NeuroMesh: IoT Security Enabled by a Blockchain Powered Botnet Vaccine

Gregory Falco
gfalco@mit.edu
Computer Science and Artificial Intelligence Laboratory (CSAIL),
Massachusetts Institute of Technology
Cambridge, Massachusetts

Caleb Li
Sloan School of Management,
Massachusetts Institute of Technology
Cambridge, Massachusetts

Pavel Fedorov
Sloan School of Management,
Massachusetts Institute of Technology
Cambridge, Massachusetts

Carlos Caldera
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology
Cambridge, Massachusetts

Rahul Arora
Sloan School of Management,
Massachusetts Institute of Technology
Cambridge, Massachusetts

Kelly Jackson
Sloan School of Management,
Massachusetts Institute of Technology
Cambridge, Massachusetts

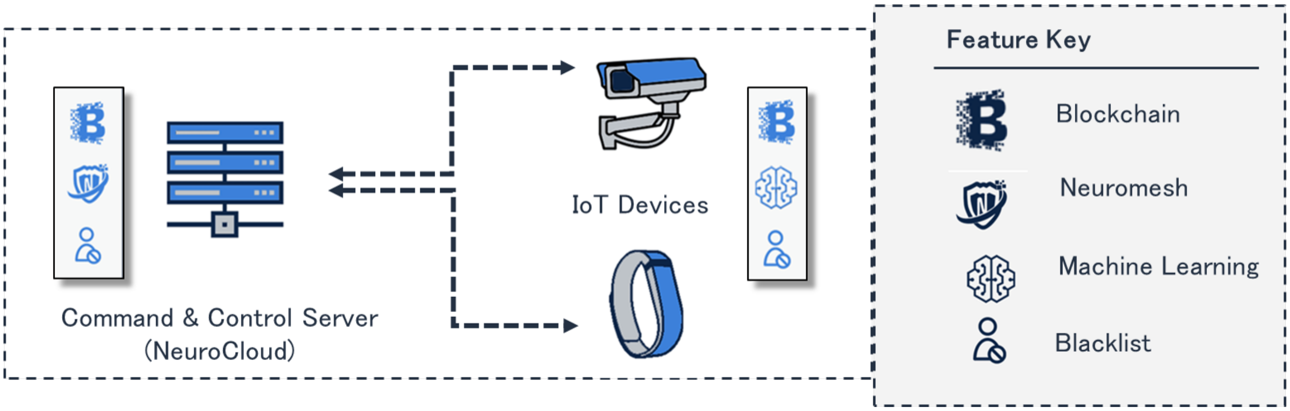

Figure 1: The NeuroMesh Solution Architecture

ABSTRACT

Internet-of-Things (IoT) devices are ubiquitous and growing rapidly in number. However, IoT manufacturers have focused on the functionality and features of the devices and made security an afterthought. Since IoT devices have small memory capacities and low-power processors, many security firms have not been able to develop anti-malware software for these devices. Current IoT security solutions are heavy and unreliable. We have developed a lightweight IoT security solution that uses hacker tools against the hackers – in essence, a vaccine for IoT. Our software provides managed security and intelligence to IoT devices using a "friendly" botnet operated through a proven, existing communication infrastructure for distributed systems – the Bitcoin blockchain.

CCS CONCEPTS

• **Security and privacy** → *Distributed systems security*; • **Computer systems organization** → *Embedded software*.

KEYWORDS

IoT Security, Botnet, Mirai, Blockchain, Bitcoin, Embedded System Security, IoT Device Management, Software Vaccine, Machine Learning, Security Architecture

ACM Reference Format:
Gregory Falco, Caleb Li, Pavel Fedorov, Carlos Caldera, Rahul Arora, and Kelly Jackson. 2019. NeuroMesh: IoT Security Enabled by a Blockchain Powered Botnet Vaccine. In *INTERNATIONAL CONFERENCE ON OMNI-LAYER INTELLIGENT SYSTEMS (COINS), May 5–7, 2019, Crete, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3312614.3312615

1 INTRODUCTION

The proliferation and widespread use of internet connected devices, also known as the internet of things (IoT) is increasing. Gartner predicts that, by 2020, the number of IoT devices will reach 20.4 billion on a global level.[1] Technological advancements such as

