is assigned a single IP address by the service provider, and additional addresses for local devices are assigned using technology called Network Address Translation [25]. For our purposes, we will assume this is the case. Because the attacker is only able to access the public IP address, they are reliant on the configuration of the network's router. In this manner, the Elasticsearch cluster acts as a shield against any TCP-based UPnP attacks such as CallStranger.

Gateway Port Forwarding Configuration

Through the use of the iptables command line utility on the gateway server, a prerouting rule is added to the server's firewall configuration. In the previous phase of the experiment, we assume that the attacker had direct access to the vulnerable port by referring directly to its local IP address. In a real-world scenario, the local IP address would be concealed through the use of NAT. In order to simulate an Internet-facing UPnP device, we perform an initial test and forward TCP port 1784 from the gateway to the target host using the commands below:

```
/usr/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp -d 10.10.20.1 --
dport 1784 -j DNAT --to 10.10.20.22:1784
```

This new configuration may be tested by rerunning the baseline test phase. The results of the attack script when specifying the gateway's IP address can be seen below:

```
*   Trying 10.10.20.1:1784...
* TCP_NODELAY set
* Connected to 10.10.20.1 (10.10.20.1) port 1784 (#0)
> SUBSCRIBE / HTTP/1.1
> Host: 10.10.20.1:1784
> User-Agent: curl/7.68.0
> Accept: */*
> NT: upnp:event
> TIMEOUT: Second-180
> CALLBACK: <http://10.0.0.1>
>
```

CallStranger specifically targets devices that are exposed on the network, so this configuration assumes that an end user has allowed for this on their home network. After verifying this attack is still successful with the use of port forwarding, we then modify our firewall configuration to redirect requests for port 1784 to the Elasticsearch aggregator at 10.10.20.41.

```
/usr/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp -d 10.10.20.1 --
dport 1784 -j DNAT --to 10.10.20.41:1784
```

Using this configuration, the UPnP TCP socket is still accessible to the internal LAN, but any attempts to access it from outside of the local network boundary is met with redirection. This retains the functionality of a given UPnP device.

Elasticsearch Results

StrangerCall Honeypot Script Output

After performing the port forwarding configuration on the gateway as described above, the attack is once again performed using the attacker machine at 192.168.10.1 with a target of 10.10.20.1 at TCP port 1784. The output of the attacker is below:

```
*   Trying 10.10.20.1:1784...
* TCP_NODELAY set
* Connection timed out after 10001 milliseconds
* Closing connection 0
curl: (28) Connection timed out after 10001 milliseconds
```

The correlating target system log output is also provided:

```
2021-04-02 18:20:22,869 hon_upnp INFO     Started UPnP listener on
0.0.0.0:1784
```

As shown by these messages, the test_hacker.sh script is no longer able to access TCP port 1784 on the target sharing a subnetwork with the Elasticsearch cluster.

While monitoring the network traffic with Zeek, no activity was witnessed traveling to or from the target in the Elasticsearch environment.

CHAPTER V:

DISCUSSION

Elasticsearch as a Security Provider

While the results of the experiment employing a CallStranger honeypot did result in the attack test failing to connect, these results don’t necessarily support the use of an Elasticsearch cluster as a suitable tool for the prevention of an Internet of Things cyberattack like CallStranger. In fact, the Elasticsearch cluster itself is a rather arbitrary element in this network configuration; the same effect could be attained using a simple firewall device or properly-configured Linux server.

Despite these findings, Elasticsearch may still be able to be applied as a security service in a different context. The primary role and purpose of Elasticsearch is for indexing and replicating recurring data across a distributed cluster. The following sections outline the use of Elasticsearch for providing two of the essential security services: data availability and data integrity.

Elasticsearch for Confidentiality

Regarding data security, the principle of least privilege states that “access to the information should be granted only on a need-to-know basis...and thus should not be accessible to everyone” [26]. While this is commonly accepted practice in a business context, individuals have the same motive to ensure the confidentiality of their own data. Public data breaches enacted on large companies such as Equifax, Sony, Adobe, Target, and Home Depot [27] [28] have become a common occurrence and are almost accepted as inevitable. These breaches can expose the personal information of millions of users each time they happen. Although users of IoT devices may assume that the traffic and data generated by each sensor device is retained on their own local network, that in many cases is not true. Amazon Ring devices report all of their camera data up to a central, Amazon-owned server. While best practices exist for securing online cameras such as 2 Factor Authentication and routinely applying software and firmware updates [29], Elasticsearch may be a suitable solution for ensuring the confidentiality of data collected by IoT devices on a home network. By sending the collected data directly to an data processor such as Logstash, the data can be retained on a local server and never be exposed to another party or the Internet at all. Furthermore, users may further increase the security of their data by leveraging Transport Layer Security (TLS) to encrypt communication between Elasticsearch nodes.