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
COINS, May 5–7, 2019, Crete, Greece
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6640-3/19/05.
https://doi.org/10.1145/3312614.3312615

smart cars, smart cities, and smart homes are a macro extension of the IoT era.[2] Tech giants Google and Apple, as well as several major automotive industry behemoths (e.g., General Motors, Honda, Ford, Toyota, etc.), are moving forward with plans to create fully automated self-driving cars.[3]

Amazon's Echo and Google's Home devices allow for syncing with other IoT devices within our homes. Beyond merely reporting the weather or playing our favorite music, each iteration of these "help bots" allows for increasing centralized functions including monitoring home security, controlling the internal temperature, and locking or unlocking home entry points. These devices are always listening and sending or receiving data from their requisite "cloud-based" database management systems.[4]

Additionally, all things IoT are used for several processes involved in medical/healthcare. For example, wearable devices monitor vital signs, remind patients the timing and dosage of their medications, and generate crucial clinical data which medical professionals then use to make accurate diagnoses.[5] IoT devices are also deployed for the industrial sectors. Sensors and other cameras assist in the predictive maintenance of equipment, smart metering for measuring energy, water, or natural gas, asset tracking, and fleet management systems.[6]

However, as will be discussed, the world of IoT faces a cybersecurity crisis which has become evident through the escalation of security breaches over the past five years.[7]

2 IOT SECURITY ISSUES

In October of 2016, hackers launched a large-scale DDoS attack using a botnet and leveraging Linux based IoT devices that had been infected with the Mirai malware.[8] Three waves of attacks disrupted popular websites including Amazon, GitHub, Slack, Visa, and HBO. This demonstrates one of many attacks against IoT.

IoT devices are easily hacked. In 2017, an 11-year-old boy demonstrated how he could hack into conference attendees Bluetooth devices using a raspberry pi. Subsequently, the young penetration tester also gained access to an internet connected teddy bear and recorded a message.[9] Though this was a benign instance, prison security control systems, heart monitors, insulin pumps, nuclear power plants, and oil pipelines are exposed through an "open source messaging protocol known as MQTT."[10] However, the protocol itself isn't necessarily the source of the issue; lack of proper device security provisioning, even as simple as implementing a username and password, is inconsistent. Hackers have also successfully entered airline systems such as a Polish airline's ground system and Vietnam's Noi Bai and Tan Son Nhat airports.[11] It's only a matter of time before cyberattacks wreak widespread havoc through gaining control of prison systems, airline control systems, and a myriad of other vulnerable IoT devices which have drastic consequences.

2.1 WHERE CURRENT IoT SYSTEMS FAIL

Due to the wide variation in IoT device firmware, and how IoT devices use open-source Embedded Linux operating system, IoT endpoints have a plethora of cybersecurity vulnerabilities. As the number of devices and their use cases grow, so do the different IoT security standards.[12] Though there are current attempts at defining a widely deployed IoT security standard, none have been pervasively adopted.[13][14] The lack of generally accepted security standards, which is partially a function of the diverse array of devices, also leaves a security gap that hackers use to gain control over the device and, eventually, the systems to which those devices are connected.

i. Embedded Operating Systems

Given the variety of IoT device manufacturers, there is also a variation in processors and device types that manage memory, data, and storage in different ways. However, Linux is the dominant operating system used for mobile operating and embedded systems. The widespread Linux adoption is predicated on several positive features including a stable OS kernel, low cost, and widespread community support due to its status as an open source OS.

Linux's popularity isn't limited to the honest and earnest use of individual programmers and commercial large-scale IoT deployments. Hackers like to use Linux because of its relative ease of customization for potentially nefarious purposes.[15] Despite Linux's benefits, a flaw in its TCP allowed hackers to inject malware by hijacking internet traffic.[16] BIOS password bypassing, data network security weaknesses, faking device entities, DNS spoofing[17] and leveraging Linux security tools[18] are additional methods hackers use to exploit the Linux system.

ii. Routers and Switches

Eighty million dollars were stolen from the central bank of Bangladesh in 2016. Other than the hackers and their illicit intentions, the primary culprit was the implementation of second hand, $10 routers which did not have an appropriate firewall in place or any file system security.[19] This was not an isolated incident. In 2018, the Department of Homeland Security (DHS), the Federal Bureau of Investigation (FBI), and the United Kingdom National Cyber Security Centre (NCSC) jointly announced that Russian attackers were hacking into routers to gain access to critical infrastructure. [20] Most routers and switches are easily hackable as there are several common security vulnerabilities found across various makes and models.[21] Routers are always "on" and perpetually connected to the internet while using open-source firmware with hardcoded default usernames and passwords. Many routers also use a Universal Plug-in-Play (UPnP) protocol that allows devices to automatically open ports for data transfer. These vulnerabilities provide easy accessibility for hackers to take control, install malware, and launch a DDoS attack.

iii. Hardware Network Monitoring

In an attempt to prevent cyberattacks, many enterprises and consumers have resorted to installing hardware devices that can monitor network traffic. The devices analyze packets flowing through the network and flag potential abnormal activity. Should abnormal activity surface, security analysts will try to determine if the source of the disruption is an intruder. If the activity is indeed an attempted attack, the analyst will blacklist the intruder's internet protocol (IP) address thus cutting off further access.

However, hackers can launch a zero-day attack that can defeat the hardware firewall in place, or use other means to bypass firewall rules. Furthermore, while IP blacklisting does inhibit further communication between a compromised device and the command and control server, hardware firewalls do not prevent the installation of malicious code on a vulnerable device after a hacker gains access. Additionally, network monitoring cannot quickly kill the malicious code once it has been installed on the vulnerable node.

iv. IoT Endpoint Security Software Solutions

There are several security measures available such as certificates, public and private keys, digital signatures, randomly generated device passwords, and IoT platforms. Each of these implementations is also susceptible to cyberattacks. Though the public and private key exchanges are robust against a direct brute force attack[22] and each public key is verified by a trusted certificate authority, they are much less effective and more cumbersome for two-way communication with IoT devices.[23] Each device would need its own public key and the key would need to be sent over the internet. Not only would this increase the costs associated with manufacturing the device, but increases the risk of a man-in-the-middle exploit.

Randomly generated device passwords applied at the time of device manufacturing is another method for increased IoT endpoint security. The main limitation of this method is the feasibility of communicating all of the device passwords from the disparate devices on the manufacturing line to a central server. As such, there is no automated way to access these devices, on boot, from a centralized server and the passwords are vulnerable to brute force attacks[24] [25] which allows an attacker admin permissions.

IoT platforms are the customized operating systems that enable end-users to securely collect data from IoT devices and to perform analytics on the data.[26] IoT platforms do not provide robust anti-malware protection. Platforms have been susceptible to code exploits[27] and to malicious apps that allowed intruders to access the device and to install botnets and ransomware[28]. Platform security can fail when a device on an IoT platform must communicate with a device outside of the IoT platform. While data may be encrypted within a platform providing security and privacy therein, as soon as the device handshakes with an external node, the entire platform becomes vulnerable.

2.2 Botnets

The notorious Mirai botnet attack in October of 2016 exemplifies how botnets can be used to infect IoT devices and launch a large-scale DDoS attack. Subsequent attacks have occurred through botnets, such as WannaCry,[29] WireX,[30] and Hajime.[31] Consequently, botnets pose the most pressing and dangerous threat to IoT device security. The Mirai attack unfolded in a seven-step process:

(1) Mirai uncovers the default credentials of weakly configured IoT devices with 62 likely username and password pairs hardcoded into Mirai.
(2) The bot forwards device characteristics to the report server using a different port.
(3) Through the command and control (CnC) server the botmaster locates new targets by continuous communication with the report server and usually through Tor.
(4) Once the targets are found the botmaster deploys an infected command in the loader which holds all the required details such as hardware infrastructure and IP addresses.
(5) The loader then logs into the target device and directs it to download and implement the malware. Meanwhile, it also activates a self-protecting script that prevents other malware from entry, and Mirai can now communicate with the CnC server signaling that the device is available to receive attack commands.
(6) After a target server has been established for the attack, the botmaster commands the launch of an attack by issuing a simple command via the CnC server.
(7) Using one of the ten available attack variants, the bots begin attacking the target server.[32]

Botnets are difficult to guard against due to their high variability as they tend to evolve quickly. However, there is a botnet based security solution that can defend IoT devices against other botnets that we've developed using Mirai as the functional model, which is described further below.