Elasticsearch for Data Availability

Data availability refers to “making sure that the services that are provided by an organization are available” [26] and may include enterprise-oriented products like websites and databases. However, this principle may also be applied to the home network. If a user relies on IoT devices to get real-time information about the physical characteristics of their home, such as temperature or the status of an alarm system, they may want to take extra steps to ensure that data is available. By exporting the system logs from a series of IoT devices to an aggregate log collector such as Elasticsearch, the duty of log availability is outsourced to an external service. Because Elasticsearch functions as a distributed cluster, availability is particularly high due to the replication abilities of the software stack. Even if one cluster member is offline, for instance, another member can display its data independently.

Elasticsearch for Data Integrity

Data integrity makes “sure that the data is not tampered either through unauthorized, intentional, or accidental changes” [26]. Though the reviewed IoT cyberattacks primarily function by executing denial of service attacks that affect availability, the integrity of the data should be considered. A networked IoT device may collect potentially sensitive data, such as a video feed of the home or personally identifiable information. Elasticsearch provides various methods of verifying the integrity of its own stored data. First, simple HTTP queries may be performed directly against the server, observing the return code to determine if a given document exists or not [30]. Additionally, Elasticsearch is able to identify missing records through the use of integer-based keys that are configured by default [30]. By iterating through each key and identifying which contain data using a bucket selector, the results will show each empty record. This method of data verification can be used by the Elasticsearch administrator through the use of a command line utility such as curl.

Benefits of IoT Log Aggregation

In his paper “Stopping IoT-based Attacks on Enterprise Networks”, G. W. Ray Davidson emphasizes the importance of a centralized and integrated logging solution for enhacing the security of networked IoT devices [31]. By using logging tools such as syslog and syslog-ng, IoT devices can reliably transmit their log data to a centralized location such as an Elasticsearch cluster through the use of a Logstash input plugin. Davidson also mentions the importance of monitoring the activity of IoT devices on a network to better create “potential threat models and defenses” [31]. Monitoring the data stored in an Elasticsearch cluster is made much simpler through the use of the Kibana web interface, one of the provided components of Elasticsearch’s integrated software stack. Kibana uses a custom query language called Kibana Query Language (KQL) [32] that allows for the filtering of log messages by TCP port, message contents, HTTP return codes, and various other formats.

CHAPTER VI:

FUTURE WORK

In this thesis, we observed the effects of an Elasticsearch implementation using minimal hardware on a UPnP-based cyber attack called CallStranger. Some experiments have been left to the future due to the distributed nature of Elasticsearch and its high compute power requirements. With an Elasticsearch cluster built on bare metal hardware with multicore processors available, an experiment may be conducted that tests the ability of commercially available IoT sensors to send real data to Elasticsearch for aggregation and indexing. Another approach may be to modify IoT devices that are configured to communicate with the Internet directly (such as Ring cameras and doorbells) to send their data to Elasticsearch, either using Elasticsearch as an intermediary or sending a copy of the data to Elasticsearch.

BIBLIOGRAPHY

[1] "About | Ring." https://shop.ring.com/pages/about (accessed Sep. 06, 2020).

[2] M. Beedham, "Amazon Ring owners foil $400K Bitcoin extortion plot by removing batteries," *TheNextWeb.com [BLOG]*, Dec. 12, 2019. https://libproxy.uhcl.edu/login?url=https://www.proquest.com/docview/2324850598?accountid=7108.

[3] "You Know, for Search," *Elastic Blog*, Feb. 08, 2010. https://www.elastic.co/blog/you-know-for-search (accessed Mar. 14, 2020).

[4] G. Acar, D. Y. Huang, F. Li, A. Narayanan, and N. Feamster, "Web-based Attacks to Discover and Control Local IoT Devices," in *Proceedings of the 2018 Workshop on IoT Security and Privacy*, Budapest, Hungary, Aug. 2018, pp. 29–35, doi: 10.1145/3229565.3229568.

[5] "CVE-2020-12695: CallStranger Vulnerability in Universal Plug and Play (UPnP) Puts Billions of Devices At Risk," *Tenable®*, Jun. 08, 2020. https://www.tenable.com/blog/cve-2020-12695-callstranger-vulnerability-in-universal-plug-and-play-upnp-puts-billions-of (accessed Oct. 29, 2020).

[6] *SK-CERT/strangercall*. SK-CERT, 2021.

[7] "TELNET | High Definition: A-Z Guide to Personal Technology - Credo Reference." https://search-credoreference-com.libproxy.uhcl.edu/content/entry/hmhighdef/telnet/0 (accessed Jul. 22, 2020).

[8] U. D. Gandhi, P. M. Kumar, R. Varatharajan, G. Manogaran, R. Sundarasekar, and S. Kadu, “HIoTPOT: Surveillance on IoT Devices against Recent Threats,” *Wirel. Pers. Commun.*, vol. 103, no. 2, pp. 1179–1194, Nov. 2018, doi: 10.1007/s11277-018-5307-3.