3 THE NEUROMESH SOLUTION

We propose a multifaceted solution to IoT endpoint security. We have devised the NeuroMesh solution consisting of the core components of NeuroNode endpoint protection, Rendezvous servers, NeuroCloud command and control (CnC) server and the NeuroPrime security operations center (SOC). NeuroMesh is named for its use of neural and mesh networks to secure IoT devices. The NeuroMesh solution proactively detects and removes IoT device malware, blacklists or whitelists IP based access control and enables secure communications and updates to IoT devices over the Bitcoin communication protocol.

3.1 Overview of the NeuroMesh Architecture

NeuroMesh's security features begin at the IoT endpoint. NeuroNode is installed on IoT devices with root access to the operating system or as a kernal module. Features of the NeuroNode can kill malware or malignant processes without network connectivity. There are both downlink and uplink communication channels from the NeuroNode. The NeuroNode can download new security commands from the NeuroCloud via the Bitcoin blockchain communication protocol. In addition, the NeuroNode can upload PCAP and log files to multiple rendezvous servers over a Diffie-Hellman communication protocol where the data is validated and then shared with the NeuroCloud. The NeuroCloud will house the master Bitcoin wallet from which security commands will be shared and also house the data for a set number of IoT devices that have been collected via the rendezvous servers. Machine learning algorithms will run against the data to determine new threats to the IoT devices. NeuroClouds will upload the newly discovered threats to NeuroPrime which then aggregates the threats and can share threats across various NeuroClouds. The threat information is then disseminated back down to the IoT devices where the NeuroNode acts on the threat intelligence. Each process is described in depth below.

3.2 The NeuroMesh Botnet Communication Protocol

Botnets have been very successful at infiltrating and controlling remote IoT devices as was seen with the Mirai attack. Therefore, we have modeled the NeuroMesh system based on an IoT botnet, Mirai - hence our "vaccine" for IoT. A challenge with using the traditional botnet architecture is that Mirai and other botnets have a central point of failure. This is because the command center is centralized on an Internet Relay Chat (IRC) server or HTTP server. One of the risks is if the attacker discovers the IRC server, the whole botnet could be compromised and taken over.[33]

Thus, one of the more secure communication protocols for botnets rely on Peer-to-Peer infrastructure. One of the implementations is the the kademlia protocol.[34] This involves using a distributed hash table (DHT) and P2P bot communication to disseminate commands. A vulnerability of using Kademlia is that if a bot is taken over, an attacker would be able to access the DHT which contains information on all other bots that are part of the botnet. With access to the routing tables, the attacker can poison the routing tables.

A conceptual paper called ZombieCoin proposes a botnet operated over the Bitcoin blockchain.[35] This concept breaks down for IoT devices because of the extensive memory capacity required for a Bitcoin node and the processing power that would be required to function is unrealistic on an IoT device. Currently, the Bitcoin blockchain is one of the most secure communication protocols that relies on Proof of Work (PoW) to verify transactions and is a trustless protocol providing pseudo-anonymity. Bitcoin has an immense amount of distributed processing power behind it and to date has been impossible to hack. NeuroMesh builds on the ZombieCoin work by establishing an IoT version of this so that we can use the Bitcoin blockchain to disseminate commands out to the IoT bots (NeuroNodes) while maintaining the security and integrity of the Bitcoin blockchain.

3.3 NeuroNode: Endpoint Security

NeuroNode consists of a 32kB bot-like bash script and a 1MB simplified payment verification (SPV) Bitcoin blockchain node from which the bash script reads and executes security commands. The bash script contains pre-stocked IoT malware signatures. Once downloaded and installed onto the device either via a direct firmware flash or cloud-based deployment, e.g. AWS Green Grass, NeuroNode will scan for these malware signatures and destroy them. Subsequently, each NeuroNode will establish its own unique system monitor for a given device. The system monitor will assume that the processes currently running at the time of installation and post malware destruction are sanctioned and will communicate with NeuroMesh that said processes are to be whitelisted. Furthermore, the system monitor will continuously scan the device to determine if any errant processes are running. Should a process outside the whitelist (sanctioned processes) arise, the NeuroNode will send a SIGKILL signal to the unauthorized process. The system monitor functions without any network connectivity, which is particularly appealing for industrial internet of things devices such as SCADA systems that may be in remote areas and lose network connectivity.

NeuroNode will conceal the totality of its processes through distribution in various parts of the file system. This hides the system monitor process from other potential SIGKILL commands orchestrated by malware. Though hiding said processes doesn't fully ensure NeuroNode processes will not be found, this concealment allows time for the NeuroNode to kill off malware using the predetermined signatures as well as the process whitelisting feature. Additionally, the system monitor will evaluate existing ports in use and then block all non-essential ports via binding a process to the unused ports. This activity blocks malware access to the ports thus preventing potential threats. To guarantee an adversary does not compromise the process whitelist, all updates to the whitelist after initial safe process enumeration must be added via the secure blockchain communication protocol. This provides functionality that enables dual-factor authentication over the Bitcoin blockchain. For any process to run that is not on the process whitelist, a second factor communication would need to be received via the blockchain confirming the process. Malware signatures can also be added via the blockchain communication protocol. Because of the root privileges of the NeuroNode, an entire firmware update can be made over the blockchain communication protocol.

Because of space and processing constraints, the SPV node is used. We have successfully shrunk the SPV node to 1MB. Using a Bloom filter, each endpoint's SPV node scans the Bitcoin blockchain for our master Bitcoin public addresses from which communications are sent (i.e. blacklisted IP addresses, processes, malware signatures, etc.). All commands from the blockchain are then executed by the NeuroNode's bash script.

i. Upstream transmission of network traffic

To analyze network traffic for each of the NeuroNodes, NeuroNodes will transmit PCAP files to three or more different rendezvous servers. The rendezvous servers are interconnected to form a mesh network with the IoT devices and NeuroCloud. Rendezvous servers are used to prevent NeuroNodes from discovering the CnC server that analyzes the network traffic in case a NeuroNode is compromised.

When data such as a PCAP file is sent to the rendezvous servers, there will be a locker for the node on each rendezvous server that receives the file. A hash of the data file is then created and sent to the the CnC server (NeuroCloud). A consensus algorithm is then run against the various rendezvous hashes to determine if the data was compromised. If there is consensus across the hashes, the PCAP file be pulled from one of the rendezvous servers to the NeuroCloud and the lockers will be emptied. If there is a discrepancy across the rendezvous server hashes, the file will be marked as compromised and discarded from the lockers without being pulled to the NeuroCloud. The NeuroCloud should only pull valid PCAPs so that any analysis done across the PCAP files can be reliable for data integrity.

3.4 NeuroCloud: Threat Detection and Communication

NeuroCloud is the CnC server that stores and analyzes network traffic. It then posts security updates to the blockchain.

i. Endpoint Data Collection and Analysis

Data collection and analysis begins with the CnC aggregating traffic data from the rendezvous point and applying our proprietary neural network for anomalous pattern detection. Once our machine learning algorithm determines malicious IP addresses, the CnC has the option to automatically communicate the threat information over the blockchain communication protocol, thus informing all network devices about the threat. Alternatively, the CnC can alert security analysts to threats who can then manually add the IP address to the blockchain.

ii. Communicating Updates to IoT Devices via Blockchain

To maximize the security of communications between the CnC and IoT devices, we utilize the public Bitcoin blockchain as a communication protocol. This provides a mechanism for externalizing traditional internal data security costs by outsourcing guarantees of data integrity to the entire public Bitcoin blockchain infrastructure. We hash our security commands to the Bitcoin blockchain (the mechanics are described below). These commands may include blacklist XYZ IP or kill ABC malware signature. The choice to use the aforementioned blockchain model is due to its widespread implementation, and the Bitcoin blockchain has proven to be resistant to data breaches. Additionally, attempting to compromise a verified transaction would require a 51% of the compute power on the entire Bitcoin network to manipulate consensus. This would be a computational feat that even a large government system would have trouble executing due to the widespread decentralization of the processing power. Moreover, should a single bot become compromised, it is impossible for a hacker to determine the location of other bots on the system because of the pseudo anonymity of the Bitcoin blockchain. Neither could a hacker determine the location of the NeuroCloud.

iii. NeuroMesh Blockchain Mechanics

Within the NeuroCloud resides two Bitcoin wallets. When a new message is created, it triggers a series of events prior to being posted to the Bitcoin blockchain. If the new message is created from wallet number one's address, it will pass Bitcoins to wallet number two for the security command to post. For example, if there is 10 BTC in wallet one, it will pass 9.9999325 BTC to wallet two. The difference between the two, 0.0000675 BTC, will be sent to the miners who will confirm the transaction. The miner fee changes based on the demand for miner services and the block space required for a transaction. Within the transaction data message, there is a generic line intended for metadata called op_return. The security command we wish to transmit will be placed in this part of the transaction. Op_return is limited to only 80 bytes in size. These wallets operate on a known address range.