[9] Bryan Pfaffenberger, “Universal Plug and Play (UPnP),” *Webster's New World™ Computer Dictionary*. Houghton Mifflin Harcourt, Accessed: Jul. 22, 2020. [Online]. Available: https://libproxy.uhcl.edu/login?url=https://search.credoreference.com/content/entry/webstercom/universal_plug_and_play_upnp/0?institutionId=7275.

[10] Andrew Donoho, Bryan Roe, Maarten Bodlaender, John Gildred, Alan Messer, YoonSoo Kim, Bruce Fairman, Jonathan Tourzan, “UPnP Device Architecture 2.0.” Open Connectivity Foundation, Inc., Apr. 17, 2020, Accessed: Apr. 11, 2021. [Online]. Available: https://openconnectivity.org/upnp-specs/UPnP-arch-DeviceArchitecture-v2.0-20200417.pdf.

[11] Y. Lee, H. Chae, and K. Lee, “Countermeasures against large-scale reflection DDoS attacks using exploit IoT devices,” *Automatika*, vol. 62, no. 1, pp. 127–136, Jan. 2021, doi: 10.1080/00051144.2021.1885587.

[12] “Telnet,” *Wiley Dictionary of Communications Technology*. John Wiley & Sons, 1998, Accessed: Jul. 22, 2020. [Online]. Available: https://libproxy.uhcl.edu/login?url=https://search.credoreference.com/content/entry/wileycommtech/telnet/0?institutionId=7275.

[13] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and Other Botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.

[14] “NVD - CVE-2020-12695.” https://nvd.nist.gov/vuln/detail/CVE-2020-12695#vulnCurrentDescriptionTitle (accessed Apr. 01, 2021).

[15] “CERT/CC Vulnerability Note VU#357851.” https://www.kb.cert.org (accessed Apr. 01, 2021).

[16] “CERT/CC Vulnerability Note VU#339275.” https://www.kb.cert.org (accessed Apr. 01, 2021).

[17] “Designing for resilience | Elasticsearch Reference [7.9] | Elastic.” https://www.elastic.co/guide/en/elasticsearch/reference/current/high-availability-cluster-design.html (accessed Oct. 11, 2020).

[18] “Scalability and resilience: clusters, nodes, and shards | Elasticsearch Reference [7.9] | Elastic.” https://www.elastic.co/guide/en/elasticsearch/reference/current/scalability.html (accessed Oct. 11, 2020).

[19] “Node | Elasticsearch Reference [7.9] | Elastic.” https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html (accessed Oct. 11, 2020).

[20] Eric Westburg, “Elasticsearch Best Practice Architecture,” Accessed: Oct. 11, 2020. [Online]. Available: https://www.elastic.co/pdf/architecture-best-practices.pdf.

[21] “Input plugins | Logstash Reference [7.12] | Elastic.” https://www.elastic.co/guide/en/logstash/current/input-plugins.html (accessed Apr. 02, 2021).

[22] “How Logstash Works | Logstash Reference [7.12] | Elastic.” https://www.elastic.co/guide/en/logstash/current/pipeline.html (accessed Apr. 02, 2021).

[23] “The Zeek Network Security Monitor,” *Zeek*. https://zeek.org/ (accessed Apr. 01, 2021).

[24] *corelight/callstranger-detector*. Corelight, Inc., 2020.

[25] D. Wing, “Network Address Translation: Extending the Internet Address Space,” *IEEE Internet Comput.*, vol. 14, no. 4, pp. 66–70, Jul. 2010, doi: http://dx.doi.org/10.1109/MIC.2010.96.

[26] Zeal Vora, *Enterprise Cloud Security and Governance*. Packt Publishing, Limited.

[27] T. Kude, H. Hoehle, and T. A. Sykes, “Big data breaches and customer compensation strategies,” *Int. J. Oper. Prod. Manag.*, vol. 37, no. 1, pp. 56–74, 2017, doi: http://dx.doi.org/10.1108/IJOPM-03-2015-0156.

[28] E. Weise, “Equifax breach: Is it the biggest data breach?,” *USA TODAY*. https://www.usatoday.com/story/tech/2017/09/07/nations-biggest-hacks-and-data-breaches-millions/644311001/ (accessed Apr. 18, 2020).

[29] Glenn Fleishman, *Taking Control of Home Security Cameras*. Take Control Books, 2021.

[30] “Elasticsearch: Verifying Data Integrity with External Data Stores,” *Elastic Blog*, Aug. 08, 2016. https://www.elastic.co/blog/elasticsearch-verifying-data-integrity-with-external-data-stores (accessed Apr. 02, 2021).

[31] G. W. R. Davidson, “Stopping IoT-based Attacks on Enterprise Networks,” p. 9.

[32] “Kibana Query Language | Kibana Guide [7.12] | Elastic.” https://www.elastic.co/guide/en/kibana/current/kuery-query.html (accessed Apr. 02, 2021).