The nodes will search for the NeuroCloud's wallet's public key. When looking at the transaction log of the NeuroCloud, each node will update its system according to the security command. For example, if a blacklisted IP or whitelisted IP was sent, IP tables would be updated accordingly. If the NeuroCloud communication is meant to deliver dual-factor authentication of a process, the process ID is shared over the Bitcoin blockchain. The endpoint system monitor and associated process whitelist previously described will be updated accordingly via the bash script that pulls the transaction from the blockchain and implements the security command. All security commands are posted in hexadecimal to the blockchain.

iv. Building a master blacklist or whitelist

We are able to build a master blacklist or whitelist aggregating blacklisted or whitelisted data over time using the Bitcoin communication protocol. In this scheme, the nodes will need to parse and pull together all previous transactions from our NeuroCloud address from multiple transaction blocks.

Data blocks will contain a header number to identity how many blocks need to be found and pulled together. For instance, if we've sent out 15 data blocks, the latest message posted to the blockchain will contain the number 15, indicating to the NeuroNode's bash script to go back into the transaction history to find the previous messages. The bash script will parse the Bitcoin transaction history to find all requested data messages, then combine the information to build a whitelist and blacklist or other pertinent notification. An entire firmware update could be delivered and received in the same fashion over multiple transactions.

3.5 NeuroPrime: The Security Operation Center

Our current architecture is configured such that a new NeuroCloud is established with associated rendezvous servers per company using NeuroMesh. Clients can choose to share threats discovered in their IoT devices by sharing the machine learning-generated blacklisted IP addresses with other organizations. In this case, the NeuroCloud for one client would send validated blacklisted IP addresses to our NeuroPrime server. NeuroPrime will cross-check the blacklisted IP addresses with other NeuroCloud client data. Blacklisted IPs that are found across NeuroClouds will be pushed back to all participating clients and NeuroPrime will instruct the respective NeuroClouds to hash the confirmed universal threats in the form of blacklisted IP addresses to their NeuroNodes over the blockchain communication protocol. This provides a global threat sharing and intelligence mechanism for all devices regardless of their owner and manufacturer.

4 FUTURE WORK

NeuroMesh has been tested on routers, CCTV cameras and smart meters to date. In all cases, NeuroMesh was able to kill the Mirai botnet and prevent errant processes from running. Further, any IP address sent via the blockchain communication channel was effectively blacklisted from access to the device. Proposed future work includes testing the NeuroMesh solution at scale. To date, testing has only been on individual devices and not on entire networks of systems which require different load balancing algorithms for highly distributed computing. We are partnering with a major Spanish utility, Iberdrola, and their US subsidiary, Avangrid, to test NeuroMesh in a realistic environment. However, it will likely take over a year to begin such a project and generate public results considering the substantial market barriers to installing such a security solution at scale. Namely, even though the utility may own their devices, the manufacturers of the IoT still have control over what is installed on them and the utility does not have root access. We believe, along with our utility partner, that the routing infrastructure for advanced metering infrastructure (AMI) should be secured first considering cyberattacks on US critical infrastructure have been linked to router and switch flaws[36].

As we work through the market barriers of testing NeuroMesh at scale, we continue to build new technical capabilities for NeuroMesh. We are developing new machine learning techniques for anomaly detection beyond the neural network we developed. Our goal will be to conduct anomaly detection for IoT log files so that we can generate new malware signatures in addition to IP-based blacklists or whitelists. We are also working to reduce the size of our already small SPV Bitcoin node in order to fit on IoT devices with less than 1MB of memory. Finally, we are exploring alternative blockchain technologies that are as secure as the Bitcoin blockchain, but less financially expensive to use (by eliminating miner fees).

5 CONCLUSION

We believe that the NeuroMesh solution is a multifaceted and comprehensive approach that targets each of the IoT security weaknesses described herein. Our lightweight architecture provides endpoint security that is hard to circumvent. Furthermore, our system can be implemented on a wide variety of devices, including industrial IoT, where heightened security is required to prevent widespread malicious attacks within the most sensitive industries such healthcare, industrial manufacturing, energy delivery systems and autonomous vehicles.

ACKNOWLEDGMENTS

The authors would like to thank the following people: Raquel Mercado, Roberto Mariscal Melero, Javier Ponce Suarez, Howard Shrobe, Michael Siegel, Erin Winslow, Harold Moss, Ron Chaney, Frank Wang, Natalie Bartlett, Isabelle Phelps, Rei Wang and Sean Dalton.

REFERENCES

1. Gartner. (2017). Gartner Says 8.4 Billion Connected “Things” Will Be in Use in 2017. http://www.gartner.com/newsroom/id/3598917
2. Meola, A. (2016). Internet of Things devices, applications & examples. http://www.businessinsider.com/internet-of-things-devices-applications-examples-2016-8
3. Venturebeat.com. (2017). Self-driving car timeline for 11 top automakers. https://venturebeat.com/2017/06/04/self-driving-car-timeline-for-11-top-automakers/
4. Wired.com. (2017). Ok, House. Get Smart: Make the Most of Your AI Home Minions. https://www.wired.com/2017/06/guide-to-ai-artificial-intelligence-at-home/
5. Wired.com. (2017). Connected Medical Devices, Apps: Are They Leading the IoT Revolution âĂŞ or Vice Versa? https://www.wired.com/insights/2014/06/connected-medical-devices-apps-leading-iot-revolution-vice-versa/
6. Tracy, P. (2017). The top 5 industrial IoT use cases. https://www.ibm.com/blogs/internet-of-things/top-5-industrial-iot-use-cases/
7. Lord, N. (2017). The history of data breaches. Available at: https://digitalguardian.com/blog/history-data-breaches
8. Etherington, D., and Conger, K. (2016). Large DDoS attacks cause outages at Twitter, Spotify, and other sites. Available at: https://techcrunch.com/2016/10/21/many-sites-including-twitter-and-spotify-suffering-outage/
9. Lui, K. (2017). This 11-year-old just schooled cybersecurity experts by “weaponizing” a teddy bear. Available at: http://fortune.com/2017/05/17/reuben-paul-cybersecurity-hacking/
10. Whittaker, Z. (2017). Exposed IoT servers let hackers unlock prison cells, modify pacemakers. Available at: http://www.zdnet.com/article/exposed-servers-hack-prison-cells-alter-pacemakers/
11. Osborne, C. (2017). Our planes are now “big flying mobile devices” and top hacking targets. Available at: http://www.zdnet.com/article/planes-as-big-flying-mobile-devices-are-top-targets-for-hackers/
12. Cisco. (2017). Securing the Internet of Things: A proposed framework. Available at: https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html
13. Postscapes. (n.d.). IoT standards and protocols. Available at: https://www.postscapes.com/internet-of-things-protocols/
14. Corser, G. (2017). Internet of Things (IoT) security and best practices. IEE Internet Technology Policy Community White Paper. Available at: https://internetinitiative.ieee.org/images/files/resources/white$_{p}$apers/internet$_{o}$f$_{t}$hings$_{f}$eb2017.pdf
15. AbdulKhaleq, A.A. (2016). Multi-level Windows exploitation using Linux Operating System. Asian Journal of Natural and Applied Sciences, Vol. 5(2). Available at: http://www.ajsc.leena-luna.co.jp/AJSCPDFs/Vol.5(2)/AJSC2016(5.2-06).pdf
16. Khandelwal, S. (2016). A Linux TCP flaw allowed hackers to hijack internet traffic and inject malware remotely. Available at: http://thehackernews.com/2016/08/linux-tcp-packet-hacking.html
17. Nguyen, C. (2016). Linux OS security mechanisms and how to implement them. Bachelor’s Thesis, University of Applied Sciences, Helsinki. Available at: https://www.theseus.fi/bitstream/handle/10024/118462/Nguyen$_{C}$uong.pdf?sequence=1
18. AbdulKhaleq, A.A. (2016). Multi-level Windows exploitation using Linux Operating System. Asian Journal of Natural and Applied Sciences, Vol. 5(2). Available at: http://www.ajsc.leena-luna.co.jp/AJSCPDFs/Vol.5(2)/AJSC2016(5.2-06).pdf
19. Condliffe, J. An $80M bank hack as been blamed on $10 routers. Available at: http://gizmodo.com/an-80m-bank-hack-has-been-blamed-on-10-routers-1772442595
20. US-CERT. Alert (TA18-106A) Russian State-Sponsored Cyber Actors Targeting Network Infrastructure Devices. Available at: https://www.us-cert.gov/ncas/alerts/TA18-106A
21. Whittaker, Z. (2017). Hundreds of Cisco switches vulnerable to flaw found in WikiLeaks files. Available at: http://www.zdnet.com/article/cisco-warns-of-critical-security-flaw-found-buried-in-wikileaks-vault-7-disclosure/
22. Blumenthal, M. (2007). Encryption: Strengths and weaknesses of public-key cryptography. Available at: http://www.csc.villanova.edu/~mdamian/Past/csc3990fa08/csrs2007/01-pp1-7-MattBlumenthal.pdf
23. Fisher, R., Lyu, M., Cheng, B., and Hancke, G. (2016). Public key cryptography: Feasible for security in modern personal area sensor networks? Available at: http://blueyetechnologies.co.in/wp-content/uploads/2017/08/Public-Key-Cryptography-Feasible-for-Security-in-Modern-Personal-Area-Sensor-Networks.pdf
24. Wollaston, V. (2013). Think you have a strong password? Hackers crack 16-character passwords in less than an hour. Available at: http://www.dailymail.co.uk/sciencetech/article-2331984/Think-strong-password-Hackers-crack-16-character-passwords-hour.html
25. Infosec Institute. (2017). Popular tools for brute-force attacks [Updated for 2017]. Available at: http://resources.infosecinstitute.com/popular-tools-for-brute-force-attacks/#gref
26. Scully, P. (2016). Insights from ongoing research on IoT platforms. Available at: https://iot-analytics.com/5-things-know-about-iot-platform/
27. Greenberg, A. (2017). Hack brief: “Devils Ivy” vulnerability cold afflict millions of IoT devices. Available at: https://www.wired.com/story/devils-ivy-iot-vulnerability/
28. IoTforAll. (2017). The 5 worst examples of IoT hacking and vulnerabilities in recorded history. Available at: https://www.iotforall.com/5-worst-iot-hacking-vulnerabilities/
29. Greenberg, A. (2017). Hackers are trying to reignite WannaCry with nonstop botnet attacks. Available at: https://www.wired.com/2017/05/wannacry-ransomware-DDoS-attack/
30. Kumar, M. WireX DDoS botnet: An army of thousands of hacked android smartphones. Available at: https://thehackernews.com/2017/08/android-DDoS-botnet.html
31. Khandelwal, S. Hajime “Vigilante Botnet” growing rapidly; hijacks 30,000 IoT devices worldwide. Available at: https://thehackernews.com/2017/04/vigilante-hacker-iot-botnet$_{2}$6.html
32. Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, Vol. 50, Issue 7.
33. Bezut, R. and Rollande, V.B. (2010). Experimental study: Study of dictionary attacks on SSH. Available at: https://files.xdec.net/TX$_{E}$N$_{B}$ezut$_{B}$ernet-Rollande$_{B}$ruteForce$_{S}$SH.pdf
34. Maymounkov, P., and Mazieres, D. (n.d.). Kademlia: A peer-to-peer information system based on the XOR metric. Available at: http://www.scs.stanford.edu/~dm/home/papers/kpos.pdf
35. Ali, Syed Taha, et al. "ZombieCoin: powering next-generation botnets with Bitcoin." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2015. Available at: http://homepages.cs.ncl.ac.uk/feng.hao/files/zcoin-camera-ready.pdf
36. Cimpanu, Catalin. "Cyber-Attacks On US Critical Infrastructure Linked To Cisco Switch Flaw." BleepingComputer. 2018. Available At: https://www.bleepingcomputer.com/news/security/cyber-attacks-on-us-critical-infrastructure-linked-to-cisco-switch-flaw/